Techniques d'intervalles pour la résolution de systèmes d'équations

Gilles Chabert

UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS - UFR Sciences
École Doctorale STIC

THÈSE

pour obtenir le titre de
Docteur en Sciences
de l'Université de Nice-Sophia Antipolis

Discipline : Informatique

présentée et soutenue par
Gilles CHABERT

Techniques d'intervalles pour la résolution de systèmes d'équations

Thèse dirigée par Gilles Trombettoni
et préparée à l'INRIA Sophia-Antipolis, projet COPRIN

Soutenue le 19 janvier 2007

Jury :

M.	**Jaulin, Luc**	Professeur à l'ENSIETA, Brest	Président, Examinateur
M.	**Frommer, Andreas**	Professeur à l'Université de Wuppertal	Rapporteur
M.	**Granvilliers, Laurent**	Professeur à l'Université de Nantes	Rapporteur
M.	**Van Hentenryck, Pascal**	Professeur à l'Université Brown	Rapporteur (non membre)
M.	**Daney, David**	Chargé de recherche à l'INRIA	Examinateur
M.	**Lhomme, Olivier**	Ingénieur R&D, ILOG	Examinateur
M.	**Trombettoni, Gilles**	Maître de Conférences à l'Université de Nice	Directeur

Remerciements

Je souhaite tout d'abord manifester ma plus grande gratitude envers Jean-Pierre Merlet, pour m'avoir accueilli dans son équipe et m'avoir offert des conditions de travail exceptionnelles tout au long de ces années.

Je tiens à remercier Luc Jaulin, Andreas Frommer, Laurent Granvilliers, Pascal Van Hentenryck, Olivier Lhomme et David Daney d'avoir accepté d'être rapporteurs ou membres du jury, avec la charge que cela implique.

Un grand merci à Gilles Trombettoni pour avoir encadré cette thèse, pour m'avoir formé sur de nombreux plans et pour ses grandes qualités de *manager* telles que son sens de l'écoute et son optimisme. Merci également à Bertrand Neveu pour sa disponibilité légendaire, son esprit critique, et la peine qu'il s'est donnée pour relire avec minutie des preuves rébarbatives.

Je voudrais ensuite exprimer ma reconnaissance envers les personnes qui ont contribué de façon significative à ma compréhension de l'analyse par intervalles. Je pense tout particulièrement à David Daney, Alexandre Goldsztejn et Marc Gouttefarde. Nos échanges m'ont beaucoup apporté.

Je suis également heureux d'aboir pu collaborer avec les autres membres du projet Coprin, et en particulier avec Carlos Grandón.

Ma famille a joué, souvent sans le savoir, un rôle considérable dans l'aboutissement de ce travail : merci à mes parents, à Daniel, Edith, Etienne, Hugues. Une pensée va également à mes amis et notamment à mes vieux complices : Pascal Bourgoin et Ludovic Cressy, toujours présents. Merci aussi à ceux qui ont assisté à ma soutenance et qui parfois sont venus de loin (Taï, Stéphanie, Antoine, Marie-Caroline).

Enfin, et par dessus tout, je souhaite remercier Maud de tout mon coeur pour sa tendresse et le soutien qu'elle m'a toujours apporté. Cette thèse, et pas seulement, lui doit beaucoup.

Table des matières

Chapitre 1

Introduction

Cette thèse propose quelques avancées sur les techniques d'intervalles pour la résolution de systèmes de contraintes numériques (équations ou inéquations non linéaires). Le terme de *techniques d'intervalles* regroupe des méthodes issues de domaines différents : l'analyse par intervalles, les intervalles modaux et la programmation par contraintes.

En analyse par intervalles, nous nous intéressons à la prise en compte de quantificateurs dans les systèmes paramétrés. Les algorithmes par intervalles proposés pour de tels systèmes peuvent être vus, notamment dans le cas linéaire, comme une extension des algorithmes classiques (pour les systèmes sans paramètre). L'objectif est de poursuivre ce travail d'extension.

Nous nous penchons également sur les intervalles modaux. Le but est de mettre au point une nouvelle approche de la théorie qui permette d'en faciliter la formulation. Il s'agit donc d'un travail de synthèse, destiné à être le plus simple possible.

En programmation par contraintes, nous avons cherché à exploiter les unions d'intervalles comme alternative possible aux intervalles pour la représentation des domaines des variables. Le but est d'étudier la possibilité d'obtenir, à l'aide de cette nouvelle représentation, la propriété d'arc-cohérence bien connue pour des domaines discrets mais difficilement adaptable aux domaines continus. L'ambition sous-jacente est de mettre au point de nouvelles cohérences partielles avec des unions d'intervalles.

Nous proposons d'introduire les thèmes développés ainsi que les diverses contributions à partir d'un exemple. Considérons le problème représenté à la figure 1.1. Il s'agit d'un robot parallèle (plan) à deux jambes. La première jambe est attachée dans le plan en un point $a = (a_1, a_2)$ fixé, la seconde en un point $b = (b_1, b_2)$. Les deux jambes peuvent tourner librement autour de leur point d'attache respectif. Les deux autres extrémités des jambes sont reliées en un point appelé *organe terminal du robot*, de coordonnées $x = (x_1, x_2)$. L'organe terminal est déplacé dans le plan en agissant sur la longueur des jambes (par exemple, avec des vérins).

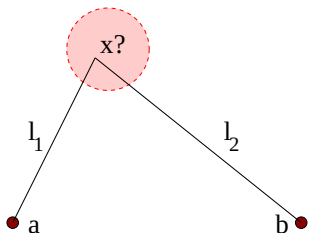


FIG. *1.1: Robot parallèle à deux jambes*

Le problème géométrique direct consiste à déterminer la position x de l'organe terminal étant donné une *commande* l, c'est à dire des longueurs l_1 et l_2 fixées pour les jambes. Le problème est modélisé par un système d'équations non linéaires, en l'occurrence deux équations de distance :

$$f(x) = 0 \iff \begin{cases} (x_1 - a_1)^2 + (x_2 - a_2)^2 = l_1^2, \\ (x_1 - b_1)^2 + (x_2 - b_2)^2 = l_2^2. \end{cases} \tag{1.1}$$

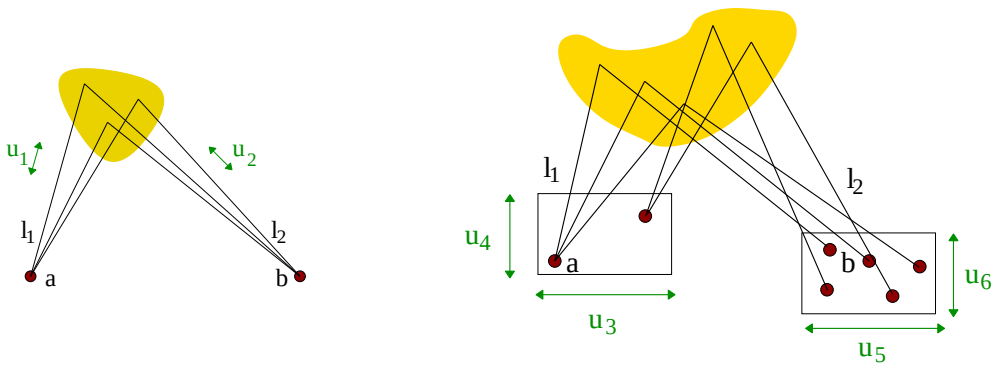
Les techniques d'intervalles permettent de résoudre de façon *fiable* de tels systèmes. La fiabilité est double : d'une part, aucune solution n'échappe à la résolution ; d'autre part, chaque solution est donnée sous la forme d'un volume de précision arbitraire auquel il est certifié qu'elle appartient.

Remarquons que le problème choisi ici est facile, et qu'il existe de nombreuses méthodes (notamment algébriques) qui permettent de le résoudre exactement. Néanmoins, les méthodes d'intervalles s'appliqueront pour un large éventail de systèmes d'équations, y compris ceux impliquant des fonctions trigonométriques pour lesquelles il n'existe pas de méthode générale.

1.1 Paramètres et quantificateurs

1.1.1 Incertitudes

Reprenons l'exemple du robot parallèle. En pratique, la longueur des jambes n'est obtenue qu'avec une certaine précision. Il en va de même des coordonnées des points d'attache. Cela signifie qu'une solution de (1.1) est certes certifiée, mais vis-à-vis d'un modèle qui traduit une situation idéalisée, donc approximative. C'est pourquoi on comprend l'intérêt de prendre en compte les perturbations d'un système, c'est à dire les variations non contrôlables des "entrées". Ces perturbations sont des données qu'il faut intégrer dans le modèle sous forme de paramètres. A chaque paramètre u_i est associé un intervalle $\mathbf{u}_i$ centré sur 0 représentant son domaine de variation possible. La figure 1.2.(a) représente des perturbations $u_1 \in \mathbf{u}_1$ et $u_2 \in \mathbf{u}_2$ sur la longueur de chacune des jambes. La figure 1.2.(b) représente des perturbations $u_3 \in \mathbf{u}_3 \dots u_6 \in \mathbf{u}_6$ sur les coordonnées des points d'attache.



(a) Perturbations sur la commande (b) Perturbations sur les points d'attache

FIG. *1.2: Exemples de perturbations.*

Bien entendu, il est possible de cumuler les deux types de perturbations. Le système d'équations s'écrit alors

$$f(x,u)=0 \iff \begin{cases} (x_1-(a_1+u_3))^2+(x_2-(a_2+u_4))^2=(l_1+u_1)^2, \\ (x_1-(b_1+u_5))^2+(x_2-(b_2+u_6))^2=(l_2+u_2)^2. \end{cases} \tag{1.2}$$

Remarquons que les perturbations permettent également de modéliser les constantes physiques irrationnelles ou non représentables par un flottant : ainsi une équation faisant intervenir la constante d'Euler peut être posée rigoureusement grâce à un paramètre e de domaine $[0.57721, 0.57722]$.

Ces paramètres intervalles modifient le problème posé. "Résoudre" un système ayant des paramètres intervalles peut avoir maintenant plusieurs significations. La signification classique est la suivante : on cherche une description des ensembles de solutions obtenus pour toutes les valeurs possibles prises par ces paramètres. Ce type de résolution répond également à une autre situation : celle où l'on souhaite observer les variations "en sortie" obtenues en fonction de variations (contrôlées) "en entrée" (par exemple, quelles sont les positions possibles de l'organe terminal si l'on déplace les points d'attaches ?). Pour cette raison, nous utilisons le terme de *paramètre* pour décrire tout type d'incertitude (que son origine soit une perturbation ou non).

Étant donné un vecteur de paramètres u et un vecteur d'intervalles $\mathbf{u}$ décrivant leurs domaines, le problème consiste désormais à trouver l'ensemble des vecteurs x vérifiant

$$\big(\exists u \in \mathbf{u}\big) \mid f(x,u)=0. \tag{1.3}$$

1.1.2 AE-systèmes

Nous avons introduit jusqu'ici des paramètres au problème géométrique direct. Est alors considéré comme solution à ce problème tout point pour lequel il *existe* une valeur des paramètres tel que le système d'équations soit satisfait. Ce type de résolution n'est pas toujours pertinent. Pour illustrer notre propos, prenons maintenant le problème géométrique inverse, qui consiste à déterminer une commande permettant de placer l'organe terminal à une position souhaitée[1]. En l'absence d'incertitude, le système d'équations est trivial. En effet, il s'agit du même système (1.1) où l devient la variable et x un paramètre fixé, sauf que les équations donnent cette fois directement l'expression de l en fonction de x. Il suffit donc d'une évaluation pour répondre au problème $f(l)=0$.

En présence de perturbations, le système se ramène de nouveau à (1.2), avec cette fois l pour variable. De même que précédemment, nous pouvons résoudre (1.2) en appliquant la définition (1.3), c'est à dire

$$\big(\exists u \in \mathbf{u}\big) \mid f(l,u)=0. \tag{1.4}$$

En extrapolant, imaginons que ce robot soit un robot chirurgical, de telle sorte que le problème inverse consiste à déterminer une commande permettant de placer le scalpel (l'organe terminal) à une position souhaitée. S'il y a des incertitudes par exemple sur les points d'attache, quelle information nous apporte la résolution de (1.4) sur la garantie qu'en injectant une solution dans les commandes, le scalpel ne touchera pas un organe vital ?

Selon la définition (1.4), une solution vérifie : *"il existe des perturbations dans* $\mathbf{u}$ *pour lesquelles la commande place l'organe terminal à la position* x *".* Il est net que cette propriété ne nous apporte rien. Notre but est plutôt d'obtenir une commande telle que "*quelles que soient les perturbations dans* $\mathbf{u}$*, la commande place l'organe terminal à la position* x*".* Mais en général, le problème posé ainsi n'a pas de solution : le scalpel ne peut rester exactement en x pour une longueur des jambes donnée si, par exemple, les points d'attaches bougent. Mathématiquement, cela signifie que $f(l,u)$ ne peut pas rester nul en faisant varier u.

Il apparaît donc le besoin de définir une borne sur l'erreur de la position, c'est à dire deux intervalles $\mathbf{v}_1$ et $\mathbf{v}_2$ tels que la boîte centrée en x et de dimension $\mathbf{v}_1 \times \mathbf{v}_2$ soit une *cible* contenant l'ensemble des positions acceptables pour le scalpel (voir figure 1.3.(a)). Le modèle adapté à cette situation fait donc intervenir deux paramètres supplémentaires, v_1 et v_2 :

[1] On ne s'intéresse pas ici au problème de trajectoire, c.a.d. à l'accessibilité de cette position depuis un état initial.

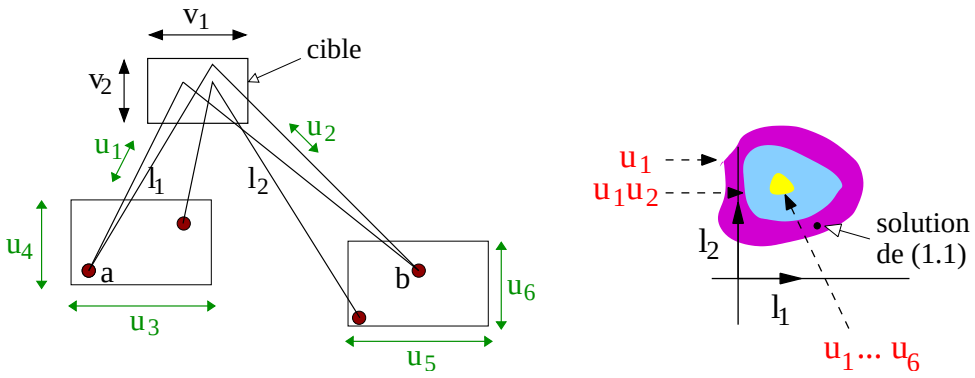
$$f(x,u,v)=0 \iff \begin{cases} ((x_1+v_1)-(a_1+u_3))^2+((x_2+v_2)-(a_2+u_4))^2=(l_1+u_1)^2, \\ ((x_1+v_1)-(b_1+u_5))^2+((x_2+v_2)-(b_2+u_6))^2=(l_2+u_2)^2. \end{cases} \tag{1.5}$$

Il est possible maintenant de reformuler le fait qu'une commande soit robuste aux perturbations ainsi : "*quelles que soient les coordonnées des points d'attache et les longueurs des jambes possibles, il existe un point dans la cible où la commande place l'organe terminal*". Plus généralement, les *AE-systèmes* permettent de modéliser les problèmes où interviennent ces deux types de paramètres :

- les paramètres quantifiés universellement (ici u) que l'on veut *contrôler*, c.a.d. dont les variations peuvent survenir sans rendre le système critique.
- les paramètres quantifiés existentiellement (ici v), qui servent à compenser le fait qu'un paramètre soit *contrôlé* en entrée par le fait qu'une erreur puisse être en contrepartie tolérée sur la sortie.

Finalement, le système à résoudre est

$$\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big) \quad f(x,u,v)=0. \tag{1.6}$$



(a) Perturbations et tolérance (b) AE-systèmes

FIG. *1.3: Exemple de paramètres quantifiés $\forall/\exists$. La figure (b) représente différents ensembles solution possibles pour (l_1,l_2), suivant le nombre de paramètres u_i pris en compte. Remarquons que la solution initiale de* (1.1) *peut ne plus être solution.*

Le problème décrit par la relation (1.6) s'appelle un **AE-système** [Sha99]. "A-E" signifie "All-Exists" car dans la formule (1.6), les paramètres universellement quantifiés précèdent les paramètres existentiellement quantifiés[2].

Insistons sur le fait que, sans hypothèse sur f, le système (1.6) est plus difficile à résoudre que (1.3). Ce n'est en fait pas le cas pour notre exemple didactique du robot à deux jambes, car le AE-système peut être ramené (par une reformulation non triviale) à un problème sans quantificateur. Mais à notre connaissance, il n'existe pas de transformation qui puisse être appliquée de façon systématique à un AE-système pour le mettre sous une forme « classique ».

1.1.3 Contributions

Dans le cas d'un système avec paramètres classique du type (1.3), la résolution d'un système non-linéaire est ramenée à celle d'un système linéaire par une linéarisation conservative appelée *méthode de Newton par*

[2] D'autres ordres entre quantificateurs seraient envisageables mais aucune méthode aujourd'hui n'appréhende de tels problèmes.

intervalles. L'un des apports de la théorie des *intervalles modaux* est d'avoir rendu cette linéarisation possible également pour les AE-systèmes : la résolution d'un AE-système non-linéaire peut aussi être ramenée à celle d'un AE-système linéaire. Pour traiter ce type de systèmes linéaires, la structure des *intervalles généralisés* (intervalles dont les bornes peuvent être inversés) semble alors parfaitement adaptée. Certaines méthodes par intervalles (Gauss-Seidel, Krawczyk) s'étendent naturellement aux AE-systèmes linéaires en remplaçant directement dans les calculs les intervalles par des intervalles généralisés. En revanche, d'autres méthodes comme celle de Hansen-Bliek ou de l'élimination de Gauss ne se prêtent pas aussi facilement à une telle extension. Dans les deux cas, aucune interprétation n'est obtenue en appliquant la méthode telle quelle avec des intervalles généralisés. Notre principale contribution est d'avoir trouvé, en dépit de ces difficultés, un moyen d'adapter ces méthodes aux AE-systèmes linéaires.

Voici la liste de nos contributions en analyse par intervalles (classiques et modaux) :

1. Extension de la méthode de Hansen-Bliek aux systèmes $\mathbf{A}x = \mathbf{b}$ où les paramètres du vecteur $\mathbf{b}$ peuvent être librement quantifiés. Cette méthode calcule l'enveloppe convexe des solutions du AE-système. L'extension change considérablement la complexité du problème car les points candidats pouvant former l'enveloppe convexe passent de 2 à $n+1$ sur chaque axe. Les AE-systèmes linéaires quantifiés « à droite » sont précisément ceux qui interviennent dans l'opérateur de Newton généralisé (proposée par Goldsztejn).
2. Nouvelle construction simplifiée de la théorie des intervalles modaux, basée sur la notion d'*image quantifiée*. La construction introduit l'arithmétique de Kaucher (intervalles généralisés) comme un moyen de calculer les images quantifiées des opérations et fonctions de base. Le théorème de Newton généralisé est redémontré dans ce formalisme.
3. Conception de la méthode LU généralisée, qui permet d'assurer l'égalité $\mathbf{A} = \mathbf{LU}$ au lieu d'une simple inclusion comme dans le cas classique. Cette décomposition peut être exploitée pour calculer des approximations intérieures ou extérieures de l'ensemble des solutions d'un AE-système linéaire, mais seulement sous certaines conditions.

1.2 Unions d'intervalles

1.2.1 Contraintes et projections

L'autre partie de notre travail porte sur la programmation par contraintes (sur les réels). Dans le cadre de la programmation par contraintes, les équations sont vues comme des contraintes entre les variables que l'on peut traiter indépendamment pour réduire l'espace de recherche. Le système (1.1) se prête particulièrement à ce point de vue puisque chaque contrainte (équation) possède une interprétation géométrique directe : elle correspond à une distance entre l'organe terminal et un point d'attache. Le système de contraintes est $\{c_1, c_2\}$ avec

$$\begin{aligned} c_1(x_1, x_2) &\iff (x_1 - a_1)^2 + (x_2 - a_2)^2 = l_1^2, \\ c_2(x_1, x_2) &\iff (x_1 - b_1)^2 + (x_2 - b_2)^2 = l_2^2. \end{aligned}$$

L'algorithme de base en programmation par contraintes consiste à *projeter* une contrainte sur une variable, c'est à dire à supprimer du domaine d'une variable les valeurs qui violent cette contrainte quelles que soient les valeurs prises par les autres variables dans leurs domaines respectifs. La figure 1.4 représente la projection de la contrainte c_1 sur x, pour deux boîtes initiales différentes. Les domaines de chaque variable forment une *boîte*.

Dans le cas de la figure 1.4.(a), la projection sur x rétrécie les bornes de son domaine. La réduction obtenue pour x peut ensuite servir à réduire le domaine de y en projetant l'autre contrainte, c_2.

Les réductions peuvent ainsi être propagées. Dans le cas où un point fixe est atteint, la propriété obtenue s'appelle l'*arc-cohérence*.

Dans le cas de la figure 1.4.(b), le domaine de x est coupé en deux : il devient une union d'intervalles. Si l'on décide d'autoriser les unions d'intervalles pour représenter les domaines, il s'avère que la propagation des

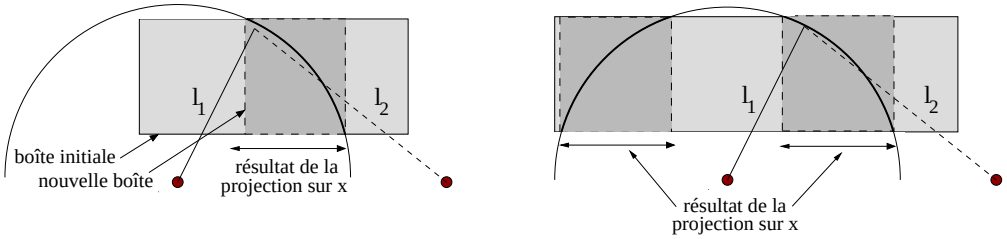


FIG. *1.4: Projection de la contrainte c_1 sur x.*

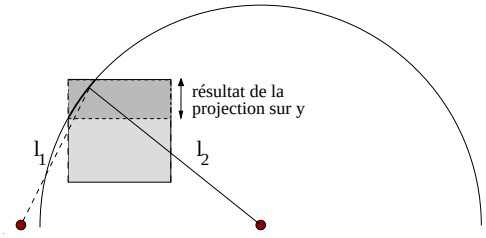


FIG. *1.5: Projection de la contrainte c_2 sur y.*

réductions peut alors faire croître la taille des unions de manière exponentielle. Le choix adopté en général est d'approximer une union par un simple intervalle, ce qui évite ce risque d'explosion tout en bénéficiant des réductions obtenues sur les bornes. La propriété qui en résulte s'appelle la *2B-cohérence*. Nous avons étudié une approche différente, qui consiste à autoriser les unions mais en calculant des réductions plus fortes, c'est à dire, impliquant plus d'information que la contrainte considérée.

1.2.2 Contributions

Voici la liste de nos contributions en programmation par contraintes :

1. Construction d'un cas où la taille de l'union nécessaire pour approximer l'arc-cohérence est de l'ordre du nombre de flottants représentant les domaines. Cet exemple exclut la possibilité d'obtenir une approximation de l'arc-cohérence avec des unions de tailles raisonnables et motive la recherche d'autres cohérences.

2. Mise au point d'un algorithme incorporant quelques étapes de bissection des domaines au cours du filtrage par arc-cohérence pour obtenir la propriété de *box-set cohérence*, qui donne une union de boîtes arc-cohérentes. Le nombre de boîtes produites est borné par une valeur ne dépendant que des équations, et non des domaines. L'algorithme a été amélioré pour donner lieu à une version paresseuse (*lazy boxset*), n'utilisant pas d'union dans les calculs intermédiaires grâce à la 2B-cohérence. Intégré dans une recherche de solutions, le surcoût qu'il introduit est négligeable.

3. Une méthode est enfin conçue pour obtenir l'arc-cohérence sans point de choix. La propriété obtenue, plus forte que l'arc-cohérence est nommée *I-global cohérence* (IGC). Elle garantit que les sous-domaines formés par les intervalles forment des cliques de supports entre eux. Grâce à une structure de données hybride (union d'intervalles et BDD) incrémentale, un ensemble de "cliques d'intervalles" est maintenue au cours de la propagation sans faire appel à un algorithme de graphe.

1.3 Organisation du document

Le chapitre 2 est une introduction à l'analyse par intervalles dite « classique » par opposition aux intervalles modaux. Les principales techniques (bissection, évaluation, filtrage, test d'existence et d'unicité) y sont décrites en détail.

Le chapitre 3 est une présentation des intervalles modaux. Il y est montré dans quelle mesure les techniques du chapitre 2 peuvent être adaptées aux AE-systèmes.

Notre contribution portant essentiellement sur les systèmes linéaires, le chapitre 4 propose un état de l'art sur les techniques d'intervalles pour les systèmes linéaires classiques et le chapitre 5 pour les AE-systèmes linéaires. Ce dernier incorpore également les nouvelles méthodes que nous proposons.

Le reste du document porte sur la programmation par contraintes. Le chapitre 6 propose un état de l'art du sous-domaine consacré aux contraintes sur des variables réelles.

Enfin, le chapitre 7 présente notre contribution concernant les cohérences liées aux unions d'intervalles.

Chapitre 2

Intervalles classiques

Sommaire

2.1 Introduction

L'analyse par intervalles est une branche de l'analyse numérique née dans les années 1960, qui possède ses propres méthodes. Le livre fondateur est sans doute celui de Moore [Moo66] (1966). D'autres ouvrages connus traitent de ce sujet [AH83, Neu90, Han92b, Kea96, JKDW01]. Bien qu'à l'origine Moore ait développé cette théorie dans le cadre de la résolution d'équations différentielles, l'objet d'étude depuis fut principalement la résolution de systèmes d'équations. Cette thèse est d'ailleurs focalisée uniquement sur cet axe.

L'analyse numérique classique propose bien entendu des méthodes pour résoudre des systèmes d'équations. Malheureusement, la solution fournie par ces méthodes est le résultat d'une série de calculs entachés d'imprécisions. Ces imprécisions peuvent être :
– des erreurs d'arrondi liées au calcul sur les flottants ;
– des imprécisions liées aux méthodes elles-mêmes, qui ne fournissent, même en théorie, qu'une approximation de la solution ;
– des incertitudes sur les paramètres. En effet, bien souvent le système que l'on résout est un modèle mathématique d'un problème physique. Or, les incertitudes inhérentes aux grandeurs physiques ne sont pas prises en compte dans le modèle. Ce dernier présente donc, dès le départ, un décalage avec la réalité.

On voit qu'il est donc nécessaire d'effectuer parallèlement aux calculs une analyse d'erreur. Il est possible, par exemple, d'utiliser un modèle probabiliste, pour connaître la distribution des sorties d'un système, étant donné une distribution des paramètres. Malheureusement, dans un contexte critique (robotique chirurgicale, contrôle

de centrales nucléaires, etc...), cela est insuffisant : on ne peut accepter qu'il y ait une probabilité, même très faible, qu'un accident se produise. Ajoutons que ces méthodes convergent vers *une* solution particulière. Elles ne conviennent pas pour décrire l'ensemble des solutions d'un système.

L'analyse par intervalles repose sur une approche différente, qui consiste en un mot à prendre en compte au niveau le plus "atomique", c'est à dire dans chaque calcul élémentaire, l'ensemble des imprécisions possibles, que l'on représente par des intervalles. Ainsi, chaque entité (variable, paramètre) se voit affecter un intervalle décrivant ses variations possibles (que l'origine de cette variation soit numérique, physique, etc...). Les calculs ne se font donc plus avec des opérandes réelles (flottantes), mais avec des opérandes de type intervalle.

Les calculs élémentaires produisent de nouveaux intervalles qui conservent les imprécisions obtenues jusqu'à ce stade, et y cumulent les nouvelles imprécisions introduites par le calcul lui-même. On parle de calcul conservatif. C'est cette propriété qui permet de garantir à l'issue de la méthode, qu'une solution reste dans un intervalle de tolérance. Combinée avec un découpage exhaustif des domaines, cette méthode ne perd aucune solution.

Ce chapitre présente les bases de l'analyse par intervalles. Il comporte tout d'abord une description de l'arithmétique (sans pour autant considérer l'aspect de *bas niveau* lié aux flottants), de ses propriétés de conservatisme et de *pessimisme* (surestimation des calculs). L'algorithme de base -de type *branch and prune*- pour la recherche de solution est donné. Des faiblesses de cet algorithme se dégagent alors trois axes fondamentaux de recherche : l'élaboration de meilleures évaluations, la mise au point d'algorithmes de filtrage et de preuve d'existence de solutions, et enfin la prise en compte des paramètres. Le chapitre se termine par une rapide comparaison entre l'approche probabiliste et la méthode par intervalles.

2.2 Notions de base

2.2.1 Intervalles

Commençons par donner quelques notations de base.

Pour tout couple de réels a et b vérifiant $a \leq b$, on appelle intervalle et on note $[a, b]$ l'ensemble suivant :

$$[a, b] := \{x \in \mathbb{R} \mid a \leq x \leq b\}$$

L'ensemble des intervalles sera noté $\mathbb{IR}$. Nous mettrons en gras les symboles qui désignent un élément de $\mathbb{IR}$ (ex : soit $\mathbf{x} \in \mathbb{IR}$, ...).

Dans un intervalle $[a, b]$, le réel a s'appelle la **borne inférieure**, b la **borne supérieure**. Si $\mathbf{x}$ est un intervalle, on note $\inf(\mathbf{x})$, ou $\underline{\mathbf{x}}$, la borne inférieure de $\mathbf{x}$, $\sup(\mathbf{x})$ ou $\overline{\mathbf{x}}$ la borne supérieure de $\mathbf{x}$.

Nous identifions l'intervalle $[a, a]$ au réel a. Un tel intervalle est dit **dégénéré**. Cette identification est possible car l'arithmétique d'intervalles que nous allons définir étend celle des nombres réels. Par exemple, si $\oplus$ est l'opérateur de somme défini pour des intervalles, on vérifie que la somme d'intervalles dégénérés coïncide avec la somme de réels :

$$\forall (a, b) \in \mathbb{R}^2 \qquad \mathbf{x} = [a, a] \oplus [b, b] \Longleftrightarrow \underline{\mathbf{x}} = \overline{\mathbf{x}} = a + b$$

Comme nous l'avons décrit en introduction, il y a souvent derrière un intervalle la volonté de représenter une valeur avec un certaine incertitude, ainsi en écrivant $\mathbf{pi} = [3.141, 3.142]$, on exprime l'idée que $\pi = 3.1415 \pm 0.0005$. La valeur 3.1415 s'appelle le milieu (*mid*) de l'intervalle **pi**, 0.0005 le rayon (*rad*). Ainsi :

$$\text{mid}\,(\mathbf{x}) := 1/2\,(\underline{\mathbf{x}} + \overline{\mathbf{x}})$$

$$\text{rad}\,(\mathbf{x}) := 1/2\,(\overline{\mathbf{x}} - \underline{\mathbf{x}})$$

Nous aurons aussi besoin de donner un nom à la plus petite valeur absolue d'un intervalle $\mathbf{x}$, c'est à dire à la valeur absolue du réel ayant la plus petite valeur absolue dans l'intervalle $\mathbf{x}$. Cette valeur s'appelle la **mignitude** de $\mathbf{x}$, et est notée $<\mathbf{x}>$. Elle correspond intuitivement au "poids" minimal de l'intervalle. On définit similairement la **magnitude** de $\mathbf{x}$, notée $|x|$.

$$<\mathbf{x}> := \inf_{x \in \mathbf{x}} |x|$$
$$|\mathbf{x}| := \sup_{x \in \mathbf{x}} |x|$$

On vérifie immédiatement que $<\mathbf{x}> = \min\{|\underline{\mathbf{x}}|, |\overline{\mathbf{x}}|\}$ si $0 \notin \mathbf{x}$, et que $<\mathbf{x}> = 0$ sinon. De même, $|\mathbf{x}| = \max\{|\underline{\mathbf{x}}|, |\overline{\mathbf{x}}|\}$.

Nous aurons recours à la relation d'inclusion entre deux intervalles, et à la relation d'infériorité entre intervalle et réel :

$$\begin{array}{ll}
\mathbf{x} \subseteq \mathbf{y} \iff \left(\underline{\mathbf{x}} \geq \underline{\mathbf{y}}\right) \wedge \left(\overline{\mathbf{x}} \leq \overline{\mathbf{y}}\right) & \text{(inclusion)} \\
\mathbf{x} \subset \mathbf{y} \iff \left(\underline{\mathbf{x}} > \underline{\mathbf{y}}\right) \wedge \left(\overline{\mathbf{x}} < \overline{\mathbf{y}}\right) & \text{(inclusion stricte)} \\
\mathbf{x} \leq \alpha \iff \overline{\mathbf{x}} \leq \alpha & \text{(infériorité)} \\
\mathbf{x} < \alpha \iff \overline{\mathbf{x}} < \alpha & \text{(infériorité stricte)}
\end{array}$$

2.2.2 Vecteurs et matrices intervalles

Soit $\mathbf{x}$ un vecteur d'intervalles de dimension n (on écrit $\mathbf{x} \in \mathbb{IR}^n$). On notera toujours $\mathbf{x}_i$ la $i^{ème}$ composante de $\mathbf{x}$.

Les définitions suivantes sont directement calquées sur le cas $n = 1$, en raisonnant composante par composante. Ainsi $\underline{\mathbf{x}}$, $\overline{\mathbf{x}}$, rad $(\mathbf{x})$, mid $(\mathbf{x})$, $<\mathbf{x}>$, et $|\mathbf{x}|$ sont six vecteurs de $\mathbb{R}^n$ tels que

$$\forall i \in [1..n] \qquad \begin{array}{lcl}
(\underline{\mathbf{x}})_i & = & \underline{\mathbf{x}_i} \\
(\overline{\mathbf{x}})_i & = & \overline{\mathbf{x}_i} \\
(\text{rad } \mathbf{x})_i & = & \text{rad } \mathbf{x}_i \\
(\text{mid } \mathbf{x})_i & = & \text{mid } \mathbf{x}_i \\
<\mathbf{x}>_i & = & <\mathbf{x}_i> \\
|\mathbf{x}|_i & = & |\mathbf{x}_i|
\end{array}$$

Les crochets sont un moyen commode de définir un intervalle à partir de ses deux bornes. Nous étendons cette notation aux vecteurs d'intervalles ; ainsi, pour $\mathbf{x} \in \mathbb{IR}^n$ on écrira $\mathbf{x} = [\underline{\mathbf{x}}, \overline{\mathbf{x}}]$.

L'inclusion entre vecteurs d'intervalles se décline de trois manières :

$$\begin{array}{ll}
\mathbf{x} \subseteq \mathbf{y} \iff \forall i \in [1..n]\ \mathbf{x}_i \subseteq \mathbf{y}_i & \text{(inclusion)} \\
\mathbf{x} \subset \mathbf{y} \iff \forall i \in [1..n]\ \mathbf{x}_i \subset \mathbf{y}_i & \text{(inclusion stricte)} \\
\mathbf{x} \subsetneq \mathbf{y} \iff (\mathbf{x} \subseteq \mathbf{y}) \wedge (\exists i \in [1..n]\ \mathbf{x}_i \subset \mathbf{y}_i) & \text{(inclusion sans égalité)}
\end{array}$$

Nous utiliserons également les opérateurs $<$, $\leq$, $>$ et $\geq$ entre vecteurs d'intervalles et vecteurs de réels (et notamment entre deux vecteurs de réels) :

$$\mathbf{x} < \alpha \iff \forall i \in [1..n]\ \mathbf{x}_i < \alpha_i.$$

Un vecteur d'intervalles, vu comme produit cartésien d'intervalles représente un parallélépipède n-dimensionnel. On utilisera souvent le terme commode de **boîte**[1].

Soit $\mathbf{A}$ une matrice intervalles. On notera toujours $\mathbf{A}_{ij}$ l'élément de $\mathbf{A}$ situé à la $i^{ème}$ ligne et à la $j^{ème}$ colonne. De nouveau, les définitions s'étendent aux matrices d'intervalles composante par composante, à l'exception de la mignitude $<\mathbf{A}>$, qui aura une définition particulière (voir chapitre 4). Par exemple, le rayon d'une matrice intervalle $\mathbf{A}$ est une matrice scalaire notée (rad $\mathbf{A}$) telle que $(\text{rad } \mathbf{A})_{ij} := \text{rad } (\mathbf{A}_{ij})$. Nous étendons également l'usage des crochets aux matrices : $\mathbf{A} = [\underline{\mathbf{A}}, \overline{\mathbf{A}}]$.

[1] On retrouve également dans la littérature française le terme *pavé*.

2.2.3 Types de boîtes

Pour finir, nous allons introduire quelques notions qui relient $\mathbb{IR}^n$ et l'ensemble des parties de $\mathbb{R}^n$ (noté $\mathcal{P}(\mathbb{R}^n)$). Pour tout ensemble D de vecteurs réels, ces notions servent à caractériser la "position" d'une boîte par rapport à D (contenue dans D ? contenant D ? intersectant D ?). Typiquement, D représentera soit l'ensemble de définition d'une fonction, soit l'ensemble des solutions d'un système d'équations.

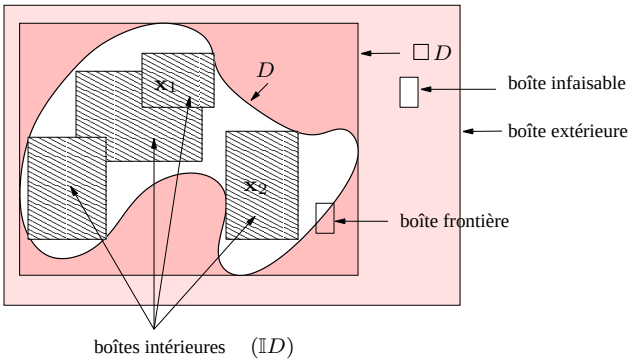


FIG. *2.1: Les différents types de boîtes (sur un exemple en dimension 2).*

Définition 2.1 (Types de boîtes) *Soit $D \in \mathcal{P}(\mathbb{R}^n)$, $\mathbf{x} \in \mathbb{IR}^n$.*
Vis-à vis de l'ensemble D, la boîte $\mathbf{x}$ est dite :
- **infaisable** *si* $\mathbf{x} \cap D = \emptyset$.
- **extérieure** *si* $D \subseteq \mathbf{x}$.
- **intérieure** *si* $\mathbf{x} \subseteq D$.
- **frontière** *si* $\mathbf{x}$ *n'est ni intérieure ni infaisable, et si* $\mathrm{rad}\,(\mathbf{x})$ *satisfait un critère de précision fixé par le contexte (ex :* $\forall i \in [1..n]\ \ \mathrm{rad}\,(\mathbf{x}_i) \leq \epsilon$*).*

Remarque : Si D est suffisamment petit, on peut avoir une boîte frontière qui soit également extérieure.

Le symbole $\mathbb{I}D$ désignera l'ensemble des boîtes intérieures de D.

$$\left(\forall D \subseteq \mathbb{R}^n\right) \quad \mathbb{I}D := \{\mathbf{x} \in \mathbb{IR}^n \mid \mathbf{x} \subseteq D\}.$$

Remarquons que sur la figure 2.1, la boîte $\mathbf{x}_1$ peut être agrandie tout en restant dans D, ce qui n'est pas le cas de $\mathbf{x}_2$. On dira que $\mathbf{x}_2$ est *non-extensible* :

Définition 2.2 *Soient $D \in \mathcal{P}(\mathbb{R}^n)$, $\mathbf{x} \in \mathbb{IR}^n$.*
$\mathbf{x}$ *est une* **boîte intérieure non-extensible** *de D si* $\left(\forall \mathbf{y} \in \mathbb{IR}^n\right) \quad \mathbf{x} \subsetneq \mathbf{y} \Longrightarrow \mathbf{y} \not\subseteq D$.

2.2.4 Opérateur d'enveloppe

L'opérateur $\square$, appliqué à un ensemble quelconque de vecteurs réels D représente la plus petite boîte extérieure de cet ensemble (l'existence d'un minimum est évidente, grâce à l'intersection). On l'appelle **enveloppe** (en anglais : *hull*) de D.

$$\left(\forall D \in \mathcal{P}(\mathbb{R}^n)\right)\ \square D := \inf_{\subseteq}\{\mathbf{x},\ (\mathbf{x} \in \mathbb{IR}^n) \wedge (D \subseteq \mathbf{x})\}. \tag{2.1}$$

On a donc trivialement : $\left(\forall \mathbf{x} \in \mathbb{IR}^n\right)\ D \subseteq \mathbf{x} \Longrightarrow \square D \subseteq \mathbf{x}$.

Soit $\mathbf{x} \in \mathbb{IR}^n$ tel que pour tout $i \in [1..n]$, $\mathbf{x}_i$ contienne la projection de tous les points de D sur la $i^{ème}$ composante. On a alors $D \subseteq \mathbf{x}$. On en déduit que $(\square D)_i$ doit nécessairement être le plus petit intervalle contenant la projection de tous les point de D sur la $i$ème composante :

$$\big(\forall D \in \mathcal{P}(\mathbb{R}^n)\big) \quad \mathbf{x} = \square D \Longleftrightarrow \big(\forall i \in [1..n]\big)\ \mathbf{x}_i := [\inf_{x \in D}(x_i), \sup_{x \in D}(x_i)] \tag{2.2}$$

L'opérateur d'enveloppe apparaît notamment dans le cadre du calcul des images d'une fonction :

Définition 2.3 (Image) *Soit f une fonction de $D \subseteq \mathbb{R}^n$ dans $\mathbb{R}^m$, Soit $\mathbf{x} \in \mathbb{I}D$. On appelle image de la fonction f sur $\mathbf{x}$ le sous-ensemble de $\mathbb{R}^m$ suivant :*

$$range(f, \mathbf{x}) := \{f(x),\ x \in \mathbf{x}\},$$

c.a.d., l'ensemble des images de la fonction f sur la boîte $\mathbf{x}$.

Volontairement, nous ne notons pas $f(\mathbf{x})$ cette image, comme cela se fait habituellement, car la notation $f(\mathbf{x})$ sera réservée à un autre usage. Une des tâches centrales de l'analyse par intervalles est de déterminer $range(f, \mathbf{x})$, pour une boîte $\mathbf{x} \in \mathbb{I}D$ donnée. Malheureusement, l'image d'une fonction a en général une forme compliquée et si l'on souhaite (comme ce sera souvent le cas) représenter cette image par une simple boîte extérieure, on est forcé d'effectuer une approximation grossière. Ce phénomène s'appelle *effet d'enveloppe*. Bien que trivial, il mérite d'être énoncé comme une propriété pour ses innombrables conséquences.

Propriété 2.1 (Effet d'enveloppe) *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ avec $m > 1$, et $\mathbf{x} \in \mathbb{I}D$. Hormis dans des cas particuliers (cf. §2.4.2)*

$$range(f, \mathbf{x}) \subsetneq \square\, range(f, \mathbf{x}) \quad \textit{(inclusion mais non égalité)}$$

L'analyse par intervalles ne manipulant que des boîtes, il y a donc dès le départ (avant le moindre calcul) une forte surestimation liée à la manière de représenter les ensembles. Un des objectifs de l'analyse par intervalles est donc d'approximer $\square\, range(f, \mathbf{x})$, pour toute boîte $\mathbf{x} \in \mathbb{I}D$. La proposition suivante montre qu'il suffit pour cela de savoir calculer l'image de chaque composante. Elle s'appuie sur le lemme suivant, bien connu :

Lemme 2.1 *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ continue.*

$$\big(\forall \mathbf{x} \in \mathbb{I}D\big) \quad \square\, range(f, \mathbf{x}) = range(f, \mathbf{x})$$

Autrement dit, l'image d'un intervalle par une fonction continue est un intervalle (l'effet d'enveloppe ne concerne donc pas le cas $m = 1$). Pour le cas général, on a :

Proposition 2.1 *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ continue.*

$$\big(\forall \mathbf{x} \in \mathbb{I}D\big) \quad \square\, range(f, \mathbf{x}) = \Big(range(f_1, \mathbf{x}) \quad \dots \quad range(f_m, \mathbf{x})\Big)^T$$

Preuve.
Prenons $\mathbf{x}$ quelconque, et montrons $(\subseteq)$.
Par définition de $\square$, $\big(\forall \mathbf{x} \in \mathbb{I}D\big) \quad \square\, range(f, \mathbf{x}) = \Big(\square\, range(f_1, \mathbf{x}) \quad \dots \quad \square\, range(f_m, \mathbf{x})\Big)^T$. Comme f est continue, pour tout $i \in [1..m]$, f_i est continue et en appliquant le lemme 2.1 on obtient $\square\, range(f, \mathbf{x}) = \Big(range(f_1, \mathbf{x}) \quad \dots \quad range(f_m, \mathbf{x})\Big)^T$. Le sens $(\supseteq)$ est évident. □

Remarque 2.1 *Enveloppe et boîte intérieure non extensible sont caractérisées par une propriété symétrique :*

$\mathbf{x}$ *est l'enveloppe de D si*	$D \subseteq \mathbf{y} \Longrightarrow \mathbf{x} \subseteq \mathbf{y}$,
$\mathbf{x}$ *est une boîte intérieure non extensible de D si*	$\mathbf{x} \subseteq \mathbf{y} \subseteq D \Longrightarrow \mathbf{x} = \mathbf{y}$.

2.3 Arithmétique d'intervalles

Dans toute cette thèse, nous manipulerons des expressions mathématiques obtenues uniquement par composition d'un certain ensemble de fonctions de base, comprenant les opérations arithmétiques binaires usuelles ($+,\times,-,/$), et des fonctions dites *élémentaires* telles que sqr, sin, exp, ...

Voici à titre d'exemple la liste des fonctions de base prises en compte dans notre bibliothèque :

Opérations binaires :	$x+y$, $x-y$, $x\times y$, x/y.
Fonctions élémentaires :	exp, ln, cos, sin, tan, $arccos$, $arcsin$, $arctan$, $cosh$, $sinh$, $tanh$, $arccosh$, $arcsinh$, $arctanh$.
Puissances :	x^a (a entier)

Bien entendu, beaucoup de fonctions sont redondantes, dans la mesure où certaines d'entre elles peuvent se calculer à partir d'autres (ex : $tan(x) = sin(x)/cos(x)$). Elles sont ajoutées à des fins d'efficacité (à chaque fonction correspond des procédures câblées).

Voici un exemple d'expression construite par composition de ces fonctions de base :

$$exp(x + sqr(y \times z)) - sin(x)/y$$

De telles expressions seront appelées *expressions arithmétiques*[2]. La grammaire des expressions arithmétiques est simple et bien connue, nous ne l'expliciterons pas. Une fonction $f : \mathbb{R}^n \to \mathbb{R}^m$ dont l'expression $f_i(x)$ est arithmétique (pour tout $i \in [1..m]$) sera appelée *fonction arithmétique*.

2.3.1 Opérations binaires

Comme nous l'avons évoqué en introduction, notre but est d'effectuer des calculs qui produisent "en sortie" des intervalles englobant toutes les combinaisons possibles obtenues avec les variations présentes "en entrée". Il est donc assez naturel d'étendre aux intervalles une opération "$\cdot$" entre réels de la façon suivante :

$$\mathbf{x} \odot \mathbf{y} := range(x \cdot y,\ x \in \mathbf{x}, y \in \mathbf{y}) \tag{2.3}$$

Nous utiliserons dans ce §2.3.1 un rond autour du symbole lorsque les opérandes sont des intervalles pour bien marquer la distinction avec le cas réel.

En vertu du lemme 2.1, $\mathbf{x} \odot \mathbf{y}$ est un intervalle et notre définition est donc équivalente à la suivante :

$$\mathbf{x} \odot \mathbf{y} = [\min_{x\in\mathbf{x},y\in\mathbf{y}} x \cdot y, \max_{x\in\mathbf{x},y\in\mathbf{y}} x \cdot y]$$

Or, si l'on prend par exemple l'opérateur "+" il est clair que

$$\min_{x\in\mathbf{x},y\in\mathbf{y}} x + y = \underline{\mathbf{x}} + \underline{\mathbf{y}} \quad \text{et} \quad \max_{x\in\mathbf{x},y\in\mathbf{y}} x + y = \overline{\mathbf{x}} + \overline{\mathbf{y}},$$

si bien que $\mathbf{x} \oplus \mathbf{y} = [\underline{\mathbf{x}} + \underline{\mathbf{y}}, \overline{\mathbf{x}} + \overline{\mathbf{y}}]$. Exemple : $[1,2] \oplus [3,4] = [5,6]$.

Cette nouvelle expression a le mérite cette fois d'être calculable directement, puisqu'elle ne fait intervenir que les bornes des intervalles $\mathbf{x}$ et $\mathbf{y}$.

Ce raisonnement peut être généralisé aux autres opérateurs car ils présentent les mêmes propriétés de continuité et de monotonie que l'addition. Considérons en effet un opérateur "$\cdot$" parmi $\{+,-,\times,/\}$, ainsi qu'un couple d'intervalles $(\mathbf{x},\mathbf{y})$ tel que $\mathbf{x} \times \mathbf{y}$ soit dans le domaine de définition de "$\cdot$". On a alors :
– $(x,y) \mapsto x \cdot y$ est continue sur $\mathbf{x} \times \mathbf{y}$

[2]Nous incorporons par abus de langage les fonctions élémentaires dans le terme d'« arithmétique ».

– Pour tout $y \in \mathbf{y}$, l'application partielle $(.,y) : x \mapsto x \cdot y$ est monotone sur $\mathbf{x}$,
– Pour tout $x \in \mathbf{x}$, l'application partielle $(x,.) : y \mapsto x \cdot y$ est monotone sur $\mathbf{y}$.
On montre alors que les extrema de la fonction $(x,y) \mapsto x \cdot y$ sur $\mathbf{x} \times \mathbf{y}$ sont nécessairement atteints en des points obtenus par combinaison des bornes de $\mathbf{x}$ et $\mathbf{y}$:

$$\mathbf{x} \odot \mathbf{y} = [\min\{\underline{\mathbf{x}} \cdot \underline{\mathbf{y}}, \underline{\mathbf{x}} \cdot \overline{\mathbf{y}}, \overline{\mathbf{x}} \cdot \underline{\mathbf{y}}, \overline{\mathbf{x}} \cdot \overline{\mathbf{y}}\}, \max\{\underline{\mathbf{x}} \cdot \underline{\mathbf{y}}, \underline{\mathbf{x}} \cdot \overline{\mathbf{y}}, \overline{\mathbf{x}} \cdot \underline{\mathbf{y}}, \overline{\mathbf{x}} \cdot \overline{\mathbf{y}}\}]$$

De la même manière que pour l'addition, il est possible de déterminer a priori quelle combinaison des bornes convient. L'intérêt étant d'éviter la formule générale, qui implique quatre calculs pour chaque borne. On obtient pour la soustraction :

$$\mathbf{x} \ominus \mathbf{y} = [\underline{\mathbf{x}} - \overline{\mathbf{y}}, \overline{\mathbf{x}} - \underline{\mathbf{y}}]$$

Pour la multiplication, le tableau suivant indique le résultat de $\mathbf{x} \otimes \mathbf{y}$:

	$\mathbf{y} < 0$	$0 \in \mathbf{y}$	$\mathbf{y} > 0$
$\mathbf{x} < 0$	$[\overline{\mathbf{x}} \times \overline{\mathbf{y}}, \underline{\mathbf{x}} \times \underline{\mathbf{y}}]$	$[\underline{\mathbf{x}} \times \overline{\mathbf{y}}, \underline{\mathbf{x}} \times \underline{\mathbf{y}}]$	$[\underline{\mathbf{x}} \times \overline{\mathbf{y}}, \overline{\mathbf{x}} \times \underline{\mathbf{y}}]$
$0 \in \mathbf{x}$	$[\overline{\mathbf{x}} \times \underline{\mathbf{y}}, \underline{\mathbf{x}} \times \overline{\mathbf{y}}]$	$[\min\{\underline{\mathbf{x}} \times \overline{\mathbf{y}}, \overline{\mathbf{x}} \times \underline{\mathbf{y}}\}, \max\{\underline{\mathbf{x}} \times \underline{\mathbf{y}}, \overline{\mathbf{x}} \times \overline{\mathbf{y}}\}]$	$[\underline{\mathbf{x}} \times \overline{\mathbf{y}}, \overline{\mathbf{x}} \times \overline{\mathbf{y}}]$
$\mathbf{x} > 0$	$[\overline{\mathbf{x}} \times \underline{\mathbf{y}}, \underline{\mathbf{x}} \times \overline{\mathbf{y}}]$	$[\overline{\mathbf{x}} \times \underline{\mathbf{y}}, \overline{\mathbf{x}} \times \overline{\mathbf{y}}]$	$[\underline{\mathbf{x}} \times \underline{\mathbf{y}}, \overline{\mathbf{x}} \times \overline{\mathbf{y}}]$

Enfin, la division intervalle $\mathbf{x} \oslash \mathbf{y}$ est définie ainsi :

	$\mathbf{y} < 0$	$\mathbf{y} > 0$
$\mathbf{x} < 0$	$[\overline{\mathbf{x}}/\overline{\mathbf{y}}, \underline{\mathbf{x}}/\underline{\mathbf{y}}]$	$[\underline{\mathbf{x}}/\overline{\mathbf{y}}, \overline{\mathbf{x}}/\underline{\mathbf{y}}]$
$0 \in \mathbf{x}$	$[\overline{\mathbf{x}}/\overline{\mathbf{y}}, \underline{\mathbf{x}}/\overline{\mathbf{y}}]$	$[\underline{\mathbf{x}}/\underline{\mathbf{y}}, \overline{\mathbf{x}}/\underline{\mathbf{y}}]$
$\mathbf{x} > 0$	$[\overline{\mathbf{x}}/\underline{\mathbf{y}}, \underline{\mathbf{x}}/\overline{\mathbf{y}}]$	$[\underline{\mathbf{x}}/\underline{\mathbf{y}}, \overline{\mathbf{x}}/\overline{\mathbf{y}}]$

Le cas $0 \in \mathbf{y}$ est indéfini. La valeur la plus cohérente que l'on peut associer à la division par un intervalle contenant 0 eu égard à notre définition, est l'intervalle $]-\infty, +\infty[$. Nous verrons au chapitre 6 une arithmétique plus sophistiquée, où la division par un intervalle contenant 0 est mieux prise en compte.

2.3.2 Fonctions élémentaires

Les fonctions élémentaires sont également redéfinies pour des opérandes de type intervalles, suivant le même principe. Nous utiliserons également (temporairement) des lettres majuscules pour distinguer une fonction "intervalle" d'une fonction "réelle". On définit donc :

$$F(\mathbf{x}) := range(f, \mathbf{x}). \tag{2.4}$$

De nouveau, cette définition se ramène à celle n'impliquant que des extrema locaux de f :

$$F(\mathbf{x}) := [\inf_{x \in \mathbf{x}} f(x), \sup_{x \in \mathbf{x}} f(x)].$$

On voit qu'il est alors possible d'exprimer $F(\mathbf{x})$ uniquement en fonction de $\underline{\mathbf{x}}$ et $\overline{\mathbf{x}}$, par connaissance des sens de variation de f. Voici deux exemples :

$$Sqr(\mathbf{x}) = \begin{array}{ll} [\underline{\mathbf{x}}^2, \overline{\mathbf{x}}^2] & \text{si } \mathbf{x} > 0, \\ [\overline{\mathbf{x}}^2, \underline{\mathbf{x}}^2] & \text{si } \mathbf{x} < 0, \\ [0, (\max\{\underline{\mathbf{x}}^2, \overline{\mathbf{x}}^2\})] = [0, |\mathbf{x}|^2] & \text{si } 0 \in \mathbf{x}. \end{array}$$

Considérons enfin la primitive cosinus, et traitons le cas $\mathbf{x} > 0$ (le cas $\mathbf{x} < 0$ se déduit automatiquement par parité, le cas $\mathbf{x} = 0$ s'obtient facilement en adaptant).

On calcule d'abord pour $\mathbf{x}$ les valeurs suivantes :

$$e_1 := E(\underline{\mathbf{x}}/\pi) \quad \text{et} \quad e_2 := E(\overline{\mathbf{x}}/\pi)$$

où $E(x)$ correspond à la partie entière de x. Ainsi, e_1 (resp. e_2) désigne le nombre de demi-périodes séparant 0 de $\underline{\mathbf{x}}$ (resp. $\overline{\mathbf{x}}$). On a alors :

$$\begin{array}{lll}
\mathrm{Cos}(\mathbf{x}) = & [-1,1] & \text{si } e_1 < e_2 - 1, \\
& [\min\{\cos(\underline{\mathbf{x}}),\cos(\overline{\mathbf{x}})\},1] & \text{si } e_1 = e_2 - 1 \quad \text{et } e_1 \text{ est impair}, \\
& [-1,\max\{\cos(\underline{\mathbf{x}}),\cos(\overline{\mathbf{x}})\}] & \text{si } e_1 = e_2 - 1 \quad \text{et } e_1 \text{ est pair}, \\
& [\cos(\underline{\mathbf{x}}),\cos(\overline{\mathbf{x}})] & \text{si } e_1 = e_2 \quad \text{et } e_1 \text{ est impair}, \\
& [\cos(\overline{\mathbf{x}}),\cos(\underline{\mathbf{x}})] & \text{si } e_1 = e_2 \quad \text{et } e_1 \text{ est pair}.
\end{array}$$

Nous utiliserons dorénavant les mêmes symboles pour les fonctions (et les opérateurs), que les opérandes soient réelles ou intervalles. Le fait que les opérandes soient en lettres grasses indiquera qu'il s'agit d'intervalles, et suffira donc à lever toute ambiguïté.

2.3.3 Fonctions quelconques

A partir des définitions précédentes, on voit qu'il est aisé de donner une sémantique aux expressions arithmétiques mettant en jeu des intervalles. Il suffit pour cela d'appliquer récursivement les définitions en "remontant" l'arbre syntaxique de cette expression. Ainsi l'expression suivante :

$$\mathbf{x} * sqr(\mathbf{y} - \mathbf{z})$$

est associée à un calcul, et pour $\mathbf{x} = [1,2]$, $\mathbf{y} = [1,3]$ et $\mathbf{z} = [2,4]$, ce calcul est

$$\begin{array}{rll}
\mathbf{y} - \mathbf{z} & = [1,3] - [2,4] & \rightarrow [-3,1] \\
sqr(\mathbf{y} - \mathbf{z}) & = sqr([-3,1]) & \rightarrow [0,9] \\
\mathbf{x} * sqr(\mathbf{y} - \mathbf{z}) & = [1,2] \times [0,9] & \rightarrow [0,18]
\end{array}$$

et la valeur de cette expression est définie comme étant l'intervalle $[0,18]$.

A ce stade, rien ne garantit que les relations (2.4) et (2.3) peuvent être étendues, c.a.d. que pour toute fonction de $\mathbb{R}^n$ dans $\mathbb{R}$ telle que $f(x)$ soit une expression arithmétique :

$$\left(\forall \mathbf{x} \in \mathbb{IR}^n\right)\ f(\mathbf{x}) = range(f, \mathbf{x}). \tag{2.5}$$

Cette relation est en fait fausse dans le cas général. Nous allons voir plus loin (cf.§2.4) que seule une inclusion peut être établie :

$$\left(\forall \mathbf{x} \in \mathbb{IR}^n\right)\ range(f, \mathbf{x}) \subseteq f(\mathbf{x}).$$

2.3.4 Calcul matriciel

Les opérations d'addition, de multiplication (interne) et de multiplication par un scalaire (de $\mathbb{IR}$), et en particulier la multiplication matrice-vecteur, sont définies en appliquant les lois arithmétiques scalaires (dans $\mathbb{IR}$) de la même manière que pour les matrices réelles. Ainsi :

$$\begin{array}{llll}
\mathbf{A} \in \mathbb{IR}^{m\times n}, \mathbf{B} \in \mathbb{IR}^{m\times n} & \forall i \in [1..m] & \forall j \in [1..n] & (\mathbf{A} + \mathbf{B})_{ij} := \mathbf{A}_{ij} + \mathbf{B}_{ij} \\
\mathbf{A} \in \mathbb{IR}^{m\times p}, \mathbf{B} \in \mathbb{IR}^{p\times n} & \forall i \in [1..m] & \forall j \in [1..n] & (\mathbf{AB})_{ij} := \displaystyle\sum_{k=1}^{p} (\mathbf{A}_{ik}\mathbf{B}_{kj}) \\
\mathbf{A} \in \mathbb{IR}^{m\times n}, \boldsymbol{\lambda} \in \mathbb{IR} & \forall i \in [1..m] & \forall j \in [1..n] & (\boldsymbol{\lambda}\mathbf{A})_{ij} = \boldsymbol{\lambda}(\mathbf{A}_{ij}) \\
\mathbf{A} \in \mathbb{IR}^{m\times n}, \mathbf{x} \in \mathbb{IR}^{n} & \forall i \in [1..n] & & (\mathbf{Ax})_{i} = \displaystyle\sum_{j=1}^{m} \mathbf{A}_{ij}\mathbf{x}_j
\end{array}$$

Des propriétés plus avancées sur les matrices intervalles sont étudiées au chapitre 4. En particulier, la résolution de systèmes linéaires par intervalles, introduite au §2.7, aura une place importante. Parmi les quatre définitions ci-dessus, nous verrons que seule la somme de matrices continue de vérifier :

$$\mathbf{A} + \mathbf{B} = range(A + B, A \in \mathbf{A}, B \in \mathbf{B}).$$

2.4 Propriétés fondamentales de l'arithmétique

La première propriété fondamentale des calculs par intervalles est qu'ils sont *conservatifs*, c.a.d., qu'ils englobent l'ensemble des calculs obtenus en combinant les réels contenus dans ces intervalles. On place directement l'énoncé du théorème dans le cas général d'une fonction de $\mathbb{R}^n$ dans $\mathbb{R}^m$:

Théorème 2.1 (Conservatisme) *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ telle que $f(x)$ soit une expression arithmétique.*

$$\big(\forall \mathbf{x} \in \mathbb{I}D\big) \quad range(f, \mathbf{x}) \subseteq f(\mathbf{x})$$

où $f(\mathbf{x})$ désigne la boîte obtenue en évaluant par l'arithmétique d'intervalles l'expression $f(x)$ sur $\mathbf{x}$.

La preuve consiste, pour chaque composante f_i, à considérer un point x quelconque de $\mathbf{x}$, puis à appliquer inductivement les relations (2.4) et (2.3) sur l'arbre syntaxique de $f_i(x)$ pour prouver que chaque sous-expression $g(x)$ de $f_i(x)$ vérifie $g(x) \in g(\mathbf{x})$. On applique pour finir la proposition 2.1.

La seconde propriété fondamentale des calculs par intervalles est qu'ils sont *pessimistes*, c.a.d., qu'ils n'offrent qu'une surestimation de l'image, dès lors qu'une même variable possède plusieurs occurrences.

Propriété 2.2 (Dépendance (ou pessimisme)) *Soit f une fonction quelconque (y compris à valeurs dans $\mathbb{R}$). Hormis dans des cas particuliers (cf. §2.4.2), si une variable apparaît plusieurs fois dans l'expression de f,*

$$range(f, \mathbf{x}) \subsetneq f(\mathbf{x}) \quad \textit{(inclusion mais non égalité)}$$

Plus on imbrique des opérations et des fonctions élémentaires dans une expression, plus le pessimisme s'étend lorsqu'il s'agit d'évaluer l'image de cette expression via l'arithmétique. Prenons l'exemple de la fonction $f : x \mapsto x^3 - 3x^2 + x$. Si on utilise l'arithmétique d'intervalles pour évaluer $range(f, [2, 3])$, on effectue le calcul suivant :

$$\mathbf{x}^3 - 3 \times \mathbf{x}^2 + \mathbf{x} = [2,3]^3 - 3 \times [2,3]^2 + [2,3] = [8,27] - 3 \times [4,9] + [2,3] = [-17,18].$$

Le résultat est mauvais, puisque en réalité $range(f, [2, 3]) = [-2, 3]$. Le problème du pessimisme est donc lié aux occurrences multiples d'une variable dans une expression. Voyons maintenant de quelle manière cela impacte les règle usuelles d'arithmétique.

2.4.1 Dépendance et opérateurs binaires

Les opérations dans $\mathbb{IR}$ héritent d'un grand nombre de propriétés des opérations correspondantes dans $\mathbb{R}$. Citons-en une : $\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$. Autrement dit, l'addition dans $\mathbb{IR}$ est commutative. Plutôt que de lister ici[3] les propriétés vérifiées par l'arithmétique dans $\mathbb{IR}$ - propriétés que l'on applique de toute manière spontanément dans les calculs - nous allons insister sur les propriétés qu'elle ne vérifie **pas**. Toutes ces "non-propriétés" découlent de la propriété 2.2 de dépendance.

Tout d'abord, la multiplication n'est pas distributive. Elle vérifie une propriété plus faible, appelée sous-distributivité :

$$\mathbf{a}(\mathbf{b} + \mathbf{c}) \subseteq \mathbf{ab} + \mathbf{ac}.$$

La raison de cette inclusion est que $\mathbf{a}$ apparaît deux fois dans la forme développée. Voici un exemple [Neu90], où seule l'inclusion est vérifiée (et non l'égalité) :

$$[-1,1]([0,1] + [-1,0]) = [-1,1] \subset [-2,2] = [-1,1][0,1] + [-1,1][-1,0]$$

[3] Une liste exhaustive est fournie dans [Neu90].

Plus flagrant encore, $\mathbf{a} - \mathbf{a}$ ne vaut pas 0 ; et de nouveau, seule l'inclusion $0 \subseteq \mathbf{a} - \mathbf{a}$ est vraie[4] en général. On obtient des inclusions similaires avec les autres opérateurs, par exemple, $1 \subseteq \mathbf{a}/\mathbf{a}$. En général ces inclusions sont bien strictes, ex : $[0,1] - [0,1] = [-1,1] \supset 0$. Une autre manière de comprendre ce phénomène est de simplement se dire que $-\mathbf{a}$ (resp. $1/\mathbf{a}$) n'est pas l'inverse de $\mathbf{a}$ pour l'addition (resp. la multiplication). La soustraction n'a donc pas de lien algébrique avec l'addition, ce sont deux lois indépendantes. Ceci empêche toute simplification formelle (comme remplacer $\mathbf{a} - \mathbf{a}$ par 0) dans des expressions mettant en jeu des intervalles. Un piège classique consiste, par exemple, à croire que la relation $\mathbf{x} = \mathbf{y} + \mathbf{z}$ implique $\mathbf{y} = \mathbf{x} - \mathbf{z}$. Cette implication court-circuite en fait les implications suivantes :

$$\mathbf{x} = \mathbf{y} + \mathbf{z} \implies \mathbf{x} - \mathbf{z} = (\mathbf{y} + \mathbf{z}) - \mathbf{z} \implies \mathbf{x} - \mathbf{z} = \mathbf{y} + (\mathbf{z} - \mathbf{z}) \implies \mathbf{x} - \mathbf{z} = \mathbf{y}$$

qui sont fausses puisque $\mathbf{z} - \mathbf{z} \neq 0$.

Remarquons que ce n'est pas une *faille* de l'arithmétique des intervalles ! Si une entité x a pour domaine de variation l'intervalle $\mathbf{a} = [0,1]$, et qu'une entité entité y a également pour domaine de variation l'intervalle $\mathbf{a} = [0,1]$, il est clair que l'intervalle $\mathbf{a} - \mathbf{a} = [-1,1]$ décrit exactement les valeurs possibles de $x - y$. Un problème se pose par contre dès qu'une même entité x apparaît plusieurs fois dans une expression : clairement, l'arithmétique décorrelle chaque occurrence de l'entité x en considérant que ces occurrences peuvent varier indépendamment les unes des autres.

2.4.2 Dépendance et effet d'enveloppe

Nous avons identifié deux causes pouvant entraîner $f(\mathbf{x}) \subsetneq range(f, \mathbf{x})$: l'effet d'enveloppe (voir section 2.2.4), et le problème de dépendance. On peut remarquer que ces deux phénomènes sont liés dans le sens où l'effet d'enveloppe n'est qu'un cas particulier du problème de dépendance. En effet, si $f : \mathbb{R}^2 \to \mathbb{R}^2$ est la fonction suivante $f(x,y) = (x,y)^T$, il est alors évident que $range(f, \mathbf{x} \times \mathbf{y}) = \square\, range(f, \mathbf{x} \times \mathbf{y})$ précisément parce que chaque composante de f varie indépendamment l'une de l'autre, et ce, parce que les variables n'apparaissent qu'une seule fois dans l'expression de f. Finalement, nous n'avons plus qu'à retenir la propriété de dépendance dont l'effet d'enveloppe est un corollaire.

Voici quelques uns des cas particuliers où $range(f, \mathbf{x}) = \square\, range(f, \mathbf{x})$, malgré la présence d'occurrences multiples :
- $\mathbf{x}$ est dégénéré.
- f est à valeurs réelles, et si x apparaît plusieurs fois dans f, f est "linéaire en x" avec des coefficients de même signe (ex : $f(x,y,z) = exp(y)x + z^2 x$).
- $\mathbf{x} \geq 0$ et f est un polynôme en x avec des coefficients positifs (ex : $f(x) = x^2 + x$)
- ...

2.4.3 Dépendance dans le cas linéaire

L'effet d'enveloppe s'observe dans le cas d'une application linéaire. Prenons par exemple $x \mapsto Ax$ avec $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ et $\mathbf{x} = ([-1,1],[-1,1])^T$. On a

$$A\mathbf{x} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} [-1,1] \\ [-1,1] \end{pmatrix} = \begin{pmatrix} [-2,2] \\ [-2,2] \end{pmatrix}.$$

Or, le vecteur $\begin{pmatrix} -2 \\ 2 \end{pmatrix}$ ne peut clairement être obtenu pour aucun $x \in \mathbf{x}$. Donc $A\mathbf{x} \supsetneq range(Ax, x \in \mathbf{x})$.
A fortiori, le produit $\mathbf{AB}$ ne peut être égal à $range(A \times B, A \in \mathbf{A}, B \in \mathbf{B})$. Dans le cas carré par exemple,

[4] Le terme *inclusion* peut paraître incorrect, mais il ne l'est pas dès lors que 0 est identifié à l'intervalle $[0,0]$.

chaque élément de A et de B apparaît dans le calcul de n éléments de la matrice produit. Les dépendances sont multiples, ce qui amène à éviter le plus possible le calcul du produit de matrices intervalles.

La sous-distributivité s'écrit comme dans le cas scalaire : $\mathbf{A}(\mathbf{B}+\mathbf{B}') \subseteq \mathbf{AB}+\mathbf{AB}'$.
Une différence avec le cas scalaire apparaît pour l'associativité :

$$\mathbf{A}(\mathbf{BC}) \neq (\mathbf{AB})\mathbf{C}.$$

L'égalité demeure dans le cas scalaire car le produit de 2 intervalles n'introduit pas de pessimisme.

2.5 Bissection-Evaluation

Nous avons rassemblé déjà suffisamment d'éléments pour pouvoir décrire un premier algorithme de recherche de solutions. Nous allons donc décrire la méthode de base de l'analyse par intervalles (qui ne repose que sur le théorème 2.1) pour déterminer de façon fiable l'ensemble des solutions d'un système d'équations. Nous verrons qu'à ce stade, « fiable » signifie que la méthode ne peut pas *perdre de solutions*. En revanche, elle ne peut pas garantir qu'une solution se trouve réellement à l'intérieur d'une boîte produite.

Nous illustrerons l'algorithme sur l'exemple donné en introduction p.1, dans le cas le plus simple, sans paramètre.

Définition du problème

Le problème est le suivant :

Soit f une fonction (arithmétique) de $\mathbb{R}^n$ dans $\mathbb{R}^m$. On note $\Sigma(f)$ l'ensemble des zéros de f, c.a.d.,

$$\Sigma(f) := \{x \mid f(x) = 0_{\mathbb{R}^m}\}.$$

Étant donnée une boîte $\mathbf{x}^{(0)}$ décrivant le domaine initial des variables, et une valeur réelle $\epsilon > 0$, déterminer un ensemble de boîtes $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}\}$ tel que

- $\forall i \in [1..k],\ \max\limits_{1\leq j\leq n} \mathrm{rad}\,(\mathbf{x}_j^{(i)}) \leq \epsilon$
- $x \in (\Sigma(f) \cap \mathbf{x}^{(0)}) \Longrightarrow \exists i \in [1..k]\ x \in \mathbf{x}^{(i)}$

Le premier point signifie que les boîtes retournées doivent être suffisamment précises (le plus grand rayon des intervalles constituant une boîte $\mathbf{x}^{(i)}$ doit être en deçà de ϵ), il s'agit donc d'un *critère d'arrêt*; le second point signifie que toutes les (éventuelles) solutions dans $\mathbf{x}^{(0)}$ doivent se trouver à l'intérieur des boîtes retournées.

Remarque 2.2 *Dans un certain nombre de cas, il est plus judicieux de choisir pour critère d'arrêt*

$$\mathrm{rad}\,(f_j(\mathbf{x}^{(i)})) \leq \epsilon \quad \textit{plutôt que} \quad \mathrm{rad}\,(\mathbf{x}_j^{(i)}) \leq \epsilon.$$

Ainsi, ce n'est plus la taille d'une boîte qui détermine sa précision, mais la taille de la boîte image. Cette nuance devient cruciale dans des applications où $f_j(x)$ représente une quantité pour laquelle on souhaite contrôler l'erreur. Le solver `Alias` *[Mer06] propose même une combinaison des deux critères d'arrêt. Ce critère nécessite d'être adapté en présence d'inéquations.*

L'algorithme

L'algorithme consiste à effectuer un arbre de recherche, où chaque noeud est une boîte représentant le domaine courant des variables. L'image de f est estimée sur cette boîte via l'arithmétique d'intervalles. Si 0 n'appartient pas au résultat de cette estimation, alors aucun point de la boîte ne peut être un zéro de f, et cette boîte peut être supprimée. Sinon, elle est coupée en deux sous-boîtes sur lesquelles la recherche est poursuivie récursivement. On parle de **bissection** (équivalent au "branch") pour le découpage en sous-boîtes, et d'**évaluation** (équivalent au "bound") pour le calcul de l'image de f. Enfin, une boîte dont la taille ne dépasse pas ϵ et que l'évaluation n'a pas permis d'exclure, est considérée comme "solution" et stockée dans une liste à part. Elle est retirée du processus de recherche.

Cherchons, par exemple, à déterminer les solutions du problème géométrique direct du robot parallèle à deux jambes (sans perturbation) dans la boîte initiale $\mathbf{x}^{(0)} = [0, 10] \times [0, 10]$. Il s'agit donc du système (1.1) p.2. En fixant les valeurs suivantes pour les coordonnées des points d'attache :

$$a = (0, 0) \quad b = (4, 0) \quad l_1 = 4 \quad l_2 = 6,$$

on a alors :

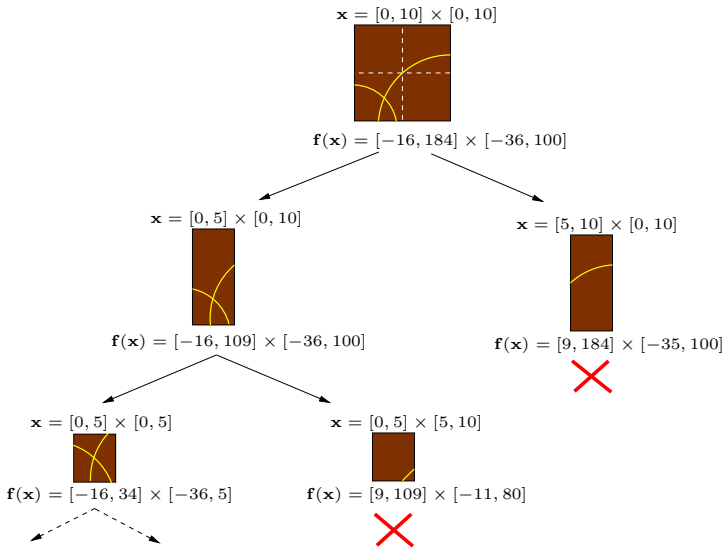
$$\begin{array}{rccl} f: & \mathbb{R}^2 & \to \mathbb{R}^2 \\ & x & \mapsto (x_1^2 + x_2^2 - 4^2,\ (x_1 - 4)^2 + x_2^2 - 6^2)^T. \end{array}$$

L'évaluation de la première boîte donne donc :

$$f\begin{pmatrix} [0,10] \\ [0,10] \end{pmatrix} = \begin{pmatrix} [0,10]^2 + [0,10]^2 - [4,4]^2 \\ ([0,10] - [4,4])^2 + [0,10]^2 - [6,6]^2 \end{pmatrix} = \begin{pmatrix} [-16,184] \\ [-36,100] \end{pmatrix}.$$

Comme $0 \in f(\mathbf{x}^{(0)})$ alors rien ne permet de conclure que $\mathbf{x}^{(0)}$ ne contient pas de solution, et la recherche est poursuivie sur deux sous-boîtes obtenues en coupant le domaine d'une des variables. Dans la figure ci-dessous, le domaine de x_1 devient $[0, 5]$ pour la première sous-boîte et $[5, 10]$ pour la seconde.

La figure illustre la suite du déroulement de l'algorithme. Les boîtes marquées d'une croix sont supprimées (dans un cas $0 \notin [9, 109]$, dans l'autre $0 \notin [9, 184]$).



Il est donc garanti, en vertu du théorème 2.1, que les boîtes éliminées ne peuvent pas contenir de solution. On répond donc bien au problème tel qu'il est posé à la section précédente. L'algorithme peut s'écrire en quelques lignes (la récursivité est gérée par une pile) :

```
empiler x^(0)
tant que la pile n'est pas vide faire
  dépiler une boîte x
  si width(x) < ε
    stocker x comme boîte contenant potentiellement une solution
  sinon si 0 ∈ f(x)
    couper x (sur une dimension) en deux sous-boîtes x^(1) et x^(2)
    empiler x^(1) et x^(2)
  fin si
fin tant que
```

L'algorithme répond au problème mais souffre de limitations importantes :

1. Il présente des performances catastrophiques dès que la dimension du problème augmente. Le nombre de bissections nécessaires pour que l'évaluation détecte l'absence de 0 est en moyenne de l'ordre du nombre de variables, ce qui entraîne presque systématiquement une explosion combinatoire.
2. Il ne permet pas de garantir l'existence ou l'unicité d'une solution dans une boîte.
3. Il ne tient pas compte des éventuelles incertitudes sur les paramètres du problème (comme dans notre exemple les coordonnées des points d'attache).

Cette thèse présente différentes techniques qui permettent de remédier (au moins partiellement) à chacun de ces points.

– Utiliser d'autres manières de calculer l'image d'une fonction que l'application directe de l'arithmétique d'intervalles. On souhaite ainsi éviter le pessimisme en présence d'occurrences multiples de variables. C'est l'objet de la section 2.6.
– Proposer un moyen de *filtrer* les boîtes, c'est à dire d'en réduire les bornes sans perdre de solution. Un filtrage doit réduire l'espace de recherche sans introduire de point de choix, il permet donc de limiter la combinatoire. Il existe beaucoup de manières de filtrer une boîte, c'est d'ailleurs la "spécialité" de la programmation par contraintes. Nous présentons dans ce chapitre le filtrage de Newton, qui est une brique classique de l'analyse par intervalles (cf. §2.8.2). Il offre également un moyen de prouver l'existence et l'unicité d'une solution dans une boîte. Les filtrages proposés en programmation par contraintes seront décrits, eux, au chapitre 6.
– Les incertitudes sur les paramètres feront l'objet du §2.9 dans le prolongement duquel s'inscrit le chapitre 3.

La conception de *solvers* repose sur une collaboration intelligente entre ces différentes techniques [VHMD97, Gra01, Mer06].

2.6 Extensions aux intervalles des fonctions réelles

2.6.1 Un exemple basé sur la monotonie

Prenons une fonction à valeur dans $\mathbb{R}$, dérivable. Nous avons vu dans la section 2.4 que, quel que soit $\mathbf{x}$, calculer $f(\mathbf{x})$ permettait d'obtenir une approximation extérieure de $range(f, \mathbf{x})$.

Si notre but est de trouver un procédé permettant d'approximer $range(f, \mathbf{x})$, il est possible de mieux faire que de prendre systématiquement $f(\mathbf{x})$ (c.a.d. évaluer par intervalles f sur $\mathbf{x}$). Une méthode plus fine, par exemple, consiste à exploiter la dérivée de f. Si on évalue f' par intervalles, et que $f'(\mathbf{x})$ est un intervalle de signe constant (c.a.d., $\underline{f'(\mathbf{x})}\,\overline{f'(\mathbf{x})} \geq 0$), alors la fonction f est forcément monotone sur $\mathbf{x}$. En effet, prenons par exemple le cas où $f'(\mathbf{x}) \geq 0$. En appliquant le théorème 2.1 :

$$range(f', \mathbf{x}) \geq 0,$$

ce qui signifie que

$$\big(\forall x \in \mathbf{x}\big) \quad f'(x) \geq 0$$

et donc que f' est de signe positif sur $\mathbf{x}$, c.a.d., que f est croissante. Or, si f est croissante, il est clair que

$$range(f, \mathbf{x}) = [f(\underline{\mathbf{x}}), f(\overline{\mathbf{x}})].$$

On a donc, dès que la monotonie d'une fonction est détectée, un moyen de calculer l'image d'une fonction de façon optimale.

Reprenons maintenant l'exemple de la fonction $f : x \mapsto x^3 - 3x^2 + x$. Sa dérivée est $f' : x \mapsto 3x^2 - 6x + 1$. Si on évalue f' sur l'intervalle $[2, 3]$, on obtient $[-5, 16]$ et on ne peut rien en déduire. Par contre, sur l'intervalle $\mathbf{x}' = [3, 4]$ on a $f'(\mathbf{x}) = [4, 31]$ et on peut en déduire que $range(f, \mathbf{x}) = [f(3), f(4)] = [3, 20]$, ce qui est bien plus précis que $f(\mathbf{x}) = [-18, 41]$. En résumé, définissons la fonction $\mathbf{f}_m$, de variables intervalles et à valeur intervalle ainsi :

$$\begin{array}{lccll} \mathbf{f}_m : & \mathbb{IR} & \to & \mathbb{IR} & \\ & x & \mapsto & [f(\underline{\mathbf{x}}), f(\overline{\mathbf{x}})] & \text{si } f'(\mathbf{x}) \geq 0, \\ & & & [f(\overline{\mathbf{x}}), f(\underline{\mathbf{x}})] & \text{si } f'(\mathbf{x}) \leq 0, \\ & & & f(\mathbf{x}) & \text{sinon.} \end{array}$$

La fonction $\mathbf{f}_m$ vérifie également $range(f, \mathbf{x}) \subseteq \mathbf{f}_m(\mathbf{x})$. Ceci nous mène à la notion d'*extension aux intervalles* d'une fonction réelle.

2.6.2 Définition et Théorème fondamental

Définition 2.4 (Extension aux intervalles d'une fonction)
Soient $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$, et $\phi : \mathbb{I}D \to \mathbb{IR}^m$. La fonction ϕ est une **extension aux intervalles** *de f si*

$$\begin{array}{ll} (1) & \big(\forall x \in D\big) \quad \phi(x) = f(x), \\ (2) & \big(\forall \mathbf{x} \in \mathbb{I}D\big) \quad range(f, \mathbf{x}) \subseteq \phi(\mathbf{x}). \end{array}$$

La fonction $\mathbf{f}_m$ définie précédemment est une extension aux intervalles de $x \mapsto x^3 - 3x^2 + x$. Nous introduisons également l'**extension optimale $\mathbf{f}^{\square}$** d'une fonction f de la façon suivante :

$$\big(\forall \mathbf{x} \in \mathbb{I}D\big) \quad \mathbf{f}^{\square}(\mathbf{x}) := \square\, range(f, \mathbf{x})$$

qui n'est pas calculable facilement (il suffit que f soit un polynôme pour que le calcul de $\mathbf{f}^{\square}(\mathbf{x})$ devienne un problème NP-difficile [KLRK97]). C'est un objet théorique.

Concernant la condition (1), rappelons tout d'abord que écrire $\phi(x) = f(x)$ est équivalent à écrire $\phi([x, x]) = [f(x), f(x)]$. Cette condition peut paraître inutile, dans le sens où la condition (2) suffit à rendre l'algorithme d'évaluation-bissection correct. En revanche, si cette condition n'était pas exigée, la fonction suivante

$$\begin{array}{ll} \mathbb{I}D & \to \mathbb{IR} \\ \mathbf{x} & \mapsto [-\infty, +\infty] \end{array}$$

où "∞" est un réel suffisamment grand, serait une extension aux intervalles valide d'une fonction quelconque f à valeurs dans $\mathbb{R}$. Il est clair alors que si ce genre d'extensions entraient en ligne de compte, il serait impossible de prouver quoi que ce soit d'intéressant (sans parler du fait que l'évaluation-bissection prendrait un temps désastreux pour ne rien produire du tout). La condition (1) est donc indispensable, et impose déjà la nécessité que, en une boîte dégénérée (réduite à un point), l'évaluation soit optimale.

Les conditions (1) et (2) ne suffisent pas toutefois à éviter d'autres cas aberrants, comme la fonction suivante

$$\begin{array}{lll} \mathbb{I}D & \to \mathbb{IR} & \\ \mathbf{x} & \mapsto \left\{ \begin{array}{ll} f(x) & \text{si (rad } \mathbf{x}) = 0, \\ [-\infty, +\infty] & \text{sinon.} \end{array} \right. \end{array}$$

On peut faire un parallèle entre l'utilisation de cette extension dans l'algorithme d'évaluation-bissection et la méthode dite de "generate & test" qui consiste à dérouler un arbre de recherche entièrement et à tester uniquement au niveau de la feuille la présence de solution.

En pratique, l'algorithme ne peut présenter des performances acceptables que si l'extension utilisée améliore la précision au fur et à mesure que la taille des boîtes diminue. Intuitivement, on peut en effet tolérer une approximation de l'image très grossière sur les boîtes larges traitées au cours de la recherche, dans la mesure où ces boîtes sont peu nombreuses et où elles ont de fortes chances de contenir une solution. En revanche, plus les boîtes deviennent petites, plus il devient crucial de savoir les filtrer, afin de limiter la combinatoire. Nous introduisons pour cela l'ordre de convergence d'une extension :

Définition 2.5 (Ordre de convergence [Moo66])
Soient $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$, et $\phi : \mathbb{I}D \to \mathbb{IR}^m$ une extension aux intervalles de f. On dit que ϕ a un ordre de convergence α ($\alpha \in \mathbb{N}^*$) si

$$\big(\forall \mathbf{x} \in \mathbb{I}D\big)\big(\exists \gamma > 0\big)\big(\forall \mathbf{y} \subseteq \mathbf{x}\big) \quad \text{rad}\,(\phi(\mathbf{y})) - \text{rad}\,(\mathbf{f}^{\square}(\mathbf{y})) < \gamma\, \text{rad}\,(\mathbf{y})^{\alpha}.$$

Le fait que γ dépende de $\mathbf{x}$ permet d' "affiner" cette constante en diminuant la taille de $\mathbf{x}$, sans changer le comportement asymptotique. L'ordre de convergence mesure en quelque sorte l'erreur relative produite par l'extension, c'est à dire le rapport entre la taille d'une boîte et le gonflement obtenu pour le calcul de l'image sur cette boîte. De façon standard, lorsque $\alpha = 1$ nous parlerons de *convergence linéaire*. Lorsque $\alpha = 2$, de *convergence quadratique*. Au vu de ce que nous avons dit précédemment, une convergence quadratique est donc préférable.

Une question se pose : comment prouver qu'une fonction d'intervalles construite par induction (comme ce sera le cas à la section suivante) vérifie la condition (2) de la définition 2.4 ? La réponse vient de Moore, et passe par la notion de *monotonie pour l'inclusion* :

Définition 2.6 (Monotonie pour l'inclusion)
Soit $\phi : \mathbb{I}D \subseteq \mathbb{IR}^n \to \mathbb{IR}^m$. La fonction ϕ est monotone pour l'inclusion si

$$\big(\forall \mathbf{x} \in \mathbb{I}D\big)\big(\forall \mathbf{y} \in \mathbb{I}D\big) \quad \mathbf{x} \subseteq \mathbf{y} \Longrightarrow \phi(\mathbf{x}) \subseteq \phi(\mathbf{y}).$$

Il est immédiat que pour tout f, l'extension optimale $\mathbf{f}^{\square}$ est monotone pour l'inclusion.

Théorème 2.2 (Théorème fondamental de Moore [Moo66])
Soient $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$, et $\phi : \mathbb{I}D \to \mathbb{IR}^m$.

$$\textit{Si} \quad \left\{ \begin{array}{c} \phi \textit{ est monotone pour l'inclusion et} \\ \big(\forall x \in D\big) \quad \phi(x) = f(x) \end{array} \right\} \textit{ alors } \phi \textit{ est une extension aux intervalles de } f.$$

Preuve.
Soit $\mathbf{x} \in \mathbb{I}D$. Comme ϕ est monotone pour l'inclusion, $x \in \mathbf{x} \Longrightarrow \phi(x) \in \phi(\mathbf{x})$. Or $\phi(x) = f(x)$ donc $\big(\forall x \in \mathbf{x}\big)$, $f(x) \in \phi(\mathbf{x})$, c.a.d., $range(f, \mathbf{x}) \subseteq \phi(\mathbf{x})$.
□

Nous avons maintenant en main les outils théoriques pour pouvoir étudier les principales extensions utilisées.

2.6.3 Extension naturelle

L'extension naturelle de f correspond à l' "évaluation simple". Elle est la fonction qui à tout $\mathbf{x} \in \mathbb{I}D$ (D étant le domaine de définition de f) associe l'évaluation par intervalles de l'expression $f(\mathbf{x})$.

Définition 2.7 *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$. On appelle* **extension naturelle** *de f la fonction* $\mathbf{f}$ *suivante :*

$$\begin{array}{rccc} \mathbf{f} : & \mathbb{I}D & \to & \mathbb{IR}^m \\ & \mathbf{x} & \mapsto & f(\mathbf{x}). \end{array}$$

Proposition 2.2 (Propriétés de l'extension naturelle [Moo66]) *L'extension naturelle est une extension aux intervalles, monotone pour l'inclusion, et de convergence linéaire.*

Remarquons que $\mathbf{f}$ est une extension, d'après le théorème 2.1. Le fait que l'extension naturelle soit monotone pour l'inclusion est fondamental, nous y reviendrons à la prochaine sous-section.

Chaque fonction élémentaire (ou opérateur binaire) prise de façon isolée calcule exactement son image (cf. les relations (2.4) et (2.3)). On montre alors par induction qu'il en est de même pour n'importe quelle expression arithmétique complexe, dès lors que les sous-expression n'ont aucune variable en commun.

Proposition 2.3 (Moore [Moo66]) *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$. Si pour tout $i \in [1..m]$, chaque composante de $\mathbf{x}$ n'apparaît qu'une seule fois dans l'expression de f_i, alors*

$$\big(\forall \mathbf{x} \in \mathbb{I}D\big) \quad \mathbf{f}(\mathbf{x}) = \mathbf{f}^{\square}(\mathbf{x}).$$

2.6.4 Extension de Taylor

Soit $f : \mathbb{R} \to \mathbb{R}$ que l'on suppose continûment dérivable sur un ensemble D de réels, et soit $\mathbf{x} \in \mathbb{I}D$. Le théorème des accroissements finis donne

$$\big(\forall a \in \mathbf{x}\big)\big(\forall b \in \mathbf{x}\big)\big(\exists c \in [a,b]\big) \quad f(b) - f(a) = f'(c)(b-a). \tag{2.6}$$

Fixons a et b, puis posons

$$\begin{array}{rccc} g_{a,b} : & [a,b] & \to & \mathbb{R} \\ & c & \mapsto & f(a) + (b-a)f'(c) \end{array}$$

On a donc $\big(\exists c \in [a,b]\big)\ f(b) = g_{a,b}(c)$. L'extension optimale $\mathbf{g}^{\square}_{a,b}$ de $g_{a,b}$ est monotone pour l'inclusion, donc

$$[c,c] \subseteq [a,b] \subseteq \mathbf{x} \Longrightarrow \mathbf{g}^{\square}_{a,b}(c) \subseteq \mathbf{g}^{\square}_{a,b}([a,b]) \subseteq \mathbf{g}^{\square}_{a,b}(\mathbf{x}).$$

Ainsi, $f(b) \subseteq \mathbf{g}^{\square}_{a,b}(\mathbf{x})$. Comme $\mathbf{g}^{\square}_{a,b}(\mathbf{x}) = f(a) + (b-a)\mathbf{f'}^{\square}(\mathbf{x})$, on obtient finalement :

$$\big(\forall a \in \mathbf{x}\big)\big(\forall b \in \mathbf{x}\big) \quad f(b) \in f(a) + (b-a)\mathbf{f'}^{\square}(\mathbf{x}). \tag{2.7}$$

Nous venons de voir que le passage de (2.6) à (2.7) s'appuie sur la monotonie pour l'inclusion de l'extension aux intervalles choisie (en l'occurrence, l'extension optimale). Cette propriété interviendra presque systématiquement dans les calculs faisant intervenir l'extension optimale ou l'extension naturelle, si bien que nous ne la référencerons plus explicitement. Un passage tel que celui de (2.6) à (2.7) se fera sans mention particulière. C'est le cas notamment dans la prochaine proposition.

Proposition 2.4
Soit $f : D \subset \mathbb{R} \to \mathbb{R}$ continûment dérivable, $\mathbf{x} \in \mathbb{I}D$, et a un point quelconque de $\mathbf{x}$.

$$\mathbf{f}^{\square}(\mathbf{x}) \quad \subseteq \quad f(a) + \mathbf{f'}^{\square}(\mathbf{x})(\mathbf{x} - a) \quad \subseteq \quad f(a) + f'(\mathbf{x})(\mathbf{x} - a).$$

Preuve. Évident à partir de (2.7), et en utilisant le fait que $\mathbf{f'}^{\square}(\mathbf{x}) \subseteq f'(\mathbf{x})$. □

Ce résultat s'observe graphiquement. Sur la figure 2.2, les pentes de f ont été calculées en tous les points de la portion de courbe délimitée par l'intervalle $\mathbf{x}$. L'ensemble de ces pentes forment un cône de dérivées, qui placé en un point $(a, f(a))$ (a quelconque dans $\mathbf{x}$) contient le graphe de cette fonction.

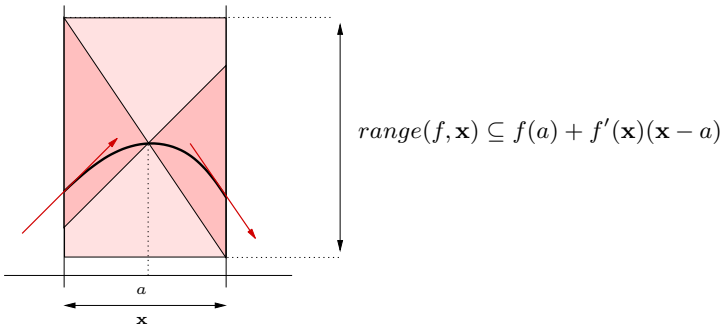


FIG. *2.2: Extension de Taylor (cas réel).*

Pour le cas général, rappelons tout d'abord la définition de gradient et de matrice jacobienne :

Définition 2.8 (Gradient, Jacobienne) *Soit f une fonction différentiable de $D \subseteq \mathbb{R}^n$ à valeurs dans $\mathbb{R}$. On appelle gradient de f la fonction $G : D \mapsto \mathbb{R}^n$ suivante :*

$$G(x) = \big(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x)\big)^T.$$

Soit f une fonction différentiable de $D \subseteq \mathbb{R}^n$ dans $\mathbb{R}^m$. On appelle jacobienne de f la fonction à valeur matricielle $J : D \mapsto \mathbb{R}^{m \times n}$, dont la $i$ème ligne est le gradient de f_i, c.a.d.,

$$J_{ij}(x) := \frac{\partial f_i}{\partial x_j}(x).$$

Le théorème des accroissements finis s'étend aux fonctions à variable vectorielle, de la façon suivante :

Théorème 2.3 (Accroissements finis)
Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ continûment différentiable, $\mathbf{x} \in \mathbb{I}D$ et soit G le gradient de f.

$$\big(\forall a \in \mathbf{x}\big)\big(\forall b \in \mathbf{x}\big)\big(\exists c \in \mathbf{x}\big) \quad f(b) = f(a) + G(c)(b - a).$$

Remarque 2.3 *Le théorème des accroissements finis apporte en réalité une information plus précise : le point c n'appartient pas seulement à $\mathbf{x}$, mais au segment d'extrémités a et b (c.a.d., l'image de $[0, 1]$ par $t \mapsto a + t(b - a)$). Comme nous n'utiliserons pas cette information supplémentaire, nous l'avons volontairement omise.*

Le théorème ne se généralise pas en revanche aux fonctions à valeurs vectorielles (voir un exemple de telle fonction figure 2.3). Il devient faux en prenant f à valeurs dans $\mathbb{R}^m$, avec $m > 1$.

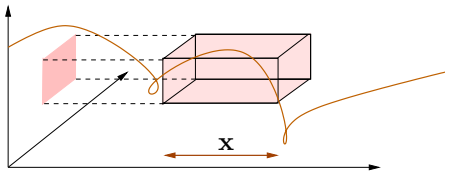


FIG. *2.3: Graphe d'une fonction de* $\mathbb{R}$ *dans* $\mathbb{R}^2$.

En analyse classique, le théorème s'adapte aux fonctions à valeurs dans $\mathbb{R}^m$ en affaiblissant l'égalité, qui se transforme en une inégalité de normes. C'est le théorème dit de *la moyenne*. L'idée ici est de conserver une égalité en considérant un sur-ensemble des jacobiennes (donc en incluant d'autres matrices que celles correspondantes aux jacobiennes calculées sur la boîte $\mathbf{x}$) :

Proposition 2.5 *Soit* $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ *continûment différentiable,* J *la jacobienne de* f*, et* $\mathbf{x} \in \mathbb{I}D$.

$$\big(\forall a \in \mathbf{x}\big)\big(\forall b \in \mathbf{x}\big)\big(\exists M \in \mathbf{J}^{\square}(\mathbf{x})\big) \quad f(b) - f(a) = M(b-a).$$

Preuve. Pour tout i dans $[1..m]$, f_i est une fonction continûment dérivable de D dans $\mathbb{R}$. En appliquant le théorème des accroissements finis, on obtient

$$\big(\exists c_i \in \mathbf{x}\big)\ f_i(b) - f_i(a) = f_i'(c_i)(b-a)$$

Or $f_i'(c_i) = J_i(c_i)$, en notant J_i la $i$ème ligne de la jacobienne de f. Finalement, grâce à la proposition 2.1 on a

$$M = \begin{pmatrix} J_1(c_1) \\ \vdots \\ J_m(c_m) \end{pmatrix} \in \begin{pmatrix} range(J_1, \mathbf{x}) \\ \vdots \\ range(J_m, \mathbf{x}) \end{pmatrix} = \square range(J, \mathbf{x}) = \mathbf{J}^{\square}(\mathbf{x})$$

□

Remarque 2.4 *La proposition précédente fait de la matrice* $\mathbf{J}^{\square}(\mathbf{x})$ *un cas particulier de "slope matrix" [Han78, Neu90].*

On en déduit l'extension de Taylor d'ordre 1 [Moo66, AH83, Neu90, Han92b] :

Proposition 2.6 (Extension de Taylor (ordre 1))
Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ *continûment différentiable,* J *la jacobienne de* f*,* $\mathbf{x} \in \mathbb{I}D$ *et* x *un point quelconque de* $\mathbf{x}$ *(typiquement : le milieu).*

$$\mathbf{f}^{\square}(\mathbf{x}) \quad \subseteq \quad f(x) + \mathbf{J}^{\square}(\mathbf{x})(\mathbf{x} - x) \quad \subseteq \quad f(x) + J(\mathbf{x})(\mathbf{x} - x).$$

En particulier, la fonction $\mathbf{f}_t : \begin{array}{ll} \mathbb{I}D & \to \mathbb{I}\mathbb{R}^m \\ \mathbf{x} & \mapsto f(x) + J(\mathbf{x})(\mathbf{x} - x) \end{array}$ *est une extension*[5] *de* f.

Remarquons que la proposition 2.6 apporte une approximation supplémentaire par rapport à la proposition 2.5, de part l'effet d'enveloppe introduit en multipliant $\mathbf{J}^{\square}(\mathbf{x})$ par $(\mathbf{x} - x)$ (voir §2.4.3). Remarquons également que contrairement à l'extension naturelle, l'extension de Taylor n'est pas monotone (cf. définition 2.6), car son calcul fait intervenir un point x choisi arbitrairement dans l'intervalle $\mathbf{x}$ en argument.

[5] L'extension $\mathbf{f}_t$ a une convergence quadratique.

2.6.5 Extension de Hansen

Nous avons vu que l'extension de Taylor $\mathbf{f}_t$ obligeait à prendre en considération la boîte $\mathbf{x}$ entière pour calculer chaque élément de la jacobienne.

Il se trouve qu'on peut remplacer cette jacobienne intervalle par une matrice beaucoup plus petite (ce n'est donc *plus* la jacobienne de f sur $\mathbf{x}$ qu'on calcule) que dans les propositions 2.5 et 2.6. La $j^{ème}$ colonne de cette nouvelle matrice (cf. proposition 2.8) contient une évaluation des dérivées partielles sur une boîte où $n-j$ dimensions sont réduites à un point. Cette seconde forme, due à [Han92b], permet donc une bien meilleure évaluation des fonctions. Elle repose sur le résultat suivant :

Proposition 2.7 *Soient $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ continûment différentiable, $\mathbf{x} \in \mathbb{I}D$.*

$$\big(\forall a \in \mathbf{x}\big)\big(\forall b \in \mathbf{b}\big)\big(\exists M \in \mathbf{J}_{a,b}\big) \quad f(b) - f(a) = M(b-a),$$

où $\mathbf{J}_{a,b}$ est la matrice intervalle définie ainsi : $\forall i \in [1..m]$, la $i^{ème}$ ligne de $\mathbf{J}_{a,b}$ vaut

$$(\mathbf{J}_{a,b})_i := \Big(\frac{\partial f_i}{\partial x_1}(\mathbf{x}_1, a_2, \ldots, a_n)\ \frac{\partial f_i}{\partial x_2}(b_1, \mathbf{x}_2, a_3 \ldots, a_n)\ \ldots\ \frac{\partial f_i}{\partial x_n}(b_1, \ldots, b_{n-1}, \mathbf{x}_n)\Big).$$

Preuve. La preuve [Han92b] s'obtient en généralisant directement l'exemple suivant donné avec une fonction f de trois variables à valeurs dans $\mathbb{R}^m$. L'idée consiste à appliquer la proposition 2.5 en projetant successivement sur les 3 axes : fixons a et b deux points quelconques de $\mathbf{x}$. On note dans ce qui suit $\dfrac{\partial f}{\partial x_i}$ le vecteur $(\dfrac{\partial f_1}{\partial x_i}, \ldots, \dfrac{\partial f_n}{\partial x_i})^T$.

D'après la proposition 2.5,

$$\Big(\exists M_1 \in \frac{\partial f}{\partial x_1}(\mathbf{x}_1, a_2, a_3)\Big) \quad f(b_1, a_2, a_3) - f(a_1, a_2, a_3) = M_1(b_1 - a_1)$$

$$\Big(\exists M_2 \in \frac{\partial f}{\partial x_2}(b_1, \mathbf{x}_2, a_3)\Big) \quad f(b_1, b_2, a_3) - f(b_1, a_2, a_3) = M_2(b_2 - a_2)$$

$$\Big(\exists M_3 \in \frac{\partial f}{\partial x_3}(b_1, b_2, \mathbf{x}_3)\Big) \quad f(b_1, b_2, b_3) - f(b_1, b_2, a_3) = M_3(b_3 - a_3)$$

En sommant, une simplification en cascade aboutit directement à :

$$\big(\exists M \in \mathbf{J}_{a,b}\big) \quad f(b) - f(a) = M(b-a). \quad \square$$

Il en découle l'extension suivante (en substituant x pour a et en notant que $b \in \mathbf{x}$) :

Proposition 2.8 (Extension de Hansen)
Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ continûment différentiable, $\mathbf{x} \in \mathbb{I}D$ et $x = (x_1, \ldots, x_n) \in \mathbf{x}$.

$$\mathbf{f}^{\square}(\mathbf{x}) \quad \subseteq \quad f(x) + \mathbf{J}(\mathbf{x}, x)(\mathbf{x} - x)$$

où $\mathbf{J}(\mathbf{x}, x)$ est la matrice intervalle définie ainsi : $\forall i \in [1..m]$, sa $i^{ème}$ ligne vaut

$$\mathbf{J}(\mathbf{x}, x)_i := \Big(\frac{\partial f_i}{\partial x_1}(\mathbf{x}_1, x_2, \ldots, x_n)\ \frac{\partial f_i}{\partial x_2}(\mathbf{x}_1, \mathbf{x}_2, \ldots, x_n)\ \ldots\ \frac{\partial f_i}{\partial x_n}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)\Big).$$

Remarque 2.5 *Ce développement ne coûte pas plus cher en théorie que celui de Taylor, mais en pratique ce n'est malheureusement pas toujours le cas : si on utilise une différentiation automatique [Gri00], les évaluations des dérivées partielles sur une boîte donnée $\mathbf{x}$ peuvent se faire simultanément, via 2 simples parcours d'arbre. Dans le cas de la formule de Hansen, chaque dérivée partielle se calcule sur une boîte différente, ce qui oblige à faire a priori 2 parcours par variable impliquée dans la composante f_i.*

2.6.6 Extension de Taylor (second ordre)

Considérons une fonction f à valeurs réelles, de classe C^2 et de gradient G. Prenons une boîte $\mathbf{x}$ incluse dans le domaine de définition de f. La formule des accroissements finis se généralise aux dérivées partielles d'ordre supérieure avec la formule de Taylor-Lagrange (ici donnée à l'ordre 2) :

$$\big(\forall a \in \mathbf{x}\big)\big(\forall b \in \mathbf{x}\big)\big(\exists c \in [a,b]\big) \quad f(b) = f(a) + G(a)(b-a) + \frac{1}{2}(b-a)^T H(c)(b-a),$$

où H désigne la *hessienne* de f (matrice $n \times n$ telle que $H_{ij} = \frac{\partial f^2}{\partial x_i \partial x_j}$). Cette formule conduit, par les mêmes raisonnements qu'à la section précédente, à la proposition suivante :

Proposition 2.9 (Extension de Taylor (ordre 2))
Soit $f : D \subset \mathbb{R}^n \to \mathbb{R}^m$ de classe C^2, J la jacobienne de f, H la hessienne de f, $\mathbf{x} \in \mathbb{I}D$ et $x \in \mathbf{x}$.

$$\mathbf{f}^{\square}(\mathbf{x}) \quad \subseteq \quad f(x) + J(x)(\mathbf{x}-x) + \frac{1}{2}(\mathbf{x}-x)^T \mathbf{H}^{\square}(\mathbf{x})(\mathbf{x}-x)$$

et de nouveau, la fonction

$$\begin{array}{rccl} \mathbf{f}_{t2}: & \mathbb{I}D & \to & \mathbb{IR}^m \\ & \mathbf{x} & \mapsto & f(x) + J(x)(\mathbf{x}-x) + \frac{1}{2}(\mathbf{x}-x)^T H(\mathbf{x})(\mathbf{x}-x) \end{array}$$

est une extension de f, de convergence quadratique.

On pourrait également envisager d'étendre la formule de Hansen à l'ordre 2.

2.7 Systèmes linéaires

En analyse par intervalles, lorsque nous parlons de *systèmes linéaires*, nous parlons toujours de *systèmes linéaires à coefficients intervalles*. En effet, les techniques d'intervalles n'apportent rien à la résolution d'un système linéaire habituel $Ax = b$ avec A et b réels, y compris pour la garantie des résultats. De très nombreuses méthodes existent et permettent de certifier un nombre arbitraire de bits corrects sur la solution.

Nous nous intéressons au cas où A et b ne sont pas connus exactement. Dans le cadre de cette thèse, nous nous limiterons au cas carré (A de taille $n \times n$).

Si on note $\mathbf{A}$ la matrice intervalle contenant l'ensemble des variations possibles de A, et $\mathbf{b}$ le vecteur intervalle des variations possibles de b, notre but est d'identifier l'ensemble $\Sigma(\mathbf{A}, \mathbf{b})$ suivant, appelé *ensemble solution* :

$$\Sigma(\mathbf{A}, \mathbf{b}) := \{x \in \mathbb{R}^n \mid \big(\exists A \in \mathbf{A}\big)\big(\exists b \in \mathbf{b}\big)\ Ax = b\}.$$

Par abus de langage, nous noterons par $\mathbf{A}x = \mathbf{b}$ le problème consistant à identifier $\Sigma(\mathbf{A}, \mathbf{b})$. Il s'agit d'un abus car un point x de $\Sigma(\mathbf{A}, \mathbf{b})$ ne vérifie pas en général $\mathbf{A}x = \mathbf{b}$ (vu comme équation). Exemple (en dimension 1) : $4 \in \Sigma([1,2],[2,4])$ puisque en prenant $x = 4$, $A = 1$ et $b = 4$ on a bien $Ax = b$; et $[1,2] \times 4 \neq [2,4]$.

En dimension 1, si la matrice $\mathbf{A}$ ne contient pas 0, $\Sigma(\mathbf{A}, \mathbf{b})$ est un simple intervalle. En dimension 2, les choses se compliquent déjà. La figure 2.4 montre graphiquement au travers d'exemples de complexité croissante, comment on obtient $\Sigma(\mathbf{A}, \mathbf{b})$ avec pour $\mathbf{A}$ la matrice identité "perturbée" sur chacun de ses coefficients par une incertitude ± 0.4. La figure 2.5 montre, pour la même matrice $\mathbf{A}$, d'autres ensembles solution obtenus en modifiant uniquement le vecteur $\mathbf{b}$.

On s'aperçoit que $\Sigma(\mathbf{A}, \mathbf{b})$, à défaut d'être une boîte, n'est même pas un ensemble convexe. Il s'avère qu'en augmentant la dimension, l'objet devient de plus en plus complexe à représenter. Nous verrons qu'il s'agit

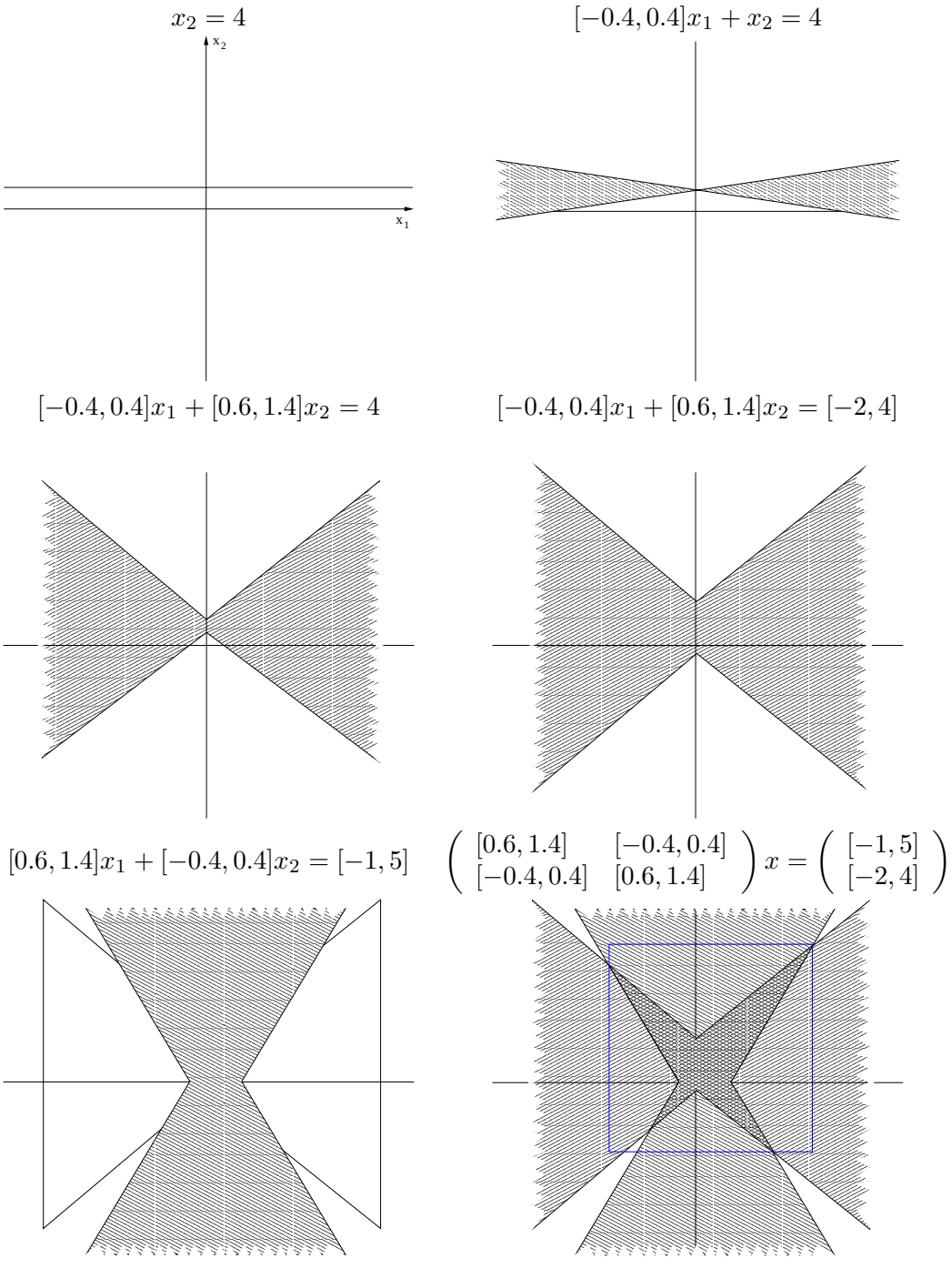


FIG. *2.4: Exemples de complexité croissante de systèmes linéaires.*

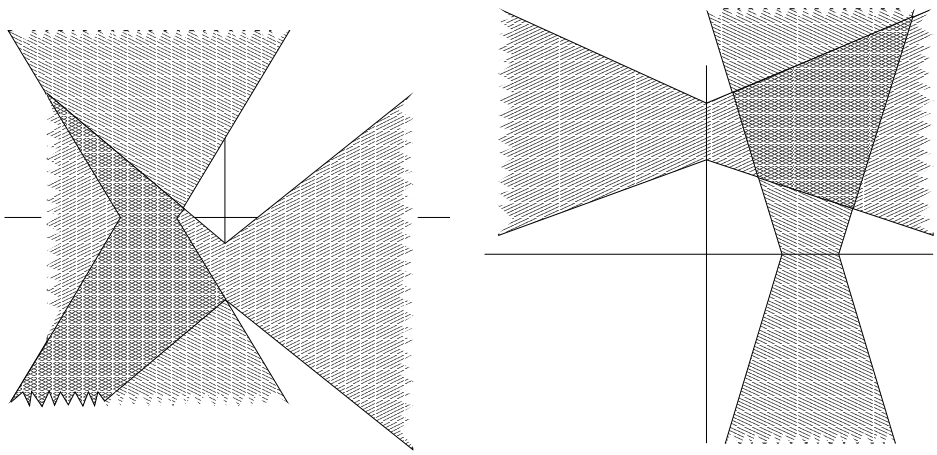


FIG. *2.5:* $\Sigma(\mathbf{A}, \mathbf{b})$ *obtenu avec d'autres vecteurs* $\mathbf{b}$.

d'un assemblage de 2^n polytopes convexes, chaque polytope étant obtenu en fixant le signe des variables. Dans la figure 2.4, on peut en effet observer que sur chaque quadrant (quart de plan) l'ensemble solution est un quadrilatère convexe.

Il serait envisageable d'utiliser un algorithme de bissection pour tenter de décrire $\Sigma(\mathbf{A}, \mathbf{b})$ avec de petites boîtes. En réalité, la résolution de systèmes linéaires intervalles n'est souvent qu'une étape dans un processus de résolution de systèmes non linéaires (nous y reviendrons au §2.8). Cette étape étant répétée un très grand nombre de fois, la rapidité prime la précision et le recours à un algorithme de bissection est exclu. On se contente de calculer en général un boîte qui englobe l'ensemble solution. Des algorithmes dédiés permettent de calculer de telles boîtes ; le chapitre 4 leur est consacré. Une boîte englobante est donc une boîte $\mathbf{B}$ vérifiant $\mathbf{B} \supseteq \square\Sigma(\mathbf{A}, \mathbf{b})$.

Notons qu'en général, $\square\,\Sigma(\mathbf{A}, \mathbf{b}) \subsetneq \mathbf{A}^{-1}\mathbf{b}$ avec $\mathbf{A}^{-1} := \square\{A^{-1},\ A \in \mathbf{A}\}$.

2.7.1 Caractérisations de $\Sigma(\mathbf{A}, \mathbf{b})$

Le lemme suivant offre deux autres façons de caractériser les solutions d'un système linéaire :

Lemme 2.2 (Beeck [Neu90])

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \iff \mathbf{A}x \cap \mathbf{b} \neq \emptyset \iff 0 \in \mathbf{A}x - \mathbf{b}$$

Preuve. Montrons la première équivalence. Supposons $x \in \Sigma(\mathbf{A}, \mathbf{b})$. Alors

$$\big(\exists A \in \mathbf{A}\big)\big(\exists b \in \mathbf{b}\big)\ Ax = b.$$

Puisque $Ax \in \mathbf{A}x$ et $b \in \mathbf{b}$ alors $Ax = b \Longrightarrow \mathbf{A}x \cap \mathbf{b} \neq \emptyset$.
Réciproquement, si $x \notin \Sigma(\mathbf{A}, \mathbf{b})$, alors

$$\big(\forall A \in \mathbf{A}\big)\big(\forall b \in \mathbf{b}\big)\ Ax \neq b$$

Et comme $\mathbf{A}x = \mathit{range}(Ax, A \in \mathbf{A})$ (il n'y a pas de dépendance !) alors $\mathbf{A}x \cap \mathbf{b} = \emptyset$. Montrons la seconde équivalence. Supposons $\mathbf{A}x \cap \mathbf{b} \neq \emptyset$. Il existe donc y tel que $y \in \mathbf{A}x$ et $y \in \mathbf{b}$. Par monotonie pour l'inclusion du calcul par intervalles (cf. proposition 2.2), $0 = y - y \in \mathbf{A}x - \mathbf{b}$. Réciproquement, $\mathbf{A}x - \mathbf{b} = \mathit{range}(Ax - b, A \in \mathbf{A}, b \in \mathbf{b})$ donc $0 \in \mathbf{A}x - \mathbf{b}$ implique

$$\big(\exists A \in \mathbf{A}\big)\big(\exists b \in \mathbf{b}\big)\ Ax - b = 0,$$

c.a.d., $x \in \Sigma(\mathbf{A}, \mathbf{b})$. □

Une autre célèbre (et en fait plus ancienne) caractérisation est due à Oettli & Prager [Oet65]. Supposons pour commencer que $x \geq 0$ (toutes les composantes de x sont positives). La formule de Beeck $\mathbf{A}x \cap \mathbf{b} \neq \emptyset$ se réécrit alors comme la conjonction

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \iff \left(\underline{\mathbf{A}}x \leq \overline{\mathbf{b}}\right) \text{ et } \left(\overline{\mathbf{A}}x \geq \underline{\mathbf{b}}\right). \tag{2.8}$$

Ceci s'obtient en observant simplement que l'intersection de deux intervalles $\mathbf{u}$ et $\mathbf{v}$ est non vide ssi $\underline{\mathbf{u}} \leq \overline{\mathbf{v}}$ et (symétriquement) $\overline{\mathbf{u}} \geq \underline{\mathbf{v}}$. En effet, appliquons cette observation pour chaque i $(1 \leq i \leq n)$ avec $\mathbf{u} := \mathbf{b}_i$ et $\mathbf{v} := (\mathbf{A}x)_i$. Comme x est positif,

$$\inf(\mathbf{A}x)_i = \inf(\sum_{j=1}^{n} \mathbf{A}_{ij}x_j) = \sum_{j=1}^{n} \inf(\mathbf{A}_{ij}x_j) = \sum_{j=1}^{n} \underline{\mathbf{A}_{ij}}x_j = \sum_{j=1}^{n} \underline{\mathbf{A}}_{ij}x_j = (\underline{\mathbf{A}}x)_i,$$

et de même, $\sup(\mathbf{A}x)_i = (\overline{\mathbf{A}}x)_i$, d'où (2.8).

Si x est maintenant quelconque, il est clair que $\inf(\mathbf{A}_{ij}x_j) = \underline{\mathbf{A}_{ij}}x_j$ si $x_j \geq 0$, $\overline{\mathbf{A}_{ij}}x_j$ sinon. Si on décompose $\mathbf{A}$ de la façon suivante : $\mathbf{A} = A \pm \Delta$ (avec $A = \text{mid}\,(\mathbf{A})$ et $\Delta = \text{rad}\,(\mathbf{A})$), on a donc $\inf(\mathbf{A}_{ij}x_j) = A_{ij} - sign(x_j)\Delta_{ij}$ où $sign(x_j) = 1$ si $x_j \geq 0$, -1 sinon.

Cette reformulation peut se réécrire directement sous forme matricielle. Introduisons la matrice diagonale Q, définie par $Q_{ii} = sign(x_i)$ (cette matrice dépend donc de x). On se convainc facilement qu'on a alors,

$$\inf(\mathbf{A}x) = (A - Q\Delta)x \quad \text{et} \quad \sup(\mathbf{A}x) = (A + Q\Delta)x.$$

Clairement, fixer Q revient à fixer le signe de chaque coordonnée donc à considérer les points x situés dans un *quadrant* (un quart de plan en dimension 2) :

Définition 2.9 (Quadrant)
Soit Q une matrice diagonale telle que $\forall i$ $(1 \leq i \leq n)$ $|Q_{ii}| = 1$.
L'ensemble $\{x \mid Qx \geq 0\}$ est appelé **quadrant** *de $\mathbb{R}^n$ (de matrice caractéristique Q).*

Pour alléger les notations, nous ne dissocions pas un quadrant de sa matrice caractéristique. Nous parlons donc de *quadrant Q* (Q étant une matrice), et parfois écrivons $x \in Q$ au lieu de $Qx \geq 0$ pour insister sur la nature "ensembliste" de cette condition. Enfin, on note que $x \in Q \iff Qx = |x|$. Par conséquent, nous convenons d'écrire $|x|$ au lieu de Qx, chaque fois que le quadrant de x est déterminé comme étant Q.

Finalement, nous avons obtenu la nouvelle caractérisation suivante :

Proposition 2.10 (Oettli-Prager)
Soient $A := \text{mid}\,(\mathbf{A})$, $\Delta := \text{rad}\,(\mathbf{A})$, $b := \text{mid}\,(\mathbf{b})$, $\delta := \text{rad}\,(\mathbf{b})$, Q un quadrant.

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \cap Q \iff \begin{cases} (A - \Delta Q)x \leq b + \delta \\ (A + \Delta Q)x \geq b - \delta \\ Qx \geq 0 \end{cases} \tag{2.9}$$

Ainsi, une équation linéaire à coefficients intervalles peut être récrite de façon équivalente en un système de $3 \times n$ inégalités linéaires scalaires dès lors que le signe des variables est fixé.

La formule d'Oettli-Prager donne déjà un moyen simple (mais trop coûteux) de calculer $\Box\,\Sigma(\mathbf{A}, \mathbf{b})$. Il s'agit de l'algorithme d'Aberth [Abe97, Bea97]. Cet algorithme calcule pour chaque quadrant Q la boîte minimale

englobant les solutions, c.a.d. $\square(\Sigma(\mathbf{A},\mathbf{b}) \cap Q)$, à l'aide d'un programme d'optimisation linéaire (algorithme du Simplexe ou point intérieur). Cette boîte s'obtient en minimisant et maximisant chaque composante x_i, sous le système de contraintes (2.9).
L'algorithme balaye les 2^n quadrants, et calcule donc pour chaque quadrant Q une boîte extérieure via $2n$ appels au Simplexe. Il fusionne ensuite (c.a.d., calcule l'enveloppe) des 2^n boîtes obtenues. Cette méthode est bien sûr trop lourde dans le cas général (elle nécessite $2^n \times 2n$ appels au Simplexe au total). Elle ne peut s'appliquer que lorsque l'on connaît *a priori* les quadrants comportant des solutions, et que le nombre de ces quadrants est petit.

Remarque : Les $2n$ premiers systèmes d'inéquations de (2.9) peuvent se regrouper en un seul :

$$(2.9) \iff \left\{ \begin{array}{l} Ax - \Delta|x| \leq b + \delta \\ -Ax - \Delta|x| \leq -b + \delta \\ Qx \geq 0 \end{array} \right. \iff \left\{ \begin{array}{l} |Ax - b| \leq \Delta|x| + \delta \\ Qx \geq 0 \end{array} \right.$$

On en déduit une forme compacte (mais moins lisible) du résultat d'Oettli-Prager :

Proposition 2.11 (Oettli-Prager)

$$x \in \Sigma(\mathbf{A},\mathbf{b}) \iff |(\text{mid } \mathbf{A})x - \text{mid } \mathbf{b}| \leq (\text{rad } \mathbf{A})|x| + \text{rad } \mathbf{b}.$$

2.7.2 Autres types de problèmes

D'autres types de problèmes existent bien sûr avec des matrices intervalles. L'un deux consiste à savoir si la matrice est régulière (c.a.d., si elle ne contient que des matrices régulières). Plus généralement, il peut être intéressant de caractériser les espaces propres d'une matrice intervalle. Enfin, le besoin de déterminer $\Sigma(\mathbf{A},\mathbf{b})$ avec $\mathbf{A}$ rectangulaire se pose dans de nombreuses applications. Des travaux fondateurs autour de ces questions sont dus à Rohn [Roh89, Roh02]. L'un des concept-clé de Rohn consiste à ramener l'étude d'une matrice intervalle à celles de ses matrices dites *extrêmes* (obtenues par combinaison des bornes supérieures et inférieures des coefficients). Des résultats probants sur la régularité des matrices jacobiennes en robotique ont été obtenus en robotique [MD06]. Ces travaux parviennent à utiliser les matrices extrêmes tout en exploitant la forme symbolique des jacobiennes.

2.8 Systèmes non linéaires

Nous considérons dans tout ce §2.8 un système carré, c.a.d. la donnée d'une fonction f de $\mathbb{R}^n$ dans $\mathbb{R}^n$.

2.8.1 Le théorème de Brouwer

L'une des premières améliorations à apporter à l'algorithme de bissection/évaluation du §2.5 est de prouver qu'il existe une solution dans une boîte retournée. Le théorème du point fixe de Brouwer est un premier outil qui permet de prouver l'existence de solution. Il s'énonce ainsi :

Théorème 2.4 (Brouwer) *Toute application f continue d'un compact de $\mathbb{R}^n$ (en particulier une boîte) dans lui-même admet un point fixe (c.a.d. un point x tel que $f(x) = x$).*

On en tire le test d'existence de solution suivant :

Corollaire 2.1 *Soient $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ continue, $\mathbf{x} \in \mathbb{I}D$ et $g : x \mapsto x - f(x)$.*

$$g(\mathbf{x}) \subseteq \mathbf{x} \Longrightarrow \left(\exists x \in \mathbf{x}\right)\ f(x) = 0.$$

Preuve. Tout d'abord, $g : x \mapsto f(x) - x$ est continue sur $\mathbf{x}$. Si $g(\mathbf{x}) \subseteq \mathbf{x}$ alors $range(g, \mathbf{x}) \subseteq \mathbf{x}$ donc, d'après le théorème de Brouwer, il existe $x \in \mathbf{x}$ tel que $g(x) = x$, c.a.d., tel que $f(x) = 0$.□

Il est bien sûr possible de remplacer $g(\mathbf{x})$ par n'importe quelle extension aux intervalles de g.

Le théorème de Brouwer n'est jamais appliqué tel quel. Si l'expression de g est de la forme $x - f(x)$, alors, quelle que soit la boîte $\mathbf{x}$, on a $\mathrm{rad}\,(g(\mathbf{x})) > \mathrm{rad}\,(\mathbf{x})$ dès que $f(\mathbf{x})$ a un rayon non-nul, et l'inclusion ne peut donc jamais être vérifiée. Le théorème requiert donc déjà une simplification formelle de $x - f(x)$, c'est à dire la possibilité d'isoler une occurrence de x dans l'expression de la fonction f. D'autre part, même dans ce cas de figure, le pessimisme rend souvent impossible une inclusion telle que $g(\mathbf{x}) \subseteq \mathbf{x}$. Ce théorème est par contre la pierre angulaire d'autres outils plus élaborés, comme le test d'existence de solutions de Newton.

2.8.2 Opérateur de Newton

Nous avons vu au §2.6.4 que le calcul d'une estimation extérieure $\mathbf{J}$ de $\mathbf{J}^{\square}(\mathbf{x})$ - matrice intervalle contenant l'ensemble des jacobiennes sur la boîte $\mathbf{x}$ - pouvait servir à améliorer l'évaluation d'une fonction. A l'aide de cette évaluation, il est également possible de filtrer la boîte $\mathbf{x}$, c'est à dire de réduire ses bornes sans perdre de solution. On peut appliquer pour cela l'*opérateur de Newton*. Cet opérateur repose sur l'approximation extérieure suivante de l'ensemble des solutions contenues dans une boîte :

Proposition 2.12
Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ une fonction continûment différentiable, $\mathbf{x} \in \mathbb{I}D$ et $\mathbf{J}$ une matrice intervalle $n \times n$ telle que $\mathbf{J} \supseteq \mathbf{J}^{\square}(\mathbf{x})$. Fixons un point $\tilde{x}$ dans $\mathbf{x}$.

$$\left(\forall x \in \mathbf{x}\right) \quad f(x) = 0 \Longrightarrow x \in \tilde{x} + \Sigma(\mathbf{J}, -f(\tilde{x})).$$

Preuve. Soit $x \in \mathbf{x}$ tel que $f(x) = 0$. D'après la proposition 2.5, $\left(\exists M \in \mathbf{J}^{\square}(\mathbf{x})\right)\ 0 - f(\tilde{x}) = M(x - \tilde{x})$. Ceci implique $x \in \tilde{x} + \Sigma(\mathbf{J}^{\square}(\mathbf{x}), -f(\tilde{x}))$. On conclut simplement en notant que $\Sigma(\mathbf{J}^{\square}(\mathbf{x}), -f(\tilde{x})) \subseteq \Sigma(\mathbf{J}, -f(\tilde{x}))$. □

Remarquons que la proposition peut s'appliquer même dans le cas où $\mathbf{J}$ n'est pas régulière. Toutefois, le cas échéant, les chances de filtrer la boîte $\mathbf{x}$ sont limitées soit parce que $\Sigma(\mathbf{J}, -f(\tilde{x}))$ est non borné (il y a donc au moins une dimension suivant laquelle on ne peut faire de réduction), soit parce que la méthode utilisée pour calculer une approximation de $\Sigma(\mathbf{J}, -f(\tilde{x})) \cap \mathbf{x}$ échoue en présence d'une matrice singulière (ex : la méthode de Hansen-Bliek, cf. p.100). La figure 2.6 représente graphiquement le filtrage dans le cas d'une fonction réelle.

Définition 2.10 (Opérateur de Newton) *Étant donnée une fonction f, l'opérateur de Newton N est défini de la façon suivante. Pour toute boîte $\mathbf{x}$,*

$$N(\mathbf{x}) := \tilde{x} + \Sigma(\mathbf{J}, -f(\tilde{x}))$$

où $\mathbf{J}$ est une approximation extérieure de $\mathbf{J}^{\square}(\mathbf{x})$, $\tilde{x}$ un point de $\mathbf{x}$. Cet opérateur possède plusieurs paramètres :
- *Une extension aux intervalles pour le calcul de $\mathbf{J}^{\square}(\mathbf{x})$ (ex., $\mathbf{J} := J(\mathbf{x})$).*
- *Une stratégie de choix[6] du point $\tilde{x}$ dans $\mathbf{x}$ (ex., $\tilde{x} := \mathrm{mid}\,(\mathbf{x})$).*
- *Une méthode de calcul de l'approximation extérieure de $\Sigma(\mathbf{J}, -f(\tilde{x}))$. En effet, on a vu au §2.7 que le calcul des solutions de $\Sigma(\mathbf{A}, \mathbf{b})$, avec $\mathbf{A}$ et/ou $\mathbf{b}$ intervalle, était un problème en soi.*

[6] Comme nous l'avons déjà observé pour l'extension de Taylor, si le point $\tilde{x}$ dépend de $\mathbf{x}$, il en résulte que l'opérateur de Newton n'est pas monotone pour l'inclusion.

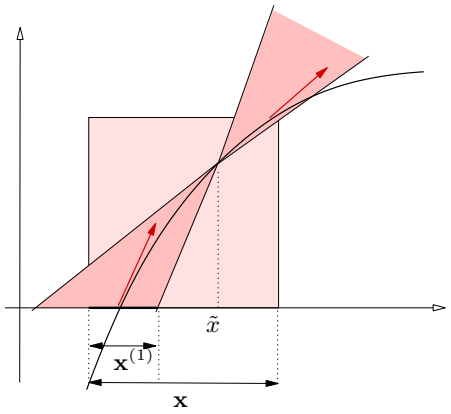


FIG. *2.6: Filtrage de Newton (cas de $\mathbb{R}$ dans $\mathbb{R}$). Le cône englobe toutes les pentes de la fonction sur la boîte $\mathbf{x}$. L'intervalle $\mathbf{x}^{(1)}$ situé à l'intersection de ce cône et de l'axe des abscisses contient l'unique solution. Le calcul effectué pour obtenir cet intervalle est bien $\tilde{x} - f(\tilde{x})/f'(\mathbf{x})$.*

Parmi les variantes de l'opérateur de Newton, mentionnons la méthode de **Moore** [Moo66] qui utilise une matrice inverse de $J(\mathbf{x})$ (cf. §2.7), et la méthode de **Hansen-Sengupta** [HS80], qui repose sur l'algorithme de Gauss-Seidel (cf. chapitre 4) et une arithmétique fermée des intervalles (cf. chapitre 6). La méthode de Moore nécessite que $J(\mathbf{x})$ soit régulière, contrairement à celle de Hansen-Sengupta.

Le filtrage de Newton consiste donc à appliquer itérativement l'opérateur

$$\mathbf{x} \leftarrow N(\mathbf{x}) \cap \mathbf{x} \tag{2.10}$$

jusqu'à ce que l'amélioration produite sur la boîte $\mathbf{x}$ soit en deçà d'un seuil prédéterminé.

Remarque 2.6 *De nombreuses optimisations sont étudiées par Hansen [Han92b]. Ayant calculé $J(\tilde{x})^{-1}$, une idée est d'appliquer une méthode de "la corde" avec $J(\tilde{x})^{-1}$ en arithmétique classique pour approcher le zéro de f. Ce zéro est ensuite utilisé comme estimée initiale, c'est à dire à la fois comme point de préconditionnement pour $\mathbf{J}$ (voir chapitre 4) et comme point d'expansion $\tilde{x}$ dans la formule de la prop. 2.12 (au lieu du milieu). Pour gagner du temps, il est possible également d'utiliser la première jacobienne calculée pour les autres itérations de* (2.10) *(même idée que pour la méthode de la corde). Enfin, si l'on utilise $|f(\mathbf{x})| < \epsilon_f$ comme critère d'arrêt (cf. remarque 2.2), au lieu d'utiliser la forme naturelle, on peut utiliser la forme de Taylor pour évaluer f sur $\mathbf{x}$ puisqu'on dispose déjà de $f(\tilde{x})$, de $\mathbf{J}$ et de $(\mathbf{x} - \tilde{x})$.*

2.8.3 Existence et Unicité

Supposons dorénavant que la jacobienne est régulière. Au delà du filtrage, l'opérateur de Newton permet de certifier l'existence et l'unicité d'une solution. Comme précédemment, considérons une fonction $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ continûment différentiable, et $\mathbf{x} \in \mathbb{I}D$. Fixons de même un point $\tilde{x}$ dans $\mathbf{x}$. L'opérateur de Newton s'appuie sur la proposition 2.5, dont nous rappelons l'énoncé :

$$\big(\forall x \in \mathbf{x}\big)\big(\exists M \in J(\mathbf{x})\big) \quad f(x) - f(\tilde{x}) = M(x - \tilde{x}). \tag{2.11}$$

Pour prouver l'existence d'une solution, nous avons de nouveau besoin de cette formule, mais également d'une information supplémentaire de continuité. La proposition suivante comporte cette information, et remplace donc la proposition 2.5.

Proposition 2.13
Avec les mêmes notations que précédemment, il existe une fonction $M : \mathbf{x} \mapsto \mathbf{J}^{\square}(\mathbf{x})$ continue telle que

$$\big(\forall x \in \mathbf{x}\big) \quad f(x) = f(\tilde{x}) + M(x)(x - \tilde{x}).$$

Nous avons utilisé le théorème des accroissements finis pour établir la proposition 2.5. Ce théorème nous permet de garantir, pour chaque x, l'existence d'un M satisfaisant la relation ; mais il ne donne en revanche aucune information sur ce M. En particulier, il ne permet pas de savoir si M varie continûment en fonction de x. La preuve utilise donc un théorème qui explicite M : le théorème de Taylor-Young avec reste intégral, à l'ordre 1.

Preuve. (i) Construisons M.

Fixons x et posons $\phi_i : [0,1] \longrightarrow \mathbb{R}^m$ telle que $\phi_i(t) = f(\tilde{x} + t(x - \tilde{x}))$.
On a $f(x) - f(\tilde{x}) = \phi(1) - \phi(0) = \int_0^1 \phi'(t)dt$ (*théorème fondamental de l'analyse*). En appliquant la règle de composition, il vient $\phi'(t) = J(\tilde{x} + t(x - \tilde{x}))(x - \tilde{x})$. On pose alors

$$M(x) = \int_0^1 J(\tilde{x} + t(x - \tilde{x}))dt.$$

Ainsi, M est une fonction de $\mathbb{R}^n$ dans $\mathbb{R}^{m \times n}$ qui vérifie bien

$$\big(\forall x \in \mathbf{x}\big) \quad f(x) = f(\tilde{x}) + M(x)(x - \tilde{x}).$$

Il s'agit de la formule de *Taylor-Young avec reste intégral*.

(ii) Prouvons que $M(x) \in J(\mathbf{x})$. Comme f est continûment différentiable, ϕ l'est également. Donc $\phi_1', \ldots, \phi_m'$ sont m fonctions continues de $\mathbb{R}$ dans $\mathbb{R}$. Pour tout $i \in [1..m]$, on a alors

$$\big(\exists c_i \in [0,1]\big) \quad \phi_i'(c_i) = \int_0^1 \phi'(t)dt.$$

Cela provient du fait que $\inf_{t\in[0,1]} \phi_i'(t) \leq \int_0^1 \phi_i'(t)dt \leq \sup_{t\in[0,1]} \phi_i'(t)$ et que, par continuité, ϕ_i' atteint ses bornes sur le compact $[0,1]$. Par définition de l'intégrale, on a alors $(\phi_1'(c_1), \ldots, \phi_m'(c_m)) = \int_0^1 \phi'(t)dt$, or $\phi_i'(c_i) = J_i(x + c_i(y - x))(y - x)$, en notant J_i la $i^{ème}$ ligne de la jacobienne de f. Donc en posant $x_i = x + c_i(y - x)$, on a $\int_0^1 \phi'(t)dt = (J_1(x_1), \ldots, J_m(x_m))(y - x)$.

On termine la preuve en notant simplement que le vecteur $(J_1(x_1), \ldots, J_m(x_m)) = \Big((J(x_1))_1, \ldots, (J(x_m))_m\Big)$ appartient à la boîte $\mathbf{J}^{\square}(\mathbf{x})$, en vertu de la proposition 2.1 p.13.

(iii) Prouvons que M est continue.

Il est bien connu que la valeur d'une intégrale varie continûment en fonction d'un paramètre x, si la fonction intégrée est elle-même continue en x (ce qui est le cas puisque f est continûment différentiable). □

Proposition 2.14 (Test d'existence) *Supposons que $\mathbf{J}^{\square}(\mathbf{x})$ soit régulière et posons $\mathbf{x}' := N(\mathbf{x})$.*
Si $\mathbf{x}' \subseteq \mathbf{x}$ alors il existe (au moins) une solution à $f(x) = 0$ dans $\mathbf{x}'$.

Preuve. La proposition 2.13 indique qu'il existe une fonction continue $M : \mathbf{x} \mapsto \mathbf{J}^{\square}(\mathbf{x})$ telle que

$$\big(\forall x \in \mathbf{x}\big)\ f(x) - f(\tilde{x}) = M(x)(x - \tilde{x}). \tag{2.12}$$

On en déduit

$$\big(\forall x \in \mathbf{x}\big)\ x - M(x)^{-1} f(x) = \tilde{x} + M(x)^{-1}(-f(\tilde{x})). \tag{2.13}$$

Posons maintenant $g : x \mapsto \tilde{x} + M(x)^{-1}(-f(\tilde{x}))$. On a $\mathbf{x}' \supseteq \tilde{x} + \square\,\Sigma(\mathbf{J}^{\square}(\mathbf{x}), -f(\tilde{x}))$ donc $range(g, \mathbf{x}) \subseteq \mathbf{x}'$. Par hypothèse, $\mathbf{x}' \subseteq \mathbf{x}$ donc $range(g, \mathbf{x}) \subseteq \mathbf{x}$. M étant continue, $x \mapsto (M(x))^{-1}$ l'est également (via les formules de Cramer), et finalement g est continue. Le théorème de Brouwer s'applique alors et prouve qu'il existe $x \in \mathbf{x}'$ tel que $x = g(x)$, c.a.d., $x = \tilde{x} + M(x)^{-1}(-f(\tilde{x}))$. On obtient alors avec (2.13) que $x - M(x)^{-1}f(x) = x$, ce qui implique $f(x) = 0$. □

Insistons sur deux différences importantes entre la partie *filtrage* et la partie *test d'existence* de l'opérateur de Newton. Dans les deux cas, le calcul de $N(\mathbf{x})$ fait intervenir un système linéaire $\mathbf{A}x = \mathbf{b}$ (avec $\mathbf{A} \supseteq \mathbf{J}^{\square}(\mathbf{x})$ et $\mathbf{b} = -f(\tilde{x})$). Mais pour le test d'existence, d'une part $\mathbf{A}$ doit être régulière, d'autre part l'ensemble calculé est $\Sigma(\mathbf{A}, \mathbf{b})$ lui-même et non $\Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x}$, c'est à dire les solutions du système restreintes à une boîte $\mathbf{x}$ fixée.

En pratique, on ne cherche pas à déterminer si $\mathbf{A}$ est régulière : on se contente en général de tester une propriété plus forte appelée *forte régularité*, que nous détaillerons au chapitre 4. Une erreur consiste à vouloir se passer du test de régularité en tentant directement d'approximer extérieurement $\Sigma(\mathbf{A}, \mathbf{b})$ et en concluant, en cas de succès, que $\mathbf{A}$ est régulière. Cette conclusion repose sur l'idée qu'un ensemble solution $\Sigma(\mathbf{A}, \mathbf{b})$ est non borné dès lors que $\mathbf{A}$ contient une matrice singulière, ce qui n'est pas tout à fait vrai car l'ensemble peut également être vide[7]. Dans ce dernier cas, une boîte extérieure peut très bien être fournie, et si cette boîte extérieure est non-vide, rien ne permet a priori de distinguer ce cas de celui d'une matrice régulière. On pourrait toutefois contourner le problème de plusieurs manières, en calculant par exemple une boîte intérieure (cf. §5.2 p.113 et §5.6 p.118) : l'obtention d'une boîte intérieure non vide et d'une boîte extérieure de $\Sigma(\mathbf{A}, \mathbf{b})$ prouve bien cette fois que $\mathbf{A}$ est régulière.

On ne cherche pas non plus en pratique à calculer $\Sigma(\mathbf{A}, \mathbf{b})$ puis à vérifier $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}$. On calcule $\Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x}$ (ce qui est plus simple, cf. chap.4) et on vérifie $\Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x} \subset \mathbf{x}$. Comme $\Sigma(\mathbf{A}, \mathbf{b})$ est connexe, cette dernière inclusion (stricte) implique bien $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}$.

La proposition 2.14 concerne l'*existence* d'une solution dans $\mathbf{x}$, mais la régularité de $\mathbf{J}^{\square}(\mathbf{x})$ permet également de certifier l'*unicité* de cette solution :

Proposition 2.15 (Test d'unicité)
Si $\mathbf{J}^{\square}(\mathbf{x})$ est régulière, $f(x) = 0$ possède au plus une solution dans $\mathbf{x}$.

Preuve. Pour tout x et y dans $\mathbf{x}$, si $f(y) = f(x)$ alors $f(y) - f(x) = 0$. D'après (2.11), $\big(\exists M \in \mathbf{J}^{\square}(\mathbf{x})\big)$ tel que $f(y) = f(x) + M(y - x)$, et $M(y - x) = 0$. Comme M est régulière, $M(x - y) = 0 \Longrightarrow x = y$. On a prouvé que f est injective. En particulier, il ne peut y avoir au plus qu'un x tel que $f(x) = 0$. □

2.8.4 Opérateur de Newton-Hansen

L'opérateur de Newton et les tests d'existence et d'unicité ont été construits à partir de la proposition 2.5, c'est à dire la forme de Taylor. Il est possible de faire le même travail, mais en partant de la proposition 2.7. Au lieu d'utiliser une approximation de $\mathbf{J}^{\square}(\mathbf{x})$, on utilise $J(\mathbf{x}, \tilde{x})$, la matrice « de Hansen » telle que définie à la proposition 2.8. Comme cette dernière est en général nettement plus fine, cela se traduit par une bien meilleure performance de l'opérateur de Newton. Toujours avec les mêmes notations, les résultats des 2 sections précédentes se déclinent ainsi :

Proposition 2.16
Si $\mathbf{J}(\mathbf{x}, \tilde{x})$ est régulière, posons $N(\mathbf{x}) := \tilde{x} + \Sigma(\mathbf{J}(\mathbf{x}, \tilde{x}), -f(\tilde{x}))$. Alors,

(i) $\big(\forall x \in \mathbf{x}\big)$ $f(x) = 0 \Longrightarrow x \in N(\mathbf{x})$,
(ii) $N(\mathbf{x}) \subseteq \mathbf{x} \Longrightarrow$ *il existe une unique solution à $f(x) = 0$ dans $N(\mathbf{x})$.*

[7] Prenons l'exemple d'une matrice A réelle. Si A est singulière alors $range(A) \subsetneq \mathbb{R}^n$, donc en choisissant $\mathbf{b}$ tel que $\mathbf{b} \cap range(A) = \emptyset$, on a bien $\Sigma(A, \mathbf{b}) = \emptyset$. Remarquons néanmoins que pour une matrice $\mathbf{A}$ intervalle, si $\mathbf{A}$ contient une seule matrice régulière alors $\Sigma(\mathbf{A}, \mathbf{b})$ ne peut pas être vide.

La preuve pour (i) est une copie de celle de la proposition 2.12 (elle sera donnée au §2.9.3 dans une version paramétrée). La preuve de la partie "existence" de (ii) est une copie de celle de la proposition 2.14, mais il faut redémontrer la proposition 2.13 avec $\mathbf{J}(\mathbf{x}, \tilde{x})$ à la place de $\mathbf{J}^{\square}(\mathbf{x})$, en s'inspirant de la preuve de la proposition 2.7. La partie "unicité" de (ii) paraît plus délicate (pas de preuve proposée).

2.8.5 Opérateur de Krawczyk

La méthode de Newton échoue souvent en présence de singularité et ne produit rien. Nous survolons rapidement dans ce paragraphe une alternative possible : l'opérateur de Krawczyk.

Nous nous appuyons de nouveau sur la proposition 2.5. Mais plutôt que de se ramener à un système linéaire par intervalles comme dans le cas de la preuve de la proposition 2.12, nous décomposons M en $(M - I) + I$. Cette décomposition permet également d'isoler x, ce que montre la chaîne d'implications suivante :

$$\begin{array}{rcll} f(x) = 0 & \Longrightarrow & \big(\exists M \in \mathbf{J}^{\square}(\mathbf{x})\big) & 0 - f(\tilde{x}) = M(x - \tilde{x}) \\ & \Longrightarrow & \big(\exists M \in \mathbf{J}^{\square}(\mathbf{x})\big) & -f(\tilde{x}) = x - \tilde{x} + (M - I)(x - \tilde{x}) \\ & \Longrightarrow & \big(\exists M \in \mathbf{J}^{\square}(\mathbf{x})\big) & x = \tilde{x} - f(\tilde{x}) + (I - M)(x - \tilde{x}). \end{array}$$

On obtient alors $f(x) = 0 \Longrightarrow x \in \tilde{x} - f(\tilde{x}) + (I - \mathbf{J})(\mathbf{x} - \tilde{x})$ où $\mathbf{J}$ est une matrice intervalle quelconque vérifiant $\mathbf{J} \supseteq \mathbf{J}^{\square}(\mathbf{x})$. Il en découle le filtrage suivant :

$$\mathbf{x} \leftarrow \tilde{x} - f(\tilde{x}) + (I - \mathbf{J})(\mathbf{x} - \tilde{x}),$$

qui s'applique sans condition particulière pour $\mathbf{J}$. Toutefois, la décomposition a introduit deux occurrences de x dont la décorélation signifie automatiquement une perte de précision par rapport à l'opérateur de Newton. Ainsi, en règle générale, pour une fonction quelconque le rayon du vecteur d'intervalles calculé en partie droite est supérieur à celui de $\mathbf{x}$. Le filtrage n'est effectif que lorsque $I - \mathbf{J}$ est proche de la matrice nulle, c'est à dire lorsque $\mathbf{J}$ est proche de I. Or, nous verrons au §4.3 p.87 qu'il est possible de *préconditionner* $\mathbf{J}$, c'est à dire de la multiplier par une matrice réelle C bien choisie de telle sorte que $C\mathbf{J} \sim I$, dans un sens que nous préciserons. Comme $f(x) = 0$ implique $Cf(x) = 0$, la chaîne d'implication précédente peut être reproduite de façon similaire en partant de $Cf(x) = 0$ pour aboutir au filtrage suivant :

$$\mathbf{x} \leftarrow \tilde{x} - Cf(\tilde{x}) + (I - C\mathbf{J})(\mathbf{x} - \tilde{x})$$

qui est cette fois opérationnel. Ce filtrage définit l'opérateur de Krawczyk. Remarquons que l'opérateur de Newton perd le minimum d'information à partir de la proposition 2.5 ; il reste donc meilleur, si tant est qu'il puisse être appliqué. L'opérateur de Krawczyk donne également lieu à un test d'existence et d'unicité :

Proposition 2.17 (Opérateur de Krawczyk) *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ une fonction continûment différentiable, $\mathbf{x} \in \mathbb{I}D$ et $\tilde{x} \in \mathbf{x}$. Soit enfin $\mathbf{J} \in \mathbb{IR}^{n \times n}$ telle que $\mathbf{J} \supseteq \mathbf{J}^{\square}(\mathbf{x})$, où J désigne la jacobienne de f.*

Posons $K(\mathbf{x}) := \tilde{x} - Cf(\tilde{x}) + (I - C\mathbf{J})(\mathbf{x} - \tilde{x})$. Alors,

$$\begin{array}{ll} \textit{(filtrage)} & f(x) = 0 \Longrightarrow x \in K(\mathbf{x}), \\ \textit{(test d'existence)} & K(\mathbf{x}) \subseteq \mathbf{x} \Longrightarrow \big(\exists x \in \mathbf{x}\big)\ f(x) = 0, \\ \textit{(test d'unicité)} & K(\mathbf{x}) \subset \mathbf{x} \Longrightarrow \big(\exists x \textit{ unique } \in \mathbf{x}\big)\ f(x) = 0. \end{array}$$

Preuve. Voir [AM00, Neu90]. □

Notons que contrairement à l'opérateur de Newton, l'unicité nécessite une inclusion stricte.

2.8.6 Opérateur de Kantorovitch, Inflation de boîtes

Il existe également un test d'existence et d'unicité de solutions particulièrement efficace, basé sur le théorème de Kantorovitch [OR70, Tap71, DM73]. Ce test fonctionne d'une manière différente de ceux présentés auparavant : au lieu de prouver l'existence d'une solution dans une boîte, il prend également un point $\tilde{x}$ en entrée et construit une boîte autour de ce point, inclus dans la boîte initiale. La boîte construite vérifie alors deux propriétés cruciales : d'une part, il est garanti qu'une solution unique au problème existe dans cette boîte et, d'autre part, qu'une méthode de Newton classique (non intervalle) convergera vers cette solution à partir du point initial $\tilde{x}$.

Nous énonçons ce test dans une forme particulière, adaptée aux intervalles. L'énoncé nécessite de définir au préalable quelques notions :
- On note $\|\cdot\|$ la norme de $\mathbb{R}^n$ définie pour tout $x \in \mathbb{R}^n$ ainsi : $\|x\| = \max_{i=1..n} |x_i|$..
- On note également $\|\cdot\|$ la norme de $\mathbb{R}^{n\times n}$ définie pour tout $M \in \mathbb{R}^{n\times n}$ ainsi : $\|M\| = \max_{i=1..n} \sum_{j=1..n} |M_{ij}|$.
- On appelle **cube** une boîte dont le rayon r est le même sur chaque dimension :
  $\mathbf{x}$ est un cube si $\forall i,j \in [1..n],\ \text{rad}\,(\mathbf{x}_i) = \text{rad}\,(\mathbf{x}_j) = r$ (on appelle alors **rayon** du cube $\mathbf{x}$ le réel r).

Théorème 2.5 (Kantorovitch) *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ une fonction de classe C^2, $\mathbf{x} \in \mathbb{I}D$ et $x^{(0)} \in \mathbf{x}$. Notons r le rayon du plus grand cube centré en $x^{(0)}$ et inclus dans $\mathbf{x}$. Supposons que la jacobienne $J(x^{(0)})$ soit inversible et notons $J_0 := J(x^{(0)})$. S'il existe trois constantes réelles A_0, B_0 et C telles que*

1. $\|J_0^{-1}\| \leq A_0$
2. $\|J_0^{-1} f(x^{(0)})\| \leq B_0 \leq r/2$
3. $\forall i \in [1..n],\ \forall j \in [1..n], \quad \sum_{k=1}^{n} |\frac{\partial^2 f_i(\mathbf{x})}{\partial x_j \partial x_k}| \leq C$
4. $nA_0B_0C \leq 1/2$

alors, d'une part il existe une unique solution à $f(x) = 0$ dans le cube centré en $x^{(0)}$ et de rayon $2B_0$. D'autre part, la méthode de Newton (réelle) initialisée avec $x^{(0)}$ convergera vers cette solution.

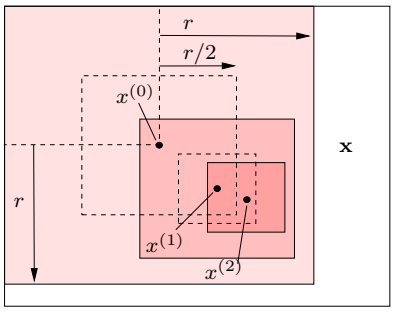
Preuve. Voir [DM73] (les grandes lignes de la preuve sont données ci-dessous). □

Considérons la suite obtenue par la méthode de Newton (réelle) à partir de $x^{(0)}$:

$$x^{(k+1)} := x^{(k)} - J(x^{(k)})^{-1} f(x^{(k)}). \tag{2.14}$$

La preuve du théorème consiste à établir les deux points suivants :
(i) Le point $x^{(1)}$ est situé dans le cube centré en $x^{(0)}$ et de rayon $r/2$.
(ii) Le cube centré en $x^{(1)}$ et de rayon $r/2$ vérifie de nouveau les conditions du théorème.

On en déduit alors facilement par récurrence que la suite des $x^{(k)}$ est une suite de Cauchy, donc convergente, contenue dans $\mathbf{x}$ (voir figure ci-dessous).



Du fait que f soit continue et que sa jacobienne soit continue et bornée sur $\mathbf{x}$, on montre enfin à partir de la relation (2.14) que la limite x^∞ de cette suite vérifie bien $f(x^\infty) = 0$.

Remarquons que le point (i) de la preuve est facile à montrer. En effet, par une application directe de (2.14), on a $\|x^{(1)} - x^{(0)}\| = \|J_0^{-1} f(x^{(0)})\| \leq B_0$, grâce à l'hypothèse 2. Comme $B_0 \leq r/2$ par cette même hypothèse, le point $x^{(1)}$ appartient bien au cube centré en $x^{(0)}$ et de rayon $r/2$.

Le point (ii) est plus technique. Tout d'abord, l'hypothèse 3 permet de borner la norme de la dérivée seconde de f sur $\mathbf{x}$, et cette borne permet de faire deux encadrements : d'une part, le théorème de la moyenne appliqué à la fonction $x \mapsto J(x)$ donne en notant $J_1 := J(x^{(1)}) : \|J_1 - J_0\| \leq nC_0\|x^{(1)} - x^{(0)}\|$, c'est à dire, $\|J_1 - J_0\| \leq nC_0B_0$. D'autre part, une version « à l'ordre 2 » du théorème de la moyenne appliquée à f permet d'écrire

$$\|f(x^{(1)}) - f(x^{(0)}) - J_0(x^{(1)} - x^{(0)})\| \leq 1/2nC_0\|x^{(1)} - x^{(0)}\|^2, \quad \text{c'est à dire,} \quad \|f(x^{(1)})\| \leq 1/2nC_0B_0^2.$$

Posons maintenant $M := J_0^{-1}(J_0 - J_1)$. On a $\|M\| \leq \|J_0^{-1}\|\|J_0 - J_1\|$ (car la norme $\|.\|$ vérifie la définition 4.1 p.81), donc $\|M\| \leq nA_0B_0C$. D'après l'hypothèse 4, on en déduit $\|M\| \leq 1/2$. Nous verrons plus loin (cf. prop. 4.2 p.81) que cela implique que $I - M$ est inversible, d'inverse $\sum_{k\geq 0} M^k$. On en conclut que $\|(I - M)^{-1}\| \leq 1/(1 - \|M\|)$, ce qui se réécrit $\|J_1^{-1}J_0\| \leq 2$ par définition de M. Finalement, J_1 est inversible et $\|J_1^{-1}\| = \|J_1^{-1}J_0J_0^{-1}\| \leq \|J_1^{-1}J_0\|\|J_0^{-1}\| \leq 2A$. De plus, $\|J_1^{-1}f(x^{(1)})\| \leq \|J_1^{-1}\|\|f(x^{(1)})\| \leq nA_0CB_0^2$. En posant $A_1 := 2A_0$ et $B_1 := nA_0CB_0^2$, un calcul direct montre alors que $nA_1B_1C \leq 1/2$. Les hypothèses du théorème sont donc de nouveau vérifiées pour le cube $x^{(1)}$ de rayon $r/2$.

Une version plus puissante de ce théorème a été obtenue dans le cas de fonctions quadratiques [Mer04]. La version obtenue permet d'une part d'affaiblir considérablement les conditions d'applicabilité : il est montré par exemple, que n peut être remplacé par une constante indépendante du nombre de variables (cette constante dans le cas d'équations de distance correspond à la dimension géométrique du problème, c.a.d., 2 ou 3). D'autre part, la boîte fournie est plus grande qu'avec une application du théorème dans sa forme générale.

D'autres tests d'existence de solutions existent en analyse par intervalles, basés par exemple sur le théorème de Miranda [Roh80], le degré topologique [FHL04] ou le théorème de Borsuk [FL04]. Des comparaisons ont pu être établies entre ces différents tests [FLS04, AFHM05, Gol07a].

Ajoutons enfin que le test d'unicité (cf. proposition 2.15) peut servir de base à une technique dite d'*inflation de boîtes* [Neu90]. En effet, l'unicité dans une boîte $\mathbf{x}$ repose uniquement sur la régularité de $\mathbf{J}^\square(\mathbf{x})$. Par conséquent, si un test (Newton, Krawczyk, Kantorovitch, etc.) a permis de prouver l'existence d'une solution dans une boîte et si en élargissant cette boîte il est possible de prouver que la jacobienne reste régulière, alors la boîte élargie contient bien, elle aussi, une unique solution.

2.9 Paramètres

Dans l'introduction de cette thèse (cf. chapitre 1), nous avons motivé l'intérêt d'ajouter des paramètres dans un système d'équations. Nous avons également vu que résoudre un système ayant des paramètres pouvait avoir plusieurs significations. Nous retenons ici la signification du §1.1.1, c'est à dire « sans quantificateur ». En effet, l'analyse par intervalles classique semble moins adaptée pour le filtrage dans le cadre des AE-systèmes (cf. §1.1.2). Le chapitre 3 décrira une approche, basée sur un autre modèle, qui visera à traiter les AE-systèmes.

On suppose donnée une fonction

$$f : \begin{array}{ccc} (D \times \mathbf{p}) \subset \mathbb{R}^n \times \mathbb{R}^k & \rightarrow & \mathbb{R}^n \\ (x, p) & \mapsto & f(x, p) \end{array} \tag{2.15}$$

où x représente les variables du système, p les paramètres et $\mathbf{p}$ les domaines de paramètres. Si on reproduit (1.3) avec nos notations, le vecteur x est une solution du système paramétré $f(x, p) = 0$ si

$$\big(\exists p \in \mathbf{p}\big) \quad f(x, p) = 0. \tag{2.16}$$

Remarquons que les systèmes linéaires à coefficients intervalles (cf. §2.7) sont des cas particuliers de systèmes paramétrés selon (2.16).

2.9.1 Continuums de solutions

Clairement, si pour une valeur fixée des paramètres, le système $f(x, p) = 0$ admet des solutions ponctuelles, en faisant varier p, chaque point-solution se transforme en un volume (en général de dimension k) qu'on appelle *continuum*. Nous nous trouvons donc presque systématiquement en présence d'une infinité de solutions. Néanmoins, il faut clairement distinguer ce type d'infinité de solutions où chacune correspond à une valeur différente du paramètre d'une infinité de solutions liées à la présence d'une singularité (ou d'un système rectangulaire ou non zéro-dimensionel). Le terme de continuum est réservé au premier cas.

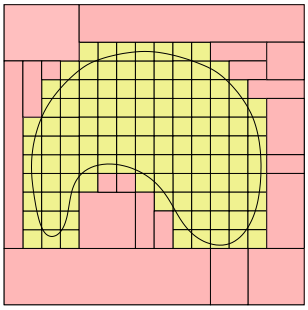


FIG. *2.7: Quadrillage d'un continuum.*

Ces continuums posent un problème de complexité : si l'on souhaite représenter par des boîtes un continuum de solutions, et que l'on exige, comme jusqu'à présent, une précision donnée ϵ sur ces boîtes (c'est à dire qu'une boîte n'ayant pu être déterminée comme infaisable doit avoir une taille en deçà de ϵ), alors tout algorithme aussi performant soit-il doit quadriller ce continuum par des boîtes de taille ϵ (voir figure 2.7). Le nombre de boîtes croît exponentiellement avec la dimension n et pour peu que le quadrillage soit fin par rapport au volume décrit, il explose très vite.

Une simplification conséquente de cette représentation consiste à regrouper les boîtes intérieures. On peut en effet avoir une description aussi fine qu'à la figure 2.7 du continuum avec le découpage suivant, moins coûteux :

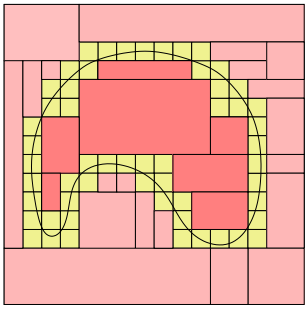


FIG. *2.8: Continuum & boîtes intérieures.*

Il apparaît donc qu'un enjeu tout aussi important que celui de la détection d'infaisabilité est la **détection de boîtes intérieures**. Tout comme pour l'infaisabilité, plus une boîte intérieure est détectée tôt, plus la combinatoire est réduite.

Nous allons maintenant décrire quelques façons d'appréhender les systèmes avec paramètres, en n'ayant recours qu'aux intervalles classiques.

Une première manière de traiter ces paramètres est de les considérer simplement comme des variables. Plusieurs inconvénients apparaissent immédiatement :
- Le système devient sous-contraint, ce qui rend impossible l'utilisation du filtrage et des tests d'existence/unicité de type Newton.
- Le fait de passer de n à $n+p$ variables alourdit considérablement les algorithmes de filtrage par propagation (voir chapitre 6) pour un gain quasi nul : intuitivement, ce sont les paramètres au même titre que les équations qui contraignent les variables. Tenter de filtrer ou bissecter les paramètres est contre-productif en général.
- Les réductions obtenues sur les domaines des paramètres sont souvent sans intérêt (obtenir un meilleur encadrement de la constante d'Euler, que l'on connaît déjà, n'est probablement pas un des objectifs !).
- D'après (2.16), réduire $\mathbf{p}$ ne peut que diminuer les chances de détecter que x est solution (nous y reviendrons au §3.4.6 p.70).
- Il n'est plus possible de détecter une boîte intérieure lorsque le quantificateur de p n'est plus $\exists$ mais $\forall$. Prenons par exemple $f(x,y,p) = \sqrt{x^2+y^2} = p$ avec $x,\ y\ \in [0,10]$ et $p \in [5,10]$. Si p est un paramètre, tout point (x,y) dans $[6,9]\times[0,1]$ est une solution puisque pour de tels couples on a $5 \leq x^2+y^2 \leq 10$. Donc $[6,9]\times[0,1]$ est intérieure. A l'inverse, si p est une variable, aucune boîte $\mathbf{x}\times\mathbf{y}\times\mathbf{p}$ de rayon non nul est intérieure (les points x partageant le même rayon p forment un cercle dans le plan).

Pour ces raisons, il est préférable de dissocier variables et paramètres, et d'avoir recours aux fonctions *épaisses*. L'idée derrière est très simple, elle consiste à résoudre le système de variable x comme si les paramètres étaient ponctuels, et d'introduire des intervalles (au lieu de réels) dans les calculs impliquant les paramètres. On a donc bien "épaissi" les fonctions.

2.9.2 Fonctions épaisses

Une fonction **épaisse** est une fonction de $\mathbb{IR}^n$ dans $\mathbb{IR}^m$ (de variables et à valeurs intervalles), définie par opposition à une fonction dite fine. Une fonction $\mathbf{f} : \mathbb{IR}^n \to \mathbb{IR}^m$ est **fine** si $\mathrm{rad}\,(\mathbf{x}) = 0 \Longrightarrow \mathrm{rad}\,(\mathbf{f}(\mathbf{x})) = 0$. Exemple avec $n = m = 1$: $f(\mathbf{x}) = \mathbf{x} + [0,1]$ est une fonction épaisse (on a $f(0) = [0,1]$).

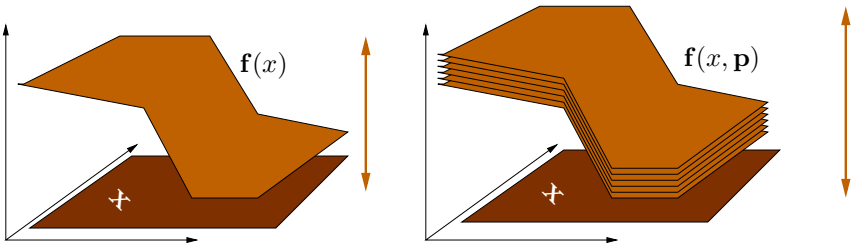


FIG. *2.9: Fonction fine et épaisse en p.*

Les fonctions épaisses interviennent naturellement dans notre système sous incertitude. Prenons une extension quelconque aux intervalles $\mathbf{f}$ de la fonction f de (2.15). La fonction $\mathbf{f^p} : \mathbf{x} \mapsto \mathbf{f}(\mathbf{x},\mathbf{p})$ est une fonction épaisse (dite fonction "épaisse en p"). Il est alors évident que pour toute boîte $\mathbf{x}$,

$$range(f, \mathbf{x}\times\mathbf{p}) \subseteq \mathbf{f^p}(\mathbf{x}).$$

De là,

$$0 \notin \mathbf{f^p}(\mathbf{x}) \Longrightarrow \big(\forall x \in D\big)\big(\forall p \in \mathbf{p}\big)\ f(x,p) \neq 0.$$

Donc $0 \notin \mathbf{f^p}(\mathbf{x})$ implique que $\mathbf{x}$ ne contient aucune solution, si bien que l'algorithme de bissection/évaluation est juste et peut être appliqué tel quel. Les boîtes $\mathbf{x}$ contenant une solution vérifient

$$\big(\exists x \in \mathbf{x}\big)\big(\exists p \in \mathbf{p}\big) \quad f(x,p) = 0. \tag{2.17}$$

Le théorème de Brouwer peut être utilisé avec une fonction épaisse pour détecter de telles boîtes. En effet, si $\mathbf{g^p}(\mathbf{x}) \subseteq \mathbf{x}$ alors

$$\big(\forall p \in \mathbf{p}\big)\ g(\mathbf{x}, p) \subseteq \mathbf{x}.$$

On en déduit que pour toute valeur de p, l'équation $g(x,p) = x$ admet un point fixe, c'est à dire,

$$\big(\forall p \in \mathbf{p}\big)\big(\exists x \in \mathbf{x}\big) \quad f(x,p) = 0, \tag{2.18}$$

ce qui implique bien (2.17). Il est important de voir que (2.18) est beaucoup plus « fort » que (2.17). Le théorème de Brouwer dans sa version épaisse apporte donc une information plus forte (l'existence d'un point fixe pour **toutes** les valeurs des paramètres), mais en contrepartie les conditions d'application sont plus difficiles à réaliser (la fonction étant épaisse, l'image $\mathbf{g^p}(\mathbf{x})$ a moins de chance d'être incluse dans $\mathbf{x}$).

On adapte de même le filtrage de Newton (proposition 2.12) et les tests d'existence & unicité (propositions 2.14 et 2.15) aux fonctions épaisses.

Proposition 2.18 (Filtrage & Tests de Newton, version épaisse)
Soit $f : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$ une fonction de variable $x \in \mathbb{R}^n$ et de paramètre $p \in \mathbf{p} \subseteq \mathbb{R}^k$. Soient $\mathbf{f}$ (resp. $\mathbf{J}$) une extension aux intervalles de f (resp. la jacobienne de f) et $\mathbf{J}_x$ la sous-matrice de $\mathbf{J}$ contenant les colonnes des dérivées partielles par rapport à x.
Pour tout $\mathbf{x} \in \mathbb{IR}^n$ et $\tilde{x} \in \mathbf{x}$,

$$\begin{array}{lllll} (i) & \big(\exists p \in \mathbf{p}\big)\ f(x,p) = 0 & \Longrightarrow & x \in \tilde{x} + \Sigma(\mathbf{J}_x(\mathbf{x},\mathbf{p}), -\mathbf{f}(\tilde{x},\mathbf{p})), \\ (ii) & \tilde{x} + \Sigma(\mathbf{J}_x(\mathbf{x},\mathbf{p}), -\mathbf{f}(\tilde{x},\mathbf{p})) \subseteq \mathbf{x} & \Longrightarrow & \big(\forall p \in \mathbf{p}\big)\big(\exists \text{ unique } \ x \in \mathbf{x}\big)\ f(x,p) = 0. \end{array}$$

La figure 2.10 reprend la figure 2.6 avec une fonction épaisse pour illustrer l'implication (i) de la proposition précédente.

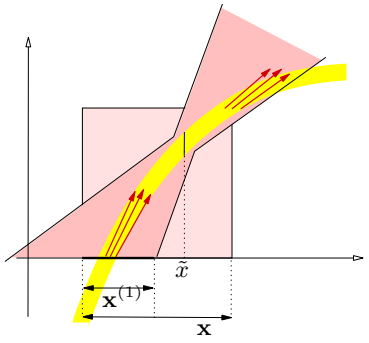


FIG. *2.10: Filtrage de Newton avec fonction épaisse. L'intervalle $\mathbf{x}^{(1)}$ est obtenu en calculant $\tilde{x} - f(\tilde{x},\mathbf{p})/f'(\mathbf{x},\mathbf{p})$.*

On étend également le filtrage de Newton-Hansen (proposition 2.16) aux fonctions épaisses en remplaçant dans la proposition précédente $J(\mathbf{x},\mathbf{p})$ par la matrice $\mathbf{J}(\mathbf{x},x,\mathbf{p})$, matrice « de Hansen » épaisse en p, dont la $i^{ème}$ ligne vaut

$$\mathbf{J}(\mathbf{x},x,\mathbf{p})_i := \Big(\frac{\partial f_i}{\partial x_1}(\mathbf{x}_1,x_2,\ldots,x_n,\mathbf{p})\ \frac{\partial f_i}{\partial x_2}(\mathbf{x}_1,\mathbf{x}_2,\ldots,x_n,\mathbf{p})\ \ldots\ \frac{\partial f_i}{\partial x_n}(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n,\mathbf{p})\Big)$$

La partie « unicité » restant à prouver. Nous montrons au paragraphe suivant qu'il est possible de mieux exploiter la méthode de Newton-Hansen que de l'appliquer directement dans sa version épaisse.

2.9.3 Méthode de Newton-Hansen paramétrique

Considérons la matrice de Hansen $\mathbf{J}((\mathbf{x},\mathbf{p}),(x,p))$ de f en un point $(x,p) \in \mathbf{x} \times \mathbf{p}$ (ce n'est plus une matrice $n \times n$ épaisse en p comme au paragraphe précédent, mais une matrice $n \times (n+k)$).

On a $\mathbf{J}((\mathbf{x},\mathbf{p}),(x,p)) := \begin{pmatrix} \mathbf{J}_x & \mathbf{J}_p \end{pmatrix}$ où $\mathbf{J}_x$ et $\mathbf{J}_p$ sont respectivement les matrices $n \times n$ et $n \times k$ suivantes :

$$\mathbf{J}_x := \begin{pmatrix} \dfrac{\partial f}{\partial x_1}(\mathbf{x}_1, x_2, \ldots, x_n, p_1, \ldots, p_k) & \ldots & \dfrac{\partial f}{\partial x_n}(\mathbf{x}_1, \ldots, \mathbf{x}_n, p_1, \ldots, p_k) \end{pmatrix},$$

$$\mathbf{J}_p := \begin{pmatrix} \dfrac{\partial f}{\partial p_1}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{p}_1, p_2, \ldots, p_k) & \ldots & \dfrac{\partial f}{\partial p_k}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{p}_1, \ldots, \mathbf{p}_k) \end{pmatrix}.$$

Reprenons rapidement les étapes menant à l'opérateur de Newton-Hansen. D'après la proposition 2.7, pour tout $x \in \mathbf{x}$ et tout $p \in \mathbf{p}$:

$$\big(\exists J \in \mathbf{J}((\mathbf{x},\mathbf{p}),(x,p))\big) \quad f(x,p) = f(\tilde{x},\tilde{p}) + J\Big((x,p) - (\tilde{x},\tilde{p})\Big).$$

Si $f(x,p) = 0$ alors

$$\begin{array}{lll} & \big(\exists J \in \mathbf{J}((\mathbf{x},\mathbf{p}),(x,p))\big) & f(\tilde{x},\tilde{p}) + J\Big((x,p) - (\tilde{x},\tilde{p})\Big) = 0 \\ \iff & \big(\exists J_x \in \mathbf{J}_x\big)\big(\exists J_p \in \mathbf{J}_p\big) & f(\tilde{x},\tilde{p}) + J_x(x-\tilde{x}) + J_p(p-\tilde{p}) = 0 \\ \iff & \big(\exists J_x \in \mathbf{J}_x\big)\big(\exists J_p \in \mathbf{J}_p\big) & x = \tilde{x} + J_x^{-1}\Big(-f(\tilde{x},\tilde{p}) - J_p(p-\tilde{p})\Big) \\ \iff & & x \in \tilde{x} + \Sigma\Big(\mathbf{J}_x, -f(\tilde{x},\tilde{p}) - \mathbf{J}_p(p-\tilde{p})\Big). \end{array}$$

Nous pouvons établir ainsi une version paramétrique de la proposition 2.16 p.36 (cf. [Gol05b] lemme V.2 p.191) :

Proposition 2.19 (Newton-Hansen, version paramétrique)
Avec les mêmes notations que précédemment, posons :

$$N(\mathbf{x}) := \tilde{x} + \Sigma(\mathbf{J}_x, -f(\tilde{x},\tilde{p}) - \mathbf{J}_p(\mathbf{p} - \tilde{p})).$$

(i) $\big(\exists p \in \mathbf{p}\big)\ f(x,p) = 0 \implies x \in N(\mathbf{x})$
(ii) $N(\mathbf{x}) \subseteq \mathbf{x} \implies \big(\forall p \in \mathbf{p}\big)\big(\exists x \in \mathbf{x}\big)\ f(x,p) = 0$

Nous venons de voir comment filtrer et prouver l'existence de solutions en présence de paramètres. Il reste à étudier les possibilités de détecter une boîte intérieure. Les deux prochaines sous-sections donnent quelques pistes.

2.9.4 Isolation de paramètres

Lorsque chaque paramètre n'apparaît que dans une seule équation, et qu'il peut être isolé (c'est à dire exprimé en fonction des variables uniquement), alors un test de boîte intérieur se ramène à une simple inclusion [GN06]. En effet, isoler chaque paramètre revient à supposer que $f(x,p) = 0 \iff g(x) = p$. D'où,

$$\big(\forall \mathbf{x} \in \mathbb{IR}^n\big) \quad \mathbf{g}(\mathbf{x}) \subseteq \mathbf{p} \implies \mathbf{x} \text{ est une boîte intérieure.}$$

Preuve.
$$\begin{array}{ll} \mathbf{g}(\mathbf{x}) \subseteq \mathbf{p} & \implies \big(\forall x \in \mathbf{x}\big)\ g(x) \subseteq \mathbf{p} \\ & \implies \big(\forall x \in \mathbf{x}\big)\big(\exists p \in \mathbf{p}\big)\ g(x) = p \\ & \implies \big(\forall x \in \mathbf{x}\big)\big(\exists p \in \mathbf{p}\big)\ f(x,p) = 0. \quad \square \end{array}$$

Ce genre de test peut s'appliquer avec notre exemple de robot (cf. chapitre 1) dans le cas où l'on considère une incertitude uniquement sur la longueur des jambes. La figure 2.11 représente les anneaux formés par les équations de distance lorsque les distances (les rayons) l_1 et l_2 varient resp. dans $\mathbf{l}_1$ et $\mathbf{l}_2$. Il apparaît alors graphiquement que la boîte $\mathbf{x}$ est intérieure puisque incluse dans l'intersection des deux anneaux (tout point de cette boîte est espacé du centre de chaque anneau d'une distance comprise dans $\mathbf{l}_1$ et $\mathbf{l}_2$).

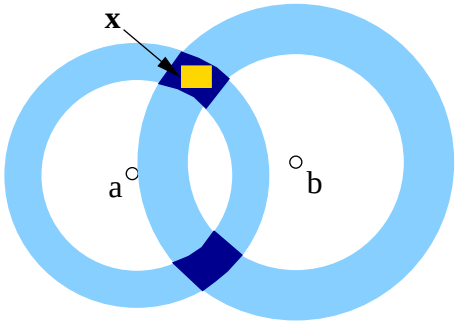


FIG. *2.11: Incertitude sur la longueur des jambes.*

Ce test a un intérêt, par exemple, en *estimation de paramètres* où p devient le vecteur des mesures [Jau02]. Il subsume également la détection de boîtes intérieures pour les systèmes d'inégalités. Dans le cas général, en revanche, les conditions ne sont pas rencontrées. Si dans notre exemple, l'incertitude est mise sur les points d'attache, par exemple les paramètres a_1 et b_1, il est déjà plus délicat d'isoler ces paramètres (il faut créer plusieurs cas en fonction du signe de $(x_1 - a_1)$ et $(x_1 - b_1)$ ou passer en coordonnées polaires). Si on cumule des incertitudes sur plus de deux paramètres présents dans une même équation, l'isolation de paramètre n'est plus applicable. Une méthode plus générale est décrite ci-dessous.

2.9.5 Permutation variable-paramètre

Lorsque le nombre de paramètres k coïncide avec le nombre d'équations n (et quelle que soit leur implication dans ces équations !) les propositions 2.18 et 2.19 donnent un moyen simple, efficace et élégant de détecter une boîte intérieure. La technique consiste à renverser le problème, c'est à dire à considérer les variables comme des paramètres et vice-versa. Cette technique est décrite dans [Gol06] et apparaît sous une autre forme dans [HM05]. Rappelons que, par définition, $\mathbf{x}$ est une boîte intérieure si

$$\big(\forall x \in \mathbf{x}\big)\big(\exists p \in \mathbf{p}\big)\ f(x,p) = 0.$$

Or cette dernière relation est précisément ce que donne le test d'existence paramétrique appliqué à la fonction f en prenant pour variable p et pour paramètre x. Dans la figure 2.12, l'incertitude est mise sur les abscisses a_1 et b_1 des points d'attache. Prenons un point x. Si on considère le problème *est-il possible de trouver un centre a et b en "glissant" le long des intervalles $\mathbf{a}_1$ et $\mathbf{b}_1$ de telle sorte que les cercles de centre a et b s'intersectent en x ?*, alors prouver qu'il existe une solution à ce problème revient bien à prouver que x est solution du problème original (2.16). Pour savoir si la boîte $\mathbf{x}$ est intérieure, il suffit donc d'appliquer un test d'existence au problème précédent avec a_1 et b_1 comme variables, et x comme paramètre variant dans $\mathbf{x}$.

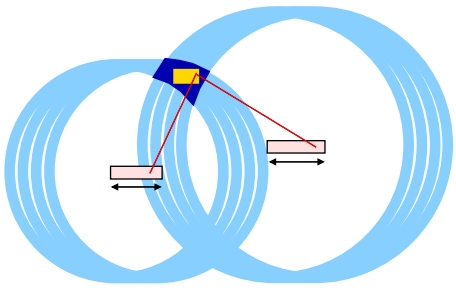


FIG. *2.12: Incertitude sur les points d'attache.*

2.9.6 Autres tests de boîtes intérieures

Nous énonçons brièvement d'autres méthodes permettant la détection de boîtes intérieures :

- Lorsque f est une fonction continue à valeurs réelles, le théorème des valeurs intermédiaires nous permet d'écrire :
$$\big(\exists p \in \mathbf{p}\big)\ f(x,p) = 0 \iff \begin{cases} \big(\exists p \in \mathbf{p}\big)\ f(x,p) \geq 0, \\ \big(\exists p \in \mathbf{p}\big)\ f(x,p) \leq 0. \end{cases}$$
Pour prouver qu'une boîte $\mathbf{x}$ est intérieure, il suffit dès lors d'isoler deux valeurs $\tilde{p}$ et $\underset{\sim}{p}$ dans $\mathbf{p}$ telles que $f(\mathbf{x}, \tilde{p}) \geq 0$ et $f(\mathbf{x}, \underset{\sim}{p}) \leq 0$.
- Lorsque f est un vecteur de polynômes, il peut être appliqué l'*élimination de quantificateurs* [Col75]. Cette technique ne s'avère en pratique applicable que pour des problèmes de taille très réduite.
- Lorsque chaque composante du vecteur de paramètres p ne possède qu'une seule occurrence dans f, nous verrons que la théorie des intervalles modaux apporte une réponse (cf §3.4.1 p.64).
- Dans le cas général, où il n'est rien supposé ni sur f ni sur p, des travaux récents remarquables ont permis de mettre au point un test de boîte intérieure basé sur la *détection de frontières* [GJ06]. Si $\mathbf{x}^{(0)}$ est le domaine initial des variables, la frontière de $\{x \in \mathbf{x}^{(0)} \mid \big(\exists p \in \mathbf{p}\big)\ f(x,p) = 0\}$ est obtenue en projetant certains points dits *singuliers* de l'ensemble $\{(x,p) \in \mathbf{x}^{(0)} \times \mathbf{p} \mid f(x,p) = 0\}$. La recherche de ces points singuliers nécessite d'appliquer un algorithme de *branch and bound* dans le produit cartésien $\mathbf{x}^{(0)} \times \mathbf{p}$, ce qui laisse un doute (pour le moment) sur un passage à l'échelle. Quoi qu'il en soit, ces travaux ont ouvert une nouvelle voie pour l'élaboration de tests de boîtes intérieures.

Enfin, remarquons que les tests de boîtes intérieures peuvent être étendus directement aux AE-systèmes. Il suffit de regrouper variables et paramètres universellement quantifiés. Avec les notations du §1.1.2, supposons données deux boîtes $\mathbf{u}$ et $\mathbf{v}$ représentant les domaines des paramètres, et une fonction

$$f : \begin{array}{ccc} (D \times \mathbf{u} \times \mathbf{v}) \subset \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q & \rightarrow & \mathbb{R}^n \\ (x,u,v) & \mapsto & f(x,u,v) \end{array} . \tag{2.19}$$

Les solutions du AE-système sont caractérisées par la formule suivante :

$$\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big) \quad f(x,u,v) = 0. \tag{2.20}$$

En assimilant u à des variables, n'importe quel test de boîte intérieur entraîne la propriété suivante :

$$\big(\forall x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big) \quad f(x,u,v) = 0,$$

qui signifie bien que $\mathbf{x}$ est une boîte intérieure.

2.10 Comparaison avec le modèle probabiliste

Le modèle probabiliste n'est pas *supplanté* par l'analyse par intervalles. Le modèle probabiliste apporte une information fine sur la distribution des sorties d'un système, ce que ne permet pas l'analyse par intervalles, où la réponse est binaire (oui : la sortie est dans un intervalle de tolérance, non : il est possible que la sortie soit en dehors). En revanche, il pose plusieurs problèmes :

- La validité du modèle probabiliste (par exemple, les erreurs sur les paramètres suivent-elles réellement la loi supposée ?)
- Le choix des paramètres de ce modèle (ce qui d'une certaine manière rajoute un nouveau type d'imprécision)
- La difficulté d'appliquer ce modèle lorsque les solutions ne s'écrivent pas de façon explicite.
- La difficulté de traiter le problème indirect : étant donnée une sortie fixée, quels sont les paramètres en entrée permettant de l'atteindre ?
- L'impossibilité de gérer le "cas le pire", comme nous l'avons déjà évoqué, c'est à dire l'impossibilité de garantir à 100 % que le cas le pire ne peut se produire (c'est à dire que l'erreur sur la sortie ne dépasse un certain seuil).

L'analyse par intervalles répond à ces questions dans la mesure où
- Aucune distribution sur les paramètres n'est supposée. Il est juste nécessaire de connaître une borne, ce qui est très souvent possible : il n'y a en effet aucun sens en réalité à considérer que l'erreur sur la longueur des jambes d'un robot de quelques mètres peut être, dans 0.0001% des cas, de 2 km !
- La distinction entre problème direct/indirect n'a plus vraiment lieu, les entrées et sorties étant traitées symétriquement. Il suffit grosso modo de choisir ce qu'on appelle *variables* et *paramètres*.
- Le cas le pire est géré. L'analyse par intervalles permet une analyse d'erreur bornée, et garantit la validité de la borne obtenue sur la sortie.

En contrepartie, de nombreux problèmes n'ont pas de solution si toutes les incertitudes sont prises en compte. Déterminer quelle est la probabilité que le "cas le pire" se produise est une information que l'analyse par intervalles ne peut pas fournir. Pour résumer, si les intervalles peuvent garantir la robustesse d'un système, les probabilités peuvent mesurer les risques. Ces deux approches sont complémentaires, et ne s'adressent pas aux mêmes types de problèmes.

Chapitre 3

Intervalles modaux

Sommaire

Les intervalles, tels qu'ils ont été définis dans le chapitre 2, présentent deux inconvénients majeurs. Le premier est purement algébrique : les intervalles ne forment ni une structure de groupe (pour l'addition ou la multiplication) ni un treillis (pour l'intersection et la réunion). Comme nous l'avons déjà indiqué, $-[0,1]$ n'est pas l'opposé de $[0,1]$, et il n'existe en fait aucun intervalle $\mathbf{a}$ tel que $\mathbf{a} + [0,1] = 0$. De même, $[0,1] \cap [2,3]$ est l'ensemble vide. Or cet ensemble n'est pas un élément de $\mathbb{IR}$.

La structure des intervalles a été étendue par Ortolf [Ort69] et Kaucher [Kau80] dans les années 70 pour donner lieu à un ensemble qui offre de meilleures propriétés algébriques, tout en maintenant la validité des lois usuelles de l'arithmétique classique des intervalles. Cet ensemble, appelé *intervalles généralisés* (dont on a coutume d'attribuer la paternité à Kaucher), étend les intervalles classiques de la même manière que les nombres relatifs étendent les nombres naturels, le corps des complexes celui des réels, etc. Un intervalle généralisé peut avoir des bornes inversées, comme par exemple, $[1,-1]$. Ces considérations algébriques n'ont, a priori, pas de rapport avec l'usage que l'on fait des intervalles, et peuvent être vus comme une pure abstraction.

Le second inconvénient est lié à l'interprétation, trop limitée, que l'on peut faire des calculs par intervalles. Un intervalle $\mathbf{x}$ représente jusqu'ici un ensemble de valeurs possibles quantifié *existentiellement*, dans le sens suivant : si $\mathbf{z}$ est le résultat de l'évaluation par f de $\mathbf{x}$, alors $\mathbf{z} \supseteq \{z \mid \left(\exists x \in \mathbf{x}\right)\ z = f(x)\}$. Si on considère une propriété sur les réels telle que $f(x) = 0$, l'intervalle $\mathbf{x}$ sera considéré comme "satisfiable" dès lors qu'il existe une valeur dans $\mathbf{x}$ qui satisfait cette propriété. Or, dans de nombreuses situations, on souhaite que **toutes** les valeurs d'un intervalle satisfassent une propriété donnée. Ces différentes façons de quantifier un intervalle ont été formalisées au chapitre 1 au travers de la notion d'*AE-systèmes*. Nous avons vu que l'analyse par intervalles classiques n'apportait a priori[1] une réponse que dans des cas de figure bien précis (détection de boîtes intérieures pour des systèmes avec autant de paramètres que d'équations...). Une équipe de chercheurs espagnols,

[1] Un exemple de problème qu'il nous semble difficile de reformuler avec des intervalles classiques serait de décrire l'ensemble des $x \in [-1000, 1000]^3$ tels que pour tout $u \in [-1,1]^3$ il existe $v \in [-1,1]^3$ satisfaisant le système suivant :

$$cos(x_1v_1u_1 + x_2v_2u_2 + x_3v_3u_3) - sin(x_1x_2v_1 + v_1v_2u_3 + u_1u_2x_3) = 0,$$
$$(x_1v_2u_3)^3 + 2exp(x_2v_3^2u_1) - (x_3v_1u_2) = 0.$$

menée par Gardeñes et Sainz, a mis en place au cours des années 80 la théorie dite des *intervalles modaux* [GnMA85, GnSJ$^+$01], qui prend pour point de départ l'association d'un quantificateur à un intervalle. Ainsi, on ne manipule plus un intervalle $\mathbf{x}$, mais l'un des deux couples suivants :

$$(\mathbf{x}, \exists) \quad \text{ou} \quad (\mathbf{x}, \forall).$$

Cette théorie a donné naissance à quelques théorèmes puissants. Elle a surtout permis d'ouvrir de nouvelles portes vers la résolution des AE-systèmes, en donnant un cadre théorique permettant d'attaquer cette classe de problèmes. Elle est en contrepartie tristement célèbre pour être hermétique.

Si les intervalles généralisés et les intervalles modaux sont regroupés dans un seul chapitre, c'est qu'ils se sont avérés être étroitement liés. Ce lien - entre inversion de borne et inversion de quantificateur - aurait été décelé déjà à l'origine par Ortolf. Les chercheurs espagnols incorporent également l'arithmétique de Kaucher dans leur théorie. Mais les intervalles généralisés ne deviennent qu'une représentation accessoire des intervalles modaux, et à ce titre, n'en simplifie guère la formalisation.

Récemment, Goldsztejn [Gol05b] a proposé une nouvelle construction de la théorie des intervalles modaux, où cette fois, les intervalles généralisés apparaissent au coeur même de la formulation. Celle-ci devient alors nettement plus simple à manipuler, donc plus rigoureuse, si bien qu'elle a déjà permis de nouveaux développements. Cette théorie est aujourd'hui approuvée par une grande partie de la communauté travaillant sur les intervalles modaux.

Nous présentons les intervalles modaux dans ce chapitre avec une démarche particulière, motivée par le constat suivant : le cadre théorique des intervalles modaux, bien que très élégant et très prometteur, paraît pour le moment disproportionné pour ce qu'il apporte en pratique. Le *filtrage de Newton généralisé* est par contre intéressant pour les AE-systèmes, et notre but est de redémontrer ce résultat en se basant sur des outils plus rudimentaires puis, dans un second temps seulement, de les replonger dans la théorie de Goldsztejn. Cette présentation des intervalles modaux propose donc des preuves originales qui n'ont recours qu'à des notions de "min-max". Selon nous, elles facilitent une compréhension rapide des théorèmes. En contrepartie, elles offrent une vision assez limitée des choses. Cette construction, à vocation pédagogique, ne prétend donc nullement se substituer aux différentes théories des intervalles modaux.

Le chapitre est organisé de la façon suivante. Dans la section 3.1, nous présentons la notion d'*image quantifiée*, clé de voûte de l'approche modale que nous traduisons ensuite par des formules min-max. Les intervalles généralisés et l'arithmétique de Kaucher seront ensuite introduits comme la structure permettant de calculer ces min-max. Nous en décrivons le minimum nécessaire au 3.2 pour pouvoir, dans la section 3.4, démontrer les résultats qui nous intéressent quant à la résolution d'AE-systèmes. Le chapitre se termine par une présentation du formalisme proposé par Goldsztejn, et enfin par un rapide tour d'horizon des propriétés algébriques des intervalles généralisés dont nous nous servirons par la suite.

3.1 Images quantifiées

Dans ce §3.1, nous considérons une fonction $f : \begin{array}{ll} \mathbb{R}^p \times \mathbb{R}^q & \to \mathbb{R} \\ (x, y) & \mapsto f(x, y) \end{array}$ continue.

Fixons $\mathbf{x} \in \mathbb{IR}^p$ et $\mathbf{y} \in \mathbb{IR}^q$ quelconques. Rappelons que pour que ϕ soit une extension aux intervalles de f, il faut et il suffit que

$$\big(\forall x \in \mathbf{x}\big)\big(\forall y \in \mathbf{y}\big)\big(\exists z \in \phi(\mathbf{x}, \mathbf{y})\big) \mid z = f(x, y).$$

Nous avons introduit au chapitre 1 la fonction $\mathbf{f}^{\square}$ comme étant l'extension aux intervalles *optimale* de f. Si f est continue, l'intervalle $\mathbf{f}^{\square}(\mathbf{x}, \mathbf{y})$ est égal à $range(f, \mathbf{x} \times \mathbf{y})$, il est donc à la fois :
– le plus petit ensemble $\mathbf{z}$ vérifiant $\big(\forall x \in \mathbf{x}\big)\big(\forall y \in \mathbf{y}\big)\big(\exists z \in \mathbf{z}\big) \mid z = f(x, y)$,
– le plus grand ensemble $\mathbf{z}$ vérifiant $\big(\forall z \in \mathbf{z}\big)\big(\exists x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big) \mid z = f(x, y)$.

Notre but est de calculer d'autres types d'images de f que celle qui englobe toutes les combinaisons possibles obtenues pour $x \in \mathbf{x}$ et $y \in \mathbf{y}$, notamment dans le cas où l'une des variables (x ou y) représente un paramètre. Deux types d'image nous intéressent en particulier ; les intervalles $\mathbf{z}$ vérifiant

$$\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\big(\exists z \in \mathbf{z}\big)\ z = f(x,y), \tag{3.1}$$

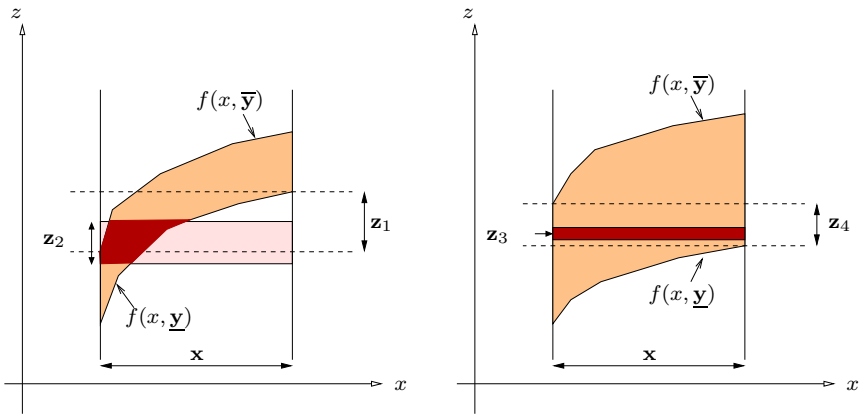
et les intervalles $\mathbf{z}$ vérifiant

$$\big(\forall z \in \mathbf{z}\big)\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x,y). \tag{3.2}$$

Le calcul exact de ces images est un problème difficile pour lequel, à notre connaissance, il n'existe pas de méthode. La première moitié de ce chapitre est entièrement consacrée à donner un moyen de calculer une approximation de ces ensembles.

Sur les deux figures ci-dessous, f est représentée comme une fonction de variable x épaisse en y (y est donc le paramètre). Les quatre intervalles exhibés sont :
- $\mathbf{z}_1$: le plus petit intervalle $\mathbf{z}$ vérifiant (3.1)
- $\mathbf{z}_2$: un intervalle $\mathbf{z}$ n'incluant pas $\mathbf{z}_1$, et par conséquent ne vérifiant pas (3.1)
- $\mathbf{z}_3$: un intervalle $\mathbf{z}$ vérifiant (3.2)
- $\mathbf{z}_4$: le plus grand intervalle $\mathbf{z}$ vérifiant (3.2).

Nous appellerons les intervalles tels que $\mathbf{z}_1$ ou $\mathbf{z}_4$ des *images quantifiées* (cf. définition 3.1 plus loin).



(a) (b)

FIG. *3.1: Images quantifiées*

Remarquons que l'intervalle $\mathbf{z}_4$ possède une caractérisation directe sous forme d'ensemble :

$$\mathbf{z}_4 = \{z \in \mathbb{R} \mid \big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x,y)\}. \tag{3.3}$$

Ce n'est pas le cas en revanche de l'intervalle $\mathbf{z}_1$.

Nous allons dès maintenant montrer qu'identifier des intervalles inclus dans (ou contenant) un intervalle tel que $\mathbf{z}_4$ permet de déterminer si une boîte est intérieure (ou extérieure) pour un AE-système. En généralisant la définition (3.3), ce résultat s'étend même directement aux fonctions quelconques à valeurs vectorielles. Considérons donc momentanément une fonction f à valeurs dans $\mathbb{R}^m$ et le AE-système $\Sigma(f, \mathbf{u}, \mathbf{v})$ suivant :

$$x \in \Sigma(f, \mathbf{u}, \mathbf{v}) \iff \big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\ f(x,u,v) = 0.$$

Alors deux lemmes s'appliquent. Ils font intervenir des ensembles de même nature que l'intervalle $\mathbf{z}_4$ de notre exemple : il peuvent en effet être caractérisés par (3.3), en réorganisant les variables (regroupement de u et x par exemple dans le lemme 3.1, changement d'ordre et regroupement de u et v dans le lemme 3.2).

Lemme 3.1

$$Si\ \mathbf{z} \subseteq \{z \in \mathbb{R}^m \mid \big(\forall x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\ z = f(x,u,v)\}\ alors$$

$$0 \in \mathbf{z} \Longrightarrow \mathbf{x}\ intérieure.$$

Preuve. $0 \in \mathbf{z} \Longrightarrow 0 \in \{z \in \mathbb{R}^m \mid \big(\forall x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\ z = f(x,u,v)\}$
$\Longrightarrow \big(\forall x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\ f(x,u,v) = 0$
$\Longrightarrow \mathbf{x}$ intérieure $\square$

Avec les intervalles classiques, il ne paraît pas commode de fournir un test de boîte intérieure générique (i.e., qui puisse s'appliquer pour n'importe quel système d'équations), hormis la permutation variable-paramètre (cf. §2.9.5 p.44). Les méthodes existantes d'extension de domaines [CDR99] ou d'élimination de quantificateurs [DH88] sont restreintes à des systèmes bien particuliers. Calculer un intervalle vérifiant (3.3) donnerait donc un test intérieur générique, mais également un test d'infaisabilité [HSVJ05] :

Lemme 3.2

$$Si\ \mathbf{z} \supseteq \{z \in \mathbb{R}^m \mid \big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\big(\exists x \in \mathbf{x}\big)\ z = f(x,u,v)\}\ alors$$

$$0 \notin \mathbf{z} \Longrightarrow \mathbf{x}\ infaisable.$$

Preuve. $0 \notin \mathbf{z} \Longrightarrow 0 \notin \{z \in \mathbb{R}^m \mid \big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\big(\exists x \in \mathbf{x}\big)\ z = f(x,u,v)\}$
$\Longrightarrow \neg\ [\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\big(\exists x \in \mathbf{x}\big)\ f(x,u,v) = 0]$
$\Longrightarrow \big(\exists u \in \mathbf{u}\big)\big(\forall v \in \mathbf{v}\big)\big(\forall x \in \mathbf{x}\big)\ f(x,u,v) \neq 0$
$\Longrightarrow \big(\forall x \in \mathbf{x}\big)\big(\exists u \in \mathbf{u}\big)\big(\forall v \in \mathbf{v}\big)\ f(x,u,v) \neq 0$
$\Longrightarrow \big(\forall x \in \mathbf{x}\big)\neg\ \big(\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\ f(x,u,v) = 0\big)$
$\Longrightarrow \big(\forall x \in \mathbf{x}\big)\ x \notin \Sigma(f,\mathbf{u},\mathbf{v})$
$\Longrightarrow \mathbf{x}$ infaisable $\square$

Clairement, le test précédent est nettement plus efficace que le test d'infaisabilité classique qui se limiterait à vérifier l'appartenance de 0 à $\mathbf{z}$ tel que

$$\mathbf{z} \supseteq \{z \in \mathbb{R}^m \mid \big(\exists u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\big(\exists x \in \mathbf{x}\big)\ z = f(x,u,v)\}.$$

Le test "classique" calculant des images beaucoup plus larges, la détection de boîtes infaisables devient plus rare.

Ainsi, pour approximer (intérieurement ou extérieurement) les ensemble solution d'un AE-système, la théorie des intervalles modaux consiste à calculer de tels intervalles, caractérisés par de nouvelles combinaisons de quantificateurs. Ces intervalles servent ensuite de base à divers tests (boîtes intérieures, infaisables) mais, comme nous le verrons ultérieurement, également à des techniques de filtrage. Revenons au cas d'une fonction f à valeurs réelles : nous généralisons la notion d'*image* à celle d'*image quantifiée*. Les notations utilisées dans cette définition ne sont pas standards.

Définition 3.1 (Image quantifiée) *Soient* $\mathbf{x} \in \mathbb{IR}^p$ *et* $\mathbf{y} \in \mathbb{IR}^q$. *On appelle* **image quantifiée** *de* f *en* $\mathbf{x} \times \mathbf{y}$ *l'un des deux intervalles suivants :*

- $f^{\exists}(\forall \mathbf{x}\ \exists \mathbf{y})$ *le plus petit intervalle* $\mathbf{z}$ *vérifiant* (3.1)*, s'il existe, et l'ensemble vide sinon. Il s'agit donc de* $\mathbf{z}_1$ *sur la figure 3.1.(a). Notons qu'il existe au moins un intervalle* $\mathbf{z}$ *vérifiant* (3.1) *:* $range(f, x \in \mathbf{x}, y \in \mathbf{y})$*. Il peut y avoir par contre plusieurs intervalles minimaux vérifiant* (3.1)*, auquel cas il n'existe pas de plus petit élément (cf. figure 3.1.(b) où tout intervalle dégénéré inclus dans* $\mathbf{z}_4$ *vérifie* (3.1)*). Dans ce cas, par convention, on pose* $f^{\exists}(\forall \mathbf{x}\ \exists \mathbf{y}) := \emptyset$.
- $f^{\forall}(\forall \mathbf{x}\ \exists \mathbf{y})$ *le plus grand ensemble* $\mathbf{z}$ *vérifiant* (3.2)*. Il s'agit donc de* $\mathbf{z}_4$ *sur la figure 3.1.(b). Notons que cet ensemble peut être vide si aucun* $\mathbf{z}$ *ne vérifie* (3.2) *(cf. figure 3.1.(a)).*

Dans la même logique, nous pouvons écrire si f est continue :

$$\mathbf{f}^{\square}(\mathbf{x},\mathbf{y}) = f^{\exists}(\forall\mathbf{x}\ \forall\mathbf{y}) = f^{\forall}(\exists\mathbf{x}\ \exists\mathbf{y}),$$

et cet intervalle existe toujours.

Les figures 3.1.(a) et 3.1.(b) suggèrent l'idée que si $f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y})$ ($\mathbf{z}_1$ sur la figure) n'est pas réduit à un point alors $f^{\forall}(\forall\mathbf{x}\ \exists\mathbf{y})$ ($\mathbf{z}_4$ sur la figure) est vide, et inversement (cf. lemme 3.3 p.55). Contrairement à l'intervalle $\mathbf{z}_4$, nous n'avons pas trouvé d'interprétation intéressante de l'intervalle $\mathbf{z}_1$, mais nous l'introduisons parce qu'il a un rôle tout à fait symétrique à celui de $\mathbf{z}_4$. On peut déjà en conclure que si la théorie permet de calculer des images quantifiées alors, en pratique, soit on est dans la situation favorable où $\mathbf{z}_4$ n'est pas vide, et les intervalles calculés ont une utilité, soit on est dans la situation où z_1 n'est pas vide, et il n'y a plus d'interprétation intéressante.

Nous allons montrer entre autres dans la section suivante que si $f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y})$ n'est pas vide, alors $f^{\forall}(\forall\mathbf{y}\ \exists\mathbf{x})$ n'est pas vide (bien noter que les quantificateurs pour x et y ont été inversés) et plus fort encore :

$$f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y}) \subseteq f^{\forall}(\forall\mathbf{y}\ \exists\mathbf{x}).$$

D'autre part, $f^{\exists}(\forall\mathbf{y}\ \exists\mathbf{x})$ et $f^{\forall}(\forall\mathbf{x}\ \exists\mathbf{y})$ sont alors vides.

3.1.1 Images quantifiées et Min-Max

Les deux propositions de cette sous-section donnent une caractérisation directe des images quantifiées d'une fonction à valeurs réelles en terme de *min-max*.

Commençons par un rappel élémentaire sur les ensembles définis par des relations :

Définition 3.2 *Soit E un ensemble, P une relation sur E, et $\mathbf{s} \subseteq E$.*

$$\begin{array}{lll} \mathbf{s} \subseteq \{s \in E \mid P(s)\} \iff \big(\forall s \in \mathbf{s}\big) & P(s) & \textit{(inclusion)} \\ \mathbf{s} \supseteq \{s \in E \mid P(s)\} \iff \big(\forall s \notin \mathbf{s}\big) & \neg P(s) & \textit{(exclusion)} \end{array}$$

Dans la proposition suivante, nous construisons un intervalle $\mathbf{z}$ "artificiellement" en calculant deux valeurs $\underline{\mathbf{z}}$ et $\overline{\mathbf{z}}$. Or il se peut que $\overline{\mathbf{z}} < \underline{\mathbf{z}}$. L'intervalle $\mathbf{z}$ peut donc avoir des bornes inversées (ex : $[1,-1]$).

Proposition 3.1 *Soient $\mathbf{x} \in \mathbb{IR}^p$, $\mathbf{y} \in \mathbb{IR}^q$, f une fonction continue de $\mathbf{x} \times \mathbf{y}$ dans $\mathbb{R}$. Posons*

$$\mathbf{z} := [\max_{x\in\mathbf{x}} \min_{y\in\mathbf{y}} f(x,y), \min_{x\in\mathbf{x}} \max_{y\in\mathbf{y}} f(x,y)]$$

$$\mathbf{z}' := f^{\forall}(\forall\mathbf{x}\ \exists\mathbf{y}) = \{z \mid \big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x,y)\} \quad (\in \mathcal{P}(\mathbb{R}))$$

Si l'une des deux conditions suivantes :

$$\begin{array}{ll} \textit{(i)} & \overline{\mathbf{z}} \geq \underline{\mathbf{z}} \quad \textit{ou} \\ \textit{(ii)} & \mathbf{z}' \neq \emptyset \end{array}$$

est vérifiée, alors l'autre condition est également vérifiée et

$$\textit{(iii)} \quad \mathbf{z} = \mathbf{z}'$$

Preuve.
Montrons $(i) \Longrightarrow (iii)$. Supposons (i) et montrons d'abord $\mathbf{z} \subseteq \mathbf{z}'$. Soient $z \in \mathbf{z}$, $\tilde{x} \in \mathbf{x}$.

$$\begin{aligned} z \in \mathbf{z} &\Longrightarrow \max_{x\in\mathbf{x}} \min_{y\in\mathbf{y}} f(x,y) \leq z \leq \min_{x\in\mathbf{x}} \max_{y\in\mathbf{y}} f(x,y) \text{ car } \overline{\mathbf{z}} \geq \underline{\mathbf{z}} \\ &\Longrightarrow \min_{y\in\mathbf{y}} f(\tilde{x},y) \leq z \leq \max_{y\in\mathbf{y}} f(\tilde{x},y). \end{aligned}$$

En utilisant le théorème des valeurs intermédiaires, puisque f est continue cela implique donc

$$\big(\exists y \in \mathbf{y}\big)\ z = f(\tilde{x}, y).$$

Les valeurs z et $\tilde{x}$ étant quelconques, on a bien démontré :

$$\big(\forall z \in \mathbf{z}\big)\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x, y),$$

ce qui implique $\mathbf{z} \subseteq \mathbf{z}'$ d'après la définition 3.2 (inclusion).

Montrons ensuite $\mathbf{z} \supseteq \mathbf{z}'$. Prenons $z \notin \mathbf{z}$.
Comme $\overline{\mathbf{z}} \geq \underline{\mathbf{z}}$, alors soit $z < \max\limits_{x\in\mathbf{x}} \min\limits_{y\in\mathbf{y}} f(x, y)$, soit $z > \min\limits_{x\in\mathbf{x}} \max\limits_{y\in\mathbf{y}} f(x, y)$. Dans le premier cas, cela implique :

$$\big(\exists \tilde{x} \in \mathbf{x}\big)\ z < \min_{y\in\mathbf{y}} f(\tilde{x}, y),$$

i.e.,

$$\big(\exists \tilde{x} \in \mathbf{x}\big)\big(\forall y \in \mathbf{y}\big)\ z \neq f(\tilde{x}, y).$$

On obtient la même chose dans le second cas. Finalement, comme z est choisi quelconque on a bien démontré

$$\big(\forall z \notin \mathbf{z}\big)\big(\exists x \in \mathbf{x}\big)\big(\forall y \in \mathbf{y}\big)\ z \neq f(x, y),$$

ce qui implique $\mathbf{z} \supseteq \mathbf{z}'$ d'après la définition 3.2 (exclusion). Ainsi, $\mathbf{z} = \mathbf{z}'$ et (iii) est vraie.

Notons que $\big(\overline{\mathbf{z}} \geq \underline{\mathbf{z}}\big) \wedge \big(\mathbf{z} = \mathbf{z}'\big) \Longrightarrow \mathbf{z}' \neq \emptyset$, donc on a également $(i) \Longrightarrow (ii)$.

Il reste à montrer $(ii) \Longrightarrow (i)$. Supposons $\mathbf{z}' \neq \emptyset$ et prenons z quelconque dans $\mathbf{z}'$. Grâce à la définition 3.2 (inclusion) on a

$$\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x, y). \tag{3.4}$$

Notons alors $\underset{\sim}{x}$ le point de $\mathbf{x}$ où la fonction $x \to \min_{y\in\mathbf{y}} f(x, y)$ (qui est bornée) atteint son maximum (ce maximum est atteint car $\mathbf{x}$ est compact), et $\tilde{x}$, le point de $\mathbf{x}$ où la fonction $x \to \max_{y\in\mathbf{y}} f(x, y)$ (également bornée) atteint son minimum. Grâce à (3.4)

$$\big(\exists \underset{\sim}{y} \in \mathbf{y}\big)\ z = f(\underset{\sim}{x}, \underset{\sim}{y}) \quad \text{et} \quad \big(\exists \tilde{y} \in \mathbf{y}\big)\ z = f(\tilde{x}, \tilde{y}),$$

donc

$$\min_{y\in\mathbf{y}} f(\underset{\sim}{x}, y) \leq z \leq \max_{y\in\mathbf{y}} f(\tilde{x}, y).$$

Ceci implique, par définition de $\underset{\sim}{x}$ et $\tilde{x}$,

$$\max_{x\in\mathbf{x}} \min_{y\in\mathbf{y}} f(x, y) \leq z \leq \min_{x\in\mathbf{x}} \max_{y\in\mathbf{y}} f(x, y),$$

C'est à dire (i). □

Proposition 3.2 *Soient $\mathbf{x} \in \mathbb{IR}^p$, $\mathbf{y} \in \mathbb{IR}^q$, f une fonction continue de $\mathbf{x} \times \mathbf{y}$ dans $\mathbb{R}$. Posons*

$$\mathbf{z} := [\min_{x\in\mathbf{x}} \max_{y\in\mathbf{y}} f(x, y), \max_{x\in\mathbf{x}} \min_{y\in\mathbf{y}} f(x, y)]$$

$$\mathbf{z}' := f^{\exists}(\forall \mathbf{x}\ \exists \mathbf{y})$$

Si l'une des deux conditions suivantes :

(i) $\overline{\mathbf{z}} \geq \underline{\mathbf{z}}$ *ou*
(ii) $\mathbf{z}' \neq \emptyset$

est vérifiée, alors l'autre condition est également vérifiée et

(iii) $\mathbf{z} = \mathbf{z}'$.

Preuve.
Montrons $(i) \Longrightarrow (iii)$. Supposons (i) et montrons d'abord $\mathbf{z}' \subseteq \mathbf{z}$. Fixons $\tilde{x} \in \mathbf{x}$, et posons $g_{\tilde{x}} : y \mapsto f(\tilde{x}, y)$. On a

$$\min_{x\in\mathbf{x}} \max_{y\in\mathbf{y}} f(x,y) \leq \max_{y\in\mathbf{y}} f(\tilde{x},y) \quad \text{et} \quad \min_{y\in\mathbf{y}} f(\tilde{x},y) \leq \max_{x\in\mathbf{x}} \min_{y\in\mathbf{y}} f(x,y),$$

c'est à dire

$$\underline{\mathbf{z}} \leq \max_{y\in\mathbf{y}} g_{\tilde{x}}(y) \quad \text{et} \quad \min_{y\in\mathbf{y}} g_{\tilde{x}}(y) \leq \overline{\mathbf{z}}. \tag{3.5}$$

D'une part, comme $g_{\tilde{x}}$ est continue sur $\mathbf{y}$, alors $range(g_{\tilde{x}}, \mathbf{y})$ est l'intervalle $[\min_{y\in\mathbf{y}} g_{\tilde{x}}(y), \max_{y\in\mathbf{y}} g_{\tilde{x}}(y)]$, d'autre part, $\mathbf{z}$ a ses bornes dans le bon sens ($\overline{\mathbf{z}} \geq \underline{\mathbf{z}}$) par hypothèse, donc (3.5) se réécrit

$$\mathbf{z} \cap range(g_{\tilde{x}}, \mathbf{y}) \neq \emptyset$$

ce qui équivaut à $\big(\exists y \in \mathbf{y}\big)\big(\exists z \in \mathbf{z}\big) z = g_{\tilde{x}}(y)$. Ainsi, $\tilde{x}$ ayant été choisi quelconque, on a bien montré que $\mathbf{z}$ vérifiait (3.1) p.49, ce qui implique $\mathbf{z}' \subseteq \mathbf{z}$. Pour montrer que $\mathbf{z}' \supseteq \mathbf{z}$, il suffit de montrer que les intervalles $]-\infty, \overline{\mathbf{z}}[$, et $]\underline{\mathbf{z}}, +\infty[$ (qui incluent $\mathbf{z}$ à l'exception d'une borne) ne vérifient pas (3.1). Notons comme à la preuve précédente $\underset{\sim}{x}$ le point de $\mathbf{x}$ où la fonction $x \to \min_{y\in\mathbf{y}} f(x,y)$ atteint son maximum, et $\tilde{x}$, le point de $\mathbf{x}$ où la fonction $x \to \max_{y\in\mathbf{y}} f(x,y)$ atteint son minimum. On a $\big(\forall y \in \mathbf{y}\big)\ f(\underset{\sim}{x}, y) \geq \min_{y\in\mathbf{y}} f(\underset{\sim}{x}, y)$, donc $\big(\forall y \in \mathbf{y}\big)\ f(\underset{\sim}{x}, y) \geq \overline{\mathbf{z}}$, donc

$$\big(\forall y \in \mathbf{y}\big)\big(\forall z \in]-\infty, \overline{\mathbf{z}}[\big)\ f(\underset{\sim}{x}, y) \neq z.$$

On démontre de même

$$\big(\forall y \in \mathbf{y}\big)\big(\forall z \in]\underline{\mathbf{z}}, +\infty[\big)\ f(\tilde{x}, y) \neq z.$$

On a bien $(i) \Longrightarrow (iii)$. Comme $\big(\overline{\mathbf{z}} \geq \underline{\mathbf{z}}\big) \wedge \big(\mathbf{z} = \mathbf{z}'\big) \Longrightarrow \mathbf{z}' \neq \emptyset$, on a également $(i) \Longrightarrow (ii)$.

Montrons $(ii) \Longrightarrow (i)$. Supposons $\mathbf{z}' \neq \emptyset$. Par l'absurde, si $\underline{\mathbf{z}} > \overline{\mathbf{z}}$ alors d'après la proposition 3.1, l'ensemble $f^{\forall}(\forall \mathbf{x}\ \exists \mathbf{y})$ n'est ni vide ni réduit à un point et tout intervalle $[z, z]$ inclus dans cet ensemble vérifie $\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\big(\exists z \in \mathbf{z}\big)\ z = f(x, y)$. Donc il n'existe pas un unique plus petit intervalle vérifiant cette propriété, ce qui par définition implique $f^{\exists}(\forall \mathbf{x}\ \exists \mathbf{y}) = \emptyset$, i.e., $\mathbf{z}' = \emptyset$. □

Nous manipulerons donc désormais ces intervalles min-max, beaucoup plus simples que des images quantifiées qui sont des ensembles, de surcroît éventuellement vides. Nous avons vu que ces intervalles définis à partir de min-max peuvent avoir leurs bornes "dans le mauvais ordre". Ces intervalles sont dits *généralisés*. Il convient à ce stade de définir cette notion formellement.

3.1.2 Intervalles généralisés

On appelle **intervalle généralisé** [Kau80, Sha02, Gol05b], et on note $[u, v]$, tout couple de réels (u, v).

On n'impose pas cette fois $u \leq v$. Voici quelques exemples d'intervalles généralisés :

$$[0,1],\ [1,0],\ [-1,1],\ [1,-1],\ [0,0].$$

Si $\mathbf{x}$ est le couple $[u, v]$, on notera $\underline{\mathbf{x}}$ ou (inf $\mathbf{x}$) le réel u, $\overline{\mathbf{x}}$ ou (sup $\mathbf{x}$) le réel v.

L'ensemble des intervalles généralisés est noté $\mathbb{KR}$. Il est partitionné en deux sous-ensembles :
- Les intervalles **propres** : ceux dont les bornes vont dans l'ordre croissant. Ainsi $\mathbf{x}$ est propre si $\underline{\mathbf{x}} \leq \overline{\mathbf{x}}$. Cet ensemble coïncide avec celui des intervalles classiques, et continuera d'être noté $\mathbb{IR}$.
- Les intervalles **impropres** : ceux dont les bornes vont dans l'ordre décroissant. Ainsi $\mathbf{x}$ est impropre si $\underline{\mathbf{x}} > \overline{\mathbf{x}}$. Remarquons qu'on utilise une inégalité stricte pour éviter d'avoir des intervalles à la fois propres et impropres. Les intervalles dégénérés (comme $[2, 2]$) sont donc considérés comme propres. Cette convention n'a aucune incidence sur les résultats, et évite les confusions. L'ensemble des intervalles impropres est noté $\overline{\mathbb{IR}}$.

On a donc $\mathbb{KR} = \mathbb{IR} \cup \overline{\mathbb{IR}}$.

Nous introduisons quelques notations pour permettre l'interversion des bornes dans des calculs, i.e., le passage d'un intervalle propre à son homologue impropre. On appelle :

– **dual** de $\mathbf{x}$, et on note (dual $\mathbf{x}$) ou dual $(\mathbf{x})$, l'intervalle généralisé $[\overline{\mathbf{x}}, \underline{\mathbf{x}}]$.

– **partie propre** de $\mathbf{x}$, et on note (pro $\mathbf{x}$) ou pro $(\mathbf{x})$, l'intervalle propre $\mathbf{x}$ si $\mathbf{x} \in \mathbb{IR}$, ou (dual $\mathbf{x}$) si $\mathbf{x} \in \overline{\mathbb{IR}}$.

– **partie impropre** de $\mathbf{x}$, et on note (imp $\mathbf{x}$) ou imp $(\mathbf{x})$, l'intervalle impropre (dual (pro $\mathbf{x}$)).

Voici quelques exemples :

$$\text{dual}\,([0,1]) = [1,0] \quad \text{dual}\,([1,0]) = [0,1] \quad \text{pro}\,([0,1]) = [0,1] \quad \text{pro}\,([1,0]) = [0,1]$$

Les implications suivantes sont triviales :

$$\begin{array}{llcll} \mathbf{x} & \text{propre} & \Longleftrightarrow & \text{dual } \mathbf{x} & \text{impropre} \\ \mathbf{x} & \text{propre} & \Longrightarrow & \text{imp } \mathbf{x} & \text{impropre} \\ \mathbf{x} & \text{impropre} & \Longrightarrow & \text{pro } \mathbf{x} & \text{propre} \\ \ldots & & & & \end{array}$$

Enfin, nous utiliserons la relation d'inclusion suivante :

Définition 3.3 (Inclusion dans $\mathbb{KR}$)

$$\mathbf{x} \subseteq \mathbf{y} \iff \underline{\mathbf{x}} \geq \underline{\mathbf{y}} \text{ et } \overline{\mathbf{x}} \leq \overline{\mathbf{y}}$$

Ainsi, $[2,-4] \subseteq [1,-3] \subseteq [-1,-1] \subseteq [-3,1] \subseteq [-4,2]$.
Il est inutile d'interpréter pour le moment cette inclusion. Elle nous servira uniquement comme un moyen commode de traiter trois cas de figure différents simultanément. En effet, considérons $\mathbf{x} \in \mathbb{KR}$ et $\mathbf{y} \in \mathbb{KR}$. Prouver l'inclusion suivante :

$$\mathbf{x} \subseteq \mathbf{y}$$

équivaut, par définition, à prouver les trois implications suivantes :

$$\mathbf{x} \in \mathbb{IR} \Longrightarrow (\mathbf{y} \in \mathbb{IR} \text{ et } \mathbf{x} \subseteq \mathbf{y}) \tag{3.6}$$

$$\mathbf{y} \in \overline{\mathbb{IR}} \Longrightarrow (\mathbf{x} \in \overline{\mathbb{IR}} \text{ et } (\text{pro } \mathbf{y}) \subseteq (\text{pro } \mathbf{x})) \tag{3.7}$$

$$\mathbf{x} \in \overline{\mathbb{IR}} \text{ et } \mathbf{y} \in \mathbb{IR} \Longrightarrow \mathbf{y} \cap (\text{pro } \mathbf{x}) \neq \emptyset. \tag{3.8}$$

3.1.3 Retour aux images quantifiées

Comme nous l'avons dit plus haut, notre but est de manipuler des intervalles min-max. Dans un souci de lisibilité, nous introduisons les notations suivantes (standards cette fois[2]) pour une fonction f à valeurs réelles :

$$\bigwedge_{x\in\mathbf{x}} \bigvee_{y\in\mathbf{y}} f(x,y) = [\max_{x\in\mathbf{x}} \min_{y\in\mathbf{y}} f(x,y), \min_{x\in\mathbf{x}} \max_{y\in\mathbf{y}} f(x,y)]$$

$$\bigvee_{x\in\mathbf{x}} \bigwedge_{y\in\mathbf{y}} f(x,y) = [\min_{x\in\mathbf{x}} \max_{y\in\mathbf{y}} f(x,y), \max_{x\in\mathbf{x}} \min_{y\in\mathbf{y}} f(x,y)]$$

[2]Nous retrouverons ces notations dans la section 3.5, où elles auront une signification plus riche.

Dans cette notation, x et y sont des vecteurs de variables. On s'autorisera à séparer un vecteur de variables x en deux sous vecteurs x_1, x_2 en répétant le symbole associé à x (exemple : $\bigwedge_{x\in\mathbf{x}} f(x) = \bigwedge_{x_1\in\mathbf{x}}\bigwedge_{x_2\in\mathbf{x}} f(x_1,x_2)$). Voir en particulier l'énoncé de la proposition 3.4 p.59. Les deux formules ci-dessous sont donc des cas particuliers :

$$\bigvee_{x\in\mathbf{x},y\in\mathbf{y}} f(x,y) = \bigvee_{x\in\mathbf{x}}\bigvee_{y\in\mathbf{y}} f(x,y) = [\min_{x\in\mathbf{x}}\min_{y\in\mathbf{y}} f(x,y), \max_{x\in\mathbf{x}}\max_{y\in\mathbf{y}} f(x,y)]$$

$$\bigwedge_{x\in\mathbf{x},y\in\mathbf{y}} f(x,y) = \bigwedge_{x\in\mathbf{x}}\bigwedge_{y\in\mathbf{y}} f(x,y) = [\max_{x\in\mathbf{x}}\max_{y\in\mathbf{y}} f(x,y), \min_{x\in\mathbf{x}}\min_{y\in\mathbf{y}} f(x,y)].$$

Ces quatre intervalles étant généralisés, il est important d'insister sur le fait qu'ils existent toujours !

Nous observons alors les deux points suivants :

$$\bigwedge_{x\in\mathbf{x}}\bigvee_{y\in\mathbf{y}} f(x,y) = \text{dual}\Big(\bigvee_{x\in\mathbf{x}}\bigwedge_{y\in\mathbf{y}} f(x,y)\Big) \tag{3.9}$$

$$\bigvee_{x\in\mathbf{x}}\bigwedge_{y\in\mathbf{y}} f(x,y) \subseteq \bigwedge_{y\in\mathbf{y}}\bigvee_{x\in\mathbf{x}} f(x,y) \tag{3.10}$$

La première relation est immédiate. La deuxième découle du fait que pour toute fonction f continue, $\max_{y\in\mathbf{y}}\min_{x\in\mathbf{x}} f(x,y) \leq \min_{x\in\mathbf{x}}\max_{y\in\mathbf{y}} f(x,y)$ et $\max_{x\in\mathbf{x}}\min_{y\in\mathbf{y}} f(x,y) \leq \min_{y\in\mathbf{y}}\max_{x\in\mathbf{x}} f(x,y)$.

Il est possible dès lors de prouver le résultat annoncé à la fin de la section 3.1 :

Lemme 3.3 *Soit $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ continue, $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{y} \in \mathbb{R}^q$.*

Si $f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y}) \neq \emptyset$	*alors*	$f^{\forall}(\forall\mathbf{y}\ \exists\mathbf{x}) \neq \emptyset$ *et* $f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y}) \subseteq f^{\forall}(\forall\mathbf{y}\ \exists\mathbf{x})$.
Si de plus, $f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y})$ *n'est pas réduit à un point*	*alors*	$f^{\exists}(\forall\mathbf{y}\ \exists\mathbf{x}) = \emptyset$ *et* $f^{\forall}(\forall\mathbf{x}\ \exists\mathbf{y}) = \emptyset$.

Preuve. De par la proposition 3.2, si $f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y}) \neq \emptyset$ n'est pas vide $f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y}) = \bigvee_{x\in\mathbf{x}}\bigwedge_{y\in\mathbf{y}} f(x,y)$, et ce dernier intervalle est propre. En appliquant (3.10) et par définition de l'inclusion dans $\mathbb{KR}$, $\bigwedge_{y\in\mathbf{y}}\bigvee_{x\in\mathbf{x}} f(x,y)$ est donc, lui aussi, propre. La proposition 3.1 s'applique alors et ce dernier intervalle coïncide avec l'ensemble $f^{\forall}(\forall\mathbf{y}\ \exists\mathbf{x})$, qui n'est pas vide. L'inclusion des ensembles se déduit immédiatement de l'inclusion des intervalles. De plus, si $\bigvee_{x\in\mathbf{x}}\bigwedge_{y\in\mathbf{y}} f(x,y)$ est propre et non réduit à un point, en appliquant (3.9), $\bigwedge_{x\in\mathbf{x}}\bigvee_{y\in\mathbf{y}} f(x,y)$ est impropre. D'après la prop. 3.1, $f^{\forall}(\forall\mathbf{x}\ \exists\mathbf{y})$ est donc vide. Finalement, par application de (3.10), par définition de l'inclusion dans $\mathbb{KR}$, et d'après la prop. 3.2, $f^{\exists}(\forall\mathbf{y}\ \exists\mathbf{x})$ est aussi vide. □

Par contraposition, on déduit également du lemme précédent que si $f^{\forall}(\forall\mathbf{x}\ \exists\mathbf{y}) \neq \emptyset$ alors $f^{\exists}(\forall\mathbf{x}\ \exists\mathbf{y}) = \emptyset$. Il est possible d'énoncer dans ce cas de figure une propriété supplémentaire :

Lemme 3.4 *Soit $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ continue, $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{y} \in \mathbb{R}^q$.*
Soient $\mathbf{z}$ et $\mathbf{z}'$ deux intervalles vérifiant respectivement :

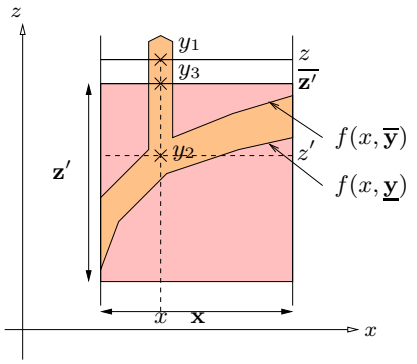
$$\mathbf{z} := f^{\forall}(\forall\mathbf{x}\ \exists\mathbf{y}) \quad et \quad \big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\big(\exists z' \in \mathbf{z}'\big)\ z' = f(x,y).$$

Si $\mathbf{z} \neq \emptyset$ alors $\mathbf{z} \cap \mathbf{z}' \neq \emptyset$.

Preuve.
Supposons que $\mathbf{z} \cap \mathbf{z}' = \emptyset$. Comme $\mathbf{z} \neq \emptyset$ alors $\exists z \notin \mathbf{z}'$ tel que $z \in \mathbf{z}$. Pour un tel z, soit $z > \overline{\mathbf{z}'}$, soit $z < \underline{\mathbf{z}'}$. Supposons $z > \overline{\mathbf{z}'}$. On a

$$\begin{array}{ll} \big(\forall x \in \mathbf{x}\big)\big(\exists y_1 \in \mathbf{y}\big)\ z = f(x, y_1) & \text{(par définition de } f^{\forall}(\forall \mathbf{x}\ \exists \mathbf{y})), \\ \big(\forall x \in \mathbf{x}\big)\big(\exists y_2 \in \mathbf{y}\big)\big(\exists z' \in \mathbf{z}'\big)\ z' = f(x, y_2). & \end{array}$$

Fixons x et posons $g_x : y \mapsto f(x, y)$. Cette fonction est continue, donc d'après ce qui précède, $[z', z] \subseteq \mathit{range}(g_x, \mathbf{y})$. Comme $z' \leq \overline{\mathbf{z}'} < z$, alors $\overline{\mathbf{z}'} \in \mathit{range}(g_x, \mathbf{y})$ (voir figure ci-dessous). Autrement dit, $\big(\exists y_3 \in \mathbf{y}\big)\ \overline{\mathbf{z}'} = f(x, y_3)$. On a bien montré que $\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ \overline{\mathbf{z}'} = f(x, y)$, c.a.d. $\overline{\mathbf{z}'} \in \mathbf{z}$, contradiction. La preuve est similaire pour $z < \underline{\mathbf{z}'}$. □



Remarque 3.1 *Comme déjà mentionné dans le §3.1, on peut donner une caractérisation directe de* $\bigwedge_{x \in \mathbf{x}} \bigvee_{y \in \mathbf{y}} f(x, y)$ *comme simple ensemble, dès lors que cet intervalle est propre :*

$$\bigwedge_{x \in \mathbf{x}} \bigvee_{y \in \mathbf{y}} f(x, y) = \{z \mid \big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x, y)\}.$$

Il n'existe pas de telle caractérisation pour $\bigvee_{x \in \mathbf{x}} \bigwedge_{y \in \mathbf{y}} f(x, y)$*, que cet intervalle soit propre ou non. Néanmoins, s'il est propre, il est quand même possible d'établir deux inclusions. Tout d'abord, d'après le lemme 3.3, on a*

$$\bigvee_{x \in \mathbf{x}} \bigwedge_{y \in \mathbf{y}} f(x, y) \subseteq \{z \mid \big(\forall y \in \mathbf{y}\big)\big(\exists x \in \mathbf{x}\big)\ z = f(x, y)\}.$$

De plus, en notant $\mathbf{z} := \bigvee_{x \in \mathbf{x}} \bigwedge_{y \in \mathbf{y}} f(x, y)$*, la proposition 3.2 permet également d'écrire* $\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\big(\exists z \in \mathbf{z}\big)\ z = f(x, y)$*, ce qui implique* $\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ f(x, y) \in \mathbf{z}$*, ce qui implique de nouveau* $\big(\forall z \notin \mathbf{z}\big)\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z \neq f(x, y)$*, c'est à dire, d'après la définition 3.2 (exclusion),*

$$\{z \mid \big(\exists x \in \mathbf{x}\big)\big(\forall y \in \mathbf{y}\big)\ z = f(x, y)\} \subseteq \bigvee_{x \in \mathbf{x}} \bigwedge_{y \in \mathbf{y}} f(x, y).$$

Finalement, on a prouvé la chaîne d'inclusions suivante :

$$\{z \mid \big(\exists x \in \mathbf{x}\big)\big(\forall y \in \mathbf{y}\big)\ z = f(x, y)\} \subseteq \bigvee_{x \in \mathbf{x}} \bigwedge_{y \in \mathbf{y}} f(x, y) \subseteq \{z \mid \big(\forall y \in \mathbf{y}\big)\big(\exists x \in \mathbf{x}\big)\ z = f(x, y)\}.$$

Toutefois, ces inclusions sont trop faibles, et ne seront pas exploitées (l'ensemble à gauche de la chaîne d'inclusions est en général vide ou réduit à un nombre fini de points).

Dans la section suivante, notre but est de calculer explicitement les images quantifiées pour les opérateurs arithmétiques et les fonctions élémentaires. Nous en déduirons une approximation des images quantifiées dans le cas général (fonctions arithmétiques).

3.2 Images quantifiées et arithmétique de Kaucher

De la même façon que l'arithmétique classique permet de calculer (approximativement) des images de fonctions, nous allons construire une arithmétique permettant de calculer des images quantifiées. Les opérations définies ci-dessous forment l'arithmétique dite de Kaucher [Kau80]. Il faut noter qu'à l'origine cette arithmétique n'avait aucun rapport avec les images quantifiées ! Une présentation plus conforme à l'esprit des travaux de Kaucher est faite au §3.6.

3.2.1 Addition

Dans une image quantifiée telle que $\bigwedge_{x\in\mathbf{x}} \bigvee_{y\in\mathbf{y}} f(x,y)$, les rôles joués par $\mathbf{x}$ et $\mathbf{y}$ ne sont pas les mêmes (par exemple, si on se réfère à l'interprétation que fournit la proposition 3.1, $\mathbf{x}$ est quantifié universellement et $\mathbf{y}$ existentiellement). Puisque l'on désire obtenir des images quantifiées par des calculs arithmétiques, il convient de distinguer d'une manière ou d'une autre les intervalles ayant le rôle de $\mathbf{x}$ de ceux ayant le rôle de $\mathbf{y}$. Une solution possible aurait été d'associer à $\mathbf{x}$ le symbole $\bigwedge$ et à $\mathbf{y}$ le symbole $\bigvee$...
Le choix qui s'est avéré le plus judicieux consiste à représenter un intervalle tel que $\mathbf{x}$ par un intervalle impropre.

Prenons deux intervalles $\mathbf{a} \in \overline{\mathbb{IR}}$ et $\mathbf{b} \in \mathbb{IR}$, et considérons la fonction $(x,y) \mapsto x+y$.

Grâce à la monotonie des applications partielles $(x+.) : y \to x+y$ et $(.+y) : x \to x+y$, on a :

$$\max_{a\in(\text{pro }\mathbf{a})} \min_{b\in\mathbf{b}} (a+b) = \min_{b\in\mathbf{b}} \max_{a\in(\text{pro }\mathbf{a})} (a+b) = \overline{(\text{pro }\mathbf{a})} + \underline{\mathbf{b}} = \underline{\mathbf{a}} + \underline{\mathbf{b}},$$

$$\min_{a\in(\text{pro }\mathbf{a})} \max_{b\in\mathbf{b}} (a+b) = \max_{b\in\mathbf{b}} \min_{a\in(\text{pro }\mathbf{a})} (a+b) = \underline{(\text{pro }\mathbf{a})} + \overline{\mathbf{b}} = \overline{\mathbf{a}} + \overline{\mathbf{b}}.$$

Nous pouvons donc calculer directement $\bigwedge_{a\in(\text{pro }\mathbf{a})} \bigvee_{b\in\mathbf{b}} a+b = \bigvee_{b\in\mathbf{b}} \bigwedge_{a\in(\text{pro }\mathbf{a})} a+b = [\underline{\mathbf{a}}+\underline{\mathbf{b}}, \overline{\mathbf{a}}+\overline{\mathbf{b}}]$.

On retrouve ainsi formellement (pour notre cas particulier) la définition classique de l'addition de deux intervalles : $\mathbf{a}+\mathbf{b} := [\underline{\mathbf{a}}+\underline{\mathbf{b}}, \overline{\mathbf{a}}+\overline{\mathbf{b}}]$. Considérons par exemple $[4,3]+[0,2]$. Le résultat de ce calcul est $[4,5]$. On a donc $[4,5] = \bigwedge_{a\in[3,4]} \bigvee_{b\in[0,2]} a+b$, et comme cet intervalle est propre, on en déduit alors avec la proposition 3.1 que $\big(\forall c \in [4,5]\big)\big(\forall a \in [3,4]\big)\big(\exists b \in [0,2]\big)\ c = a+b$.

On définit finalement l'addition $\oplus$ dans $\mathbb{KR}$ entre deux intervalles $\mathbf{a} \in \mathbb{KR}$ et $\mathbf{b} \in \mathbb{KR}$, comme étant l'opération suivante :

$$\mathbf{a} \oplus \mathbf{b} = \bigbigwedge_{a\in\text{pro }(\mathbf{a})} \bigbigwedge_{b\in\text{pro }(\mathbf{b})} a+b, \qquad \text{où } \bigbigwedge_{x\in\text{pro }(\mathbf{x})} = \begin{cases} \bigvee & \text{si } \mathbf{x} \in \mathbb{IR} \\ \bigwedge & \text{si } \mathbf{x} \in \overline{\mathbb{IR}} \end{cases}.$$

de telle sorte que cette opération coïncide avec l'addition classique lorsque $\mathbf{a}$ et $\mathbf{b}$ sont tous deux dans $\mathbb{IR}$ (ce qui justifie a posteriori le choix d'associer les intervalles impropres au symbole $\bigwedge$ et pas au symbole $\bigvee$).

Le calcul pour les autres combinaisons propre/impropre pour $\mathbf{a}$ et $\mathbf{b}$ découlent de la même manière que pour le calcul ci-dessus effectué avec $\mathbf{a} \in \overline{\mathbb{IR}}$ et $\mathbf{b} \in \mathbb{IR}$. On s'aperçoit alors facilement que l'addition cherchée dans $\mathbb{KR}$ s'écrit dans tous les cas de la façon suivante :

$$\mathbf{a} \oplus \mathbf{b} := [\underline{\mathbf{a}}+\underline{\mathbf{b}}, \overline{\mathbf{a}}+\overline{\mathbf{b}}].$$

3.2.2 Multiplication

La multiplication sur $\mathbb{KR}$ peut être construite exactement de la même manière.

$$\mathbf{a} \otimes \mathbf{b} = \bigwedge\!\!\bigvee_{a \in \mathrm{pro}\,(\mathbf{a})} \bigwedge\!\!\bigvee_{b \in \mathrm{pro}\,(\mathbf{b})} a \times b, \qquad \text{où } \bigwedge\!\!\bigvee_{x \in \mathrm{pro}\,(\mathbf{x})} = \begin{cases} \bigvee & \text{si } \mathbf{x} \in \mathbb{IR} \\ \bigwedge & \text{si } \mathbf{x} \in \overline{\mathbb{IR}} \end{cases}.$$

Tout d'abord, puisque cette multiplication doit aussi étendre celle de $\mathbb{IR}$, on n'échappe pas à la distinction de cas liée à la présence ou non du zéro dans les opérandes, puisque cette distinction est déjà présente dans l'arithmétique classique. On note par commodité (cf. [Sha02]) :

$\mathcal{P}$	:=	$\{\mathbf{x} \in \mathbb{KR} \mid \underline{\mathbf{x}} \geq 0 \;\&\; \overline{\mathbf{x}} \geq 0\}$	(intervalles positifs)
$\mathcal{Z}$	:=	$\{\mathbf{x} \in \mathbb{KR} \mid \underline{\mathbf{x}} \leq 0 \leq \overline{\mathbf{x}}\}$	(intervalles (propres) contenant 0)
$-\mathcal{P}$	:=	$\{\mathbf{x} \in \mathbb{KR} \mid \underline{\mathbf{x}} \leq 0 \;\&\; \overline{\mathbf{x}} \leq 0\}$	(intervalles négatifs)
dual $\mathcal{Z}$	:=	$\{\mathbf{x} \in \mathbb{KR} \mid (\text{dual } \mathbf{x}) \in \mathcal{Z}\}$	(intervalles (impropres) contenus dans 0)

Prenons par exemple $\mathbf{a} \in (\mathbb{IR} \cap \mathcal{P})$ et $\mathbf{b} \in \text{dual } \mathcal{Z}$. On obtient immédiatement

$$\mathbf{a} \otimes \mathbf{b} = [\min_{a \in \mathbf{a}} \max_{b \in \mathrm{pro}\,(\mathbf{b})} a \times b, \max_{a \in \mathbf{a}} \min_{b \in \mathrm{pro}\,(\mathbf{b})} a \times b] = [\underline{\mathbf{ab}}, \underline{\mathbf{a}}\overline{\mathbf{b}}].$$

On peut retrouver ainsi de la même façon les 10 cas (16 si on ne tient pas compte de la commutativité) résumés dans la table 3.1.

$\mathbf{a} \times \mathbf{b}$	$\mathbf{b} \in \mathcal{P}$	$\mathbf{b} \in \mathcal{Z}$	$\mathbf{b} \in -\mathcal{P}$	$\mathbf{b} \in \text{dual } \mathcal{Z}$
$\mathbf{a} \in \mathcal{P}$	$[\underline{\mathbf{a}}\,\underline{\mathbf{b}}, \overline{\mathbf{a}}\,\overline{\mathbf{b}}]$	$[\overline{\mathbf{a}}\,\underline{\mathbf{b}}, \overline{\mathbf{a}}\,\overline{\mathbf{b}}]$	$[\overline{\mathbf{a}}\,\underline{\mathbf{b}}, \underline{\mathbf{a}}\,\overline{\mathbf{b}}]$	$[\underline{\mathbf{a}}\,\underline{\mathbf{b}}, \underline{\mathbf{a}}\,\overline{\mathbf{b}}]$
$\mathbf{a} \in \mathcal{Z}$	$[\underline{\mathbf{a}}\,\overline{\mathbf{b}}, \overline{\mathbf{a}}\,\overline{\mathbf{b}}]$	$[\min\{\underline{\mathbf{a}}\,\overline{\mathbf{b}}, \overline{\mathbf{a}}\,\underline{\mathbf{b}}\}, \max\{\underline{\mathbf{a}}\,\underline{\mathbf{b}}, \overline{\mathbf{a}}\,\overline{\mathbf{b}}\}]$	$[\overline{\mathbf{a}}\,\underline{\mathbf{b}}, \underline{\mathbf{a}}\,\underline{\mathbf{b}}]$	0
$\mathbf{a} \in -\mathcal{P}$	$[\underline{\mathbf{a}}\,\overline{\mathbf{b}}, \overline{\mathbf{a}}\,\underline{\mathbf{b}}]$	$[\underline{\mathbf{a}}\,\overline{\mathbf{b}}, \underline{\mathbf{a}}\,\underline{\mathbf{b}}]$	$[\overline{\mathbf{a}}\,\overline{\mathbf{b}}, \underline{\mathbf{a}}\,\underline{\mathbf{b}}]$	$[\overline{\mathbf{a}}\,\overline{\mathbf{b}}, \overline{\mathbf{a}}\,\underline{\mathbf{b}}]$
$\mathbf{a} \in \text{dual } \mathcal{Z}$	$[\underline{\mathbf{a}}\,\underline{\mathbf{b}}, \overline{\mathbf{a}}\,\underline{\mathbf{b}}]$	0	$[\overline{\mathbf{a}}\,\overline{\mathbf{b}}, \underline{\mathbf{a}}\,\overline{\mathbf{b}}]$	$[\max\{\underline{\mathbf{a}}\,\underline{\mathbf{b}}, \overline{\mathbf{a}}\,\overline{\mathbf{b}}, \min\{\underline{\mathbf{a}}\,\overline{\mathbf{b}}, \overline{\mathbf{a}}\,\underline{\mathbf{b}}\}\}]$

TAB. *3.1: Multiplication de Kaucher*

3.2.3 Soustraction et Division

La soustraction et la division suivent de nouveau le même schéma de définition. On prouve alors aisément que leurs définitions peuvent être ramenées à celle de l'addition et de la multiplication, tout comme leur pendant dans $\mathbb{IR}$:

$$\begin{array}{rcll} \mathbf{a} \ominus \mathbf{b} & := & \mathbf{a} \oplus [-\overline{\mathbf{b}}, -\underline{\mathbf{b}}], & \\ \mathbf{a} \oslash \mathbf{b} & := & \mathbf{a} \otimes [1/\overline{\mathbf{b}}, 1/\underline{\mathbf{b}}] & \text{si } 0 \not\subseteq \mathbf{b} \text{ et } \mathbf{b} \not\subseteq 0. \end{array}$$

3.2.4 Fonctions élémentaires

Les fonctions élémentaires ont été introduites au §2.3 puis étendues aux intervalles au §2.3.2. Pour que l'extension aux intervalles généralisés respecte notre interprétation en termes d'images quantifiées, on doit avoir (cf. §3.1.3) :

$$\begin{array}{rcl} \mathbf{a} \in \mathbb{IR} & \Longrightarrow & f(\mathbf{a}) := [\min_{a \in \mathbf{a}} f(a), \max_{a \in \mathbf{a}} f(a)] \\ \mathbf{a} \in \overline{\mathbb{IR}} & \Longrightarrow & f(\mathbf{a}) := [\max_{a \in \mathrm{pro}\,(\mathbf{a})} f(a), \min_{a \in \mathrm{pro}\,(\mathbf{a})} f(a)] \end{array}$$

En conséquence de quoi, une fonction élémentaire appliquée à un intervalle propre retourne le résultat de l'évaluation classique (telle que définie au 2.3.2), et appliquée à un intervalle impropre retourne ce même résultat impropre (c.a.d., avec les bornes inversées). Par exemple, $sqr([2,1]) = [4,1]$.

3.2.5 Propriétés des calculs dans $\mathbb{KR}$

La proposition suivante se démontre par une induction immédiate :

Proposition 3.3 *Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction arithmétique et $\mathbf{x} \in \mathbb{KR}^n$.*

$$f(\text{dual}\,(\mathbf{x})) = \text{dual}\,(f(\mathbf{x}))$$

3.3 Approximation d'images quantifiées

Nous allons maintenant utiliser l'arithmétique de Kaucher pour approximer les images quantifiées de fonctions arithmétiques quelconques (ou presque).

Les deux propositions suivantes montrent qu'en enchaînant plusieurs opérations entre intervalles généralisés, les résultats des calculs continuent d'être interprétables par des min-max, donc des images quantifiées. Néanmoins, la relation devient plus faible que pour les opérateurs de base. Une image quantifiée ne peut plus être obtenue exactement via l'arithmétique de Kaucher. Seule une inclusion est possible.

Les inclusions dans la proposition suivante sont toutes dans $\mathbb{KR}$, c'est donc volontairement qu'il n'est rien supposé sur la nature des intervalles mis en jeu (propres ou impropres). Rappelons, à ce stade, que l'inclusion entre intervalles généralisés nous permet simplement de faire d'une pierre "trois" coups (voir §3.1.2). Pour alléger le texte, nous supposerons également que les fonctions sont définies sur $\mathbb{R}^n$, et non plus sur un sous-ensemble $D \subseteq \mathbb{R}^n$. Les preuves s'adaptent facilement au cas d'un domaine de définition restreint (en adjoignant des conditions supplémentaires d'inclusion dans ce domaine).

Proposition 3.4 (Opérateur binaire, cas général)
Soient $\mathbf{a} \in \mathbb{IR}$, $\mathbf{b} \in \mathbb{IR}$, $\mathbf{x} \in \mathbb{IR}^p$, $\mathbf{y} \in \mathbb{IR}^q$, et $f : \mathbb{R}^{p+q+1} \to \mathbb{R}$ continue. Soit $\star$ un opérateur parmi $\{+, -, \times, /\}$ (on interdit d'avoir $0 \in \mathbf{b}$ si $\star = /$).

$$\bigwedge_{x\in\mathbf{x}} \bigvee_{a\in\mathbf{a}} \bigvee_{b\in\mathbf{b}} \bigvee_{y\in\mathbf{y}} f(a \star b, x, y) \supseteq \bigwedge_{x\in\mathbf{x}} \bigvee_{c\in\mathbf{c}} \bigvee_{y\in\mathbf{y}} f(c, x, y) \text{ avec } \mathbf{c} = \mathbf{a} \star \mathbf{b}, \tag{3.11}$$

$$\bigwedge_{x\in\mathbf{x}} \bigwedge_{a\in\mathbf{a}} \bigwedge_{b\in\mathbf{b}} \bigvee_{y\in\mathbf{y}} f(a \star b, x, y) \supseteq \bigwedge_{x\in\mathbf{x}} \bigwedge_{c\in\mathbf{c}} \bigvee_{y\in\mathbf{y}} f(c, x, y) \text{ avec } \mathbf{c} = \mathbf{a} \star \mathbf{b}, \tag{3.12}$$

Si $\text{dual}\,(\mathbf{a}) \star \mathbf{b} \in \mathbb{IR}$ *alors*

$$\bigwedge_{x\in\mathbf{x}} \bigwedge_{a\in\mathbf{a}} \bigvee_{b\in\mathbf{b}} \bigvee_{y\in\mathbf{y}} f(a \star b, x, y) \supseteq \bigwedge_{x\in\mathbf{x}} \bigvee_{c\in\mathbf{c}} \bigvee_{y\in\mathbf{y}} f(c, x, y) \text{ avec } \mathbf{c} = \text{dual}\,(\mathbf{a}) \star \mathbf{b}. \tag{3.13}$$

Sinon,

$$\bigwedge_{x\in\mathbf{x}} \bigwedge_{a\in\mathbf{a}} \bigvee_{b\in\mathbf{b}} \bigvee_{y\in\mathbf{y}} f(a \star b, x, y) \supseteq \bigwedge_{x\in\mathbf{x}} \bigwedge_{c\in\text{dual}\,(\mathbf{c})} \bigvee_{y\in\mathbf{y}} f(c, x, y) \text{ avec } \mathbf{c} = \text{dual}\,(\mathbf{a}) \star \mathbf{b}. \tag{3.14}$$

Preuve. Nous montrons pour chaque inclusion que la borne inférieure du membre de gauche est plus petite que celle du membre de droite ; l'inégalité sur les bornes supérieures s'obtenant de façon totalement symétrique.

De plus, chacune de ces inégalités est de la forme $\max_{x\in\mathbf{x}} \phi(x,\mathbf{a},\mathbf{b},\mathbf{y}) \leq \max_{x\in\mathbf{x}} \psi(x,\mathbf{c},\mathbf{y})$. On fixe donc $x \in \mathbf{x}$, et on se contente de montrer que $\phi(x,\mathbf{a},\mathbf{b},\mathbf{y}) \leq \psi(x,\mathbf{c},\mathbf{y})$. Ayant choisi x quelconque, cela implique effectivement l'inégalité souhaitée.

Montrons (3.11).

Soient $\tilde{c}$ et $\tilde{y}$ tels que $\min_{c\in\mathbf{c}} \min_{y\in\mathbf{y}} f(x,c,y) = f(x,\tilde{c},\tilde{y})$. On a $\big(\forall c \in \mathbf{c}\big)\big(\exists a \in \mathbf{a}\big)\big(\exists b \in \mathbf{b}\big)$ tels que $c = a \star b$. En particulier, $\big(\exists \tilde{a} \in \mathbf{a}\big)\big(\exists \tilde{b} \in \mathbf{b}\big)$ tel que $\tilde{c} = \tilde{a} \star \tilde{b}$. Ainsi, $f(x,\tilde{c},\tilde{y}) = f(x,\tilde{a}\star\tilde{b},\tilde{y})$. On a alors $\min_{a\in\mathbf{a}} \min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y) \leq f(x,\tilde{a}\star\tilde{b},\tilde{y}) = f(x,\tilde{c},\tilde{y})$, c'est à dire,

$$\min_{a\in\mathbf{a}} \min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y) \leq \min_{c\in\mathbf{c}} \min_{y\in\mathbf{y}} f(x,c,y).$$

Montrons (3.12).

Supposons que $\max_{a\in\mathbf{a}} \max_{b\in\mathbf{b}} \Big(\min_{y\in\mathbf{y}} f(x,a\star b,y)\Big)$ soit atteint en $(\tilde{a},\tilde{b})$. On a $\big(\forall a \in \mathbf{a}\big)\big(\forall b \in \mathbf{b}\big)\big(\exists c \in \mathbf{c}\big) \mid c = a\star b$. En particulier, il existe $\tilde{c} \in \mathbf{c}$ satisfaisant $\tilde{c} = \tilde{a} \star \tilde{b}$. On peut également définir $\tilde{y}$ tel que $\min_{y\in\mathbf{y}} f(x,\tilde{c},y)$ soit atteint en $\tilde{y}$. Ainsi, d'une part, $\max_{a\in\mathbf{a}} \max_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y) = \min_{y\in\mathbf{y}} f(x,\tilde{a}\star\tilde{b},y) \leq f(x,\tilde{a}\star\tilde{b},\tilde{y}) = f(x,\tilde{c},\tilde{y})$. D'autre part, $f(x,\tilde{c},\tilde{y}) = \min_{y\in\mathbf{y}} f(x,\tilde{c},y) \leq \max_{c\in\mathbf{c}} \min_{y\in\mathbf{y}} f(x,c,y)$. Cette chaîne d'inégalités se compacte en :

$$\max_{a\in\mathbf{a}} \max_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y) \leq \max_{c\in\mathbf{c}} \min_{y\in\mathbf{y}} f(x,c,y).$$

Montrons (3.13).

Soient $\tilde{c}$ et $\tilde{y}$ tels que $\min_{c\in\mathbf{c}} \min_{y\in\mathbf{y}} f(x,c,y) = f(x,\tilde{c},\tilde{y})$. Supposons maintenant que $\max_{a\in\mathbf{a}} \Big(\min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y)\Big)$ soit atteint en $\tilde{a}$. Comme $\mathbf{c}$ est propre alors d'après la proposition 3.1 $\big(\forall c \in \mathbf{c}\big)\big(\forall a \in \mathbf{a}\big)\big(\exists b \in \mathbf{b}\big)$ tels que $c = a \star b$. En particulier, $\big(\exists \tilde{b} \in \mathbf{b}\big)$ tel que $\tilde{c} = \tilde{a} \star \tilde{b}$. Ainsi, d'une part, $\max_{a\in\mathbf{a}} \min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y) = \min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,\tilde{a}\star b,y) \leq f(x,\tilde{a}\star\tilde{b},\tilde{y})$. D'autre part, $f(x,\tilde{a}\star\tilde{b},\tilde{y}) = f(x,\tilde{c},\tilde{y}) = \min_{c\in\mathbf{c}} \min_{y\in\mathbf{y}} f(x,c,y)$. Cette chaîne d'inégalités se compacte en :

$$\max_{a\in\mathbf{a}} \min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y) \leq \min_{c\in\mathbf{c}} \min_{y\in\mathbf{y}} f(x,c,y).$$

Montrons (3.14).

Supposons que $\max_{a\in\mathbf{a}} \Big(\min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y)\Big)$ soit atteint en $\tilde{a}$. La proposition 3.2 nous donne : $\big(\forall a \in \mathbf{a}\big)\big(\exists b \in \mathbf{b}\big)\big(\exists c \in (\text{dual } \mathbf{c})\big) \mid c = a \star b$. En particulier, il existe un couple $(\tilde{b},\tilde{c})$ de réels dans $\mathbf{b} \times (\text{dual } \mathbf{c})$ satisfaisant $\tilde{c} = \tilde{a} \star \tilde{b}$. On peut également définir $\tilde{y}$ tel que $\min_{y\in\mathbf{y}} f(x,\tilde{c},y)$ soit atteint en $\tilde{y}$. Ainsi, d'une part, $\max_{a\in\mathbf{a}} \min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y) = \min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,\tilde{a}\star b,y) \leq f(x,\tilde{a}\star\tilde{b},\tilde{y}) = f(x,\tilde{c},\tilde{y})$. D'autre part, $f(x,\tilde{c},\tilde{y}) = \min_{y\in\mathbf{y}} f(x,\tilde{c},y) \leq \max_{c\in(\text{dual } \mathbf{c})} \min_{y\in\mathbf{y}} f(x,c,y)$. Cette chaîne d'inégalités se compacte en :

$$\max_{a\in\mathbf{a}} \min_{b\in\mathbf{b}} \min_{y\in\mathbf{y}} f(x,a\star b,y) \leq \max_{c\in(\text{dual } \mathbf{c})} \min_{y\in\mathbf{y}} f(x,c,y).$$

□

Proposition 3.5 (Fonction élémentaire, cas général)
Soient $\mathbf{a} \in \mathbb{IR}$, $\mathbf{x} \in \mathbb{IR}^p$, $\mathbf{y} \in \mathbb{IR}^q$, *et* $f : \mathbb{R}^{p+q+1} \to \mathbb{R}$ *continue. Soit* g *une fonction élémentaire,* $\mathbf{a}$ *étant inclus*

dans le domaine de définition de g.

$$\bigwedge_{x\in\mathbf{x}}\bigvee_{a\in\mathbf{a}}\bigvee_{y\in\mathbf{y}} f(g(a),x,y) = \bigwedge_{x\in\mathbf{x}}\bigvee_{c\in\mathbf{c}}\bigvee_{y\in\mathbf{y}} f(c,x,y) \text{ avec } \mathbf{c} = g(\mathbf{a}) \tag{3.15}$$

$$\bigwedge_{x\in\mathbf{x}}\bigwedge_{a\in\mathbf{a}}\bigvee_{y\in\mathbf{y}} f(g(a),x,y) = \bigwedge_{x\in\mathbf{x}}\bigwedge_{c\in\mathrm{dual}\,(\mathbf{c})}\bigvee_{y\in\mathbf{y}} f(c,x,y) \text{ avec } \mathbf{c} = g(\mathrm{dual}\,(\mathbf{a}) \tag{3.16}$$

Preuve. La preuve est évidente si on note que $g(\mathbf{a}) = range(g,\mathbf{a})$. □

Remarque 3.2 *La proposition 3.4 aurait pu être donnée sous une forme plus générale, avec exactement la même preuve. Pour une fonction g continue quelconque, la relation* (3.13) *peut en effet être remplacée par*

$$\bigwedge_{x\in\mathbf{x}}\bigwedge_{a\in\mathbf{a}}\bigvee_{b\in\mathbf{b}}\bigvee_{y\in\mathbf{y}} f(g(a,b),x,y) \supseteq \bigwedge_{x\in\mathbf{x}}\bigvee_{c\in\mathbf{c}}\bigvee_{y\in\mathbf{y}} f(c,x,y) \text{ avec } \mathbf{c} \subseteq \bigwedge_{a\in\mathbf{a}}\bigvee_{b\in\mathbf{b}} g(a,b) \tag{3.17}$$

dans la mesure où l'intervalle $\mathbf{c}$ *est propre. De même, la relation* (3.14) *peut être remplacée par*

$$\bigwedge_{x\in\mathbf{x}}\bigwedge_{a\in\mathbf{a}}\bigvee_{b\in\mathbf{b}}\bigvee_{y\in\mathbf{y}} f(g(a,b),x,y) \supseteq \bigwedge_{x\in\mathbf{x}}\bigwedge_{c\in\mathrm{dual}\,(\mathbf{c})}\bigvee_{y\in\mathbf{y}} f(c,x,y) \text{ avec } \mathbf{c} \supseteq \bigvee_{a\in\mathbf{a}}\bigwedge_{b\in\mathbf{b}} g(a,b) \tag{3.18}$$

dans le cas contraire.

Nous en venons enfin à une première proposition permettant d'approximer intérieurement une image quantifiée, où la fonction impliquée a un caractère plus général (c'est à dire : arithmétique, mais avec une seule occurrence de chaque variable).

Proposition 3.6 *Soient $\mathbf{x}\in\mathbb{IR}^p$, $\mathbf{y}\in\mathbb{IR}^q$ et f une fonction arithmétique continue de $\mathbb{IR}^{p+q}$ dans $\mathbb{IR}$, telle que chaque variable x_i ou y_j apparaisse au plus une fois dans l'expression de f.*

$$f((\mathrm{dual}\ \mathbf{x}),\ \mathbf{y}) \subseteq \bigwedge_{x\in\mathbf{x}}\bigvee_{y\in\mathbf{y}} f(x,y).$$

Preuve. Le raisonnement se fait par induction sur l'arbre de syntaxe de f. Rappelons que chaque feuille de f correspond à une variable. Posons

$$\mathcal{S}^0 := \bigwedge_{x\in\mathbf{x}}\bigvee_{y\in\mathbf{y}} f(x,y) \tag{3.19}$$

Considérons un arbre d'évaluation $f^{(0)}$, identique à celui de f, où chaque feuille représentant une composante x_i de x est remplacée par l'intervalle (dual $\mathbf{x}_i$), et chaque feuille représentant une composante y_j par $\mathbf{y}_j$.

Construisons alors une suite d'arbre d'évaluation $(f^{(n)})_{n\geq 1}$ de la façon suivante. Soit on prend dans $f^{(n)}$ un couple quelconque de feuilles $(\mathbf{a},\mathbf{b})$ ayant même nœud-père (représentant un opérateur $\star$), et on construit $f^{(n+1)}$ en remplaçant dans $f^{(n)}$ le sous-arbre $(\star,\mathbf{a},\mathbf{b})$ par $\mathbf{c} := \mathbf{a}\star\mathbf{b}$. Soit on prend dans $f^{(n)}$ un sous-arbre $(f,\mathbf{a})$ (f est une fonction élémentaire) et on le remplace par $\mathbf{c} := f(\mathbf{a})$.

Ce procédé étant répété jusqu'à l'étape $n = k$ où l'arbre $f^{(k)}$ est réduit à une seule feuille. Clairement, cette feuille représente l'intervalle $f((\mathrm{dual}\ \mathbf{x}),\ \mathbf{y})$.

Construisons maintenant la suite d'ensembles $(\mathcal{S}^{(n)})_{1\leq n\leq k}$ définie ainsi :

$$\mathcal{S}^{(n)} := \bigwedge_{dual(feuilles\ impropres)} \bigvee_{feuilles\ propres} f^{(n)}(feuilles)$$

Montrons que pour tout n $(0 \leq n < k)$ $\mathcal{S}^{(n+1)} \subseteq \mathcal{S}^{(n)}$.

Prenons n quelconque, et considérons le sous-arbre $(\star, \mathbf{a}, \mathbf{b})$ de $f^{(n)}$ remplacé par $\mathbf{c} := \mathbf{a} \star \mathbf{b}$ dans $f^{(n+1)}$.

Si $\mathbf{a} \in \mathbb{IR}$ et $\mathbf{b} \in \mathbb{IR}$ alors, a et b sont quantifiés $\exists$ dans $\mathcal{S}^{(n)}$. $\mathbf{c}$ est forcément propre, donc quantifié $\exists$ dans $\mathcal{S}^{(n+1)}$. On applique alors l'inclusion (3.11) et on obtient $\mathcal{S}^{(n+1)} \subseteq \mathcal{S}^{(n)}$.

Si une seule des deux feuilles ($\mathbf{a}$ ou $\mathbf{b}$) est impropre, alors, sans perte de généralité, on peut supposer $\mathbf{a} \in \overline{\mathbb{IR}}$ et $\mathbf{b} \in \mathbb{IR}$ (par symétrie). Ainsi, $\mathbf{a}$ est quantifié $\forall$ et $\mathbf{b}$ est quantifié $\exists$ dans $\mathcal{S}^{(n)}$. Si $\mathbf{c}$ est propre, $\mathbf{c}$ est quantifié $\exists$ dans $\mathcal{S}^{(n+1)}$. On applique alors l'inclusion (3.13) et on obtient $\mathcal{S}^{(n+1)} \subseteq \mathcal{S}^{(n)}$. Si $\mathbf{c}$ est impropre, $\mathbf{c}$ est quantifié $\forall$ dans $\mathcal{S}^{(n+1)}$. On applique alors l'inclusion (3.14) et on obtient $\mathcal{S}^{(n+1)} \subseteq \mathcal{S}^{(n)}$.

Lorsque les deux feuilles sont impropres, $\mathbf{c}$ est forcément impropre et l'inclusion (3.12) aboutit encore à la même conclusion. Si le sous-arbre est $(f, \mathbf{a})$, on applique la proposition 3.5 de la même manière.

Finalement, par une récurrence immédiate, $\forall n \in [0..k],\ \mathcal{S}^{(n)} \subseteq \mathcal{S}^{(0)}$. En particulier, $\mathcal{S}^{(k)} \subseteq \mathcal{S}^{(0)}$.

Clairement, $f^{(k)}$ est un arbre composé d'une unique feuille dont l'intervalle associé $\mathbf{z}$ est $f((\text{dual } \mathbf{x}),\ \mathbf{y})$ (en tant que résultat de ce calcul). Si $\mathbf{z}$ est propre, alors $\mathcal{S}^{(k)} = \bigvee_{z\in\mathbf{z}} z \quad = \mathbf{z}$. Si $\mathbf{z}$ est impropre, alors $\mathcal{S}^{(k)} = \bigwedge_{z\in(\text{dual } \mathbf{z})} z =$ dual $(\bigvee_{z\in(\text{dual } \mathbf{z})} z) =$ dual $(\text{dual } \mathbf{z}) = \mathbf{z}$. Dans tous les cas,

$$\mathcal{S}^{(k)} = \mathbf{z} = f((\text{dual } \mathbf{x}),\ \mathbf{y}). \tag{3.20}$$

En utilisant (3.19), le fait que $\mathcal{S}^{(k)} \subseteq \mathcal{S}^{(0)}$ et (3.20), on obtient $f((\text{dual } \mathbf{x}),\ \mathbf{y}) \subseteq \bigwedge_{x\in\mathbf{x}} \bigvee_{y\in\mathbf{y}} f(x, y)$.

□

Le fait que dans la proposition précédente les vecteurs $\mathbf{x}$ et $\mathbf{y}$ soient des vecteurs quelconques d'intervalles généralisés, permet d'établir très facilement l'approximation extérieure suivante :

Corollaire 3.1 *Soient $\mathbf{x} \in \mathbb{IR}^p$, $\mathbf{y} \in \mathbb{IR}^q$ et f une fonction arithmétique continue de $\mathbb{IR}^{p+q}$ dans $\mathbb{IR}$, telle que chaque variable x_i ou y_j apparaisse au plus une fois dans l'expression de f.*

$$\bigvee_{y\in\mathbf{y}} \bigwedge_{x\in\mathbf{x}} f(x, y) \subseteq f((\text{dual } \mathbf{x}), \mathbf{y})$$

Preuve. La proposition 3.6 donne :

$$f(\mathbf{x}, (\text{dual } \mathbf{y})) \subseteq \bigwedge_{y\in\mathbf{y}} \bigvee_{x\in\mathbf{x}} f(x, y).$$

Par passage au dual, le sens de l'inclusion est inversé :

$$\text{dual } (\bigwedge_{y\in\mathbf{y}} \bigvee_{x\in\mathbf{x}} f(x, y)) \subseteq \text{dual } f(\mathbf{x}, (\text{dual } \mathbf{y})),$$

ce qui s'écrit également (d'après la proposition 3.3)

$$\bigvee_{y\in\mathbf{y}} \bigwedge_{x\in\mathbf{x}} f(x, y) \subseteq (f((\text{dual } \mathbf{x}), \mathbf{y})).$$

□

Remarque 3.3 *On n'a en général qu'une inclusion stricte dans la proposition 3.6. Voici le célèbre contre-exemple suivant, emprunté au groupe SIGLA/X [GnMA85, GnSJ+01]. Considérons* $\mathbf{x}_1 = [-2, 2]$, $\mathbf{x}_2 = [-1, 1]$, $\mathbf{x}_3 = [-1, 1]$ *et* $\mathbf{x}_4 = [-2, 2]$. *On peut vérifier "à la main" que*

$$\bigwedge_{x_1 \in \mathbf{x}_1} \bigwedge_{x_3 \in \mathbf{x}_3} \bigvee_{x_2 \in \mathbf{x}_2} \bigvee_{x_4 \in \mathbf{x}_4} (x_1 + x_2)(x_3 + x_4) = [-\frac{3}{2}, \frac{3}{2}],$$

alors que $(\text{dual}\,(\mathbf{x}_1) + \mathbf{x}_2)(\text{dual}\,(\mathbf{x}_3) + \mathbf{x}_4) = [0, 0] \subset [-\frac{3}{2}, \frac{3}{2}]$ *(cf. table 3.1).*

Il y a toutefois un cas notable où l'égalité demeure, c'est celui où f est une somme de produits. Une des avancées récentes de la théorie des intervalles modaux réside dans l'introduction de linéarisations [Gol05b, Gol07c], et le cas particulier considéré ci-dessous s'appliquera en fait pour les fonctions mises sous forme de Taylor. Il est donc d'une grande importance.

Proposition 3.7 *Pour toute fonction* $f : \mathbb{R}^n \to \mathbb{R}$, *telle que*

$$f(x) = x_{\sigma(1)}x_{\sigma(2)} + \ldots + x_{\sigma(n-1)}x_{\sigma(n)}$$

où σ *est une permutation de* $[1..n]$, *et pour tout* $\mathbf{x} \in \mathbb{IR}^p$ *et* $\mathbf{y} \in \mathbb{IR}^q$ *avec* $p + q = n$,

$$\bigvee_{y \in \mathbf{y}} \bigwedge_{x \in \mathbf{x}} f(x, y) = f((\text{dual}\ \mathbf{x}),\ \mathbf{y}) = \bigwedge_{x \in \mathbf{x}} \bigvee_{y \in \mathbf{y}} f(x, y).$$

Preuve. Il suffit de prouver que $\bigvee_{y \in \mathbf{y}} \bigwedge_{x \in \mathbf{x}} f(x, y) = \bigwedge_{x \in \mathbf{x}} \bigvee_{y \in \mathbf{y}} f(x, y)$, puisque de par la proposition 3.6 et le corollaire 3.1 l'évaluation $f((\text{dual}\ \mathbf{x}), \mathbf{y})$ est "coincée" entre les deux. Faisons-le par exemple pour la borne inférieure de ces deux intervalles. Dans la récurrence suivante, on omet pour la lisibilité de spécifier à chaque fois que le *max* vaut pour $x \in \mathbf{x}$ et le *min* pour $y \in \mathbf{y}$. De plus, pour que les notations soient homogènes, on pose $x_{p+1} := y_1, \ldots, x_n := y_q$.

$$\begin{aligned}
\inf(\bigwedge_{x \in \mathbf{x}} \bigvee_{y \in \mathbf{y}} f(x, y)) &= \max(\min(x_{\sigma(1)}x_{\sigma(2)} + \ldots + x_{\sigma(n-1)}x_{\sigma(n)})) \\
&= \max(\min x_{\sigma(1)}x_{\sigma(2)} + \ldots + \min x_{\sigma(n-1)}x_{\sigma(n)}) && \text{(distributivité du min)} \\
&= \max \min x_{\sigma(1)}x_{\sigma(2)} + \ldots + \max \min x_{\sigma(n-1)}x_{\sigma(n)} && \text{(distributivité du max)} \\
&= \min \max x_{\sigma(1)}x_{\sigma(2)} + \ldots + \min \max x_{\sigma(n-1)}x_{\sigma(n)} && \text{(hypothèse de récurrence)} \\
&= \min(\max x_{\sigma(1)}x_{\sigma(2)} + \ldots + \max x_{\sigma(n-1)}x_{\sigma(n)}) && \text{(distributivité du min)} \\
&= \min(\max(x_{\sigma(1)}x_{\sigma(2)} + \ldots + x_{\sigma(n-1)}x_{\sigma(n)})) && \text{(distributivité du max)} \\
&= \inf(\bigvee_{y \in \mathbf{y}} \bigwedge_{x \in \mathbf{x}} f(x, y)).
\end{aligned}$$

□

3.4 Résolution d'AE-systèmes

En nous appuyant sur notre étude des images quantifiées, nous allons maintenant décrire quelques techniques permettant la résolution d'AE-systèmes. Il faut noter que ces techniques n'ont été applicables à ce jour que sur des problèmes académiques et que de nombreux progrès restent à faire. L'ensemble de ce qui est décrit ici est dû à Goldsztejn [Gol05b].

Dans ce §3.4 nous considérons une fonction continue $f : \begin{array}{ll} \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q & \to \mathbb{R}^m \\ (x,u,v) & \mapsto f(x,u,v) \end{array}$.

Nous fixons $\mathbf{u} \in \mathbb{IR}^p$ et $\mathbf{v} \in \mathbb{IR}^q$ et noterons $\Sigma(f,\mathbf{u},\mathbf{v})$ le AE-système correspondant (cf. §1.1.2). Nous considérons enfin une boîte $\mathbf{x} \in \mathbb{IR}^n$ quelconque. Rappelons que x désigne le vecteur de variables, u le vecteur de paramètres universellement quantifiés et v le vecteur de paramètres existentiellement quantifiés.

3.4.1 Test intérieur

Le plus simple à considérer est le test intérieur, dans le cas où les paramètres existentiels apparaissent une seule fois dans l'expression de f :

Lemme 3.5 (Test intérieur) *Si chaque composante de v n'apparaît au plus qu'une fois dans l'expression de f, alors,*

$$0 \subseteq f(\text{dual } \mathbf{x}, \text{dual } \mathbf{u}, \mathbf{v}) \Longrightarrow \mathbf{x} \textit{ est une boîte intérieure.}$$

Preuve. Quitte à en réordonner les composantes, on peut partitionner le vecteur v en m sous-vecteurs $v_1, \ldots, v_m$ tels que pour tout $i \in [1..m]$, les composantes de v_i n'apparaissent que dans f_i (et une seule fois). On peut donc écrire $f(x,u,v) = (f_1(x,u,v_1), \ldots, f_m(x,u,v_m))^T$. Or,

$$\begin{array}{lll} 0 \subseteq f(\text{dual } \mathbf{x}, \text{dual } \mathbf{u}, \mathbf{v}) & \Longrightarrow \big(\forall i \in [1..m]\big)\ 0 \subseteq f_i(\text{dual } \mathbf{x}, \text{dual } \mathbf{u}, \mathbf{v}_i) & \\ & \Longrightarrow \big(\forall i \in [1..m]\big)\ 0 \subseteq \bigwedge_{x \in \mathbf{x}} \bigwedge_{u \in \mathbf{u}} \bigvee_{v_i \in \mathbf{v}_i} f_i(x,u,v_i) & \text{(prop. 3.6)} \\ & \Longrightarrow \big(\forall i \in [1..m]\big)\big(\forall x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\exists v_i \in \mathbf{v}_i\big)\ f_i(x,u,v_i) = 0 & \text{(prop. 3.1)} \\ & \Longrightarrow \big(\forall x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\ f(x,u,v) = 0. & \end{array}$$

□

Lorsqu'un paramètre existentiel possède plusieurs occurrences, il est possible d'étendre ce test en *rendant universelles toutes les occurrences de ce paramètre sauf une*. Ce résultat n'est applicable que sous certaines conditions que nous allons étudier maintenant.

3.4.2 Occurrences multiples de paramètres existentiels

Par simplicité, nous ne considérons pas dans ce paragraphe de paramètres u universels puisqu'ils peuvent être assimilés à des variables dans le cadre de détection de boîtes intérieures.

Illustrons notre propos tout d'abord sur une fonction à valeurs réelles. Soit $f : \mathbb{R}^2 \to \mathbb{R}$ de variable $x \in \mathbb{R}$ et de paramètre $v \in \mathbb{R}$, v apparaissant deux fois dans l'expression de f. Considérons maintenant la fonction $g : \mathbb{R}^3 \to \mathbb{R}$ de variable $x \in \mathbb{R}$ et de paramètres $v_1 \in \mathbb{R}$ et $v_2 \in \mathbb{R}$, dont l'expression est celle de f où une occurrence de v est remplacée par v_1, l'autre par v_2. Exemple :

$$f(x,v) = ln(x-v) - v \qquad g(x,v_1,v_2) := ln(x-v_1) - v_2.$$

Si on observe l'une des inclusions suivantes :

$$0 \subseteq g(\text{dual } \mathbf{x}, \text{dual } \mathbf{v}, \mathbf{v}) \quad \text{ou} \quad 0 \subseteq g(\text{dual } \mathbf{x}, \mathbf{v}, \text{dual } \mathbf{v})$$

alors $\mathbf{x}$ est une boîte intérieure. Supposons par exemple $0 \subseteq g(\text{dual } \mathbf{x}, \text{dual } \mathbf{v}, \mathbf{v})$. Puisqu'il n'y a plus d'occurrence multiple de paramètres, la proposition 3.6 (puis la proposition 3.1) s'applique à g et

$$\big(\forall x \in \mathbf{x}\big)\big(\forall v_1 \in \mathbf{v}\big)\big(\exists v_2 \in \mathbf{v}\big)\ g(x,v_1,v_2) = 0.$$

Cette dernière relation implique-t-elle bien $\big(\forall x \in \mathbf{x}\big)\big(\exists v \in \mathbf{v}\big)\ g(x,v,v) = 0$? En général, la réponse est non. Il est possible que, quel que soit v_1, la valeur de v_2 permettant de satisfaire $g(x,v_1,v_2) = 0$ soit systématiquement différente de v_1, auquel cas il ne peut exister de valeur $v = v_1 = v_2$ satisfaisant $g(x,v,v) = 0$, i.e., $f(x,v) = 0$. Cependant, dans de nombreux cas comme celui de notre exemple, la réponse est oui. L'idée repose sur le principe de continuité des paramètres existentiellement quantifiés par rapport aux paramètres universellement quantifiées [Gol05b] : Tout d'abord, si on fixe x, puisque $\big(\forall v_1 \in \mathbf{v}\big)\big(\exists v_2 \in \mathbf{v}\big)\ g(x,v_1,v_2) = 0$, il est possible, pour chaque valeur v_1 (dans $\mathbf{v}$) d'associer une valeur v_2 (également dans $\mathbf{v}$) telle que $g(x,v_1,v_2) = 0$; il est donc possible de définir une fonction (peut-être plusieurs) $\phi_x : \mathbf{v} \to \mathbf{v}$ telle que $v_2 = \phi_x(v_1) \Longrightarrow g(x,v_1,v_2) = 0$. On remarque alors que ϕ étant une fonction d'un intervalle $\mathbf{v}$ dans lui-même, il suffit que cette fonction soit continue pour que le théorème de Brouwer s'applique ! Donc si ϕ est continue, il existe bien v dans $\mathbf{v}$ tel que $v = \phi_x(v)$, i.e., tel que $f(x,v) = g(x,v,v) = 0$.

Dans notre exemple, en prenant $\mathbf{x} = [5,6]$ et $\mathbf{v} = [1,2]$, l'évaluation $g(\text{dual } \mathbf{x}, \text{dual } \mathbf{v}, \mathbf{v})$ donne $[-0.39, 0.098]$, qui contient bien 0. Clairement, $g(x,v_1,v_2) = 0 \iff v_2 = ln(x - v_1)$. La seule fonction ϕ possible est $\phi_x(v_1) = ln(x - v_1)$, et elle est bien continue sur $[1,2]$. Donc $\mathbf{x}$ est intérieur. Par ailleurs, cette fonction étant forcément "la bonne", elle doit vérifier pour tout x, $\phi_x(\mathbf{v}) \subseteq \mathbf{v}$, ce qu'on peut vérifier facilement en calculant $ln(\mathbf{x} - \mathbf{v}) = ln([5,6] - [1,2]) = [1.098, 1.609]$. En revanche, $g(\text{dual } \mathbf{x}, \mathbf{v}, \text{dual } \mathbf{v}) = [0.386, -0.614]$ ne contient pas 0.

Avec la même fonction, en prenant $\mathbf{x} := [-0.3, 0.08]$ et $\mathbf{v} = [-2,0]$ on est dans le cas contraire. C'est $g(\text{dual } \mathbf{x}, \mathbf{v}, \text{dual } \mathbf{v})$ qui vaut l'intervalle $[-0.526, 0.53]$ contenant 0. Comme $g(x,v_1,v_2) = 0 \iff v_1 = x - exp(v_2)$. La seule fonction ϕ possible est $\phi_x(v_2) = x - exp(v_2)$, qui est bien continue sur $[-2,0]$ pour $x \in [-0.3, 0.08]$. Donc $\mathbf{x}$ est intérieure. On vérifie de nouveau au passage que $\mathbf{x} - exp(\mathbf{v}) = [-1.3, -0.055] \subseteq \mathbf{v}$.

Une fonction telle que ϕ associant implicitement un paramètre existentiel à un paramètre universel dans une proposition quantifiée s'appelle une *fonction de Skolem* [Gol05b]. Les fonctions de Skolem permettent également de traiter le cas des paramètres ayant des occurrences dans plusieurs composantes de f. Prenons le cas d'une fonction de $\mathbb{R}^3$ dans $\mathbb{R}^2$, de variable $x \in \mathbb{R}$ et de paramètres $v \in \mathbb{R}^2$.
Supposons $\begin{cases} 0 \subseteq f_1(\text{dual } \mathbf{x}, \text{dual } \mathbf{v}_1, \mathbf{v}_2) \\ \qquad\qquad \text{et} \\ 0 \subseteq f_2(\text{dual } \mathbf{x}, \mathbf{v}_1, \text{dual } \mathbf{v}_2) \end{cases}$. Prenons alors $x \in \mathbf{x}$ quelconque, et soient ϕ_x et ψ_x deux fonctions de Skolem telles que $\begin{cases} \big(\forall v_1 \in \mathbf{v}_1\big)\ v_2 = \phi_x(v_1) \Longrightarrow f_1(x,v_1,v_2) = 0 \\ \big(\forall v_2 \in \mathbf{v}_2\big)\ v_1 = \psi_x(v_2) \Longrightarrow f_2(x,v_1,v_2) = 0 \end{cases}$.
Si ϕ_x et ψ_x sont continues respectivement sur $\mathbf{v}_1$ et $\mathbf{v}_2$, alors $\mathbf{x}$ est intérieure. En effet, la fonction $(v_2,v_1) \mapsto (\phi_x(v_1), \psi_x(v_2))$ est alors une fonction continue de $\mathbf{v}$ dans lui-même. Le théorème de Brouwer s'applique et ainsi, il existe un couple (v_2,v_1) vérifiant $(v_2,v_1) = (\phi_x(v_1), \psi_x(v_2))$, ce qui implique simultanément $f_1(x,v_1,v_2) = 0$ et $f_2(x,v_1,v_2) = 0$.

La continuité des fonctions de Skolem permet donc de traiter de façon élégante le problème des occurrences multiples de paramètres. Goldsztejn [Gol05b, Gol07b] place ce concept au coeur de sa formulation de la théorie des intervalles modaux. Bien que ce soit ingénieux, il nous semble toutefois que nous contournons le problème plutôt que nous le résolvons vraiment, et que ce contournement est souvent impossible. Il suffit pour s'en convaincre de prendre l'exemple suivant :

Exemple 3.1 *Soit* $f : (x,v) \mapsto ((x_1 - v_1)^2 + (x_2 - v_2)^2 - 1, (v_2 - x_2)/(v_1 - x_1) - 1)$ *de telle sorte que* $f(x,v) = 0$ *ssi le point* x *est à distance* 1 *du point* v *et la pente formée par les points* x *et* v *vaut* 1. *Il est clair qu'alors pour toute boîte* v *du paramètre, le translaté de* v *par le vecteur* $-\frac{1}{2}(\sqrt{2}, \sqrt{2})$ *est une boîte intérieure pour* x *(voir figure 3.2). Or, le test ci-dessus ne pourra pas le vérifier :*

Considérons la boîte $\mathbf{v} = [-2,2] \times [-2,2]$. *L'approche des intervalles modaux ne peut qu'effectuer les calculs suivants pour prouver qu'une boîte* $\mathbf{x}$ *est intérieure :*

$$(\mathbf{z}_1, \mathbf{z}_2) := \Big(f_1(\text{dual } \mathbf{x}_1, \text{dual } \mathbf{x}_2, \text{dual } \mathbf{v}_1, \mathbf{v}_2),\ f_2(\text{dual } \mathbf{x}_1, \text{dual } \mathbf{x}_2, \mathbf{v}_1, \text{dual } \mathbf{v}_2)\Big).$$

Or $f_1(\text{dual } \mathbf{x}_1, \text{dual } \mathbf{x}_2, \text{dual } \mathbf{v}_1, \mathbf{v}_2)$ *est forcément impropre. En effet, sinon cela impliquerait*

$$\big(\forall x \in \mathbf{x}\big)\big(\forall v_2 \in \mathbf{v}_2\big)\big(\exists v_1 \in \mathbf{v}_1\big) \quad (x_1 - v_1)^2 + (x_2 - v_2)^2 = 1,$$

ce qui est absurde (un point x *ne peut être à distance* 1 *d'un ensemble de points* (v_1, v_2) *avec* $v_2 \in [-2, 2]$*).*

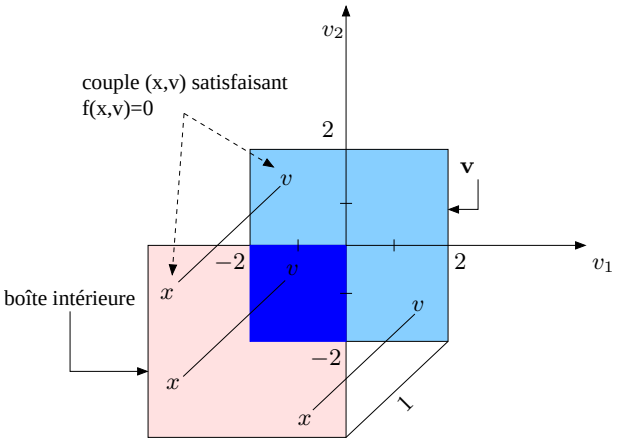


FIG. *3.2: Illustration de l'exemple 3.1*

Ce genre d'exemple illustre le fait que notre test de boîte intérieure (c'est à dire, l'évaluation aux intervalles généralisées sous continuité des fonctions de Skolem) implique une propriété beaucoup plus forte que celle de boîte intérieure. Cette propriété dit simplement que toutes les occurrences (sauf une) d'un même paramètre doivent pouvoir être quantifiées universellement, ce qui est la plupart du temps absurde.

En conclusion, les intervalles modaux ne peuvent intervenir pour la détection de boîtes intérieures que dans le cas où chaque paramètre est limité à une seule occurrence : nous appliquons l'évaluation aux intervalles généralisés (lemme 3.5). Notons toutefois que dans le cas où le système peut se ramener à une seule équation, les intervalles modaux peuvent également être utilisés quel que soit le nombre d'occurrences des paramètres (voir [HSVJ05]). Ces différents cas s'ajoutent à ceux que nous avons traités dans les paragraphes 2.9.4, 2.9.5 et 2.9.6.

Ceci clôt la question des boîtes intérieures pour ce chapitre. La suite concerne l'exclusion et le filtrage de boîtes pour la résolution d'AE-systèmes.

3.4.3 Test d'infaisabilité

Au tout début de ce chapitre, nous avons donné dans le lemme 3.2 un test d'infaisabilité, qui peut s'écrire sous une forme plus faible :

Lemme 3.6 *Soit* $\mathbf{z} \in \mathbb{KR}^m$ *tel que*

$$\big(\forall i \in [1..m]\big) \quad \mathbf{z}_i \supseteq \bigwedge_{u \in \mathbf{u}} \bigvee_{x \in \mathbf{x}} \bigvee_{v \in \mathbf{v}} f_i(x, u, v).$$

Si $0 \not\subseteq \mathbf{z}$ *alors* $\mathbf{x}$ *est infaisable.*

Preuve.

$$\begin{aligned}
0 \not\subseteq \mathbf{z} &\implies \big(\exists i \in [1..m]\big)\ 0 \notin \bigwedge_{u\in\mathbf{u}} \bigvee_{x\in\mathbf{x}} \bigvee_{v\in\mathbf{v}} f_i(x,u,v) \\
&\implies \big(\exists i \in [1..m]\big)\big(\exists u \in \mathbf{u}\big)\big(\forall x \in \mathbf{x}\big)\big(\forall v \in \mathbf{v}\big)\ f_i(x,u,v) \neq 0 \\
&\implies \neg\Big(\big(\forall i \in [1..m]\big)\big(\forall u \in \mathbf{u}\big)\big(\exists x \in \mathbf{x}\big)\big(\exists v \in \mathbf{v}\big)\ f_i(x,u,v) = 0\Big) \\
&\implies \neg\Big(\big(\exists x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\big(\forall i \in [1..m]\big)\ f_i(x,u,v) = 0\Big) \\
&\implies \neg\Big(\big(\exists x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\exists v \in \mathbf{v}\big)\ f(x,u,v) = 0\Big).
\end{aligned}$$

□

Malheureusement, l'évaluation dans $\mathbb{KR}$ ne nous procure en général qu'une approximation (strictement) intérieure d'une image quantifiée de la forme $\bigwedge\bigvee$ (cf. proposition 3.6 et l'exemple qui suit). Il reste toutefois le cas décrit dans la proposition 3.7, où l'égalité demeure. Notre but est de linéariser la fonction f pour pouvoir entrer dans ce cas de figure.

3.4.4 Linéarisations

En linéarisant, le nombre de paramètres croît, et les formules quantifiées deviennent vite fastidieuses à écrire. Pour soulager le texte, nous allons autoriser temporairement quelques abus d'écriture. Tout d'abord, fixons trois points $\tilde{x}$, $\tilde{u}$ et $\tilde{v}$ respectivement dans les boîtes $\mathbf{x}$, $\mathbf{u}$ et $\mathbf{v}$.

Plaçons-nous dans le cas $m = 1$ (f à valeurs réelles). Par analogie avec la proposition 2.8 p.27, considérons alors trois vecteurs-lignes intervalles $\mathbf{g}_x$, $\mathbf{g}_u$ et $\mathbf{g}_v$ tels que :

$$\begin{array}{l}
\mathbf{g}_x \supseteq df_x(\mathbf{x},\mathbf{u},\mathbf{v}) \\
\mathbf{g}_u \supseteq df_u(\tilde{x},\mathbf{u},\mathbf{v}) \\
\mathbf{g}_v \supseteq df_v(\tilde{x},\tilde{u},\mathbf{v})
\end{array} \tag{3.21}$$

où
– $df_x(\mathbf{x},\mathbf{u},\mathbf{v})$ est le gradient de f suivant le vecteur de variables x, calculé sur la boîte $\mathbf{x}\times\mathbf{u}\times\mathbf{v}$.
– $df_u(\tilde{x},\mathbf{u},\mathbf{v})$ est le gradient de f suivant le vecteur de paramètres u, calculé sur la boîte $\mathbf{u}\times\mathbf{v}$ en $x = \tilde{x}$.
– $df_v(\tilde{x},\tilde{u},\mathbf{v})$ est le gradient de f suivant le vecteur de paramètres v, calculé sur l'intervalle $\mathbf{v}$ en $(x,u) = (\tilde{x},\tilde{u})$.
On définit alors la fonction g suivante de $\mathbf{x}\times\mathbf{u}\times\mathbf{v}\times\mathbf{g}_x\times\mathbf{g}_u\times\mathbf{g}_v$ dans $\mathbb{R}$:

$$g(x,u,v,g_x,g_u,g_v) := f(\tilde{x},\tilde{u},\tilde{v}) + g_x(x-\tilde{x}) + g_u(u-\tilde{u}) + g_v(v-\tilde{v}). \tag{3.22}$$

L'abus de notation suivant sera alors autorisé : il consiste à écrire u (resp. g_u, etc...) au lieu de $(u \in \mathbf{u})$ (resp. $(g_u \in \mathbf{g}_u)$, etc...).

Le lemme suivant n'est qu'une version "quantifiée" de la proposition 2.8 relative à l'extension de Hansen. Elle montre que l'inclusion reste valable lorsque u n'est pas quantifiée existentiellement (comme dans le cas classique), mais universellement.

Lemme 3.7 *Si $m = 1$ (f est à valeurs dans $\mathbb{R}$) alors $\mathbf{x}$ est faisable seulement si*

$$0 \subseteq \bigwedge_{u} \bigvee_{(x,v)} f(x,u,v) \subseteq \bigwedge_{u} \bigvee_{(x,v,g_x,g_u,g_v)} g(x,u,v,g_x,g_u,g_v),$$

où $\mathbf{g}_x$, $\mathbf{g}_u$, $\mathbf{g}_v$ et g sont définis par les relations (3.21) *et* (3.22).

Preuve.

Tout d'abord, d'après le lemme 3.6, si $\mathbf{x}$ est faisable alors $0 \subseteq \bigwedge_{u} \bigvee_{(x,v)} f(x,u,v)$. En particulier, $\bigwedge_{u} \bigvee_{(x,v)} f(x,u,v)$ est propre.

Par ailleurs, nous pouvons effectuer trois linéarisations classiques successives, en procédant exactement de la même manière que pour l'extension de Hansen :

$$\begin{array}{rl} \big(\forall x \in \mathbf{x}\big)\big(\forall u \in \mathbf{u}\big)\big(\forall v \in \mathbf{v}\big)\big(\exists g_x \in \mathbf{g}_x\big) & f(x,u,v) = f(\tilde{x},u,v) + g_x(x-\tilde{x}) \\ \big(\forall u \in \mathbf{u}\big)\big(\forall v \in \mathbf{v}\big)\big(\exists g_u \in \mathbf{g}_u\big) & f(\tilde{x},u,v) = f(\tilde{x},\tilde{u},v) + g_u(u-\tilde{u}) \\ \big(\forall v \in \mathbf{v}\big)\big(\exists g_v \in \mathbf{g}_v\big) & f(\tilde{x},\tilde{u},v) = f(\tilde{x},\tilde{u},\tilde{v}) + g_v(v-\tilde{v}) \end{array}$$

Remarquons que x, u et v ont exactement le même rôle ici (leurs quantificateurs spécifiques ne sont pas pris en compte). Ces linéarisations peuvent se rassembler en une seule formule :

$$\begin{array}{c} \big(\forall x \in \mathbf{x}\big)\big(\forall u \in \mathbf{x}\big)\big(\forall v \in \mathbf{u}\big)\big(\exists g_x \in \mathbf{g}_x\big)\big(\exists g_u \in \mathbf{g}_u\big)\big(\exists g_v \in \mathbf{g}_v\big) \\ f(x,u,v) = f(\tilde{x},\tilde{u},\tilde{v}) + g_x(x-\tilde{x}) + g_u(u-\tilde{u}) + g_v(v-\tilde{v}). \end{array}$$

La formule précédente se réécrit dans nos nouvelles conventions de notation :

$$\forall x\ \forall u\ \forall v\ \exists g_x\ \exists g_u\ \exists g_v\quad f(x,u,v) = g(x,u,v,g_x,g_u,g_v). \tag{3.23}$$

Ceci nous amène directement à la seconde inclusion :

$$\begin{array}{rl} z \in \bigwedge\limits_u \bigvee\limits_{(x,v)} f(x,u,v) & \Longrightarrow \forall u\ \exists x\ \exists v\quad z = f(x,u,v) \\ & \Longrightarrow \forall u\ \exists x\ \exists v\ \exists g_x\ \exists g_u\ \exists g_v\quad z = g(x,u,v,g_x,g_u,g_v) \quad \text{(avec (3.23))} \\ & \Longrightarrow z \in \bigwedge\limits_u \bigvee\limits_{(x,v,g_x,g_u,g_v)} g(x,u,v,g_x,g_u,g_v). \end{array}$$

□

Cette linéarisation nous place dans une configuration où le test d'infaisabilité du lemme 3.6 peut être appliqué.

Proposition 3.8 (Test d'infaisabilité) *Soient $\mathbf{J}_x$, $\mathbf{J}_u$ et $\mathbf{J}_v$ trois matrices vérifiant*

$$\begin{array}{l} \mathbf{J}_x \supseteq J_x(\mathbf{x},\mathbf{u},\mathbf{v}), \\ \mathbf{J}_u \supseteq J_u(\tilde{x},\mathbf{u},\mathbf{v}), \\ \mathbf{J}_v \supseteq J_v(\tilde{x},\tilde{u},\mathbf{v}), \end{array}$$

où $J_x(\mathbf{x},\mathbf{u},\mathbf{v})$ est la jacobienne de f suivant x, calculé sur la boîte $\mathbf{x}\times\mathbf{u}\times\mathbf{v}$;
$J_u(\tilde{x},\mathbf{u},\mathbf{v})$ est la jacobienne de f suivant u, calculé sur $\mathbf{u}\times\mathbf{v}$ avec $x=\tilde{x}$;
$J_v(\tilde{x},\tilde{u},\mathbf{v})$ est la jacobienne de f suivant v, calculé sur $\mathbf{v}$ avec $(x,u)=(\tilde{x},\tilde{u})$.

$$\textit{Si } 0 \notin f(\tilde{u},\tilde{v},\tilde{x}) + \mathbf{J}_x(\mathbf{x}-\tilde{x}) + \mathbf{J}_u((\text{dual } \mathbf{u}) - \tilde{u}) + \mathbf{J}_v(\mathbf{v}-\tilde{v}) \textit{ alors } \mathbf{x} \textit{ est infaisable}$$

Preuve. *Posons* $\mathbf{z} := f(\tilde{u},\tilde{v},\tilde{x}) + \mathbf{J}_x(\mathbf{x}-\tilde{x}) + \mathbf{J}_u((\text{dual } \mathbf{u}) - \tilde{u}) + \mathbf{J}_v(\mathbf{v}-\tilde{v})$.

D'après le lemme 3.7, on a, pour tout $i \in [1..m]$,

$$\bigwedge_u \bigvee_{(x,v)} f_i(x,u,v) \subseteq \bigwedge_u \bigvee_{(x,v,g_x,g_u,g_v)} g(x,u,v,g_x,g_u,g_v),$$

où $\mathbf{g}_x$, $\mathbf{g}_u$, $\mathbf{g}_v$ représentent respectivement la $i^{ème}$ ligne de $\mathbf{J}_x$, $\mathbf{J}_u$ ou $\mathbf{J}_v$, et où g est défini par

$$g(x,u,v,g_x,g_u,g_v) := f_i(\tilde{x},\tilde{u},\tilde{v}) + g_x \cdot (x-\tilde{x}) + g_u \cdot (u-\tilde{u}) + g_v \cdot (v-\tilde{v}).$$

D'après la proposition 3.7, on a

$$\mathbf{z}_i = g(\mathbf{x},(\text{dual } \mathbf{u}),\mathbf{v},\mathbf{g}_x,\mathbf{g}_u,\mathbf{g}_v) = \bigwedge_{u\in\mathbf{u}} \bigvee_{(v,v,g_x,g_u,g_v)} g(x,u,v,g_x,g_u,g_v).$$

Donc $\big(\forall i \in [1..m]\big)\quad \mathbf{z}_i \supseteq \bigwedge\limits_u \bigvee\limits_{(x,v)} f_i(x,u,v)$. *Il suffit ensuite d'appliquer le lemme 3.6.* □

3.4.5 Filtrage généralisé de Newton

Tout se passe comme dans le cas classique, où la forme centrée calculée en une boîte dégénérée x (l'inconnue) permet de définir le filtrage de Newton par intervalles. Dans le chapitre 2, le filtrage de Newton ramène à chaque itération un système classique à un système linéaire où les coefficients $\mathbf{A}$ et $\mathbf{b}$ sont quantifiés existentiellement. Dans le cas d'un AE-système, le filtrage de Newton *généralisé* se ramène à un AE-système linéaire, que nous allons introduire brièvement en nous limitant à la sous-classe qui nous intéresse (celle comprenant les systèmes où seul $\mathbf{b}$ peut être affecté de quantificateurs universels). Une étude plus approfondie des AE-systèmes linéaires est menée au chapitre 5.

Définition 3.4 (AE-système linéaire « quantifié à droite »)
Soit $\mathbf{A} \in \mathbb{IR}^{n\times n}$, $\mathbf{b} \in \mathbb{KR}^n$. Construisons $\mathbf{b}_\exists$ et $\mathbf{b}_\forall$, les deux vecteurs de $\mathbb{KR}$ suivants :

$$\forall i \in [1..n] \quad (\mathbf{b}_\exists)_i := \begin{cases} \mathbf{b}_i & si\ \mathbf{b}_i \in \mathbb{IR} \\ 0 & sinon \end{cases} \qquad (\mathbf{b}_\forall) := \begin{cases} \mathbf{b}_i & si\ \mathbf{b}_i \in \overline{\mathbb{IR}} \\ 0 & sinon \end{cases},$$

de telle sorte que les composantes soient "réparties" dans $\mathbf{b}_\exists$ ou $\mathbf{b}_\forall$ suivant qu'elles sont quantifiées existentiellement ou universellement.

On note[3] *alors $\Sigma(\mathbf{A},\mathbf{b})$ l'ensemble solution du AE-système linéaire défini ainsi :*

$$\Sigma(\mathbf{A},\mathbf{b}) := \{x \mid \big(\forall b_\forall \in (\text{pro } \mathbf{b}_\forall)\big)\big(\exists b_\exists \in \mathbf{b}_\exists\big)\big(\exists A \in \mathbf{A}\big)\ Ax = b\}.$$

Nous verrons qu'il existe, tout comme pour les systèmes linéaires classiques, plusieurs manières de caractériser cet ensemble solution. Le lemme 2.2 p.30 de Beeck se retrouve sous la forme suivante, dont la démonstration sera donnée au chapitre 4 (cf. prop 5.2).

Lemme 3.8 (Shary-Beeck)

$$\big(\forall \mathbf{b} \in \mathbb{KR}^n\big) \quad x \in \Sigma(\mathbf{A},\mathbf{b}) \iff 0 \in \mathbf{A}x - \mathbf{b}.$$

Nous avons maintenant suffisamment d'éléments pour pouvoir énoncer le filtrage de Newton pour les AE-systèmes. La proposition suivante [Gol05b, Gol07c] est la seule à notre connaissance qui propose un moyen d'approximer extérieurement les solutions d'un AE-système (sans ignorer les quantificateurs universels), dans le cas général (c.a.d., sans supposer quoi que ce soit sur la forme de f).

Proposition 3.9 (Goldsztejn) *Avec les notations de la proposition 3.8,*

$$x \in \Sigma(f,\mathbf{u},\mathbf{v}) \Longrightarrow x \in \tilde{x} + \Sigma(\mathbf{A},\mathbf{b}),$$

avec $\mathbf{A} = \mathbf{J}_x$ et $\mathbf{b} = -f(\tilde{x},\tilde{u},\tilde{v}) - \mathbf{J}_u((\text{dual } \mathbf{u}) - \tilde{u}) - \mathbf{J}_v(\mathbf{v} - \tilde{v})$.

Preuve. D'après la contraposée de la proposition 3.8, si $x \in \Sigma(f,\mathbf{u},\mathbf{v})$, alors

$$0 \in f(\tilde{u},\tilde{v},\tilde{x}) + \mathbf{J}_x(x - \tilde{x}) + \mathbf{J}_u((\text{dual } \mathbf{u}) - \tilde{u}) + \mathbf{J}_v(\mathbf{v} - \tilde{v}),$$

c'est à dire, $0 \in \mathbf{A}(x - \tilde{x}) - \mathbf{b}$, i.e., $x - \tilde{x} \in \Sigma(\mathbf{A},\mathbf{b})$ d'après le lemme 3.8. □

[3] Les notations $\Sigma(\mathbf{A},\mathbf{b})$ et $\Sigma(f,\mathbf{u},\mathbf{v})$ sont assez répandues, mais notons qu'elles ne sont pas homogènes : $\mathbf{b}$ est un vecteur d'intervalles généralisés tandis que $\mathbf{u}$ et $\mathbf{v}$ sont deux vecteurs d'intervalles propres représentant chacun un quantificateur différent.

Remarque 3.4 *La proposition précédente cache une subtilité. Supposons par exemple* $\mathbf{b}$ *de dimension 1, et propre.*

Comme $\mathbf{b} = \bigwedge_{u\in\mathbf{u}} \bigvee_{v,J_u,J_v} -f(\tilde{x},\tilde{u},\tilde{v}) - J_u(u-\tilde{u}) - J_v(v-\tilde{v})$*, on est tenté d'écrire*

$$\begin{array}{rcll} (x-\tilde{x}) \notin \Sigma(\mathbf{J}_x,\mathbf{b}) & \iff & \mathbf{J}_x(x-\tilde{x}) \cap \mathbf{b} = \emptyset & \textit{(d'après le lemme 2.2 p.30)} \\ & \iff & \big(\forall J_x \in \mathbf{J}_x\big)\ J_x(x-\tilde{x}) \notin \mathbf{b} & \\ & \iff & \big(\forall J_x \in \mathbf{J}_x\big)\big(\exists u \in \mathbf{u}\big)\big(\forall v \in \mathbf{v}\big)\big(\forall J_u \in \mathbf{J}_u\big)\big(\forall J_v \in \mathbf{J}_v\big) & \\ & & J_x(x-\tilde{x}) + f(\tilde{x},\tilde{u},\tilde{v}) + J_u(u-\tilde{u}) + J_v(v-\tilde{v}) \neq 0 & \end{array}$$

Curieusement, cette dernière relation n'implique pas $x \notin \Sigma(g,\mathbf{u},(\mathbf{v},\mathbf{J}_x,\mathbf{J}_u,\mathbf{J}_v))$*. En effet,*

$$x \notin \Sigma(g,\mathbf{u},(\mathbf{v},\mathbf{J}_x,\mathbf{J}_u,\mathbf{J}_v)) \iff \big(\exists u \in \mathbf{u}\big)\big(\forall J_x \in \mathbf{J}_x\big)\big(\forall v \in \mathbf{v}\big)\big(\forall J_u \in \mathbf{J}_u\big)\big(\forall J_v \in \mathbf{J}_v\big),$$

ce qui n'est pas une condition nécessaire de notre dernière relation. La proposition paraît fausse !

L'explication est la suivante : dans cette chaîne d'équivalences, on a perdu une information, l'optimalité de $\mathbf{J}_x(x-\tilde{x}) - \mathbf{b}$*, c'est à dire de son interprétation en terme de quantificateurs que l'on utilise en écrivant* $0 \notin \mathbf{J}_x(x-\tilde{x}) - \mathbf{b}$*, et que l'on perd en écrivant* $\mathbf{J}_x(x-\tilde{x}) \cap \mathbf{b} = \emptyset$ *(les intervalles* $\mathbf{J}_x$ *et* $\mathbf{b}$ *deviennent "indépendants").*

3.4.6 Bissections des paramètres

Nous venons de voir plusieurs moyens d'effectuer le pavage de l'espace de recherche pour un AE-système. Le test intérieur du §3.4.1 n'est par contre pas souvent applicable. Plaçons-nous alors dans le cas où le nombre de paramètres existentiels coïncide avec le nombre d'équations. Comme cela a été indiqué au chapitre 2, il peut être appliqué le test de boîte intérieure par permutation variable-paramètre (cf. §2.9.5 p.44), qui lui-même repose sur le test de Newton-Hansen paramétrique (cf. §2.9.3 p.42) :

$$\mathbf{v} \subseteq \tilde{v} + \Sigma(\mathbf{J}_v, -f(\tilde{x},\tilde{u},\tilde{v}) - \mathbf{J}_x(\mathbf{x}-\tilde{x}) - \mathbf{J}_u(\mathbf{u}-\tilde{u})) \ ?$$

Remarquons alors que ce test permet également un filtrage du paramètre $\mathbf{v}$, puisqu'il utilise le test de Newton-Hansen paramétrique qui peut servir immédiatement de filtrage. Il peut même se produire que $\mathbf{v}$ devienne vide, auquel cas la boîte $\mathbf{x}$, en plus de ne pas être intérieure, se trouve être infaisable. Ce test intérieur sert donc, par effet de bord, de filtrage (du paramètre v seulement) et de test d'infaisabilité.

Nous allons décrire dans cette section une nouvelle version de l'algorithme de bissection/évaluation, qui incorpore le filtrage et le test intérieur, et qui tient compte des paramètres. La partie bissection ne change pas. L'évaluation, par contre, est remplacée par un "traitement" de la boîte (filtrage et test intérieur) que nous détaillons.

Le plus important à noter est que le filtrage et le test intérieur, qui font donc aussi office tous les deux de test d'infaisabilité, utilisent des jacobiennes qui sont fonction de $\mathbf{x}$, mais également de $\mathbf{u}$ et $\mathbf{v}$. On peut observer facilement que l'efficacité de ces deux tests dépendant de la qualité de l'évaluation de ces jacobiennes. Il s'en suit qu'en raison du pessimisme des intervalles, il faut réduire $\mathbf{u}$ et $\mathbf{v}$ pour permettre une évaluation plus fine.

Rappelons que les domaines des paramètres $\mathbf{u}$ et $\mathbf{v}$ sont fixés une fois pour toute. Les réduire signifie qu'il faut intégrer les paramètres dans un algorithme de bissection de type *constructif*. La bissection des paramètres repose sur le principe très simple suivant. Si $\mathbf{u} = \mathbf{u}_1 \cup \mathbf{u}_2$ et $\mathbf{v} = \mathbf{v}_1 \cup \mathbf{v}_2$ alors par définition de $\Sigma(f,\mathbf{u},\mathbf{v})$,

$$x \in \Sigma(f,\mathbf{u},\mathbf{v}) \iff x \in \Sigma(f,\mathbf{u}_1,\mathbf{v}) \quad \textbf{et} \quad x \in \Sigma(f,\mathbf{u}_2,\mathbf{v}) \tag{3.24}$$

$$x \in \Sigma(f,\mathbf{u},\mathbf{v}_1) \quad \textbf{ou} \quad x \in \Sigma(f,\mathbf{u},\mathbf{v}_2) \Longrightarrow x \in \Sigma(f,\mathbf{u},\mathbf{v}) \tag{3.25}$$

On est donc, pour une boîte $\mathbf{x}$, amené à effectuer un arbre de recherche **et**/**ou** pour diminuer la taille des boîtes des paramètres jusqu'à ce que l'un des tests fonctionne. L'arbre peut être géré de la façon suivante par un

mécanisme de "descente-remontée". Cette gestion astucieuse est décrite dans [Gol06] pour une bissection limitée aux paramètres universels. Nous l'avons étendue aux paramètres existentiels, mais cette extension est bénigne à cause du sens de l'implication (3.25). En effet, la bissection des paramètres existentiels pour le test de boîte intérieure pose un réel paradoxe : d'une part, plus la boîte $\mathbf{v}$ est large, plus d'un point de vue ensembliste le test a des chances de réussir (ex : la relation $\big(\exists v \in \mathbf{v}\big)\ x = v$ avec $\mathbf{x} = [0,1]$ et $\mathbf{v} = [0,1]$ ne peut plus être satisfaite pour tout x dès que $\mathbf{v}$ est réduit) ; d'autre part, plus la boîte $\mathbf{v}$ est large, moins bon est le résultat de l'évaluation de sa jacobienne par intervalles, donc moins le test à de chances de réussir ! Cela signifie qu'en pratique, la bissection de $\mathbf{v}$ n'a de chance d'aboutir à un test que si la boîte $\mathbf{x}$ est petite par rapport à $\mathbf{v}$ (on découpe $\mathbf{v}$ suffisamment pour améliorer l'évaluation sans pour autant enlever de "support" de $\mathbf{x}$).

Venons-en à l'algorithme. Il n'est pas fourni de preuve, la validité de cet algorithme découle directement des deux formules (3.24) et (3.25). Tout nœud de l'arbre est la donnée d'un triplet $(\mathbf{x}, \mathbf{u}, \mathbf{v})$. Les nœuds qui ne sont pas des feuilles ont un opérateur **et**/**ou** associé. L'algorithme démarre par la phase descendante « **et** », sur l'arbre réduit à un seul nœud : celui des domaines $(\mathbf{x}, \mathbf{u}, \mathbf{v})$ initiaux.

Phase descendante « **et** »

Pour chaque nœud,
- Effectuer un test de boîte intérieure. S'il réussit, marquer le nœud comme *intérieur*. Si le test prouve que la boîte est infaisable, marquer le nœud comme *infaisable*.
- Filtrer le nœud (c'est à dire la boîte $\mathbf{x}$) en utilisant la proposition de Goldsztejn. Si le filtrage prouve que la boîte est infaisable, marquer le nœud comme *infaisable*. Sinon,
  - Si le diamètre de $\mathbf{u}$ est plus grand que ϵ_u, couper $\mathbf{u}$ en 2 sous-boîtes $\mathbf{u}_1$ et $\mathbf{u}_2$ puis poursuivre la descente « **et** » en créant un sous-arbre (**et**, ($\mathbf{u}_1$,$\mathbf{v}$,$\mathbf{x}$), ($\mathbf{u}_2$,$\mathbf{v}$,$\mathbf{x}$))).
  - Sinon, si le diamètre de $\mathbf{v}$ est plus grand que ϵ_v, couper $\mathbf{v}$ en 2 sous-boîtes $\mathbf{v}_1$ et $\mathbf{v}_2$ puis entamer la descente « **ou** » en créant un sous-arbre (**ou**, ($\mathbf{u}$,$\mathbf{v}_1$,$\mathbf{x}$), ($\mathbf{u}$,$\mathbf{v}_2$,$\mathbf{x}$))).

Phase descendante « **ou** »

Pour chaque nœud,
- Effectuer un test de boîte intérieure. S'il réussit, marquer le nœud comme *intérieur*. Sinon, si le diamètre de $\mathbf{v}$ est plus grand que ϵ_v, couper $\mathbf{v}$ en 2 sous-boîtes $\mathbf{v}_1$ et $\mathbf{v}_2$ puis poursuivre la descente en créant un sous-arbre (« **ou** », ($\mathbf{u}$,$\mathbf{v}_1$,$\mathbf{x}$), ($\mathbf{u}$,$\mathbf{v}_2$,$\mathbf{x}$))).

Phase ascendante

Dans un arbre « **et** » de racine $(\mathbf{x}, \mathbf{u}, \mathbf{v})$ et de nœuds-fils $(\mathbf{x}', \mathbf{u}', \mathbf{v}')$:
- Si un nœud est marquée comme étant infaisable, le sous-arbre est remplacé entièrement par un nœud marqué comme *infaisable*.
- Si tous les nœuds sont marqués comme intérieurs, $\mathbf{v}$ est remplacé par l'union des $\mathbf{v}'$ et le sous-arbre est remplacé entièrement par un nœud marqué comme *intérieur*.
- Sinon, remplacer $\mathbf{x}$ par l'intersection des $\mathbf{x}'$, remplacer $\mathbf{v}$ par l'union des $\mathbf{v}'$.

Dans un arbre « **ou** » de racine $(\mathbf{x}, \mathbf{u}, \mathbf{v})$ et de nœuds-fils $(\mathbf{x}', \mathbf{u}', \mathbf{v}')$:
- Si un nœud est marquée comme étant intérieur, le sous-arbre est remplacé entièrement par un nœud marqué comme *intérieur*. Remplacer $\mathbf{v}$ par l'union des $\mathbf{v}'$.

Une optimisation immédiate de cet algorithme consiste à faire partager (ex : par un pointeur commun) les boîtes $\mathbf{x}$ des nœuds de l'arbre. Tous ces pointeurs peuvent adresser le même objet : la boîte attachée à la racine de cet arbre. Cet algorithme est en cours d'amélioration (cf. conclusion).

3.5 Une introduction aux intervalles modaux

Cette section survole la théorie proposée par Goldsztejn [Gol05b, GDRT05, Gol07b, Gol07c], avec toutefois une coloration différente[4]. Notre but est d'en dégager les concepts de base, en occultant les résultats plus avancés[5] malgré leur importance.

Étant donné une fonction f réelle et deux vecteurs d'intervalles $\mathbf{x}$ et $\mathbf{y}$, nous avons cherché jusqu'ici à obtenir une boîte $\mathbf{z}$ qui satisfasse l'une des relations (3.1) ou (3.2) p. 49. Ces deux relations peuvent se réécrire aussi :

$$\big(\forall x \in \mathbf{x}\big)\big(Qz \in \mathbf{z}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x,y), \tag{3.26}$$

avec $Q = \exists$ dans (3.1) et $Q = \forall$ dans (3.2).

Les propositions 3.1 et 3.2 ont répondu (en partie) à cette question.

Le point de départ des intervalles modaux est le suivant : déterminer un moyen de calculer un intervalle $\mathbf{z}$ et un quantificateur Q qui satisfasse (3.26). La nuance fondamentale entre ce problème et celui traité auparavant est qu'il n'y a plus qu'une seule relation à satisfaire (la relation (3.26)), et que le quantificateur de $\mathbf{z}$ fait désormais partie du résultat. Le résultat cherché est donc un couple intervalle-quantificateur $(\mathbf{z}, Q)$, qu'on appelle *intervalle modal*.

Pour un intervalle dégénéré, les quantificateurs $\exists$ et $\forall$ sont équivalents. On convient alors d'associer le quantificateur $\exists$.

De même qu'auparavant, il est choisi de représenter l'intervalle ($\mathbf{z}$,$\exists$) par l'intervalle $\mathbf{z}$ lui-même (intervalle propre), et de représenter l'intervalle ($\mathbf{z}$,$\forall$) par l'intervalle impropre (dual $\mathbf{z}$). Mais ce choix n'est plus du tout arbitraire. Il se justifie par une interprétation pertinente de la relation d'inclusion (et par conséquence, des opérations de treillis « meet » et « join »).

3.5.1 Interprétation du treillis des intervalles généralisés

Étant donné qu'un intervalle modal est l'association d'un domaine $\mathbf{z}$ et d'un quantificateur Q, il est naturel de définir l'inclusion entre deux intervalles modaux ainsi :

Définition 3.5 (Inclusion entre intervalles modaux)
Soient $(\mathbf{z}, Q)$ et $(\mathbf{z}', Q')$ deux intervalles modaux. $(\mathbf{z}, Q)$ est inclus dans $(\mathbf{z}', Q')$ si pour toute relation ϕ définie sur les réels,

$$(Qz \in \mathbf{z})\ \phi(z) \Longrightarrow (Q'z \in \mathbf{z}')\ \phi(z)$$

Exemple 3.2 $([2,4], \forall) \subseteq ([3,5], \exists)$ *: en effet, pour toute relation $\phi(z)$ (par ex. $(z \geq 0)$, $(z^2 \leq 1)$, etc...), si ϕ est vérifiée pour* **toutes** *les valeurs entre 2 et 4, ϕ sera en particulier vérifiée pour la valeur 4. Donc il* **existe** *bien une valeur dans l'intervalle $[3,5]$ vérifiant ϕ.*

Cette définition signifie qu'un intervalle « plus petit » est « plus précis » (il vérifie une propriété plus forte). Cette définition subsume l'inclusion classique entre intervalles vu comme ensemble de réels. Il suffit d'identifier pour cela un intervalle classique $\mathbf{z}$ avec le couple $(\mathbf{z}, \exists)$. En effet, l'inclusion classique est la relation suivante :

$$\mathbf{z} \subseteq \mathbf{z}' \iff \big(\forall z \in \mathbf{z}\big) \quad z \in \mathbf{z}' \tag{3.27}$$

[4] L'association intervalle-quantificateur est notamment proscrite dans la formulation de Goldsztejn, alors que nous en faisons justement un point central ! mais cela n'a pas de grande conséquence.

[5] Entre autres : l'AE-extension naturelle des fonctions à valeurs vectorielles, l'AE-extension de la valeur moyenne, et les approximations intérieures de fonctions.

Montrons qu'elle est alors équivalente à notre définition 3.5. Soient $\mathbf{z}$ et $\mathbf{z}'$ tels que $\mathbf{z} \subseteq \mathbf{z}'$ au sens de la définition 3.5. Soit z_0 un point de $\mathbf{z}$, et ϕ la relation suivante : $\phi(z)$ ssi $z = z_0$. Clairement, il existe $z \in \mathbf{z}$ vérifiant $\phi(z)$, donc il existe $z \in \mathbf{z}'$ vérifiant $\phi(z)$, ce qui implique forcément $z_0 \in \mathbf{z}'$. On a bien $\mathbf{z} \subseteq \mathbf{z}'$ au sens de (3.27). La réciproque est évidente.

La définition 3.5, cas par cas, est équivalente aux quatre implications suivantes :

$(\mathbf{z}, \exists) \not\subseteq (\mathbf{z}', \forall)$	(les relations vraies au moins une fois sur $\mathbf{z}$ ne peuvent toutes l'être sur tout $\mathbf{z}'$)
$(\mathbf{z}, \exists) \subseteq (\mathbf{z}', \exists) \Longrightarrow \mathbf{z} \subseteq \mathbf{z}'$	(si toute relation vraie au moins une fois sur $\mathbf{z}$ est vraie au moins une fois sur $\mathbf{z}'$, $\mathbf{z}$ est inclus dans $\mathbf{z}'$)
$(\mathbf{z}, \forall) \subseteq (\mathbf{z}', \forall) \Longrightarrow \mathbf{z} \supseteq \mathbf{z}'$	(si toute relation vraie sur tout $\mathbf{z}$ est vraie sur tout $\mathbf{z}'$, $\mathbf{z}'$ est inclus dans $\mathbf{z}$)
$(\mathbf{z}, \forall) \subseteq (\mathbf{z}', \exists) \Longrightarrow \mathbf{z} \cap \mathbf{z}' \neq \emptyset$	(voir exemple 3.2)

Ces implications coïncident exactement avec les relations 3.6, 3.7 et 3.8 p.54, c'est à dire l'inclusion dans $\mathbb{KR}$, en choisissant de représenter $(\mathbf{z}, \forall)$ par un intervalle impropre. Le choix de représenter un intervalle modal par un intervalle généralisé est donc tout à fait justifié, et nous n'utiliserons désormais plus que cette notation.

Comme d'habitude, l'inclusion entre vecteurs d'intervalles est définie en appliquant la définition 3.5 composante par composante. De nouveau, cette définition coïncide avec l'inclusion entre vecteurs d'intervalles généralisés. Étant donné un ensemble muni d'une relation d'ordre $\subseteq$, les opérations « meet » et « join », notées resp. $\wedge$ et $\vee$, sont définies ainsi :

Définition 3.6 (Meet & Join pour un ensemble muni d'une inclusion)

$\mathbf{z}_1 \wedge \mathbf{z}_2$ *est le plus grand intervalle* $\mathbf{z}$ *tel que* $\mathbf{z} \subseteq \mathbf{z}_1$ *et* $\mathbf{z} \subseteq \mathbf{z}_2$.
$\mathbf{z}_1 \vee \mathbf{z}_2$ *est le plus petit intervalle* $\mathbf{z}$ *tel que* $\mathbf{z}_1 \subseteq \mathbf{z}$ *et* $\mathbf{z}_2 \subseteq \mathbf{z}$.

Il découle de l'interprétation de l'inclusion une interprétation des opérateurs $\wedge$ et $\vee$. Prenons par exemple deux intervalle propres $\mathbf{z}_1 := [1, 2]$ et $\mathbf{z}_2 := [4, 5]$. L'intervalle $\mathbf{z} := \mathbf{z}_1 \wedge \mathbf{z}_2$ est un intervalle tel que toute relation vraie sur $\mathbf{z}$ (soit une fois si $\mathbf{z}$ est propre, soit partout si $\mathbf{z}$ est impropre) soit vraie au moins une fois sur $\mathbf{z}_1$ et $\mathbf{z}_2$. Supposons $\mathbf{z}$ propre. Pour tout point $z_0 \in \mathbf{z}$, la relation $z = z_0$ est vérifié au moins une fois sur $\mathbf{z}$, donc elle doit l'être au moins une fois sur $\mathbf{z}_1$ et $\mathbf{z}_2$, ce qui est absurde puisque ces intervalles sont disjoints (au mieux z_0 appartient à l'un des deux). On a donc $\mathbf{z}$ impropre. Clairement, si une relation est vraie sur tout $\mathbf{z}$ il suffit que l'intervalle $\mathbf{z}$ ait un élément en commun avec les intervalles $\mathbf{z}_1$ et $\mathbf{z}_2$ pour que cette relation soit vraie une fois sur $\mathbf{z}_1$ et une fois sur $\mathbf{z}_2$. On voit alors que l'intervalle le plus grand (c'est à dire ayant le plus petit domaine, ou le moins « précis ») est $[4, 2]$.

Mathématiquement, la définition 3.6 se traduit dans $\mathbb{KR}$:

Définition 3.7 (Meet & Join dans $\mathbb{KR}$)

$$\mathbf{z}_1 \wedge \mathbf{z}_2 = [\sup\{\underline{\mathbf{x}}, \underline{\mathbf{y}}\}, \inf\{\overline{\mathbf{x}}, \overline{\mathbf{y}}\}]$$
$$\mathbf{z}_1 \vee \mathbf{z}_2 = [\inf\{\underline{\mathbf{x}}, \underline{\mathbf{y}}\}, \sup\{\overline{\mathbf{x}}, \overline{\mathbf{y}}\}]$$

Les définitions 3.6 et 3.7 se généralisent facilement à un ensemble quelconques d'intervalles. On montre alors [Gol05b] que pour toute fonction $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ et tout vecteur d'intervalles $\mathbf{x} \in \mathbb{IR}^p$ et $\mathbf{y} \in \mathbb{IR}^q$,

$$\bigwedge_{x \in \mathbf{x}} \bigvee_{y \in \mathbf{y}} f(x, y) = [\max_{x \in \mathbf{x}} \min_{y \in \mathbf{y}} f(x, y), \min_{x \in \mathbf{x}} \max_{y \in \mathbf{y}} f(x, y)],$$

$$\bigvee_{x \in \mathbf{x}} \bigwedge_{y \in \mathbf{y}} f(x, y) = [\min_{x \in \mathbf{x}} \max_{y \in \mathbf{y}} f(x, y), \max_{x \in \mathbf{x}} \min_{y \in \mathbf{y}} f(x, y)].$$

On retrouve donc précisément la définition donnée au §3.1.3 p.54.

Remarque : Comme $\mathbb{KR}$ est clos pour les opérations « meet » et « join », on dit qu'il s'agit d'un treillis.

3.5.2 AE-extension d'une relation réelle

Nous venons de voir que les intervalles généralisés (modaux) formaient un treillis dont les opérations s'interprètent par le biais de relations sur les réels. Étant donnés un intervalle généralisé $\mathbf{x}$, et une relation ϕ sur les réels, il est peu commode de dire que ϕ est satisfaite au moins une fois ou partout sur $\mathbf{x}$ suivant que $\mathbf{x}$ est propre ou impropre. On étend donc ϕ aux intervalles généralisés, et on se contente d'écrire directement $\phi(\mathbf{x})$. Le même principe se généralise aux relations sur des vecteurs de réels :

Définition 3.8 (AE-extension d'une relation réelle)
Soit ϕ une relation sur $\mathbb{R}^n$, L'AE-extension de ϕ est la relation ϕ sur $\mathbb{KR}^n$ telle que

$$\big(\forall \mathbf{x} \in \mathbb{KR}^n\big)\ \phi(\mathbf{x}) \iff \big(\forall x_{\mathrm{I}} \in (\text{dual } \mathbf{x}_{\mathrm{I}})\big)\big(\exists x_{\mathrm{P}} \in \mathbf{x}_{\mathrm{P}}\big)\ \phi(x)$$

où $\mathbf{x}_{\mathrm{P}}$ (resp. $\mathbf{x}_{\mathrm{I}}$) désigne les composantes propres (resp. impropres) de $\mathbf{x}$.

Exemple : Soit $\phi(x, y)$ ssi $x^2 \geq y$. Alors $\phi([2, 1], [0, 4])$ est vraie. Les AE-extensions des relations permettent de redéfinir encore plus simplement l'inclusion :

$$\mathbf{x} \subseteq \mathbf{x}' \text{ si pour toute relation } \phi, \quad \phi(\mathbf{x}) \Longrightarrow \phi(\mathbf{x}'). \tag{3.28}$$

3.5.3 AE-extension d'une fonction réelle

Dans les intervalles classiques, si $\mathbf{f}$ est une extension aux intervalles d'une fonction f, cela signifie (cf. §2.6 p.21) que pour toute boîte $\mathbf{x}$,

$$\big(\forall x \in \mathbf{x}\big)\big(\exists z \in \mathbf{f}(\mathbf{x})\big)\ \mid z = f(x),$$

autrement dit, que $\phi(\text{dual } \mathbf{x}, \mathbf{f}(\mathbf{x}))$, où $\phi(x, z)$ est la relation $z = f(x)$. Ceci mène à la définition suivante, qui est la pierre angulaire de la théorie [Gol05b].

Définition 3.9 (AE-extension d'une fonction)
Soit $f : \mathbb{R}^n \to \mathbb{R}^m$ continue. On appelle AE-extension de f, toute fonction $\mathbf{f} : \mathbb{KR}^n \to \mathbb{KR}^m$ vérifiant

$$\big(\forall \mathbf{x} \in \mathbb{KR}^n\big)\ \phi(\text{dual } \mathbf{x}, \mathbf{f}(\mathbf{x})),$$

où ϕ est la relation dans $\mathbb{R}^n \times \mathbb{R}^m$ suivante : $\phi(x, z)$ ssi $z = f(x)$.

Un vecteur $\mathbf{z}$ pour lequel $\phi(\text{dual } \mathbf{x}, \mathbf{z})$ est vrai est qualifié de « $(f, \mathbf{x})$-interprétable ». Si $\mathbf{x}$ est propre, un vecteur $\mathbf{z}$ $(f, \mathbf{x})$-interprétable n'est rien d'autre qu'une boîte contenant $range(f, \mathbf{x})$. Plus généralement, si $\mathbf{x}$ a des composantes impropres, $\mathbf{z}$ ne contient (est moins précis) qu'une image quantifiée (voir exemple suivant). On dit alors que $\mathbf{f}$ est une AE-extension *minimale* de f si pour tout $\mathbf{x} \in \mathbb{KR}^n$, $\mathbf{f}(\mathbf{x})$ est un vecteur d'intervalles généralisés le plus « précis » possible, c.a.d., si pour tout $\mathbf{z} \in \mathbb{KR}^m$ $(f, \mathbf{x})$-interprétable, on a $\mathbf{f}(\mathbf{x}) \subseteq \mathbf{z}$. Finalement, il est donné la proposition suivante [Gol05b] :

Proposition 3.10 (Caractérisation d'une AE-extension minimale unique)
Soit $f : \mathbb{R}^n \to \mathbb{R}^m$ continue et $\mathbf{f} : \mathbb{KR}^n \to \mathbb{KR}^m$. La fonction $\mathbf{f}$ est l'unique AE-extension minimale de f ssi

$$\big(\forall \mathbf{x} \in \mathbb{KR}^n\big)\big(\forall \mathbf{z} \in \mathbb{KR}^m\big) \quad \mathbf{z} \text{ est } (f, \mathbf{x})\text{-interprétable} \iff \mathbf{f}(\mathbf{x}) \subseteq \mathbf{z}.$$

Preuve. : Le schéma de la preuve est le suivant : le sens ($\Longleftarrow$) signifie que **f** est une AE-extension, le sens ($\Longrightarrow$) qu'elle est minimale et unique. □

Prouver l'existence d'une *unique* AE-extension optimale n'est pas un problème trivial. En général, il n'existe pas d'AE-extension minimale unique : cela est lié au fait qu'il n'existe pas une unique boîte intérieure non extensible (donc minimale) pour un ensemble, comme nous l'avons mentionné au tout début de cette thèse (cf. §2.2.4 p.12).

Exemple 3.3
Soit $f(x) = \begin{pmatrix} x_1 \\ |x_2 - x_1| \end{pmatrix}$, *et posons :*

$$\mathbf{x}_{\exists\exists} = ([0,1],[0,1]),$$
$$\mathbf{x}_{\forall\forall} = ([1,0],[1,0]),$$
$$\mathbf{x}_{\forall\exists} = ([1,0],[0,1]).$$

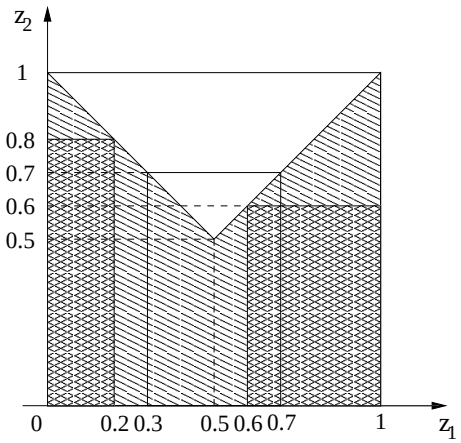


FIG. *3.3: AE-extensions minimales*

La zone hachurée en forme de "M" sur la figure 3.3 représente $range(f,[0,1]\times[0,1])$. *La boîte* $([0,1],[0,1])^T$ *est alors le plus petit vecteur* $(f,\mathbf{x}_{\exists\exists})$*-interprétable minimal. En effet, comme il est mentionné ci-dessus, un vecteur* $(f,\mathbf{x}_{\exists\exists})$*-interprétable est forcément quantifié* $\exists$, *et la plus petite boîte* $\mathbf{z}$ *vérifiant* $\big(\forall x \in \mathbf{x}_{\exists\exists}\big)\big(\exists z \in \mathbf{z}\big)\ z = f(x)$ *est précisément* $\square range(f,\mathbf{x}_{\exists\exists})$, *c'est à dire* $([0,1],[0,1])^T$. *De même, les boîtes* $([0,0.2],[0,0.8])^T$ *et* $([0.6,1],[0,0.6])^T$ *sont deux exemples de boîtes intérieures non extensibles. On en déduit que les vecteurs* $([0.2,0],[0.8,0])^T$ *et* $([1,0.6],[0.6,0])^T$ *sont deux exemples de vecteurs* $(f,\mathbf{x}_{\forall\forall})$*-interprétables minimaux puisqu'ils vérifient* $\big(\forall z \in (\text{dual } \mathbf{z})\big)\big(\exists x \in (\text{dual } \mathbf{x}_{\forall\forall})\big)\ z = f(x)$, *et qu'ils ne peuvent être plus "précis". Or ces deux boîtes ne sont pas comparables, il n'existe donc pas une unique AE-extension minimale de* f. *Enfin, notons que* $([0.7,0.3],[0,0.7])^T$ *est un exemple de vecteur* $(f,\mathbf{x}_{\forall\exists})$*-interprétable. En effet, nous avons bien*

$$\big(\forall z_1 \in [0.3,0.7]\big)\big(\forall x_2 \in [0,1]\big)\big(\exists x_1 \in [0,1]\big)\big(\exists z_2 \in [0,0.7]\big) \quad (z_1 = x_1)\ et\ (z_2 = |x_2 - x_1|).$$

De plus, ce vecteur est minimal. Si $\mathbf{z}_1$ *est "réduit" par exemple à* $[0.8,0.3]$ *alors en fixant* $z_1 = 0.8$ *et* $x_2 = 0$, *l'égalité* $x_1 = z_1$ *force* x_1 *à valoir* 0.8. *On a donc* $|x_2 - x_1| = 0.8$ *et cette valeur n'appartient pas à* $\mathbf{z}_2$. *La formule précédente ne peut plus être vérifiée. De même, si* $\mathbf{z}_2$ *est réduit à* $[0.1,0.7]$, *en choisissant* $z_1 = 0.3$, $x_2 = 0.3$ *on a* $x_1 = 0.3$ *donc* $z_2 = |x_2 - x_1| = 0$ *et cette valeur n'appartient pas à* $[0.1,0.7]$.

Toutefois, si f est à valeur dans $\mathbb{R}$, l'AE-extension minimale existe bien.

Proposition 3.11 (AE-extension minimale)
Soit $f : \mathbb{R}^n \to \mathbb{R}$ continue. Soit f^* la fonction telle que

$$\big(\forall \mathbf{x} \in \mathbb{KR}^n\big) \quad f^*(\mathbf{x}) := \bigvee_{x_{\mathrm{P}} \in \mathbf{x}_{\mathrm{P}}} \bigwedge_{x_{\mathrm{I}} \in (\mathrm{dual}\ \mathbf{x}_{\mathrm{I}})} f(x),$$

où $\mathbf{x}_{\mathrm{P}}$ (resp. $\mathbf{x}_{\mathrm{I}}$) désigne les composantes propres (resp. impropres) de $\mathbf{x}$.
f^* est une AE-extension de f minimale.

Preuve. Fixons $\mathbf{x} \in \mathbb{KR}^n$ et posons $\mathbf{z} = f^*(\mathbf{x})$.

Quitte à en réordonner les composantes, on peut écrire $\mathbf{x} = (\mathbf{x}_{\mathrm{P}}, \mathbf{x}_{\mathrm{I}})$. On peut donc remplacer en toute rigueur ce vecteur $\mathbf{x}$ de $\mathbb{KR}^n$ par un couple $(\mathbf{x}, \mathbf{y}) \in \mathbb{IR}^p \times \mathbb{IR}^q$ tel que $(\mathbf{x}, (\mathrm{dual}\ \mathbf{y})) = (\mathbf{x}_{\mathrm{P}}, \mathbf{x}_{\mathrm{I}})$. Nous effectuons cette transformation par souci de conformité avec les choix de notations précédents.

On a donc $(\mathbf{x}, \mathbf{y}) \in \mathbb{IR}^p \times \mathbb{IR}^q$ quelconque fixé, et $\mathbf{z} = f^*(\mathbf{x}, (\mathrm{dual}\ \mathbf{y}))$. Nous dirons qu'un intervalle est "interprétable" pour dire qu'il est "$(f, ((\mathrm{dual}\ \mathbf{x}), \mathbf{y}))$)-interprétable", c'est à dire qu'il vérifie $\phi((\mathrm{dual}\ \mathbf{x}), \mathbf{y}, \mathbf{z})$ avec $\phi(x, y, z) \iff z = f(x, y)$.

D'après la proposition 3.10, il suffit de montrer que $\mathbf{z}$ est interprétable et que pour tout $\mathbf{z}'$ interprétable, on a $\mathbf{z} \subseteq \mathbf{z}'$.

Considérons donc un intervalle $\mathbf{z}'$ interprétable quelconque.

Supposons $\mathbf{z} \in \mathbb{IR}$. On a $\mathbf{z} = f^{\exists}(\forall \mathbf{x}\ \exists \mathbf{y})$ d'après la proposition 3.2 p.52. D'où, en appliquant le lemme 3.3 p.55, $\mathbf{z} \neq \emptyset \Longrightarrow f^{\forall}(\forall \mathbf{x}\ \exists \mathbf{y}) = \emptyset$. Il n'existe donc pas de z' tel que $\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z' = f(x, y)$. Le quantificateur de $\mathbf{z}'$ est donc forcément $\exists$, i.e., cet intervalle est propre, et $\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\big(\exists z' \in \mathbf{z}'\big)\ z' = f(x, y)$. Mais par définition de $f^{\exists}(\forall \mathbf{x}\ \exists \mathbf{y})$, $\mathbf{z}$ est l'*unique* intervalle minimal vérifiant $\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\big(\exists z \in \mathbf{z}\big)\ z = f(x, y)$. Donc, d'une part $\mathbf{z} \subseteq \mathbf{z}'$ et d'autre part $\mathbf{z}$ est interprétable.

Supposons $\mathbf{z} \in \overline{\mathbb{IR}}$. D'après (3.9) p.55, $(\mathrm{dual}\ \mathbf{z}) = \mathrm{dual}\ \Big(\bigvee_{x \in \mathbf{x}} \bigwedge_{y \in \mathbf{y}} f(x, y)\Big) = \bigwedge_{x \in \mathbf{x}} \bigvee_{y \in \mathbf{y}} f(x, y)$. Grâce à la proposition 3.1 p.51, il vient $(\mathrm{dual}\ \mathbf{z}) = f^{\forall}(\forall \mathbf{x}\ \exists \mathbf{y})$, c'est à dire,

$$(\mathrm{dual}\ \mathbf{z}) = \{z \mid \big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x, y)\}, \tag{3.29}$$

ce qui implique $\big(\forall z \in (\mathrm{dual}\ \mathbf{z})\big)\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x, y)$, c'est à dire $\mathbf{z}$ est interprétable. Si $\mathbf{z}' \in \overline{\mathbb{IR}}$ alors $\big(\forall z' \in (\mathrm{dual}\ \mathbf{z}')\big)\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\ z = f(x, y)$, ce qui implique $\mathbf{z} \subseteq \mathbf{z}'$ d'après (3.29). Si $\mathbf{z}'$ est propre, alors $\big(\forall x \in \mathbf{x}\big)\big(\exists y \in \mathbf{y}\big)\big(\exists z' \in \mathbf{z}'\big)\ z' = f(x, y)$. On applique alors le lemme 3.4 p.55 qui donne $(\mathrm{dual}\ \mathbf{z}) \cap \mathbf{z}'$, c'est à dire $\mathbf{z} \subseteq \mathbf{z}'$. □

3.5.4 Extension naturelle (cas à valeurs réelles)

Sans surprise, l'évaluation des expressions avec l'arithmétique de Kaucher permet de construire une AE-extension de la même manière que l'arithmétique classique permettait de construire une extension aux intervalles :

Définition-Proposition 3.1 (AE-extension naturelle) *Soit $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ une fonction continue sans occurrence multiple de variables. La fonction*

$$\begin{array}{rccc} \mathbf{f} : & \mathbb{KD} & \to \mathbb{KR} \\ & \mathbf{x} & \mapsto f(\mathbf{x}) \end{array}$$

est une AE-extension de f, appelée **AE-extension naturelle**.

Preuve. Soit $\mathbf{x} \in \mathbb{KR}^n$. D'après le corollaire 3.1 p.62, $f^*(\mathbf{x}) \subseteq f(\mathbf{x})$ donc $f(\mathbf{x})$ est $(f, \mathbf{x})$-interprétable. □

3.6 Structure des intervalles généralisés

Terminons ce chapitre par une brève description de $\mathbb{KR}$, du point de vue algébrique. Ce point de vue nous sera utile au chapitre 5.

L'ensemble des intervalles $\mathbb{IR}$ est une structure algébrique "bancale", pour essentiellement deux raisons :

- Muni des opérations de borne supérieure et inférieure suivantes :
$$\mathbf{x} \wedge \mathbf{y} := \mathbf{x} \cap \mathbf{x} \quad \text{et} \quad \mathbf{x} \vee \mathbf{y} := \square(\mathbf{x} \cup \mathbf{y}),$$
c'est à dire des opérations d'intersection et d'enveloppe, $\mathbb{IR}$ ne forme pas un treillis. En effet, la borne inférieure de deux intervalles disjoints est l'ensemble vide, qui n'appartient pas à $\mathbb{IR}$. les intervalles généralisés permettent de remédier à ce problème, ce que nous avons vu à la section 3.5.
- $\mathbb{IR}$ n'est pas un groupe pour l'addition et la multiplication. En autorisant l'inversion des bornes, les intervalles généralisés forment un sur-ensemble des intervalles classiques où il devient possible de définir un symétrique de chaque élément pour l'addition, et un symétrique de chaque élément "sans 0" pour la multiplication. Nous allons maintenant détailler un peu plus cet aspect.

Proposition 3.12

$\mathbb{KR}$ est un groupe pour l'addition.

Preuve. L'associativité, la commutativité et l'existence d'un élément neutre s'obtiennent comme dans le cas classique. Il reste à montrer que tout intervalle $\mathbf{x}$ possède un opposé. Or $\mathbf{x} + [-\underline{\mathbf{x}}, -\overline{\mathbf{x}}] = [\underline{\mathbf{x}} - \underline{\mathbf{x}}, \overline{\mathbf{x}} - \overline{\mathbf{x}}] = [0, 0]$. □

On a de même que tout intervalle ne contenant pas $\mathbf{0}$ et non contenu dans $\mathbf{0}$ possède un inverse dans $\mathbb{KR}$ (pour la multiplication).

Ainsi l'opposé de $\mathbf{x}$, noté (opp $\mathbf{x}$), est l'intervalle $[-\underline{\mathbf{x}}, -\overline{\mathbf{x}}]$. Voici maintenant comment on peut interpréter cette addition. $[-1, -2]$ est l'opposé de $[1, 2]$, cela signifie qu'en ajoutant $[-1, -2]$, on soustrait "formellement" (opération notée $\ominus$) $[1, 2]$. Il ne faut pas confondre la soustraction dans $\mathbb{KR}$ avec cette soustraction formelle $\ominus$. Si on veut se raccrocher à la notion classique d'intervalles, il ne faut donc pas a priori interpréter ces opérations arithmétiques en terme de *calculs* mais en terme d'*équations*. Exemples :

- Quel est l'intervalle $\mathbf{x}$ tel que $[1, 2] + \mathbf{x} = [4, 7]$? → C'est $[4, 7] \ominus [1, 2] = [4, 7] + \text{opp}\,[1, 2] = [4, 7] + [-1, -2] = [3, 5]$, ce qu'on peut vérifier. Il faut bien noter que dans l'équation, $\mathbf{x}$ est écrit en gras : ce n'est donc pas une variable réelle pour laquelle on cherche à déterminer un ensemble-intervalle de solutions, mais une variable intervalle (ce qui est inhabituel).
- Que signifie l'égalité $\mathbf{z} = \mathbf{x} + \mathbf{y}$ lorsque : $\mathbf{x}$ est propre, $\mathbf{y}$ impropre, $\mathbf{z}$ est propre ?
→ Qu'en ajoutant (opp $\mathbf{y}$) à $\mathbf{z}$, on obtient $\mathbf{x}$ (ce qui a une signification puisqu'on coïncide de nouveau avec l'arithmétique classique).
- Lorsque $\mathbf{x}$ est propre, $\mathbf{y}$ impropre, $\mathbf{z}$ est impropre ?
→ Qu'en ajoutant (opp $\mathbf{z}$) à $\mathbf{x}$ on obtient (opp $\mathbf{y}$) (même remarque).

Si on veut interpréter le résultat d'une addition en se limitant à cette vision algébrique, il faut écrire une équation où chaque terme impropre est rendu propre par passage de l'autre côté du signe "=".

Remarquons que la multiplication externe est calquée sur le cas classique :

$$\lambda\mathbf{x} := \begin{cases} [\lambda\underline{\mathbf{x}}, \lambda\overline{\mathbf{x}}] \text{ si } \lambda \geq 0 \\ [\lambda\overline{\mathbf{x}}, \lambda\underline{\mathbf{x}}] \text{ sinon} \end{cases}$$

et que $\mathbb{KR}$ n'est pas un espace vectoriel. En effet, il n'y a toujours pas de distributivité de la multiplication externe sur l'addition scalaire, exemple :

$$[0, 0] = (1 + (-1))[0, 1] \neq 1 \times [0, 1] + (-1 \times [0, 1]) = [-1, 1].$$

3.7 Conclusion

Nous avons proposé dans ce chapitre une introduction aux intervalles modaux avec pour objectif de nous raccrocher à un minimum de concepts nouveaux. Nous avons choisi de baser la théorie sur un seul : celui d'*image quantifiée*. Une fois ce concept défini, les preuves n'utilisent dès lors que des notions connues (formules logiques du premier ordre, min-max, linéarisations, etc...). Cette présentation permet également de construire l'arithmétique des intervalles généralisés commme un moyen de calculer des images quantifiées, et non comme une extension algébrique (abstraite) de l'arithmétique classique.

Nous avons précisé dans quelle mesure cette théorie pouvait jouer un rôle dans la résolution d'AE-systèmes non linéaires. L'opérateur de Newton généralisé est l'outil le plus général issu de cette théorie, et le seul qui paraisse a priori systématisable. Pour le reste, le problème central est celui de *dépendance*. Dans le cas classique, nous avons vu que ce problème rendait les calculs plus mauvais mais pas faux. Ce n'est plus le cas lorsque les deux types de quantificateurs sont mélangés : il devient alors un handicap incontournable. C'est peut-être pour cette raison que les intervalles modaux sont restés à ce jour assez confidentiels.

Un des points clés mis en exergue pour la résolution d'AE-systèmes est la nécessité de bissecter les paramètres, c'est à dire de gérer une combinatoire multiple (variables, paramètres existentiels et universels). Ce besoin est apparu avec le filtrage de Newton généralisé, qui repose en effet sur un calcul dont l'efficacité dépend directement de la précision du domaine des paramètres. Il est fort à parier que n'importe quel filtrage ou test intérieur obéisse à cette même règle, et ce, indépendamment des intervalles modaux. Des travaux en cours corroborent cette thèse : il y est utilisé un test intérieur basé sur la théorie de Rohn [Roh02] qui pose la question de la bissection des paramètres exactement dans les mêmes termes. Cette problématique est donc une question en soi qui ouvre un espace de recherche intéressant ; nous y reviendrons dans la conclusion générale.

Chapitre 4

Systèmes d'équations linéaires par intervalles

Sommaire

Ce chapitre présente les principaux algorithmes permettant de calculer une approximation extérieure de l'ensemble des solutions d'un système linéaire par intervalle $\mathbf{A}x = \mathbf{b}$, où $\mathbf{A}$ est une matrice carrée et dont x est solution si

$$\big(\exists A \in \mathbf{A}\big)\big(\exists b \in \mathbf{b}\big)\ Ax = b.$$

Ces algorithmes sont ceux de Gauss-Seidel, de Krawczyk, de Hansen-Bliek et l'élimination de Gauss.

Le chapitre est organisé ainsi : les trois premières sections contiennent des résultats théoriques qui serviront à énoncer les conditions précises de convergence (ou d'applicabilité) des algorithmes susnommés. Les sections suivantes détaillent ensuite ces algorithmes et leurs propriétés.

En pratique, les algorithmes ne garantissent un bon comportement que si la matrice $\mathbf{A}$ est une *H-matrice* intervalle. La notion de H-matrice intervalle repose sur la notion de H-matrice réelle, elle-même reposant sur la notion de *M-matrice* réelle. Dans la section 4.1, nous définissons les M-matrices et H-matrices réelles, avec quelques autres outils numériques classiques utiles (qui ne font pas intervenir d'intervalle). Les M-matrices et H-matrices intervalles sont ensuite introduites au §4.2.

Les H-matrices sont en fait très particulières et ne se rencontrent pas "dans la nature". Il est souvent possible par contre de transformer une matrice intervalle régulière quelconque en une H-matrice intervalle, par un *préconditionnement*. Lorsque ce préconditionnement produit effectivement une H-matrice, on dit que la matrice de départ est *fortement régulière*. Ainsi, les algorithmes sont étroitement liés à la notion de *forte régularité*. Le préconditionnement et la forte régularité sont décrits au §4.3.

Ajoutons enfin que déterminer si une matrice intervalle est régulière ou non est un problème NP-difficile [RP93]

si bien qu'en pratique des conditions suffisantes de régularité vérifiables en temps polynomial s'avèrent intéressantes. La plus "faible" d'entre elles connue (donc la plus intéressante) est justement la propriété de forte régularité. Cette remarque rend d'autant plus légitime l'étude de cette classe de matrices.

4.1 Quelques outils d'analyse numérique

Nous allons introduire quelques notions d'analyse numérique classique utiles pour l'étude théorique des systèmes linéaires par intervalles. Notre but est d'établir des liens entre diverses caractéristiques des matrices comme la norme, le rayon spectral et l'inverse. Ces liens serviront à définir et à mieux cerner les types de matrices qui jouent un rôle particulier en analyse par intervalles : les M-matrices et H-matrices. Dans ce §4.1, tous les résultats ne portent que sur les réels.

Remarque 4.1 *Pour dire que A^{-1} existe (c.a.d.* $\det(A) \neq 0$*), nous utiliserons de façon équivalente les termes "A est régulière", "A est non-singulière" et "A est inversible".*

4.1.1 Norme pondérée

Soit u un vecteur de $\mathbb{R}^n$, strictement positif. On définit la norme *infinie* pondérée $\|.\|_u$ pour un vecteur x de $\mathbb{R}^n$ ainsi :

$$\|x\|_u := \max_{i=1..n} \frac{|x_i|}{u_i}.$$

Toute norme $\|.\|$ de $\mathbb{R}^n$ induit une norme matricielle dans $\mathbb{R}^{n\times n}$ (notée également $\|.\|$), dite *subordonnée*, qui possède quelques propriétés utiles comme le fait d'être une norme d'algèbre (voir plus bas). La norme subordonnée $A \to \|A\|$ est définie à partir de $x \to \|x\|$ ainsi :

$$\|A\| := \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|}{\|x\|}.$$

La proposition suivante donne l'expression de la norme subordonnée de $\|.\|_u$.

Proposition 4.1 *Soit $u \in \mathbb{R}^n$, $u > 0$. L'application $\|.\|_u : \mathbb{R}^{n\times n} \to \mathbb{R}$ telle que :*

$$\forall A \in \mathbb{R}^{n\times n}, \quad \|A\|_u = \max_{i=1..n} \sum_{j=1}^{n} \frac{|A_{ij}|u_j}{u_i}$$

est une norme matricielle, subordonnée de la norme vectorielle $\|.\|_u$.

Preuve. On a $\|A\|_u = \sup_{x\in\mathbb{R}^n\setminus\{0\}} \frac{\|Ax\|_u}{\|x\|_u}$. Or,

$$\forall x \in \mathbb{R}^n\setminus\{0\} \quad \|Ax\|_u = \max_{i=1..n} \frac{1}{u_i} |\sum_{j=1}^{n} A_{ij}x_j| \leq \max_{i=1..n} \frac{1}{u_i} \sum_{j=1}^{n} \frac{|A_{ij}u_j||x_j|}{u_j}.$$

$$\text{Donc } \|Ax\|_u \leq \max_{j=1..n} \frac{|x_j|}{u_j} \times \max_{i=1..n} \sum_{j=1}^{n} \frac{|A_{ij}u_j|}{u_i} = \|x\|_u \max_{i=1..n} \sum_{j=1}^{n} \frac{|A_{ij}u_j|}{u_i}.$$

Ainsi, $\|A\|_u \leq \max_{i=1..n} \sum_{j=1}^{n} \frac{|A_{ij}|u_j}{u_i}$. Notons maintenant i_0 la valeur de i en laquelle le maximum de cette somme est atteint. On obtient l'égalité dans la relation précédente en prenant le vecteur x défini ainsi :

$$x_j = \begin{cases} u_j & \text{si} \quad A_{i_0 j} \geq 0, \\ -u_j & \text{sinon.} \end{cases} \tag{4.1}$$

On a bien, avec un tel x, $\frac{\|Ax\|_u}{\|x\|_u} = \max_{i=1..n} \sum_{j=1}^{n} \frac{|A_{ij}|u_j}{u_i}$. $\square$

Il est possible de choisir u de telle sorte que $\|A\|_u$ soit arbitrairement grand. En effet, pour un u donné, notons encore i_0 (comme dans la preuve précédente) la valeur qui maximise la somme. Pour tout vecteur u' vérifiant $u'_{i_0} < u_{i_0}$ et $u'_j > u_j$ (pour $j \neq i_0$), on a $|A_{i_0 j}|u'_j/u'_{i_0} > |A_{i_0 j}|u_j/u_{i_0}$ et donc $\|A\|_{u'} > \|A\|_u$. Par contre, il existe une borne inférieure pour $\|A\|_u$, et c'est évidemment celle-ci qui nous intéresse. Par conséquent, une inégalité impliquant la norme pondérée d'une matrice A aura en général cette allure :

$$\big(\forall u > 0\big)\ \|A\|_u > \alpha \qquad \text{ou} \qquad \big(\exists u > 0\big)\ \|A\|_u < \alpha.$$

La dernière inéquation peut s'écrire sous une forme équivalente. On a en effet (en notant $|A|$ la matrice des valeurs absolues des coefficients de A) :

$$\|A\|_u < \alpha \iff |A|u < \alpha u. \tag{4.2}$$

Preuve.
$\|A\|_u < \alpha \iff \max_{i=1..n} \sum_{j=1}^{n} |A_{ij}|u_j/u_i < \alpha \iff \forall i \in [1..n]\ \sum_{j=1}^{n} |A_{ij}|u_j/u_i < \alpha$
$\iff \forall i \in [1..n]\ (|A|u)_i < \alpha u_i \iff |A|u < \alpha u.$ $\square$

Définition 4.1 $\|.\|$ *est une norme d'algèbre (ou norme sous-multiplicative) si*

$$\forall A \in \mathbb{R}^{n\times n},\ \forall B \in \mathbb{R}^{n\times n}, \|AB\| \leq \|A\|\|B\|.$$

Par une récurrence immédiate, on en déduit que si $\|.\|$ est une norme d'algèbre, alors,

$$\forall A \in \mathbb{R}^{n\times n},\ \forall k \in \mathbb{N},\ \ \|A^k\| \leq \|A\|^k. \tag{4.3}$$

On montre que toutes les normes subordonnées sont des normes d'algèbre. Or, nous ne manipulerons que des normes de type $\|.\|_u$ (voir définition 4.1). On supposera donc toujours par la suite que la norme considérée est bien une norme d'algèbre. Ceci permet de prouver notamment la convergence de la série suivante :

Proposition 4.2 (Neumann) *Soit $A \in \mathbb{R}^{n\times n}$ et $\|.\|$ une norme d'algèbre. Si $\|A\| < 1$ alors $I - A$ est inversible et la série $\sum_{k=0}^{\infty} A^k$ converge vers $(I - A)^{-1}$.*

Preuve. On a $\forall k > 0$, $\|A^k\| \leq \|A\|^k$ d'après (4.3). Or la série $\sum_{k=0}^{\infty} \|A\|^k$ converge si $\|A\| < 1$ donc $\sum_{k=0}^{\infty} A^k$ converge *normalement*. De plus, on a $\forall l \geq 0$, $(I - A)(\sum_{k=0}^{l} A^k) = I - A^{l+1}$ et cette dernière expression tend vers I quand $l \to \infty$ (car $\|A^{l+1}\| \leq \|A\|^{l+1}$). On en déduit que $(I - A)(\sum_{k=0}^{\infty} A^k) = I$. On montre de même que $(\sum_{k=0}^{\infty} A^k)(I - A) = I$, et donc que $\sum_{k=0}^{\infty} A^k = (I - A)^{-1}$. $\square$

La série de Neumann interviendra uniquement dans la preuve de la proposition 4.4 ci-après.

4.1.2 Rayon spectral

Dans tout ce §4.1.2, A désigne une matrice de $\mathbb{R}^{n\times n}$. Nous dirons que A est *positive* et noterons $A \geq 0$ lorsque tous ses coefficients sont positifs. Il ne s'agit pas de matrice *définie positive*.

Définition 4.2 (Rayon spectral) *Soit $Sp_{\mathbb{C}}(A)$ l'ensemble des valeurs propres complexes de A. On appelle rayon spectral de A et on note $\rho(A)$ le réel suivant :*

$$\rho(A) = \max_{\lambda \in Sp_{\mathbb{C}}(A)} |\lambda|.$$

Le rayon spectral est une caractéristique intrinsèque de chaque matrice qui est notamment indépendante de la norme choisie. A l'instar du déterminant et du rang, qui renseigne sur la régularité d'une matrice, le rayon spectral va servir à énoncer des conditions de convergence d'algorithmes tels que Gauss-Seidel ou Krawczyk. De plus, nous verrons plus loin que quelle que soit la norme $\|.\|$ choisie, on a $\rho(A) \leq \|A\|$. Ainsi, une condition de convergence telle que $\rho(A) < \alpha$ est plus faible que $\|A\| < \alpha$, donc plus intéressante, puisqu'elle englobe plus de cas.

Un théorème central est celui de Perron-Frobenius, qui "plonge" le rayon spectral d'une matrice positive dans une équation algébrique. En pratique, ce théorème permet de faire le pont entre une (in)égalité sur le rayon spectral et une (in)égalité qui implique la matrice elle-même (voir exemples ci-dessous).

Théorème 4.1 (Perron-Frobenius) *Soit A une matrice positive. Alors $\rho(A)$ est une valeur propre de A pour laquelle il existe un vecteur propre positif (non-nul) x de $\mathbb{R}^n$ (appelé* **vecteur de Perron***), i.e.,*

$$A \geq 0 \Longrightarrow \exists x \gneqq 0 \mid Ax = \rho(A)x.$$

Nous allons donner un lien entre rayon spectral et norme (proposition 4.3), et un lien entre rayon spectral et inverse (proposition 4.4). Ces liens reposent sur le lemme suivant :

Lemme 4.1 *Soit $\|.\|$ une norme de $\mathbb{R}^n$, et $\|.\|$ la norme matricielle subordonnée. Pour toute matrice A positive,*

$$\inf_{u>0} \|A\|_u = \rho(A).$$

Preuve. Voir [Neu90]. Remarquons qu'une partie du résultat : l'inégalité $\inf_{u>0} \|A\|_u \geq \rho(A)$ est vraie pour toute matrice (pas nécessairement positive) et se montre aisément. En effet, si λ est une valeur propre, par définition $\exists x \neq 0$ tel que $Ax = \lambda x$. Pour un tel x, $\|Ax\| = |\lambda|\|x\|$ et donc $|\lambda| = \|Ax\|/\|x\| \leq \|A\|$ (par définition de la norme subordonnée). Pour toute valeur propre λ de A et toute norme $\|.\|$, on a donc $|\lambda| \leq \|A\|$, d'où l'inégalité. Cette inégalité fournit déjà un minorant meilleur que 0 pour $\|A\|_u$. L'inégalité inverse, valable uniquement pour les matrices positives est plus difficile à obtenir ; elle repose sur le théorème de Perron-Frobenius.

Proposition 4.3 *Soit $A \geq 0$, $\alpha > 0$.*

$$\rho(A) < \alpha \Longleftrightarrow \exists u > 0 \mid Au < \alpha u.$$

Preuve. Avec le lemme 4.1, $\rho(A) < \alpha \iff \exists u > 0 \ \|A\|_u < \alpha$. On applique ensuite (4.2). □

Proposition 4.4 *Soit $A \in \mathbb{R}^{n \times n}$*

$$\rho(A) < 1 \Longrightarrow (I - A) \text{ inversible.}$$

Si de plus, $A \geq 0$ alors

$$\rho(A) < 1 \Longleftrightarrow (I - A) \text{ inversible et } (I - A)^{-1} \geq 0.$$

Preuve. Supposons que $\rho(A) < 1$. Pour montrer que $(I - A)$ est régulière, on montre que son noyau est réduit à $\{0\}$. Or, on a $(I - A)x = 0 \Longrightarrow Ax = x \Longrightarrow x = 0$, car 1 ne peut être valeur propre. Si de plus $A \geq 0$, alors avec le lemme 4.1, il existe $u > 0$ tq $\|A\|_u < 1$ donc la série de Neumann $\sum_{k=0}^{\infty} A^k$ converge (Proposition 4.2) vers $(I - A)^{-1}$ et comme chaque terme est positif, sa limite l'est également. Réciproquement (voir [NK97]), d'après le théorème de Perron-Frobenius, $\exists x \gneqq 0$ tel que $Ax = \rho(A)x$. On a $(I - A)x = (1 - \rho(A))x$ donc $x = (1 - \rho(A))(I - A)^{-1}x$. Si $(I - A)^{-1} \geq 0$ et $x \gneqq 0$ alors nécessairement $(1 - \rho(A)) > 0$, i.e., $\rho(A) < 1$.□

Enfin, nous aurons également recours à l'inégalité suivante :

Proposition 4.5 *Soient $A \geq 0$, $B \geq 0$.* $\quad A \geq B \Longrightarrow \rho(A) \geq \rho(B)$.

Preuve. Voir [Neu90].

4.1.3 M-matrices

Notre objectif est de caractériser les matrices "presque diagonales", c'est à dire les matrices où le "poids" est sur la diagonale. On parle encore de *dominance diagonale*. Ces matrices ont l'intérêt de posséder des propriétés proches de celles des matrices diagonales. Concrètement, A est une matrice à **diagonale dominante** si :

$$\forall i \in [1..n] \quad |a_{ii}| > \sum_{j \neq i} |a_{ij}| \tag{4.4}$$

Si on est en présence d'une matrice A vérifiant "l'équilibre des signes", c'est à dire les conditions (i) et (ii) suivantes :

$$(i) \quad \forall i \in [1..n] \quad a_{ii} \geq 0 \qquad \text{et} \qquad (ii) \quad \forall j \neq i,\ a_{ij} \leq 0, \tag{4.5}$$

alors la condition de dominance diagonale (4.4) s'écrit plus simplement :

$$\forall i \in [1..n], \quad \sum_{j=1}^{n} a_{ij} > 0. \tag{4.6}$$

On a donc déjà identifié une première classe de matrices "presque diagonales" : celles qui présentent l'équilibre des signes et dont la somme des éléments de chaque ligne est strictement positive, puisque (4.5)+(4.6) $\Longrightarrow$ (4.4). On voit alors facilement que la positivité des éléments diagonaux devient une condition superflue et qu'elle se déduit directement des deux autres conditions. Autrement dit, (4.5).(ii)+(4.6) $\Longrightarrow$ (4.4).

Nous allons généraliser la classe de matrices vérifiant (4.5).(ii)+(4.6) en assouplissant (4.6) pour obtenir celle des M-matrices, dont la définition la plus parlante est la suivante :

Définition 4.3 (M-matrice) *Soit $A \in \mathbb{R}^{n \times n}$*

$$A \text{ est une M-matrice si } \begin{cases} \forall i \neq j \ a_{ij} \leq 0, \\ \exists u > 0 \in \mathbb{R}^n \mid Au > 0. \end{cases}$$

La condition $\exists u > 0 \in \mathbb{R}^n \mid Au > 0$ peut s'interpréter ainsi : si on considère la matrice $U = diag(u)$, alors la matrice AU vérifie (4.6), puisque $\forall i, \sum_{j=1}^{n} (AU)_{ij} = \sum_{j=1}^{n} a_{ij}u_j = (Au)_i > 0$. Or la multiplication par une matrice diagonale strictement positive n'est qu'un changement d'échelle. Une M-matrice n'est donc autre qu'une matrice vérifiant (4.5).(ii) et (4.6) modulo un changement d'échelle.

La propriété fondamentale des M-matrices est qu'elles sont d'inverse positive :

Proposition 4.6 *Soit A une matrice ayant des éléments non-diagonaux négatifs. On a l'équivalence suivante :*

$$A \text{ est une M-matrice } \iff A^{-1} \geq 0.$$

La proposition suivante nous sera également d'une grande utilité :

Proposition 4.7 *Soient $A \in \mathbb{R}^{n\times n}$ et $B \in \mathbb{R}^{n\times n}$. Si A est une M-matrice et $B \geq 0$ alors*

$$A - B \text{ est une M-matrice } \iff \rho(A^{-1}B) < 1.$$

Preuve.
Remarquons pour commencer que A étant une M-matrice, $A^{-1} \geq 0$, d'après la proposition 4.6. Comme B est positive, on a donc $A^{-1}B \geq 0$.

($\Longrightarrow$) $(A-B)^{-1}(A-B) \geq 0$ implique $(A-B)^{-1}A \geq (A-B)^{-1}B$. Comme $A-B$ est une M-matrice, alors $(A-B)^{-1} \geq 0$, d'après la proposition 4.6. Comme $B \geq 0$, on en déduit $(A-B)^{-1}B \geq 0$, d'où $(A-B)^{-1}A \geq 0$. Ceci se réécrit $(A-B)^{-1}(A^{-1})^{-1} \geq 0$, ou encore $(A^{-1}(A-B))^{-1} \geq 0$, c'est à dire $(I - A^{-1}B)^{-1} \geq 0$. Il s'ensuit que $I - A^{-1}B$ est inversible et d'inverse positif. Comme $A^{-1}B \geq 0$, cela implique bien d'après la proposition 4.4 que $\rho(A^{-1}B) < 1$.

($\Longleftarrow$) $\rho(A^{-1}B) < 1$ et $A^{-1}B \geq 0$ impliquent $I - A^{-1}B$ inversible et d'inverse positive, en vertu de la proposition 4.4. Comme $A^{-1} \geq 0$ alors $(I - A^{-1}B)^{-1}A^{-1} \geq 0$ donc $(A(I - A^{-1}B))^{-1} \geq 0$, c'est à dire $(A-B)^{-1} \geq 0$. Comme $A - B$ a ses éléments non diagonaux négatifs (puisque $B \geq 0$), la proposition 4.6 nous permet de conclure que $A - B$ est une M-matrice.

□

Cette proposition montre sous quelles conditions en modifiant "légèrement" A, on conserve la propriété de M-matrice. En effet, il faut remarquer que $\rho(A^{-1}B) < 1 \Longrightarrow I - A^{-1}B \sim I \Longrightarrow A - B \sim A$. Donc modifier "légèrement" A peut s'écrire $\rho(A^{-1}B) < 1$.

Cette proposition ne peut pas être étendue telle quelle au cas intervalle. Quand bien même elle le serait, elle nécessiterait le calcul de l'inverse d'une matrice intervalle, qui est difficile. On verra plus loin comment néanmoins se servir de cette proposition, dans le cas par exemple de matrices intervalles $\mathbf{A}$ pour lesquelles on souhaite avoir $\underline{\mathbf{A}}^{-1} \geq 0$. Si $\mathbf{A}$ est centrée sur l'identité (ce qui deviendra courant grâce au préconditionnement), on aura alors $\underline{\mathbf{A}} = I - \text{rad } \mathbf{A}$, et comme I est évidemment une M-matrice et rad $\mathbf{A} \geq 0$, on se basera sur la relation suivante :

$$\underline{\mathbf{A}} \text{ est une M-matrice ssi } \rho(\text{rad } \mathbf{A}) < 1.$$

4.1.4 H-matrices

Les M-matrices sont des objets mathématiquement bien "identifiés" (plus de 50 manières de les caractériser ont été répertoriées !), auxquels on aimerait se ramener. On a pu généraliser la dominance diagonale à la notion de M-matrice, mais en ajoutant la contrainte d'équilibre des signes. L'idée derrière les H-matrices est de s'affranchir de cette contrainte en se ramenant à une M-matrice via la matrice dite *de comparaison* (ou d'Ostrowski).

Définition 4.4 (Matrice de comparaison)
Soit $A \in \mathbb{R}^{n\times n}$. On appelle **matrice de comparaison** *et on note $<A>$ la matrice définie ainsi :*

$$<A>_{ij} := \begin{cases} |A_{ij}| & si \;\; i = j, \\ -|A_{ij}| & sinon\ . \end{cases}$$

Définition 4.5 (H-matrice)
Soit $A \in \mathbb{R}^{n\times n}$.

$$A \text{ est une } \textbf{H-matrice } si \;\; <A> \;\; \text{est une M-matrice.}$$

Comme par définition les éléments non-diagonaux de $<A>$ sont négatifs, il suffit donc que $<A>$ soit inversible et d'inverse positive, ou que $\exists u > 0$ tel que $<A>\ u > 0$, pour que A soit une H-matrice.

Les matrices à diagonale dominante sont donc un cas particulier de H-matrices : en effet, une matrice vérifiant (4.4) vérifie bien $<A>\ u > 0$ avec $u = (1, \ldots, 1)^T$. A notre niveau, l'intérêt principal de la notion de H-matrice est qu'elle englobe et généralise la notion de matrice à diagonale dominante, ce qui n'était pas le cas des M-matrices. On peut donc imaginer "en pratique" qu'une H-matrice est une matrice à diagonale dominante. Cette remarque s'appliquera également pour les H-matrices intervalles et les matrices intervalles à diagonale dominante.

Proposition 4.8 *Toute matrice triangulaire à diagonale non-nulle est une H-matrice.*

Preuve.
Soit T une matrice triangulaire qu'on suppose d'abord inférieure. Il suffit de montrer que $<T>$ est une M-matrice. On construit pour cela un vecteur $u > 0$ tel que $<T>\ u > 0$. Pour ce faire, il suffit de prendre $u_1 = 1$, puis u_2 tel que $<T>_{21}\ u_1 + <T>_{22}\ u_2 > 0$, et ainsi de suite de telle sorte que $<T>\ u > 0$. On pose donc $u_i > -(\sum_{j<i} <T>_{ij}\ u_j)/\ <T>_{ii}$, et par une récurrence immédiate on a $u_i > 0$. Si T est triangulaire supérieure, il suffit de procéder dans le sens contraire, c'est à dire de construire u en partant de u_n jusqu'à u_1. □

4.2 H-matrices intervalles

Définition 4.6 (Matrice de comparaison -cas intervalle-)
On appelle matrice de comparaison de $\mathbf{A}$ *que l'on note* $<\mathbf{A}>$ *la matrice réelle :*

$$<\mathbf{A}>_{ij} := \begin{cases} <\mathbf{A}_{ii}> & \text{si } i = j, \\ -|\mathbf{A}_{ij}| & \text{sinon .} \end{cases}$$

Rappelons que $|\mathbf{x}|$ *désigne la plus grande valeur absolue de* $\mathbf{x}$*, et* $<\mathbf{x}>$ *la plus petite petite (cf. p.11).*

Définition 4.7 (H-matrice intervalle) *Soit* $\mathbf{A} \in \mathbb{IR}^{n\times n}$.
$\mathbf{A}$ *est une H-matrice si* $<\mathbf{A}>$ *est une M-matrice (réelle)*

Comme par définition les éléments non-diagonaux de $<\mathbf{A}>$ sont négatifs, il suffit donc que $<\mathbf{A}>^{-1} \geq 0$.

Remarque 4.2 *Il est utile de remarquer que la notion de H-matrice intervalle se ramène à celle de M-matrice réelle et non à celle de M-matrice intervalle. Car il existe bien une extension de la notion de M-matrice aux intervalles, mais elle n'a pas d'utilité dans le contexte de cette thèse :*

$$\mathbf{A} \in \mathbb{IR}^{n\times n} \text{ est une M-matrice (intervalle) si } \begin{cases} \forall i \in [1..n]\ \mathbf{a}_{ij} \leq 0, \\ \exists u > 0 \in \mathbb{R}^n \mid \mathbf{A}u > 0. \end{cases}$$

Il existe alors une caractérisation intéressante :

$$\mathbf{A} \text{ est une M-matrice } \Longleftrightarrow \underline{\mathbf{A}} \text{ et } \overline{\mathbf{A}} \text{ sont des M-matrices.}$$

Kuttler [Neu90] montre alors qu'une M-matrice $\mathbf{A}$ *vérifie* $\mathbf{A}^{-1} \geq 0$*, où* $\mathbf{A}^{-1} = \square\{A^{-1}, A \in \mathbf{A}\}$*, ce qui n'est pas une conséquence triviale de la proposition 4.6 (puisque* $\mathbf{A}^{-1} \neq \{A^{-1}, A \in \mathbf{A}\}$*). Kuttler montre plus précisément que pour toute matrice* $\mathbf{A}$ *vérifiant uniquement* $\overline{\mathbf{A}}^{-1} \geq 0$ *et* $\underline{\mathbf{A}}^{-1} \geq 0$ *(donc* $\mathbf{A}$ *n'est pas tout à fait une M-matrice d'après ce qui est dit au dessus), on a* $\mathbf{A}^{-1} = [\overline{\mathbf{A}}^{-1}, \underline{\mathbf{A}}^{-1}]$*. Enfin, concernant les matrices intervalles d'inverse positive, Beeck [Neu90] donne une formule pour* $\square\,\Sigma(\mathbf{A}, \mathbf{b})$ *lorsque* $\mathbf{A}^{-1} \geq 0$*. Malheureusement, cette formule est constructive uniquement dans l'un des cas suivant :* $\mathbf{b} \geq 0$*,* $\mathbf{b} \leq 0$*,* $0 \in \mathbf{b}$*. Nous verrons par contre au §4.7 une formule dans le cas où* $\underline{\mathbf{A}}^{-1} \geq 0$*, si* $\mathbf{A}$ *est centrée sur* I*.*

La définition 4.7 est cohérente dans la mesure où une H-matrice intervalle représente un ensemble de H-matrice réelles :

Proposition 4.9

$\mathbf{A}$ *est une H-matrice ssi* $\forall A \in \mathbf{A}$, A *est une H-matrice.*

Preuve.

($\Longrightarrow$) De façon évidente on a $\mathbf{B} \subseteq \mathbf{A} \Longrightarrow <\mathbf{B}> \geq <\mathbf{A}>$. Or $\mathbf{A}$ est une H-matrice $\Longrightarrow <\mathbf{A}>$ est une M-matrice $\Longrightarrow \exists u > 0 \mid <\mathbf{A}> u > 0 \Longrightarrow \exists u > 0 \mid <\mathbf{B}> u > 0 \Longrightarrow <\mathbf{B}>$ est une M-matrice $\Longrightarrow \mathbf{B}$ est une H-matrice. Cela vaut en particulier pour les matrices $\mathbf{B}$ réelles.

($\Longleftarrow$) Il suffit de considérer en particulier la matrice $A \in \mathbf{A}$ définie ainsi :

$$A_{ii} := \begin{cases} \overline{\mathbf{A}_{ii}} \text{ si } \overline{\mathbf{A}_{ii}} < 0 \\ \underline{\mathbf{A}_{ii}} \text{ si } \underline{\mathbf{A}_{ii}} > 0 \\ 0 \text{ sinon} \end{cases} \quad \text{et pour } k \neq i \quad A_{ik} := \begin{cases} \overline{\mathbf{A}_{ik}} \text{ si } |\overline{\mathbf{A}_{ik}}| > |\underline{\mathbf{A}_{ik}}| \\ \underline{\mathbf{A}_{ik}} \text{ sinon} \end{cases}$$

On voit alors que $<A>=<\mathbf{A}>$ et donc $\mathbf{A}$ est une H-matrice.

□

On peut prouver facilement que, quelle que soit $\mathbf{A}$, on a $<\mathbf{A}> \geq <\text{mid } \mathbf{A}> - \text{rad } \mathbf{A}$. Si de plus, $\forall i \in [1..n]$ $0 \notin \mathbf{A}_{ii}$ alors $<\mathbf{A}>=<\text{mid } \mathbf{A}> - \text{rad } \mathbf{A}$. Cette dernière formule s'applique dans le cas des H-matrices (car $0 \in \mathbf{A}_{ii} \Longrightarrow <\mathbf{A}_{ii}>= 0 \Longrightarrow <\mathbf{A}>_{ii}= 0$ et $<\mathbf{A}>$ ne peut pas être une M-matrice).

Or, on sait transmettre, via la proposition 4.7, la propriété de M-matrice d'une matrice réelle à une autre pour peu qu'elles soient suffisamment "proches". Ainsi, si (rad $\mathbf{A}$) n'est pas trop grand, $<\mathbf{A}>\sim<\text{mid } \mathbf{A}>$ et on en déduit avec la proposition 4.7 que si (mid $\mathbf{A}$) est une H-matrice, $\mathbf{A}$ est une H-matrice. Formellement, on énonce une nouvelle caractérisation des H-matrices intervalles :

Proposition 4.10

$\mathbf{A}$ *est une H-matrice* $\Longleftrightarrow$ (mid $\mathbf{A}$) *est une H-matrice et* $\rho(<\text{mid } \mathbf{A}>^{-1} \text{ rad } \mathbf{A}) < 1$.

Preuve. D'après les définitions 4.7 et 4.5, il s'agit de montrer :

$<\mathbf{A}>$ est une M-matrice $\Longleftrightarrow <\text{mid } \mathbf{A}>$ est une M-matrice et $\rho(<\text{mid } \mathbf{A}>^{-1} \text{ rad } \mathbf{A}) < 1$,

ou encore, en utilisant la proposition 4.7 :

$<\mathbf{A}>$ est une M-matrice $\Longleftrightarrow <\text{mid } \mathbf{A}> - \text{rad } \mathbf{A}$ est une M-matrice.

($\Longrightarrow$) Si $<\mathbf{A}>$ est une M-matrice, alors $\forall i \in [1..n]$, $0 \notin \mathbf{A}_{ii}$. On a alors $<\mathbf{A}>=<\text{mid } \mathbf{A}> - \text{rad } \mathbf{A}$, et donc $<\text{mid } \mathbf{A}> - \text{rad } \mathbf{A}$ est une M-matrice.

($\Longleftarrow$) Si $<\text{mid } \mathbf{A}> - \text{rad } \mathbf{A}$ est une M-matrice, $\exists u > 0$ tel que $(<\text{mid } \mathbf{A}> - \text{rad } \mathbf{A})u > 0$ et comme $<\mathbf{A}> \geq (<\text{mid } \mathbf{A}> - \text{rad } \mathbf{A})$ alors $<\mathbf{A}> u > 0$ et $<\mathbf{A}>$ est une M-matrice.

□

Terminons ce paragraphe en indiquant que l'inverse d'une H-matrice intervalle peut être "facilement" bornée (Ostrowski, cf. [Neu90]).

Proposition 4.11 (Ostrowski)

$\mathbf{A}$ *est une H-matrice* $\Longrightarrow |\mathbf{A}^{-1}| \leq <\mathbf{A}>^{-1}$ (*rappelons que* $\mathbf{A}^{-1} = \square\{A^{-1}, A \in \mathbf{A}\}$).

Cette formule permet déjà de donner une première manière d'approximer extérieurement $\Sigma(\mathbf{A}, \mathbf{b})$ lorsque $\mathbf{A}$ est régulière, puisque $x \in \Sigma(\mathbf{A}, \mathbf{b}) \Longrightarrow x \in \mathbf{A}^{-1}\mathbf{b} \Longrightarrow |x| \leq |\mathbf{A}^{-1}||\mathbf{b}| \Longrightarrow |x| \leq <\mathbf{A}>^{-1} |\mathbf{b}|$. Ainsi $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq [-1, 1] <\mathbf{A}>^{-1} |\mathbf{b}|$. Quoique le résultat soit grossier, nous y aurons recours.

4.3 Préconditionnement

Le préconditionnement est une technique qui consiste à transformer un système linéaire intervalle $\mathbf{A}x = \mathbf{b}$ en multipliant (à gauche) $\mathbf{A}$ et $\mathbf{b}$ par une même matrice scalaire C de telle sorte que le système $(C\mathbf{A})x = (C\mathbf{b})$ se "comporte mieux" numériquement, tout en préservant l'ensemble solution du système original. Le préconditionnement est lié à la notion de *condition* bien connue des numériciens :

Exemple 4.1 (Hilbert) *La matrice de Hilbert est un exemple classique de matrice dite "mal conditionnée". Elle est définie ainsi :* $A := (|a_{ij}|)_{1\leq i\leq n, 1\leq j\leq n}$ *où* $a_{ij} = 1/(i+j-1)$. *Dans le cas* 3×3 *cela donne (avec une précision de* 10^{-4}*) :*

$$A := \begin{pmatrix} 1 & 0.5 & 0.3333 \\ 0.5 & 0.3333 & 0.25 \\ 0.3333 & 0.25 & 0.2 \end{pmatrix}$$

Considérons deux vecteurs proches, par exemple :

$$b = (1,1,1)^T \quad et \quad b' = (1.1,1,1)^T.$$

Si on résout par une méthode classique $Ax = b$ *et* $Ax' = b'$ *on trouve :*

$$x = (3.04, -24.21, 30.19) \qquad x' = (3.95, -27.84, 33.22)$$

On s'aperçoit que l'incertitude de 10^{-1} sur b a produit une incertitude sur le résultat 10 à 30 fois plus grande, c'est ce qu'on traduit en qualifiant cette matrice de *mal conditionnée.*

Dans les algorithmes par intervalles, le mauvais conditionnement se traduit par un accroissement très rapide de la taille des intervalles produits par les calculs, ce qui empêche d'obtenir des résultats précis avec ce genre de systèmes.

Exemple 4.2 *Reprenons l'exemple 4.1 de Hilbert, la résolution de* $Ax = \mathbf{b}$ *par intervalles (où* $\mathbf{b} = [1, 1.1] \times [1,1] \times [1,1]$*) avec une élimination de Gauss (cf. §4.6) produit la boîte* $[-3.01, 10.00] \times [-33.89, -18.16] \times [24.15, 39.27]$*, où le rayon du premier intervalle est plus de 100 fois supérieur à l'incertitude introduite sur b !*

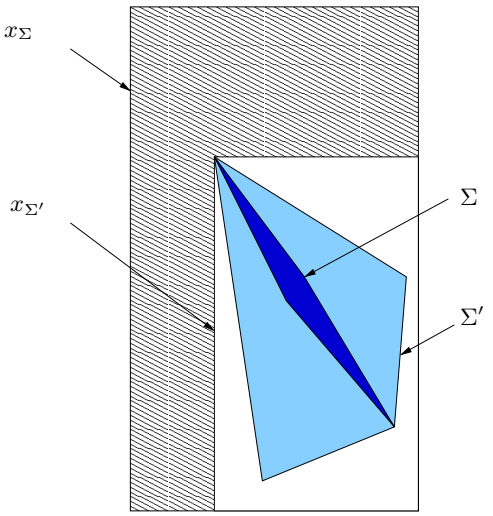
Exemple 4.3 *Si on place maintenant une incertitude de* 10^{-2} *cette fois sur un coefficient de A, et que l'on résout le système* $\mathbf{A}x = b$ *(avec* $\mathbf{A}_{11} = [1, 1.01]$*, et* $b = (1,1,1)$*), on produit la boîte* $[-21.99, 35.57] \times [-80.33, -10.43] \times [16.79, 84.61]$*, qui est toujours aussi inexploitable.*

Pour remédier à cela, l'idée est d'appliquer un *préconditionnement* qui consiste à prémultiplier $\mathbf{A}$ et $\mathbf{b}$ par une matrice scalaire bien choisie, en général l'inverse du milieu de $\mathbf{A}$. Inverser une matrice scalaire pour résoudre un système est raisonnable dans le contexte de l'analyse par intervalles pour des raisons d'échelle : les systèmes linéaires réels résolus aujourd'hui peuvent présenter des millions d'inconnues, ce qui exclut tout calcul d'inverse. Les méthodes multi-grilles, par exemple, permettent de traiter de tels systèmes. La contrepartie étant qu'elles ne présentent que des solutions approximatives. A l'inverse, les problèmes intervalles traités à l'heure actuelle sont, en comparaison, d'une dimension très réduite, l'intérêt étant plus porté sur la qualité des résultats.

Cumulons les incertitudes sur A et b des exemples 4.2 et 4.3. Si on considère $C := (\text{mid } \mathbf{A})^{-1}$, on a alors :

$$C\mathbf{A} = \begin{pmatrix} [0.957, 1.043] & [0,0] & [0,0] \\ [-0.174, 0.174] & [1,1] & [0,0] \\ [-0.145, 0.145] & [0,0] & [1,1] \end{pmatrix} \quad \text{et} \quad C\mathbf{b} = \begin{pmatrix} [2.909, 3.776] \\ [-24.156, -23.681] \\ [29.753, 32.652]) \end{pmatrix}$$

La résolution de $(C\mathbf{A})x = (C\mathbf{b})$ par la même méthode produit la boîte $[2.79, 3.95] \times [-27.84, -23.00] \times [29.18, 33.22]$, qui a un rayon nettement plus acceptable. Pourtant, nous verrons plus loin que le préconditionnement agrandit (au mieux conserve) l'ensemble solution, i.e., $\Sigma(C\mathbf{A}, C\mathbf{b}) \supseteq \Sigma(\mathbf{A}, \mathbf{b})$. Il peut paraître étonnant alors qu'un même algorithme de résolution produise un boîte plus fine pour approximer un ensemble plus gros. Le dessin suivant illustre ce phénomène sur un exemple similaire à 2 dimensions.



Légende

Σ	$\Sigma(\mathbf{A}, \mathbf{b})$
Σ'	$\Sigma(C\mathbf{A}, C\mathbf{b})$ avec $C = (\text{mid } \mathbf{A})^{-1}$
x_Σ	approximation de $\square\,\Sigma$ (Gauss élim)
x'_Σ	approximation de $\square\,\Sigma'$ (Gauss élim)

$$\mathbf{A} = \begin{pmatrix} [1,1.1] & [0.4,0.5] \\ [0.4,0.5] & 0.3333333 \end{pmatrix}$$
$$\mathbf{b} = \begin{pmatrix} [1,1.1] \\ 1 \end{pmatrix}$$

L'explication est qu'une fois préconditionnée, la matrice intervalle est centrée autour de l'identité, elle est donc cette fois *bien conditionnée*. Cette structure de la matrice permet de limiter très efficacement le pessimisme des calculs par intervalles.

Les premiers travaux sur le préconditionnement de matrices intervalles sont dus à Hansen [Han65].

4.3.1 Forte régularité

Maintenant que nous avons illustré l'intérêt du préconditionnement, la première chose à faire est de définir la classe des matrices sur lesquelles il peut être appliqué. Or, pour qu'il soit applicable, il suffit que le milieu de la matrice soit inversible (on note $\check{\mathbf{A}} := \text{mid } \mathbf{A}$). Cela étant, notre but est de manipuler des matrices régulières, et la multiplication d'une matrice intervalle régulière par une matrice réelle régulière peut produire une matrice intervalle non-régulière :

Exemple 4.4
Soit $\mathbf{A} = \begin{pmatrix} [1,2] & [1,15] \\ -1 & [1,15] \end{pmatrix}$. *On a* $\check{\mathbf{A}}^{-1}\mathbf{A} = \begin{pmatrix} [0.8,1.2] & [-5.6,5.6] \\ [-0.025,0.025] & [0.12,1.9] \end{pmatrix}$, *qui est non-régulière puisqu'elle contient la matrice* $\begin{pmatrix} 0.8 & 5.6 \\ 0.02 & 0.14 \end{pmatrix}$ *de déterminant nul.*

Nous nous intéressons donc aux matrices intervalles qui n'entrent pas dans ce cas-là, c'est à dire les matrices qui *restent régulières après préconditionnement*, et que l'on qualifie de *fortement* régulières[1] :

[1] Récemment Rohn [Roh05] a trouvé un moyen de caractériser les matrices régulières et non-fortement régulières (c.a.d., de clairement départager les deux propriétés) en utilisant une condition de régularité basée sur les matrices "extrêmes" [Roh89].

Définition 4.8 *Soit* $\mathbf{A} \in \mathbb{IR}^{n\times n}$.

$\mathbf{A}$ *est* **fortement régulière** *si* $\mathbf{A}$ *et* $\check{\mathbf{A}}^{-1}\mathbf{A}$ *sont régulières.*

Évidemment, cette définition implique que $\mathbf{A}$ est régulière. Nous avons employé le terme "recentrer" en introduction. Vérifions que le milieu d'une matrice préconditionnée est bien l'identité :

Soit $\mathbf{A} \in \mathbb{IR}^{n\times n}$. Supposons que $\check{\mathbf{A}}$ soit inversible. La matrice préconditionnée est donc $\mathbf{A}' := \check{\mathbf{A}}^{-1}\mathbf{A}$, et on a bien :

$$\begin{aligned}\text{mid}\,(\mathbf{A}') &= \text{mid}\,(\check{\mathbf{A}}^{-1}\mathbf{A}) = \text{mid}\,(\check{\mathbf{A}}^{-1}(\check{\mathbf{A}} + [-\text{rad}\ \mathbf{A}, +\text{rad}\ \mathbf{A}])) \\ &= \text{mid}\,(\check{\mathbf{A}}^{-1}\check{\mathbf{A}}) + \text{mid}\,(\check{\mathbf{A}}^{-1}[-\text{rad}\ \mathbf{A}, +\text{rad}\ \mathbf{A}]) \\ &= \text{mid}\ I + |\check{\mathbf{A}}^{-1}|\,\text{mid}\,[-\text{rad}\ \mathbf{A}, +\text{rad}\ \mathbf{A}] = I + 0 = I.\end{aligned}$$

Dans le reste de ce paragraphe, nous noterons $\mathbf{A}' := \check{\mathbf{A}}^{-1}\mathbf{A}$. Le lien fondamental entre le préconditionnement et les types de matrices étudiées auparavant réside dans la proposition suivante :

Proposition 4.12 *Si* $\check{\mathbf{A}}$ *est inversible, alors :*

$$\begin{array}{lcll}(i) & \mathbf{A} \textit{ est fortement régulière} & \Longleftrightarrow & (ii) \quad \mathbf{A}' \textit{ est une H-matrice,} \\ & & \Longleftrightarrow & (iii) \quad \rho(\text{rad}\ \mathbf{A}') < 1, \\ & & \Longleftrightarrow & (iv) \quad (I - \text{rad}\ \mathbf{A}') \textit{ inversible et } (I - \text{rad}\ \mathbf{A}')^{-1} \geq 0.\end{array}$$

Preuve.
$(i) \iff (iii)$. Si $\mathbf{A}'$ est régulière, alors $x \in \Sigma(\mathbf{A}', 0) \Longrightarrow x = 0$. Or, d'après la proposition 2.11 p.32, $x \in \Sigma(\mathbf{A}', 0)$ équivaut à $|(\text{mid}\ \mathbf{A}')x| \leq (\text{rad}\ \mathbf{A}')|x|$, donc $|x| \leq (\text{rad}\ \mathbf{A}')|x| \Longrightarrow x = 0$. Supposons $\rho(\text{rad}\ \mathbf{A}') \geq 1$ et notons u un vecteur de Perron (cf. théorème 4.1) de $(\text{rad}\ \mathbf{A}')$, ce vecteur étant non nul. Comme $\text{rad}\,(\mathbf{A}')$ est positive, $\text{rad}\,(\mathbf{A}')u = \rho(\text{rad}\ \mathbf{A}')u$, d'où $\text{rad}\,(\mathbf{A}')u \geq u$. On a vu que $|x| \leq (\text{rad}\ \mathbf{A}')|x| \Longrightarrow x = 0$; en posant $|x| = u$ on a donc $u = 0$ ce qui est absurde. Ainsi, $\rho(\text{rad}\ \mathbf{A}') < 1$. Réciproque : voir [Neu90].
$(iii) \iff (ii)$. Comme annoncé déjà dans notre remarque qui suit la proposition 4.7, on a $<\mathbf{A}'> = I - (\text{rad}\ \mathbf{A}')$ et donc $\rho(\text{rad}\ \mathbf{A}') < 1$ ssi $<\mathbf{A}'>$ est une M-matrice, i.e., ssi $\mathbf{A}'$ est une H-matrice.
$(iii) \iff (iv)$. Découle de la proposition 4.4 p.82. □

Les H-matrices $\mathbf{A}'$ obtenues après préconditionnement sont donc bien particulières. En effet, elles vérifient $\text{mid}\,(\mathbf{A}') = I$ et $\rho(\text{rad}\ \mathbf{A}') < 1$, ce qui ne forme qu'une condition suffisante pour être une H-matrice. En pratique, toute matrice intervalle régulière ayant un rayon suffisamment petit est fortement régulière. La forte régularité n'est donc pas si *forte*. Il reste à énoncer que le préconditionnement agrandit l'ensemble des solutions (au mieux le conserve), ou en un mot, que le préconditionnement est conservatif.

Théorème 4.2 *Soit* C *une matrice scalaire inversible.*

$$\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \Sigma(C\mathbf{A}, C\mathbf{b}).$$

Preuve. $x \in \Sigma(\mathbf{A}, \mathbf{b}) \Longrightarrow \big(\exists A \in \mathbf{A}\big)\big(\exists b \in \mathbf{b}\big) Ax = b$. On a $Ax = C^{-1}Cb$, donc $(CA)x = Cb$. On termine en notant que $CA \in C\mathbf{A}$ et $Cb \in C\mathbf{b}$. □

Curieusement, si on généralise le préconditionnement en prémultipliant $\mathbf{A}$ par une matrice C scalaire quelconque et en le postmultipliant par une autre matrice C' quelconque, on n'étend pas la classe des matrices restant régulières après préconditionnement [Neu90]. Autrement dit, si $C\mathbf{A}C'$ est une H-matrice, alors $\mathbf{A}$ est fortement

régulière ! Remarquons qu'on pourrait se contenter d'exiger que $C\mathbf{A}C'$ soit simplement régulière (dans le cas où $C = \text{mid } \mathbf{A}$ et $C' = I$, c'est donc équivalent d'après la proposition 4.12), auquel cas la classe serait sans doute étendue, mais cette condition n'est pas vérifiable aisément, alors que la propriété de H-matrice, elle, l'est.

Ainsi, la propriété de forte régularité est très générale, et non spécifique comme pourrait le laisser penser le choix très particulier de la matrice (mid $\mathbf{A}$). Il ne faut pas en conclure que le choix de (mid $\mathbf{A}$) est toujours au moins aussi bon que celui d'une autre matrice C, dans le sens où le préconditionnement par le milieu serait celui qui agrandirait le moins l'ensemble solution.

Concluons par une rapide classification des différents types de matrices rencontrées. On montre qu'une M-matrice, une H-matrice (voir ci-dessous) ou une matrice d'inverse positif est fortement régulière, ainsi, la forte régularité apparaît comme une généralisation supplémentaire.

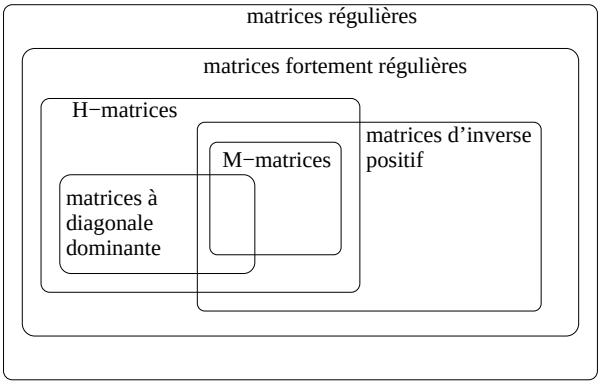
Proposition 4.13 *Une H-matrice est fortement régulière.*

Preuve. Soit $\mathbf{A}$ une H-matrice. Alors $\check{\mathbf{A}}$ est une H-matrice (proposition 4.9) et donc $|\check{\mathbf{A}}|^{-1} \leq <\check{\mathbf{A}}>^{-1}$ (proposition 4.11), d'où :

$$|\check{\mathbf{A}}^{-1}(\text{rad } \mathbf{A})| \leq |\check{\mathbf{A}}^{-1}|(\text{rad } \mathbf{A}) \leq <\check{\mathbf{A}}> (\text{rad } \mathbf{A}).$$

Ceci implique $\rho(\check{\mathbf{A}}^{-1}(\text{rad } \mathbf{A})) \leq \rho(<\check{\mathbf{A}}>^{-1} (\text{rad } \mathbf{A}))$ (proposition 4.5), et comme $\rho(<\check{\mathbf{A}}>^{-1} (\text{rad } \mathbf{A})) < 1$ (proposition 4.10), on a bien $\rho(\check{\mathbf{A}}^{-1}(\text{rad } \mathbf{A})) < 1$, c'est à dire que $\mathbf{A}$ est fortement régulière (proposition 4.12 (ii)). □

On obtient finalement le schéma récapitulatif suivant :



4.4 Itération de Gauss-Seidel

Les bases théoriques étant posées, nous décrivons maintenant les opérateurs les plus connus calculant une approximation extérieure de l'ensemble des solutions d'un système linéaire par intervalles, en commençant par ceux de Gauss-Seidel et Krawczyk.

On peut distinguer dans ces opérateurs deux versions différentes :
- la version contractante,
- la version à point fixe « pur » (sans calcul d'intersection).

La version contractante consiste, à partir d'une boîte initiale, à calculer une approximation extérieure des solutions contenues dans cette boîte en réduisant successivement ses bornes jusqu'à ce qu'il n'y ait plus d'amélioration possible. L'avantage de la version contractante est de pouvoir être appliquée dans toutes les situations,

y compris en présence de singularité. Son inconvénient est qu'elle ne calcule que l'ensemble des solutions dans une boîte donnée. Obtenir l'ensemble de toutes les solutions nécessite une estimation initiale de cet ensemble, et une estimation trop large peut empêcher toute réduction.

La version à point fixe permet de trouver une approximation de l'ensemble de *toutes* les solutions en partant de n'importe quelle boîte (la boîte dégénérée 0 par exemple) et en itérant jusqu'à l'obtention d'un point fixe. L'outil fondamental sous-jacent est le théorème de Schröder (voir plus bas) qui garantit l'existence d'un point fixe unique de l'itération. L'avantage de la version à point fixe est qu'elle ne nécessite pas d'estimation initiale. Son inconvénient est d'une part qu'elle requiert de fortes conditions sur la matrice pour assurer la convergence, d'autre part, il n'est jamais garanti que le point fixe soit atteint en un nombre fini d'itérations : ce n'est qu'une limite. Or, seul le point fixe est une approximation extérieure de l'ensemble cherché ; rien ne garantit qu'à une itération précédent le point fixe, la boîte obtenue comprenne toutes les solutions. Il faudrait donc en pratique quelques adaptations pour que la méthode reste conservative. La version à point fixe a donc surtout un intérêt théorique, mais elle pourrait toutefois être utile dans la partie *existence* du test de Newton qui implique une approximation extérieure de *toutes* les solutions d'un système linéaire (cf. commentaires de la proposition 2.14).

L'opérateur de Gauss-Seidel se décline dans les deux versions. Nous commençons par décrire la version à point fixe.

Théorème 4.3 (Schröder [Neu90])
Soit P une matrice positive telle que $\rho(P) < 1$, et soit $f : \mathbb{IR}^n \to \mathbb{IR}^n$ une fonction intervalle telle que

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{IR}^n \times \mathbb{IR}^n, \quad q(f(\mathbf{x}), f(\mathbf{y})) < Pq(\mathbf{x}, \mathbf{y}),$$

où $q : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ est la distance de Hausdorff (vectorielle) définie par

$$q(\mathbf{x}, \mathbf{y}) := |(\text{mid } \mathbf{x}) - (\text{mid } \mathbf{y})| + |(\text{rad } \mathbf{x}) - (\text{rad } \mathbf{y})|.$$

Alors pour tout $\mathbf{x}_0 \in \mathbb{IR}$, la suite récurrente $\mathbf{x}_k := f(\mathbf{x}_{k-1})$ converge vers un unique point fixe.

Remarque 4.3 *La distance q s'interprète ainsi. Elle est entre deux intervalles $\mathbf{x}$ et $\mathbf{y}$ l'agrandissement minimal nécessaire du rayon de chacun des deux intervalles pour qu'il recouvre l'autre, c'est à dire :*

$$q(\mathbf{x}, \mathbf{y}) = \inf\{\delta \geq 0 \mid \mathbf{x} \subseteq \mathbf{y} + [-\delta, +\delta] \text{ et } \mathbf{y} \subseteq \mathbf{x} + [-\delta, +\delta]\}.$$

4.4.1 Principe

Soit $\mathbf{A}x = \mathbf{b}$ un système linéaire à coefficients intervalles. La méthode de Gauss-Seidel (et ses dérivés) permet le calcul d'une approximation extérieure de $\square\,\Sigma(\mathbf{A}, \mathbf{b})$.

Dans ce §4.4.1, on définit $\mathbf{D}$ (resp. $\mathbf{L}$, $\mathbf{U}$) comme étant la matrice diagonale (resp. triangulaire strictement inférieure, triangulaire strictement supérieure) de $\mathbf{A}$ de telle sorte que $\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}$.

L'idée consiste à voir $\mathbf{A}x = \mathbf{b}$ comme un système de n équations à n variables, et à modifier le domaine de la variable x_i en résolvant (par projection) la $i^{\text{ème}}$ équation pour cette variable-ci. Chaque équation ne présentant pas d'occurrence multiple de variables, la projection fournit un ensemble optimal de solutions. Ce principe de fonctionnement rappelle l'approche de la programmation par contraintes (cf. chapitre 6) et tout particulièrement l'algorithme 2B (cf. §6.4) qui utilise la même technique de projection pour n'importe quel système non-linéaire. Voir [Gou05] pour un travail sur cette analogie. On écrit :

$$(\mathbf{x}_i)^{(k+1)} := \frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{j \neq i} \mathbf{A}_{ij}\mathbf{x}_j^{(k)}), \tag{4.7}$$

en supposant pour le moment que $\forall i \in [1..n]$, $0 \notin \mathbf{A}_{ii}$.

La boîte $\mathbf{x}^{(k+1)}$ obtenue en calculant $(\mathbf{x}_i)^{(k+1)}$ pour i allant de 1 à n avec (4.7) est une approximation extérieure[2] des solutions de l'équation

$$\mathbf{D}x^{(k+1)} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} \tag{4.8}$$

d'inconnue $x^{(k+1)}$. Ce balayage des n équations correspond à ce que nous appellerons une *étape* de l'itération. La méthode de **Jacobi** consiste à calculer une suite $\mathbf{x}^{(k)}$ en répétant en boucle l'étape décrite par la relation (4.8) jusqu'à l'obtention d'un point fixe.

Cette méthode utilise donc le fait que l'on cherche à obtenir un point fixe en réinjectant la boîte obtenue par une étape en entrée de l'étape suivante. Mais on peut accélérer le processus de convergence si au cours d'une même étape les domaines sont réactualisés. En effet, une fois $\mathbf{x}_i^{(k+1)}$ calculé, on peut le substituer directement à $\mathbf{x}_i^{(k)}$ dans les calculs suivants. L'équation (4.7) devient :

$$(\mathbf{x}_i)^{(k+1)} := \frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{j=1}^{i-1} \mathbf{A}_{ij}\mathbf{x}_j^{(k+1)} - \sum_{j=i+1}^{n} \mathbf{A}_{ij}\mathbf{x}_j^{(k)}). \tag{4.9}$$

Cet algorithme correspond à la méthode de **Gauss-Seidel**.

Plus généralement, si les équations formées par les lignes de $\mathbf{A}$ sont traitées dans l'ordre, pour toute matrice triangulaire inférieure $\mathbf{M}$ et toute matrice $\mathbf{N}$ telles que $\mathbf{A} = \mathbf{M} + \mathbf{N}$, on est capable de résoudre de la même manière que pour (4.8) le système

$$\mathbf{M}x^{(k+1)} = \mathbf{b} - \mathbf{N}\mathbf{x}^{(k)} \tag{4.10}$$

par un simple balayage (i.e., n projections). On appelle *opérateur de Gauss* l'algorithme qui étant donné une matrice $\mathbf{M}$ triangulaire inférieure et un vecteur $\mathbf{y}$ produit la boîte $\mathbf{x}$ englobant toutes les solutions du système $\mathbf{M}x = \mathbf{y}$. Cette boîte sera notée $IGA(\mathbf{M}, \mathbf{y})$[3].

Ainsi, on peut décrire les méthodes de Jacobi et Gauss-Seidel comme étant les itérations suivantes :

Jacobi	$\mathbf{x}^{(k+1)} := IGA(\mathbf{D}, \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)})$
Gauss-Seidel	$\mathbf{x}^{(k+1)} := IGA(\mathbf{L} + \mathbf{D}, \mathbf{b} - \mathbf{U}\mathbf{x}^{(k)})$

D'autres découpages[4] de $\mathbf{A}$ en 2 matrices $\mathbf{M}$ et $\mathbf{N}$ sont possibles, mais $\mathbf{M}$ doit évidement rester triangulaire inférieure, ce qu'on supposera toujours par la suite. Finalement, on va étudier la convergence de la méthode itérative générale :

$$\mathbf{x}^{(k+1)} := IGA(\mathbf{M}, \mathbf{b} - \mathbf{N}\mathbf{x}^{(k)}). \tag{4.11}$$

4.4.2 Convergence

On comprend intuitivement que la méthode de Jacobi ne peut être opérante que si dans l'expression (4.7) le diviseur est grand (en terme de rayon) en comparaison au dividende (de telle sorte que le rayon de l'intervalle produit soit petit). Si on généralise ce principe avec deux matrices $\mathbf{M}$ et $\mathbf{N}$, cela peut s'écrire $<\mathbf{M}> \gg |\mathbf{N}|$, ou encore $<\mathbf{M}> \sim <\mathbf{M}> + |\mathbf{N}|$. Si on suppose un instant que $<\mathbf{M}>$ est une M-matrice, on vérifie alors les conditions de la proposition 4.7 p.84, et on obtient ainsi une condition qui s'écrit $\rho(<\mathbf{M}>^{-1} |\mathbf{N}|) < 1$.

De façon rigoureuse :

[2] La boîte obtenue n'est plus qu'une approximation des solutions, du fait de l'effet d'enveloppe.
[3] Remarquons que $IGA(\mathbf{M}, \mathbf{y})$ ne résout **pas** le système-intervalle $\mathbf{M}\mathbf{x} = \mathbf{y}$, d'inconnue $\mathbf{x} \in \mathbb{IR}^n$. Cf. §5.2 p.113
[4] *Splitting* en anglais.

Théorème 4.4 *Supposons* $\mathbf{M}$ *régulière. L'itération (4.11) converge quelle que soit la boîte initiale* $\mathbf{x}^{(0)}$ *vers un unique point fixe* $\mathbf{x}^*$ *ssi* $\rho(<\mathbf{M}>^{-1} |\mathbf{N}|) < 1$.

Preuve. On reprend ici la preuve donnée dans [AM00].

($\Longrightarrow$) Par l'absurde. Supposons $\rho(<\mathbf{M}>^{-1} |\mathbf{N}|) \geq 1$.
A partir des trois règles arithmétiques suivantes :

$$\begin{array}{ll} \big(\forall \mathbf{a} \in \mathbb{IR}\big)\big(\forall \mathbf{b} \in \mathbb{IR}\big) & \text{rad}\,(\mathbf{a}+\mathbf{b}) = \text{rad}\,\mathbf{a} + \text{rad}\,\mathbf{b}, \\ & \text{rad}\,\mathbf{ab} \geq |\mathbf{a}|\,\text{rad}\,\mathbf{b}, \\ & \text{rad}\,\mathbf{a}/\mathbf{b} \geq (\text{rad}\,\mathbf{a})/ <\mathbf{b}>, \end{array}$$

on prouve facilement que $\mathbf{x}^{(k+1)} = IGA(\mathbf{M}, \mathbf{b} + \mathbf{N}\mathbf{x}^{(k)}) \Longrightarrow <\mathbf{M}>$ rad $\mathbf{x}^{(k+1)} \geq |\mathbf{N}|\,\text{rad}\,\mathbf{x}^{(k)}$. Or, d'après la proposition 4.8 p.85, $<\mathbf{M}>$ est une M-matrice car $\mathbf{M}$ est triangulaire inférieure, donc $<\mathbf{M}>^{-1} \geq 0$ et $<\mathbf{M}>^{-1}$ $|\mathbf{N}| \geq 0$. Ainsi $(\text{rad}\,\mathbf{x}^{(k+1)}) \geq <\mathbf{M}>^{-1} |\mathbf{N}|\,\text{rad}\,\mathbf{x}^{(k)}$. Posons $P = <\mathbf{M}>^{-1} |\mathbf{N}|$. Comme $P \geq 0$, prenons un vecteur $\mathbf{x}$ quelconque vérifiant simplement rad $\mathbf{x} = u$, où u est un vecteur de Perron de P. Comme $u \gneqq 0$, pour au moins une coordonnée i, $u_i > 0$, donc rad $\mathbf{x}_i > 0$. Multiplions $\mathbf{x}$ par un scalaire λ de telle sorte que rad $\lambda\mathbf{x}_i >$ rad $\mathbf{x}_i^* + \epsilon$ pour un $\epsilon > 0$. Et posons $\mathbf{y}^{(0)} = \lambda\mathbf{x}$. Remarquons que rad $\mathbf{y}^{(0)} = \text{rad}\,\lambda\mathbf{x} = |\lambda|\,\text{rad}\,\mathbf{x}$ est toujours un vecteur de Perron de P. Appliquons maintenant l'itération (4.11) sur $\mathbf{y}^{(0)}$. On a $(\text{rad}\,\mathbf{y}^{(1)}) \geq P\,\text{rad}\,\mathbf{y}^{(0)} = \rho(P)\,\text{rad}\,\mathbf{y}^{(0)}$ donc $(\text{rad}\,\mathbf{x}_i^{(1)}) > \rho(P)(\text{rad}\,\mathbf{x}_i^* + \epsilon)$, et par une récurrence immédiate $(\text{rad}\,\mathbf{y}_i^{(k)}) > \rho(P)^k(\text{rad}\,\mathbf{x}_i^* + \epsilon)$. Par passage à la limite, on obtient rad $\mathbf{x}_i^* \geq$ rad $\mathbf{x}_i^* + \epsilon$. Contradiction.

($\Longleftarrow$) Les trois propriétés suivantes peuvent être vérifiées pour des intervalles $\mathbf{a}$, $\mathbf{a}'$, $\mathbf{b}$, $\mathbf{b}'$ et $\mathbf{c}$ quelconques :

$$\begin{array}{l} q(\mathbf{a}/\mathbf{c}, \mathbf{b}/\mathbf{c}) = (q(\mathbf{a}, \mathbf{b}))/ <\mathbf{c}> \quad (\text{si } 0 \notin \mathbf{c}), \\ q(\mathbf{a}+\mathbf{a}', \mathbf{b}+\mathbf{b}') = q(\mathbf{a}, \mathbf{a}') + q(\mathbf{b}, \mathbf{b}'), \\ q(\mathbf{ac}, \mathbf{bc}) = |\mathbf{c}| q(\mathbf{a}, \mathbf{b}). \end{array}$$

De ces propriétés, on déduit facilement $< M > q(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) \leq |N| q(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)})$, i.e., $q(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) \leq Pq(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)})$. Il suffit ensuite d'appliquer le théorème de Schröder. □

Corollaire 4.1 *Les itérations de Jacobi et de Gauss-Seidel convergent pour toute boîte initiale ssi* $\mathbf{A}$ *est une H-matrice.*

Preuve. Tout d'abord, si $\mathbf{A}$ est une H-matrice alors diag$(\mathbf{A}) > 0$, donc $\mathbf{M}$ est une H-matrice aussi bien dans la méthode de Jacobi que Gauss-Seidel puisque dans les 2 cas, $\mathbf{M}$ comporte la diagonale de $\mathbf{A}$ (on applique alors la proposition 4.8). Comme $<\mathbf{M}>$ est une M-matrice, $\rho(<\mathbf{M}>^{-1} |\mathbf{N}|) < 1$ est vrai ssi $<\mathbf{M}> - |\mathbf{N}|$ est une M-matrice. Mais $\mathbf{N}$ ayant sa diagonale nulle, $-|\mathbf{N}| = <\mathbf{N}>$ et $<\mathbf{M}> - |\mathbf{N}| = <\mathbf{M}> - <\mathbf{N}> = <\mathbf{A}>$. Ainsi $<\mathbf{A}>$ est une M-matrice, i.e., $\mathbf{A}$ est une H-matrice. Au final $\rho(<\mathbf{M}>^{-1} |\mathbf{N}|) < 1 \Longleftrightarrow \mathbf{A}$ est une H-matrice. □

Remarque 4.4 *L'une de ces itérations peut converger pour une boîte initiale particulière sans que* $\mathbf{A}$ *soit pour autant une H-matrice (ni même régulière).*

Rappelons que l'itération (4.11) prenant n'importe quelle estimée initiale (et pas forcément une boîte contenant déjà $\Sigma(\mathbf{A}, \mathbf{b})$), il n'est pas évident a priori que cette méthode calcule bien une approximation extérieure de $\Sigma(\mathbf{A}, \mathbf{b})$. La proposition suivante le montre.

Proposition 4.14

Pour toute H-matrice $\mathbf{A} \in \mathbb{IR}^{n \times n}$ *et* $\forall \mathbf{b} \in \mathbb{IR}^n \quad \square\,\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}^*$,

où $\mathbf{x}^*$ *désigne le point fixe de l'itération (4.11).*

Preuve. Il suffit de montrer que pour tout $x \in \Sigma(\mathbf{A}, \mathbf{b})$, en prenant l'intervalle dégénéré $[x, x]$ comme estimée initiale, on a $x \in \mathbf{x}^*$. Cela signifie bien, par unicité du point fixe, que $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}^*$ (donc que $\square\,\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}^*$ puisque $\mathbf{x}^*$ est une boîte). Pour montrer que $x \in \mathbf{x}^*$, il suffit de montrer que $x \in IGA(\mathbf{M}, \mathbf{b} + \mathbf{N}[x, x])$, i.e., que chaque itération conserve x. Or $x \in \Sigma(\mathbf{A}, \mathbf{b}) \Longrightarrow \exists A \in \mathbf{A}, \exists b \in \mathbf{b}, Ax = b \Longrightarrow \exists M \in \mathbf{M}, \exists N \in \mathbf{N}, \exists b \in \mathbf{b}, (M - N)x = b$. Or $(M - N)x = b \Longrightarrow Mx = b + Nx \Longrightarrow x \in IGA(\mathbf{M}, \mathbf{b} + \mathbf{N}[x, x])$ (en considérant chaque équation séparément). □

Proposition 4.15 (Barth & Nuding)
Si $\mathbf{A}$ *est une M-matrice alors l'itération de Gauss-Seidel converge vers* $\square\,\Sigma(\mathbf{A}, \mathbf{b})$.

Remarque 4.5 *Cette proposition peut être légèrement étendue [AM00] aux itérations qui vérifient :* $\mathbf{M}$ *est une M-matrice, et* $\underline{\mathbf{N}} > 0$.

4.4.3 Synthèse

Considérons un système $\mathbf{A}x = \mathbf{b}$, où $\mathbf{A}$ est une matrice quelconque.

Supposons tout d'abord que $\mathbf{A}$ n'est pas une H-matrice et faisons l'hypothèse raisonnable que le seul moyen d'obtenir à partir de $\mathbf{A}x = \mathbf{b}$ un système $\mathbf{A}'x = \mathbf{b}'$, où $\mathbf{A}'$ est une H-matrice, est le préconditionnement. D'après le corollaire 4.1, l'algorithme de Gauss-Seidel fonctionne **ssi** la matrice en entrée est une H-matrice. Or, pour obtenir une H-matrice, il faut (d'après notre hypothèse) et suffit de préconditionner le problème. Le résultat de ce préconditionnement est effectivement une H-matrice **ssi** $\mathbf{A}$ est fortement régulière (proposition 4.12). On a donc déterminé une condition nécessaire et suffisante à l'applicabilité de la méthode de Gauss-Seidel : $\mathbf{A}$ doit être fortement régulière.

Reste le cas où $\mathbf{A}$ est déjà, initialement, une H-matrice, auquel cas le préconditionnement devient inutile. Mais nous avons vu que, dans ce cas, $\mathbf{A}$ était aussi fortement régulière (proposition 6.18). Donc finalement, nous en concluons :

> *En admettant que le préconditionnement soit le seul moyen d'obtenir une H-matrice à partir d'une matrice quelconque (qui n'est pas déjà une H-matrice), alors, les seules matrices intervalles sur lesquelles l'itération de Gauss-Seidel peut être appliquée sont les matrices fortement régulières.*

Remarquons que dans un certain nombre de cas, il a été prouvé qu'en présence d'une M-matrice il était possible de préconditionner (mais autrement que par le milieu) de telle sorte que l'ensemble solution reste inchangé (voir [Neu90], proposition 4.1.5 et 4.1.6), donc ce préconditionnement peut également être appliqué sans incidence.

4.4.4 Version contractante

En général, l'ensemble $\Sigma(\mathbf{A}, \mathbf{b})$ intéresse peu. La résolution de systèmes linéaires sert principalement comme brique dans une méthode de Newton par intervalles (cf. §2.8.2 p.33), et en général, l'ensemble recherché est plutôt de la forme $\Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x}$, où $\mathbf{x}$ désigne une "boite courante" (y compris pour la partie "existence" de l'opérateur de Newton).

Évidemment, pour calculer $\Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x}$, il est possible d'appliquer Gauss-Seidel comme décrit ci-dessus, puis d'intersecter le résultat $\mathbf{x}^*$ obtenu avec $\mathbf{x}$, si tant est que $\mathbf{x}^*$ soit atteint en un nombre fini d'itérations. Mais si on choisit comme estimée initiale $\mathbf{x}$ lui-même, on s'attend alors au résultat suivant :

Proposition 4.16 *Soit $\mathbf{x} \in \mathbb{IR}^n$. Soit $\mathbf{x}^{(k)}$ défini par l'itération (4.11) calculée pour $\mathbf{x}^{(0)} := \mathbf{x}$. Alors*

$$\forall k > 0, \quad \Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x} \subseteq \mathbf{x}^{(k)}$$

Preuve. Soit $x \in \Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x}$. Alors $\exists A \in \mathbf{A}, b \in \mathbf{b}$ tels que $Ax = b$, donc $\exists M \in \mathbf{M}, N \in \mathbf{N}, b \in \mathbf{b}$ tels que $Mx = b - Nx$. Par récurrence, si $x \in \mathbf{x}^{(k)}$, comme $IGA(\mathbf{M}, b - \mathbf{N}\mathbf{x}^{(k)})$ calcule la boîte extérieure du système (4.10), on a $x \in \mathbf{x}^{(k+1)}$. □

Puisque l'ensemble recherché est toujours contenu dans la boîte produite par une itération, on peut donc anticiper en calculant, à l'intérieur d'une étape, pour chaque composante, l'intersection de sa projection avec son domaine actuel. On écrit donc :

$$(\mathbf{x}_i)^{(k+1)} := \frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{j \neq i} \mathbf{A}_{ij}\mathbf{x}_j^{(k)}) \cap \mathbf{x}_i^{(k)} \tag{4.12}$$

On définit de même un opérateur de *Gauss contractant*, qui étant donné une matrice $\mathbf{M}$ (triangulaire inférieure), un vecteur $\mathbf{y}$ et une boîte $\mathbf{x}$, calcule la boîte extérieure $\mathbf{x}'$ des solutions du système $\mathbf{M}x = \mathbf{y}$ situées à l'intérieur de $\mathbf{x}$. Cette boîte sera notée $IGAC(\mathbf{M}, \mathbf{y}, \mathbf{x})$. L'itération de Gauss-Seidel, sous sa forme générique et contractante s'écrit maintenant :

$$\mathbf{x}^{(k+1)} := IGAC(\mathbf{M}, \mathbf{b} - \mathbf{N}\mathbf{x}^{(k)}, \mathbf{x}^{(k)}) \tag{4.13}$$

L'intérêt de cette version est d'une part de détecter l'absence de solution dans une boîte plus rapidement (plutôt que d'attendre que le point fixe soit atteint pour faire l'intersection avec le domaine considéré)[5], d'autre part de ne pas avoir à se soucier des conditions de convergence. L'itération (4.13) converge en effet dans tous les cas de figure. Le point fixe peut être seulement une surestimation de $\mathbf{x}^* \cap \mathbf{x}$, ou même au pire la boîte de départ.

Cette version permet également d'obtenir une réduction des bornes dans les cas où un 0 apparaît dans la diagonale de $\mathbf{A}$. En effet, supposons $0 \in \mathbf{A}_{ii}$. Une solution x de (4.8) doit vérifier $\mathbf{A}_{ii}x_i = \mathbf{y}_i$ où $\mathbf{y}$ est le membre de droite de (4.8). Nous devons donc résoudre à la $i^{ème}$ étape de projection une équation par intervalles du type $\mathbf{a} \times x = \mathbf{b}$ où $\mathbf{a}$ et $\mathbf{b}$ sont deux intervalles, avec $0 \in \mathbf{a}$. En supposant par exemple $\mathbf{b} > 0$, un raisonnement simple permet alors d'établir que

$$\big(\exists a \in \mathbf{a}\big)\big(\exists b \in \mathbf{b}\big)\ ax = b \iff x \in]-\infty, \underline{\mathbf{b}}/\underline{\mathbf{a}}] \cup [\underline{\mathbf{b}}/\overline{\mathbf{a}}, +\infty[.$$

Les solutions forment donc une union $\mathbf{x}_1 \cup \mathbf{x}_2$. Cette union est non bornée mais puisque dans le cas de la version contractante on ne cherche que les solutions dans un intervalle $\mathbf{x}$, il se peut que $\overline{\mathbf{x}_1} < \underline{\mathbf{x}}$ ou $\overline{\mathbf{x}} < \underline{\mathbf{x}_2}$ auquel cas décomposer l'intersection $\mathbf{x} \cap (\mathbf{x}_1 \cup \mathbf{x}_2)$ en $(\mathbf{x} \cap \mathbf{x}_1) \cup (\mathbf{x} \cap \mathbf{x}_2)$ permet bien de réduire une borne de $\mathbf{x}$. La division étendue (c.a.d. avec diviseur contenant 0) est détaillée au §6.9.5 p.170.

4.5 Itération de Krawczyk

Si en général les méthodes linéaires servent de base à des méthodes non-linéaires, l'opérateur de Krawczyk fait un peu figure d'exception, dans le sens où c'est une méthode conçue initialement comme alternative à la méthode de Newton (cf. §2.8.5 p.37).

Notre but est simplement de réécrire la proposition 2.17 dans le cas d'une fonction linéaire, avec toutefois une nuance : la fonction (non linéaire) est remplacée par une fonction épaisse ($x \mapsto \mathbf{b} - \mathbf{A}x$). On définit donc l'itération de Krawczyk de la façon suivante :

$$\mathbf{x}^{(k+1)} := \tilde{x} + \mathbf{b} - \mathbf{A}\tilde{x} + (I - \mathbf{A})(\mathbf{x}^{(k)} - \tilde{x}), \tag{4.14}$$

[5] On montre cependant que le seul cas où $IGAC(\mathbf{A}, \mathbf{b}, \mathbf{x})$ détecte l'absence de solution est lorsque $\mathbf{A}\mathbf{x} \cap \mathbf{b} = \emptyset$.

avec $\tilde{x} \in \mathbf{x}^{(k)}$, ce qui peut être réécrit

$$\mathbf{x}^{(k+1)} := \mathbf{b} + (I - \mathbf{A})\mathbf{x}^{(k)}. \tag{4.15}$$

Cette nouvelle forme est toujours meilleure : si on considère deux suites $x^{(k)}$ et $y^{(k)}$, définies respectivement par les itérations (4.14) et (4.15), quel que soit l'intervalle initial $x^{(0)} = y^{(0)}$ on a pour tout $k > 0$, $y^{(k)} \subseteq x^{(k)}$. Cela découle simplement du fait que l'expression (4.14) n'est autre que l'expression (4.15) dans laquelle $\mathbf{x}^{(k)}$ a été décorellé en $x + (\mathbf{x}^{(k)} - x)$. Il suffit ensuite d'appliquer par récurrence la sous-distributivité.

Remarque 4.6 *La factorisation* (4.14)→(4.15) *n'est possible que dans le cas linéaire, où chaque fonction $x \mapsto Ax$, $A \in \mathbf{A}$, coïncide avec sa différentielle de telle sorte que $f(\tilde{x}) = b - A\tilde{x} = b - M\tilde{x}$ (cf. notations du §2.8.5).*

Finalement, l'itération de Krawczyk s'écrit :

Krawczyk $\qquad\qquad \mathbf{x}^{(k+1)} := IGA(I, \mathbf{b} - (I - \mathbf{A})\mathbf{x}^{(k)})$

Remarquons que l'argument utilisé pour introduire la méthode de Krawczyk n'est plus tout à fait valable dans le cas linéaire, puisque Gauss-Seidel, version contractante, permet également de traiter des matrices singulières. Néanmoins la méthode de Krawczyk linéaire hérite d'autres propriétés de son pendant non-linéaire : l'existence et l'unicité de solutions.

4.5.1 Convergence

Proposition 4.17
L'itération de Krawczyk converge quelle que soit l'estimation initiale $\mathbf{x}^{(0)}$ ssi $\rho(|I - \mathbf{A}|) < 1$.

Preuve. Il suffit d'appliquer le théorème 4.4 avec $\mathbf{M} = I$ et $\mathbf{N} = \mathbf{A} - I$. □

Remarque 4.7 *Le corollaire 4.1 p.93 ne s'applique pas à l'itération de Krawczyk car la diagonale de $\mathbf{N}$ (i.e., $\mathbf{A} - I$) n'est pas nulle en général. Si par contre $\mathbf{A}$ est centrée sur I (ce qui devient fréquent en préconditionnant), alors on a $\rho(|I - \mathbf{A}|) = \rho(\mathrm{rad}\ \mathbf{A})$ et on retrouve donc bien exactement la même condition de convergence que pour Gauss-Seidel.*

Remarque 4.8 *Une proposition similaire peut être établie dans le cas non-linéaire [AM00]. En reprenant les notations de la proposition 2.17, si $\rho(|I - C\mathbf{J}|) < 1$, alors l'itération $\mathbf{x} \leftarrow K(\mathbf{x})$ produit une séquence de boîte qui converge vers le zéro (unique) de f. La condition signifie $I - \mathbf{J} \sim 0$, et donc l'itération $\mathbf{x} \leftarrow K(\mathbf{x})$ donne $K(\mathbf{x}) \sim \tilde{x} - f(\tilde{x})$, c'est à dire des calculs quasiment réels. On comprend intuitivement que le rayon des intervalles tende vers 0.*

4.5.2 Existence de solutions

Proposition 4.18 (Test d'existence de Krawczyk) *Soit $\mathbf{x} \in \mathbb{IR}^n$.*

$$\mathbf{b} + (I - \mathbf{A})\mathbf{x} \subseteq \mathbf{x} \implies \big(\forall A \in \mathbf{A}\big)\big(\forall b \in \mathbf{b}\big)\big(\exists x \in \mathbf{x}\big)\big(Ax = b\big).$$

Preuve. Il suffit d'appliquer le théorème de Brouwer p.32 à chaque fonction $x \to b + (I - A)x$. □

En utilisant la monotonie pour l'inclusion, on montre facilement par récurrence que si les hypothèses de la proposition sont vérifiées, alors l'itération (4.15) calculée avec $\mathbf{x}^{(0)} := \mathbf{x}$, produit une suite décroissante de boîtes, i.e., $\forall k > 0$, $\mathbf{x}^{(k+1)} \subseteq \mathbf{x}^{(k)}$. Il est fort probable qu'une telle suite possède une limite, mais il ne faut pas en déduire que l'itération converge. La convergence pour $\mathbf{x}^{(0)} := \mathbf{x}$ ne prouve en effet rien sur la convergence de l'itération (4.15) en général. La proposition 4.17 ne s'applique donc pas. En particulier, l'existence de solutions n'implique pas la régularité de $\mathbf{A}$.

4.5.3 Unicité des solutions

De même que dans le cas non-linéaire, l'itération de Krawczyk permet de prouver l'unicité des solutions en cas de convergence. L'unicité signifie ici :

$$\forall A \in \mathbf{A}, \quad \forall b \in \mathbf{b} \quad \exists x \text{ unique tel que } Ax = b,$$

ce qui équivaut à dire que $\forall A \in \mathbf{A}$, A est régulière, et donc tout simplement que $\mathbf{A}$ est régulière.

Lemme 4.2

Si $\rho(|I - \mathbf{A}|) < 1$ alors $\mathbf{A}$ est régulière.

Preuve. $\forall A \in \mathbf{A}$, $I - A \subseteq I - \mathbf{A}$ donc $|I - A| \leq |I - \mathbf{A}|$. Ceci implique $\rho(I - A) \leq \rho(|I - \mathbf{A}|)$ (cf. proposition 4.5 p.83), d'où $\rho(I - A) < 1$. Avec la proposition 4.4 p.82, on obtient que $I - (I - A)$, c'est à dire A, est régulière. □

On ne peut pas calculer en pratique le rayon spectral de $I - \mathbf{A}$. Il existe toutefois une situation qui permet de détecter la régularité facilement : on part d'une boîte initiale $\mathbf{x}^{(0)}$, puis on calcule $\mathbf{x}^{(1)}$. Si on observe $\mathbf{x}^{(1)} \subset \mathbf{x}^{(0)}$, alors la régularité (en plus de la convergence) est assurée.

Proposition 4.19 (Test d'unicité de Krawczyk) *Soit $\mathbf{x} \in \mathbb{IR}^n$.*

$$\mathbf{b} + (I - \mathbf{A})\mathbf{x} \subset \mathbf{x} \implies \mathbf{A} \text{ est régulière et } \big(\forall A \in \mathbf{A}\big)\big(\forall b \in \mathbf{b}\big)\big(\exists x \in \mathbf{x} \text{ unique}\big)\big(Ax = b\big).$$

Preuve. Notons $\mathbf{x}^{(1)} := \mathbf{b} + (I - \mathbf{A})\mathbf{x}$. D'une part ; $\mathrm{rad}\,(\mathbf{x}^{(1)}) \geq |I - \mathbf{A}|\,\mathrm{rad}\,(\mathbf{x})$, et d'autre part $\mathbf{x}^{(1)} \subset \mathbf{x}$ implique $0 \leq \mathrm{rad}\,(\mathbf{x}^{(1)}) < \mathrm{rad}\,(\mathbf{x})$. Ainsi $|I - \mathbf{A}|\,\mathrm{rad}\,(\mathbf{x}) < \mathrm{rad}\,(\mathbf{x})$. Si on note par exemple $u := \mathrm{rad}\,(\mathbf{x})$, on a donc $|I - \mathbf{A}|u < u$, pour un $u > 0$, ce qui implique $\||I - \mathbf{A}|\|_u < 1$, c'est à dire $\rho(|I - \mathbf{A}|) < 1$ (voir proposition 4.3). Il suffit ensuite d'appliquer le lemme 4.2 pour obtenir la régularité, et la proposition 4.18 pour obtenir l'existence (dans $\mathbf{x}$!), et donc forcément l'unicité, des solutions. □

En adaptant la preuve précédente, on peut montrer de façon plus générale :

$$\big(\exists \mathbf{x} \in \mathbb{IR}^n\big)\big(IGA(\mathbf{M}, \mathbf{b} - \mathbf{N}\mathbf{x}) \subset \mathbf{x}\big) \Longrightarrow \rho(<\mathbf{M}>^{-1} |\mathbf{N}|) < 1.$$

Ainsi, si dans une étape d'une itération telle que Krawczyk ou Gauss-Seidel, on observe une réduction des bornes sur chaque dimension, on a $\rho(<\mathbf{M}>^{-1} |\mathbf{N}|) < 1$. On déduit donc d'une propriété "locale" (la stricte monotonie de l'itération lorsque $\mathbf{x}^{(0)} := \mathbf{x}$), une information globale (la convergence systématique quelle que soit l'estimée initiale, cf. théorème 4.4).

4.5.4 Version contractante

De même que pour Gauss-Seidel, on définit l'itération de Krawczyk contractante :

$$\mathbf{x}^{(k+1)} := IGAC(I, \mathbf{b} - (I - \mathbf{A})\mathbf{x}^{(k)}, \mathbf{x}^{(k)}).$$

La version contractante est tout à fait compatible avec le test d'unicité puisque

$$IGA(I, \mathbf{b} - (I - \mathbf{A})\mathbf{x}^{(k)}) \subset \mathbf{x}^{(k)} \Longleftrightarrow IGAC(I, \mathbf{b} - (I - \mathbf{A})\mathbf{x}^{(k)}, \mathbf{x}^{(k)}) \subset \mathbf{x}^{(k)}.$$

4.6 Élimination de Gauss

L'élimination de Gauss est une méthode classique de résolution de systèmes linéaires qui s'adapte facilement au cas intervalle. Son avantage principal est de "préparer" la matrice A pour faciliter la résolution de plusieurs équations linéaires partageant la même matrice (donc avec seulement le vecteur b qui varie). Plus précisément, cette méthode transforme la matrice A en un produit de matrice LU par un algorithme de complexité $O(n^3)$. Une fois cette transformation effectuée, la résolution du système $(LU)x = b$ n'est plus qu'en $O(n^2)$.

4.6.1 Méthode dans le cas réel

Déroulons l'algorithme d'élimination de Gauss en prenant un système $Ax = b$, avec A régulière de dimension $n = 3$. Il consiste à simplifier le système

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} x = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{4.16}$$

en effectuant des combinaisons linéaires entre les lignes. L'idée est de faire apparaître des zéros sur la première colonne en soustrayant convenablement la première ligne aux lignes suivantes. On a :

$$(4.16) \iff \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} - \alpha_2 A_{11} & A_{22} - \alpha_2 A_{12} & A_{23} - \alpha_2 A_{13} \\ A_{31} - \alpha_3 A_{11} & A_{32} - \alpha_3 A_{13} & A_{33} - \alpha_3 A_{13} \end{pmatrix} x = \begin{pmatrix} b_1 \\ b_2 - \alpha_2 b_1 \\ b_3 - \alpha_3 b_1 \end{pmatrix}.$$

Si A_{11} (qu'on appelle *pivot*) est non nul, en choisissant $\alpha_2 := A_{21}/A_{11}$ et $\alpha_3 := A_{31}/A_{11}$, on obtient

$$(4.16) \iff \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ 0 & A'_{22} & A'_{23} \\ 0 & A'_{32} & A'_{33} \end{pmatrix} x = \begin{pmatrix} b_1 \\ b'_2 \\ b'_3 \end{pmatrix} \tag{4.17}$$

avec $A'_{ij} = A_{ij} - (A_{i1}/A_{11})A_{1j}$ et $b'_i = b_i - (A_{i1}/A_{11})b_1$. Puisque A est régulière, l'unique solution de (4.17) s'obtient en résolvant d'abord

$$\begin{pmatrix} A'_{22} & A'_{23} \\ A'_{32} & A'_{33} \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b'_2 \\ b'_3 \end{pmatrix} \tag{4.18}$$

puis en calculant $x_1 = \dfrac{1}{A_{11}}\big(b_1 - A_{12}x_2 - A_{13}x_3\big)$. Si A'_{22} (le pivot) est non nul, on peut alors réappliquer l'élimination précédente au système (4.18) pour obtenir

$$\begin{pmatrix} A'_{22} & A'_{23} \\ 0 & A''_{33} \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b'_2 \\ b''_3 \end{pmatrix} \tag{4.19}$$

avec $A''_{33} = A'_{33} - (A'_{32}/A'_{22})A'_{23}$ et $b''_3 = b'_3 - (A'_{32}/A'_{22})b'_2$. Résoudre (4.19) se ramène de nouveau à résoudre

$$\begin{pmatrix} A''_{33} \end{pmatrix} \begin{pmatrix} x_3 \end{pmatrix} = \begin{pmatrix} b''_3 \end{pmatrix} \tag{4.20}$$

puis à calculer $x_2 = \dfrac{1}{A'_{22}}\big(b'_2 - A'_{23}x_2 - A'_{33}x_3\big)$. Le système (4.20) se résout alors directement : $x_3 = \dfrac{1}{A''_{33}}b''_3$.

Pour une matrice A de taille n, on peut appliquer l'élimination précédente récursivement. Sous réserve de pivots non nuls, on peut ainsi ramener le système $Ax = b$ de façon équivalente à un système $Ux = y$, où U est une matrice triangulaire supérieure, c.a.d. de la forme

$$\begin{pmatrix} U_{11} & \dots & U_{1n} \\ & \ddots & \vdots \\ 0 & & U_{nn} \end{pmatrix},$$

qui ne dépend que de A, et y un vecteur dont chaque composante y_i s'exprime comme une combinaison linéaire de $(b_1, \ldots, b_i)$ avec 1 pour coefficient de b_i. Ceci équivaut à ce que b_i s'exprime comme une combinaison linéaire de $(y_1, \ldots, y_i)$ avec 1 pour coefficient de y_i[6]. On a donc $Ly = b$ où L est une matrice triangulaire inférieure, c.a.d. de la forme

$$\begin{pmatrix} L_{11} & & 0 \\ \vdots & \ddots & \\ L_{n1} & \ldots & L_{nn} \end{pmatrix},$$

ne dépendant que de A et vérifiant $L_{ii} = 1$ pour tout i. Finalement $Ax = b \iff (LU)x = b$, et comme L et U ne dépendent que de A, cette équivalence est vraie pour tout couple de vecteurs (x, b) ce qui implique $A = LU$. Remarquons enfin qu'il est facile d'éviter le problème du pivot nul en permutant des colonnes ou des lignes de A (ce qui revient à réordonner les composantes de x ou de b).

Construction de L et U

Plutôt que de calculer L et U en utilisant l'élimination de Gauss (méthode directe), il est possible de partir de la relation $A = LU$ (méthode indirecte). Les matrices L et U étant respectivement triangulaire inférieure et supérieure, cette égalité se réécrit pour tout $i \in [1..n]$ et tout $j \in [1..n]$:

$$\sum_{k=1}^{\max\{i,j\}} L_{ik}U_{kj} = A_{ij}. \tag{4.21}$$

En imposant que chaque élément diagonal de L vaille 1, les autres coefficients de L et ceux de U s'obtiennent en cascade à partir de (4.21) :

$$\big(\forall i \leq j\big) \quad U_{ij} = A_{ij} - \sum_{k<i} L_{ik}U_{kj} \quad \text{et} \quad \big(\forall i < j\big) \quad L_{ij} = \frac{1}{U_{ii}}(A_{ij} - \sum_{k<j} L_{ik}U_{kj}), \tag{4.22}$$

ce qui prouve au passage l'unicité de la décomposition $A = LU$ (en imposant $\mathrm{diag}(L) = \mathrm{diag}(I)$). On construit d'abord la première ligne de U, puis la seconde de L, puis la seconde de U, et ainsi de suite. On montre que la complexité de cette algorithme est en $O(2n^3/3)$.

Résolution

Une fois L et U construite, $Ax = b$ se résout en deux étapes. On résout $Ly = b$ d'inconnue y, puis $Ux = y$ d'inconnue x. Comme les deux systèmes sont triangulaires, la résolution se fait par substitutions successives (comme dans l'exemple), pour un coût total en $O(2n^2)$.

4.6.2 Méthode dans le cas intervalle

La relation (4.22) définit un procédé de construction des matrices L et U par induction. Il est possible en développant les calculs formellement et en interdisant toute simplification, d'exprimer chaque composante de L et de U en fonction de A, donc d'écrire $L = l(A)$ et $U = u(A)$, de telle sorte que l'évaluation de $l(A)$ et de $u(A)$ coïncide avec les matrices L et U obtenues par induction, quelle que soit la nature des coefficients et la sémantique des opérateurs choisie.

Si on décide dans le procédé (4.22) de remplacer A par une matrice intervalle $\mathbf{A}$, on obtiendra donc deux matrices intervalles $\mathbf{L} := l(\mathbf{A})$ et $\mathbf{U} := u(\mathbf{A})$. Il est évident qu'en dimension supérieure à 2, A possède de multiples

[6] Ou plus simplement : l'inverse d'une matrice triangulaire inférieure est triangulaire inférieure.

occurrences de variables dans l'expression de l et de u. On a donc $\mathbf{L} \supseteq range(l, \mathbf{A})$ et $\mathbf{U} \supseteq range(u, \mathbf{A})$, ce qui implique

$$\mathbf{LU} \supseteq range\Big(l(A) \times u(A'), A \in \mathbf{A}, A' \in \mathbf{A}'\Big)$$

ce qui implique de nouveau

$$\mathbf{LU} \supseteq range((l \times u)(A), A \in \mathbf{A}),$$

c'est à dire

$$\mathbf{A} \subseteq \mathbf{LU}.$$

Une fois cette décomposition effectuée, une approximation extérieure de $\Sigma(\mathbf{A}, \mathbf{b})$ s'obtient de façon similaire à la résolution dans le cas réel, c'est à dire en considérant deux boîtes $\mathbf{x}$ et $\mathbf{y}$ vérifiant :

$$\mathbf{y} \supseteq \Sigma(\mathbf{L}, \mathbf{b}) \quad \text{et} \quad \mathbf{x} \supseteq \Sigma(\mathbf{U}, \mathbf{y}).$$

On a en effet la série d'implications suivantes :

$$\begin{aligned} x \in \Sigma(\mathbf{A}, \mathbf{b}) &\implies \big(\exists A \in \mathbf{A}\big)\ Ax \in \mathbf{b} \\ &\implies \big(\exists L \in \mathbf{L}\big)\big(\exists U \in \mathbf{U}\big)\ (LU)x \in \mathbf{b} \\ &\implies \big(\exists U \in \mathbf{U}\big)\big(\exists L \in \mathbf{L}\big)\ L(Ux) \in \mathbf{b} \\ &\implies \big(\exists U \in \mathbf{U}\big)\ Ux \in \mathbf{y} \\ &\implies x \in \mathbf{x}. \end{aligned}$$

On en conclut que $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}$. Pour construire des boîtes $\mathbf{y}$ et $\mathbf{x}$ qui conviennent, il suffit d'écrire :

$$\mathbf{y}_i := \mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}\mathbf{y}_j \quad \text{et} \quad \mathbf{x}_i := \frac{1}{\mathbf{U}_{ii}}(\mathbf{y}_i - \sum_{j>i} \mathbf{U}_{ij}\mathbf{x}_j),$$

ce qu'on prouve aisément.

4.7 Méthode de Hansen-Bliek

La méthode de Hansen-Bliek concerne les matrices fortement régulières, préconditionnées par l'inverse de leur milieu. On suppose donc être en présence d'une matrice intervalle $\mathbf{A}$ de taille $n \times n$ centrée sur I, et telle que $\underline{\mathbf{A}}^{-1} \geq 0$[7] (car $\underline{\mathbf{A}} = <\mathbf{A}>$ est une M-matrice), et d'un vecteur intervalle $\mathbf{b}$ de dimension n.

La méthode s'appuie sur la formule d'Oettli-Prager exposée au §2.7.1 p.30. Nous reprenons la décomposition utilisée dans ce paragraphe (avec cette fois mid $(\mathbf{A}) = I$), c.a.d.,

$$\mathbf{A} = I + [-\Delta, +\Delta] \quad \mathbf{b} = b + [-\delta, +\delta].$$

Au vu de la proposition 4.12 p.89, la forte régularité de la matrice de départ (avant préconditionnement) s'écrit de façon équivalente :

- $I - \Delta$ est une M-matrice,
- $(I - \Delta)^{-1} \geq 0$, ou encore,
- $\rho(\Delta) < 1$.

Le système linéaire étant fixé une fois pour toute dans cette section, nous noterons $\Sigma := \Sigma(\mathbf{A}, \mathbf{b})$. Nous noterons enfin $M := (I - \Delta)^{-1}$, et m_{ij} les coefficients de M.

La méthode de Hansen-Bliek donne une formule décrivant explicitement $\square\,\Sigma$. Cette formule fait toutefois intervenir l'inverse d'une matrice scalaire, ce qui fait automatiquement monter sa complexité à $O(n^3)$.

[7] Hansen [Han92a] est légèrement plus restrictif (sans que cela soit nécessaire) en supposant que $\underline{\mathbf{A}}$ est à diagonale dominante.

Remarque 4.9 *Ning & Kearfott [NK97] montrent comment utiliser la technique de Hansen-Bliek pour calculer une approximation extérieure (cette fois non optimale) du solution set dans le cas où* $\mathbf{A}$ *est simplement une H-matrice. Pour cela, ils introduisent une manière originale de centrer une matrice sur* I *sans utiliser de préconditionnement (mais il y a toujours une surestimation...). On obtient donc* $\mathbf{A} = I + [-\Delta, +\Delta]$ *et ils montrent alors que si* $\mathbf{A}$ *est une H-matrice,* Δ *est une M-matrice ! L'avantage de leur approche est que le calcul de* $\square\,\Sigma(\mathbf{A}, \mathbf{b})$ *ne nécessite plus d'autre inversion que celle de* $(I - \Delta)$*, alors qu'une utilisation directe de la méthode de Hansen-Bliek nécessiterait deux inversions : celle de* $\check{\mathbf{A}}$ *(pour préconditionner), et celle de* $I - |\check{\mathbf{A}}^{-1}|\,\mathrm{rad}\,\mathbf{A}$*. Voir également [Neu99].*

Soit Q un quadrant (voir définition 2.9 p.31). Réécrivons le résultat de la proposition 2.9 p.31 (en prenant en compte que $A = I$) :

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \cap Q \Longleftrightarrow \begin{cases} (I - \Delta Q)x \leq b + \delta & (a) \\ (I + \Delta Q)x \geq b - \delta & (b) \\ Qx \geq 0 & (c) \end{cases} \tag{4.23}$$

Rappelons enfin que pour calculer $\square\,\Sigma$ à partir de cette dernière formule, cela nécessite de lancer a priori un algorithme d'optimisation linéaire un nombre de fois de l'ordre de 2^n. Le fait que la matrice $\mathbf{A}$ soit centrée autour de I crée une symétrie qui permettra d'éviter la combinatoire des quadrants. L'algorithme a été conçu par Hansen [Han92a] et Bliek [Bli92]. Nous utiliserons ici les notations introduites par Rohn [Roh93], qui proposa une démonstration plus rigoureuse de ce résultat.

Remarquons que si $I + \Delta$ était également d'inverse positif, alors $[I - \Delta, I + \Delta]$ serait d'inverse positif (cf. remarque 4.2 p.85, à propos du résultat de Kuttler) et que sous certaines conditions, il existe alors une méthode plus simple pour déterminer Σ (méthode mentionnée également dans la même remarque).

4.7.1 Maximisation de $|x|$

Dans un premier temps, notre but est de calculer pour chaque composante k $(1 \leq k \leq n)$ une borne max pour $|x_k|$, lorsque x décrit Σ, c'est à dire

$$x_k^* := \max_{x \in \Sigma} |x_k|.$$

Dans un second temps, nous calculons un max soit pour $x_k \geq 0$, soit pour $x_k \leq 0$, c'est à dire

$$\tilde{x}_k := \max_{x \in \Sigma, x_k \geq 0} |x_k| \quad \text{et} \quad \underset{\sim}{x}_k := -\max_{x \in \Sigma, x_k \leq 0} |x_k|.$$

L'idée de base consiste à extraire de la formule (4.23) la matrice $I - \Delta$ (notée M). L'inverse de cette matrice étant positif, multiplier chaque membre d'une inéquation $Mx \leq y$ par M^{-1} conserve l'inégalité et permet d'écrire $x \leq M^{-1}y$. De cette façon on parvient à isoler x, c'est à dire à le borner (puisqu'il s'agit d'une inéquation).

Proposition 4.20

$$x \in \Sigma \cap Q \Longleftrightarrow \begin{cases} (I - \Delta)Qx \leq Qb + \delta & (a') \\ (I + \Delta)Qx \geq Qb - \delta & (b') \\ Qx \geq 0 & (c') \end{cases} \tag{4.24}$$

Preuve. Soit $i \in [1..n]$. Si $Q_{ii} = 1$ alors (a) implique $[(I - \Delta Q)x]_i \leq [b + \delta]_i$ i.e., $x_i - \sum_{j=1}^n (\Delta_{ij} Q_{jj}) x_j \leq b_i + \delta_i$. Ceci peut être réécrit $Q_{ii} x_i - \sum_{j=1}^n \Delta_{ij}(Q_{jj} x_j) \leq Q_{ii} b_i + \delta_i$, ou, sous une forme plus compacte, $[(I - \Delta)Qx]_i \leq [Qb + \delta]_i$.

Si $Q_{ii} = -1$, alors (b) implique $[(I + \Delta Q)x]_i \geq [b - \delta]_i$, i.e., $x_i + \sum_{j=1}^n (\Delta_{ij} Q_{jj}) x_j \geq b_i - \delta_i$. Comme précédemment, on obtient $Q_{ii} x_i - \sum_{j=1}^n \Delta_{ij}(Q_{jj} x_j) \leq Q_{ii} b_i + \delta_i$, et de nouveau $[(I - \Delta)Qx]_i \leq [Qb + \delta]_i$. Cette dernière relation étant vraie pour tout i, nous avons prouvé (a').

En répétant le même argument en échangeant les rôles de (a) et (b), on obtient de la même manière (b'). De plus, chaque inéquation de (a) et (b) est utilisée une fois et une seule, et sous une forme équivalente, dans le système réunissant (a') et (b'). Par conséquent, la conjonction de (a'), (b') et (c') est bien une caractérisation de $\Sigma \cap Q$. □

Comme $x \in Q \iff |x| = Qx$, l'inéquation (a') nous permet de borner la norme des composantes de x dans n'importe quelle quadrant, pour le coût d'une seule inversion de matrice. En effet, $(a') \iff (I - \Delta)|x| \leq Qb + \delta \implies |x| \leq M(Qb + \delta)$. On peut déjà à ce stade conclure, par l'absurde, que si $M(Qb + \delta) \not\geq 0$ alors $\Sigma \cap Q = \emptyset$. Supposons donc que ce vecteur est positif et posons $x_Q := QM(Qb + \delta)$, de telle sorte que x_Q soit un point du quadrant Q de norme $M(Qb + \delta)$. On a alors $|x| \leq Qx_Q$, c'est à dire $|x| \leq |x_Q|$.

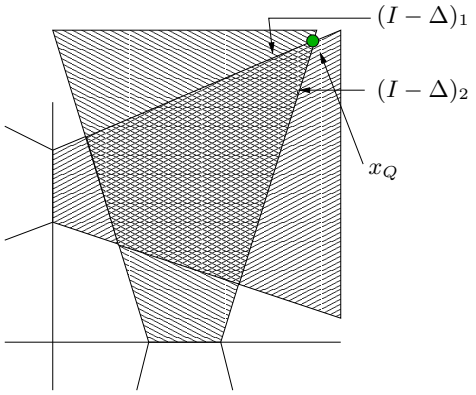


FIG. *4.1: Le point x_Q.*

Dans la figure 4.1, la droite étiquetée $(I-\Delta)_1$ représente la première ligne du système (a'), c'est à dire l'hyperplan délimitant l'espace $((I - \Delta)x)_1 \leq (Qb + \delta)_1$. De même pour $(I - \Delta)_2$. Le point x_Q est bien à l'intersection de ces n hyperplans.

Viennent ici alors trois questions :

1. Cette majoration est-elle optimale (i.e., une borne max) pour chaque quadrant ?
2. Si cette majoration n'est pas optimale, pour chaque quadrant, et pour tout $i, j \in 1..n$, les bornes maximales de $|x_i|$ et $|x_j|$ sont-elles atteintes en un même point ?
3. Comment évite-t-on de calculer x_Q pour les 2^n quadrants ?

La proposition suivante répond aux deux premières questions.

Proposition 4.21 *Soit Q un quadrant, $x_Q = QM(Qb + \delta)$. D'une part,*

$$\Sigma \cap Q \neq \emptyset \iff x_Q \in Q;$$

D'autre part, si l'une de ces deux conditions est vérifiée alors toutes les composantes de $|x|$ lorsque x décrit $\Sigma \cap Q$ atteignent leur maximum au point x_Q. Autrement dit :

$$|x_Q| = \max_{x \in \Sigma \cap Q} |x|.$$

Preuve. On a déjà vu que $\Sigma \cap Q \neq \emptyset \implies x_Q \in Q$. Réciproquement, supposons $x_Q \in Q$ et prouvons que x_Q est un point de $\Sigma \cap Q$. D'une part, cela impliquera bien que $\Sigma \cap Q \neq \emptyset$. D'autre part, comme $|x| \leq |x_Q|$ pour tout $x \in \Sigma \cap Q$, il en découlera automatiquement que toutes les composantes de $|x|$ atteignent leur maximum en x_Q lorsque x décrit $\Sigma \cap Q$.

On a $(I-\Delta)(Qx_Q) = (Qb+\delta)$. En multipliant chaque membre par Q, on obtient $(Q-Q\Delta)Qx_Q = Q^2b + Q\delta$, c'est à dire $(I - Q\Delta Q)x_Q = b + Q\delta$. Or $I - \Delta \leq I - Q\Delta Q \leq I + \Delta$ et $b - \delta \leq b + Q\delta \leq b + \delta$. D'où, $\big(\exists A \in I \pm \Delta\big)\big(\exists b \in b \pm \delta\big) Ax_Q = b$. On a donc bien $x_Q \in \Sigma \cap Q$. □

Répondons à la troisième question. A ce stade, il nous est possible pour tout k de borner $|x_k|$ en effectuant le calcul suivant (il est utile de remarquer que le max y est bien défini car Σ n'est jamais vide !) :

$$\max\{|x_k|, x \in \Sigma\} = \max\{(Qx_Q)_k \quad Q \text{ quadrant de } R^n\}.$$

Mais puisque M est positive, $M(Qb+\delta)$ est maximal lorsque chaque composante de $Qb+\delta$ est maximale, c'est à dire, lorsque chaque composante de Qb est maximale. Cette fois, la combinatoire des quadrants joue en notre faveur, puisqu'il existe un quadrant qui, composante par composante, maximise Qb. Il s'agit en effet du quadrant Q^* défini par la matrice diagonale suivante :

$$Q^*_{ii} := sign(b_i),$$

où $sign(b_i)$ vaut ± 1 suivant que b_i est positif ou négatif. On voit alors que pour tout quadrant Q, $Qb+\delta \leq Q^*b+\delta \iff Qx_Q \leq Q^*x_{Q^*}$. La simple évaluation de $Q^*(x_{Q^*})$ nous confère donc le vecteur $\max\{|x|, x \in \Sigma\}$.

Proposition 4.22 *En posant $x^* := M(|b|+\delta)$ on a $x^* = \max\{|x|, x \in \Sigma\}$.*

Cela ne nous permet pas encore de calculer les bornes de $\square\,\Sigma$ puisqu'on ne sait pas si ce max est atteint pour des composantes positives (auquel cas $\sup(\square\,\Sigma)_k = x^*_k$) ou négatives (auquel cas $\inf(\square\,\Sigma)_k = -x^*_k$).

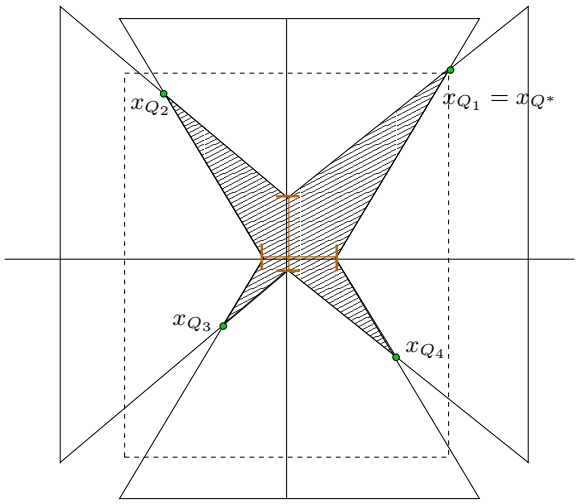


FIG. *4.2: Le quadrant Q^*.*

L'idée est alors de ne pas considérer Q^* pour traiter l'espace entier mais de se restreindre pour chaque k aux quadrants $Q^{(k+)}$ et $Q^{(k-)}$ définis par les matrices diagonales suivantes, dont la seule différence par rapport à Q^* est d'imposer le signe de x_k :

$$Q^{(k+)}_{ii} = \begin{cases} sign(b_i) & \text{si } i \neq k, \\ +1 & \text{sinon.} \end{cases} \qquad Q^{(k-)}_{ii} = \begin{cases} sign(b_i) & \text{si } i \neq k, \\ -1 & \text{sinon.} \end{cases}$$

Il est clair que le vecteur $Q^{(k+)}x_{Q^{(k+)}}$ maximise $|x|$, où x est une solution de $k^{\text{ème}}$ coordonnée positive et que ce vecteur a une composante négative ssi il n'existe pas de tel point x. On ne balaye donc plus que 2^{n-1} quadrants. Ainsi, $Q^{(k+)}x_{Q^k} \geq 0$ ssi $\tilde{x}_k := [Q^{(k+)}x_{Q^{(k+)}}]_k$ est la borne sup de la $k^{\text{ème}}$ composante de Σ. De même, $Q^{(k-)}x_{Q^-} \geq 0$ ssi $\underset{\sim}{x}_k := [-Q^{(k-)}x_{Q^{(k-)}}]_k$ est la borne inf de la $k^{\text{ème}}$ composante de Σ. Les valeurs $\tilde{x}_k$ et $\underset{\sim}{x}_k$ suffisent donc à décrire la projection de $\square\,\Sigma$ sur la dimension k, dès lors qu'il existe des points de Σ de $k^{\text{ème}}$ coordonnée négative et d'autres de $k^{\text{ème}}$ coordonnée positive :

Proposition 4.23
Posons $\tilde{x}_k := [Q^{(k+)} x_{Q^{(k+)}}]_k$ *et* $\underset{\sim}{x}_k := [-Q^{(k-)} x_{Q^{(k-)}}]_k$.

$$\begin{array}{ll} (\exists x \in \Sigma)\ x_k \geq 0 \iff \tilde{x}_k \geq 0, & \text{et dans ce cas } \tilde{x}_k = \max_{x\in\Sigma, x_k \geq 0} |x_k|; \\ (\exists x \in \Sigma)\ x_k \leq 0 \iff \underset{\sim}{x}_k \leq 0, & \text{et dans ce cas } \underset{\sim}{x}_k := -\max_{x\in\Sigma, x_k \leq 0} |x_k|. \end{array}$$

Une optimisation intéressante permet de déterminer $\tilde{x}_k$ (ou $\underset{\sim}{x}_k$) pour tout k sans avoir à calculer entièrement le vecteur $x_{Q^{(k+)}}$. Il est possible en effet de déduire la $k^{\text{ème}}$ composante de ce vecteur directement à partir de x^*.

Corollaire 4.2

$$\tilde{x}_k = \begin{cases} x^*_k & \text{si } b_k \geq 0, \\ x^*_k + 2m_{kk}b_k & \text{sinon.} \end{cases} \qquad \underset{\sim}{x}_k = \begin{cases} -x^*_k & \text{si } b_k \leq 0, \\ -x^*_k + 2m_{kk}b_k & \text{sinon.} \end{cases}$$

Preuve. Prouvons-le pour $\tilde{x}_k$. Si $sign(b_k) = 1$, on a $Q^{(k+)}_{kk} = Q^*_{kk} = 1$, c.a.d., $Q^{(k+)} = Q^*$, et $\tilde{x}_k = x^*_k$. Sinon, $Q^{(k+)} = Q^* + 2e_k e_k^T$ (où e_k est la $k^{\text{ème}}$ colonne de la matrice identité) et $Q^{(k+)} x_{Q^{(k+)}} = M(Q^{(k+)}b + \delta) = M((Q^* + 2e_k e_k^T)b + \delta)$, d'où on tire $\tilde{x}_k = x^* + 2m_{kk}b_k$. La preuve est similaire pour $\underset{\sim}{x}_k$. □

Dans la section suivante, nous traitons le cas où la borne sup de x_k est atteinte pour $x_k \leq 0$ (c'est à dire lorsque $\tilde{x}_k \leq 0$). Il suffit donc pour cela de savoir minimiser $|x_k|$. On obtiendra de façon symétrique la borne inf de x_k, lorsque celle-ci est atteinte pour $x_k \geq 0$.

4.7.2 Minimisation de $|x|$

La difficulté est que, sur un quadrant donné, les $|x_k|$ n'atteignent plus leur minimum au même point (pour différente valeur de k), ce qui est illustré sur la figure suivante.

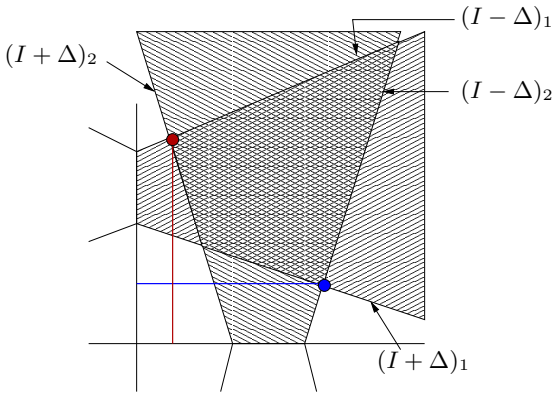


FIG. *4.3: Un exemple où le minimum pour* $|x_1|$ *n'est pas atteint au même point que le minimum pour* $|x_2|$, *lorsque* x *décrit* $\Sigma \cap Q$.

Fixons k, et cherchons à maximiser x_k pour x_k négatif. On suppose donc dans cette section $\tilde{x}_k \leq 0$.

Tout d'abord, nous aurons recours à deux petits lemmes techniques :

Lemme 4.3 $\Delta M = M\Delta = M - I$.

Preuve. $(I - \Delta)M = I \Longrightarrow M - \Delta M = I \Longrightarrow \Delta M = M - I$, de même pour $M\Delta$. □

Lemme 4.4 *Soit $k \in [1..n]$. Posons $\nu_k := 1/(2m_{kk} - 1)$. Alors*

$$0 < \nu_k \leq 1.$$

Preuve. $M > 0$ et $\Delta \geq 0 \Longrightarrow M\Delta \geq 0 \Longrightarrow M - I \geq 0$, avec le lemme 4.3. Donc $\forall\ k \in [1..n]\ m_{kk} \geq 1$, ou encore, $2m_{kk} - 1 \geq 1$. □

Considérons un quadrant Q tel que $Q_{kk} = -1$. Pour obtenir le maximum de x_k dans $\Sigma \cap Q$, l'intuition est géométrique. On peut voir sur la figure 4.3 que le point qui minimise $|x_1|$ dans $\Sigma \cap Q$ est obtenu à l'intersection de la droite délimitant la première équation du système (a') et la deuxième équation du système (b'), et inversement pour le point qui minimise $|x_2|$. Un exemple en trois dimension suggère alors la généralisation suivante : pour obtenir le point qui maximise x_k, on considère les n hyperplans formés des lignes du système (a') auxquelles on a ôté la $k^{\text{ème}}$, et de la ligne k du système (b'), c'est à dire de la ligne k du système (a) (puisque $Q_{kk} = -1$).

Si l'on se fie à cette intuition, la borne doit s'obtenir en considérant les n inégalités associées aux hyperplans que l'on a identifiés, c'est à dire

$$\left\{\begin{array}{rcl} [(I-\Delta)Qx]_1 & \leq & [Qb+\delta]_1 \\ & \vdots & \\ [(I-\Delta Q)x]_k & \leq & [b+\delta]_k \\ & \vdots & \\ [(I-\Delta)Qx]_n & \leq & [Qb+\delta]_n \end{array}\right. \tag{4.25}$$

Pour uniformiser les inéquations, il est aisé de voir que si on note Q^+ le quadrant symétrique de Q par rapport à $x_k = 0$, alors (4.25) se transforme de façon équivalente en :

$$(Q^+ - \Delta Q)x \leq Q^+b + \delta. \tag{4.26}$$

Pour de tels quadrant Q et Q^+, on a également $Q^+ = 2e_ke_k^T + Q$, d'où,

$$\begin{array}{lll} (4.26) & \Longrightarrow (2e_ke_k^T + Q - \Delta Q)x \leq Q^+b + \delta & \\ & \Longrightarrow 2e_ke_k^Tx + (I-\Delta)Qx \leq Q^+b + \delta & \\ & \Longrightarrow M2e_ke_k^Tx + Qx \leq M(Q^+b+\delta) = Q^+x_{Q^+} & (\text{car } M \geq 0) \\ & \Longrightarrow 2m_{kk}x_k - x_k \leq (Q^+x_{Q^+})_k & (\text{cf. section précédente}) \\ & \Longrightarrow (2m_{kk}-1)x_k \leq (Q^+x_{Q^+})_k & \\ & \Longrightarrow x_k \leq \nu_k(Q^+x_{Q^+})_k & (\text{d'après le lemme 4.4}) \end{array}$$

Il reste à vérifier que cette borne est optimale. Pour cela, il suffit de trouver un point de $\Sigma \cap Q$ dont la $k^{ème}$ composantes est $\nu_k\ (Q^+x_{Q^+})_k$. La deuxième difficulté est qu'il n'est pas possible de montrer l'optimalité en extrayant un point de Q directement de (4.26) comme cela a été fait avec le système (a') de la proposition 4.20. En effet, supposons que x soit solution de $(Q^+ - \Delta Q)x = Q^+b + \delta$. On trouve bien $x_k = \nu_k\ (Q^+x_{Q^+})_k$ mais on a également $x_i = (Qx_Q)_i$ pour $i \neq k$. Or, intuitivement, il est clair que le maximum de x_k ne s'atteint pas en un point où la norme de toutes les autres composantes se maximisent. Le point considéré ici n'appartient donc pas à Σ.

Rohn [Roh93] considère le point $Q^+M(Q^+b + \delta - 2\nu_k(Q^+x_{Q^+})_k\Delta e_k)$. La proposition suivante justifie le choix de ce point (nous expliquons dans la preuve comment ce point est obtenu).

Proposition 4.24
Soit Q un quadrant tel que $Q_{kk} = -1$, et Q^+ le quadrant symétrique de Q par rapport à $x_k = 0$ (on a donc $Q^+_{kk} = +1$).

Le point

$$x := Q^+M(Q^+b + \delta - 2\nu_k(Q^+x_{Q^+})_k\Delta e_k) \tag{4.27}$$

vérifie les trois points suivants :

1. $x \in Q \iff \Sigma \cap Q \neq \emptyset$,
2. $x_k = \nu_k\,(Q^+x_{Q^+})_k$,
3. $(I - Q^+\Delta Q)x = b + Q^+\delta$ *(c'est à dire $x \in \Sigma$).*

Preuve. Montrons tout d'abord comment est obtenu la formule pour x. D'après ce qui précède, le point x doit satisfaire le système suivant :

$$\begin{cases} [(I-\Delta)Qx]_1 &= [Qb+\delta]_1 \\ & \vdots \\ [(I-\Delta Q)x]_k &= [b+\delta]_k \\ & \vdots \\ [(I-\Delta)Qx]_n &= [Qb+\delta]_n \end{cases}$$

Or $(I-\Delta)Qx = (I-\Delta)(Q^+ - 2e_ke_k^T)x = (I-\Delta)Q^+x - 2(I-\Delta)e_ke_k^Tx = (I-\Delta)Q^+x - 2x_ke_k + 2x_k\Delta e_k$. Donc, pour $i \neq k$, le terme $2x_ke_k$ disparaît et

$$[(I-\Delta)Qx]_i = [(I-\Delta)Q^+x + 2x_k\Delta e_k]_i.$$

Cette formule s'étend à $i = k$ avec notre système, car :

$$(I-\Delta Q)x = x - \Delta(Q^+ - 2e_ke_k^T)x = x - \Delta Q^+x + 2\Delta e_ke_k^Tx = x - \Delta Q^+x + 2x_k\Delta e_k.$$

Comme $Q^+_{kk} = 1$, on a $[x - \Delta Q^+x + 2x_k\Delta e_k]_k = [(I-\Delta)Q^+x + 2x_k\Delta e_k]_k$ et $[b+\delta]_k = [Q^+b+\delta]_k$. De plus, $\forall i$ $(1 \leq i \leq n)$ $Q^+_{ii} = Q_{ii}$, Finalement, le système caractérisant le point x s'écrit de façon équivalente :

$$(I-\Delta)Q^+x = Q^+b + \delta - 2x_k\Delta e_k$$

Or on souhaite que $x_k = \nu_k\,(Q^+x_{Q^+})_k$, d'où l'expression pour x obtenue.

On commence la preuve par le deuxième point.

2-

$$\begin{aligned} (4.27) &\implies x = Q^+M(Q^+b+\delta) - Q^+M2\nu_k(Q^+x_{Q^+})_k\Delta e_k \\ &\implies Q^+x = Q^+x_{Q^+} - 2\nu_k(Q^+x_{Q^+})_kM\Delta e_k \end{aligned}$$

Puis, en utilisant le lemme 4.3 :

$$(4.27) \implies Q^+x = Q^+x_{Q^+} - 2\nu_k(Q^+x_{Q^+})_k(Me_k - e_k). \tag{4.28}$$

De là,

$$\begin{aligned} (4.27) &\implies x_k = (Q^+x_{Q^+})_k - 2\nu_k(Q^+x_{Q^+})_k(m_{kk}-1) \\ &\implies x_k = (Q^+x_{Q^+})_k - 2\tfrac{1}{2m_{kk}-1}(Q^+x_{Q^+})_k(m_{kk}-1) \\ &\implies x_k = (Q^+x_{Q^+})_k(1 - \tfrac{2}{2m_{kk}-1}(m_{kk}-1)) \\ &\implies x_k = (Q^+x_{Q^+})_k(\tfrac{1}{2m_{kk}-1}(2m_{kk} - 1 - 2m_{kk} + 2) \\ &\implies x_k = \nu_k(Q^+x_{Q^+})_k. \end{aligned}$$

1-

On a déjà $x_k \leq 0$ d'après **2-** (puisque $(Q^+x_{Q^+})_k \leq \tilde{x}_k \leq 0$, par hypothèse). Pour x_i avec $i \neq k$, on a, (4.28) $\implies Q^+_{ii}x_i = (Q^+x_{Q^+})_i - 2x_km_{ik}$. Or $(Q^+x_{Q^+})_i = (Qx_Q)_i \geq 0$ (sinon, il n'y a pas de solution dans Q), $x_k \leq 0$ et $m_{ik} \geq 0$ donc $Q^+_{ii}x_i \geq 0$, c'est à dire, $Q_{ii}x_i \geq 0$. On a donc bien $Qx \geq 0$, i.e., $x \in Q$.

3-

On a $I - Q^+\Delta Q = I - Q^+\Delta(Q^+ - 2e_ke_k^T) = (I - Q^+\Delta Q^+) - 2Q^+\Delta e_ke_k^T$

Or, d'une part,

$$\begin{aligned}(I - Q^+\Delta Q^+)(x) &= Q^+(I-\Delta)Q^+x \\ &= Q^+(I-\Delta)Q^+Q^+M(Q^+b+\delta-2x_k\Delta e_k) \\ &= Q^+(Q^+b+\delta-2\nu_k(Q^+x_{Q^+})_k\Delta e_k) \\ &= b+Q^+\delta-2Q^+x_k\Delta e_k \\ &= b+Q^+\delta-2Q^+x_k\Delta e_k.\end{aligned}$$

D'autre part, $2Q^+\Delta e_k e_k^T(x) = 2Q^+\Delta x_k e_k$. Ainsi, on obtient bien $(I - Q^+\Delta Q)x = b + Q^+\delta$.

□

Il en découle le calcul de borne suivant pour Σ :

Proposition 4.25 *Si $\Sigma \neq \emptyset$ et si pour tout $x \in \Sigma$ on a $x_k \leq 0$ alors $\max_{x\in\Sigma} x_k = \nu_k\ \tilde{x}_k$.*

Preuve. Pour tous les quadrants Q comportant des solutions on a $\max\{x_k,\ x \in \Sigma \cap Q\} = \nu_k(Q^+x_{Q^+})_k$. Il suffit ensuite de voir que $\tilde{x}_k = \max\{(Q^+x_{Q^+})_k, Q_{kk} = -1\}$. □

Le minimum de x_k pour $x \in \Sigma$ et $x_k \geq 0$, s'obtient de nouveau en considérant le système $\mathbf{A}x = -\mathbf{b}$. On peut également calculer cette borne en adaptant de façon symétrique les calculs précédents : l'équation (4.27) s'écrit $(Q^- - \Delta Q)x \leq (Q^-b + \delta)$, où Q est un quadrant vérifiant $Q_{kk} = +1$, et Q^- son symétrique par rapport à $x_k = 0$. On trouve également $x_k \leq -\nu_k(Q^-x_{Q^-})_k$ sachant que $(Q^-x_{Q^-})_k$ est bien négatif. Le point minimisant x_k est $Q^-M(Q^-b+\delta-2\nu_k(Q^-x_{Q^-})_k\Delta e_k)$, qui vérifie l'équation $(I-Q^-\Delta Q)x = b+Q^-\delta$. Ce point est solution du système suivant :

$$\left\{\begin{array}{rcl}[(I-\Delta)Qx]_1 &=& [Qb+\delta]_1 \\ &\vdots& \\ [(I+\Delta Q)x]_k &=& [b-\delta]_k \\ &\vdots& \\ [(I-\Delta)Qx]_n &=& [Qb+\delta]_n\end{array}\right.$$

En effet, $(I-\Delta)Qx = (I-\Delta)(Q^- + 2e_ke_k^T)x = (I-\Delta)Q^-x + 2(I-\Delta)e_ke_k^Tx = (I-\Delta)Q^-x + 2x_ke_k - 2x_k\Delta e_k$. Donc, pour $i \neq k$, le terme $2x_ke_k$ disparaît et

$$[(I-\Delta)Qx]_i = [(I-\Delta)Q^-x - 2x_k\Delta e_k]_i.$$

Cette formule s'étend ensuite à $i = k$, car

$$(I+\Delta Q)x = x + \Delta(Q^- + 2e_ke_k^T)x = x + \Delta Q^-x + 2\Delta e_ke_k^Tx = x + \Delta Q^-x + 2x_k\Delta e_k.$$

Comme $Q^-_{kk} = -1$, on a : $[x + \Delta Q^-x + 2x_k\Delta e_k]_k = [(-I+\Delta)Q^-x + 2x_k\Delta e_k]_k$, et $[b-\delta]_k = [-Q^-b-\delta]_k$. Donc $[I+\Delta Qx]_k = [b-\delta]_k \Longleftrightarrow [(I-\Delta)Q^-x - 2x_k\Delta e_k]_k = [Q^-b+\delta]_k$. De plus, $\forall i\ (1 \leq i \leq n)\ Q^-_{ii} = Q_{ii}$, donc le système caractérisant le point x s'écrit de façon équivalente :

$$(I-\Delta)Q^-x = Q^-b + \delta + 2x_k\Delta e_k.$$

On remplace ensuite x_k par $-\nu_k\ (Q^-x_{Q^-})_k$, d'où l'expression pour x obtenue.

La preuve de la proposition, s'adapte facilement :
– Le point **2-** se démontre strictement de la même manière.
– Le point **1-** s'obtient en écrivant $Q^-_{ii}x_i = (Q^-x_{Q^-})_i + 2x_km_{ik} \geq 0$.
– Le point **3-** s'obtient en écrivant $Q = Q^- + 2e_ke_k^T$.

Corollaire 4.3 *Si $\Sigma \neq \emptyset$ et si pour tout $x \in \Sigma$ on a $x_k \geq 0$ alors $\min_{x\in\Sigma} x_k = \nu_k\ \underline{x}_k$.*

Le théorème suivant synthétise les résultats de cette section.

Théorème 4.5 (Hansen-Bliek-Rohn) *Soit $\Delta \in \mathbb{R}^{n\times n}$ telle que $\Delta \geq 0$ et $\rho(\Delta) < 1$. Soient b et δ deux vecteurs de $\mathbb{R}^n$, avec $\delta \geq 0$. Pour tout k, $1 \leq k \leq n$, posons :*

$$\overline{x}_k := \max\{\tilde{x}_k, \nu_k \tilde{x}_k\},$$

$$\underline{x}_k := \min\{\underline{x}_k, \nu_k \underline{x}_k\}.$$

avec $\tilde{x}_k$, $\underline{x}_k$ et ν_k définis comme au corollaire 4.2 et au lemme 4.4. Alors, $\Sigma(I \pm \Delta, b \pm \Delta)$ est vide ssi $\underline{x}_k > \overline{x}_k$ pour une valeur de k. Sinon, la $k^{ème}$ projection de $\square\,\Sigma(I \pm \Delta, b \pm \Delta)$ est l'intervalle $[\underline{x}_k, \overline{x}_k]$.

Preuve. D'après la proposition 4.23, il existe une solution x qui vérifie $x_k \geq 0$ ssi $\tilde{x}_k \geq 0$, et $\tilde{x}_k$ est la borne sup de l'intervalle cherché. Comme $\tilde{x}_k \geq 0 \iff \nu_k \tilde{x}_k \leq \tilde{x}_k$ (d'après le lemme 4.4), $\overline{x}_k$ est bien la borne sup. S'il existe également une solution x avec $x_k \leq 0$, on trouve de même que la borne inf coïncide avec $\underline{x}_k$. Sinon, $\underline{x}_k \geq 0$, ce qui équivaut à $\nu_k \underline{x}_k \leq \underline{x}_k$. Or, d'après le corollaire 4.3, la borne inf est $\nu_k \underline{x}_k$, donc $\underline{x}_k$.

De même, s'il n'existe pas de solution x avec $x_k \geq 0$, on a $\tilde{x}_k \leq 0$ donc $\nu_k \tilde{x}_k \geq \tilde{x}_k$, et $\overline{x}_k$ est bien la borne sup, d'après la proposition 4.25.

Finalement, supposons $\underline{x}_k > \overline{x}_k$. Si Σ n'était pas vide, il rentrerait dans l'un des trois cas de figure ci-dessus, qui impliquent tous $\underline{x}_k \leq \overline{x}_k$, contradiction. $\square$

4.8 Conclusion

De nombreuses méthodes permettent l'approximation des solutions d'un système linéaire par intervalles. Nous en avons décrit certaines. D'autres méthodes, issues de l'analyse par intervalles ou de la programmation linéaire ont été conçues, par exemple [Rum83, Sha95, Bea97, Jan04]. Le problème paraît donc bien résolu.

Ces méthodes présentent toutefois un défaut en commun : elles ne tiennent pas compte de l'éventuelle corrélation entre les coefficients de $\mathbf{A}$ et $\mathbf{b}$. Le fait de considérer que chaque coefficient d'un système linéaire par intervalles varie indépendamment des autres introduit une approximation grossière dès le départ. Ceci est particulièrement vrai lorsque le système est obtenu à partir de l'opérateur de Newton, puisque la matrice est une matrice jacobienne pouvant impliquer de nombreuses occurrences de la même variable. Par exemple, si la jacobienne a la forme suivante :

$$J(x) := \begin{pmatrix} x & 1 \\ 1 & 1-x \end{pmatrix}$$

alors pour tout $x \in [0,1]$, la matrice $J(x)$ est régulière. Le remplacement de chaque coefficient de la matrice par un simple intervalle produit la matrice intervalle suivante

$$J(\mathbf{x}) := \begin{pmatrix} [0,1] & 1 \\ 1 & [0,1] \end{pmatrix}$$

qui n'est plus régulière. Aucune des méthodes ne s'applique donc. Il est clair que de nombreuses améliorations peuvent être apportées en exploitant la forme symbolique des jacobiennes. Une collaboration entre le calcul formel et les algorithmes par intervalles semble une voie prometteuse : l'étape de préconditionnement, par exemple, se prête bien à cette collaboration et des travaux ont déjà été menés sur le sujet [Kea02, MD06][8].

[8] Des méthodes performantes incorporant un préconditionnement symbolique ont été élaborées au sein de l'équipe [MD06] et testées avec succès sur des problèmes de robotique.

Chapitre 5

AE-systèmes linéaires

Sommaire

5.1 Introduction et définitions

Considérons une matrice $\mathbf{A} \in \mathbb{IR}^{n\times n}$ et un vecteur $\mathbf{b} \in \mathbb{IR}^n$. Au chapitre précédent, nous avons traité, à partir de ces données, le système linéaire $\mathbf{A}x = \mathbf{b}$ dont l'ensemble solution était le suivant :

$$\big(\exists A \in \mathbf{A}\big)\big(\exists b \in \mathbf{b}\big)\ Ax = b.$$

La matrice $\mathbf{A}$ et le vecteur $\mathbf{b}$ représentent ici un total de $(n^2 + n)$ paramètres quantifiés existentiellement. Par ailleurs, nous avons motivé dans le chapitre 1 l'intérêt d'introduire des quantificateurs universels pour des paramètres dans le cadre de systèmes d'équations en général, ce qui vaut en particulier pour les systèmes d'équations linéaires. De plus, nous avons montré au §3.4.5 p.69 que des systèmes linéaires avec paramètres quantifiés pouvaient servir à résoudre des problèmes non-linéaires de même nature.

Nous nous intéressons dans ce chapitres aux AE-systèmes linéaires, c'est à dire à toutes les manières possibles de quantifier les coefficients de $\mathbf{A}$ et de $\mathbf{b}$, en respectant l'ordre "A-E" qui consiste à placer en premier les paramètres universels dans les formules. Nous nous limitons toujours aux systèmes carrés. Voici pour commencer deux exemples canoniques :

$$\big(\forall A \in \mathbf{A}\big)\big(\exists b \in \mathbf{b}\big)\ Ax = b \quad (\textit{tolerable solution set},$$

$$\big(\forall b \in \mathbf{b}\big)\big(\exists A \in \mathbf{A}\big)\ Ax = b \quad (\textit{controllable solution set}.$$

Ces exemples sont particuliers puisque dans les deux cas, les coefficients de $\mathbf{A}$ partagent tous le même quantificateur (et de même pour $\mathbf{b}$). Or nous autorisons tout type de combinaison. Voici un exemple plus compliqué en dimension 3 :

Exemple 5.1 *Le système* $\mathbf{A}x = \mathbf{b}$ *en dimension 3 peut être quantifié de la façon suivante :*

$$\begin{pmatrix} \forall & \exists & \exists \\ \exists & \forall & \exists \\ \forall & \exists & \forall \end{pmatrix} x = \begin{pmatrix} \exists \\ \forall \\ \exists \end{pmatrix}.$$

Le AE-système correspondant est :

$$\begin{array}{l} \big(\forall A_{11} \in \mathbf{A}_{11}\big)\big(\forall A_{22} \in \mathbf{A}_{22}\big)\big(\forall A_{31} \in \mathbf{A}_{31}\big)\big(\forall A_{33} \in \mathbf{A}_{33}\big)\big(\forall b_2 \in \mathbf{b}_2\big) \\ \big(\exists A_{12} \in \mathbf{A}_{12}\big)\big(\exists A_{13} \in \mathbf{A}_{13}\big)\big(\exists A_{21} \in \mathbf{A}_{21}\big)\big(\exists A_{23} \in \mathbf{A}_{23}\big)\big(\exists A_{32} \in \mathbf{A}_{32}\big)\big(\exists b_1 \in \mathbf{b}_1\big)\big(\exists b_3 \in \mathbf{b}_3\big) \\ Ax = b. \end{array}$$

Pour des raisons d'homogénéité avec la littérature sur le sujet, nous ne manipulerons plus désormais un AE-système mais directement *l'ensemble des solutions d'un AE-système*, appelé *AE-solution set*.

Dans le cas général (dimension n), il est fastidieux d'écrire la formule d'un AE-système (c.a.d, la formule qui caractérise un AE-solution set) à partir de la donnée d'une matrice, d'un vecteur et de quantificateurs associé à chaque coefficient. La raison étant que l'ordre "A-E" impose de réordonner les coefficients dans la formule c'est à dire de gérer des indices, etc. Un langage commode proposé par Shary consiste à séparer $\mathbf{A}$ en deux matrices, l'une contenant les coefficients existentiels, l'autre universels. On sépare de même $\mathbf{b}$ en deux vecteurs, un pour chaque quantificateur. On aboutit à la définition suivante :

Définition 5.1 (AE-Solution set linéaire)
Soient $\mathbf{A}^\exists$ *et* $\mathbf{A}^\forall$ *deux matrices de* $\mathbb{IR}^{n\times n}$ *telles que*

$$\forall (i,j) \in [1..n] \times [1..n] \quad (\mathbf{A}^\exists)_{ij} = 0 \quad ou \quad (\mathbf{A}^\forall)_{ij} = 0,$$

De même, soient $\mathbf{b}^\exists$ *et* $\mathbf{b}^\forall$ *deux vecteurs de* $\mathbb{IR}^n$ *tels que*

$$\forall i \in [1..n] \quad (\mathbf{b}^\exists)_i = 0 \quad ou \quad (\mathbf{b}^\forall)_i = 0.$$

On appelle **AE-solution set linéaire** *de matrice existentiellement quantifiée* $\mathbf{A}^\exists$*, de matrice universellement quantifiée* $\mathbf{A}^\forall$*, de vecteur existentiellement quantifié* $\mathbf{b}^\exists$ *et de vecteur universellement quantifié* $\mathbf{b}^\forall$*, l'ensemble suivant*

$$\Sigma(\mathbf{A}^\exists, \mathbf{A}^\forall, \mathbf{b}^\exists, \mathbf{b}^\forall) = \{x \mid \big(\forall A^\forall \in \mathbf{A}^\forall\big)\big(\forall b^\forall \in \mathbf{b}^\forall\big)\big(\exists A^\exists \in \mathbf{A}^\exists\big)\big(\exists b^\exists \in \mathbf{b}^\exists\big) \quad (A^\forall + A^\exists)x = (b^\forall + b^\exists)\}.$$

Revenons à l'exemple 5.1. On effectue le découpage suivant :

$$\mathbf{A}^\forall := \begin{pmatrix} \mathbf{A}_{11} & 0 & 0 \\ 0 & \mathbf{A}_{22} & 0 \\ \mathbf{A}_{31} & 0 & \mathbf{A}_{33} \end{pmatrix} \quad \mathbf{A}^\exists := \begin{pmatrix} 0 & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & 0 & \mathbf{A}_{23} \\ 0 & \mathbf{A}_{32} & 0 \end{pmatrix}$$

$$\mathbf{b}^\forall := \begin{pmatrix} 0 \\ \mathbf{b}_2 \\ 0 \end{pmatrix} \quad \mathbf{b}^\exists := \begin{pmatrix} \mathbf{b}_1 \\ 0 \\ \mathbf{b}_3 \end{pmatrix},$$

en remarquant au passage que $\mathbf{A} = \mathbf{A}^\exists + \mathbf{A}^\forall$ et $\mathbf{b} = \mathbf{b}^\exists + \mathbf{b}^\forall$. Le AE-solution set donné dans l'exemple 5.1 est alors précisément $\Sigma(\mathbf{A}^\exists, \mathbf{A}^\forall, \mathbf{b}^\exists, \mathbf{b}^\forall)$.

Plutôt que d'avoir à manipuler deux matrices et deux vecteurs, l'idée est d'utiliser les intervalles généralisés pour pouvoir représenter à l'intérieur de la même matrice et du même vecteur les paramètres des deux types. Ainsi, on peut ne manipuler qu'une seule matrice $\mathbf{A} \in \mathbb{KR}^{n\times n}$, où les coefficients propres (resp. impropres) désignent les paramètres existentiellement (resp. universellement) quantifiés. De même pour $\mathbf{b}$. La convention choisie ici pour associer une qualité propre/impropre à un quantificateur est cohérente avec celle du chapitre 3 et différente de celle proposée initialement par Shary, principal instigateur de la théorie [Sha02]. Ce choix de convention permet une homogénéisation des notations et surtout des algorithmes [GC06b], comme il sera vu plus loin.

Définition 5.2 (Matrice & vecteur caractéristiques)
Soit $\Sigma(\mathbf{A}^\exists, \mathbf{A}^\forall, \mathbf{b}^\exists, \mathbf{b}^\forall)$ *un AE-solution set (selon la définition 5.1). Posons*

$$\mathbf{A}^c := \mathbf{A}^\exists + (\text{dual } \mathbf{A}^\forall),$$
$$\mathbf{b}^c := \mathbf{b}^\exists + (\text{dual } \mathbf{b}^\forall).$$

On appelle **AE-solution set linéaire** *de matrice caractéristique* $\mathbf{A}^c$ *et de vecteur caractéristique* $\mathbf{b}^c$*, et on note* $\Sigma(\mathbf{A}^c, \mathbf{b}^c)$*, l'ensemble*

$$\Sigma(\mathbf{A}^c, \mathbf{b}^c) := \Sigma(\mathbf{A}^\exists, \mathbf{A}^\forall, \mathbf{b}^\exists, \mathbf{b}^\forall).$$

Cette nouvelle définition, hormis le fait qu'elle constitue un langage commode pour décrire les AE-solution sets, permet d'utiliser toute la "mécanique" des intervalles généralisés, que nous devons tout d'abord étoffer de quelques définitions et propriétés supplémentaires. Nous en viendrons ainsi à la caractérisation fondamentale.

Terminons en signalant que dans la définition précédente, nous avons noté la matrice caractéristique $\mathbf{A}^c$ pour ne pas la confondre avec la matrice originale (c.a.d., $\mathbf{A}^\exists + \mathbf{A}^\forall$) notée jusqu'ici $\mathbf{A}$. Dorénavant, nous n'aurons plus recours à cette dernière matrice : tout système sera d'emblée décrit par son couple matrice/vecteur caractéristique. Nous noterons donc $\Sigma(\mathbf{A}, \mathbf{b})$ un AE-solution set linéaire, en gardant à l'esprit que $\mathbf{A}$ et $\mathbf{b}$ désignent désormais la matrice et le vecteur caractéristiques. Ce sont donc une matrices et un vecteur d'intervalles généralisés.

Nous pouvons redéfinir avec ces conventions les différents exemples de AE-solution sets exhibés précédemment. Le AE-solution set correspondant au cas classique porte le nom de *united* solution set.

Exemples de AE-solution sets $\Sigma(\mathbf{A}, \mathbf{b})$

United solution set	$\mathbf{A}$ propre	$\mathbf{b}$ propre
Tolerable solution set	$\mathbf{A}$ impropre	$\mathbf{b}$ propre
Controllable solution set	$\mathbf{A}$ propre	$\mathbf{b}$ impropre

5.1.1 Compléments sur les intervalles généralisés

Une propriété fondamentale de l'arithmétique de Kaucher que nous avons occultée jusqu'à ce stade est la suivante :

Proposition 5.1 *Toutes les opérations arithmétiques*[1] *dans* $\mathbb{KR}$ *(définies au §3.2 p.57) sont monotones pour l'inclusion dans* $\mathbb{KR}$ *(définie au §3.3 p.54).*

Exemples :

$$\mathbf{x} \subseteq \mathbf{y} \Longrightarrow -\mathbf{x} \subseteq -\mathbf{y}. \tag{5.1}$$

$$\mathbf{x} \subseteq \mathbf{y} \Longrightarrow \mathbf{x} + \mathbf{a} \subseteq \mathbf{y} + \mathbf{a}. \tag{5.2}$$

Nous aurons également recours à quelques règles de « passage au dual ». Pour tout opérateur $\star \in \{+, -, \times, /\}$

$$\text{dual}\,(\mathbf{x} \star \mathbf{y}) = \text{dual}\,(\mathbf{x}) \star \text{dual}\,(\mathbf{y}),$$

en excluant bien sûr le cas $0 \in \text{pro}\,(\mathbf{y})$ pour la division. Ce passage au dual peut s'étendre donc à la somme et au produit de matrices. Si $\mathbf{A}$ et $\mathbf{B}$ sont deux matrices,

$$\text{dual}\,(\mathbf{A} + \mathbf{B}) = \text{dual}\,(\mathbf{A}) + \text{dual}\,(\mathbf{B}), \tag{5.3}$$

$$\text{dual}\,(\mathbf{AB}) = \text{dual}\,(\mathbf{A}) \times \text{dual}\,(\mathbf{B}). \tag{5.4}$$

[1] Ces opérations n'incluent pas dual , pro et imp .

La règle d'associativité suivante continue d'être vraie pour un vecteur dégénéré x :

$$(\mathbf{AB})x = \mathbf{A}(\mathbf{B}x). \tag{5.5}$$

On montre également aisément (par exemple à l'aide de la proposition 3.3 p.59), en remarquant bien que x est un vecteur d'intervalles dégénéré :

$$\big(\forall \mathbf{A} \in \mathbb{KR}^{m\times n}\big)\big(\forall x \in \mathbb{R}^n\big) \quad (\text{dual } \mathbf{A})x = \text{dual}\,(\mathbf{A}x). \tag{5.6}$$

Enfin, la distributivité se décline de trois façons. Pour $\mathbf{a}$, $\mathbf{b}$, $\mathbf{x}$ dans $\mathbb{KR}$:

$$(\mathbf{a}+\mathbf{b})\mathbf{x} \subseteq \mathbf{ax}+\mathbf{bx} \quad \text{si } \mathbf{x} \text{ est propre,} \tag{5.7}$$
$$(\mathbf{a}+\mathbf{b})\mathbf{x} \supseteq \mathbf{ax}+\mathbf{bx} \quad \text{si } \mathbf{x} \text{ est impropre,} \tag{5.8}$$
$$(\mathbf{a}+\mathbf{b})\mathbf{x} = \mathbf{ax}+\mathbf{bx} \quad \text{si } \mathbf{x} \text{ est dégénéré.} \tag{5.9}$$

Remarquons que (5.9) se déduit de (5.7) et (5.8).

5.1.2 Caractérisation fondamentale des AE-solution sets linéaires

Nous énonçons ici un moyen de caractériser un point appartenant à un AE-solution set linéaire par une simple inclusion entre intervalles généralisés. Cette inclusion servira ensuite de base aux algorithmes d'approximation d'AE-solution sets présentés dans les paragraphes suivants (à l'exception de Hansen-Bliek généralisé).

Proposition 5.2 (Shary)

$$x \in \Sigma(\mathbf{A},\mathbf{b}) \iff 0 \in \mathbf{A}x-\mathbf{b} \iff (\text{dual } \mathbf{A})x \subseteq \mathbf{b},$$

où les opérations arithmétiques et l'inclusion se font dans $\mathbb{KR}^n$.

Preuve. Soient $\mathbf{A}^\exists$, $\mathbf{A}^\forall$, $\mathbf{b}^\exists$ et $\mathbf{b}^\forall$ définis à partir de $\mathbf{A}$ et $\mathbf{b}$ comme à la définition 5.2. Nous proposons deux preuves. Tout d'abord, cette proposition peut être vue comme un corollaire des résultats de la section 3.3, et énoncée comme un cas particulier du lemme 3.4.1 p.64. Sauf que, contrairement à ce lemme, nous obtenons une équivalence et non une simple implication. Ceci tient au fait que la boîte $\mathbf{x}$ est dégénérée (on cherche à caractériser une solution, au lieu d'une boîte intérieure) et que, grâce à la linéarité, "évaluation aux intervalles généralisés" coïncide avec "image quantifiée' (on applique la proposition 3.7 p. 63). Voici la preuve correspondante. Nous notons A_i la $i^{ème}$ ligne d'une matrice A (cette notation n'est valable que pour cette preuve). En décomposant ligne par ligne la relation $0 \in \mathbf{A}x-\mathbf{b}$, on obtient

$$\begin{array}{lll}
0 \subseteq \mathbf{A}x-\mathbf{b} & \iff & 0 \subseteq \mathbf{A}^\exists x + (\text{dual } \mathbf{A}^\forall)x - \mathbf{b}^\exists - (\text{dual } \mathbf{b}^\forall) \\
& \iff \forall i \in [1..n] & 0 \subseteq \mathbf{A}_i^\exists x + (\text{dual } \mathbf{A}^\forall)_i x - \mathbf{b}_i^\exists - (\text{dual } \mathbf{b}^\forall)_i \\
& \iff \forall i \in [1..n] & 0 \subseteq \bigwedge_{A_i^\forall \in \mathbf{A}_i^\forall} \bigwedge_{b_i^\forall \in \mathbf{b}_i^\forall} \bigvee_{A_i^\exists \in \mathbf{A}_i^\exists} \bigvee_{b_i^\exists \in \mathbf{b}_i^\exists} A_i^\exists x + A_i^\forall x - b_i^\exists - b_i^\forall \quad \text{d'après la prop. 3.7} \\
& \iff \forall i \in [1..n] & 0 \subseteq \bigwedge_{A_i^\forall \in \mathbf{A}_i^\forall} \bigwedge_{b_i^\forall \in \mathbf{b}_i^\forall} \bigvee_{A_i^\exists \in \mathbf{A}_i^\exists} \bigvee_{b_i^\exists \in \mathbf{b}_i^\exists} (A^\exists x + A^\forall x - b^\exists - b^\forall)_i \\
& \iff \forall i \in [1..n] & \big(\forall A_i^\forall \in \mathbf{A}_i^\forall\big)\big(\forall b_i^\forall \in \mathbf{b}_i^\forall\big)\big(\exists A_i^\exists \in \mathbf{A}_i^\exists\big)\big(\exists b_i^\exists \in \mathbf{b}_i^\exists\big) \\
& & (A^\exists x + A^\forall x - b^\exists - b^\forall)_i = 0 \\
& \iff & \big(\forall A^\forall \in \mathbf{A}^\forall\big)\big(\forall b^\forall \in \mathbf{b}^\forall\big)\big(\exists A^\exists \in \mathbf{A}^\exists\big)\big(\exists b^\exists \in \mathbf{b}^\exists\big) \quad A^\exists x + A^\forall x - b^\exists - b^\forall = 0 \\
& \iff & x \in \Sigma(\mathbf{A}^\exists, \mathbf{A}^\forall, \mathbf{b}^\exists, \mathbf{b}^\forall) \\
& \iff & x \in \Sigma(\mathbf{A},\mathbf{b}).
\end{array}$$

□

Shary propose également une preuve directe :

Preuve. [Sha02]

$$\begin{array}{rll} x \in \Sigma(\mathbf{A},\mathbf{b}) & \iff & x \in \Sigma(\mathbf{A}^{\exists},\mathbf{A}^{\forall},\mathbf{b}^{\exists},\mathbf{b}^{\forall}) \\ & \iff & \big(\forall A^{\forall} \in \mathbf{A}^{\forall}\big)\big(\forall b^{\forall} \in \mathbf{b}^{\forall}\big)\big(\exists A^{\exists} \in \mathbf{A}^{\exists}\big)\big(\exists b^{\exists} \in \mathbf{b}^{\exists}\big) \quad (A^{\forall}+A^{\exists})x = (b^{\forall}+b^{\exists}) \\ & \iff & \big(\forall A^{\forall} \in \mathbf{A}^{\forall}\big)\big(\forall b^{\forall} \in \mathbf{b}^{\forall}\big)\big(\exists A^{\exists} \in \mathbf{A}^{\exists}\big)\big(\exists b^{\exists} \in \mathbf{b}^{\exists}\big) \quad A^{\forall}x - b^{\forall} = -A^{\exists}x + b^{\exists} \\ & \iff & \mathbf{A}^{\forall}x - \mathbf{b}^{\forall} \subseteq -\mathbf{A}^{\exists}x + \mathbf{b}^{\exists}, \end{array}$$

car du fait de l'absence d'occurrences multiples, $\mathbf{A}^{\forall}x - \mathbf{b}^{\forall} = \{A^{\forall}x - b^{\forall}, A^{\forall} \in \mathbf{A}^{\forall}, b^{\forall} \in \mathbf{b}^{\forall}\}$, et $\mathbf{A}^{\exists}x - \mathbf{b}^{\exists} = \{A^{\exists}x - b^{\exists}, A^{\exists} \in \mathbf{A}^{\exists}, b^{\exists} \in \mathbf{b}^{\exists}\}$ (les composantes de x apparaissent plusieurs fois dans l'expression, mais elles sont réduites à un réel). On a donc

$$\begin{array}{rll} x \in \Sigma(\mathbf{A},\mathbf{b}) & \iff & \mathbf{A}^{\forall}x + (\text{dual } \mathbf{A}^{\exists})x \subseteq \mathbf{b}^{\exists} + (\text{dual } \mathbf{b}^{\forall}) \quad (\text{car dual}\,(\mathbf{A}^{\exists}x) = (\text{dual } \mathbf{A}^{\exists})x) \\ & \iff & (\text{dual}\,((\text{dual } \mathbf{A}^{\forall}) + \mathbf{A}^{\exists}))x \subseteq \mathbf{b}^{\exists} + (\text{dual } \mathbf{b}^{\forall}) \\ & \iff & (\text{dual } \mathbf{A})x \subseteq \mathbf{b} \end{array}$$

Il reste enfin à prouver :

$$(\text{dual } \mathbf{A})x \subseteq \mathbf{b} \iff 0 \subseteq \mathbf{A}x - \mathbf{b}.$$

On écrit :

$$\begin{array}{rll} (\text{dual } \mathbf{A})x \subseteq \mathbf{b} & \iff & \text{dual}\,(\mathbf{A}x) \subseteq \mathbf{b} \qquad (\text{grâce à (5.6)}). \\ & \iff & -\text{dual}\,(\mathbf{A}x) \subseteq -\mathbf{b} \quad (\text{grâce à (5.1)}) \\ & \iff & 0 \subseteq \mathbf{A}x - \mathbf{b} \qquad (\text{grâce à (5.2)}) \end{array}$$

□

5.2 Approche formelle pour l'approximation intérieure

La proposition 5.2 permet de donner une condition suffisante (mais en général non nécessaire) pour qu'une boîte soit intérieure à un AE-solution set $\Sigma(\mathbf{A},\mathbf{b})$. Cette proposition est rendue efficace en pratique moyennant un préconditionnement « à droite » du système [Gol05a].

Proposition 5.3 *Soit* $\mathbf{A} \in \mathbb{KR}^{n\times n}$, $\mathbf{b} \in \mathbb{KR}^{n}$. *Si* $\mathbf{x}$ *est un vecteur propre tel que*

$$(\text{dual } \mathbf{A})\mathbf{x} = \mathbf{b},$$

alors $\mathbf{x}$ *est une boîte intérieure de* $\Sigma(\mathbf{A},\mathbf{b})$.

Preuve. Si $(\text{dual } \mathbf{A})\mathbf{x} = \mathbf{b}$, alors par monotonie pour l'inclusion, $\forall x \in \mathbf{x}$, $(\text{dual } \mathbf{A})x \subseteq \mathbf{b}$, i.e., x est solution d'après la proposition 5.2. Remarquons qu'il est correct d'écrire "$x \in \mathbf{x}$" car $\mathbf{x}$ est propre. □

Avant de s'interroger sur la façon de résoudre une telle équation, remarquons qu'il aurait été possible, et même quelque part plus simple, de dire qu'il suffisait pour que $\mathbf{x}$ soit une boîte intérieure que $(\text{dual } \mathbf{A})\mathbf{x} \subseteq \mathbf{b}$. Considérer une égalité au lieu d'une inclusion permet en fait de ne s'intéresser qu'aux boîtes intérieures maximales, c'est à dire, non extensibles. Informellement, le vecteur le plus grand inclus dans $\mathbf{b}$ est bien $\mathbf{b}$ lui-même, donc si $\mathbf{x}$ est solution de $(\text{dual } \mathbf{A})\mathbf{x} = \mathbf{b}$, on s'attend à ce que

$$\mathbf{y} \supsetneq \mathbf{x} \Longrightarrow \exists y \in \mathbf{y} \mid (\text{dual } \mathbf{A})y \not\subseteq \mathbf{b}. \tag{5.10}$$

Ce raisonnement est toutefois approximatif, pour deux raisons. Reprenons-le pas-à-pas.

Soit $\mathbf{x}$ tel que $(\text{dual } \mathbf{A})\mathbf{x} = \mathbf{b}$, et posons $\mathbf{y} \supsetneq \mathbf{x}$. Il n'est pas évident que

$$\mathbf{y} \supsetneq \mathbf{x} \Longrightarrow (\text{dual } \mathbf{A})\mathbf{y} \supsetneq (\text{dual } \mathbf{A})\mathbf{x}. \tag{5.11}$$

Or clairement, nous avons besoin que cela soit vrai pour conclure (dual $\mathbf{A}$)$\mathbf{y} \supseteq \mathbf{b}$, c.a.d. que $\mathbf{x}$ est "maximal". Cette implication n'est pas toujours vraie, et lorsqu'elle l'est, nous parlons de *stricte monotonie pour l'inclusion*. Les conditions de stricte monotonie ont été étudiées par Sharaya [Sha01].

Admettons néanmoins que (5.11) soit vraie. Il n'est de nouveau pas évident que

$$(\text{dual } \mathbf{A})\mathbf{y} \supseteq \mathbf{b} \Longrightarrow \exists y \in \mathbf{y} \mid (\text{dual } \mathbf{A})y \not\subseteq \mathbf{b}, \tag{5.12}$$

du fait du problème de dépendance. Or, nous avons de nouveau besoin de cette implication pour conclure que $\mathbf{y}$ n'est pas intérieure. Shary a prouvé que cette implication était vraie (cf. [Sha02] p.380). Intuitivement, le problème de dépendance n'est pas *suffisant* : une boîte $\mathbf{b}$ strictement incluse dans (dual $\mathbf{A}$)$\mathbf{y}$ peut ne pas entièrement contenir tous les vecteurs (dual $\mathbf{A}$)y pour $y \in \mathbf{y}$. On en conclut :

Si (5.11) *est vraie alors* (5.10) *est vraie.*

Ceci s'exprime de façon équivalente en :

Si $\mathbf{x}$ *est un vecteur d'intervalles propres vérifiant* $\mathbf{Ax} = \mathbf{b}$ *"maximal",*
alors $\mathbf{x}$ *est une boîte intérieure non extensible.*

Concernant maintenant la résolution du système $\mathbf{Ax} = \mathbf{b}$, nous renvoyons à l'article de Shary [Sha02] p.377 et aux nombreuses références qui s'y trouvent. Voici toutefois un bref survol du problème. Tout d'abord il a été prouvé dans le cas général comme étant NP-difficile [Lak96]. La résolution est cependant rendue possible en pratique sur de nombreuses instances grâce notamment à des méthodes de point fixe [Gol05a] ou des méthodes dites de "Newton sous-différentiel" [Sha02]. Ces dernières nécessitent d'exprimer l'arithmétique de Kaucher autrement que sous forme de tables (cf. la définition de la multiplication au §3.2 p.57) puisque ces tables, de par leur caractère morcelé, ne rendent pas aisé le calcul de dérivées. Ce sont les formules de Lakeyev : la multiplication s'écrit ainsi avec une seule formule, dès lors que la qualité propre/impropre des opérandes est connue (la formule fait intervenir toutefois des min et des max entre les bornes des intervalles).

5.3 Gauss-Seidel généralisé

L'opérateur de Gauss-Seidel introduit au §4.4 p.90 se généralise pour permettre l'approximation extérieure de n'importe quel AE-solution set (au lieu de se limiter aux *united solution sets*). Nous poursuivons donc cette section en nous plaçant dans le cadre plus général des AE-systèmes linéaires.

Remarque 5.1 *Shary a mis au point l'opérateur de Gauss-Seidel généralisé en s'intéressant directement à la version contractante. Inversement, nous verrons au paragraphe suivant qu'il a aussi adapté l'opérateur de Krawczyk en s'intéressant uniquement à la version point fixe. Une version contractante de Krawczyk est proposée dans [GC06b], et tout porte à croire que Gauss-Seidel s'adapte également dans sa version point fixe en hybridant les preuves (version point fixe) de Gauss-Seidel classique et de Krawczyk généralisé. Néanmoins, ce travail reste à faire et n'est pas présenté dans cette thèse.*

D'après la caractérisation fondamentale des AE-solutions sets linéaires,

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \iff 0 \in (\text{dual } \mathbf{A})x \subseteq \mathbf{b}.$$

Cette inclusion se décompose ligne par ligne en :

$$(\sum_{j=1}^{n} \text{dual } \mathbf{A}_{ij}) \cdot x \subseteq \mathbf{b}_i \tag{5.13}$$

Grâce à la distributivité (5.9),

$$(5.13) \iff (\text{dual } \mathbf{A}_{ii})x_i + \sum_{j\neq i}(\text{dual } \mathbf{A}_{ij})x_j \subseteq \mathbf{b}_i.$$

Grâce à (5.6),

$$(5.13) \iff (\text{dual } \mathbf{A}_{ii})x_i + \sum_{j\neq i}\text{dual}\,(\mathbf{A}_{ij}x_j) \subseteq \mathbf{b}_i.$$

Ajoutons de part et d'autre l'opposé de dual $(\mathbf{A}_{ij}x_j)$ (c'est à dire $\mathbf{A}_{ij}x_j$) pour $j \neq i$. La propriété de groupe permet une simplification à gauche, et l'inclusion est conservée en vertu de (5.1). Ainsi,

$$(5.13) \iff (\text{dual } \mathbf{A}_{ii})x_i \subseteq \mathbf{b}_i - \sum_{j\neq i}\mathbf{A}_{ij}x_j.$$

Finalement, en multipliant de part et d'autre par l'inverse de (dual $\mathbf{A}_{ii}$), les mêmes arguments de simplification et de monotonie pour l'inclusion permettent d'écrire

$$(5.13) \iff x_i \subseteq \frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{j\neq i}\mathbf{A}_{ij}x_j).$$

Prenons maintenant $\mathbf{x}$, une boîte quelconque. Si en plus d'appartenir à $\Sigma(\mathbf{A},\mathbf{b})$, on souhaite que x appartienne à $\mathbf{x}$ alors, par monotonie, l'inclusion continue d'être valide en remplaçant x_j par $\mathbf{x}_j$. On en conclut

$$x \in \Sigma(\mathbf{A},\mathbf{b}) \cap \mathbf{x} \Longrightarrow x \subseteq \frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{j\neq i}\mathbf{A}_{ij}\mathbf{x}_j).$$

Dès lors que $\Sigma(\mathbf{A},\mathbf{b}) \neq \emptyset$, ceci implique en particulier que $\frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{j\neq i}\mathbf{A}_{ij}\mathbf{x}_j)$ est propre, bien que des intervalles impropres puissent intervenir dans les calculs. A partir de cette dernière relation, on établit ainsi l'itération de **Gauss-Seidel généralisée** $(x^{(n)})_{n\geq 0}$ -version contractante- qui produit une suite décroissante de boîtes extérieures de $\Sigma(\mathbf{A},\mathbf{b}) \cap \mathbf{x}^{(0)}$:

$$\forall i \in [1..n] \quad \mathbf{x}_i^{(k+1)} := \frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{j\neq i}\mathbf{A}_{ij}\mathbf{x}_j^{(k)}) \cap \mathbf{x}^{(k)}, \tag{5.14}$$

en précisant que l'itération s'arrête[2] dès qu'un intervalle impropre apparaît au moment de l'intersection. Puisque la présence d'un intervalle impropre implique que $\mathbf{x}^{(0)}$ ne contient aucune solution, il est en effet possible de conclure immédiatement que $\Sigma \cap \mathbf{x}^{(0)} = \emptyset$. L'intersection est donc correcte puisqu'elle ne met en jeu que des intervalles propres. En résumé, on a prouvé que la proposition 4.16 p.95 peut être généralisée à $\mathbf{A} \in \mathbb{KR}^{n\times n}$ et $\mathbf{b} \in \mathbb{KR}^n$:

Proposition 5.4 *Soit $\mathbf{x} \in \mathbb{IR}^n$. Soit $\mathbf{x}^{(k)}$ défini par l'itération* (5.14) *calculée à partir de $\mathbf{x}^{(0)} := \mathbf{x}$. Alors*

$$\forall k > 0, \quad \Sigma(\mathbf{A},\mathbf{b}) \cap \mathbf{x} \subseteq \mathbf{x}^{(k)}.$$

Dans le cas où l'on souhaite une boîte extérieure de $\Sigma(\mathbf{A},\mathbf{b})$ lui-même, l'itération de Gauss-Seidel nécessite une estimée initiale, c'est à dire une boîte $\mathbf{x}$ qui soit elle-même déjà une boîte extérieure. Il est possible alors de prendre une approximation de $\Sigma(\text{pro }\mathbf{A}, \text{pro }\mathbf{b})$, qui est un *united solution set* contenant[3] $\Sigma(\mathbf{A},\mathbf{b})$, en utilisant l'une des méthodes du chapitre 4.

Remarque 5.2 *Le changement proposé dans [Gol05b, GC06b] sur la manière de construire la matrice caractéristique a permis d'étendre l'algorithme de Gauss-Seidel, jusqu'ici décliné différemment pour les intervalles classiques et généralisés.*

[2] Rigoureusement, on peut écrire par exemple $\mathbf{x}^{(n)} = \emptyset$ pour tous les n qui suivent.

[3] Cette inclusion peut se voir directement à partir des formules quantifiées (car remplacer le quantificateur d'un paramètre universel par $\exists$ affaiblit la relation, donc étend l'ensemble des solutions), ou en écrivant simplement à l'aide de la caractérisation fondamentale $\mathbf{A}x \subseteq \mathbf{b} \Longrightarrow (\text{pro }\mathbf{A})x \subseteq (\text{pro }\mathbf{b})$.

5.4 Krawczyk généralisé

Encore une fois, l'idée est tout à fait similaire à ce qui a été exposé au §4.5 p.95. Si $x \in \Sigma(\mathbf{A}, \mathbf{b})$, notre but est d'établir l'inclusion suivante, qui *doit* effectivement être vraie au moins pour les *united solution sets* (i.e., lorsque $\mathbf{A} \in \mathbb{IR}^{n\times n}$), puisque cela a été prouvé au début du §4.5 :

$$x \subseteq (I - \mathbf{A})x + \mathbf{b}.$$

Proposition 5.5

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \iff x \subseteq (I - \mathbf{A})x + \mathbf{b}.$$

Preuve. D'après la caractérisation fondamentale des AE-solution sets linéaires :

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \iff 0 \subseteq \mathbf{A}x - \mathbf{b}.$$

Il s'ensuit, par monotonie pour l'inclusion dans $\mathbb{KR}^n$:

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \iff x \subseteq x - \mathbf{A}x + \mathbf{b}.$$

Il suffit alors d'appliquer la règle de distributivité (5.9) :

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \iff x \subseteq (I - \mathbf{A})x - \mathbf{b}.$$

□

On prouve de même l'existence d'un point fixe. Dans la proposition suivante, ρ désigne le rayon spectral (cf. 4.1.2 p.81).

Proposition 5.6 (Shary-Krawczyk) *Si $\rho(|I - \mathbf{A}|) < 1$, alors la suite suivante :*

$$\mathbf{x}^{(n+1)} := (I - \mathbf{A})\mathbf{x}^{(n)} + \mathbf{b}$$

converge vers un unique point fixe $\mathbf{x}^*$ quelle que soit l'estimée initiale $\mathbf{x}^{(0)}$. Ce point fixe vérifie

$$\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}^*.$$

Si de plus $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}^{(0)}$, alors $\big(\forall n \geq 0\big)$, $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}^{(n)}$. Ceci implique en particulier que si pour une valeur de $n \geq 0$, l'une des composantes de $\mathbf{x}^{(n)}$ est impropre, alors $\Sigma(\mathbf{A}, \mathbf{b})$ est vide.

Preuve. (schéma) En considérant la fonction de $\mathbb{KR}^n \to \mathbb{KR}^n$ suivante $f : \mathbf{x} \mapsto (I - \mathbf{A})\mathbf{x} + \mathbf{b}$, on établit la relation suivante à partir de la définition de la pseudo-distance dans $\mathbb{KR}^n$:

$$\operatorname{dist}(f(\mathbf{x}), f(\mathbf{y})) \leq |I - \mathbf{A}| \operatorname{dist}(\mathbf{x}, \mathbf{y}),$$

qui implique, grâce au théorème de Schröder (dans une version plus générale que celle exposée au chapitre 4) l'existence et l'unicité d'un point fixe $\mathbf{x} = f(\mathbf{x})$. Pour le reste, il suffit de montrer (en utilisant la monotonie pour l'inclusion) que si un point solution x appartient à une itération, il appartient aux suivantes. Il appartient donc au point fixe. Mais le point étant unique, x appartient à $\mathbf{x}^*$ quelle que soit l'estimée initiale. □

Le fait de savoir que $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}^*$ apparaît a priori comme un résultat théorique sans utilité pratique (notamment parce qu'on ne peut jamais être sûr d'atteindre le point fixe en un nombre fini d'itérations). Toutefois, il a un intérêt grâce à l' "approche formelle". Dans la section précédente, nous avons montré que la solution de l'équation intervalle (dual $\mathbf{A}$)$\mathbf{x} = \mathbf{b}$ était une approximation intérieure de $\Sigma(\mathbf{A}, \mathbf{b})$. Nous venons de montrer que $\mathbf{x}^*$, c.a.d. la solution de l'équation intervalle $\mathbf{x} = (I - \mathbf{A})\mathbf{x} + \mathbf{b}$, est une approximation extérieure $\Sigma(\mathbf{A}, \mathbf{b})$. Nous pouvons donc utiliser les mêmes techniques de résolution (Newton sous-différentiel) pour calculer une solution de l'équation $\mathbf{x} = (I - \mathbf{A})\mathbf{x} + \mathbf{b}$. Cette approche, plus compliquée, nous donne toutefois la meilleure approximation extérieure possible calculable avec l'itération de Shary-Krawczyk !

5.5 Méthode exhaustive

Les *united solution sets* ont été ramenés à des systèmes de programmation linéaire au §2.7.1 p.30. Il est possible d'en faire de même pour les AE-solution sets linéaires. Contrairement aux méthodes des paragraphes précédents, cela permet d'éviter ainsi l'arithmétique des intervalles généralisés et de travailler directement avec des simplexes. On démontre que le théorème d'Oettli-Prager p.31 peut être généralisé aux matrices $\mathbf{A}$ de $\mathbb{KR}^{n\times n}$ et aux vecteurs $\mathbf{b}$ de $\mathbb{KR}^n$. L'énoncé choisi s'appuie sur la définition suivante :

Définition 5.3 *Soit* $\mathbf{x} \in \mathbb{KR}$

On appelle **milieu** *de* $\mathbf{x}$ *et on note* $(\text{mid } \mathbf{x})$ *le réel* $\quad (\text{mid } \mathbf{x}) := \frac{1}{2}(\overline{\mathbf{x}} + \underline{\mathbf{x}})$.

On appelle **rayon** *de* $\mathbf{x}$ *et on note* $(\text{rad } \mathbf{x})$ *le réel* $\quad (\text{rad } \mathbf{x}) := \frac{1}{2}(\overline{\mathbf{x}} - \underline{\mathbf{x}})$.

Ces définitions s'étendent aux vecteurs et aux matrices composante par composante.

Exemple :

$$\mathbf{A} = \begin{pmatrix} [0,2] & [3,1] \\ [1,-1] & [-1,1] \end{pmatrix} \quad (\text{mid } \mathbf{A}) = \begin{pmatrix} 1 & 2 \\ 0 & 0 \end{pmatrix} \quad (\text{rad } \mathbf{A}) = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

Proposition 5.7 (Shary-Rohn-Oettli-Prager)
Soient $A := \text{mid } (\mathbf{A})$, $\Delta := \text{rad } (\mathbf{A})$, $b := \text{mid } (\mathbf{b})$, $\delta := \text{rad } (\mathbf{b})$, Q *un quadrant.*

$$x \in \Sigma(\mathbf{A}, \mathbf{b}) \cap Q \Longleftrightarrow \begin{cases} (A - \Delta Q)x \leq b + \delta \\ (A + \Delta Q)x \geq b - \delta \\ Qx \geq 0 \end{cases} \tag{5.15}$$

Il est important de noter qu'à la différence de la proposition 2.9 p.31, Δ et δ peuvent avoir ici des composantes **négatives**.

Preuve. Cette preuve ressemble à celle faite au §2.7.1. Rappelons que l'inclusion dans $\mathbb{KR}$ s'écrit $\mathbf{x} \subseteq \mathbf{y} \iff \underline{\mathbf{x}} \geq \underline{\mathbf{y}}$ et $\overline{\mathbf{x}} \leq \overline{\mathbf{y}}$. Par conséquence, la caractérisation fondamentale se décompose :

$$(\text{dual } \mathbf{A})x \subseteq \mathbf{b} \iff \begin{cases} \inf((\text{dual } \mathbf{A})x) \geq \underline{\mathbf{b}}, \\ \sup((\text{dual } \mathbf{A})x) \leq \overline{\mathbf{b}}. \end{cases}$$

Il est crucial d'avoir à l'esprit qu'ici « inf » désigne la borne inférieure d'un intervalle, et non nécessairement sa plus petite valeur ! Prenons la première inclusion, $\inf((\text{dual } \mathbf{A})x) \geq \underline{\mathbf{b}}$, et décomposons-la ligne par ligne. Par définition de l'addition dans $\mathbb{KR}$, on a $\inf\Big((\text{dual } \mathbf{A})x\Big)_i = \sum_{j=1}^{n} \inf\Big((\text{dual } \mathbf{A}_{ij}x_j\Big)$. Comme la multiplication d'un intervalle de $\mathbb{KR}$ par un nombre négatif a pour effet d'inverser les bornes, on a :

$$\inf\Big((\text{dual } \mathbf{A}_{ij})x_j\Big) = \begin{cases} \inf(\text{dual } \mathbf{A}_{ij})x_j & \text{si } x_j > 0, \\ \sup(\text{dual } \mathbf{A}_{ij})x_j & \text{sinon.} \end{cases}$$

Donc, par définition du dual :

$$\inf((\text{dual } \mathbf{A}_{ij})x_j) = \begin{cases} \overline{\mathbf{A}}_{ij}x_j & \text{si } x_j > 0, \\ \underline{\mathbf{A}}_{ij}x_j & \text{sinon.} \end{cases}$$

Or, d'après la définition 5.3 ci-dessus, on a $\overline{\mathbf{A}} = A + \Delta$ et $\mathbf{A} = A - \Delta$, d'où

$$\inf((\text{dual } \mathbf{A}_{ij})x_j) = \begin{cases} (A+\Delta)_{ij}x_j & \text{si } x_j > 0, \\ (A-\Delta)_{ij}x_j & \text{sinon.} \end{cases}$$

Finalement, on trouve bien

$$\inf((\text{dual } \mathbf{A})x) = (A + Q\Delta)x,$$

où Q désigne le quadrant de x. On peut montrer de même $\sup((\text{dual } \mathbf{A})x) = (A - Q\Delta)x$. □

Les conclusions sont les mêmes qu'au §2.7.1 : la proposition 5.7 permet de calculer l'approximation extérieure optimale de $\Sigma(\mathbf{A}, \mathbf{b})$. Il suffit de faire appel à un algorithme de programmation linéaire. Cette approche comporte les mêmes inconvénients : complexité exponentielle, et, en pratique, non-certification des bornes pour les résultats fournis par les outils de programmation linéaire standards.

5.6 Méthode LU généralisée

(*Cette section a fait l'objet de la publication* [GC06a].)

L'élimination de Gauss, et donc la méthode LU, présente une différence notable avec les autres méthodes jusqu'ici revisitées : elle ne peut pas se généraliser directement aux AE-systèmes linéaires. Ceci peut s'expliquer en remarquant que les autres méthodes en question sont basées sur une caractérisation d'un point x appartenant à $\Sigma(\mathbf{A}, \mathbf{b})$. Or cette caractérisation du point x se généralise aux AE-solution sets, ce qui explique pourquoi la méthode elle-même se généralise. La méthode LU intervalle, en revanche, part d'une décomposition sur les réels de $A \in \mathbb{R}^{n\times n}$, qu'elle étend ensuite aux intervalles grâce à l'inclusion $A \in \mathbf{A}$. Lorsque $\mathbf{A} \in \mathbb{KR}^{n\times n}$, cette appartenance n'a plus de sens (et $A \subseteq \mathbf{A}$ est en général faux).

La méthode LU peut par contre être pensée d'une façon différente. Puisque nous sommes dans une structure algébrique qui permet de résoudre des équations de variables intervalles, il est alors censé de chercher, pour une matrice $\mathbf{A} \in \mathbb{KR}^{n\times n}$, à résoudre l'équation de n^2 inconnues suivante :

$$\mathbf{A} = \mathbf{L} \times \mathbf{U}, \tag{5.16}$$

avec $\mathbf{L}$ (resp. $\mathbf{U}$) triangulaire inférieure (resp. supérieure) et $\text{diag}(\mathbf{L}) = I$. Les coefficients de $\mathbf{L}$ et $\mathbf{U}$ forment les n^2 inconnues. Lorsqu'il existe une solution à cette équation, on appelle le couple $(\mathbf{L}, \mathbf{U})$ une *décomposition LU généralisée* de $\mathbf{A}$. Nous allons d'abord montrer au §5.6.1 comment résoudre cette équation. Nous montrerons au §5.6.2 comment exploiter cette décomposition pour l'approximation de AE-solution sets. Nous finirons par quelques considérations pratiques au §5.6.3.

Pour obtenir une décomposition LU généralisée, le principe consiste à étendre directement l'algorithme d'élimination de Gauss sur les réels (au lieu de sa version intervalle) car ce dernier permet effectivement de résoudre l'équation (5.16) dans le cas particulier $(\text{rad } \mathbf{A}) = 0$. Il suffit de prendre soin de conserver à chaque étape le symbole "=", grâce aux propriétés de groupe de $\mathbb{KR}$.

5.6.1 Algorithme

L'élimination de Gauss est généralisée ainsi. Considérons une matrice $\mathbf{A} \in \mathbb{KR}^{n\times n}$. Dès lors que $0 \notin \text{pro } \mathbf{A}_{11}$, pour $i \in [2..n]$, on multiplie la première ligne de $\mathbf{A}$ par $\mathbf{A}_{i1}/(\text{dual } \mathbf{A}_{11})$. Comme $\mathbf{A}_{11}/(\text{dual } \mathbf{A}_{11}) = 1$, la ligne suivante est produite :

$$\left(\mathbf{A}_{i1} \quad , \quad \frac{\mathbf{A}_{12}\mathbf{A}_{i1}}{\text{dual } \mathbf{A}_{11}} \quad , \quad \cdots \quad , \quad \frac{\mathbf{A}_{1n}\mathbf{A}_{i1}}{\text{dual } \mathbf{A}_{11}} \right). \tag{5.17}$$

La seconde étape consiste à soustraire pour les lignes i $(2 \leq i \leq n)$ le dual de la ligne précédemment calculée à la $i^{ème}$ ligne de la matrice $\mathbf{A}$. Comme $\mathbf{A}_{i1} - \text{dual } \mathbf{A}_{i1} = 0$, on obtient

$$\left(0 \quad , \quad \mathbf{A}_{i2} - \frac{(\text{dual } \mathbf{A}_{12})(\text{dual } \mathbf{A}_{i1})}{\mathbf{A}_{11}} \quad , \quad \cdots \quad , \quad \mathbf{A}_{in} - \frac{(\text{dual } \mathbf{A}_{1n})(\text{dual } \mathbf{A}_{i1})}{\mathbf{A}_{11}}\right). \tag{5.18}$$

Une fois cette transformation faite pour $i \in [2..n]$, la matrice intervalle $\mathbf{A}$ apparaît :

$$\mathbf{A} := \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1n} \\ 0 & \mathbf{A}'_{22} & \cdots & \mathbf{A}'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \mathbf{A}'_{n2} & \cdots & \mathbf{A}'_{nn} \end{pmatrix}, \quad \text{avec} \quad \mathbf{A}'_{ij} := \mathbf{A}_{ij} - \frac{(\text{dual } \mathbf{A}_{1j})(\text{dual } \mathbf{A}_{i1})}{\mathbf{A}_{11}}. \tag{5.19}$$

On réitère cette technique du pivot sur les lignes suivantes. De même que dans le contexte des réels, cet algorithme est équivalent à une décomposition LU de la matrice d'intervalles généralisés $\mathbf{A}$, et cette décomposition peut se formuler de la façon suivante :

$$\mathbf{L}_{ii} = 1 \ \text{ et } \ \mathbf{L}_{ij} = 0 \ \text{ for } \ i < j, \tag{5.20}$$

$$\mathbf{L}_{ij} = \Big(\mathbf{A}_{ij} - \sum_{k<j} \text{dual}\,(\mathbf{L}_{ik}\mathbf{U}_{kj})\Big)/(\text{dual } \mathbf{U}_{ii}) \quad \text{ pour } j < i, \tag{5.21}$$

$$\mathbf{U}_{ij} = 0 \ \text{ pour } \ i > j, \tag{5.22}$$

$$\mathbf{U}_{ij} = \mathbf{A}_{ij} - \sum_{k<i} \text{dual}\,(\mathbf{L}_{ik}\mathbf{U}_{kj}) \quad \text{ pour } i \leq j. \tag{5.23}$$

Les expressions précédentes forment une définition récursive des matrices $\mathbf{L}$ et $\mathbf{U}$ vérifiant $\mathbf{A} = \mathbf{LU}$. La construction démarre par la première ligne de $\mathbf{U}$, qui est calculée trivialement à l'aide de (5.23). Cette ligne est identique à la première ligne de $\mathbf{A}$. Ensuite, en supposant $0 \notin \text{pro } \mathbf{U}_{ii}$, la $i^{ème}$ colonne de $\mathbf{L}$ se calcule à partir de (5.21) et la $i^{ème}$ ligne de $\mathbf{U}$ à partir de (5.23). Le procédé est répété récursivement pour $i + 1$.

Proposition 5.8 *Soient $\mathbf{A} \in \mathbb{KR}^{n\times n}$. Supposons que les matrices d'intervalles généralisés $\mathbf{L}$ et $\mathbf{U}$ définies par (5.20-5.23) puissent être construites. Alors, elles satisfont $\mathbf{A} = \mathbf{LU}$.*

Preuve. Considérons $i, j \in [1..n]$ tel que $i \leq j$. D'après (5.23)

$$\mathbf{U}_{ij} = \mathbf{A}_{ij} - \sum_{k<i} \text{dual}\,(\mathbf{L}_{ik}\mathbf{U}_{kj}). \tag{5.24}$$

En ajoutant $\sum_{k<i} \mathbf{L}_{ik}\mathbf{U}_{kj}$ de chaque côté de l'égalité, on obtient

$$\mathbf{U}_{ij} + \sum_{k<i} \mathbf{L}_{ik}\mathbf{U}_{kj} = \mathbf{A}_{ij}.$$

Comme $\mathbf{L}_{jj} = 1$ et $\mathbf{L}_{ik} = 0$ pour $i < k$, l'égalité s'écrit plus simplement

$$\sum_{k=1}^{n} \mathbf{L}_{ik}\mathbf{U}_{kj} = \mathbf{A}_{ij}.$$

Pour le cas $i > j$, un argument similaire s'applique en considérant l'équation (5.21).□

Voici deux exemples de décompositions LU généralisées.

Exemple 5.2 *Soit*

$$\mathbf{A} = \begin{pmatrix} [9,11] & [-1,1] & [-1,1] \\ [-11,11] & [8,12] & [-2,2] \\ [-11,11] & [-12,12] & [7,13] \end{pmatrix}. \tag{5.25}$$

Détail des calculs : Tout d'abord, la première ligne de $\mathbf{U}$ *est la première ligne de* $\mathbf{A}$*. Ensuite, avec* (5.21) *et* (5.23)*,*

$$\begin{array}{lcll} \mathbf{L}_{21} & = & \mathbf{A}_{21}/(\text{dual } \mathbf{U}_{11}) = [-11,11]/[11,9] & = [-1,1] \\ \mathbf{L}_{31} & = & \mathbf{A}_{31}/(\text{dual } \mathbf{U}_{11}) = [-11,11]/[11,9] & = [-1,1] \\ \mathbf{U}_{22} & = & \mathbf{A}_{22} - \text{dual}\,(\mathbf{L}_{21}\mathbf{U}_{12}) & = [9,11] \\ \mathbf{U}_{23} & = & \mathbf{A}_{23} - \text{dual}\,(\mathbf{L}_{21}\mathbf{U}_{13}) & = [-1,1] \\ \mathbf{L}_{32} & = & \big(\mathbf{A}_{32} - \text{dual}\,(\mathbf{L}_{31}\mathbf{U}_{12})\big)/(\text{dual } \mathbf{U}_{22}) & = [-1,1] \\ \mathbf{U}_{33} & = & \mathbf{A}_{33} - \text{dual}\,(\mathbf{L}_{31}\mathbf{U}_{13}) - \text{dual}\,(\mathbf{L}_{32}\mathbf{U}_{23}) & = [9,11]. \end{array}$$

On obtient la décomposition suivante :

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ [-1,1] & 1 & 0 \\ [-1,1] & [-1,1] & 1 \end{pmatrix} \quad et \quad \mathbf{U} = \begin{pmatrix} [9,11] & [-1,1] & [-1,1] \\ 0 & [9,11] & [-1,1] \\ 0 & 0 & [9,11] \end{pmatrix}. \tag{5.26}$$

et ces matrices satisfont bien $\mathbf{A} = \mathbf{LU}$*.*

Exemple 5.3 *Soit*

$$\mathbf{A} = \begin{pmatrix} [2,3] & 1 & 0 \\ 1 & [2,3] & 1 \\ 0 & 1 & [2,3] \end{pmatrix}. \tag{5.27}$$

La décomposition LU généralisée de $\mathbf{A}$ *donne :*

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ [0.5, \frac{1}{3}] & 1 & 0 \\ 0 & [\frac{2}{3}, 0.375] & 1 \end{pmatrix} \quad et \quad \mathbf{U} = \begin{pmatrix} [2,3] & 1 & 0 \\ 0 & [1.5, \frac{8}{3}] & 1 \\ 0 & 0 & [\frac{4}{3}, 2.625] \end{pmatrix}. \tag{5.28}$$

De nouveau, ces matrices satisfont $\mathbf{A} = \mathbf{LU}$*. On note alors que* $\mathbf{L}$ *n'est plus entièrement propre.*

Une conséquence de la propriété $\mathbf{A} \subseteq (\text{pro } \mathbf{A})$ est qu'étant donnée une matrice $\mathbf{A} \in \mathbb{KR}^{n\times n}$, si (pro $\mathbf{A}$) possède une décomposition LU "classique" (celle du §4.6), alors $\mathbf{A}$ possède une décomposition LU généralisée. En effet, si on note $(\mathbf{L}', \mathbf{U}')$ la décomposition *LU* classique de pro $\mathbf{A}$, et $(\mathbf{L}, \mathbf{U})$ la décomposition généralisée de $\mathbf{A}$, on s'aperçoit en "annulant les dual" dans la définition récursive de $\mathbf{L}$ et $\mathbf{U}$ que $\mathbf{L} \subseteq \mathbf{L}'$ et $\mathbf{U} \subseteq \mathbf{U}'$. Donc si la décomposition classique ne fait pas apparaître de pivot nul, la décomposition généralisée non plus. Ainsi, toutes les matrices $\mathbf{A}$ dont la projection propre remplit l'une des conditions[4] pour admettre une décomposition LU classique admettent une décomposition LU généralisée.

5.6.2 Approximation basée sur la décomposition LU généralisée

La décomposition *LU* peut servir à approximer intérieurement ou extérieurement des AE-solution sets. L'approximation intérieure consiste simplement à remarquer que l'équation formelle de la proposition 5.3 est résolue plus simplement lorsque $\mathbf{A} = \mathbf{LU}$. L'approximation extérieure est plus délicate : elle repose intuitivement sur le fait qu'en présence de matrices entièrement impropres, les inclusions s'inversent.

[4] Voir [Neu90] où de telles conditions sont énoncées.

Proposition 5.9 *Soient* $\mathbf{A} \in \mathbb{KR}^{n\times n}$, $\mathbf{b} \in \mathbb{KR}^n$. *Supposons que la décomposition LU généralisée de* $\mathbf{A}$ *soit possible et notons* $\mathbf{L}$ *et* $\mathbf{U}$ *le résultat de cette décomposition. Définissons alors les vecteurs d'intervalles généralisés* $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}' \in \mathbb{KR}^n$ *tels que pour* $i \in [1..n]$,

$$\mathbf{y}_i = \mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}(\text{dual } \mathbf{y}_j) \quad et \quad \mathbf{x}_i = \Big(\mathbf{y}_i - \sum_{j>i} \mathbf{U}_{ij}(\text{dual } \mathbf{x}_j)\Big)/\mathbf{U}_{ii},$$

$$\mathbf{y}'_i = \mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}\mathbf{y}'_j \quad et \quad \mathbf{x}'_i = \Big(\mathbf{y}'_i - \sum_{j>i} \mathbf{U}_{ij}\mathbf{x}'_j\Big)/\mathbf{U}_{ii}.$$

Alors,
(i) Si $\mathbf{L}$ *et* $\mathbf{x}$ *sont propres, alors* $\mathbf{x} \subseteq \Sigma(\mathbf{A}, \mathbf{b})$.
(ii) Supposons $\mathbf{U}$ *et* $\mathbf{L}$ *impropres. Si* $\mathbf{x}'$ *est propre alors* $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}'$. *Sinon,* $\Sigma(\mathbf{A}, \mathbf{b}) = \emptyset$.

La preuve de la proposition utilise le lemme suivant :

Lemme 5.1 *Soient* $\mathbf{L}$ *et* $\mathbf{U}$ *deux matrices de* $\mathbb{KR}^{n\times n}$, x *un vecteur de* $\mathbb{R}^n$.

$$\begin{array}{lll} (i) & \textit{Si } \mathbf{L} \textit{ est propre} & (\text{dual } \mathbf{L})\Big((\text{dual } \mathbf{U})x\Big) \supseteq \Big((\text{dual } \mathbf{L})(\text{dual } \mathbf{U})\Big)x. \\ (ii) & \textit{Si } \mathbf{L} \textit{ est impropre} & (\text{dual } \mathbf{L})\Big((\text{dual } \mathbf{U})x\Big) \subseteq \Big((\text{dual } \mathbf{L})(\text{dual } \mathbf{U})\Big)x. \end{array}$$

Preuve. (i) Posons $\mathbf{b} := (\text{dual } \mathbf{L})\Big((\text{dual } \mathbf{U})x\Big)$ et cherchons à prouver l'inclusion pour la $i^{ème}$ composante ($i \in [1..n]$ quelconque). On a

$$\mathbf{b}_i = \sum_{j=1}^{n} (\text{dual } \mathbf{L}_{ij}) \sum_{k=1}^{n} \Big((\text{dual } \mathbf{U}_{jk})x_k\Big).$$

Comme $\mathbf{L}$ est propre, (dual $\mathbf{L}$) est impropre. On peut donc appliquer n fois la super-distributivité (5.8) p.112 et obtenir

$$\mathbf{b}_i \supseteq \sum_{j=1}^{n} \sum_{k=1}^{n} (\text{dual } \mathbf{L}_{ij}) \Big((\text{dual } \mathbf{U}_{jk})x_k\Big).$$

On fait alors appel à (5.5) p.112 pour changer le parenthésage :

$$\mathbf{b}_i \supseteq \sum_{j=1}^{n} \sum_{k=1}^{n} \Big((\text{dual } \mathbf{L}_{ij})(\text{dual } \mathbf{U}_{jk})x_k\Big)$$

Le reste repose sur un changement d'indice classique :

$$\begin{array}{ll} \mathbf{b}_i & \supseteq \sum_{k=1}^{n} \sum_{j=1}^{n} \Big((\text{dual } \mathbf{L}_{ij})(\text{dual } \mathbf{U}_{jk})x_k\Big) \\ & \supseteq \sum_{k=1}^{n} \Big(\sum_{j=1}^{n} ((\text{dual } \mathbf{L}_{ij})(\text{dual } \mathbf{U}_{jk})\Big)x_k \\ & \supseteq \sum_{k=1}^{n} \Big((\text{dual } \mathbf{L})(\text{dual } \mathbf{U})\Big)_{ik} x_k \\ & \supseteq \Big[\Big((\text{dual } \mathbf{L})(\text{dual } \mathbf{U})\Big)\mathbf{x}\Big]_i. \end{array}$$

(ii) s'obtient de la même façon, via la sous-distributivité (5.7). □

Preuve. [de la proposition 5.9] Tout d'abord, si $\mathbf{A} = \mathbf{LU}$ alors d'après (5.4)(dual $\mathbf{A}$) = (dual $\mathbf{L}$)(dual $\mathbf{U}$).

(i) Par définition de $\mathbf{y}$ et $\mathbf{x}$, on a (dual $\mathbf{L}$)$\mathbf{y} = \mathbf{b}$ et (dual $\mathbf{U}$)$\mathbf{x} = \mathbf{y}$. Par conséquent, (dual $\mathbf{L}$)$\big((\text{dual } \mathbf{U})\mathbf{x}\big) = \mathbf{b}$. Comme $\mathbf{L}$ est supposé propre, (dual $\mathbf{L}$) est impropre. Considérons $x \in \mathbf{x}$ quelconque. D'après le lemme 5.1 (i), $\big((\text{dual } \mathbf{L})(\text{dual } \mathbf{U})\big)x \subseteq (\text{dual } \mathbf{L})\big((\text{dual } \mathbf{U})x\big)$. On en déduit (dual $\mathbf{A}$)$x \subseteq \mathbf{b}$ et finalement $x \in \Sigma(\mathbf{A}, \mathbf{b})$, grâce à la caractérisation fondamentale (prop. 5.2).

(ii) Supposons $x \in \Sigma(\mathbf{A}, \mathbf{b})$. Grâce à la caractérisation fondamentale, $\big((\text{dual } \mathbf{L})(\text{dual } \mathbf{U})\big)x \subseteq \mathbf{b}$. Comme (dual $\mathbf{L}$) et (dual $\mathbf{U}$) sont tous deux propres, $\mathbf{b}$ ne peut être que propre.

Grâce au lemme 5.1 (ii), (dual $\mathbf{L}$)$\big((\text{dual } \mathbf{U})x\big) \subseteq \big((\text{dual } \mathbf{L})(\text{dual } \mathbf{U})\big)x$, d'où (dual $\mathbf{L}$)$\big((\text{dual } \mathbf{U})x\big) \subseteq \mathbf{b}$. Comme (dual $\mathbf{U}$) est propre, $\mathbf{u} := (\text{dual } \mathbf{U})x$ est également propre. Nous allons montrer que $\mathbf{u} \subseteq \mathbf{y}'$. Nous procédons par induction. La première ligne de (dual $\mathbf{L}$)$\mathbf{u} \subseteq \mathbf{b}$ produit $\mathbf{u}_1 \subseteq \mathbf{b}_1$ ou $\mathbf{b}_1 = \mathbf{y}'_1$ par définition de $\mathbf{y}'$. Fixons maintenant $i \in [2..(n-1)]$ et supposons $j < i \Longrightarrow \mathbf{u}_j \subseteq \mathbf{y}'_j$. La $i^{ème}$ ligne de (dual $\mathbf{L}$)$\mathbf{u} \subseteq \mathbf{b}$ produit $\mathbf{u}_i + \sum_{j<i}(\text{dual } \mathbf{L}_{ij})\mathbf{u}_j \subseteq \mathbf{b}_i$. Par conséquent $\mathbf{u}_i \subseteq \mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}(\text{dual } \mathbf{u}_j) \subseteq \mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}\mathbf{u}_j$, la seconde inclusion étant une conséquence de la monotonie pour l'inclusion (dual $\mathbf{u}_j \subseteq \mathbf{u}_j$ car $\mathbf{u}_j$ est propre). Finalement, en appliquant l'hypothèse d'induction puis de nouveau la monotonie pour l'inclusion, on obtient $\mathbf{u}_i \subseteq \mathbf{b}_i - \sum_{j<i} \mathbf{L}_{ij}\mathbf{y}'_j$, ce dernier intervalle étant précisément par définition $\mathbf{y}'_i$. Jusqu'ici, nous avons prouvé (dual $\mathbf{U}$)$x \subseteq \mathbf{y}'$. A l'aide d'une induction similaire, on peut également prouver $x \subseteq \mathbf{x}'$. Il s'ensuit que si $\mathbf{x}'$ est propre alors $\Sigma(\mathbf{A}, \mathbf{b}) \subseteq \mathbf{x}'$. Sinon $\Sigma(\mathbf{A}, \mathbf{b}) = \emptyset$. □

5.6.3 Application

Les fortes conditions exigées par la proposition 5.9 la rendent peu utile en pratique. Elle est susceptible d'être applicable pour le calcul d'une boîte extérieure lorsque $\mathbf{A}$ est impropre, i.e. lorsqu'un *tolerable solution set* est traité, et pour le calcul d'une boîte intérieure lorsque $\mathbf{A}$ est propre, i.e. lorsqu'un *united solution set* est traité. Sous ces conditions, il apparaît empiriquement que la décomposition LU généralisée appliquées à des matrices diagonalement dominantes centrées autour de l'identité satisfait les conditions de la proposition 5.9. Hélas, pour cette classe de matrices, l'opérateur de Gauss-Seidel généralisé (**GIGS**) (cf. §5.3) et l'approche formelle (cf. §5.2) semblent donner des meilleurs résultats, le premier pour la boîte extérieure et la seconde pour la boîte intérieure.

Il ne reste a priori à décharge pour la méthode LU que le gain de temps obtenu lorsque plusieurs systèmes sont considérés avec la même matrice (puisque la décomposition ne doit être calculée qu'une seule fois). Dans certaines situations cependant, la décomposition LU généralisée peut fournir une approximation bien plus précise que celle fournie par l'opérateur GIGS, comme illustré sur l'exemple suivant :

Exemple 5.4 *Considérons la matrice tridiagonale* $\mathbf{A}$ *et le vecteur* $\mathbf{b}$ *définis par*

$$\mathbf{A} = \begin{pmatrix} \mathbf{a} & \mathbf{a} & 0 \\ \mathbf{a} & \mathbf{a} & \mathbf{a} \\ 0 & \mathbf{a} & \mathbf{a} \end{pmatrix} \quad et \quad \mathbf{b} = \begin{pmatrix} [0, 10] \\ [-10, 0] \\ [-5, 5] \end{pmatrix}$$

avec $\mathbf{a} = [1.1, 0.9]$ *et le tolerable solution set* $\Sigma(\mathbf{A}, \mathbf{b})$*. La décomposition LU généralisée et l'opérateur GIGS ont le même comportement sur cet exemple. Cependant, le fait de permuter les lignes* 2 *et* 3 *change la donne. A partir de l'estimée initiale* $(\pm 1000, \pm 1000, \pm 1000)^T$ *l'opérateur GIGS retourne alors*

$$([-18.4091, 27.5], [-22.5, 22.5], [-27.5, 18.4091])^T,$$

tandis que l'approximation extérieure basée sur la décomposition LU généralisée produit :

$$([-12.8099, 11.157], [-9.12847, 19.5718], [-13.6364, 4.54545])^T,$$

ce qui est un bien meilleur résultat.

Il doit être noté qu'un changement léger dans les incertitudes des matrices intervalles de l'exemple précédent conduit à une matrice $\mathbf{L}$ qui n'est plus propre. En conclusion, certaines instances favorables rendent l'approche LU digne d'intérêt, mais un travail d'investigation a priori délicat doit être conduit pour mieux cerner la classe regroupant ces instances.

5.7 Méthode de Hansen-Bliek généralisée

(*Cette section a fait l'objet de la publication* [CG06].)

Nous proposons une extension de la méthode de Hansen-Bliek (cf. §4.7 p.100) aux AE-systèmes linéaires « quantifiés à droite ». Cette classe de problèmes a été rencontrée au chapitre 3 p.69, où elle s'est avérée être la pierre angulaire de l'opérateur de Newton généralisé. Bien que d'apparence restrictive, cette classe possède donc une utilité particulière. L'exemple 5.5 et la figure 5.1 ci-dessous montrent un exemple de *controllable solution set*.

Exemple 5.5

$$\begin{pmatrix} [1,1] & [-0.5,0.5] \\ [-0.5,0.5] & [1,1] \end{pmatrix} x = \begin{pmatrix} [1,-0.5] \\ [5.5,4.5] \end{pmatrix}$$

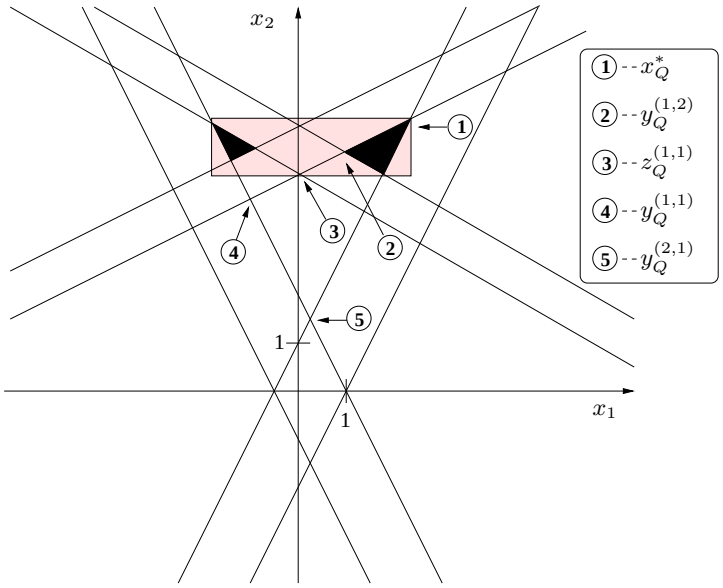


FIG. *5.1: Controllable solution set de l'exemple 5.5 (en noir), et son enveloppe (en gris). Les annotations servent à illustrer le propos des sections suivantes : les points* 1 *à* 5 *sont des exemples de points candidats. L'indice Q désigne le quadrant positif* ($x_1 \geq 0$, $x_2 \geq 0$).

On remarque sur la figure que ce AE-solution set est non connexe.

5.7.1 Résultat principal

De même que dans le cas classique, nous nous intéressons ici aux matrices généralisées $\mathbf{A}$ vérifiant (mid $\mathbf{A}$) $= I$ et $\rho(\text{rad}\,(\mathbf{A})) < 1$ (voir chapitre 4). Nous noterons simplement Σ au lieu de $\Sigma(\mathbf{A}, \mathbf{b})$ le *solution set* quantifié à droite considéré. $\mathbf{A}$ est donc entièrement quantifiée existentiellement, $\mathbf{b}$ est quantifié librement.

Rappelons que l'on note $\Delta := \text{rad}\ \mathbf{A}$ et $b := \text{mid}\ \mathbf{b}$ (cf. définition 5.3 et prop. 5.7). Nous notons M l'inverse de $(I - \Delta)$, et m_{ij} les coefficients de M. D'après l'étude faite au 4.1.3 p.83, $I - \Delta$ est donc une M-matrice, ce qui implique $M \geq 0$. Enfin, nous héritons du §4.7 les notations dues à Rohn [Roh93].

La formule dans le cas généralisé est la suivante :

Fixons $k \in [1..n]$, et considérons

$$x^* := M(|b| + \delta);$$

$$\tilde{x}_k := \begin{cases} x^*_k & \text{si } b_k \geq 0, \\ x^*_k + 2m_{kk}b_k & \text{sinon}\,; \end{cases}$$

$$\underline{x}_k := \begin{cases} -x^*_k & \text{si } b_k \leq 0, \\ -x^*_k + 2m_{kk}b_k & \text{sinon}\,; \end{cases}$$

$$\nu_k = \frac{1}{2m_{kk} - 1}.$$

Définissons également $\hat{x}_k$ de la façon suivante :

$$\begin{array}{lll} \text{Si } (\forall i \neq k \quad m_{ik} = 0) & \text{alors} & \hat{x}_k = -\infty, \\ & \text{sinon} & \hat{x}_k = \max\{\theta_{ki}, i \neq k \text{ et } m_{ik} \neq 0\}, \end{array}$$

où

$$\theta_{ki} := \alpha_{ki}|b_i| + \sum_{j=1, j\neq k}^{n} (m_{kj} - m_{ij}\alpha_{ki})(|b_j| + \delta_j) \quad \text{avec} \quad \alpha_{ki} = m_{kk}/m_{ik}.$$

Calculons enfin $\underline{x}_k$ et $\overline{x}_k$ en suivant la procédure ci-dessous :

```
si (x̃_k ≥ max{0, ν_k x_k, x̂_k}) alors x̄_k = x̃_k
   sinon x̄_k = min{ν_k x̃_k, −x̂_k}
   si (x_k ≤ min{0, ν_k x̃_k, −x̂_k}) alors x_k = x_k
   sinon x_k = max{ν_k x_k, x̂_k}
```

Nous allons prouver le résultat suivant :
– Σ est vide ssi il existe $k \in [1..n]$ tel que[5] $\underline{x}_k > \overline{x}_k$.
– Si Σ n'est pas vide, alors pour tout $k \in [1..n]$, la $k^{\text{ème}}$ composante de $\square\Sigma$ est $[\underline{x}_k, \overline{x}_k]$.

La preuve suit le même découpage qu'au §4.7. Tout d'abord (§5.7.2), nous supposons qu'il existe des solutions vérifiant $x_k \geq 0$ et d'autres vérifiant $x_k \leq 0$. On calcule alors la borne supérieure de $|x_k|$ pour $x \in \Sigma$ et $x_k \geq 0$, puis pour $x \in \Sigma$ et $x_k \leq 0$. Nous montrons que la borne vaut dans le premier cas $\tilde{x}_k$, et dans le second $-\underline{x}_k$ (corollaires 5.1 & 5.2). Ainsi, $[\underline{x}_k, \tilde{x}_k]$ est la projection de $\square\,\Sigma$ sur la $k^{\text{ème}}$ coordonnée. Ces bornes sont valides sous l'hypothèse que x_k peut être à la fois positif et négatif dans Σ. Cette hypothèse peut être vérifiée, grâce à la proposition 5.16 et au corollaire 5.6, au §5.7.8.

Dans un second temps (§5.7.3), nous supposons que les $k^{\text{ème}}$ coordonnées des points solutions sont soit toutes positives soit toutes négatives. Dans ce cas, il est nécessaire de calculer la borne inférieure de $|x_k|$. Avec $x_k \geq 0$, cette borne peut être $\nu_k\underline{x}_k$ ou $\hat{x}_k$, et avec $x_k \leq 0$, elle peut être $\nu_k\tilde{x}_k$ ou $-\hat{x}_k$ (cf proposition 5.15 et corollaire 5.5).

Au §4.7 p.100 nous avons montré que le théorème d'Oettli-Prager du chapitre 2 (prop 2.10 p.31) pouvait être mis sous une forme mieux exploitable, donnée à la proposition 4.20 p.101. Le théorème d'Oettli-Prager a été étendu aux intervalles généralisés pour donner la proposition 5.7 p.117. La caractérisation restant formellement la même, la transformation de la proposition 4.20 reste valable avec exactement la même preuve, puisque celle-ci n'utilise pas le fait que Δ ou δ soit positif. Nous réécrivons par commodité cette proposition, en nous plaçant directement dans le cas qui nous concerne : mid $\mathbf{A} = I$.

Proposition 5.10 *Soit Q un quadrant*

$$x \in \Sigma \Longleftrightarrow \begin{cases} (I - \Delta)Qx \leq Qb + \delta & (a) \\ (I + \Delta)Qx \geq Qb - \delta & (b) \\ Qx \geq 0 & (c) \end{cases} \tag{5.29}$$

[5] $\underline{x}_k > \overline{x}_k$ ne peut survenir que si $\underline{x}_k$ et $\overline{x}_k$ sont tous deux issus des "sinon".

5.7.2 Maximisation de $|x_k|$

Rappelons que k est fixé. On note $\Sigma^{(k+)}$ l'ensemble $\{x \in \mathbb{R}^n | x \in \Sigma \wedge x_k \geq 0\}$. Similairement, $\Sigma^{(k-)}$ désigne $\{x \in \mathbb{R}^n | x \in \Sigma \wedge x_k \leq 0\}$. Dans ce paragraphe, nous donnons une expression pour la borne supérieure de $|x_k|$ pour $x \in \Sigma^{(k+)}$. La borne supérieure de $|x_k|$ pour $x \in \Sigma^{(k-)}$ découle par symétrie.

L'approche de Hansen et Bliek repose sur la capacité à casser le balayage combinatoire de la méthode exhaustive (cf. §5.5) en localisant un quadrant particulier où le maximum global est atteint.

Dans une première étape, nous fixons arbitrairement un quadrant Q, et identifions formellement le point qui maximise $(Qx)_k$ *localement*, i.e., pour $x \in \Sigma \cap Q$. On constate qu'il n'y a alors qu'une seule expression possible (notée x_Q) pour ce maximum. Comme tous les quadrants partagent cette expression pour leur maximum, il est alors facile d'isoler **le** quadrant (noté $Q^{(k+)}$) ayant le maximum global.

Bien que les résultats soient ici exactement les mêmes qu'au §4.7.1 (ce que la figure 5.2 ci-dessous illustre avec deux cas de figure), les preuves proposées sont différentes car nous devons prendre en compte les quantificateurs de $\mathbf{b}$[6].

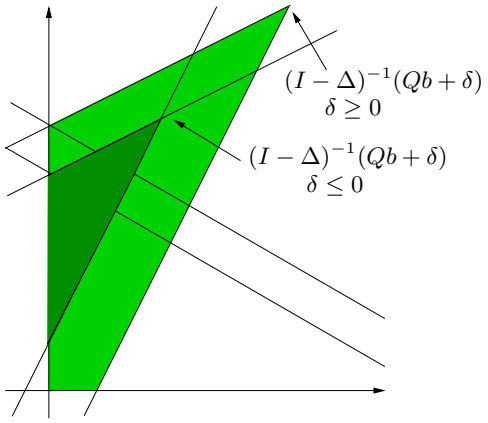


FIG. *5.2: Point maximisant $|x|$ dans le quadrant positif, dans le cas d'un united solution set ($\delta \geq 0$) et dans le cas d'un controllable solution set ($\delta \leq 0$).*

Pour un quadrant Q donné, définissons $x_Q := QM(Qb+\delta)$ de telle sorte que $(I-\Delta)Qx_Q = (Qb+\delta)$. Le lemme suivant montre que $|x_Q|$ maximise $|x|$ dans $\Sigma \cap Q$.

Lemme 5.2 (Maximum local) *Les trois conditions suivantes sont équivalentes :*
(i) $\Sigma \cap Q \neq \emptyset$ *(ii)* $x_Q \in \Sigma \cap Q$ *(iii)* $Qx_Q \geq 0 \wedge \Delta Qx_Q \geq -\delta$.
De plus, si l'une de ces conditions est vérifiée, alors :
(iv) $|x_Q| = \max_{x \in \Sigma \cap Q} |x|$

Preuve. Il est évident que (ii) implique (i). Prouvons (i) implique (ii). D'abord, il est clair que x_Q satisfait (5.29.a). Ensuite, d'après (i) il existe $x \in \Sigma \cap Q$. Ce point x satisfait donc (5.29.a), c.a.d. $(I-\Delta)Qx \leq Qb+\delta$. Multiplier cette inégalité par la matrice positive M entraîne $Qx \leq Qx_Q$. Or, $0 \leq Qx$ de par (5.29.c), donc x_Q satisfait également (5.29.c). L'inégalité $Qx \leq Qx_Q$ entraîne également $(I+\Delta)Qx \leq (I+\Delta)Qx_Q$ car $I+\Delta \geq 0$. Et puisque x satisfait (5.29.b) (i.e. $Qb-\delta \leq (I+\Delta)Qx$), x_Q satisfait également (5.29.b) ce qui complète la preuve de (i) implique (ii). Prouvons maintenant que (ii) est équivalent à (iii). Comme x_Q satisfait (5.29.a)

[6]Par exemple, contrairement au cas où $\mathbf{b}$ est entièrement existentiel, nous ne pouvons plus extraire une matrice $\tilde{A} \in \mathbf{A}$ et un vecteur $\tilde{b} \in \mathbf{b}$ vérifiant $\tilde{A}x = \tilde{b}$ pour prouver l'appartenance de x à Σ.

par construction, et comme $Qx_Q \geq 0$ est présent à la fois dans (ii) (via (5.29.c)) et (iii), nous devons juste prouver que x_Q satisfait (5.29.b) (c.a.d. $(I+\Delta)Qx_Q \geq Qb - \delta$)) ssi $\Delta Qx_Q \geq -\delta$. Il suffit alors de soustraire l'inégalité $(I-\Delta)Qx_Q = (Qb+\delta)$ (qui définit x_Q) à $(I+\Delta)Qx_Q \geq Qb-\delta$ pour obtenir l'inégalité équivalente $2\Delta Qx_Q \geq -2\delta$. Par conséquence, $\Delta Qx_Q \geq -\delta$ est équivalent à (5.29.b). Pour prouver que (i)$\Longrightarrow$(iv), prenons de nouveau x dans $\Sigma \cap Q$. On a vu que (i) implique $Qx \leq Qx_Q$, i.e. $|x| \leq Qx_Q$. Mais $x_Q \in Q$, du fait que (i) implique (ii). Ainsi, $|x_Q| = \max_{x \in \Sigma \cap Q} |x|$. □

Nous calculons maintenant le maximum de $|x_k|$ (ou simplement x_k) lorsque x décrit $\Sigma^{(k+)}$.

Définissons Q^* et $Q^{(k+)}$ comme les matrices diagonales suivantes[7] :

$$Q^*_{ii} := sign(b_i); \qquad\qquad Q^{(k+)}_{ii} := \begin{cases} 1 & \text{si } i = k, \\ sign(b_i) & \text{sinon.} \end{cases}$$

Proposition 5.11 (Maximum global)

$$\Sigma^{(k+)} \neq \emptyset \implies \begin{cases} \Sigma \cap Q^{(k+)} \neq \emptyset \\ \big(x_{Q^{(k+)}}\big)_k = \max\{x_k \mid x \in \Sigma^{(k+)}\} \end{cases}$$

Preuve. Considérons $x \in \Sigma^{(k+)}$ et notons Q son quadrant. On a $Q_{kk} = 1$ et $Qb \leq Q^{(k+)}b$. Par conséquent, M étant positive,

$$M\big(Qb+\delta\big) \leq M\big(Q^{(k+)}b+\delta\big).$$

Par définition de x_Q et $x_{Q^{(k+)}}$ cela signifie $Qx_Q \leq Q^{(k+)}x_{Q^{(k+)}}$. Par ailleurs, en utilisant (i)$\Longrightarrow$(iv) du lemme 5.2, on a $Qx \leq Qx_Q$. Donc $x_k \leq (x_{Q^{(k+)}})_k$ (puisque $Q_{kk} = Q^{(k+)}_{kk} = 1$).

Cette inégalité étant valable pour tout $x \in \Sigma^{(k+)}$, on a montré

$$\big(x_{Q^{(k+)}}\big)_k \geq \max\{x_k | x \in \Sigma^{(k+)}\}.$$

D'autre part, de par "(i) implique (iii)" du lemme 5.2, $\Sigma \cap Q \neq \emptyset$ implique $0 \leq Qx_Q$ et $-\delta \leq \Delta Qx_Q$. Ceci entraîne à la fois $0 \leq Q^{(k+)}x_{Q^{(k+)}}$ et $-\delta \leq \Delta Q^{(k+)}x_{Q^{(k+)}}$ (car $Qx_Q \leq Q^{(k+)}x_{Q^{(k+)}}$ et $\Delta \geq 0$). En appliquant "(iii) implique (ii)" du lemme 5.2, on obtient $x_{Q^{(k+)}} \in \Sigma \cap Q^{(k+)} \subseteq \Sigma^{(k+)}$. Ainsi $\Sigma \cap Q^{(k+)} \neq \emptyset$ et

$$\big(x_{Q^{(k+)}}\big)_k \leq \max\{x_k | x \in \Sigma^{(k+)}\},$$

ce qui complète la preuve. □

Remarque 5.3 *Par contraposition, si* $\big(x_{Q^{(k+)}}\big)_k < 0$ *alors* $\Sigma^{(k+)}$ *est forcément vide.*

La proposition 5.11 montre que le calcul du maximum de x_k dans $\Sigma^{(k+)}$ ne requiert que le calcul de la $k^{ème}$ composante de $Q^{(k+)}x_{Q^{(k+)}}$. Par conséquent, il est inutile de calculer toutes les composantes de tels vecteurs, et le corollaire suivant montre que ces n réels peuvent être calculés directement à partir du vecteur x^*, de même que dans le cas classique.

Corollaire 5.1 *Soit* $\tilde{x}$ *défini à partir de* x^* *(voir §5.7.1).*

$$\begin{aligned} (x_{Q^{(k+)}})_k = \tilde{x}_k \qquad & (i) \\ \Sigma^{(k+)} \neq \emptyset \Longrightarrow \tilde{x}_k = \max\{x_k \mid x \in \Sigma^{(k+)}\} \qquad & (ii) \end{aligned}$$

[7] Le signe de 0 peut être choisi comme étant 1 ou -1, sans impact.

Preuve. Grâce à la proposition 5.11, nous devons juste prouver (i).
La preuve est la même que celle du corollaire 4.2 p.104. Clairement, $Q^*b = |b|$ et $x^* = Q^*x_{Q^*} = M(Q^*b + \delta)$. Maintenant, si $sign(b_k) = 1$, on a $Q^{(k+)}_{kk} = Q^*_{kk}$, c.a.d. $Q^{(k+)} = Q^*$, et donc $(x_{Q^{(k+)}})_k = x^*_k$. Sinon, $Q^{(k+)} = Q^* + 2e_k e_k^T$ (e_k est la $k^{ème}$ colonne de I) et $|x_{Q^{(k+)}}| = M(Q^{(k+)}b + \delta) = M((Q^* + 2e_k e_k^T)b + \delta)$ d'où $(x_{Q^{(k+)}})_k = x^*_k + 2m_{kk}b_k$. Finalement, $(x_{Q^{(k+)}})_k = \tilde{x}_k$. □

On en déduit le calcul de la borne inférieure de x_k, lorsque x décrit $\Sigma^{(k-)}$.

Corollaire 5.2

$$\Sigma^{(k-)} \neq \emptyset \Longrightarrow \underset{\sim}{x}_k = \min\{x_k \mid x \in \Sigma^{(k-)}\}.$$

Preuve. On applique le corollaire 5.1 au AE-solution set $\Sigma' := \Sigma(\mathbf{A}, -\mathbf{b})$. La borne supérieure obtenue pour x_k positif dans Σ' et l'opposé de la borne inférieure recherchée. □

5.7.3 Minimisation de $|x_k|$: introduction

Le découpage logique du paragraphe précédent (optimum local/optimum global) peut s'appliquer de nouveau au minimum de $|x_k|$, à ceci près que la situation se complique nettement dans la mesure où un minimum local peut avoir jusqu'à $2n$ expressions différentes, au lieu d'une seule (cf. figure 5.3). Chacune de ces expressions sera appelée un *candidat* (formel).

Dans le cas classique (cf. §4.7.2 p.104), il n'y a toujours qu'un seul candidat pour la minimisation. Les quantificateurs introduisent donc à partir de maintenant un changement considérable dans la complexité du problème.

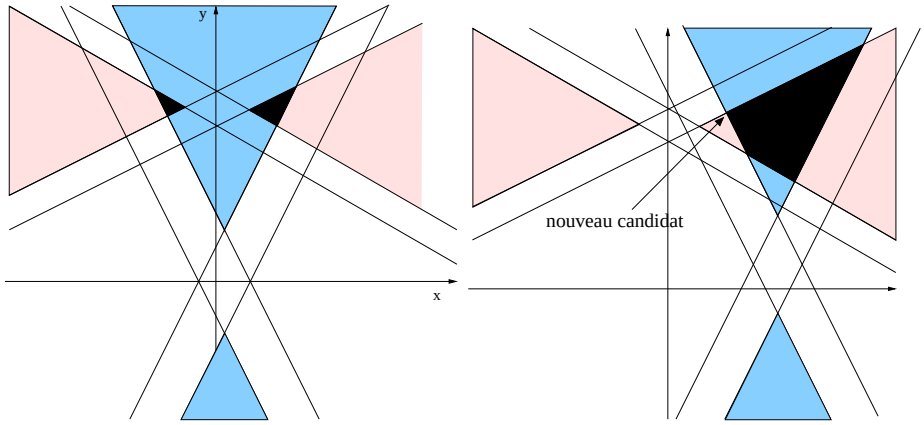


FIG. *5.3: Deux exemples de controllable solution sets. A gauche, les candidats sont les mêmes que dans le cas classique. Le fait que les zones intersectées (en gris) pour former Σ soient non connexes fait apparaître dans la partie droite un nouveau candidat par rapport au cas classique.*

Nous allons prouver d'abord au §5.7.4 que le minimum local de $|x_k|$ (le minimum pour un quadrant Q donné) est nécessairement atteint en un point appartenant à une droite spécifique appelée "droite support" (de Q). Ensuite, au §5.7.5, nous établissons que "seulement" $2n$ points de cette droite peuvent être des *candidats* pour la minimisation. Nous donnons une formule explicite pour chacun d'entre eux, en les regroupant en deux catégories (les *candidats intérieurs* et les *candidats limitrophes*). Plus loin, au §5.7.6 nous montrons que le minimiseur est précisément celui qui maximise la $k^{ème}$ composante (en valeur absolue) parmi les candidats. Ceci permet d'identifier formellement le minimum local.

Au §5.7.7 nous prouvons, comme dans le cas de la borne supérieure, que le minimum global (c.a.d. le minimum de tous les minimums locaux) se trouve dans un quadrant formellement identifié, qui, par ailleurs, n'est autre que $Q^{(k+)}$.

5.7.4 La droite support

La droite support d'un quadrant est définie comme l'intersection de $(n-1)$ hyperplans obtenus en supprimant la $k^{ème}$ ligne dans le système (5.29.a).

Le fait de supprimer certaines lignes (ou colonnes) d'une matrice X pose un problème de notation. Pour y remédier, nous introduisons un système d'indice "évolué", afin de continuer à être rigoureux sans trop alourdir le texte. Ce système s'applique aussi bien aux matrices qu'aux vecteurs. Si X est une matrice, X_{ij} continue de représenter un coefficient lorsque i et j sont deux entiers. Mais i peut avoir d'autres types de valeurs comme ":", "$\neq k$" ou "$< k$" qui désignent respectivement toutes les lignes, toutes les lignes sauf k, et les $(k-1)$ premières lignes. De la même façon, j peut prendre ces valeurs spéciales pour représenter un sous-ensemble de colonnes. Ainsi, par exemple, $X_{\neq k,:}$ représente la matrice X dont la $k^{ème}$ ligne a été ôtée.

Nous aurons besoin plus loin de la propriété suivante pour certaines sous-matrices :

Lemme 5.3

$$\big(\forall k \in [1..n]\big)\ (I-\Delta)_{\neq k,\neq k} \text{ est une M-matrice .}$$

Preuve. Nous rappelons deux propriétés des M-matrices introduites au §4.1.3 p.83 :

$$A \text{ est une M-matrice } \iff \big(\exists u > 0\big)\ Au > 0, \tag{5.30}$$
$$A \text{ est une M-matrice } \implies \big(\forall i \neq k\big)\ A_{ik} \leq 0. \tag{5.31}$$

Comme $I-\Delta$ est une M-matrice, d'après (5.30), $\exists u > 0$ tel que $(I-\Delta)u > 0$. D'après (5.31), pour un tel u, $(I-\Delta)_{\neq k,\neq k}(u_{\neq k}) > 0$ et de nouveau grâce à (5.30), $(I-\Delta)_{\neq k,\neq k}$ est une M-matrice. □

Entrons maintenant dans le vif du sujet.

Définition 5.4
Soit Q un quadrant. On appelle droite support et on note $\mathcal{L}_Q^{(k)}$ l'ensemble suivant :

$$\mathcal{L}_Q^{(k)} := \{x \mid (I-\Delta)_{\neq k,:}Qx = (Qb+\delta)_{\neq k}\}.$$

Pour prouver que le minimum de $|x_k|$ est atteint sur $\mathcal{L}_Q^{(k)}$ (voir Proposition 5.12 ci-dessous), nous montrons d'abord que tout point $\tilde{x}$ solution peut être projeté sur $\mathcal{L}_Q^{(k)}$ orthogonalement à l'axe x_k, tout en restant dans l'ensemble des solutions (voir figure 5.4). Nous n'aurons ensuite plus qu'à appliquer cette propriété à *n'importe quel* point qui minimise $|x_k|$.

Lemme 5.4
Soit Q un quadrant.

$$\big(\forall \tilde{x} \in \Sigma \cap Q\big)\big(\exists x \in \Sigma \cap Q \cap \mathcal{L}_Q^{(k)}\big)\big(|x_k| = |\tilde{x}_k|\big)$$

Preuve. Cette preuve nécessite de manipuler la $k^{ème}$ ligne d'une matrice. Les notations pouvant devenir vite lourdes, nous supposons ici pour les alléger que $k = 1$. Il n'y a aucune perte de généralité car cette supposition revient simplement à réordonner les variables.

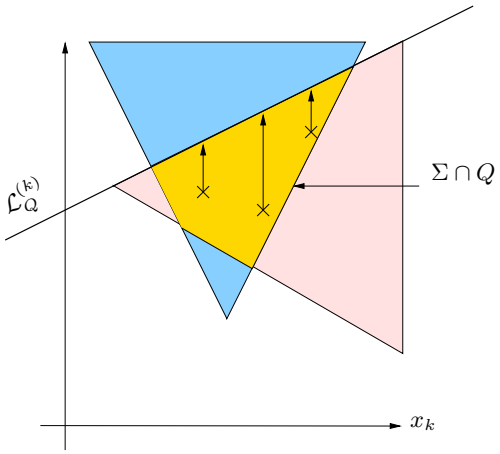


FIG. *5.4: Projection d'un point de $\Sigma \cap Q$ sur la droite support*

Soit $\check{x}$ quelconque dans $\Sigma \cap Q$. Nous allons construire un point x qui convient. D'abord, posons

$$\Delta' := \begin{pmatrix} 0 \\ (I-\Delta)_{\neq 1,:} \end{pmatrix}.$$

D'après la proposition 4.5 p.83,

$$0 \leq \Delta' \leq \Delta \Longrightarrow \rho(\Delta') \leq \rho(\Delta).$$

Par conséquent, $\rho(\Delta') < 1$, et $I - \Delta'$ est toujours une M-matrice. Nous pouvons alors considérer l'unique x tel que

$$(I-\Delta')Qx = \begin{pmatrix} |\check{x}_1| \\ (Qb+\delta)_{\neq 1} \end{pmatrix}.$$

De cette façon, x vérifie déjà $(Qx)_1 = |\check{x}_1|$ et $x \in \mathcal{L}_Q^{(1)}$. Nous devons ensuite prouver que $x \in \Sigma \cap Q$. Pour ce faire, nous commençons par prouver que $|\check{x}| \leq |x|$.
Notons M' l'inverse de $(I-\Delta)_{\neq 1,\neq 1}$. D'après le lemme 5.3, M' est aussi positive.

$$\text{On a} \quad \begin{cases} (Qx)_1 &= |\check{x}|_1, \\ (Qx)_{\neq 1} &= M'(Qb+\delta)_{\neq 1} + M'\Delta_{\neq 1,1}|\check{x}|_1. \end{cases}$$

Puisque $\check{x} \in \Sigma \cap Q$, il satisfait (5.29.a), c.a.d. $(I-\Delta)|\check{x}| \leq Qb + \delta$. En particulier,

$$(I-\Delta)_{\neq 1,:}|\check{x}| \leq (Qb+\delta)_{\neq 1}.$$

En faisant passer $\check{x}_1$ à droite, on obtient $(I-\Delta)_{\neq 1,\neq 1}|\check{x}|_{\neq 1} \leq (Qb+\delta)_{\neq 1} + \Delta_{\neq 1,1}|\check{x}|_1$.

En multipliant chaque côté par M' : $|\check{x}|_{\neq 1} \leq M'(Qb+\delta)_{\neq 1} + M'\Delta_{\neq 1,1}|\check{x}|_1$.

On reconnaît $(Qx)_{\neq 1}$ à droite, si bien que $|\check{x}|_{\neq 1} \leq (Qx)_{\neq 1}$ et finalement $|\check{x}| \leq Qx$. Nous avons déjà établi que $x \in \mathcal{L}_Q^{(1)}$, et il reste à prouver que $x \in \Sigma \cap Q$. Pour cela, nous utilisons la caractérisation donnée par le système (5.29).
$\rightarrow$ x satisfait (5.29.a). D'une part, $[(I-\Delta)|x|]_1 = |\check{x}|_1 - \sum_{i=1}^n \Delta_{1i}|x_i|$.
Mais $|\check{x}|_1 - \sum_{i=1}^n \Delta_{1i}|x_i| \leq |\check{x}|_1 - \sum_{i=1}^n \Delta_{1i}|\check{x}|_i = [(I-\Delta)|\check{x}|]_1 \leq [Qb+\delta]_1$.
D'autre part, par définition de x, $[(I-\Delta)|x|]_{\neq 1} = (Qb+\delta)_{\neq 1}$. En rassemblant les deux dernières relations, on obtient les inégalités souhaitées, c.a.d. $(I-\Delta)|x| \leq Qb + \delta$.

$\rightarrow$ x satisfait (5.29.b). En effet, $\check{x} \in \Sigma \cap Q$ implique $(I+\Delta)|\check{x}| \geq Qb - \delta$, qui à son tour implique bien $(I+\Delta)|x| \geq Qb - \delta$.

$\rightarrow$ x satisfait (5.29.c) car $Qx \geq |\check{x}| \geq 0$. □

Proposition 5.12
Soit Q un quadrant

$$\big(\exists w \in \Sigma \cap Q \cap \mathcal{L}_Q^{(k)}\big)\big(\forall x \in \Sigma \cap Q\big)\big(|w_k| \leq |x_k|\big)$$

Preuve. Comme $\Sigma \cap Q$ est un ensemble compact, le minimum de $|x_k|$, pour $x \in \Sigma \cap Q$, est atteint en un point $\check{x}$ de l'ensemble. En appliquant le lemme 5.4 à $\check{x}$, on obtient un point w de la droite $\mathcal{L}_Q^{(k)}$ qui minimise aussi la $k^{ème}$ composante (en valeur absolue) à l'intérieur de $\Sigma \cap Q$. □

Enfin, énonçons une propriété-clef de $\mathcal{L}_Q^{(k)}$ relative à son orientation et à laquelle nous nous référerons au §5.7.6. Informellement, elle signifie qu'en suivant la trajectoire de $\mathcal{L}_Q^{(k)}$ dans le quadrant Q, toutes les composantes augmentent simultanément (en valeur absolue) lorsque $|x_k|$ augmente.

Lemme 5.5 *Soit Q un quadrant, $x \in \mathcal{L}_Q^{(k)}$ et $x' \in \mathcal{L}_Q^{(k)}$.*

$$(Qx)_k \leq (Qx')_k \iff \forall i \neq k\ (Qx)_i \leq (Qx')_i$$

Preuve.

$$\begin{aligned} x \in \mathcal{L}_Q^{(k)} &\iff (I-\Delta)_{\neq k,:}Qx = (Qb+\delta)_{\neq k} \\ &\iff (I-\Delta)_{\neq k,\neq k}(Qx)_{\neq k} = (Qb+\delta)_{\neq k} + (Qx)_k\Delta_{k,:} \\ &\iff \forall i \neq k,\ \ (Qx)_i = (M'(Qb+\delta)_{\neq k})_i + (Qx)_k M'\Delta_{ki}, \end{aligned}$$

où M' dénote l'inverse de $(I-\Delta)_{\neq k,\neq k}$, qui est positive d'après le lemme 5.3. La dernière relation s'obtient en substituant x' à x. Il suffit ensuite de se souvenir que $\Delta_{ki} \geq 0$. □

5.7.5 Les points candidats

Rappelons que nous cherchons le minimum de $|x_k|$ dans le simplexe défini par le système 5.29. Une propriété bien connue de la programmation linéaire est qu'un point optimisant un critère linéaire dans un simplexe se situe toujours à l'un des sommets [Dan63].

Jusqu'ici, nous savons que le point de $\Sigma^{(k+)} \cap Q$ minimisant $|x_k|$ appartient à la droite $\mathcal{L}_Q^{(k)}$. Il en découle que ce point est nécessairement situé à l'intersection de $\mathcal{L}_Q^{(k)}$ et de l'un des $(2n+1)$ hyperplans restants de $\Sigma^{(k+)} \cap Q$ (parmi ceux que donne le système (5.29)).

Nous pouvons déjà mettre de côté l'hyperplan dont l'équation est $(I-\Delta)_{k,:}Qx = (Qb+\delta)_k$, puisque l'intersection de ce dernier avec $\mathcal{L}_Q^{(k)}$ est précisément x_Q, le point maximisant $|x_k|$ (cf. §5.7.2).

Les $2n$ autres points sont appelés *candidats*. Le point $y_Q^{(k,i)}$ obtenu en considérant l'hyperplan défini par la $i^{ème}$ ligne de (5.29.b) est appelé *candidat intérieur*, tandis que le point $z_Q^{(k,i)}$ obtenu par la $i^{ème}$ ligne de (5.29.c) (c.a.d. $(Qx)_i = 0$) est appelé *candidat limitrophe*.

Définition 5.5 (Candidat intérieur)
Soit Q un quadrant, $i \in [1..n]$, et

$$\Lambda_i := \begin{pmatrix} (I-\Delta)_{<k,:} \\ (I+\Delta)_{i,:} \\ (I-\Delta)_{>k,:} \end{pmatrix}$$

Si Λ_i est régulière, on appelle **candidat intérieur** *et on note $y_Q^{(k,i)}$ le point tel que*

$$Qy_Q^{(k,i)} := \Lambda_i^{-1} \begin{pmatrix} (Qb+\delta)_{<k} \\ (Qb-\delta)_i \\ (Qb+\delta)_{>k} \end{pmatrix} \tag{5.32}$$

Ainsi, $y_Q^{(k,i)}$ se situe à l'intersection de $\mathcal{L}_Q^{(k)}$ et de l'hyperplan

$$\{x \mid (I+\Delta)_{i,:}Qx = (Qb-\delta)_i\}.$$

Nous définissons de façon similaire les *candidats limitrophes.*

Définition 5.6 (Candidat limitrophe)
Soit Q un quadrant, $i \in [1..n]$, et

$$\Gamma_i := \begin{pmatrix} (I-\Delta)_{<k,:} \\ I_{i,:} \\ (I-\Delta)_{>k,:} \end{pmatrix}$$

Si Γ_i est régulière, on appelle **candidat limitrophe** *et note $z_Q^{(k,i)}$ le point tel que*

$$Qz_Q^{(k,i)} := \Gamma_i^{-1} \begin{pmatrix} (Qb+\delta)_{<k} \\ 0 \\ (Qb+\delta)_{>k} \end{pmatrix} \tag{5.33}$$

De nouveau, $z_Q^{(k,i)}$ est situé à l'intersection de $\mathcal{L}_Q^{(k)}$ et de l'hyperplan $\{x \mid (Qx)_i = 0\}$.

Bien évidemment, si un hyperplan est parallèle à $\mathcal{L}_Q^{(k)}$, le candidat correspondant n'existe pas. Cela ne change en rien le fait que le minimum est atteint en l'un des autres points candidats. Il est utile de remarquer que les candidats $y_Q^{(k,k)}$ et $z_Q^{(k,k)}$ existent toujours puisque Λ_k et Γ_k sont régulières. Cela signifie que, dans le "pire" des cas, le minimum se trouve parmi eux deux.

Nous montrerons au §5.7.10 la relation suivante (cf. proposition 5.17) :

$$\forall i \neq k \qquad \Lambda_i \text{ est régulière} \iff \Gamma_i \text{ est régulière} \iff m_{ik} \neq 0.$$

Candidats intérieurs

La question qui se pose désormais est : "pour un quadrant Q quelconque, peut-on déterminer la $k^{\text{ème}}$ composante de $Qy_Q^{(k,i)}$?". La proposition suivante fournit la réponse.

Proposition 5.13 *Soit Q un quadrant. On a :*

$$(Qy_Q^{(k,k)})_k = \nu_k \left[m_{kk}(Qb-\delta)_k - \sum_{j\neq k}^{n} m_{kj}(Qb+\delta)_j \right] \quad \textit{avec} \quad \nu_k = \frac{1}{2m_{kk}-1}$$

Et pour $i \neq k$, si $m_{ik} \neq 0$ alors

$$(Qy_Q^{(k,i)})_k = \alpha_{ki}(Qb)_i + \sum_{j\neq k}^{n} (m_{kj} - m_{ij}\alpha_{ki})(Qb+\delta)_j \quad \textit{avec} \quad \alpha_{ki} = m_{kk}/m_{ik}$$

Preuve. Pour des raisons de clarté, D_i représentera dans la suite $e_k e_i^T$, c.a.d. la matrice vérifiant $(D_i)_{ki} := 1$ et contenant des zéros partout ailleurs. On a $\Lambda_i = (I - D_k)(I - \Delta) + D_i(I + \Delta) = (I - D_k - D_i)(I - \Delta) + 2D_i$.

Comme noté précédemment, Λ_k est régulière. De plus, si $i \neq k$, $m_{ik} \neq 0$ implique avec la proposition 5.17 que Λ_i est régulière. In fine, nous pouvons écrire $\Lambda_i^{-1}\Lambda_i = I$, c.a.d. $\Lambda_i^{-1}\Lambda_i M = M$.

De là, $\Lambda_i^{-1}\left[(I - D_k - D_i)(I - \Delta) + 2D_i\right] M = M$; et puisque $(I - \Delta)M = I$, on obtient

$$\Lambda_i^{-1}\left[(I - D_k - D_i) + 2D_i M\right] = M. \tag{5.34}$$

Utilisons λ comme symbole pour désigner les coefficients de Λ^{-1}. A partir de (5.34), il est aisé de calculer la $k^{\text{ème}}$ ligne de Λ_i^{-1}.
Pour $i = k$, on obtient :
• $\lambda_{kk}(2m_{kk} - 1) = m_{kk} \Longrightarrow \lambda_{kk} = (m_{kk}/(2m_{kk} - 1)) = m_{kk}\nu_k$.
• $2m_{kj}\lambda_{kk} + \lambda_{kj} = m_{kj} \Longrightarrow \lambda_{kj} = m_{kj}(1 - 2\lambda_{kk}) = m_{kj}(1 - 2m_{kk}\nu_k) = -m_{kj}\nu_k$.

Si bien que la projection de (5.32) sur la $k^{\text{ème}}$ coordonnée donne :

$$\begin{aligned}(Qy_Q^{(k,k)})_k &= \lambda_{kk}(Qb - \delta)_k + \textstyle\sum_{j\neq k}^n \lambda_{kj}(Qb + \delta)_j \\ &= \nu_k\left[m_{kk}(Qb - \delta)_k - \textstyle\sum_{j\neq k}^n m_{kj}(Qb + \delta)_j\right].\end{aligned}$$

Avec $i \neq k$, on obtient :
• $2\lambda_{kk}m_{ik} = m_{kk} \Longrightarrow \lambda_{kk} = 1/2(m_{kk}/m_{ik}) = 1/2\alpha_{ki}$.
• $2m_{ij}\lambda_{kk} + \lambda_{kj} = m_{kj} \Longrightarrow \lambda_{kj} = m_{kj} - (m_{ij}m_{kk})/m_{ik} = m_{kj} - m_{ij}\alpha_{ki}$.
• $(2m_{ii} - 1)\lambda_{kk} + \lambda_{ki} = m_{ki} \Longrightarrow \lambda_{ki} = m_{ki} - (m_{ii}m_{kk})/m_{ik} + 1/2(m_{kk}/m_{ik}) = m_{ki} - m_{ii}\alpha_{ki} + 1/2\alpha_{ki}$.

Le même raisonnement qu'au dessus produit :

$$\begin{aligned}(Qy_Q^{(k,i)})_k &= \lambda_{kk}(Qb - \delta)_i + \lambda_{ki}(Qb + \delta)_i + \textstyle\sum_{j\neq k, j\neq i}^n \lambda_{kj}(Qb + \delta)_j \\ &= (\lambda_{kk} + \lambda_{ki})(Qb)_i + (\lambda_{ki} - \lambda_{kk})\delta_i + \textstyle\sum_{j\neq k, j\neq i}^n \lambda_{kj}(Qb + \delta)_j \\ &= \alpha_{ki}(Qb)_i + \textstyle\sum_{j\neq k}^n (m_{kj} - m_{ij}\alpha_{ki})(Qb + \delta)_j.\end{aligned}$$

□

Remarque 5.4 *En particulier, nous avons :*

$$(Qy_Q^{(k,i)})_k = (\alpha_{ki} + m_{ki} - m_{ii}\alpha_{ki})(Qb)_i + \sum_{\substack{j\neq k\\ j\neq i}}^n (m_{kj} - m_{ij}\alpha_{ki})(Qb)_j + \phi(\delta),$$

où $\phi(\delta)$ est une expression que ne dépend pas de Q.

Dans le corollaire suivant, nous réintroduisons x_Q, tel qu'il a été défini au §5.7.2.

Corollaire 5.3 *Soit Q un quadrant. Soit S une matrice diagonale remplie de 1 sauf pour $S_{kk} := -1$ (cela signifie que (SQ) est le quadrant symétrique de Q par rapport à $(x_k = 0)$). On a :*

$$(Qy_Q^{(k,k)})_k = -\nu_k((SQ)x_{(SQ)})_k.$$

Preuve. Il suffit de comparer l'expression obtenue à la proposition 5.13 avec la définition de $x_{(SQ)}$ (cf. §5.7.2).□

Corollaire 5.4 *Soient Q un quadrant, $x \in \mathcal{L}_Q^{(k)}$ et $i \in [1..n]$ tel que $m_{ik} \neq 0$.*

$$(Qx)_k \geq (Qy_Q^{(k,i)})_k \quad \Longrightarrow \quad (I+\Delta)_{i,:}Qx \geq (Qb-\delta)_i.$$

Preuve. De par le lemme 5.5, $(Qx)_k \geq (Qy_Q^{(k,i)})_k$ implique $Qx \geq Qy_Q^{(k,i)}$. Combiner cette inégalité avec $I+\Delta \geq 0$ et $(I+\Delta)_{i,:}Qy_Q^{(k,i)} = (Qb-\delta)_i$ aboutit au résultat escompté.□

Candidats limitrophes

Les formules explicites des candidats limitrophes se déduisent de façon analogue, et avec des calculs plus simples. La preuve est donc plus concise.

Proposition 5.14 *Soit Q un quadrant, $i \in [1..n]$. Si $m_{ik} \neq 0$, alors*

$$(Qz_Q^{(k,i)})_k = (Qy_Q^{(k,i)})_k - \alpha_{ki}(Qb)_i.$$

Preuve. On a $\Gamma_i = (I-D_k)(I-\Delta)+D_i$, si bien que $\Gamma_i^{-1}((I-D_k)+D_iM) = M$. Utilisons γ comme symbole pour les coefficients de Γ_i^{-1}. On en déduit d'une part $\gamma_{kk} = m_{kk}/m_{ik} = \alpha_{ki}$. D'autre part, $\forall j \neq k, \quad \gamma_{kk}m_{ij} + \gamma_{kj} = m_{kj}$, c.a.d. $\gamma_{kj} = m_{kj} - (m_{kk}m_{ij})/m_{ik}$. Finalement, on a $(Qz_Q^{(k,i)})_k = \sum_{j\neq k}(m_{kj} - m_{ij}\alpha_{ki})(Qb+\delta)_j = (Qy_Q^{(k,i)})_k - \alpha_{ki}(Qb)_i$. □

5.7.6 Minimum local

Maintenant que nous avons caractérisé les candidats intérieurs et limitrophes, il est plus confortable de ne considérer plus qu'un seul ensemble de (au plus $2n$) candidats possibles. Définissons :

$$\forall i \in [1..n], m_{ik} \neq 0, \quad \left\{ \begin{array}{ll} c_Q^{(k,i)} & := y_Q^{(k,i)} \\ c_Q^{(k,n+i)} & := z_Q^{(k,i)} \end{array} \right.$$

Comme noté plus haut, $y_Q^{(k,k)}$ et $z_Q^{(k,k)}$ appartiennent toujours à cet ensemble, qui n'est donc jamais vide.

Introduisons également un ensemble d'indices approprié :

$$\mathcal{I} := \bigcup_{i\in[1..n], m_{ik}\neq 0} \{i, (n+i)\}$$

Lemme 5.6 (Minimum local) *Soit Q un quadrant.*

$$\Sigma \cap Q \neq \emptyset \quad \Longrightarrow \quad \inf_{x\in\Sigma\cap Q} |x_k| = \max_{i\in\mathcal{I}} (Qc_Q^{(k,i)})_k$$

Preuve. Soit c un candidat quelconque, et supposons qu'il existe $x \in \mathcal{L}_Q^{(k)}$ tel que $(Qx)_k < (Qc)_k$. Nous allons prouver que $x \notin \Sigma\cap Q$. Ainsi, si $\Sigma\cap Q \neq \emptyset$, $\inf_{x\in\Sigma\cap Q}|x_k|$ est bien obtenu en maximisant $(Qc)_k$, pour c décrivant l'ensemble des points candidats (voir figure 5.5).
Prouvons maintenant $x \notin \Sigma \cap Q$. Premièrement, $(Qx)_k < (Qc)_k$ implique $Qx \leq Qc$, en vertu du lemme 5.5. Supposons maintenant que c soit un candidat limitrophe. Il existe i tel que $(Qc)_i = 0$ (cf. définition 5.6).

Comme $Qx \leq Qc$ implique $(Qx)_i \leq (Qc)_i$, alors $(Qx)_i \leq 0$. Mais $(Qx)_i = 0$ impliquerait[8] $x = c$, contredisant $(Qx)_k < (Qc)_k$. Par conséquent, $(Qx)_i < 0$ et $x \notin \Sigma \cap Q$. Similairement, si c est un candidat intérieur, il existe i tel que $(I + \Delta)_{i,:}Qc = (Qb - \delta)_i$ (cf. définition 5.5). $Qx \leq Qc$ implique $(I + \Delta)_{i,:}Qx \leq (I + \Delta)_{i,:}Qc$, et donc $(I + \Delta)_{i,:}Qx \leq (Qb - \delta)_i$. Mais cette dernière inégalité est en fait stricte puisque $x \neq c$ [9]. De nouveau, $x \notin \Sigma \cap Q$.□

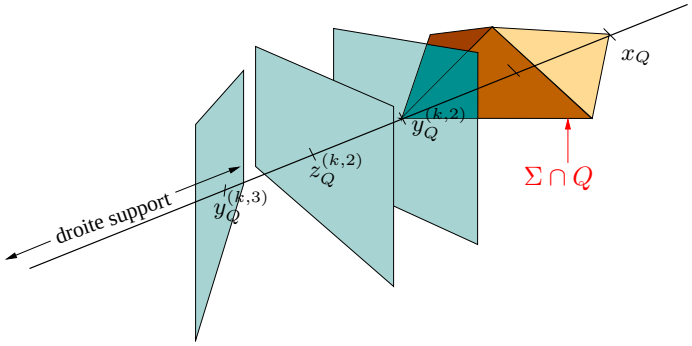


FIG. *5.5: Candidat minimisant $|x_k|$ pour un quadrant fixé*

5.7.7 Minimum global

Dans le paragraphe précédent, nous avons donné un moyen de calculer le minimum de $|x_k|$ parmi les candidats, pour un quadrant fixé. Ici, nous faisons l'opération inverse, c.a.d. nous calculons le minimum d'un candidat fixé (défini par une matrice Λ_i ou Γ_i)[10] parmi l'ensemble des quadrants.

On définit

$$\mathcal{O}^+ := \{Q \text{ quadrant de } \mathbb{R}^n \mid Q_{kk} = +1\}$$
$$\mathcal{O}^- := \{Q \text{ quadrant de } \mathbb{R}^n \mid Q_{kk} = -1\}$$

Lemme 5.7 (Meilleur quadrant pour un candidat)

$$\begin{array}{lll} \forall i \in \mathcal{I} & \min_{Q\in\mathcal{O}^+}(Qc_Q^{(k,i)})_k = (Q^{(k+)}c_{Q^{(k+)}}^{(k,i)})_k & \\ \textit{Plus précisément, } \forall i \in [1,n] \textit{ tel que } i \neq k \textit{ et } m_{ik} \neq 0 & & \\ (i) & \min_{Q\in\mathcal{O}^+}(Qy_Q^{(k,i)})_k = \theta_{ki} & \textit{(cf. §5.7.1)} \\ (ii) & \min_{Q\in\mathcal{O}^+}(Qz_Q^{(k,i)})_k = \theta_{ki} - \alpha_{ki}|b_i| & \\ \textit{Et} & & \\ (iii) & \min_{Q\in\mathcal{O}^+}(Qy_Q^{(k,k)})_k = \nu_k x_k & \\ (iv) & \min_{Q\in\mathcal{O}^+}(Qz_Q^{(k,k)})_k = 0 & \end{array}$$

Preuve.
Pour un candidat $c_Q^{(k,i)}$, la fonction $Qb \longrightarrow (Qc_Q^{(k,i)})_k$ est une forme affine (cf. remarque 5.4 après la proposition 5.13) avec, excepté pour $(Qb)_k$, des coefficients négatifs (cf. proposition 5.18). Par conséquence, le minimum de $\mathcal{O}^+$ est atteint lorsque toutes les composantes de $(Qb)_{\neq k}$ sont maximisées, c.a.d. lorsque $Q = Q^{(k+)}$. Plus

[8] $\mathcal{L}_Q^{(k)}$ n'est pas parallèle à $(x_i = 0)$ puisque $i \in \mathcal{I}$ signifie que Γ_i est régulière.
[9] $\mathcal{L}_Q^{(k)}$ n'est pas parallèle à $((I + \Delta)_{i,:}Qx = (Qb - \delta)_i)$ puisque $i \in \mathcal{I}$ signifie que Λ_i est régulière.
[10] Il se doit d'être précisé que ni Λ_i ni Γ_i ne dépend du quadrant. En d'autres termes, pour n'importe quels quadrants Q et Q', $y_Q^{(k,i)}$ existe ssi $y_{Q'}^{(k,i)}$ existe. L'énoncé du lemme 5.7 est donc bien cohérente.

précisément :
(i) et (ii). D'après les propositions 5.13 & 5.14, et par définition de θ_{ki} :

$$\theta_{ki} = (Q^{(k+)} y^{(k,i)}_{Q^{(k+)}})_k \quad \text{et} \quad \theta_{ki} - \alpha_{ki}|b_i| = (Q^{(k+)} z^{(k,i)}_{Q^{(k+)}})_k.$$

(iii). $\min_{Q\in\mathcal{O}^+} Qy^{(k,k)}_Q$
$= \min_{Q\in\mathcal{O}^+} -\nu_k((SQ)x_{(SQ)})_k$ d'après le corollaire 5.3,
$= -\nu_k \max_{Q\in\mathcal{O}^+}((SQ)x_{(SQ)})_k$ car $\nu_k \geq 0$ (cf. lemme 4.4 p.105),
$= -\nu_k \max_{Q\in\mathcal{O}^-}(Qx_Q)_k$
$= \nu_k \underline{x}_k$ d'après le corollaire 5.2.

(iv) est évident.

□

Tous les éléments sont maintenant réunis pour pouvoir fournir le minimum global de $|x_k|$ dans $\Sigma^{(k+)}$.

Proposition 5.15

$$\Sigma^{(k+)} \neq \emptyset \Longrightarrow \min_{x\in\Sigma^{(k+)}} x_k = \max\{0, \nu_k \underline{x}_k, \hat{x}_k\}$$

Preuve. Supposons $\Sigma^{(k+)} \neq \emptyset$. Une propriété min-max bien connue (que nous avons déjà exploitée p.55) permet d'écrire

$$\max_{i\in\mathcal{I}} \min_{Q\in\mathcal{O}^+} (Qc^{(k,i)}_Q)_k \leq \min_{Q\in\mathcal{O}^+} \max_{i\in\mathcal{I}} (Qc^{(k,i)}_Q)_k. \tag{5.35}$$

D'autre part, grâce au lemme 5.6 :

$$\min_{Q\in\mathcal{O}^+} \max_{i\in\mathcal{I}} (Qc^{(k,i)}_Q)_k = \min_{Q\in\mathcal{O}^+} \min_{x\in\Sigma\cap Q} |x_k| = \min_{x\in\Sigma^{(k+)}} |x_k| = \min_{x\in\Sigma^{(k+)}} x_k, \tag{5.36}$$

et grâce au lemme 5.7 :

$$\max_{i\in\mathcal{I}} \min_{Q\in\mathcal{O}^+} (Qc^{(k,i)}_Q)_k = \max_{i\in\mathcal{I}} (Q^{(k+)} c^{(k,i)}_{Q^{(k+)}})_k. \tag{5.37}$$

En combinant (5.35), (5.36) et (5.37), on obtient :

$$\max_{i\in\mathcal{I}} (Q^{(k+)} c^{(k,i)}_{Q^{(k+)}})_k \leq \min_{x\in\Sigma^{(k+)}} x_k. \tag{5.38}$$

Par ailleurs, d'après la proposition 5.11, $\Sigma^{(k+)} \neq \emptyset \Longrightarrow Q^{(k+)} \neq \emptyset$, et avec le lemme 5.6,

$$\max_{i\in\mathcal{I}} (Q^{(k+)} c^{(k,i)}_{Q^{(k+)}})_k = \min_{x\in\Sigma\cap Q^{(k+)}} x_k. \tag{5.39}$$

En combinant (5.38) et (5.39), on obtient :

$$\max_{i\in\mathcal{I}} (Q^{(k+)} c^{(k,i)}_{Q^{(k+)}})_k = \min_{x\in\Sigma^{(k+)}} x_k. \tag{5.40}$$

De nouveau grâce au lemme 5.7, on en déduit :

$$\max_{i\in\mathcal{I}} (Q^{(k+)} c^{(k,i)}_{Q^{(k+)}})_k = \max\{\max_{i\in\mathcal{I}} \theta_{ki},\ \max_{i\in\mathcal{I}} (\theta_{ki} - \alpha_{ki}|b_i|),\ \nu_k \underline{x}_k,\ 0\},$$

où $\mathcal{I} := \{i \in [1..n], i \neq k \mid \ m_{ik} \neq 0\}$. Mais du fait que $\forall i \in \mathcal{I}$, $\theta_{ki} - \alpha_{ki}|b_i| \leq \theta_{ki}$ (car $\alpha_{ki} \geq 0$), cela peut se simplifier en :

$$\max_{i\in\mathcal{I}} (Q^{(k+)} c^{(k,i)}_{Q^{(k+)}})_k = \max\{0, \nu_k \underline{x}_k, \hat{x}_k\}. \tag{5.41}$$

Finalement, en regroupant (5.40) et (5.41) :

$$\max\{0, \nu_k \underline{x}_k, \hat{x}_k\} = \min_{x\in\Sigma^{(k+)}} x_k.$$

□

Corollaire 5.5

$$\Sigma^{(k-)} \neq \emptyset \Longrightarrow \max_{x \in \Sigma^{(k-)}} x_k = \min\{0, \nu_k \tilde{x}_k, -\hat{x}_k\}$$

Preuve. On considère $\Sigma(\mathbf{A}, -\mathbf{b})$ et on applique la proposition 5.15. □

Nous finissons ce paragraphe par une petite astuce. Il peut être noté que 0 n'apparaît pas comme candidat dans la partie "else" de l'algorithme donné au §5.7.1. Cet omission est justifiée par le fait suivant : dès lors que 0 appartient à $[\underline{x}_k, \overline{x}_k]$ (la $k^{ème}$ composante de $\square\,\Sigma$), on a

$$(\square\,\Sigma) \cap \Sigma^{(k+)} \neq \emptyset \quad \text{et} \quad (\square\,\Sigma) \cap \Sigma^{(k-)} \neq \emptyset,$$

si bien que $[\underline{x}_k, \overline{x}_k] = [\underline{x}_k, \tilde{x}_k]$. En conséquence, 0 peut être écarté dans tous les autres cas de figure.

5.7.8 Existence de solutions

Jusqu'ici, nous avons prouvé que les bornes de $\square\,\Sigma^{(k+)}$ que nous calculons sont correctes. Mais nous avons toujours admis $\Sigma^{(k+)} \neq \emptyset$. Dans ce paragraphe, nous montrons que $(\tilde{x}_k \geq \max\{0, \nu_k \underset{\sim}{x}_k, \hat{x}_k\})$ est une condition suffisante pour que des solutions existent dans $\Sigma^{(k+)}$, pourvu que cette même condition ou son pendant négatif soit également vérifiée pour toutes les autres composantes.

Lemme 5.8 $\forall k \in [1..n]$,

Si $b_k \geq 0$ **alors** $\tilde{x}_k < \max\{0, \nu_k \underset{\sim}{x}_k\} \Longrightarrow \underset{\sim}{x}_k > 0$
sinon $\underset{\sim}{x}_k > \min\{0, \nu_k \tilde{x}_k\} \Longrightarrow \tilde{x}_k < 0$

Preuve. Tout d'abord, il découle des expressions de $\tilde{x}_k$ et de $\underset{\sim}{x}_k$ que $\underset{\sim}{x}_k = -\tilde{x}_k + 2m_{kk}b_k$ quel que soit le signe de b_k.
Supposons $b_k \geq 0$. Si $\tilde{x}_k < 0$ alors $-\tilde{x}_k > 0$ d'où $\underset{\sim}{x}_k > 0$. Si $\tilde{x}_k < \nu_k \underset{\sim}{x}_k$ alors $-\tilde{x}_k > -\nu_k \underset{\sim}{x}_k$ d'où $\underset{\sim}{x}_k > -\nu_k \underset{\sim}{x}_k + 2m_{kk}b_k$. Il s'ensuit que $(1+\nu_k)\underset{\sim}{x}_k > 2m_{kk}b_k$, ce qui de nouveau implique $\underset{\sim}{x}_k > 0$ ($m_{kk} \geq 0$). Ainsi, $\tilde{x}_k < \max\{0, \nu_k \underset{\sim}{x}_k\} \Longrightarrow \underset{\sim}{x}_k > 0$.
Supposons $b_k < 0$. Si $\underset{\sim}{x}_k > 0$ alors $-\underset{\sim}{x}_k < 0$ donc $\tilde{x}_k < 0$. Si $\underset{\sim}{x}_k > \nu_k \tilde{x}_k$ alors $-\tilde{x}_k + 2m_{kk}b_k > \nu_k \tilde{x}_k$. Il s'ensuit que $(1+\nu_k)\tilde{x}_k < 2m_{kk}b_k$, ce qui de nouveau implique $\tilde{x}_k < 0$. Ainsi $\underset{\sim}{x}_k > \max\{0, \nu_k \tilde{x}_k\} \Longrightarrow \tilde{x}_k < 0$. □

Avant de donner une condition suffisante pour que $\Sigma^{(k+)}$ ne soit pas vide, nous devons introduire une condition suffisante pour que $\Sigma \cap Q^*$ ne soit pas vide (la définition de Q^* est donnée au §5.7.2).

Lemme 5.9

$$\Sigma \cap Q^* \neq \emptyset \Longleftrightarrow \big(\forall k \in [1..n]\big) \quad \tilde{x}_k \geq \max\{0, \nu_k \underset{\sim}{x}_k\} \textbf{ ou } \underset{\sim}{x}_k \leq \min\{0, \nu_k \tilde{x}_k\}$$

Preuve. ($\Longrightarrow$) Supposons $\Sigma \cap Q^* \neq \emptyset$. Si $Q^*_{kk} = 1$, alors $\Sigma^{(k+)}$ est non vide. Il vient alors avec la proposition 5.15 que 0 et $\nu_k \underset{\sim}{x}_k$ sont tous deux plus petits que $\min\{x_k, x \in \Sigma^{(k+)}\}$, et avec le corollaire 5.1 que $\max\{x_k, x \in \Sigma^{(k+)}\}$ est $\tilde{x}_k$. Ainsi, $\tilde{x}_k \geq \max\{0, \nu_k \underset{\sim}{x}_k\}$. Si $Q^*_{kk} = -1$, un argument similaire mène à $\underset{\sim}{x}_k \leq \min\{0, \nu_k \tilde{x}_k\}$.
($\Longleftarrow$) Nous prouvons que $x_{Q^*} \in \Sigma$. Tout d'abord, x_{Q^*} satisfait (5.29.a) par définition. Maintenant, prenons $k \in [1..n]$ quelconque, et souvenons-nous que x^* est un raccourci pour désigner $Q^* x_{Q^*}$ (cf. §5.7.2). Si $b_k \geq 0$ alors $x^*_k = \tilde{x}_k$. D'une part, $\tilde{x}_k \geq 0$ (car $\tilde{x}_k < 0$ impliquerait $\underset{\sim}{x}_k > 0$ d'après le lemme 5.8), d'où $x^*_k \geq 0$; d'autre part, $\tilde{x}_k \geq \nu_k \underset{\sim}{x}_k$ (pour la même raison). Mais d'après le lemme 5.7, $\nu_k \underset{\sim}{x}_k = (Q^* y^{(k,k)}_{Q^*})_k$ (puisque $Q^* = Q^{(k+)}$), donc on a $x^*_k \geq (Q^* y^{(k,k)}_{Q^*})_k$ ce qui implique $(I + \Delta)_{k,:} x^* \geq (Q^* b - \delta)_k$, d'après le corollaire 5.4. Si $b_k \leq 0$,

$x_k^* = -\underline{x}_k$. Alors, $\underline{x}_k \leq 0$ implique $x_k^* \geq 0$ et $\underline{x}_k \leq \nu_k \tilde{x}_k$ implique $-\underline{x}_k \geq (Q^* y_{Q^*}^{(k,k)})_k$. Nous concluons de la même façon. Nous avons prouvé que $x_k^* \geq 0$ et $(I+\Delta)_{k,:}x^* \geq (Q^*b - \delta)_k$ pour tout $k \in [1..n]$. Ainsi, x_{Q^*} satisfait (5.29.b) et (5.29.c). Finalement $x_{Q^*} \in \Sigma$, et $\Sigma \cap Q^* \neq \emptyset$. □

Nous avons maintenant les outils pour pouvoir établir la condition suffisante rendant $\Sigma^{(k+)}$ non vide.

Proposition 5.16

$$\Sigma^{(k+)} \neq \emptyset \Longleftrightarrow \left\{ \begin{array}{l} \tilde{x}_k \geq \max\{0, \nu_k \underline{x}_k, \hat{x}_k\} \\ (\forall i \neq k) \quad \tilde{x}_i \geq \max\{0, \nu_i \underline{x}_i\} \textbf{ ou } \underline{x}_i \leq \min\{0, \nu_i \tilde{x}_i\} \end{array} \right.$$

Preuve. Le sens ($\Longrightarrow$) s'obtient aisément en adaptant la première partie de la preuve du lemme précédent. Pour le sens ($\Longleftarrow$) nous prouvons que $x_{Q^{(k+)}}$ satisfait le système (5.29). Pour la lisibilité, posons $t := Q^{(k+)} x_{Q^{(k+)}}$ et notons que $x_{Q^*} \in \Sigma \cap Q^*$ d'après le lemme 5.9. Tout d'abord, $x_{Q^{(k+)}}$ satisfait (5.29.a) par définition. Ensuite, $\forall i \neq k$, $t_i = x_i^*$ et $x_i^* \geq 0$ puisque $x_{Q^*} \in \Sigma \cap Q^*$. De plus, $t_k = \tilde{x}_k$ de par le corollaire 5.1 et $\tilde{x}_k \geq 0$ par hypothèse. Ainsi $t \geq 0$, c.a.d. $x_{Q^{(k+)}}$ satisfait (5.29.c). Il reste à prouver que $x_{Q^{(k+)}}$ satisfait (5.29.b) :

• Considérons $i \neq k$.

→ Supposons d'abord $m_{ik} \neq 0$. Cela signifie que le candidat intérieur $y_{Q^{(k+)}}^{(k,i)}$ existe (cf. prop. 5.17). On a $t_k \geq \hat{x}_k$ par hypothèse, et $\hat{x}_k \geq \theta_{ki} = (Q^{(k+)} y_{Q^{(k+)}}^{(k,i)})_k$ par définition. Grâce au corollaire 5.4 il vient alors $(I+\Delta)_{i,:}t \geq (Q^{(k+)}b - \delta)_i$.

→ Supposons maintenant $m_{ik} = 0$. Il s'avère que les expressions pour $(Q^{(k+)} y_{Q^{(k+)}}^{(i,i)})_i$ et $(Q^* y_{Q^*}^{(i,i)})_i$ (cf. prop. 5.13) ne diffèrent que par leurs termes impliquant respectivement $Q_{kk}^{(k+)}$ et Q_{kk}^*. Comme ces termes ont pour facteur m_{ik}, il sont tous les deux nuls. Par conséquence, ces deux expressions coïncident. De plus, $t_i = x_i^*$ et $x_i^* \geq (Q^* y_{Q^*}^{(i,i)})_i$ puisque $x_{Q^*} \in \Sigma \cap Q^*$. Donc $t_i \geq (Q^{(k+)} y_{Q^{(k+)}}^{(i,i)})_i$, et nous pouvons de nouveau appliquer le corollaire 5.4. Nous obtenons encore $(I+\Delta)_{i,:}t \geq (Q^{(k+)}b - \delta)_i$.

• On a $t_k \geq \nu_k \underline{x}_k$ par hypothèse, et $\nu_k \underline{x}_k = (Q^{(k+)} y_{Q^{(k+)}}^{(k,k)})$ d'après le lemme 5.7. Donc, toujours avec le corollaire 5.4, nous obtenons $(I+\Delta)_{k,:}t \geq (Q^{(k+)}b - \delta)_k$. □

Le pendant négatif de la proposition précédente est :

Corollaire 5.6

$$\Sigma^{(k-)} \neq \emptyset \Longleftrightarrow \left\{ \begin{array}{l} \underline{x}_k \leq \min\{0, \nu_k \tilde{x}_k, -\hat{x}_k\} \\ (\forall i \neq k) \quad \tilde{x}_i \geq \max\{0, \nu_i \underline{x}_i\} \textbf{ ou } \underline{x}_i \leq \min\{0, \nu_i \tilde{x}_i\} \end{array} \right.$$

5.7.9 Application

Vérifions la validité de notre algorithme sur un exemple, et comparons le résultat obtenu avec celui que fournissent d'autres méthodes.

Prenons $\mathbf{A}x = \mathbf{b}$, avec $\mathbf{A} = I \pm \Delta$ et :

$$\Delta = \begin{pmatrix} .1 & .1 & .1 & .1 & .1 \\ .1 & .2 & .1 & .1 & .1 \\ .2 & .3 & .1 & .2 & .2 \\ .1 & .4 & .1 & .1 & .1 \\ .1 & .5 & .1 & .1 & .1 \end{pmatrix}$$

$$\mathbf{b} = ([1,7] \quad [-4,-10] \quad [8,-6] \quad [9,8] \quad [2,-10])^T.$$

Le vecteur de quantificateurs pour **b** est donc :

$$(\exists, \forall, \forall, \forall, \forall)^T.$$

Dans le tableau ci-dessous, pour $k = 1..5$, nous montrons le résultat obtenu pour la $k^{ème}$ composante de $\square\,\Sigma$. Dans la seconde colonne, les bornes sont données. Dans la troisième colonne, nous indiquons quels sont les candidats qui forment les bornes. Les deux colonnes suivantes montrent le résultat fourni par l'opérateur de Gauss-Seidel généralisé (cf. §5.3) et l'opérateur de Krawczyk généralisé (cf. §5.4).

k	résultat numérique	résultat formel	Gauss-Seidel	Krawczyk
1	[4.71053,11.8576]	$[\theta_{13} \quad , \tilde{x}_1]$	[-2.96871,11.8576]	[-3.8576,11.8576]
2	[-9.84177,-6.09412]	$[x_2 \quad , -\theta_{25}]$	[-9.84177,-5.10549]	[-9.84177,-4.15830]
3	[-1.36076,4.27215]	$[x_3 \quad , \tilde{x}_3]$	[-2.04993,4.27215]	[-2.27215,4.27215]
4	[8.09474,15.81013]	$[\theta_{43} \quad , \tilde{x}_4]$	[2.51899,15.81013]	[1.18987,15.81013]
5	[-6.7943,-2.53322]	$[x_5 \quad , \nu_5\tilde{x}_5]$	[-6.7943,-1.71375]	[-6.7943,-1.2057]

Ajoutons que, comme prévu, la méthode exhaustive (cf. §5.5) fournit des bornes[11] qui coïncident avec la colonne "résultat numérique".

Il peut être observé au travers de la colonne "résultat formel" que n'importe quel candidat peut être le "bon", ce qui veut dire qu'il est peu probable qu'une formule plus simple que la nôtre puisse être donnée pour $\square\,\Sigma$.

5.7.10 Propriétés annexes

Dans ce paragraphe, nous fixons $i \neq k$, et définissons la matrice Δ' ainsi :

$$\Delta' := (I - D_k + D_i)\Delta \quad \text{avec } D_k = e_k e_k^T \text{ et } D_i = e_k e_i^T.$$

Δ' est donc une copie de Δ à l'exception de la $k^{ème}$ ligne, qui est une duplication de la $i^{ème}$ ligne. Comme $(I - \Delta)$ est une M-matrice, il existe $u > 0$ tel que $(I - \Delta)u > 0$. Chaque ligne de $I - \Delta'$ étant aussi une ligne de $I - \Delta$, on a alors également $(I - \Delta')u > 0$ et par conséquent $I - \Delta'$ est aussi une M-matrice. On pose alors

$$M' := (I - \Delta')^{-1}.$$

Lemme 5.10

$$m'_{kk} = m'_{ik} + 1.$$

Preuve. On a $(I - \Delta')M' = I$, d'où $M' = I + \Delta' M'$ et $M' - I = \Delta' M'$. De là, $\Delta'_{k,:} = \Delta'_{i,:} \Longrightarrow (M' - I)_{k,:} = (M' - I)_{i,:}$, et en particulier $m'_{kk} = m'_{ik} + 1$. □

Lemme 5.11

$$m'_{kk} = \frac{m_{kk}}{m_{kk} - m_{ik}}$$

$$m'_{ki} = m_{ki} - m_{kk}\frac{1 + m_{ki} - m_{ii}}{m_{kk} - m_{ik}}$$

$$\forall j \neq k,\ j \neq i,\ m'_{kj} = m_{kj} - m_{kk}\frac{m_{kj} - m_{ij}}{m_{kk} - m_{ik}}$$

[11] Les 2^5 simplexes dérivés de la proposition 5.7 ont été résolus via la méthode du Simplexe sous Maple.

Preuve. On a :

$$\begin{aligned}
M'(I-\Delta') = I &\iff M'(I-(I-D_k+D_i)\Delta) = I \\
&\iff M'(I-\Delta+(D_k-D_i)\Delta)M = M \\
&\iff M'((I-\Delta)-(D_k-D_k\Delta)+(D_i-D_i\Delta)+(D_k-D_i))M = M \\
&\iff M'(I-D_k+D_i+(D_k-D_i)M) = M.
\end{aligned}$$

La relation précédente donne par exemple pour $k=1$:

$$M'\begin{pmatrix} (m_{11}-m_{i1}) & (m_{12}-m_{i2}) & \dots & (1+m_{1i}-m_{ii}) & \dots & (m_{1n}-m_{in}) \\ & & & & & \\ 0_{n-1} & & & I_{n-1} & & \end{pmatrix} = M.$$

Les coefficients de M' peuvent alors être calculés directement à partir de cette relation. On a par exemple $m'_{kk}(m_{kk}-m_{ik}) = m_{kk}$, ce qui donne $m'_{kk} = m_{kk}/(m_{kk}-m_{ik})$. Les autres coefficients s'obtenant de la même façon, nous passons les détails. □

Proposition 5.17 (Existence de candidats)

$$\Lambda_i \textit{ est régulière} \iff \Gamma_i \textit{ est régulière} \iff m'_{kk} \neq 1 \iff m_{ik} \neq 0.$$

Preuve. Nous prouvons que $Ker(\Lambda_i) = \{0\} \iff m_{ik} \neq 0$.

Commençons par transformer Λ_i :

$$\Lambda_i = (I-D_k)(I-\Delta) + D_i(I+\Delta) = (I-(I-D_k-D_i)\Delta) + D_i - D_k.$$

Définissons alors (comme au corollaire 5.3) $S := I - 2D_k$. On a

$$\begin{aligned}
S\Lambda_i &= (S - S(I-D_k-D_i)\Delta) + S(D_i-D_k) \\
&= (S-(I-D_k+D_i)\Delta) + (D_k-D_i) \\
&\qquad (\text{car } S(I-D_k-D_i) = I-D_k+D_i \text{ et } S(D_i-D_k) = D_k-D_i) \\
&= (I-2D_k-(I-D_k+D_i)\Delta) + D_k - D_i \qquad (\text{par définition de } S) \\
&= (I-(I-D_k+D_i)\Delta) - D_k - D_i \\
&= (I-\Delta') - D_k - D_i.
\end{aligned}$$

$$\begin{aligned}
\Lambda_i x = 0 &\iff S\Lambda_i x = 0 \qquad (\text{car } S \text{ est une matrice diagonale régulière}) \\
&\iff (I-\Delta')x - (D_k+D_i)x = 0 \\
&\iff x = M'(D_k+D_i)x \\
&\iff \forall j \in [1..n] \quad x_j = m'_{jk}(x_k+x_i).
\end{aligned}$$

La dernière relation donne $x_k = m'_{kk}(x_k+x_i)$ et $x_i = m'_{ik}(x_k+x_i)$. D'où, $x_i = (m'_{kk}-1)(x_k+x_i) = x_k - x_k - x_i = -x_i$ c.a.d. $x_i = 0$. Finalement,

$$\Lambda_i x = 0 \iff (x_i = 0 \quad \text{et} \quad \forall j \neq i,\ x_j = m'_{jk}x_k). \tag{5.42}$$

Clairement $m'_{kk} = 1$ implique que $\forall x_k \in \mathbb{R}$ le vecteur $(m'_{1k}x_k, \dots, 0, \dots, m'_{nk}x_k)$ satisfait la partie droite de (5.42), donc la partie gauche, et ainsi $dim(Ker(\Lambda_i)) > 0$.

Réciproquement, si $m'_{kk} \neq 1$, on peut voir que seul 0_n satisfait la partie droite de (5.42).

Par conséquent, $Ker(\Lambda_i) = \{0\} \Longleftrightarrow m'_{kk} \neq 1$. D'après le lemme 5.11 on a alors

$$m'_{kk} = 1 \Longleftrightarrow \frac{m_{kk}}{m_{kk} - m_{ik}} = 1 \Longleftrightarrow m_{ik} = 0.$$

En résumé :

$$\Lambda_i \text{ régulière } \Longleftrightarrow Ker(\Lambda_i) = \{0\} \Longleftrightarrow m'_{kk} \neq 1 \Longleftrightarrow m_{ik} \neq 0.$$

La preuve est similaire pour Γ_i. On a

$$\Gamma_i = (I - D_k)(I - \Delta) + D_i = I - (I - D_k)\Delta + D_i - D_k.$$

Si on multiplie cette matrice par la matrice régulière $S = I - 2D_k + D_i$ on tombe de nouveau sur

$$S\Gamma_i = (I - \Delta') - D_k - D_i,$$

en nous aidant du fait que $S(I - D_k) = I - D_k + D_i$ et $S(D_i - D_k) = D_k - D_i$. □

Proposition 5.18 *Si $m_{ik} \neq 0$, alors*

$$(i) \qquad \alpha_{ki} + m_{ki} - m_{ii}\alpha_{ki} \leq 0$$

$$(ii) \qquad \forall j \neq k,\ j \neq i, \quad m_{kj} - m_{ij}\alpha_{ki} \leq 0$$

Preuve. (i). On a

$$\alpha_{ki} + m_{ki} - m_{ii}\alpha_{ki} = \frac{m_{kk}}{m_{ik}} + m_{ki} - \frac{m_{ii}m_{kk}}{m_{ik}} = m_{ki} + m_{kk}\frac{1 - m_{ii}}{m_{ik}}.$$

Calculons $(1 - m'_{kk})(\alpha_{ki} + m_{ki} - m_{ii}\alpha_{ki})$. On trouve

$$(1 - \frac{m_{kk}}{m_{kk} - m_{ik}})(m_{ki} + m_{kk}\frac{1 - m_{ii}}{m_{ik}}),$$

puis en développant,

$$m_{ki} + m_{kk}(\frac{1 - m_{ii}}{m_{ik}} - \frac{m_{ki} + m_{kk}(1 - m_{ii})/m_{ik}}{m_{kk} - m_{ik}}), \text{ c.a.d. } m_{ki} - m_{kk}(\frac{m_{ki} + (1 - m_{ii})}{m_{kk} - m_{ik}}),$$

qui n'est autre que m'_{ki}. D'après la proposition 5.17, $m_{ik} \neq 0$ équivaut à $m'_{kk} \neq 1$; et en vertu du lemme 5.10, $m'_{kk} \geq 1$. D'où $m'_{kk} > 1$. On peut donc écrire $\alpha_{ki} + m_{ki} - m_{ii}\alpha_{ki} = m'_{ki}/(1 - m'_{kk})$. Il suffit alors d'utiliser $m'_{ki} \geq 0$ et $m'_{kk} > 1$ pour en conclure $\alpha_{ki} + m_{ki} - m_{ii}\alpha_{ki} \leq 0$. Si on calcule de façon similaire $(1 - m'_{kk})(m_{kj} - m_{ij}\alpha_{ki})$, on trouve m'_{kj} et on conclut de la même manière. □

5.8 Conclusion

Nous avons présenté dans ce chapitre quelques méthodes permettant d'approximer intérieurement et extérieurement un AE-solution set linéaire. Un choix de présentation légèrement différent du choix habituel a été adopté, qui a permis d'homogénéiser la théorie des AE-systèmes linéaires avec celle des systèmes linéaires classiques.

Nous avons de plus proposé deux nouvelles méthodes : la méthode généralisée LU, qui est séduisante d'un point de vue théorique (en tant qu'application remarquable des intervalles généralisés) mais pour laquelle une utilité en pratique reste à établir ; et la méthode de Hansen-Bliek généralisée qui se pose comme une alternative efficace à la recherche exhaustive pour l'obtention d'une boîte extérieure. Cette nouvelle méthode est par contre limitée aux AE-systèmes quantifiés à droite. Il est légitime de trouver que cette classe est restrictive. Cependant, trois arguments peuvent être avancés :

- Le saut de complexité dû à la quantification à droite peut laisser imaginer que la boîte extérieure optimale d'un AE-solution set quelconque reste un problème NP-complet même après préconditionnement. La géométrie des AE-solution sets est bien plus inextricable que celle des AE-solution sets quantifiés à droite.
- La méthode de Hansen-Bliek ne peut de toute façon pas être étendue telle quelle aux AE-systèmes, tout simplement parce qu'elle repose fondamentalement sur la positivité de $\Delta = \mathrm{rad}\,(\mathbf{A})$.
- D'un point de vue pratique, les AE-systèmes quantifiés à droite jouent un rôle privilégié pour le filtrage d'AE-systèmes non linéaires carrés via l'opérateur de Newton généralisé.

Plusieurs améliorations sont envisageables :

- Une évaluation expérimentale de l'influence de la méthode de Hansen-Bliek généralisée par rapport aux autres méthodes, lorsque celle-ci est utilisée comme brique dans le filtrage de Newton généralisé.
- Le calcul des enveloppes de chaque composante connexe du AE-solution set.
- Une extension de la méthode à certaines matrices quantifiées bien particulières (par exemple, avec les quantificateurs universels disposés en colonne).

Chapitre 6

Programmation par contraintes sur les réels

Sommaire

La programmation par contraintes est un *paradigme* de programmation. La communauté travaillant dans ce domaine étudie notamment :

- la *programmation logique avec contraintes* (en anglais : CLP) qui intègre des contraintes dans le modèle classique de la programmation logique, notamment des relations arithmétiques sur les réels ou les rationnels (mal prises en compte par la logique du premier ordre) ;
- le *problème de satisfaction de contraintes* (en anglais : CSP) issu de l'intelligence artificielle et qui permet de modéliser et de résoudre de nombreux problèmes NP-complets ;
- le célèbre problème NP-complet de *satisfiabilité de formules logiques booléennes* (en anglais : SAT) issu également de l'intelligence artificielle.

Ces trois sous-communautés sont aujourd'hui assez proches les unes des autres, proposent des logiciels libres ou privés. Elles ont des liens forts avec la recherche opérationnelle pour résoudre des problèmes combinatoires difficiles de taille industrielle.

Un problème de satisfaction de contraintes se décrit dans un formalisme simple et uniforme comprenant un ensemble de variables pouvant chacune prendre un ensemble de valeurs possibles et un ensemble de contraintes (relations) entre ces variables. La programmation par contraintes propose aussi un arsenal de méthodes pour résoudre n'importe quelle instance exprimée dans ce formalisme. La plupart des méthodes se partagent en deux grandes catégories. D'abord, des algorithmes combinatoires de recherche de solutions instancient les variables

de toutes les façons possibles jusqu'à ce qu'une solution soit trouvée. Ensuite, ces algorithmes sont combinés à des algorithmes de filtrage maintenant une *cohérence locale* (propriété vérifiée par chaque contrainte individuellement) au cours de la recherche. Ces algorithmes de filtrage permettent d'éliminer en temps polynomial des parties de l'espace de recherche.

A partir de la fin des années 1980, quelques chercheurs de programmation par contraintes ont adapté certains algorithmes de filtrage aux systèmes d'équations sur les réels où les variables ont un intervalle comme domaine initial. Principalement trois opérateurs performants viennent ainsi enrichir des solveurs proposés par les communautés de programmation par contraintes ou d'analyse par intervalles : la 2B-cohérence, la Box-cohérence et la 3B-cohérence.

Ce chapitre donne un tour d'horizon de ces techniques issues de la programmation par contraintes. Nous nous efforçons dans notre présentation de séparer le plus possible les propriétés de cohérence locale, c'est à dire ce que l'on souhaite calculer, de la façon de les calculer. Les propriétés sont d'abord étudiées dans un modèle théorique qui fait abstraction des problèmes d'arrondis sur les machines. Ce modèle permet de définir simplement un CSP (sur intervalles), la projection et l'arc-cohérence. Nous exhibons des propriétés liées à la projection (monotonie et idempotence). Nous montrons l'infaisabilité de l'arc-cohérence dans le cas général à partir d'un exemple-type. Les opérateurs mentionnés ci-dessus sont alors détaillés.

Pour éviter des convergences lentes lors de ces filtrages, nous expliquons comment gérer un paramètre de précision. Nous explicitons les défauts du paramètre w communément utilisé et introduisons comme alternative un paramètre σ qui offrirait en quelque sorte un découpage prédéfini des intervalles manipulés.

Enfin, les flottants sont pris en compte pour définir une cohérence qui soit calculable avec une machine.

6.1 Introduction aux CSP

Les problèmes traités par la programmation par contraintes sont appelés CSP (*constraint satisfaction problems*). Ils sont définis ainsi :

Définition 6.1 (CSP) *Un problème de satisfaction de contraintes (***CSP***) est un triplet $(\mathcal{C}, \mathcal{X}, \mathcal{D})$, où :*
- *$\mathcal{X}$ est un ensemble $\{x_1, \ldots, x_n\}$ de symboles appelés* **variables**.
- *$\mathcal{D}$ est le produit cartésien $\mathcal{D}_{x_1} \times \ldots \times \mathcal{D}_{x_n}$ d'ensembles indicés par les variables et appelés* **domaines**. *L'ensemble $\mathcal{D}$ est appelé* **espace de recherche**.
- *$\mathcal{C}$ un ensemble $\{c_1, \ldots, c_m\}$ de relations sur $\mathcal{D}$ appelées* **contraintes**.

On appelle **solution** *du CSP tout n-uplet $(v_1, \ldots, v_n) \in \mathcal{D}$ satisfaisant[1] simultanément toutes les contraintes.*

A chaque variable x est donc associé un domaine noté $\mathcal{D}_x$ appelé *domaine de x*. On distingue habituellement trois types de CSP :
- Les CSP **booléens**, où pour tout $x \in \mathcal{X}$, $\mathcal{D}_x = \{0, 1\}$. Nous ne les utilisons pas dans cette thèse.
- Les CSP **discrets**, où pour tout $x \in \mathcal{X}$, $\mathcal{D}_x$ est un ensemble fini d'entiers. Nous les utiliserons pour introduire quelques notions et algorithmes.
- Les CSP **continus**, où pour tout $x \in \mathcal{X}$, $\mathcal{D}_x \subseteq \mathbb{R}$. Ce sont bien sûr ceux qui nous intéressent. Les contraintes sont en général des équations (ou inéquations) et les domaines des intervalles.

[1] En tant que relation sur $\mathcal{D}$, la contrainte c est une fonction $c : \mathcal{D} \to \{\texttt{vrai}, \texttt{faux}\}$ qu'on associe canoniquement au sous-ensemble de $\mathcal{D}$ des éléments dont l'image par c est `vrai`. On dit que "$x \in \mathcal{D}$ satisfait c" ou tout simplement $c(x)$ pour dire que $c(x) = \texttt{vrai}$. Prenons par exemple $\mathcal{D} = \{1, 2, 3\} \times \{1, 2, 3\}$. La contrainte c peut alors être soit définie en *extension*, c'est à dire en listant l'ensemble des éléments vérifiant c (par exemple : $c(x, y) \iff (x, y) \in \{(1, 2), (1, 3), (2, 3)\}$), soit en *intention*, c'est à dire en donnant une propriété vérifiée par (et uniquement par) ces éléments (c.a.d., $c(x, y) \iff x < y$). Les contraintes seront toujours données par la suite en intention, donc à l'aide d'une expression formelle faisant intervenir les variables de $\mathcal{X}$.

Voici un exemple de CSP discret :

Exemple 6.1 (CSP) *Soit $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ le CSP suivant :*
$\mathcal{X} = \{x, y, z\}$, $\mathcal{D} = \{\mathcal{D}_x \times \mathcal{D}_y \times \mathcal{D}_z\}$, $\mathcal{C} = \{c_1, c_2\}$ *avec* $\mathcal{D}_x = \{1, 2, 3\}, \mathcal{D}_y = \{2, 3, 4\}, \mathcal{D}_z = \{1, 2, 3\}$, *et*

(c_1) $x \geq y$,
(c_2) $y = z + x$,
(c_3) *les valeurs prises par les variables doivent être toutes différentes.*

On note $Var(c)$ les variables impliquées dans la contrainte c. Exemple : $Var(c_1) = \{x, y\}$.

Nous avons introduit au §2.5 p.19 l'algorithme de bissection & évaluation pour un système d'équations, ainsi que l'intérêt de filtrer le domaine des variables. Des algorithmes de recherche arborescente similaires existent pour les CSP discrets, où la bissection est souvent remplacée par une instanciation de valeur[2]. L'évaluation peut être assimilée à une simple vérification des contraintes dont les variables ont été instanciées (*backtrack chronologique*), La nécessité de filtrer les domaines se pose de la même façon. Nous définissons un filtrage ainsi[3] :

Définition 6.2 (Filtrage) *Soient $\mathcal{C}$ un ensemble de contraintes et $\mathcal{X}$ un ensemble de variables. On appelle* **filtrage** *toute fonction Φ qui à un espace de recherche $\mathcal{D}$ associe un espace de recherche $\Phi(\mathcal{D})$ tel que*

- *Si $(v_1, \ldots, v_n)$ est une solution de $(\mathcal{C}, \mathcal{X}, \mathcal{D})$, alors $(v_1, \ldots, v_n) \in \Phi(\mathcal{D})$* *(conservatisme),*
- *$\Phi(\mathcal{D}) \subseteq \mathcal{D}$, c.a.d., $\big(\forall x \in \mathcal{X}\big)\ (\Phi(\mathcal{D}))_x \subseteq \mathcal{D}_x$* *(contractance),*
- *Si $\mathcal{D} \subseteq \mathcal{D}'$ alors $\Phi(\mathcal{D}) \subseteq \Phi(\mathcal{D}')$* *(monotonie).*

L'algorithme de filtrage souvent cité comme fondateur en programmation par contraintes consiste, dans un CSP discret, à éliminer des domaines les valeurs ne pouvant satisfaire une contrainte particulière (on parle de cohérence *locale*), donc ne pouvant satisfaire le système entier, et de propager ces réductions d'une contrainte à l'autre [Wal75]. Cet algorithme de propagation sera plus tard amélioré dans le formalisme des contraintes et rebaptisé `AC3` [Mac77]. Il existe aujourd'hui sous de multiples variantes [Rég05] et semble encore alimenter de nombreux développements.

Nous allons introduire l'algorithme `AC3` pour deux raisons : d'une part, les concepts derrière cet algorithme sont clés pour la programmation par contraintes en domaine continu ; d'autre part, un travail important de cette thèse a consisté à essayer d'étendre cet algorithme aux CSP continus.

L'algorithme `AC3` repose sur deux notions fondamentales : la *projection* et la *propagation*. Dans le §6.1.1, nous présentons la projection puis donnons la boucle principale de l'algorithme `AC3`. Nous détaillons ensuite dans le §6.1.2 la partie propagation de l'algorithme.

6.1.1 Projection et Arc-cohérence

Étant donné une contrainte c et une variable x, l'opération de base de l'algorithme `AC3`, appelé *projection*, consiste à éliminer du domaine $\mathcal{D}_x$ les valeurs "localement" incohérentes vis-à-vis de c. Sa définition repose sur celle de *support*.

Définition 6.3 (Support d'une valeur pour une contrainte) *Soit $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP. Soient $c \in \mathcal{C}$, $x \in Var(c)$, $v \in \mathcal{D}_x$ et notons $\{x, y_1, \ldots, y_k\}$ l'ensemble $Var(c)$.*

[2] Un choix non-déterministe est fait à chaque étape pour fixer une variable à une valeur dans son domaine.

[3] Il ne faut pas voir la troisième condition sur le même plan que les deux premières : la monotonie et la contractance suffisent à définir un filtrage, mais la monotonie est utile pour justifier (par exemple) les algorithmes de réfutation (cf. §6.6). L'opérateur de Newton (cf. p.33) est bien conservatif et contractant, mais non monotone. Il échapperait donc à la définition de filtrage : pour cette raison, la monotonie doit être vue plutôt comme une propriété annexe.

On appelle **support** *de $x = v$ pour c, tout k-uplet $(v_1, \ldots, v_k)$ de $\mathcal{D}_{y_1} \times \ldots \times \mathcal{D}_{y_k}$ tel que la contrainte c soit satisfaite avec $(x = v), (y_1 = v_1) \ldots, (y_k = v_k)$.*

Les supports de l'exemple 6.1 sont donnés dans la table suivante :

contrainte c_1

valeur	supports
$x = 1$	$y \in \emptyset$
$x = 2$	$y \in \{2\}$
$x = 3$	$y \in \{2, 3\}$
$y = 2$	$x \in \{2, 3\}$
$y = 3$	$x \in \{3\}$
$y = 4$	$x \in \emptyset$

contrainte c_2

valeur	supports
$x = 1$	$(y, z) \in \{(2, 1), (3, 2), (4, 3)\}$
$x = 2$	$(y, z) \in \{(3, 1), (4, 2)\}$
$x = 3$	$(y, z) \in \{(4, 1)\}$
$y = 2$	$(x, z) \in \{(1, 1)\}$
$y = 3$	$(x, z) \in \{(2, 1), (1, 2)\}$
$y = 4$	$(x, z) \in \{(1, 3), (2, 2), (3, 1)\}$
$z = 1$	$(x, y) \in \{(1, 2), (2, 3), (3, 4)\}$
$z = 2$	$(x, y) \in \{(1, 3), (2, 4)\}$
$z = 3$	$(x, y) \in \{(1, 4)\}$

contrainte c_3

valeur	supports
$x = 1$	$(y, z) \in \{(2, 3), (3, 2)\}$
$x = 2$	$(y, z) \in \{(1, 3), (3, 1)\}$
$x = 3$	$(y, z) \in \{(1, 2), (2, 1)\}$
...	...

Définition 6.4 (Projection d'une contrainte sur une variable) *Soient $\mathcal{C}$ un ensemble de contraintes, $\mathcal{X}$ un ensemble de variables, $c \in \mathcal{C}$ et $x \in \mathcal{X}$.*

On appelle **projection** *de c sur x, et on note Π_x^c la fonction qui à un espace de recherche $\mathcal{D}$ associe l'espace de recherche $\mathcal{D}'$ vérifiant*

$$\forall y \in \mathcal{X} \backslash \{x\} \quad D'_y = \mathcal{D}_y,$$
$$D'_x := \{v \in \mathcal{D}_x \mid \text{ il existe dans le CSP } (\mathcal{C}, \mathcal{X}, \mathcal{D}) \text{ un support de } x = v \text{ pour } c\}.$$

Remarquons que, dans notre définition, il n'est pas imposé que c et x soit liés (s'il ne le sont pas, nous avons $\Pi_x^c(\mathcal{D}) = \mathcal{D}$). Le domaine D'_x obtenu en projetant une contrainte c sur x permet d'obtenir une réduction du domaine de x en remplaçant dans le CSP l'ancien domaine $\mathcal{D}_x$ par le nouveau D'_x (cette réduction est conservative car les valeurs sans support ne peuvent former une solution).

Exemple 6.2 (Projection dans le cas discret) *Considérons un CSP discret, et une contrainte $c(x, y)$ binaire. Une implémentation possible de la projection de c sur x consiste à tester pour chaque valeur de $\mathcal{D}_x$ s'il existe une valeur compatible dans le domaine de $\mathcal{D}_y$, c'est à dire un couple (v_x, v_y) satisfaisant c :*

$D'_x \leftarrow \emptyset$
`pour tout` $v_x \in \mathcal{D}_x$
 `pour tout` $v_y \in \mathcal{D}_y$
  `si` $c(v_x, v_y)$ `alors` $D'_x \leftarrow D'_x \cup v_x$
`retourner` D'_x

L'algorithme `AC3` effectue des projections avec des couples (contrainte, variable) jusqu'à ce qu'il n'y ait plus de réduction possible, c.a.d. l'obtention d'un point fixe. Lorsque celui-ci est atteint, la propriété obtenue est appelée *arc-cohérence* :

Définition 6.5 (Arc-cohérence) *Soit $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP. On dit que l'espace de recherche $\mathcal{D}$ est* **arc-cohérent** *si*

$$\forall (c, x) \in \mathcal{C} \times \mathcal{X} \quad \Pi_x^c(\mathcal{D}) = \mathcal{D}.$$

Remarque 6.1 *Pour être rigoureux, il conviendrait de dire qu'un CSP est arc-cohérent (au lieu de « un espace de recherche est arc-cohérent »). Nous adoptons néanmoins cet abus de langage, qui permet d'éviter des lourdeurs.*

Une manière naïve d'obtenir ce point fixe (`AC1`) consiste à parcourir en boucle la liste des couples (contrainte, variable) jusqu'à ce qu'un parcours entier de la liste ne produise plus aucun changement. L'algorithme `AC3` implémente un moyen d'obtenir ce point fixe sans faire de parcours systématique de la liste. Il gère un *agenda*. Cet agenda contient les couples pouvant potentiellement entraîner une réduction de domaine, donc à soumettre à l'opérateur de projection. Le gain de temps se fait en ajoutant dans l'agenda uniquement les couples (c, x) impactés par une réduction. Lorsque l'agenda est vide, on prouve que le point fixe est atteint. Cette technique (`AC3`) porte le nom de *propagation de contraintes*. Avant de la détailler, en voici la boucle principale :

`ALGORITHME AC3`(paramètre : CSP $(\mathcal{C}, \mathcal{X}, \mathcal{D})$)

mettre dans un agenda tous les couples $(c, x) \in \mathcal{C} \times \mathcal{X}$ tels que $x \in Var(c)$.
`tant-que` l'agenda n'est pas vide :
 enlever un couple (c, x) de l'agenda
 $D'_x \leftarrow \Pi^c_x(\mathcal{D})$
 `si` $D'_x \neq \mathcal{D}_x$ `alors`
  `propager`(c, x)
  $\mathcal{D}_x \leftarrow D'_x$
 `fin si`
`fin tant-que`

Définition 6.6 (Clôture par arc-cohérence) *On suppose fixés par le contexte un ensemble $\mathcal{C}$ de contraintes et un ensemble $\mathcal{X}$ de variables. On appelle clôture par arc-cohérence d'un espace de recherche $\mathcal{D}$ l'unique espace de recherche maximal arc-cohérent inclus dans $\mathcal{D}$.*

`AC3` calcule donc la clôture par arc-cohérence d'un CSP. Ce résultat sera prouvé plus loin (cf. proposition 6.2 p.150). La définition de clôture permet de désigner l'objet calculé par `AC3` indépendamment de `AC3`, elle sera utilisée en domaine continu bien qu'`AC3` n'existe qu'en domaine discret.

6.1.2 Propagation

Si la projection de c sur x réduit son domaine, quels sont les couples pouvant être impactés ? Tout d'abord, il est clair que ces couples sont au moins inclus dans l'ensemble suivant :

$$\{(c', x') \in \mathcal{C} \times \mathcal{X},\ x \in Var(c') \text{ et } x' \in Var(c')\}. \tag{6.1}$$

Nous allons montrer qu'il est possible de retirer de cet ensemble tous les couples impliquant la variable x ou la contrainte c. L'algorithme de propagation s'écrit ainsi :

`ALGORITHME PROPAGER`(contrainte c, variable x)

`pour tout` $c' \in \mathcal{C}\backslash\{c\}$ tel que $x \in Var(c')$
 `pour tout` $x' \in \mathcal{X}\backslash\{x\}$ tel que $x' \in Var(c')$
  ajouter le couple (c', x') dans l'agenda
 `fin pour`
`fin pour`

Pour prouver que cet algorithme est correct, nous supposons qu'avant d'en extraire le couple (c, x) l'agenda était *complet*, c.a.d., contenait au moins tous les couples pouvant entraîner une réduction des domaines. Notons de plus $\mathcal{D}_x$ l'espace de recherche avant la projection de c sur x et $\mathcal{D}'_x$ le nouvel espace. Le retrait de l'ensemble (6.1) des couples impliquant x repose sur le lemme suivant :

Lemme 6.1 (Monotonie de la projection)
Soit $\mathcal{D}$ et $\mathcal{D}'$ deux espaces de recherche tels que $\mathcal{D}' \subseteq \mathcal{D}$ (cf. def. 6.2). Alors, pour tout couple $(c, x) \in \mathcal{C} \times \mathcal{X}$:

$$\begin{array}{ll} (i) & \Pi^c_x(\mathcal{D}') \subseteq \Pi^c_x(\mathcal{D}), \\ (ii) & \big(\forall y \neq x\big)\ \mathcal{D}'_y = D_y \quad \Longrightarrow \quad \Pi^c_x(\mathcal{D}') = \Pi^c_x(\mathcal{D}) \cap D'_x. \end{array}$$

Montrons qu'il est inutile d'ajouter dans l'agenda tout couple (c', x), avec $c' \neq c$. En effet, si le couple (c', x) est déjà dans l'agenda avant qu'on en extraie (c, x), il est inutile de le remettre. Supposons qu'il n'y soit pas. L'agenda étant complet par hypothèse avant d'en extraire (c, x), on a $\mathcal{D}_x = \Pi^{c'}_x(\mathcal{D})$. Or, la projection de c sur x ne modifie que le domaine de x, donc en appliquant le lemme 6.1 (ii), on a

$$\begin{array}{lll} \Pi^{c'}_x(\mathcal{D}') & = \Pi^{c'}_x(\mathcal{D}) \cap \mathcal{D}'_x & \text{(lemme 6.1 (ii)),} \\ & = \mathcal{D}_x \cap \mathcal{D}'_x & \text{(car l'agenda était complet),} \\ & = \mathcal{D}'_x & \text{(car } \mathcal{D}'_x \subseteq \mathcal{D}_x\text{).} \end{array}$$

Aucune réduction n'est possible en projetant c' sur x, il est donc inutile d'ajouter (c', x). Le retrait de l'ensemble (6.1) des couples impliquant c repose sur le lemme suivant[4] :

Lemme 6.2 (Idempotence de la projection)
Soit $\mathcal{D}$ un espace de recherche. Alors, pour tout couple $(c, x) \in \mathcal{C} \times \mathcal{X}$, si on pose

$$\mathcal{D}' := \Pi^c_x(\mathcal{D}) \quad \textit{et} \quad \mathcal{D}'' := \Pi^c_x \circ \Pi^c_{x_1} \circ \ldots \circ \Pi^c_{x_n}(\mathcal{D}),$$

On a

$$D'_x = D''_x.$$

Le lemme signifie que projeter une contrainte c sur une variable x *avant* ou *après* une projection de c sur toutes les autres variables, ne change rien pour x. Par monotonie (lemme 6.1 (i)), il s'ensuit que les projections successives d'une même contrainte peuvent être ordonnées d'une façon arbitraire et qu'il est inutile de projeter deux fois sur la même variable.

Venons-en au fait. Il est inutile d'ajouter dans l'agenda tout couple (c, x'), avec $x \neq x'$. En effet, si le couple (c, x') est déjà dans l'agenda avant que (c, x) soit retiré, il est inutile de le remettre. Supposons donc que le couple (c, x') n'y soit pas et remarquons que chaque fois que la contrainte c est impactée, tous les couples (c, x') sont ajoutés dans l'agenda simultanément, à l'exception du couple dont la variable est à l'origine de l'impact (si on tient compte de la première optimisation). Si le couple (c, x') n'est pas présent dans l'agenda avant qu'on traite (c, x), c'est donc soit qu'il a déjà été traité depuis la *dernière* réduction qui avait impacté c, soit qu'il est lui-même à l'origine de cet impact auquel cas on peut de nouveau considérer qu'il a été traité[5]. Dans les deux cas, par idempotence (lemme 6.2), il est inutile de le retraiter.

6.2 Généralités sur les CSP continus

Par souci de clarté, nous supposons que les contraintes sont données sous forme d'équations $f_i(x) = 0$ $(i \in [1..m])$. Autoriser des inéquations $f_i(x) \leq 0$ ou un mélange des deux ne pose pas de difficulté théorique mais nécessite d'adapter les résultats Chaque fonction f_i doit être une expression *arithmétique* (cf. §2.3 p.14), c'est à dire obtenue par composition de fonctions de base, ou *primitives* (voir table 6.1).

[4] Nous employons le terme *idempotence* car ce lemme implique notamment $\Pi^c_x(\mathcal{D}) = \Pi^c_x(\Pi^c_x(\mathcal{D}))$.

[5] En effet, si la dernière projection qui impacte la contrainte c est sur la variable x', en n'ajoutant pas le couple (c, x') dans l'agenda, on "anticipe" justement sur le fait que son traitement ne provoquera pas de réduction.

$y = x$	$y = \exp(x)$	$y = \cos(x)$	$y = \arccos(x)$
$y = x + z$	$y = \ln(x)$	$y = \sin(x)$	$y = \arcsin(x)$
$y = x - z$	$y = \log(x)$	$y = \tan(x)$	$y = \arctan(x)$
$y = x \times z$	$y = \text{sqr}(x)$	$y = \cosh(x)$	$y = \text{arccosh}\ (x)$
$y = x/z$	$y = \sqrt{x}$	$y = \sinh(x)$	$y = \text{arcsinh}\ (x)$
$y = x^a$	$y = 1/x$	$y = \tanh(x)$	$y = \text{arctanh}\ (x)$

TAB. *6.1: Contraintes primitives*

Dans un CSP continu, en général, $\mathcal{D}_x$ est un intervalle de $\mathbb{IR}$, auquel cas l'espace de recherche $\mathcal{D}$ est simplement une boîte. Le modèle de CSP coïncide alors avec les systèmes d'équations par intervalles décrits dans la partie 1 de cette thèse. C'est pourquoi les algorithmes décrits ici peuvent être combinés avec ceux d'analyse par intervalles. Nous aurons également recours à une représentation des domaines plus fine appelée *gruyère*, basée sur les unions d'intervalles :

Définition 6.7 (Union) *Soit* $\lceil \mathbf{x} \rceil$ *une union finie d'intervalles fermés de réels. On appelle* **représentation** *de* $\lceil \mathbf{x} \rceil$ *l'unique ensemble d'intervalles disjoints* $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ *tels que*

$$\lceil \mathbf{x} \rceil = \mathbf{x}_1 \cup \ldots \cup \mathbf{x}_k.$$

Une union d'intervalles fermés $\lceil \mathbf{x} \rceil$ *est assimilée à sa représentation, et sera dorénavant considérée comme disjointe. On appelle alors* **union** *toute union disjointe d'intervalles fermés.*

Soit $\mathbf{y} \in \mathbb{IR}$. *On note* $\mathbf{y} \in \lceil \mathbf{x} \rceil$ *lorsque* $\mathbf{y}$ *est un intervalle de la représentation de* $\lceil \mathbf{x} \rceil$. *On appelle* **taille** *de* $\lceil \mathbf{x} \rceil$ *et on note* $|\lceil \mathbf{x} \rceil|$ *le nombre d'intervalles de la représentation de* $\lceil \mathbf{x} \rceil$.

Exemple 6.3 *Soit* $\lceil \mathbf{x} \rceil = [-1, 0] \cup [1, 4] \cup [2, 5]$.
- *La représentation de* $\lceil \mathbf{x} \rceil$ *est* $[-1, 0] \cup [1, 5]$.
- *La taille de* $\lceil \mathbf{x} \rceil$ *est* $|\lceil \mathbf{x} \rceil| = 2$.
- *On a* $[3, 4] \subseteq \lceil \mathbf{x} \rceil$ *mais* $[3, 4] \notin \lceil \mathbf{x} \rceil$.
- *On a* $3 \in \lceil \mathbf{x} \rceil$ *(en tant que réel) mais* $[3, 3] \notin \lceil \mathbf{x} \rceil$ *(en tant qu'élément de la représentation). Il y a donc là une ambiguïté, que nous éviterons.*
- $[1, 5] \in \lceil \mathbf{x} \rceil$ *(il n'y a pas d'ambiguïté).*

Définition 6.8 (Gruyère) *Un* **gruyère** *est le produit cartésien d'unions (voir figure 6.1).*

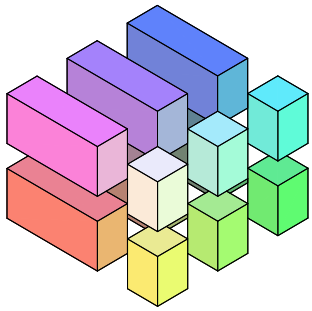


FIG. *6.1: Exemple de gruyère en dimension 3.*

Ce chapitre présente quelques *cohérences partielles* qui permettent de définir des filtrages, tels que nous les avons introduits au chapitre 2. Ces filtrages sont destinés à être intégrés à l'algorithme de bissection et évaluation. Puisque nous ne manipulons plus une seule fonction $f : \mathbb{R}^n \to \mathbb{R}^m$ mais m contraintes de type $f_i(x) = 0$ avec $f_i : \mathbb{R}^n \to \mathbb{R}$, nous devons adapter notre définition d'*évaluation* :

Définition 6.9 (Évaluation d'une contrainte)
Soit $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP. Pour toute contrainte $c : f(x_1, \ldots, x_n) = 0$ de $\mathcal{C}$, on appelle évaluation de c le calcul de

$$range(f, \prod_{x \in Var(c)} \mathcal{D}_x).$$

Si $0 \notin range(f, \mathcal{D})$, on en déduit que la contrainte est insatisfiable sur $\mathcal{D}$, ce qui montre que l'espace de recherche courant est infaisable. On retrouve bien la notion d'évaluation présentée au §2.5 p.19.

6.3 Arc-cohérence

6.3.1 Le modèle théorique

Le modèle théorique consiste à calculer la clôture par arc-cohérence d'un CSP, en travaillant avec des ensembles de réels. Il n'y a aucune contre-indication à cela puisque la clôture par arc-cohérence ne fait intervenir que des définitions mathématiques.
Notons Π l'opérateur $(\Pi_{x_n}^{c_m} \circ \ldots \circ \Pi_{x_1}^{c_m}) \circ \ldots \circ (\Pi_{x_n}^{c_1} \circ \ldots \circ \Pi_{x_1}^{c_1})$, c'est à dire la fonction qui à partir d'un espace de recherche $\mathcal{D}$ compose toutes les projections possibles (la propagation permet en pratique d'éviter les calculs inutiles). Alors nous pouvons définir la suite d'espaces de recherches :

$$\begin{array}{l} \mathcal{D}^{(0)} := \mathcal{D}, \\ \mathcal{D}^{(k)} := \Pi(\mathcal{D}^{(k-1)}). \end{array} \tag{6.2}$$

Cette suite est décroissante pour l'inclusion, et minorée par l'ensemble vide. Nous admettons le résultat suivant (voir la seconde édition de [JKDW01] p.92 pour une discussion à ce propos) :

Proposition 6.1 *Si le graphe de chaque contrainte est fermé*[6] *(ce que l'on supposera dorénavant), la suite $\mathcal{D}^{(k)}$ converge*[7] *(pour une certaine distance) vers une limite $\mathcal{D}^{(\infty)}$ qui vérifie $\mathcal{D}^{(\infty)} = \Pi(\mathcal{D}^{(\infty)})$.*

On ne sait pas en général quelle est la nature de l'espace $\mathcal{D}^{(\infty)}$; notamment si dans le cas où $\mathcal{D}^{(0)}$ est une boîte, il peut être représenté par un ensemble dénombrable d'intervalles. Quoi qu'il en soit, cette limite est arc-cohérente, et vérifie même la proposition suivante :

Proposition 6.2 *La limite $\mathcal{D}^{(\infty)}$ de la suite définie par l'itération* (6.2) *est la clôture par arc-cohérence du CSP $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ au sens de la définition 6.6, c'est à dire l'unique sous-espace maximal arc-cohérent de $\mathcal{P}$.*

[6] Une inégalité stricte est un exemple de contrainte dont le graphe est un ensemble ouvert.

[7] Pour prouver ce résultat, il faut d'abord choisir une distance sur les sous-ensembles (compacts) de $\mathbb{R}$, par exemple celle de Hausdorff. Il faut montrer alors qu'avec cette distance, la suite $\mathcal{D}^{(k)}$ converge et que la fonction Π est continue. On peut alors en déduire que la limite est également un point fixe, puisque $\mathcal{D}^{(\infty)} = \lim_{n\to\infty} \Pi(\mathcal{D}^{(k)}) = \Pi(\lim_{n\to\infty} \mathcal{D}^{(k)}) = \Pi(\mathcal{D}^{(\infty)})$.

Preuve. Tout d'abord, l'existence et l'unicité d'un sous-espace arc-cohérent maximal vient du fait que $\emptyset$ est arc-cohérent, et que l'union de deux sous-espaces arc-cohérents est arc-cohérente. Par l'absurde, supposons que la proposition soit fausse et donc que $\mathcal{D}^{(\infty)}$ ne soit pas maximal. Il existe alors un espace $\mathcal{D}'$ arc-cohérent tel que

$$\mathcal{D}^{(\infty)} \subsetneq \mathcal{D}' \subseteq \mathcal{D}.$$

On en déduit qu'il existe $k \geq 0$ tel que

$$\mathcal{D}^{k+1} \subsetneq \mathcal{D}' \subseteq \mathcal{D}^{(k)}.$$

On en déduit de nouveau qu'il existe un couple (c_i, x_j) tel que

$$\Pi_{x_j}^{c_i}(\mathcal{D}^{(k*)}) \subsetneq \mathcal{D}' \subseteq \mathcal{D}^{(k*)} \text{ avec } \mathcal{D}^{(k*)} = \Big((\Pi_{x_{j-1}}^{c_i} \circ \ldots \circ \Pi_{x_1}^{c_i}) \circ \ldots \circ (\Pi_{x_n}^{c_1} \circ \ldots \circ \Pi_{x_1}^{c_1})\Big)(\mathcal{D}^{(k)}).$$

En projetant c_i sur x_j, le domaine $\mathcal{D}_{x_j}^{(k*)}$ est réduit à un domaine strictement plus petit que $\mathcal{D}'_{x_j}$, puisqu'il s'agit du seul domaine impacté. On en conclut qu'il existe une valeur x_j dans $\mathcal{D}'_{x_j}$ n'ayant pas de support dans les domaines $\mathcal{D}_x^{(k*)}$ des autres variables $x \in \mathcal{X}\backslash\{x_j\}$. Or cette valeur possède un support dans les domaines $\mathcal{D}'_x$ puisque $\mathcal{D}'$ est arc-cohérente. Comme $\mathcal{D}' \subseteq \mathcal{D}^{(k*)}$, il y a bien une contradiction. □

Définition 6.10 (Filtrage par arc-cohérence)
On appelle filtrage par arc-cohérence la fonction $\Phi_{\texttt{AC}}$ qui pour tout CSP retourne la clôture par arc-cohérence de ce CSP. La fonction $\Phi_{\texttt{AC}}$ vérifie les conditions de la définition 6.2.

6.3.2 Infaisabilité de l'arc-cohérence

Le modèle théorique nous permet d'obtenir des propriétés sur l'arc-cohérence des CSP continus qui ont des répercussions en pratique. Tout d'abord, en admettant pouvoir travailler sur les réels, il y a bien entendu des cas où le calcul de la suite $\mathcal{D}^{(k)}$ ne termine pas, c.a.d., où $\forall k \geq 0$, $\mathcal{D}^{(k)} \neq \mathcal{D}^{(\infty)}$:

Exemple 6.4 *Soit $(\{c_1, c_2\}, \{x, y\}, \{\mathcal{D}_x, \mathcal{D}_y\})$ un CSP, avec $\mathcal{D}_x = \mathcal{D}_y = [0, 1]$ et*

$$\begin{array}{ll} (c_1) & y = 2x. \\ (c_2) & y = x, \end{array}$$

Calculons $\mathcal{D}^{(1)}$. Voici les domaines obtenus en appliquant successivement les projections :

projection	domaine courant de x	domaine courant de y
$\Pi_x^{c_1}$	$[0, 0.5]$	$[0, 1]$
$\Pi_y^{c_1}$	$[0, 0.5]$	$[0, 1]$
$\Pi_x^{c_2}$	$[0, 0.5]$	$[0, 1]$
$\Pi_y^{c_2}$	$[0, 0.5]$	$[0, 0.5]$

On en déduit les premiers termes de la suite $\mathcal{D}^{(k)}$:

$\mathcal{D}^{(1)}$	$[0, 1] \times [0, 1]$
$\mathcal{D}^{(2)}$	$[0, 0.5] \times [0, 0.5]$
$\mathcal{D}^{(3)}$	$[0, 0.25] \times [0, 0.25]$
$\mathcal{D}^{(4)}$	$[0, 0.0625] \times [0, 0.0625]$
$\ldots$	$\ldots$

Il est clair que $\mathcal{D}^{(\infty)} = [0, 0] \times [0, 0]$ mais que cette limite n'est pas atteinte avec un nombre fini d'itérations. L'algorithme `AC3` ne termine pas. D'un point de vue pratique, ce phénomène est qualifié de *convergence lente*.

Toutefois, la clôture par arc-cohérence dans ce cas a une structure simple : une boîte dégénérée. Cela signifie qu'il existe un algorithme (qui termine) calculant la clôture par arc-cohérence de ce CSP (l'algorithme qui retourne la boîte $[0,0]\times[0,0]$!) ; il n'est donc pas exclu de mettre au point un algorithme calculant, par exemple, la clôture arc-cohérente d'un CSP dont l'espace de recherche initial est une boîte. Il est juste montré pour le moment que cet algorithme n'est pas `AC3`.

Nous allons en fait montrer qu'un tel algorithme n'existe pas, grâce à l'exemple suivant, appelé *exemple-type*.

Exemple 6.5 (Exemple-type) *Soit* $\mathcal{P} = (\{c_1, c_2\}, \{x, y\}, \mathcal{D}_x \times \mathcal{D}_y)$ *le CSP suivant :*
$\mathcal{D}_x = [1, 9]$, $\mathcal{D}_y = [1, 9]$, *et*

$$\begin{array}{ll} (c_1) & y = x, \\ (c_2) & (\frac{3}{4}(x-5))^2 = y. \end{array}$$

Proposition 6.3 *Dans la clôture par arc-cohérence de* $\mathcal{P}$*, les domaines des variables* x *et* y *sont une infinité d'intervalles non-vides disjoints.*

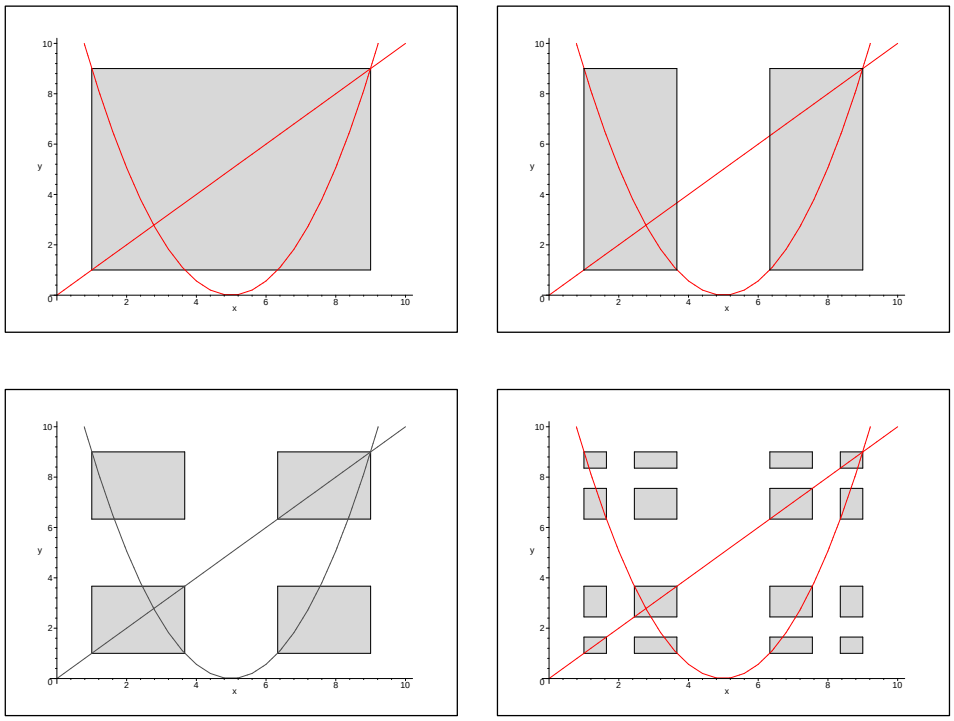


FIG. *6.2: Premières étapes de projection*

Pour prouver cette proposition, nous énonçons quelques lemmes.

Soient f_1 et f_2 les deux fonctions suivantes :

$$f_1 : y \longrightarrow \frac{4}{3}\sqrt{y} + 5, \quad f_2 : y \longrightarrow 5 - \frac{4}{3}\sqrt{y}.$$

Construisons les suites suivantes :

$$\begin{array}{ll} X_0 = \mathcal{D}_x & Y_0 = \mathcal{D}_y, \\ X_{k+1} = \Phi(Y_k) & Y_{k+1} = X_k, \end{array}$$

où on note

$$\Phi(Y_k) := range(f_1, Y_k) \cup range(f_2, Y_k).$$

Il s'avère que pour un nombre k fini d'itérations, l'espace de recherche $X_k \times Y_k$ est un gruyère.

Remarquons que, à la différence de l'opérateur Π, nous omettons volontairement les intersections avec les domaines. Nous prouverons cependant qu'on a bien $X_n \times Y_n = \Pi(X_{n-1} \times Y_{n-1})$. Ce sont donc bien, en réalité, des projections que l'on calcule. La figure 6.2 décrit les premières itérations. Les gruyères représentés sont successivement $X_0 \times Y_0$, $X_1 \times Y_0$, $X_1 \times Y_1$ et $X_2 \times Y_2$.

Il apparaît visuellement que la taille des unions croît exponentiellement. Montrons-le en établissant trois propriétés :

Lemme 6.3 $\forall n \geq 1,\ X_n \subseteq X_{n-1}$ *et* $Y_n \subseteq Y_{n-1}$.

Preuve. Par induction. On peut vérifier que $X_1 \subseteq X_0$ et $Y_1 \subseteq Y_0$. Supposons $X_n \subseteq X_{n-1}$. On a $X_n \subseteq X_{n-1} \Longrightarrow Y_{n+1} \subseteq Y_n \Longrightarrow \Phi(Y_{n+1}) \subseteq \Phi(Y_n)$ et donc $X_{n+1} \subseteq X_n$. □

Lemme 6.4 *Le nombre d'intervalles double à chaque itération ($\forall n \geq 0\ |X_n| = 2 \times |X_{n-1}|$), et plus précisément, chaque intervalle donne lieu à deux sous-intervalles disjoints.*

Preuve. Supposons que X_n et Y_n contiennent 2^n intervalles disjoints. Tout d'abord, les bornes de ces intervalles sont comprises entre 1 et 9. En effet, comme $X_0 \times Y_0 \subseteq [1,9] \times [1,9]$, le lemme 6.3 implique par une induction immédiate $X_n \times Y_n \subseteq [1,9] \times [1,9]$.

D'une part, les fonctions f_1 étant f_2 continues et monotones sur $[1,9]$, $range(f_1, X_n)$ et $range(f_2, Y_n)$ contiennent chacun 2^n intervalles disjoints. D'autre part, $range(f_1, [1,9]) \cap range(f_2, [1,9]) = \emptyset$, car $\frac{4}{3}\sqrt{[1,9]} + 5 = [\frac{19}{3}, 9]$ et $5 - \frac{4}{3}\sqrt{[1,9]} = [1, \frac{11}{3}]$, donc les 2×2^n intervalles obtenus sont tous disjoints et ainsi $|Y_{n+1}| = |X_{n+1}| = |\Phi(Y_n)| = 2^{n+1}$.
□

Lemme 6.5 *A chaque itération, les bornes des intervalles sont maintenues dans les domaines. C'est à dire, si $[a,b]$ est un intervalle*[8] *de X_n, alors $\forall m \geq n$, a et b sont les bornes d'intervalles de X_m.*

Preuve. Tout d'abord, puisque f_1 et f_2 sont monotones, pour tout intervalle I, les bornes de $range(f_1, I)$ et $range(f_2, I)$ prennent support sur les bornes de I. Nous montrons maintenant le lemme par induction. Les bornes 1 et 9 de x et y sont toujours maintenues dans les domaines car :
- (x=9, y=9) est une solution du CSP.
- La valeur $x = 1$ ne peut être supprimée en calculant $X_n \leftarrow \Phi(Y_n)$ puisque $y = 9$ est un support.
- La valeur $y = 1$ ne peut être supprimée en affectant $Y_n \leftarrow X_n$ puisque $x = 1$ est un support.

Si nous supposons que les bornes de tous les intervalles de l'union X_n sont maintenues pour tout $m \geq n$, alors les bornes de X_{n+1} le seront également pour tout $m \geq n+1$ puisqu'elles prennent support sur les bornes de Y_n, c.a.d., X_n, et qu'elles sont incluses dans X_n (lemme 6.3). □

Preuve de la proposition 6.3. Le lemme 6.3 nous permet de dire que

$$X_{k+1} = \Phi(Y_k) \cap (X_k) \quad \text{et} \quad Y_{k+1} = X_k \cap Y_k,$$

[8] Rappelons qu'en disant "$[a,b]$ est un intervalle de X_n" on signifie que $[a,b]$ est un élément de l'union, vue comme ensemble d'intervalles. Donc nous considérons $[a,b] \in X_n$, et pas seulement $[a,b] \subseteq X_n$ (voir déf. 6.7).

c'est à dire qu'ajouter une intersection avec le domaine courant à chaque itération est sans effet. Par conséquent, notre suite calcule bien successivement les projections sur x et y :

$$X_n \times Y_n = \Pi(X_{n-1} \times Y_{n-1}).$$

Il découle alors du lemme 6.4 le fait suivant : le nombre d'intervalles dans X_n tend vers l'infini ; et même si la taille des intervalles tend vers zéro, chaque intervalle apparaissant à la $n^{ème}$ itération contient un point appartenant à toutes les itérations suivantes (d'après le lemme 6.5), donc à la limite $X_\infty \times Y_\infty$. Nous avons bien montré :
Dans la clôture par arc-cohérence de $\mathcal{P}$, les domaines sont une infinité d'intervalles non-vides disjoints. □

Remarque : Dans la clôture par arc-cohérence de $\mathcal{P}$, les intervalles peuvent être dégénérés.

Le modèle théorique nous permet d'affirmer qu'aucun algorithme de filtrage calculant la clôture par arc-cohérence ne peut être établi. Ceci n'est pas très étonnant, puisqu'il s'agit d'une propriété définie sur les réels. En revanche, le modèle théorique nous renseigne sur la possibilité de calculer concrètement une *approximation* de l'arc-cohérence : la proposition 6.3 a pour conséquence pratique qu'un filtrage de type `AC3` calculant une approximation de l'arc-cohérence adaptée à la structure des flottants conduirait sous certaines hypothèses raisonnables (e.g., les arrondis des projections n'agrègent pas deux intervalles disjoints entre eux lorsque ces intervalles n'ont pas un rayon de l'ordre du flottant, etc...) à des unions dont la taille atteindrait le nombre de flottants à une vitesse exponentielle (problème de complexité en espace).

Cette affirmation a été observée en pratique sur l'exemple-type. Ainsi, l'arc-cohérence n'est pas une cohérence applicable en domaine continu. L'intuition de ce résultat existait depuis longtemps [Cle87, Hyv92, BO97], et a amené à concevoir d'autres types de cohérences locales. La plus connue d'entre elles est la 2B-cohérence. Elle sert également de base à d'autres cohérences (3B, etc...) qui ne sont plus strictement locales (c.a.d., où les supports ne sont pas calculés que pour une seule contrainte à la fois). Nous proposerons d'autres alternatives au chapitre suivant.

6.4 2B-cohérence

Informellement, la 2B cohérence[9] consiste à empêcher l'explosion du nombre d'intervalles en ne considérant non pas la projection, mais son enveloppe. De cette façon, les domaines obtenus restent des intervalles à chaque itération. La clôture par 2B-cohérence est donc une boîte, ce qui permet d'intégrer très facilement le filtrage correspondant dans des méthodes d'analyse par intervalles. Cette cohérence est décrite entre autres dans [Lho93, BO97, Col98, BGGP99, CDR99, GB01]. Notons $\Pi^\Box$ l'opérateur $(\Box\,\Pi^{c_m}_{x_n} \circ \ldots \circ \Box\,\Pi^{c_m}_{x_1}) \circ \ldots \circ (\Box\,\Pi^{c_1}_{x_n} \circ \ldots \circ \Box\,\Pi^{c_1}_{x_1})$, c'est à dire la fonction qui à partir d'un espace de recherche $\mathcal{D}$ enchaîne toutes les projections mais en ne conservant à chaque fois que l'enveloppe du résultat (l'envellope □ a été définie page 12). Nous définissons la suite d'espaces de recherches :

$$\begin{array}{l} \mathcal{D}^{(0)} := \mathcal{D}, \\ \mathcal{D}^{(k)} := \Pi^\Box(\mathcal{D}^{(k-1)}). \end{array} \tag{6.3}$$

De nouveau, cette suite possède une limite. Remarquons que la preuve est beaucoup plus facile ici (les suites $\underline{\mathcal{D}}^{(k)}$ et $\overline{\mathcal{D}}^{(k)}$ sont monotones et bornées, donc convergentes (Bolzano-Weirstrass)).

Définition 6.11 (2B-cohérence) *Soit $\mathcal{P} = (\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP. On dit que $\mathcal{D}$ est 2B-cohérent si $\mathcal{D} = \Pi^\Box(\mathcal{D})$.*

Proposition 6.4 (Clôture et filtrage par 2B-cohérence)
Soit $\mathcal{P} = (\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP, tel que $\mathcal{D}$ soit une boîte[10]*. La limite $\mathcal{D}^{(\infty)}$ de la suite définie par l'itération* (6.3)*,*

[9] Cette cohérence apparaît également très souvent en anglais sous le nom de *hull consistency*.
[10] Nous pourrions supposer que $\mathcal{D}$ est un ensemble quelconque, mais cela n'a pas tellement d'intérêt.

est la clôture par 2B-cohérence de $\mathcal{P}$. C'est à dire, $\mathcal{D}^{(\infty)}$ est la plus grande sous-boîte de $\mathcal{D}$ 2B-cohérente. On appelle **filtrage par 2B-cohérence** *la fonction $\Phi_{\mathtt{2B}}$ qui pour tout CSP retourne la clôture par 2B-cohérence de ce CSP (la fonction $\Phi_{\mathtt{2B}}$ vérifie les conditions de la définition 6.2).*

Preuve. S'obtient facilement en adaptant celle de la proposition 6.2. □

On a coutume de dire que la 2B-cohérence est une approximation de l'arc-cohérence "aux bornes", puisqu'en remplaçant une projection par son enveloppe, on ne fait que "combler les trous" (sans toucher aux bornes). Il est utile alors de remarquer deux faits plutôt en opposition à cette formule, le premier étant évident, le second moins. En général :
- Les bornes d'une boîte 2B-cohérente ne forment pas entre elles un système arc-cohérent, dans le sens où elles ne sont pas forcément support entre elles (voir exemple 6.6).
- Contrairement à ce qu'on trouve parfois dans la littérature, la clôture par 2B-cohérence d'un CSP n'est pas l'enveloppe de la clôture par arc-cohérence de ce CSP (voir exemple 6.7).

Exemple 6.6 *Soit $\mathcal{P} = (\{c\}, \{x, y\}, [-1, 1] \times [-1, 1])$ un CSP, avec :*

$$(c) \quad (x - y)^2 = 1.$$

La boîte $[-1, 1] \times [-1, 1]$ est 2B-cohérente (et même arc-cohérente) mais les bornes $x = -1$ et $x = 1$ prennent support en $y = 0$, et inversement les bornes $y = -1$ et $y = 1$ prennent support en $x = 0$.

Exemple 6.7 *Soit $\mathcal{P} = (\{c_1, c_2\}, \{x, y\}, [-1, 1] \times [-1, 1])$ un CSP, avec :*

$$\begin{array}{ll} (c_1) & x^2 = 1, \\ (c_2) & (x - y)^2 = 1. \end{array}$$

La boîte $[-1, 1] \times [-1, 1]$ est 2B-cohérente, car les bornes -1 et 1 du domaine de x sont cohérentes pour c_1 et prennent support en $y = 0$ pour c_2. Les bornes $y = -1$ et $y = 1$ prennent support en $x = 0$.

Or la clôture par arc-cohérence de ce CSP est $[-1, -1] \cup [1, 1]$ pour le domaine de x, $[0, 0]$ pour le domaine de y. L'enveloppe de cet ensemble est $[-1, 1] \times [0, 0] \subsetneq [-1, 1] \times [-1, 1]$.

6.5 w-cohérence versus σ-cohérence

Dans ce paragraphe, on note $\mu(\lceil \mathbf{x} \rceil)$ la *mesure* d'une union, c.a.d. la somme des diamètres des intervalles qui la composent. Exemple : $\lceil \mathbf{x} \rceil = [1, 2] \cup [3, 3.5] \Longrightarrow \mu(\lceil \mathbf{x} \rceil) = 1.5$. On définit également la mesure d'un gruyère $\mathcal{D}$:

$$\mu(\mathcal{D}) := \sum_{x \in \mathcal{X}} \mu(\mathcal{D}_x).$$

Reprenons l'exemple 6.4 p.151. Que l'on applique un filtrage par arc-cohérence ou par 2B-cohérence, il y a une *convergence lente* : la suite $\mathcal{D}^{(k)}$ converge vers $[0, 0] \times [0, 0]$ linéairement sans jamais l'atteindre. En pratique, cela signifie qu'un filtrage prend beaucoup de temps avant d'arriver à isoler le flottant 0, et que beaucoup d'itérations sont effectuées pour retirer des intervalles de plus en plus négligeables par rapport à la taille initiale des domaines.

L'idée couramment utilisée [Lho93, Lho94] est d'interrompre cette propagation lorsque la réduction des domaines est en deçà d'un certain plancher. Tout d'abord, admettons le fait suivant : lorsque le domaine initial $\mathcal{D}^{(0)}$ est un gruyère (en particulier une boîte), pour tout $k \geq 0$, $\mathcal{D}^{(k)}$ est un gruyère[11]. Le principe consiste alors à ne

[11] Nous l'avons déjà mentionné dans la preuve de la proposition 6.3 ; cela vient du fait que les fonctions arithmétiques sont continues sur leur ensemble de définition et que les gruyères sont bornés.

pas prendre en compte le calcul d'une projection Π_x^c ou plus généralement d'un filtrage Φ, si celui-ci ne fait pas diminuer la mesure du gruyère d'au moins w, où w est une constante réelle positive[12] prédéfinie (on appelle **gain** la diminution obtenue).

Nous décrivons maintenant formellement l'itération. Tout d'abord, pour tout filtrage Φ, notons $\Phi^{(w)}$ la fonction suivante :

$$\Phi^{(w)}(\mathcal{D}) := \begin{cases} \mathcal{D}_{temp} \text{ avec } \mathcal{D}_{temp} := \Phi(\mathcal{D}) & \text{si } \mu(\mathcal{D}) - \mu(\mathcal{D}_{temp}) \geq w, \\ \mathcal{D} & \text{sinon.} \end{cases}$$

Notons alors Ψ l'opérateur $(\Pi_{x_n}^{c_m(w)} \circ \ldots \circ \Pi_{x_1}^{c_m(w)}) \circ \ldots \circ (\Pi_{x_n}^{c_1(w)} \circ \ldots \circ \Pi_{x_1}^{c_1(w)})$, c'est à dire la fonction qui à partir d'un espace de recherche $\mathcal{D}$ compose toutes les projections dont le gain est au moins w. Alors nous pouvons définir la suite d'espaces de recherche :

$$\begin{array}{l} \mathcal{D}^{(0)} := \mathcal{D}, \\ \mathcal{D}^{(k)} := \Psi(\mathcal{D}^{(k-1)}). \end{array} \tag{6.4}$$

La propriété obtenue pour $\mathcal{D}^{(\infty)}$ s'appelle la w-arc-cohérence. Remarquons que $\mathcal{D}^{(\infty)}$ est cette fois atteinte en un nombre fini d'itérations, puisque chaque itération qui précède le point fixe fait diminuer d'au moins w la taille d'un domaine. Nous pouvons également définir la suite d'espaces de recherches :

$$\begin{array}{l} \mathcal{D}^{(0)} := \mathcal{D}, \\ \mathcal{D}^{(k)} := \Psi^{\Box}(\mathcal{D}^{(k-1)}) \end{array} \tag{6.5}$$

avec $\Psi^{\Box} = ((\Box\, \Pi_{x_n}^{c_m})^{(w)} \circ \ldots \circ (\Box\, \Pi_{x_1}^{c_m})^{(w)}) \circ \ldots \circ ((\Box\, \Pi_{x_n}^{c_1})^{(w)} \circ \ldots \circ (\Box\, \Pi_{x_1}^{c_1})^{(w)})$. La propriété obtenue pour $\mathcal{D}^{(\infty)}$ s'appelle la w-2B-cohérence. Remarquons que $(\Box\, \Pi_{x_j}^{c_i})^{(w)}$ est différent de $\Box(\Pi_{x_j}^{c_i(w)})$: dans le premier cas le gain w est calculé *après* enveloppe, dans le second *avant*.

Définition 6.12 (w-cohérence) *Soit $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP.*
*On dit que l'espace de recherche $\mathcal{D}$ est w-***arc-cohérent** *si $\Psi(\mathcal{D}) = \mathcal{D}$.*
*On dit que l'espace de recherche $\mathcal{D}$ est w-***2B-cohérent** *si $\Psi^{\Box}(\mathcal{D}) = \mathcal{D}$.*

De plus, il existe une clôture par w-cohérence (w-arc-cohérence ou w-2B-cohérence), dans le sens de la définition 6.6 p.147 et de la proposition 6.2 p.150 ; c'est à dire un plus grand sous-CSP w-cohérent.

Ce type de cohérence possède toutefois un gros inconvénient : le point fixe $\mathcal{D}^{(\infty)}$ atteint ne correspond *pas* forcément à la clôture par w-cohérence[13], mais parfois à un domaine plus petit. La proposition 6.2 ne s'étend pas à la w-cohérence[14]. Par ailleurs, le point fixe dépend de l'ordre des contraintes et/ou des variables. Tout ceci est illustré dans l'exemple suivant.

Exemple 6.8 (Perte de l'unicité du point fixe)
Soit $\mathcal{P} = (\{c_1, c_2, c_3\}, \{x, y, z\}, \mathcal{D} = [0,1] \times [0,2] \times [0,2])$ un CSP avec :

$$\begin{array}{ll} (c_1) & x = 2z \\ (c_2) & y = z \\ (c_3) & x = y \end{array}$$

Calculons le point fixe de $\mathcal{D}^{(k)}$ en prenant $w = 1$.

[12] Dans la littérature, w est introduit comme un nombre de flottants, mais cela ne change rien au propos ; nous préférons utiliser un réel dans la mesure où nous sommes dans un modèle théorique.

[13] Du coup, le terme de "clôture" n'est pas approprié. Toutefois, nous le maintenons par simplicité.

[14] La dernière partie de la preuve contient en effet un argument de *support* qui ne peut plus être appliqué : des valeurs sans support pouvant apparaître dans un CSP w-cohérent.

Supposons que l'ordre de traitement (projection) entre les contraintes soit $c_1 \prec c_2 \prec c_3$. L'espace $\mathcal{D}^{(1)}$ s'obtient ainsi :

	domaine courant	*projection*	*gain*	*nouveau domaine*
$\Psi_x^{c_1}$	$\mathcal{D}_x = [0,1]$	$[0,1]$	0	*(pas de changement)*
$\Psi_z^{c_1}$	$\mathcal{D}_z = [0,2]$	$[0,0.5]$	1.5	$\mathcal{D}_z \leftarrow [0,0.5]$
$\Psi_y^{c_2}$	$\mathcal{D}_y = [0,2]$	$[0,0.5]$	1.5	$\mathcal{D}_y \leftarrow [0,0.5]$
$\Psi_z^{c_2}$	$\mathcal{D}_z = [0,0.5]$	$[0,0.5]$	0	*(pas de changement)*
$\Psi_x^{c_3}$	$\mathcal{D}_x = [0,1]$	$[0,0.5]$	0.5	$(gain \leq w)$
$\Psi_y^{c_3}$	$\mathcal{D}_y = [0,0.5]$	$[0,0.5]$	0	*(pas de changement)*

Le point fixe est atteint. Le résultat est donc $[0,1] \times [0,0.5] \times [0,0.5]$.

Supposons que l'ordre entre les contraintes soit $c_3 \prec c_2 \prec c_1$. L'espace $\mathcal{D}^{(1)}$ s'obtient ainsi :

	domaine courant	*projection*	*gain*	*nouveau domaine*
$\Psi_x^{c_3}$	$\mathcal{D}_x = [0,1]$	$[0,1]$	0	*(pas de changement)*
$\Psi_y^{c_3}$	$\mathcal{D}_y = [0,2]$	$[0,1]$	1	$\mathcal{D}_y \leftarrow [0,1]$
$\Psi_y^{c_2}$	$\mathcal{D}_y = [0,1]$	$[0,1]$	0	*(pas de changement)*
$\Psi_z^{c_2}$	$\mathcal{D}_z = [0,2]$	$[0,1]$	1	$\mathcal{D}_z \leftarrow [0,1]$
$\Psi_x^{c_1}$	$\mathcal{D}_x = [0,1]$	$[0,1]$	0	*(pas de changement)*
$\Psi_z^{c_1}$	$\mathcal{D}_z = [0,1]$	$[0,0.5]$	0.5	$(gain \leq w)$

Le point fixe est atteint. Le résultat est donc $[0,1] \times [0,1] \times [0,1]$.

Or, on peut vérifier "à la main" que la clôture par w-arc-cohérence de $\mathcal{P}$ est l'ensemble $[0,1] \times [0,2[\times [0,1.5[$.

L'exemple peut induire en erreur : on observe que l'opérateur Ψ produit un résultat qui dépend de l'ordre entre les contraintes (puisque $\mathcal{D}^{(1)} = \Psi(\mathcal{D}^{(0)})$ diffère en changeant cet ordre). On en déduit alors que la suite entière $\mathcal{D}^{(k)}$ dépend de l'ordre, et il paraît alors cohérent que le point fixe $\mathcal{D}^{(\infty)}$ dépende de cet ordre. En réalité, le fait que Ψ dépende de l'ordre entre les contraintes n'est pas l'explication à retenir puisque l'opérateur Π lui-même dépend de cet ordre (alors qu'il possède un unique point fixe). Le point-clé est davantage que la clôture par w-cohérence ne joue plus le rôle de borne inférieure vers laquelle les suites $\mathcal{D}^{(k)}$ convergent.

De façon beaucoup plus intuitive, la w-cohérence mélange deux choses contradictoires : le fait de vouloir obtenir une propriété *approchée* (à w près) en effectuant des projections *exactes*. Partant de ce constat, nous proposons un autre type de cohérence, palliant à la fois les problèmes de convergence lente et d'unicité du point fixe. En outre, elle possède une sémantique très proche de la $\mathcal{F}$-arc-cohérence (voir §6.10.1 plus loin) ce qui permet de faire facilement un pont entre le modèle théorique et une machine manipulant des nombres flottants.

Nous appelons cette nouvelle cohérence la σ-cohérence (une ébauche de cette cohérence a déjà été proposée dans [CTN05b]). L'idée derrière cette cohérence est simple : forcer les bornes des intervalles a coïncider avec une subdivision[15] prédéfinie σ des réels plus ou moins fine.

Définition 6.13 (σ-tranche) *Soit $\sigma = \{-\infty, \ldots, +\infty\}$ une subdivision finie de $\mathbb{R} \cup \{-\infty, +\infty\}$. Pour tout $x \in \mathbb{R}$ on appelle* **tranche** *de x et on note $\sigma(x)$, l'unique intervalle $[f_1, f_2]$ dont les bornes appartiennent à σ et vérifiant $f_1 < x < f_2$.*

Remarquons que dans le cas où x est lui-même un élément de σ, la tranche $\sigma(x)$ comporte trois éléments[16] (le prédécesseur de x, x, et son successeur).

[15] Quoique ce ne soit pas obligatoire, nous choisirons par simplicité la même subdivision pour chaque variable.

[16] Nous pourrions aussi établir d'autres conventions : poser $\sigma(x) = [x, x^{+1}]$ ou $\sigma(x) = [x^{-1}, x]$. Mais le choix de "doubler" la tranche est plus judicieux pour des questions d'implémentation (si x est une solution, je ne peux supprimer ni $[x^{-1}, x]$, ni $[x, x^{+1}]$). Voir §6.10.1.

La figure 6.3 illustre alors le principe de la σ-arc-cohérence : si deux réels sont supports entre eux, alors leurs tranches sont support entre elles.

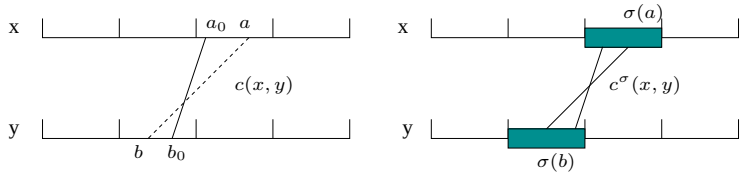


FIG. *6.3: Tranches support entre elles*

Une manière intéressante de formaliser cette cohérence sans avoir à tout redéfinir (opérateur de projection, clôture, etc...) consiste à définir à partir d'une contrainte c la contrainte notée c^σ qui est satisfaite pour un n-uplet de réels (e.g., (a,b) sur la figure) ssi il existe un n-uplet suffisamment "proche", c.a.d., dans les mêmes tranches ((a_0,b_0) sur la figure) satisfaisant c :

Définition 6.14 (σ-approximation d'une contrainte) *Soit c une contrainte n-aire. On appelle σ-approximation de c et on note c^σ la contrainte*[17] *telle que*

$$\forall (x_1,\ldots,x_n)\in\mathbb{R}^n \quad c^\sigma(x_1,\ldots,x_n) \iff \exists (y_1,\ldots,y_n)\in\sigma(x_1)\times\ldots\times\sigma(x_n) \quad c(y_1,\ldots,y_n).$$

L'avantage de cette définition est que nous pouvons conserver notre modèle théorique, c.a.d. raisonner sur les réels et appliquer ce qui a déjà été fait à la *σ-approximation* du CSP :

Définition 6.15 (σ-cohérence) *Soient $(\{c_1,\ldots,c_m\},\mathcal{X},\mathcal{D})$ un CSP, et σ une subdivision finie de $\mathbb{R}\cup\{-\infty,+\infty\}$. On dit que $\mathcal{D}$ est σ-arc-cohérent (resp. σ-2B-cohérent) si $\mathcal{D}$ est arc-cohérent (resp. 2B-cohérent) vis-à-vis du CSP $(\{c_1^\sigma,\ldots,c_m^\sigma\},\mathcal{X},\mathcal{D})$.*

Exemple 6.9 *Reprenons le CSP de l'exemple 6.8, et σ tel que*

$$\sigma=\{-\infty,\ldots,\ -0.5,\ 0,\ 0.5,\ 1,\ 1.5,\ 2,\ 2.5,\ \ldots,+\infty\}.$$

Calculons le point-fixe par σ-arc-cohérence.

	domaine courant	*projection de c*	*projection de c^σ*	*nouveau domaine*
$\Psi_x^{c_1}$	$\mathcal{D}_x=[0,1]$	$[0,1]$	$[-0.5,1.5]\cap[0,1]$	*(pas de changement)*
$\Psi_z^{c_1}$	$\mathcal{D}_z=[0,2]$	$[0,0.5]$	$[-0.5,1]\cap[0,2]$	$\mathcal{D}_z\leftarrow[0,1]$
$\Psi_y^{c_2}$	$\mathcal{D}_y=[0,2]$	$[0,1]$	$[-0.5,1.5]\cap[0,2]$	$\mathcal{D}_y\leftarrow[0,1.5]$
$\Psi_z^{c_2}$	$\mathcal{D}_z=[0,1]$	$[0,1]$	$[-0.5,1.5]\cap[0,1]$	*(pas de changement)*
$\Psi_x^{c_3}$	$\mathcal{D}_x=[0,1]$	$[0,1]$	$[-0.5,1.5]\cap[0,1]$	*(pas de changement)*
$\Psi_y^{c_3}$	$\mathcal{D}_y=[0,1.5]$	$[0,1]$	$[-0.5,1.5]\cap[0,1.5]$	*(pas de changement)*

Le point fixe est atteint. Le résultat est donc $[0,1]\times[0,1.5]\times[0,1]$. On peut vérifier qu'il s'agit bien de la clôture par σ-arc-cohérence de $\mathcal{P}$.

Les avantages de la σ-cohérence sont nets :

[17] Remarquons que cette définition en « intention » d'une contrainte sur les réels est la seule dans ce chapitre qui n'est pas une équation.

– Par construction, tout CSP σ-cohérent est un gruyère où la taille de chaque union est bornée par la taille de la subdivision (une preuve directe est proposée dans [CTN05b]). Le gruyère est bien sûr une boîte dans le cas de la σ-2B-cohérence.
– Le point fixe du filtrage par σ-cohérence est unique (de par l'unicité du point fixe de l'arc-cohérence).
– Le phénomène de convergence lente est évité en choisissant σ de façon appropriée.
– L'obtention de la propriété de σ-cohérence sur les flottants est plus aisée que la w-cohérence : il suffit que σ soit un sous-ensemble de flottants (voir §6.10.2 p.171).

En contrepartie, le filtrage par σ-cohérence paraît moins précis que le filtrage par w-cohérence, puisqu'il suffit que la borne d'une tranche ait un support pour que la tranche entière soit maintenue dans le domaine. Cette perte de précision est effectivement vraie *localement* au niveau d'une projection, et elle constitue justement le prix à payer pour obtenir un point fixe unique. On ne peut rien conclure en revanche au niveau du filtrage lui-même : si nous reprenons l'exemple 6.8 avec cette fois la boîte initiale $\mathcal{D} = [0,1] \times [0,1.9] \times [0,1.4]$, aucune projection ne peut diminuer de plus de 1 la taille d'un domaine. Cette boîte est donc w-cohérente, or, la clôture par σ-cohérence est toujours $[0,1] \times [0,1.5] \times [0,1]$, c'est à dire un domaine plus précis. Ces deux cohérences ne sont donc pas comparables en théorie. Une étude comparative expérimentale mériterait d'être menée pour pouvoir établir un rapport entre les deux, mais ce travail n'a pas été réalisé dans le cadre de cette thèse pour deux raisons. D'une part, il est peu probable que la σ-cohérence soit significativement meilleure (elle n'a d'ailleurs pas été conçue dans ce but). D'autre part, il demeure un grand doute sur l'utilité en pratique d'un point fixe unique. Dans le cadre d'une recherche de solutions, seule l'efficacité du filtrage compte ; les propriétés vérifiées par les domaines n'ont a priori qu'un intérêt théorique[18].

6.6 Rognage

Nous présentons dans ce paragraphe un exemple de cohérence "à deux niveaux" que nous illustrons d'abord sur un CSP discret.

Il existe un filtrage en domaine discret connu sous le nom de SAC (*singleton arc-consistency*) [DB97b, BE04, BR05]. Il consiste à balayer le domaine de chaque variable x, et à appliquer à chaque valeur v de ce domaine un *test de réfutation* qui consiste à fabriquer une copie $\mathcal{D}'$ de l'espace de recherche courant où le domaine de x est restreint à la valeur $\{v\}$ puis à appliquer un filtrage par arc-cohérence sur $\mathcal{D}'$. Si le résultat de ce filtrage est $\emptyset$, alors aucune solution n'existe dans $\mathcal{D}'$, et comme celui-ci inclut intégralement les domaines des autres variables, cela signifie qu'aucune solution dans $\mathcal{D}$ ne peut comporter la valeur v. Dans ce cas, on écrit que le test $AC(.) = \emptyset$? a réussi pour v, ce qui signifie qu'elle peut être supprimée.

De façon plus abstraite, supposons disposer d'un filtrage Φ quelconque qui respecte la définition 6.2 p.145. Appliqué à un espace de recherche, l'opérateur Φ supprime un certain nombre de valeurs (*contractance*) ne pouvant appartenir à une solution (*conservatisme*). Si le résultat de ce filtrage est l'ensemble vide, alors, il n'existe aucune solution. Un filtrage Φ permet donc de définir le test de réfutation[19] "$\Phi(.) = \emptyset$?" qui peut être appliqué à n'importe quel sous-espace $\mathcal{D}'$ de l'espace de recherche courant $\mathcal{D}$. Par *monotonie*, $\mathcal{D}'$ étant plus petit, le test a plus de chance de prouver l'infaisabilité qu'avec $\mathcal{D}$. Le cas échéant, il ne reste plus qu'à remplacer $\mathcal{D}$ par $\mathcal{D} \backslash \mathcal{D}'$. La difficulté est de savoir représenter $\mathcal{D} \backslash \mathcal{D}'$ (voir figure 6.4).

Passons aux CSP continus, où nous introduisons une cohérence basée sur le même principe. Nous choisissons $\Phi = \Phi_{\mathtt{2B}}$ et représentons les espaces de recherche par de simples boîtes ; ces choix sont justifiés par la pratique[20]. La réfutation est appliquée avec la restriction suivante : au lieu de chercher à ce que chaque valeur soit 2B-

[18] Pour nuancer ce propos, signalons que l'unicité du point fixe a toute de même l'avantage de faciliter le rejeu d'expérimentations (donc la reproductibilité d'erreur) car il devient inutile de spécifier l'ordre des contraintes.

[19] « Réfutation » car Φ prouve l'absence de solution lorsqu'il retourne l'ensemble vide (dans le cas contraire, on ne peut pas statuer sur l'existence ou non de solution).

[20] Nous utiliserons également comme test de réfutation une simple évaluation par intervalles du sous-domaine (voir §6.7).

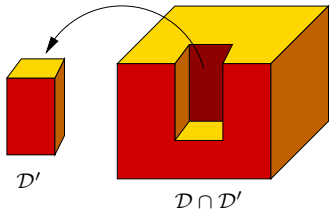


FIG. *6.4: Réfutation d'un sous-espace de recherche*

cohérente (c.a.d., non réfutable par 2B-cohérence), nous nous limitons à ce que les bornes des domaines le soient. Pour un espace de recherche courant, le fait de réduire les bornes d'un domaine jusqu'à tomber sur une valeur 2B-cohérence porte le nom de *rognage*[21] :

Définition 6.16 (Rognage d'une variable) *Soient $\mathcal{C}$ un ensemble de contraintes et $\mathcal{X} = \{x_1, \ldots, x_n\}$ un ensemble de variables. On appelle* **rognage** *de x_i, et on note $R^{\mathsf{2B}}_{x_i}$ la fonction qui à un espace de recherche $\mathcal{D} = \mathcal{D}_1 \times \ldots \times \mathcal{D}_n$ retourne l'espace de recherche $\mathcal{D}'$ vérifiant*

$$\forall j \neq i \quad D'_{x_j} = \mathcal{D}_{x_j},$$
$$D'_{x_i} := \square\{v_i \in \mathcal{D}_{x_i} \mid \Phi_{\mathsf{2B}}(\mathcal{C}, \mathcal{X}, \mathcal{D}_{x_1} \times \ldots \times \mathcal{D}_{x_{i-1}} \times \{v_i\} \times \mathcal{D}_{x_{i+1}} \times \ldots \times \mathcal{D}_{x_n}) \neq \emptyset\}.$$

Remarquons que le rognage - contrairement à la projection - ne dépend pas d'une seule contrainte et n'est pas local : l'incohérence d'une valeur dépend du CSP entier. Il n'est donc plus possible d'utiliser l'algorithme de propagation décrit au §6.1.2 pour propager les réductions d'une variable à l'autre : lorsque le domaine d'une variable est réduit, toutes les autres variables sont a priori impactées. Le point fixe du rognage permet de définir une nouvelle cohérence, appelée 3B-cohérence [Lho93, Lho94].

Définition 6.17 (3B-cohérence) *Soit $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP. On dit que l'espace de recherche $\mathcal{D}$ est* **3B-cohérent** *si*

$$\forall x \in \mathcal{X} \quad R^{\mathsf{2B}}_x(\mathcal{D}) = \mathcal{D}.$$

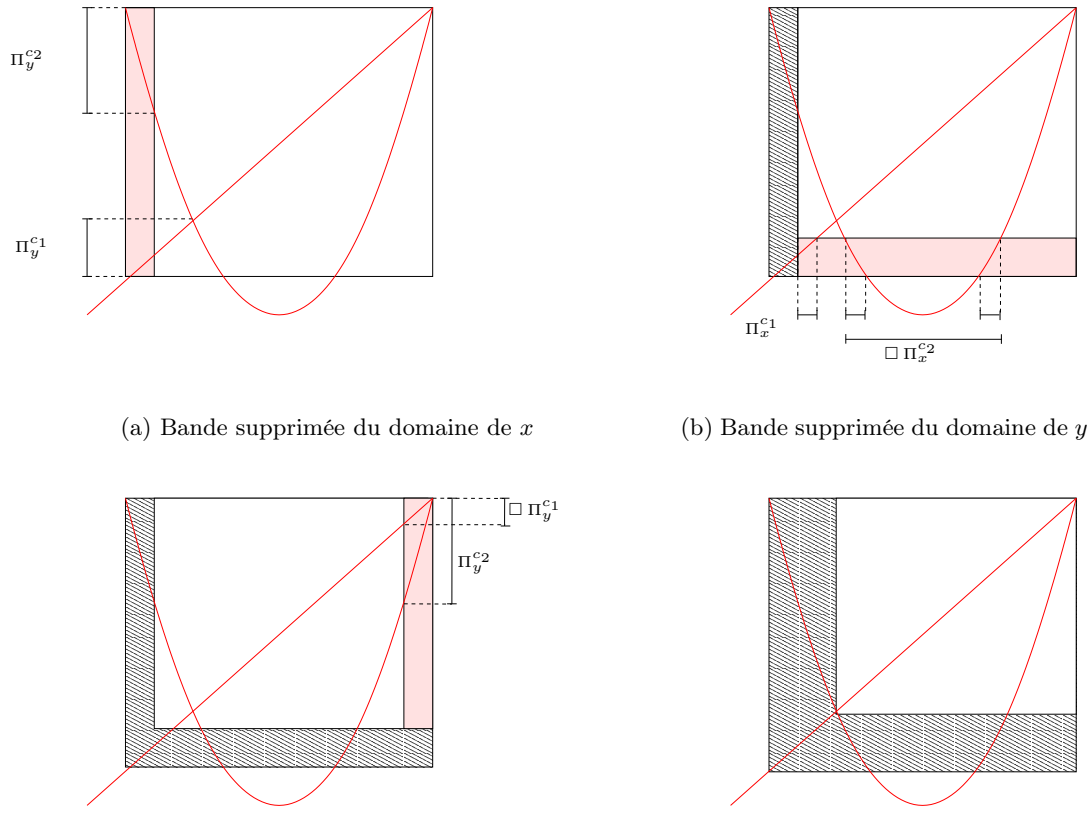
Les définitions de clôture par 3B-cohérence et de filtrage par 3B-cohérence sont similaires à celles de l'arc-cohérence ou de la 2B-cohérence. Enfin, la 3B-cohérence peut être elle-même utilisée pour réfuter. On obtient alors un filtrage à trois niveaux, appelé 4B-cohérence, et ainsi de suite. La k-B cohérence est définie ainsi (mais en pratique la 4B-cohérence est déjà beaucoup trop chère). On peut voir que dans son niveau le plus imbriqué, le rognage de la kB-cohérence cherche à réfuter des espaces de recherche où $k-1$ variables sont instanciées à des valeurs. Il en découle que la kB-cohérence calcule la *cohérence globale*, c'est à dire l'enveloppe de toutes les solutions, d'un CSP à $k-1$ variables. Remarquons qu'il existe des moyens beaucoup plus efficaces de calculer la cohérence globale (cf. par exemple [JKDW01] §5.4.2).

6.6.1 Implémentation du rognage

Considérons une variable x, de domaine $\mathcal{D}_x$. Le rognage est lui-même basé sur un procédé itératif qui converge vers le résultat, c.a.d., un nouveau domaine $\mathcal{D}_x$ dont les bornes sont 2B-cohérentes. Supposons que $\mathcal{D}_x$ soit l'intervalle $[a, b]$. Si le filtrage 2B prouve que le sous-intervalle $[a, a+\epsilon]$ ($\epsilon \geq 0$) est infaisable, alors l'intervalle $[a, a+\epsilon]$ peut être supprimé du domaine de x, ce qui revient à ramener la borne inférieure de $\mathcal{D}_x$ à $a+\epsilon$. Le sous-CSP obtenu en restreignant $\mathcal{D}_x$ à un sous-intervalle tel que $[a, a+\epsilon]$ situé à une extrémité du domaine

[21] *Shaving* en anglais.

s'appelle une **bande** (voir figure 6.5). Lorsqu'une bande ne peut plus être supprimée, l'idée alors est de prendre une bande plus fine (choisir ϵ plus petit). Intuitivement, l'idée est d' "emprisonner" la plus petite valeur 2B-cohérence du domaine dans des bandes dont la largeur tend vers zéro. La figure 6.5 ci-dessous illustre le filtrage par 3B-cohérence sur l'exemple-type 6.5 p.152. On peut y voir des exemples de bandes supprimées par 2B-cohérence (les projections des deux contraintes ont une intersection vide). Remarquons que la 3B-cohérence calcule la cohérence globale pour un CSP à deux variables, ce que nous pouvons observer sur la figure 6.5.(d).



(a) Bande supprimée du domaine de x

(b) Bande supprimée du domaine de y

(c) Bande non supprimée du domaine de x

(d) Point fixe de la 3B-cohérence

FIG. *6.5: Rognage*

L'algorithme n'est toutefois pas aussi simple. Il peut y avoir nécessité de réduire la largeur de bande dans une zone sans valeur 2B-cohérente. La figure 6.6 représente ce phénomène avec une fonction univariée dont on cherche le zéro. Dans ce cas, le filtrage $\Phi_{\texttt{2B}}$ se ramène à une simple évaluation et le pessimisme de l'évaluation par intervalles entraîne un découpage plus fin autour d'un pseudo-zéro. Une fois cette zone passée, il faut ré-augmenter la largeur de bande.

Supposons que nous travaillons sur la borne gauche. Sans détailler l'implémentation, l'idée principale, illustrée sur la figure 6.7, consiste à gérer trois intervalles :

- `intérieur`, qui représente l'intervalle à droite de la borne cherchée le plus grand obtenu jusqu'à présent. Un moyen d'élargir cet intervalle consiste à applique le "test de la borne" (si une borne quelconque est 2B-cohérente, `intérieur` peut être étendu jusqu'à celle-ci). `intérieur` est initialisé à un intervalle dégénéré représentant la borne droite de l'intervalle.
- `frontière` représente la partie de l'intervalle contenant la borne cherchée, et que l'on essaye de réduire. `frontière` est initialisé au domaine entier.
- `courant` représente le sous-intervalle que l'on tente de réfuter par 2B-cohérence à chaque étape (partie hachurée).

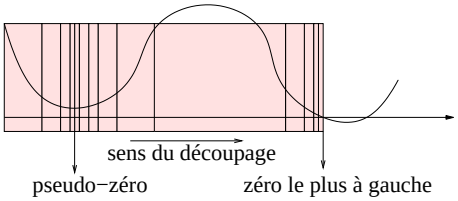


FIG. *6.6: Adaptation dynamique de la largeur des bandes.*

Il existe un grand nombre de stratégies pour passer de l'étape initiale à l'étape finale (approche *optimiste* : commencer par des bandes le plus large possible, puis diviser la largeur à chaque échec, approche *pessimiste* : commencer par une bande minimale puis doubler à chaque succès, entrelacement avec des "tests de borne", etc...)

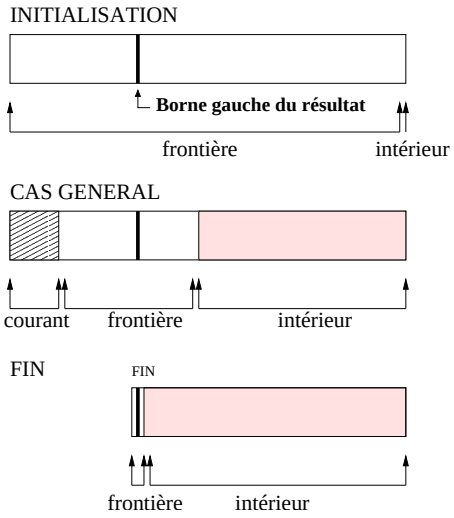


FIG. *6.7: Implémentation du rognage.*

6.7 Box-cohérence

La Box-cohérence [BMVH94, VHMD97, VHMK97, GGB99] est la cohérence obtenue en remplaçant dans la définition du rognage (définition 6.16) l'opérateur $\Phi_{\texttt{2B}}$ par l'opérateur $\Phi_{\texttt{eval}}$, qui consiste à évaluer les contraintes via l'extension naturelle (cf. §2.6.3 p.24) et à tester la présence de 0 dans le résultat de cette évaluation.

Une boîte est donc **box-cohérente** si $\quad \forall x \in \mathcal{X}\ R_x^{\texttt{eval}}(\mathcal{D}) = \mathcal{D}$.

Considérons un CSP $(\{c\}, \{x, y_1, \ldots, y_n\}, \mathbf{x} \times \mathbf{y}_1 \times \ldots \times \mathbf{y}_n)$ réduit à une seule contrainte c :

$$(c) \quad f(x, y_1 \ldots, y_n) = 0.$$

Le rognage de la variable x consiste alors à trouver la plus petite valeur x_l et la plus grande valeur x_u du domaine $\mathbf{x}$ telles que

$$0 \in f(x_l, \mathbf{y}_1, \ldots, \mathbf{y}_n), \quad 0 \in f(x_u, \mathbf{y}_1, \ldots, \mathbf{y}_n).$$

L'intervalle $[x_l, x_u]$ est alors une approximation de la projection de c sur x que l'on peut noter $\Pi_x^{c(\texttt{BOX})}(\mathcal{D})$. Une première manière de calculer $[x_l, x_u]$ est d'utiliser le rognage « par bandes » décrit ci-dessus et d'appliquer le

test $0 \in f(\mathbf{x}', \mathbf{y}_1, \ldots, \mathbf{y}_n)$ où $\mathbf{x}'$ désigne l'intervalle courant. Une manière de procéder plus efficace consiste à voir la fonction f comme une fonction univariée (d'une seule variable) et de n paramètres $y_1, \ldots, y_n$. Trouver x_l et x_u consiste à trouver le zéro le plus à gauche et le plus à droite de f. Nous pouvons utiliser alors la méthode de Newton paramétrique version « épaisse » décrite au chapitre 1 (voir prop. 2.18 p.42).

Considérons maintenant le cas général de m contraintes $c_1, \ldots, c_m$ avec $c_i : f_i(x, y_1, \ldots, y_n) = 0$. Le traitement de la variable x peut se faire en rognant avec $\Phi_{\texttt{eval}}$. On prouve facilement que ce rognage équivaut à rogner indépendamment la variable avec chaque contrainte, c.a.d. que l'intervalle produit vérifie

$$R_x^{\texttt{eval}}(\mathcal{D}) = \Pi_x^{c_1(\texttt{BOX})} \circ \ldots \circ \Pi_x^{c_m(\texttt{BOX})}(\mathcal{D}),$$

et ce quel que soit l'ordre entre les contraintes. Le point fixe de la box-cohérence équivaut donc à un point fixe des projections $\Pi_x^{c(\texttt{BOX})}$:

$$\mathcal{D} \text{ est box-cohérent} \iff \forall (c, x) \in \mathcal{C} \times \mathcal{X} \quad \Pi_x^{c(\texttt{BOX})}(\mathcal{D}) = \mathcal{D}.$$

On en conclut qu'il est possible d'appliquer la box-cohérence avec la boucle d'`AC3` en travaillant directement avec des couples (c, x) plutôt que de travailler avec des variables uniquement et d'évaluer « en bloc » toutes les contraintes à chaque rognage. L'avantage de travailler avec des couples est de pouvoir propager intelligemment les réductions, ce qu'il n'est pas possible de faire dans l'autre cas.

Bien entendu, la projection $\Pi_x^{c(\texttt{BOX})}$ peut se calculer aussi avec l'opérateur de Newton paramétrique. Souvent, l'opérateur de Newton ne permet pas de calculer directement le zéro le plus à gauche et le plus à droite. En particulier, si ces deux zéros sont des continuums de solutions distincts, la dérivée s'annule dans l'intervalle initial et aucune réduction de borne n'est possible. Le problème de dépendance dans le calcul des dérivées peut être également à l'origine d'une absence de réduction. On combine donc cet opérateur avec un découpage des domaines tel que celui de la figure 6.7.

Il existe un moyen simple d'améliorer cet algorithme[22]. Supposons que la contrainte soit $f(x, y) = 0$, et notons $\mathbf{x}$ le domaine de la variable x. L'idée est de choisir un point de découpage ayant plus de chance d'isoler un zéro de f que le milieu de $\mathbf{x}$. Ce point peut être obtenu en calculant une étape du Newton scalaire (paramétrique) mais en restreignant le domaine de x à une borne de $\mathbf{x}$. La convergence vers le zéro le plus à gauche (ou à droite) devient ainsi quadratique. Ceci est illustré sur la figure 6.8.

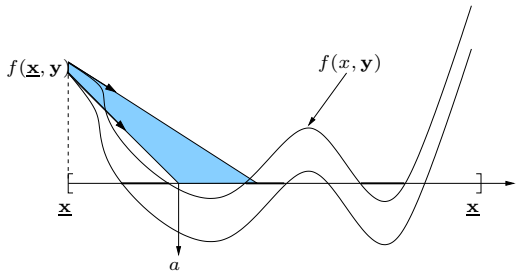


FIG. *6.8: Accélération du calcul de la box-cohérence. Le point a permet de séparer* $\mathbf{x}$ *en deux sous-intervalles* $[\underline{\mathbf{x}}, a]$ *et* $[a, \overline{\mathbf{x}}]$*, donc d'isoler l'un des continuums de solution.*

Il est prouvé dans [Col98, CDR99] qu'en l'absence d'occurrence multiple de x dans la contrainte c, on a $\Pi_x^{c(\texttt{BOX})} = \Pi_x^c$, autrement dit, que le rognage de la variable x équivaut à une projection. Or, nous allons voir dans le paragraphe suivant qu'il est justement possible de calculer une projection en l'absence d'occurrence multiple. Des algorithmes hybrides 2B-cohérence/Box-cohérence ont été mis au point [BGGP99] depuis pour limiter l'invocation du Newton paramétrique au cas où la variable rognée possède de multiples occurrences.

[22] Cette astuce nous a été soufflée par Luc Jaulin.

6.8 L'algorithme HC4Revise

Jusqu'ici, le modèle théorique suppose que nous disposons d'une *boîte noire* calculant la projection de n'importe quelle contrainte (que ce soit pour l'arc-cohérence ou la 2B-cohérence). Nous allons donner un algorithme [BGGP99, Leb99, GB01] calculant l'évaluation (cf. définition 6.9) et la projection d'une contrainte quelconque $f(x_1, \ldots, x_n) = 0$, sans occurrence multiple de variable. Il peut également être utilisé pour des contraintes avec occurrences multiples, mais pour obtenir seulement une approximation extérieure de la projection. Nous verrons alors que cette approximation permet toutefois de calculer la 2B-cohérence (ou arc-cohérence) mais d'un CSP "dérivé", obtenu par une *décomposition* du CSP original. Nous commençons par introduire la décomposition.

6.8.1 Décomposition de CSP

D'un point de vue pratique, il peut être préférable de manipuler uniquement des contraintes primitives (cf. tableau 6.1 p.149) : la projection est plus facile à obtenir. C'est donc pour ces raisons qu'apparaît la notion de décomposition.

La décomposition (en contraintes primitives) d'un CSP consiste en un CSP sémantiquement équivalent mais ne comportant que des contraintes primitives. Le coût de cette simplification est l'ajout de variables intermédiaires dites *implicites*. Cette décomposition se comprend mieux à travers un exemple :

Exemple 6.10 (Décomposition) *La contrainte $c : (x-y)^2 = z$ est équivalente au système formé des trois contraintes primitives $\{c'_1, c'_2, c'_3\}$ suivantes impliquant cinq variables (x, y, z, w_1, w_2) :*

$$\begin{array}{ll} (c'_1) & w_1 = x - y \\ (c'_2) & w_2 = w_1^2 \\ (c'_3) & z = w_2 \end{array}$$

dans le sens où $c(x,y,z) \iff \big(\exists w_1 \in \mathbb{R}\big)\big(\exists w_2 \in \mathbb{R}\big) \quad \Big(c'_1(x,y,w_1)$ *et* $c'_2(w_1,w_2)$ *et* $c'_3(z,w_2)\Big)$.
Les variables w_1 et w_2 sont appelées variables **implicites**. *Leur domaine initial est $\mathbb{R} =]-\infty, +\infty[$.*

Les propositions suivantes montrent que la clôture par arc-cohérence (ou 2B-cohérence) d'un CSP donne un espace de recherche plus réduit que la clôture de la décomposition de ce CSP.

Proposition 6.5 *Soit $\mathcal{P} = (\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP, $\mathcal{P}' = (\mathcal{C}', \mathcal{X} \cup \{w_1, \ldots, w_k\}, \mathcal{D} \times \mathcal{D}')$ sa décomposition, avec $\mathcal{D}' := \mathbb{R}^k$. Soit $\tilde{\mathcal{D}}$ la clôture par arc-cohérence (resp. 2B-cohérence) de $\mathcal{P}$, $\underset{\sim}{\mathcal{D}} \times \underset{\sim}{\mathcal{D}'}$ la clôture par arc-cohérence (resp. 2B-cohérence) de $\mathcal{P}'$. Alors $\tilde{\mathcal{D}} \subseteq \underset{\sim}{\mathcal{D}}$.*

Preuve. Le lecteur se référera à [CDR99] pour la comparaison entre les clôtures par 2B-cohérence. La preuve pour les clôtures par arc-cohérence repose sur le même principe : en décomposant les contraintes, les projections agissent sur un sous-problème plus petit, et on perd l'équivalence. Ceci se produit particulièrement lorsqu'on décorelle plusieurs occurrences d'une même variable. □

6.8.2 L'algorithme

L'algorithme HC4Revise repose lui-même sur deux procédures :

1. l'évaluation d'une contrainte *primitive*.
2. la projection d'une contrainte *primitive*.

Cet algorithme implémente une *projection automatique*, à l'instar de la différenciation automatique [Gri00], avec un double parcours de type synthèse/héritage de l'arbre[23] de syntaxe de la contrainte.

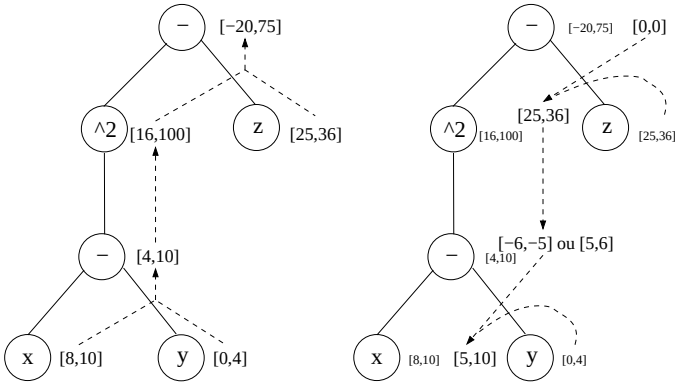
Commençons par l'évaluation. L'évaluation d'une contrainte $f(x) = 0$ correspond au parcours "en remontée" de l'arbre. On commence par associer aux feuilles de l'arbre représentant une variable x le domaine $\mathcal{D}_x$, et aux feuilles représentant une constante c le singleton $\{c\}$.

On applique alors récursivement l'évaluation des fonctions primitives (voir la figure 6.9(a) pour une illustration de l'évaluation avec des arguments intervalles). Pour un noeud w_2 représentant une fonction élémentaire $w_2 = e(w_1)$ (w_1 et w_2 sont des variables implicites, elles correspondent à des sous-expressions de f), si le noeud-fils w_1 a été évalué et qu'on lui a associé un domaine $\mathcal{D}_{w_1}$, alors on associe au noeud w_2 le domaine

$$\mathcal{D}_{w_2} \leftarrow range(e, \mathcal{D}_{w_1}).$$

L'évaluation d'un noeud représentant un opérateur binaire $\star$ suit le même principe : une fois évalués, les deux noeuds-fils w_1 et w_3, on écrit

$$\mathcal{D}_{w_2} \leftarrow range(w_1 \star w_3, w_1 \in \mathcal{D}_{w_1}, w_3 \in \mathcal{D}_{w_3}).$$



(a) Évaluation de $(x-y)^2 = z$ avec $\mathcal{D}_x = [8, 10]$, $\mathcal{D}_y = [0, 4]$ et $\mathcal{D}_z = [25, 36]$.

(b) Projection sur x

FIG. *6.9:* HC4Revise

Venons-en au calcul de projection, et notons x la variable sur laquelle on projette. Il est nécessaire d'avoir fait au préalable une évaluation de la contrainte, et donc que chaque variable implicite w ait un domaine $\mathcal{D}_w$ correspondant à l'image de la sous-expression correspondante. La projection se calcule par un parcours "en descente" de l'arbre de syntaxe (voir la figure 6.9(b)). La descente se fait en suivant la branche dont x est la feuille (rappelons qu'il est supposé que la contrainte ne contient qu'une seule occurrence de la variable x, il n'y a donc qu'une seule branche possible).

On commence par affecter 0 au domaine de la racine, puis :
– Pour un noeud w_2 représentant une fonction élémentaire $w_2 = e(w_1)$, on effectue l'opération suivante :

$$\mathcal{D}_{w_1} \leftarrow \Pi_{w_1}^{w_2=e(w_1)}(\mathcal{D}_{w_1} \times \mathcal{D}_{w_2}).$$

– Pour un noeud w_2 représentant une fonction élémentaire $w_2 = w_1 \star w_3$, où w_1 est la branche contenant x, on effectue l'opération suivante :

$$\mathcal{D}_{w_1} \leftarrow \Pi_{w_1}^{w_2=w_1\star w_3}(\mathcal{D}_{w_1} \times \mathcal{D}_{w_2} \times \mathcal{D}_{w_3}).$$

[23]Remarquons qu'il est possible de factoriser certains calculs en utilisant un DAG (*directed acyclic graph*) au lieu d'un arbre.

Proposition 6.6 (Correction d'HC4Revise [BGGP99, GB01])
Pour toute contrainte c ne présentant pas d'occurrence multiple de variable, HC4Revise *calcule les projections de c.*

Preuve.

Soit $\mathcal{P}$ la décomposition de c.

1. Le réseau de contraintes de $\mathcal{P}$ est un arbre (l'arbre de syntaxe de c)
2. Évaluer une contrainte primitive avec HC4Revise revient à effectuer une projection sur la variable implicite représentant le noeud courant (comme w_1 et w_2 dans l'exemple 6.10). Rappelons en effet que les domaines initiaux des variables implicites sont $\mathbb{R}$. Une telle projection est équivalente à une étape de DAC (*directed arc consistency*) en domaine fini [DP87].
3. Chaque fois qu'une contrainte primitive est projetée, il s'agit de nouveau d'une étape de DAC.
4. Il est montré dans [Fal94] qu'effectuer un double parcours d'un arbre de contraintes en appliquant DAC permettait d'obtenir l'arc-cohérence du CSP. Or, chaque parcours effectué par HC4Revise est équivalent à un parcours de DAC, des feuilles de l'arbre (les variables de c) vers la racine, et inversement.
5. L'arc-cohérence d'un arbre de contraintes est équivalente à la *cohérence globale* [Fre82]. Donc HC4Revise donne la cohérence globale de $\mathcal{P}$. En d'autres mots, chaque valeur dans le domaine d'une variable de c appartient à une solution de $\mathcal{P}$, c.a.d., satisfait la contrainte c, puisque c et $\mathcal{P}$ sont sémantiquement équivalent (la cohérence globale de $\mathcal{P}$ est équivalente à l'arc-cohérence de c). Finalement, le double parcours d'HC4Revise calcule bien la projection. □

De nombreuses optimisations sont également proposées dans [BGGP99, GB01]. Indiquons par exemple que le double parcours d'arbre peut servir à calculer simultanément la projection sur chaque variable x de la contrainte, ainsi que la dérivée partielle $\partial f/\partial x$. Une variante très utilisée de HC4Revise consiste à ne manipuler que des intervalles, c'est à dire, à systématiquement prendre l'enveloppe de l'évaluation ou de la projection d'une contrainte primitive. Nous notons cette variante HC4Revise$_\Box$. Un « piège » consiste à croire qu'en l'absence d'occurrence multiple de variables, HC4Revise$_\Box$ calcule l'enveloppe de la projection d'une contrainte non primitive (puisqu'on ne fait que "combler les trous" à chaque étape intermédiaire) ; l'intérêt étant alors d'utiliser cette variante pour obtenir la clôture par 2B-cohérence. Or ce n'est pas le cas, comme le montre l'exemple 6.11.

Exemple 6.11 *Soit $\mathcal{P} = (\{c\}, \{x, y\}, \mathcal{D})$ le CSP suivant :*
$\mathcal{D}_x = [0, 1]$, $\mathcal{D}_y = [-1, 1]$ *et*
$(c) \quad (x \times y)^2 = 1$.

Remarquons que la borne $x = 0$ ne fait pas partie de la projection $\Pi_x^c(\mathcal{D})$, puisqu'elle n'a pas de support dans $\mathcal{D}_y$. Or, appliquer HC4Revise$_\Box$ *ne permet pas de réduire cette borne. En effet, la projection de $w^2 = 1$ sur w, où w est le noeud $x \times y$, produira (après enveloppe) l'intervalle $[-1, 1]$ qui inclut le "mauvais" support $w = 0$ de $x = 0$. En conséquence, l'intervalle $\mathcal{D}_x$ ne sera pas modifié.*

Si on considère par contre le CSP obtenu en décomposant la contrainte de l'exemple précédent, c.a.d.,
$\mathcal{P}' = (\{c_1, c_2\}, \{x, y, w\}, \mathcal{D} \times \mathbb{R})$, avec $\mathcal{D}_x = [0, 1]$, $\mathcal{D}_y = [-1, 1]$, $\mathcal{D}_w = \mathbb{R}$
$(c1) \quad x \times y = w$,
$(c2) \quad w^2 = 1$,
on prouve alors (en adaptant la preuve de la proposition 6.6) que HC4Revise$_\Box$ appliqué sur la contrainte c calcule la 2B-cohérence de $\mathcal{P}'$ (la boîte obtenue est $[0, 1] \times [-1, 1]$, et celle-ci est bien la restriction de la boîte 2B-cohérence $[0, 1] \times [-1, 1] \times [-1, 1]$ de $\mathcal{P}'$). Ce résultat se généralise :

Proposition 6.7 (Point fixe d'HC4Revise [BGGP99, GB01])
Soit $\mathcal{P}$ un CSP, $\mathcal{P}'$ la décomposition de $\mathcal{P}$. Si on note Π' l'opérateur composant les $m \times n$ pseudo-projections possibles calculées avec HC4Revise$_\Box$*, alors le point fixe de l'itération $\mathcal{D}^{(k+1)} \leftarrow \Pi'(\mathcal{D}^{(k)})$ est la restriction de la clôture par 2B-cohérence de $\mathcal{P}'$ aux domaines des variables de $\mathcal{P}$.*

Remarque 6.2 *L'opérateur* `HC4Revise` *(ou* $\texttt{HC4Revise}_{\square}$*) peut être facilement adapté pour traiter une contrainte comportant des occurrences multiples de variables. Il suffit, dans la phase descendante, de calculer autant de projections qu'il y a d'occurrences de la même variable, chaque projection se faisant pour une feuille représentant une occurrence différente. Les domaines obtenus pour les différents feuilles peuvent être ensuite intersectés. Remarquons alors que cet opérateur ne vérifie plus la propriété de monotonie (cf. lemme 6.1 p.148). En effet, le graphe de la décomposition d'une telle contrainte contient un cycle et l'arc-cohérence n'est pas atteinte par deux simples parcours comme dans le cas d'un arbre. Si une contrainte c contient plusieurs occurrences de x, on est donc obligé, dans l'algorithme de propagation, d'insérer dans l'agenda tous les couples* (c', x) *y compris pour* $c' = c$ *(voir p. 147).*

6.9 Évaluation des contraintes avec flottants

Les sections précédentes décrivent des cohérences que l'on peut qualifier de *théoriques*. En effet, elles reposent sur une opération de projection Π_x^c dont le calcul avec une machine n'a pas été précisé. L'algorithme `HC4Revise` ramène la projection d'une contrainte quelconque à celle d'une contrainte primitive, ce qui simplifie déjà conséquemment le problème ; mais il reste à déterminer comment calculer la projection d'une contrainte primitive avec une machine, c'est à dire, avec des nombres flottants. Bien entendu, ce calcul ne peut pas être exact, et notre but est également de définir des cohérences qui tiennent compte de l'approximation liée aux calculs des projections, de telle sorte que ces cohérences soient effectivement applicables dans la réalité.

Le calcul de la projection d'une contrainte primitive (ainsi que la définition de ces nouvelles cohérences) repose sur l'évaluation d'une contrainte primitive. Nous commençons donc par décrire l'évaluation d'une contrainte primitive sur les flottants. La projection sera donnée au §6.10. Rappelons que l'évaluation $f(\mathcal{D})$ peut aussi servir à simplement *tester* l'espace de recherche $\mathcal{D}$ (si 0 n'apparaît pas dans le résultat, $\mathcal{D}$ est infaisable).

L'évaluation d'une expression $f(x)$ (x représente un vecteur de $\mathcal{D}$) a le même sens qu'au chapitre 2 ; notre but est de calculer :

$$\{y \in \mathbb{R} \mid \big(\exists x \in \mathcal{D}\big)\ y = f(x)\},$$

le plus précisément possible. Comme $\mathcal{D}$ est au mieux une boîte, au pire un gruyère, il suffit a priori de calculer $f(\mathcal{D})$ en utilisant les techniques d'évaluation basées sur l'arithmétique d'intervalles. Mais du point de vue algorithmique, nous devons maintenant tenir compte de la structure sous-jacente utilisée pour les calculs (les flottants), du sort des "bornes infinies", ainsi que des problèmes liés au domaine de définition des fonctions ($\mathcal{D}$ peut ne pas être entièrement inclus dans le domaine $\mathcal{D}_f$).

6.9.1 Calcul sur les flottants

Sur un ordinateur, nous ne disposons que d'un ensemble fini de flottants $\mathcal{F}$ pour représenter la droite réelle. Si l'on souhaite être conservatif, un intervalle $\mathbf{x}$ peut donc au mieux être représenté par un intervalle

$$[\underline{\mathbf{x}}^-, \overline{\mathbf{x}}^+],$$

où pour tout réel x, x^- désigne le plus grand flottant à gauche de x, x^+ le plus petit flottant à droite.

Calculons maintenant par exemple la somme de deux intervalles $\mathbf{x}$ et $\mathbf{y}$, avec un ordinateur. Sachant que la somme exacte de deux flottants peut ne pas être un flottant, le résultat de cette somme doit lui-même être arrondi au flottant le plus proche : soit à gauche (pour le calcul de la borne inférieure de $\mathbf{x}+\mathbf{y}$) soit à droite (pour la borne supérieure). Le calcul de $\mathbf{x}+\mathbf{y}$ obtenu avec des flottants est donc au mieux :

$$[(\underline{\mathbf{x}}^- + \underline{\mathbf{y}}^-)^-, (\overline{\mathbf{x}}^+ + \overline{\mathbf{y}}^+)^+].$$

Il y a donc un double décalage qui peut amener à introduire des flottants inutiles. L'implémentation d'une arithmétique qui garantisse que le résultat d'un calcul (fait sur les flottants) est à un flottant près du véritable

résultat de ce calcul (fait sur les réels), ou même la question de savoir quelle est la distance en termes de flottants entre le calcul effectué et le véritable résultat, sont des problématiques très complexes que nous n'aborderons pas : nous considérons dans cette thèse que le calcul sur les flottants de $f(\mathbf{x})$ donne une surestimation de l'intervalle $f(\mathbf{x})$ (réel) dont nous ne savons pas mesurer l'ampleur. Terminons par quelques notations :

Définition 6.18
Soit $x \in \mathbb{R}$. On note :
– *x^+ le plus petit flottant supérieur à x (x lui-même si $x \in \mathcal{F}$),*
– *x^- le plus grand flottant inférieur à x (x lui-même si $x \in \mathcal{F}$).*
Soit $x \in \mathcal{F}$. On note :
– *x^{+1} le successeur (s'il existe) de x, c.a.d., le plus petit flottant strictement supérieur à x,*
– *x^{-1} le prédécesseur (s'il existe) de x, c.a.d., le plus grand flottant strictement inférieur à x.*

6.9.2 Bornes infinies

Pour que les calculs puissent être conservatifs, il est indispensable de définir le plus petit flottant comme étant $-\infty$ et le plus grand comme étant $+\infty$, de telle sorte que $\mathcal{F}$ devienne une approximation de la droite achevée $\mathbb{R} \cup \{-\infty, +\infty\}$. En effet, sinon, quel intervalle de flottants peut représenter $exp(\mathtt{f})$, où `f` est le plus grand flottant ?

Il en découle qu'une évaluation peut retourner un intervalle contenant des bornes infinies. Il apparaît donc la nécessité de redéfinir l'arithmétique d'intervalles en intégrant les bornes infinies (cette arithmétique sera donnée au §6.9.5). Nous pouvons alors profiter des bornes infinies pour définir proprement l'inverse par un intervalle contenant 0, ainsi que la tangente d'un intervalle contenant un multiple de $\pi/2$. Comme cela sera justifié au paragraphe suivant, ces extensions sont utiles même en présence de variables dont les domaines sont bornés, car les domaines de définition de ces deux fonctions contiennent des bornes "ouvertes" (en 0 pour $1/x$, ou en $\pi/2$ pour $\tan(x)$). A l'inverse, les autres fonctions ont leur domaine non seulement fermés dans la droite achevée, mais représentables par des flottants[24].

6.9.3 Gestion du domaine de définition

Faut-il lever une exception lorsqu'il est demandé d'évaluer une fonction f sur un intervalle $\mathbf{x}$ non inclus dans le domaine de définition de f (voire totalement en dehors) ?

La réponse est non, pour au moins deux raisons. Tout d'abord, dans une approche « contraintes », lorsqu'une personne cherche à obtenir les solutions d'un système d'équations, les domaines spécifiés pour les variables ne correspondent qu'à un espace de recherche. Il n'y a alors aucune raison de distinguer un point dans cet espace ne convenant pas car non solution du système d'équations d'un point ne convenant pas car n'entrant pas dans le domaine de définition des fonctions : le fait que $x \notin \mathcal{D}_f$ est bien un cas particulier de « x n'est pas solution de $f(x) = 0$ ». La seconde raison est que l'arrondi utilisé pour représenter $\mathcal{D}_x$ peut introduire des flottants n'entrant pas dans le domaine de définition. Si l'on souhaite effectuer le calcul par intervalles suivant : $\tan(\mathbf{x})$ avec $\mathbf{x} = [1, \pi/2 - 10^{-99}]$, il est fort probable que $\overline{\mathbf{x}}$ ne soit pas un flottant, et que le flottant `f` le plus proche à droite de $\overline{\mathbf{x}}$ soit plus grand que $\pi/2$. Pour être conservatif, nous devons représenter $\mathbf{x}$ par l'intervalle $[\underline{\mathbf{x}}^-, \overline{\mathbf{x}}^+]$ et sortir du domaine de définition. Le problème de « dépassement de domaine » doit être géré au niveau de l'arithmétique sur les flottants : dans un tel cas de figure, elle doit détecter $\overline{\mathbf{x}} > \pi/2$ et en déduire que la borne supérieure de $\tan(\mathbf{x})$ est le flottant $+\infty$. Le même problème se pose pour $1/\mathbf{x}$ avec $\underline{x} \sim 0$.

[24] Les valeurs intervenant pour décrire les domaines de $+, -, \times, exp, ln, cos$, etc... sont : $-\infty, 0, +\infty$ (qui sont bien des flottants) ainsi que -1 et 1 (qui sont aussi en général des flottants) pour les fonctions comme arcsin.

Fort de ces arguments, nous pouvons chercher désormais

$$\{y \in \mathbb{R} \mid \big(\exists x \in \mathcal{D} \cap \mathcal{D}_f\big)\ y = f(x)\}.$$

Si l'intervalle $\mathbf{x}$ est totalement en dehors de $\mathcal{D}_f$, on a

$$\{y \in \mathbb{R} \mid \big(\exists x \in \mathcal{D} \cap \mathcal{D}_f\big)\ y = f(x)\} = \emptyset,$$

ce qui signifie qu'une évaluation peut retourner $\emptyset$ sans que cela soit une erreur. Il y a donc nécessité d'introduire également dans l'arithmétique l'intervalle $\emptyset$.

Nous en venons à la définition d'intervalles (et d'unions d'intervalles) de flottants, et à l'extension d'une fonction aux (unions de) intervalles de flottants. Ces extensions permettent de décrire formellement ce que nous calculons avec une arithmétique d'intervalles basée sur les flottants.

6.9.4 Extension aux intervalles de flottants

Définition 6.19 (Intervalle de flottants) *Soit $\mathcal{F}$ un ensemble de flottants (contenant $-\infty$ et $+\infty$). On définit l'ensemble $\mathbb{I}\mathcal{F}$ des intervalles de flottants comme suit :*

$$\mathbb{I}\mathcal{F} := \Big(\{[f_1, f_2], f_1 \in \mathcal{F}, f_2 \in \mathcal{F}, f_1 \leq f_2\} \backslash \{[-\infty, -\infty], [+\infty, +\infty]\}\Big) \cup \{\emptyset\}.$$

Définition 6.20 (Union d'intervalles de flottants)
Soit $\mathcal{F}$ un ensemble de flottants. On note $\mathbb{U}\mathcal{F}$ l'ensemble des unions finies d'éléments disjoints de $\mathbb{I}\mathcal{F}\backslash\{\emptyset\}$ auquel est ajouté l'élément $\emptyset$. Exemples :

$$\emptyset, \quad [0,1], \quad [-\infty, -1] \cup [1, +\infty], \quad [-\infty, -\infty^{+1}] \cup [+\infty^{-1}, +\infty].$$

Définition 6.21 (Extension à $\mathbb{I}\mathcal{F}$ d'une fonction)
Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction arithmétique. On appelle extension à $\mathbb{I}\mathcal{F}$ de f toute fonction $g : \mathbb{I}\mathcal{F}^n \to \mathbb{U}\mathcal{F}$ telle que

$$\big(\forall \mathbf{x} \in \mathbb{I}\mathcal{F}^n\big) \quad \{y \in \mathbb{R} \mid \big(\exists x \in \mathbf{x} \cap \mathcal{D}_f\big)\ y = f(x)\} \subseteq g(\mathbf{x}).$$

Il est utile de remarquer dans la définition précédente que g est à valeur dans $\mathbb{U}\mathcal{F}$ (des arguments intervalles peuvent produire une union d'intervalles).

Définition 6.22 (Extension à $\mathbb{U}\mathcal{F}$ d'une fonction)
Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction arithmétique. On appelle extension à $\mathbb{U}\mathcal{F}$ de f toute fonction $h : \mathbb{U}\mathcal{F}^n \to \mathbb{U}\mathcal{F}$ telle que

$$\big(\forall \lceil \mathbf{x} \rceil \in \mathbb{U}\mathcal{F}^n\big) \quad \{y \in \mathbb{R} \mid \big(\exists x \in \lceil \mathbf{x} \rceil \cap \mathcal{D}_f\big)\ y = f(x)\} \subseteq h(\lceil \mathbf{x} \rceil).$$

Pour implémenter une extension à $\mathbb{U}\mathcal{F}$, nous pouvons nous ramener à une extension à $\mathbb{I}\mathcal{F}$:

Proposition 6.8 *Soit $f : \mathbb{R}^n \to \mathbb{R}$ une fonction arithmétique, g une extension à $\mathbb{I}\mathcal{F}$ de f. Alors la fonction h telle que*

$$\big(\forall \lceil \mathbf{x} \rceil \in \mathbb{U}\mathcal{F}^n\big) \quad h(\lceil \mathbf{x} \rceil) = \bigcup_{\mathbf{x}_1 \in \lceil \mathbf{x}_1 \rceil, \ldots, \mathbf{x}_n \in \lceil \mathbf{x}_n \rceil} g(\mathbf{x}_1, \ldots, \mathbf{x}_n)$$

est une extension à $\mathbb{U}\mathcal{F}$ de f.

6.9.5 Évaluation des contraintes primitives

Nous donnons ci-dessous une implémentation possible d'une extension à $\mathbb{I}\mathcal{F}$ des fonctions primitives. Cette extension sera notée $f^\diamond$. En appliquant le parcours "en remontée" (c.a.d. la phase d'évaluation) de l'algorithme `HC4Revise` (cf. §6.8), nous en déduisons une extension à $\mathbb{I}\mathcal{F}$ de n'importe quelle fonction arithmétique.

calcul de $\mathbf{y} = \mathbf{x} +^\diamond \mathbf{z}$

si ($\mathbf{x} = \emptyset$ ou $\mathbf{z} = \emptyset$)	$\mathbf{y} \leftarrow \emptyset$
sinon	
si ($\underline{\mathbf{x}} = -\infty$ ou $\underline{\mathbf{z}} = -\infty$)	$\underline{\mathbf{y}} \leftarrow -\infty$
sinon	$\underline{\mathbf{y}} \leftarrow (\underline{\mathbf{x}} + \underline{\mathbf{z}})^-$
si ($\overline{\mathbf{x}} = +\infty$ ou $\overline{\mathbf{z}} = +\infty$)	$\overline{\mathbf{y}} \leftarrow +\infty$
sinon	$\overline{\mathbf{y}} \leftarrow (\overline{\mathbf{x}} + \overline{\mathbf{z}})^+$

La soustraction est similaire à l'addition.

calcul de $\mathbf{y} = \mathbf{x} \times^\diamond \mathbf{z}$

si ($\mathbf{x} = \emptyset$ ou $\mathbf{z} = \emptyset$)	$\mathbf{y} \leftarrow \emptyset$
sinon si (($\underline{\mathbf{x}} = 0$ et $\overline{\mathbf{x}} = 0$) ou ($\underline{\mathbf{z}} = 0$ et $\overline{\mathbf{z}} = 0$))	$\mathbf{y} \leftarrow [0, 0]$
sinon si (($\underline{\mathbf{x}} < 0$ et $\overline{\mathbf{x}} > 0$) et ($\underline{\mathbf{z}} = -\infty$ ou $\overline{\mathbf{z}} = +\infty$))	$\mathbf{y} \leftarrow [-\infty, +\infty]$
sinon si (($\underline{\mathbf{z}} < 0$ et $\overline{\mathbf{z}} > 0$) et ($\underline{\mathbf{x}} = -\infty$ ou $\overline{\mathbf{x}} = +\infty$))	$\mathbf{y} \leftarrow [-\infty, +\infty]$
sinon si (($\underline{\mathbf{x}} = -\infty$ et $\overline{\mathbf{z}} = 0$) ou ($\overline{\mathbf{z}} = +\infty$ et $\underline{\mathbf{x}} = 0$))	
si ($\overline{\mathbf{x}} \leq 0$ ou $\underline{\mathbf{z}} \geq 0$)	$\mathbf{y} \leftarrow [0, +\infty]$
sinon	$\mathbf{y} \leftarrow [(\overline{\mathbf{x}} \times \underline{\mathbf{z}})^-, +\infty]$
sinon si (($\underline{\mathbf{x}} = -\infty$ et $\underline{\mathbf{z}} = 0$) ou ($\underline{\mathbf{z}} = -\infty$ et $\underline{\mathbf{x}} = 0$))	
si ($\overline{\mathbf{x}} \leq 0$ ou $\overline{\mathbf{z}} \leq 0$)	$\mathbf{y} \leftarrow [-\infty, 0]$
sinon	$\mathbf{y} \leftarrow [-\infty, (\overline{\mathbf{x}} \times \overline{\mathbf{z}})^+]$
sinon si (($\underline{\mathbf{z}} = -\infty$ et $\overline{\mathbf{x}} = 0$) ou ($\overline{\mathbf{x}} = +\infty$ et $\underline{\mathbf{z}} = 0$))	
si ($\overline{\mathbf{z}} \leq 0$ ou $\underline{\mathbf{x}} \geq 0$)	$\mathbf{y} \leftarrow [0, +\infty]$
sinon	$\mathbf{y} \leftarrow [(\overline{\mathbf{z}} \times \underline{\mathbf{x}})^-, +\infty]$
sinon si (($\overline{\mathbf{x}} = +\infty$ et $\overline{\mathbf{z}} = 0$) ou ($\overline{\mathbf{z}} = +\infty$ et $\overline{\mathbf{x}} = 0$))	
si ($\underline{\mathbf{x}} \geq 0$ ou $\underline{\mathbf{z}} \geq 0$)	$\mathbf{y} \leftarrow [-\infty, 0]$
sinon	$\mathbf{y} \leftarrow [-\infty, (\underline{\mathbf{x}} \times \underline{\mathbf{z}})^+]$
sinon	$\mathbf{y} \leftarrow \mathbf{x} \times \mathbf{y}$

calcul de $\mathbf{y} = \mathbf{x} /^\diamond \mathbf{z}$

si ($\mathbf{x} = \emptyset$ ou $\mathbf{z} = \emptyset$)	$\mathbf{y} \leftarrow \emptyset$
sinon si ($\underline{\mathbf{z}} = 0$ et $\overline{\mathbf{z}} = 0$)	$\mathbf{y} \leftarrow \emptyset$
sinon si ($\underline{\mathbf{x}} = 0$ et $\overline{\mathbf{x}} = 0$)	
si ($\underline{\mathbf{z}} \leq 0$ et $\overline{\mathbf{z}} \geq 0$)	$\mathbf{y} \leftarrow [-\infty, +\infty]$
sinon	$\mathbf{y} \leftarrow [0, 0]$
sinon si ($\underline{\mathbf{z}} > 0$ ou $\overline{\mathbf{z}} < 0$)	$\mathbf{y} \leftarrow \mathbf{x}/\mathbf{z}$
sinon si ($\overline{\mathbf{x}} \leq 0$ et $\overline{\mathbf{z}} = 0$)	$\mathbf{y} \leftarrow [(\overline{\mathbf{x}}/\underline{\mathbf{z}})^-, +\infty]$
sinon si ($\overline{\mathbf{x}} \leq 0$ et $\underline{\mathbf{z}} < 0$ et $\overline{\mathbf{z}} > 0$)	$\mathbf{y} \leftarrow [-\infty, (\overline{\mathbf{x}}/\overline{\mathbf{z}})^+] \cup [(\overline{\mathbf{x}}/\underline{\mathbf{z}})^-, +\infty]$
sinon si ($\overline{\mathbf{x}} \leq 0$ et $\underline{\mathbf{z}} = 0$)	$\mathbf{y} \leftarrow [-\infty, (\overline{\mathbf{x}}/\overline{\mathbf{z}})^+]$
sinon si ($\underline{\mathbf{x}} \geq 0$ et $\overline{\mathbf{z}} = 0$)	$\mathbf{y} \leftarrow [-\infty, (\underline{\mathbf{x}}/\underline{\mathbf{z}})^+]$
sinon si ($\underline{\mathbf{x}} \geq 0$ et $\underline{\mathbf{z}} < 0$ et $\overline{\mathbf{z}} > 0$)	$\mathbf{y} \leftarrow [-\infty, (\underline{\mathbf{x}}/\underline{\mathbf{z}})^+] \cup [(\underline{\mathbf{x}}/\overline{\mathbf{z}})^-, +\infty]$
sinon si ($\underline{\mathbf{x}} \geq 0$ et $\underline{\mathbf{z}} = 0$)	$\mathbf{y} \leftarrow [(\underline{\mathbf{x}}/\overline{\mathbf{z}})^-, +\infty]$
sinon	$\mathbf{y} \leftarrow [-\infty, +\infty]$ (on a $\underline{\mathbf{x}} < 0 < \overline{\mathbf{x}}$ et $\underline{\mathbf{z}} \leq 0 \leq \overline{\mathbf{z}}$)

Nous donnons quelques exemples de fonctions élémentaires :

calcul de $\mathbf{y} = \exp^{\diamond}(\mathbf{x})$

si $(\mathbf{x} = \emptyset)$	$\mathbf{y} \leftarrow \emptyset$
si $(\underline{\mathbf{x}} = -\infty)$	$\underline{\mathbf{y}} \leftarrow 0$
sinon	$\underline{\mathbf{y}} \leftarrow \exp(\underline{\mathbf{x}})^{-}$
si $(\overline{\mathbf{x}} = +\infty)$	$\overline{\mathbf{y}} \leftarrow +\infty$
sinon	$\overline{\mathbf{y}} \leftarrow \exp(\overline{\mathbf{x}})^{+}$

calcul de $\mathbf{y} = \log^{\diamond}(\mathbf{x})$

si $(\overline{\mathbf{x}} < 0)$	$\mathbf{y} \leftarrow \emptyset$
sinon	
si $(\underline{\mathbf{x}} < 0)$	$\underline{\mathbf{y}} \leftarrow -\infty$
sinon	$\underline{\mathbf{y}} \leftarrow \log(\underline{\mathbf{x}})^{-}$
si $(\overline{\mathbf{x}} = +\infty)$	$\overline{\mathbf{y}} \leftarrow +\infty$
sinon	$\overline{\mathbf{y}} \leftarrow \log(\overline{\mathbf{x}})^{+}$

calcul de $\mathbf{y} = \tan^{\diamond}(\mathbf{x})$

si $(\mathbf{x} = \emptyset)$	$\mathbf{y} \leftarrow \emptyset$
si $(\underline{\mathbf{x}} = -\infty$ ou $\overline{\mathbf{x}} = +\infty)$	$\mathbf{y} \leftarrow [-\infty, +\infty]$
$\mathbf{p} \leftarrow (\mathbf{x} + [\text{sign}\,(\underline{\mathbf{x}}) * (\pi/2)^{-}, \text{sign}\,(\overline{\mathbf{x}}) * (\pi/2)^{+}]/[\pi^{-}, \pi^{+}]$	(calcul du nombre de périodes traversées)
si $\overline{\mathbf{p}} - \underline{\mathbf{p}} \geq 2$	$\mathbf{y} \leftarrow [-\infty, +\infty]$
sinon si $\overline{\mathbf{p}} - \underline{\mathbf{p}} = 1$	$\mathbf{y} \leftarrow [-\infty, \tan(\overline{\mathbf{x}})^{+}] \cup [\tan(\underline{\mathbf{x}})^{-}, +\infty]$
sinon	$\mathbf{y} \leftarrow [\tan(\underline{\mathbf{x}})^{-}, \tan(\overline{\mathbf{x}})^{+}]$

Remarque 6.3 *La borne ouverte* 0 *pour la division par* $\mathbf{x}/\mathbf{z}$ *oblige à traiter le cas* $\mathbf{z} = [0, 0]$ *(le résultat étant* $\emptyset$*). Le problème ne se pose pas avec les bornes ouvertes de* tan*, car* $\pi/2$ *n'est pas représentable par un flottant !*

6.10 Projections des contraintes avec flottants

Dans le prolongement de la section précédente, nous définissons maintenant une manière de calculer une approximation extérieure de la projection d'une contrainte primitive. Nous supposons disposer d'une extension à $\mathbb{U}\mathcal{F}$ de toute fonction f, notée $f^{\diamond}$.

La notion de projection est basée sur la notion de *satisfaction* : la définition 6.4 p.146 fait intervenir tous les n-uplets des domaines. Une énumération des réels étant impossible, cette définition continue d'avoir un sens pour un modèle théorique (cf. §6.3.1), mais ne correspond plus à quelque chose de calculable. Il faut remplacer la projection par une notion adaptée à la structure de flottants. Nous verrons alors qu'il est possible (sous certaines hypothèses) de caractériser l'approximation de l'arc-cohérence ou de la 2B-cohérence obtenue avec cette nouvelle projection. Néanmoins, ceci n'est valable qu'en présence de contraintes primitives, car une caractérisation dans le cas général nécessiterait des hypothèses beaucoup plus fortes (et non réalistes) sur la précision des calculs.

6.10.1 $\mathcal{F}$-projection

Informellement, la satisfaction d'une contrainte est redéfinie ainsi : nous fabriquons artificiellement un CSP fini en assimilant l'ensemble des réels entre deux flottants consécutifs à une "valeur", appelée *tranche*. Nous retrouvons exactement la notion de tranche définie au §6.5 p.155, où la subdivision σ n'était rien d'autre que des flottants "virtuels". Le pont évoqué entre la σ-cohérence et la $\mathcal{F}$-cohérence se fait en prenant pour subdivision de la σ-cohérence un sous-ensemble des flottants. Le nombre de flottants étant fini dans un domaine, il ne reste plus alors qu'à définir la satisfaction d'une contrainte avec des tranches.

Définition 6.23 (Tranche) *On appelle* **tranche** *tout intervalle* $[x, x^{+1}]$ *formé de deux flottants consécutifs. On note cette tranche* $\sigma(x)$*.*

Définition 6.24 ($\mathcal{F}$-support) *Soient* $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ *un CSP-gruyère et* $c \in \mathcal{C}$ *de la forme* $f(x, y_1, \ldots, y_k) = 0$*. Notons* $\lceil \mathbf{x} \rceil$ *(resp.*$\lceil \mathbf{y}_i \rceil$*) le domaine* $\mathcal{D}_x$ *(resp* $\mathcal{D}_{y_i}$*, pour tout* $i \in [1..k]$*).*

Pour toute tranche $\sigma(x)$ *de* $\lceil \mathbf{x} \rceil$*, on appelle* **$\mathcal{F}$-support** *de* $\sigma(x)$ *(pour* c*) toute boîte à* k *dimensions* $\sigma(y_1) \times \ldots \times \sigma(y_k) \in \lceil \mathbf{y}_1 \rceil \times \ldots \times \lceil \mathbf{y}_k \rceil$ *telle que* $0 \in f^{\diamond}(\sigma(x) \times \sigma(y_1) \times \ldots \times \sigma(y_k))$*.*

Remarque 6.4 *Dans la définition précédente, on interdit une "micro-boîte" $\sigma(x) \times \sigma(y_1) \times \ldots \times \sigma(y_k)$ d'avoir sur une dimension un intervalle dégénéré. Cette interdiction permet uniquement de simplifier les définitions suivantes, mais les intervalles dégénérés ne constituent en fait ni un obstacle en théorie, ni en pratique. Quoi qu'il en soit, cette interdiction n'a que deux conséquences : le gruyère initial $\mathcal{D}$ ne doit pas contenir de domaine $\mathcal{D}_x$ réduit à un seul point (ce qui n'est pas très gênant, une variable x avec un tel domaine peut être remplacée par une constante), et une contrainte comme $x = 0$ réduit $\mathcal{D}_x$ à l'intervalle $[0^{-1}, 0^{+1}]$ (au lieu de $[0, 0]$).*

Définition 6.25 ($\mathcal{F}$-projection) *Soient $\mathcal{C}$ un ensemble de contraintes et $\mathcal{X} = \{x_1, \ldots, x_n\}$ un ensemble de variables. Soient $f(x_1, \ldots, x_n) \in \mathcal{C}$ et $x_i \in \mathcal{X}$.*

On appelle **$\mathcal{F}$-projection** *de c sur x_i, et on note $\Pi^{c\diamond}_{x_i}$ la fonction qui à un gruyère $\mathcal{D} = \mathcal{D}_{x_1} \times \ldots \times \mathcal{D}_{x_n}$ retourne le gruyère $\mathcal{D}'$ vérifiant*
- $\big(\forall j \neq i\big) \quad D'_{x_j} = \mathcal{D}_{x_j}$,
- *D'_{x_i} est l'union des tranches $\sigma(x_i)$ de $\mathcal{D}_{x_i}$ ayant un $\mathcal{F}$-support pour c.*

On en déduit une manière naïve de calculer la $\mathcal{F}$-projection d'une contrainte, qui consiste à découper en tranches les domaines des variables impliquées dans la contrainte de la même manière que l'on énumère des valeurs en domaine fini. Du fait du nombre gigantesque de flottants que peut contenir un intervalle, cette méthode est disqualifiée d'office. Dans le cas d'une contrainte binaire $f(x, y) = 0$, elle s'écrit :

$D'_x \leftarrow \emptyset$
`pour tout` $[x, x^{+1}] \subseteq \mathcal{D}_x$
 `pour tout` $[y, y^{+1}] \subseteq \mathcal{D}_y$
  `si` $0 \in f^{\diamond}([x, x^{+1}] \times [y, y^{+1}])$
   $D'_x \leftarrow D'_x \cup [x, x^{+1}]$
  `fin si`
 `fin pour`
`fin pour`
retourner D'_x

Tout comme la projection permet de définir l'arc-cohérence et la 2B-cohérence, la $\mathcal{F}$-projection permet de définir les cohérences suivantes :

Définition 6.26 ($\mathcal{F}$-Arc-cohérence) *Soit $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP avec $\forall x \in \mathcal{X}$, $\mathcal{D}_x \in \mathbb{U}\mathcal{F}$. On dit que l'espace de recherche $\mathcal{D}$ est* **$\mathcal{F}$-arc-cohérent** *si*

$$\forall (c, x) \in \mathcal{C} \times \mathcal{X} \quad \Pi^{c\diamond}_x(\mathcal{D}) = \mathcal{D}.$$

Définition 6.27 ($\mathcal{F}$-2B-cohérence) *Soit $(\mathcal{C}, \mathcal{X}, \mathcal{D})$ un CSP avec $\forall x \in \mathcal{X}$, $\mathcal{D}_x \in \mathbb{I}\mathcal{F}$. On dit que l'espace de recherche $\mathcal{D}$ est* **$\mathcal{F}$-2B-cohérent** *si*

$$\forall (c, x) \in \mathcal{C} \times \mathcal{X} \quad \Box\, \Pi^{c\diamond}_x(\mathcal{D}) = \mathcal{D}.$$

On montre alors facilement que l'algorithme `AC3` (cf. page 147) permet d'obtenir ces cohérences, en remplaçant la ligne :
calculer $D'_x \leftarrow \Pi^c_x(\mathcal{D})$;
dans le cas de la $\mathcal{F}$-arc-cohérence par :
calculer $D'_x \leftarrow \Pi^{c\diamond}_x(\mathcal{D})$;
et dans le cas de la $\mathcal{F}$-2B-cohérence par :
calculer $D'_x \leftarrow \Box\, \Pi^{c\diamond}_x(\mathcal{D})$.

Les notions de clôture (par $\mathcal{F}$-arc-cohérence ou $\mathcal{F}$-2B-cohérence) se définissent comme auparavant, et s'obtiennent grâce à ces différentes versions d'`AC3`. Il n'y a plus de souci de terminaison du fait que la projection ne peut enlever moins qu'un flottant dans un domaine (c.a.d., une tranche). Ajoutons que la σ-$\mathcal{F}$-arc-cohérence et la σ-$\mathcal{F}$-2B-cohérence s'obtienne alors facilement, en "élargissant" simplement le résultat de toute $\mathcal{F}$-projection aux tranches de σ.

Nous allons maintenant décrire un algorithme implémentant l'opérateur $\Pi^{c\diamond}$, plus efficace que le découpage en tranches décrit ci-dessus. Tout comme pour l'évaluation, nous nous limitons aux contraintes primitives puisque `HC4Revise` permet de traiter le cas général, moyennant une approximation, à partir de l'évaluation et de la projection des contraintes primitives (cf. §6.8).

Nous allons voir qu'en se limitant à des contraintes primitives, l'algorithme calcule une projection exacte (c.a.d. $\Pi_x^{c\diamond}$) si on émet quelques hypothèses. Généraliser ces hypothèses de telle sorte que `HC4Revise` calcule également une projection exacte n'est pas réaliste : cela supposerait qu'aucune erreur ne se propage en composant les fonctions apparaissant dans l'expression de la fonction[25].

6.10.2 Hypothèse

Nous donnons dans les deux paragraphes suivants un moyen de calculer $\Pi_x^{c\diamond}(\mathcal{D})$ lorsque c est une contrainte primitive. La validité de ce calcul repose sur une hypothèse : *toute paire de tranches $\mathcal{F}$-supports entre elles pour une contrainte $y = e(x)$ contient une paire de valeurs supports entre elles pour cette contrainte*, ce qui est bien sûr faux dans le cas général. Cette hypothèse traduit néanmoins l'idée que si on ne peut prouver l'infaisabilité d'un intervalle entre deux flottants consécutifs, alors il a effectivement des chances de contenir une solution. L'usage d'une arithmétique des flottants de précision infinie peut rendre cette hypothèse crédible en pratique : comme une fonction est en général monotone sur une tranche, l'évaluation de la tranche peut se ramener à un calcul sur chaque borne où la précision infinie est appliquée.

Hypothèse

Pour toute fonction élémentaire e, et pour toutes tranches $\sigma(x)$ et $\sigma(y)$,

$$0 \in \sigma(y) -^{\diamond} e^{\diamond}(\sigma(x)) \quad \text{ssi} \quad \big(\exists y_0 \in \sigma(y)\big)\big(\exists x_0 \in \sigma(x)\big)\ y_0 = e(x_0).$$

De même, pour toute opération binaire $\star$ et toutes tranches $\sigma(x)$, $\sigma(y)$ et $\sigma(z)$,

$$0 \in \sigma(y) -^{\diamond} (\sigma(x) \star^{\diamond} \sigma(z)) \quad \text{ssi} \quad \big(\exists y_0 \in \sigma(y)\big)\big(\exists x_0 \in \sigma(x)\big)\ y_0 = x_0 \star z_0.$$

La partie supposée vraie (et en général fausse) dans cette hypothèse est bien sûr la condition nécessaire.

Dans ce qui suit, on note c la contrainte, x la variable et $\mathcal{D}^{\diamond} := \Pi_x^{c\diamond}(\mathcal{D})$. Notre but est donc de calculer $\mathcal{D}_x^{\diamond}$. Nous notons $\mathbf{x}$ le domaine de x, en supposant qu'il s'agit d'un intervalle de $\mathbb{I}\mathcal{F}$. Nous pouvons en effet appliquer la proposition suivante (analogue à la proposition 6.8) pour le calcul d'une projection sur une union de $\mathbb{U}\mathcal{F}$ à partir des projections sur $\mathbb{I}\mathcal{F}$:

Proposition 6.9 *Soit c une contrainte.*

$$\big(\forall \lceil \mathbf{x} \rceil \in \mathbb{U}\mathcal{F}^n\big) \quad \Pi_x^{c\diamond}(\lceil \mathbf{x} \rceil) = \bigcup_{\mathbf{x}_1 \in \lceil \mathbf{x}_1 \rceil, \ldots, \mathbf{x}_n \in \lceil \mathbf{x}_n \rceil} \Pi_x^{c\diamond}(\mathbf{x}_1, \ldots, \mathbf{x}_n).$$

[25] Rump donne un exemple [Rum88] où le simple fait d'évaluer une fonction en un point conduit à des erreurs d'arrondi produisant un résultat aberrant et ce, même avec une précision très grande (128 bits).

6.10.3 Projection des fonctions élémentaires

Nous nous intéressons ici au cas d'une contrainte $c : y = e(x)$, où e est une fonction élémentaire. Nous montrons comment projeter sur x (la projection de e sur y est un cas plus simple, que nous évoquerons ensuite).

Projection théorique

Mettons de côté un instant les domaines $\mathcal{D}_x$ et $\mathcal{D}_y$ (d'où l'attribut "théorique"). Les fonctions élémentaires sont toutes strictement monotones par morceaux sur leur domaine de définition. Elles sont donc inversibles par morceaux, c'est à dire que pour tout fonction élémentaire e il existe un ensemble de fonctions ϕ_1, ϕ_2,... éventuellement infini avec pour tout i, $\phi_i : e(\mathcal{D}_e) \to \mathcal{D}_e$, et telles que

$$e(x) = y \iff x = \phi_1(y) \text{ ou } x = \phi_2(y) \text{ ou } \ldots \tag{6.6}$$

Évidemment, au delà de leur existence, les fonctions ϕ_i sont identifiées grâce à notre connaissance de e (on sait parmi les fonctions élémentaires lesquelles sont inverses l'une de l'autre), et de son éventuelle périodicité.

Exemple 6.12 *La projection théorique de $y = cos(x)$ est :*

$$y = cos(x) \iff x = arccos(y) \text{ ou } x = -arccos(y) \text{ ou } x = arccos(y) + 2\pi \text{ ou } \ldots.$$

Nous avons maintenant les outils pour pouvoir calculer une $\mathcal{F}$-projection (cf. définition 6.25).

Proposition 6.10 *Soit e une fonction élémentaire, et c la contrainte $y = e(x)$. Notons $\mathbf{x} = \mathcal{D}_x$, $\mathbf{y} = \mathcal{D}_y$ et supposons $\mathbf{x} \in \mathbb{I}\mathcal{F}$ et $\mathbf{y} \in \mathbb{I}\mathcal{F}$. Alors,*

$$\mathcal{D}_x^\diamond = (\phi_1^\diamond(\mathbf{y})^{\pm 1} \cap \mathbf{x}) \cup (\phi_2^\diamond(\mathbf{y})^{\pm 1} \cap \mathbf{x}) \cup \ldots \tag{6.7}$$

où on note $[f_1, f_2]^{\pm 1}$ l'intervalle $[f_1, f_2]$ de $\mathbb{I}\mathcal{F}$ "agrandi" d'un flottant, c.a.d.,

$$[f_1, f_2]^{\pm 1} := [{f_1}^{-1}, {f_2}^{+1}].$$

La proposition utilise le lemme suivant :

Lemme 6.6 *Quels que soient $\mathbf{x} \in \mathbb{I}\mathcal{F}$ et $\mathbf{y} \in \mathbb{I}\mathcal{F}$,* $0 \in \mathbf{x} -^\diamond \mathbf{y} \iff \mathbf{x} \cap \mathbf{y} \neq \emptyset$.

Preuve. Montrons le sens ($\Longrightarrow$). Nous ne pouvons prouver l'implication sans "descendre" dans la structure des flottants. Tout d'abord $\mathbf{x} -^\diamond \mathbf{y}$ retourne l'intervalle de flottant le plus précis contenant $range(x-y, x \in \mathbf{x}, y \in \mathbf{y})$. Si 0 appartient à cet intervalle, il existe deux flottants $x \in \mathbf{x}$ et $y \in \mathbf{y}$ pour lesquels on a soit $0 = (x-y)^-$ soit $0 = (x-y)^+$. Dans les deux cas, $0 = |x-y|^-$. Tout d'abord, si $|x| = |y|$, clairement $x = y$ et on a bien $\mathbf{x} \cap \mathbf{y} \neq \emptyset$. Supposons $|y| < |x|$. Les flottants sont distribués de telle sorte que pour deux flottants f_1 et f_2, $|f_1| < |f_2| \Longrightarrow |f_1| < |f_2 - f_1|$ (plus on s'éloigne de 0, plus les flottants sont espacés). Donc

$$\begin{array}{lll}
|y| < |x| & \Longrightarrow |y| < |x-y| & \text{d'après la distribution des flottants,} \\
& \Longrightarrow |y| < 0^{+1} & \text{car } 0 = |x-y|^-, \\
& \Longrightarrow y = 0 & \text{car } y \text{ est un flottant,} \\
& \Longrightarrow |x-y| = |x| & \\
& \Longrightarrow 0 < |x| < 0^{+1} & \text{car } 0 = |x-y|^- \text{ et } x \neq y,
\end{array}$$

ce qui est impossible. Le cas $|y| \geq |x|$ se montre de la même façon. Le sens ($\Longleftarrow$) est évident. □

Preuve de la proposition. Tout d'abord,

$$x \in \mathcal{F} \cap \mathcal{D}_x^\diamond \iff [x^{-1}, x] \subseteq \mathcal{D}_x^\diamond \text{ ou } [x, x^{+1}] \subseteq \mathcal{D}_x^\diamond.$$

Nous supposons $\sigma(x) = [x, x^{+1}] \in \mathcal{D}_x^\diamond$ (introduire le cas $[x^{-1}, x] \subseteq \mathcal{D}_x^\diamond$ n'apporte pas de difficulté mais alourdit le texte). On a donc $x \in \mathcal{F} \cap \mathcal{D}_x^\diamond$ ssi

$$x \in \mathbf{x}, \tag{6.8}$$

$$\big(\exists \sigma(y) \subseteq \mathbf{y}\big) \quad \sigma(y) \text{ est un } \mathcal{F}\text{-support de } \sigma(x) \text{ pour } y = e(x). \tag{6.9}$$

D'après l'hypothèse 1, on a :

$$(6.9) \iff \big(\exists \sigma(y) \subseteq \mathbf{y}\big)\big(\exists y_0 \in \sigma(y)\big)\big(\exists x_0 \in \sigma(x)\big) \quad y_0 = e(x_0).$$

En appliquant la projection théorique ci-dessus, on obtient

$$(6.9) \iff \big(\exists \sigma(y) \subseteq \mathbf{y}\big)\big(\exists y_0 \in \sigma(y)\big)\big(\exists x_0 \in \sigma(x)\big)\big(\exists i \geq 0\big) \quad x_0 = \phi_i(x_0).$$

De nouveau, d'après l'hypothèse 1,

$$(6.9) \iff \big(\exists \sigma(y) \subseteq \mathbf{y}\big)\big(\exists i \geq 0\big) \quad \sigma(y) \text{ est un } \mathcal{F}\text{-support de } \sigma(x) \text{ pour } x = \phi_i(y).$$

D'après la définition de $\mathcal{F}$-support,

$$(6.9) \iff \big(\exists \sigma(y) \subseteq \mathbf{y}\big)\big(\exists i \geq 0\big) \quad 0 \in \sigma(x) -^\diamond \phi_i^\diamond(\sigma(y)).$$

En utilisant le lemme 6.6,

$$(6.9) \iff \big(\exists \sigma(y) \subseteq \mathbf{y}\big)\big(\exists i \geq 0\big) \quad \sigma(x) \cap \phi_i^\diamond(\sigma(y)) \neq \emptyset.$$

On en déduit,

$$\begin{array}{lll} (6.9) & \iff \big(\exists \sigma(y) \subseteq \mathbf{y}\big)\big(\exists i \geq 0\big) & x \in \phi_i^\diamond(\sigma(y))^{\pm 1} \\ & \iff \big(\exists i \geq 0\big) & x \in \phi_i^\diamond(\mathbf{y})^{\pm 1} \\ & \iff & x \in \Big(\phi_1^\diamond(\mathbf{y})^{\pm 1} \cup \phi_1^\diamond(\mathbf{y})^{\pm 1} \cup \ldots\Big). \end{array}$$

Finalement, $x \in \mathcal{D}_x^\diamond$ ssi x vérifie à la fois (6.8) et (6.9), donc ssi $x \in (\phi_1^\diamond(\mathbf{y})^{\pm 1} \cap \mathbf{x}) \cup (\phi_2^\diamond(\mathbf{y})^{\pm 1} \cap \mathbf{x}) \cup \ldots$, d'où le résultat. □

Exemple 6.13 *La projection théorique de $x^2 = y$ est :*

$$x^2 = y \iff x = \sqrt{y} \text{ ou } x = -\sqrt{y}.$$

Si $\mathbf{y} = [-2, -1]$, l'évaluation $sqrt^\diamond(y)$ retourne $\emptyset$, donc la projection sur x est $\emptyset$.

Si $\mathbf{y} = [-2, 1]$, l'évaluation $sqrt^\diamond(y)$ "ramène" le domaine de $\mathbf{y}$ à $[-2, 1] \cap \mathcal{D}_{sqrt} = [0, 1]$ et retourne $[0, 1]$. La projection sur x est $[-1^{-1}, 0^{+1}] \cup [0^{-1}, 1^{+1}] = [-1^{-1}, 1^{+1}]$.

Si $\mathbf{y} = [1, 4]$, l'évaluation $sqrt^\diamond(y)$ retourne $[1, 2]$. La projection sur x est $[-2^{-1}, -1^{+1}] \cup [1^{-1}, 2^{+1}]$.

Sélection des intervalles

Pour calculer (6.7), il reste le problème qu'il peut y avoir un nombre infini de termes $\phi_i^\diamond(\mathbf{y}) \cap \mathbf{x}$. Mais dans ce cas, il s'agit forcément d'une fonction périodique et le domaine de $\mathbf{x}$ permet alors de préselectionner un nombre fini de périodes ou (demi-périodes) admissibles, c'est à dire contenant les intervalles $\phi_i^\diamond(\mathbf{y})^{\pm 1}$ dont l'intersection avec $\mathbf{x}$ est non vide.

Exemple 6.14 *Soit la contrainte $y = cos(x)$ avec $\mathbf{x} = [-4, 5]$, $\mathbf{y} = [0, 0.5]$. La projection de c sur $\mathbf{x}$ est restreinte aux demi-périodes $[-\pi, 0]$, $[0, \pi]$ et $[\pi, 2\pi]$ puisque $[-4, 5] \subseteq [-\pi, 2\pi]$ (cf.figure 6.10).*

La formule de l'exemple 6.12 mène donc au calcul des 3 intervalles suivants :

$$\mathbf{x}_1 = \arccos^{\diamond}([0, 0.5]) = [\frac{\pi}{6}^{-}, \frac{\pi}{2}^{+}] \subseteq [0.52, 1.58],$$

$$\mathbf{x}_2 = -\arccos^{\diamond}([0, 0.5]) = [-\frac{\pi}{2}^{-}, -\frac{\pi}{6}^{+}] \subseteq [-1.58, -0.52],$$

$$\mathbf{x}_3 = 2\pi - \arccos^{\diamond}([0, 0.5]) = [\frac{3\pi}{2}^{-}, \frac{11\pi}{6}^{+}] \subseteq [4.71, 7.58].$$

La projection sur $\mathbf{x}$ obtenue est donc l'union $(\mathbf{x}_1^{\pm 1} \cap \mathbf{x}) \cup (\mathbf{x}_2^{\pm 1} \cap \mathbf{x}) \cup (\mathbf{x}_3^{\pm 1} \cap \mathbf{x}) \sim [-1.58, -0.52] \cup [0.52, 1.58] \cup [4.71, 5]$. Si on ne gère que des intervalles, la meilleure approximation possible de cette union est son enveloppe, c.a.d., l'intervalle $\mathbf{x}' = [-1.58, 5]$.

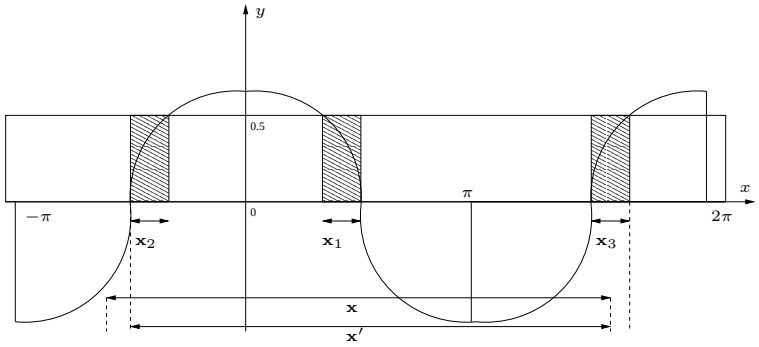


FIG. *6.10: Projection de $y = cos(x)$ sur x*

La projection de $y = e(x)$ sur y est un cas particulier de l'étude qui précède (la fonction implicite est f elle-même). On obtient :

$$\mathcal{D}'_x = e^{\diamond}(\mathbf{x})^{\pm 1} \cap \mathbf{y}.$$

6.10.4 Projection des opérations binaires

Nous nous intéressons ici au cas d'une contrainte $c : y = x \star z$, avec $\star \in \{+, -, \times, /\}$. Nous montrons comment projeter sur x (la projection sur y étant un corollaire, comme dans le cas des fonctions élémentaires). La projection théorique coule de source :

$$\begin{array}{lcl} y = x + z & \Longleftrightarrow & x = y - z \\ y = x - z & \Longleftrightarrow & x = y + z \\ y = x \times z & \Longleftrightarrow & x = y/z \\ y = x/z & \Longleftrightarrow & x = y \times z \end{array}$$

Il n'y a pas de problème de sélection des intervalles, le nombre d'intervalles produits étant fini (au plus deux intervalles pour la division). On obtient finalement l'équivalent de la proposition 6.11 :

Proposition 6.11 *Soit $\oplus$ une opérateur binaire parmi $\{+, -, \times, /\}$. On note $\ominus$ l'opérateur inverse (l'inverse de $+$ (resp. $\times$) est $-$ (resp. /) et réciproquement). Notons $\mathbf{x} = \mathcal{D}_x$, $\mathbf{y} = \mathcal{D}_y$, $\mathbf{z} = \mathcal{D}_z$ et supposons $\mathbf{x} \in \mathbb{I}\mathcal{F}$, $\mathbf{y} \in \mathbb{I}\mathcal{F}$ et $\mathbf{z} \in \mathbb{I}\mathcal{F}$. Alors,*

$$\mathcal{D}_x^{\diamond} = (\mathbf{y} \ominus^{\diamond} \mathbf{z})^{\pm 1} \cap \mathbf{x}.$$

6.11 Conclusion

Nous avons présenté dans ce chapitre les principaux filtrages issus de la programmation par contraintes pour les systèmes d'équations de variables réelles. La projection et l'évaluation sont les deux opérations « atomiques » impliquant l'arithmétique des intervalles. Chaque filtrage se distingue par la façon dont ces opérations sont combinées avec des techniques de gestion des domaines (intersection et union, calcul d'enveloppe, découpage, rognage, etc...).

Ces filtrages constituent une alternative particulièrement intéressante pour des problèmes dont la structure ne permet pas d'appliquer les outils d'analyse par intervalles présentés au chapitre 2. Il est montré, par exemple, dans [Jau06] un problème concret comportant 300000 variables que la 2B-cohérence permet d'attaquer efficacement.

En programmation par contraintes, un soin particulier est toujours apporté à identifier formellement les propriétés que l'on obtient en appliquant un filtrage. L'intérêt est de pouvoir comparer les propriétés entre elles et d'établir une hiérarchie entre les différents filtrages. Ainsi, par exemple, la box-cohérence est-elle équivalente à la 2B-cohérence en l'absence d'occurrence multiple de variable. Comme la 2B-cohérence s'obtient plus rapidement, cela permet de faire un choix judicieux entre les deux filtrages. Cette démarche n'est toutefois pas entièrement rigoureuse car la 2B-cohérence n'est pas vraiment obtenue en pratique, ni même sa forme d'approximation la plus connue, c.a.d., la w-2B cohérence. En effet, sa définition repose sur la projection (sur les réels) qui n'est pas calculable dans la grande majorité des cas.

Nous avons proposé dans ce chapitre une manière de définir une cohérence à partir d'une projection sur les flottants appelée $\mathcal{F}$-projection. Nous avons en particulier explicité deux cohérences : la $\mathcal{F}$-arc-cohérence et la $\mathcal{F}$-2B-cohérence. Ces « $\mathcal{F}$-cohérences » d'une part tiennent compte des erreurs produites par l'arithmétique des flottants, et d'autre part sont bien calculables. Ce travail n'est toutefois qu'une première étape vers une construction de cohérences calculables. En effet, il ne concerne que les contraintes primitives et repose sur une hypothèse raisonnable mais pas forcément vérifiée (cf. p.§6.10.2). L'étape suivante consisterait à lever cette hypothèse en intégrant dans la définition d'une cohérence la précision garantie pour chaque fonction élémentaire par l'arithmétique sous-jacente. Une cohérence serait donc à la fois paramétrée par un ensemble de flottants et par la précision des calculs sur cet ensemble.

Nous avons également introduit dans ce chapitre un moyen de définir l'approximation d'une cohérence appelé σ-approximation. Ce type d'approximation cumule plusieurs avantages : celui de présenter un « point fixe unique » et d'être immédiat à implémenter pour toute « $\mathcal{F}$-cohérence ». Ainsi, par exemple, la σ-$\mathcal{F}$-arc cohérence s'obtient en remplaçant dans un filtrage par $\mathcal{F}$-arc cohérence chaque $\mathcal{F}$-projection par son approximation aux intervalles délimités par la subdivision σ.

Chapitre 7

Cohérences sur les unions d'intervalles

Sommaire

Dans ce chapitre, nous proposons deux nouvelles cohérences partielles sur les CSP continus utilisant la représentation par unions d'intervalles. Cette approche est motivée par le caractère infructueux de l'algorithme `AC3`, conçu pour des problèmes discrets et appliqué tel quel en domaine continu (cf. §6.3.2). En effet, calculer le point fixe de l'arc–cohérence sur les flottants est un problème de complexité – en temps et en espace – polynomiale en le nombre de flottants. Nous avons vu au §6.5 que la complexité en temps peut être acceptée moyennant l'introduction d'approximations (w–arc–cohérence ou σ–arc–cohérence). En revanche, la complexité en espace est rédhibitoire : il est difficilement concevable de gérer des unions d'intervalles dont la taille est de l'ordre du nombre de flottants.

Face à ce constat, plusieurs travaux ont déjà été menés [Hyv92, Lho93, Fal94, BGGP99]. En général, ils ont abouti à la conclusion suivante : le meilleur moyen de pallier l'inadéquation d'`AC3` aux valeurs réelles consiste à limiter le filtrage aux bornes des domaines. La propriété obtenue s'appelle la *2B*–cohérence (cf. §6.4). La solution proposée *affaiblit* donc la propriété d'arc–cohérence. Nous proposons dans ce chapitre une approche différente, qui consiste à calculer des cohérences plus fortes, basées sur la structure d'union.

Nous introduisons au §7.1 la *box–set cohérence*. Un espace de recherche est box–set cohérent s'il est un *ensemble de boîtes arc–cohérentes*. Nous donnons un moyen très simple de calculer la clôture par box–set cohérence en combinant un filtrage par 2B–cohérence, une projection d'unions d'intervalles et un découpage particulier des domaines appelé *découpage naturel*. Les propriétés théoriques de cet algorithme sont ensuite étudiées, notamment les conditions sous lesquelles l'arc–cohérence est effectivement obtenue.

Nous introduisons au §7.2 la *I-cohérence globale* (IGC). Sa définition repose sur une nouvelle abstraction des CSP continus : les intervalles qui forment le domaine d'une variable sont assimilés à de simples valeurs, si bien que le gruyère peut lui-même être assimilé à un ensemble discret (qui toutefois, change dynamiquement). Il est possible de définir alors une notion de support entre ces différentes valeurs, et donc de calculer d'autres types de cohérences. Parmi ces cohérences, IGC semble la seule à pouvoir être combinée avec l'arc-cohérence (la cohérence obtenue est notée AC–IGC). Nous montrons alors qu'il est possible à partir de l'algorithme de base `AC3` de l'obtenir, moyennant le recours à des ROBDD (*reduced ordered binary decision diagrams*), au lieu

de simples intervalles, pour la représentation des domaines.

Les complexités des algorithmes calculant la box–set cohérence et AC–IGC sont au pire exponentielles, car les problème sont NP-difficiles, mais la complexité en espace est dans les deux cas indépendante du nombre de flottants, ce qui fait leur intérêt ; elle ne dépend en effet que de caractéristiques du CSP (e.g., arité des variables, nombre de périodes des fonctions). D'autre part, en pratique, les complexités ne semblent pouvoir être atteintes que dans des cas pathologiques vraiment particuliers.

7.1 Box–set cohérence

(*Cette section a fait l'objet de la publication* [CTN05a].)

Nous avons vu au §6.3.2 que l'arc–cohérence présentait de très mauvaises performances sur les CSPs continus. La raison étant la complexité en espace, qui est de l'ordre du nombre de flottants. Nous proposons une cohérence plus forte, appelée box–set cohérence qui présente une complexité en espace indépendante du nombre de flottants.

Reprenons l'exemple–type 6.5 p.152. Considérons, à une étape quelconque, un intervalle du domaine de x dans lequel aucune des deux solutions ne se projette. Si nous construisons une boîte avec cet intervalle et n'importe quel intervalle du domaine de y, cette boîte n'est pas arc–cohérente et la clôture par arc–cohérence de cette boîte (cf. définition 6.6) est même l'ensemble vide. Dans cet exemple, la boîte initiale $[1,9] \times [1,9]$ ne contient en fait que deux sous–boîtes arc–cohérentes, qui sont les boîtes dégénérées représentées par les solutions. Notre but est désormais d'isoler ces sous–boîtes particulières. Nous pouvons illustrer ce but de façon plus générale à l'aide de la figure 7.1, qui décrit un système à trois variables liées deux à deux par des contraintes binaires. Les domaines sont des unions, et une arête entre un intervalle I d'une union et un intervalle J d'une autre union signifie que chaque valeur de I possède un support dans J (et réciproquement).

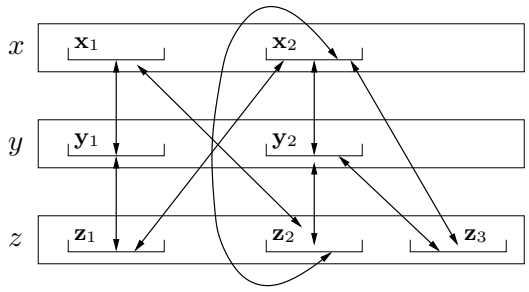


FIG. *7.1: Sous-boîtes arc-cohérentes d'un gruyère.*

Nous voyons que le gruyère $(\mathbf{x}_1 \cup \mathbf{x}_2) \times (\mathbf{y}_1 \cup \mathbf{y}_2) \times (\mathbf{z}_1 \cup \mathbf{z}_2 \cup \mathbf{z}_3)$ est arc–cohérent, mais qu'il n'y a que deux sous–boîtes arc–cohérentes formées avec les intervalles de ce gruyère : $\mathbf{x}_2 \times \mathbf{y}_2 \times \mathbf{z}_2$ et $\mathbf{x}_2 \times \mathbf{y}_2 \times \mathbf{z}_3$.

La boîte $\mathbf{x}_1 \times \mathbf{y}_1 \times \mathbf{z}_1$ n'est pas arc–cohérente car $\mathbf{x}_1$ et $\mathbf{z}_1$ ne sont pas reliés. Plus précisément, $\mathbf{x}_1$, $\mathbf{y}_1$ et $\mathbf{z}_1$ n'appartiennent à aucune sous–boîte arc–cohérente ; ils peuvent donc être supprimés des domaines. Nous étudions dans ce paragraphe une nouvelle cohérence permettant d'isoler les sous–boîtes arc–cohérentes maximales d'un CSP, ainsi que quelques algorithmes qui permettent de la calculer.

Définition 7.1 (Box–set cohérence) *Soit $\mathcal{P} = (\mathcal{C}, \mathcal{X}, \mathcal{E})$ un CSP où $\mathcal{E}$ est un ensemble de boîtes.*
On dit que $\mathcal{E}$ est **box–set cohérent** *si pour toute boîte $B \in \mathcal{E}$, B est arc–cohérente.*
On appelle **clôture par box–set cohérence** *d'un CSP–boîte $(\mathcal{C}, \mathcal{X}, B)$ l'ensemble des sous–boîtes de B arc–cohérentes maximales (pour l'inclusion).*

Supposons momentanément que la clôture $\mathcal{E}$ par box–set cohérence d'un CSP $\mathcal{P}$ soit un ensemble fini de boîtes.

Remarquons alors que le calcul de $\mathcal{E}$ peut être exploité de deux manières.

1. Version *en profondeur d'abord* : chaque boîte de $\mathcal{E}$ peut être utilisée comme nouveau point de choix pour la recherche de solutions de $\mathcal{P}$. Cela signifie qu'avant de calculer une nouvelle boîte de $\mathcal{E}$, la recherche est poursuivie dans cette première boîte (elle est bissectée, etc...). Dans ce cas, le filtrage par box–set cohérence ne fait que produire des sous–arbres de l'arbre de recherche.
2. Version *filtrage pur* : il est également possible de rassembler les boîtes de $\mathcal{E}$ pour former un gruyère. Dans l'exemple de la figure 7.1, le gruyère obtenu est $\mathbf{x}_2 \times \mathbf{y}_2 \times (\mathbf{z}_2 \cup \mathbf{z}_3)$. Cette transformation permet d'éviter la combinatoire, mais implique en général une perte importante d'information.

L'exemple suivant illustre le fait que la box–set cohérence est une propriété plus forte que l'arc–cohérence :

Exemple 7.1 *Soit le CSP* $\{\mathcal{C}, \{x, y, z\}, [-1, 1]^3\}$ *avec pour contraintes* $x^2 = 1$, $x = y$, $y = z$ *et* $z = -x$.
La clôture par arc–cohérence est le gruyère $([-1, -1] \cup [1, 1]) \times ([-1, -1] \cup [1, 1]) \times ([-1, -1] \cup [1, 1])$.
La clôture par box–set cohérence est l'ensemble vide (une boîte $\mathbf{x} \times \mathbf{y} \times \mathbf{z}$ *formée avec* $\mathbf{x} = [1, 1]$ *ou* $\mathbf{x} = [-1, -1]$ *devrait vérifier* $\mathbf{x} = -\mathbf{x}$ *ce qui est impossible).*

La box–set cohérence n'est pas comparable avec la *cohérence globale* (l'enveloppe de toutes les solutions). Le CSP $(\{x = y, y = -x\}, \{x, y\}, [-1, 1] \times [-1, 1])$ par exemple est box–set cohérent alors que seule la boîte $[0, 0] \times [0, 0]$ est globalement cohérente. A l'inverse, dans l'exemple-type la cohérence globale est le point fixe obtenu par 3B-cohérence, c.a.d. la boîte représentée à la figure 6.5 p.161 alors que la clôture par box–set cohérence contient exactement les deux boîtes-solutions.

Nous montrons dans le paragraphe suivant que, contrairement à l'arc–cohérence, il est possible de borner la taille de la clôture par box–set cohérence, c'est à dire la complexité en espace, par une valeur qui ne dépend pas du nombre de flottants. Nous nous intéresserons ensuite (cf. §7.1.2 et §7.1.3) aux moyens d'appliquer cette cohérence.

7.1.1 Convexité, monotonie, complexité

La box–set cohérence repose sur deux propriétés clefs : la convexité [BO97] et la monotonie. Une contrainte c est *convexe* sur une boîte donnée lorsque les projections de c ne créent pas de trou, autrement dit lorsque le résultat de la projection de c sur n'importe quelle variable ne produit qu'un seul intervalle. Une contrainte est *monotone* sur une boîte lorsqu'elle est convexe pour toute sous–boîte. Formellement :

Définition 7.2 (Convexité, monotonie) *Soit* $(\mathcal{X}, \mathcal{C}, B)$ *un CSP-boîte,* $c \in \mathcal{C}$ *une contrainte.*
On dit que c est convexe (sur B) si $\forall x \in \mathcal{X}$, $|\Pi_x^c(B)| = 1$.
On dit que c est monotone (sur B) si c est convexe sur toute sous–boîte de B.

La monotonie est bien une propriété plus forte que la convexité. Par exemple, la contrainte $y = x^2$ est convexe sur $B = B_x \times B_y = [-2, 2] \times [0, 4]$ (puisque $\Pi_x^c(B) = \Pi_y^c(B) = B$) mais non monotone, puisque $\Pi_x^c([-2, 2] \times [1, 4]) = ([-2, -1] \cup [1, 2]) \times [1, 4]$.

Remarque 7.1 *La convexité d'une contrainte c n'a pas de lien avec la continuité de la fonction impliquée dans c. Par exemple, la fonction* $f_1 : (x, y) \longrightarrow x^2 - y$ *est continue sur* $B = \mathbf{x} \times \mathbf{y} = [-2, 2] \times [1, 4]$ *alors qu'en posant* $c_1 : f_1(x, y) = 0$ *la contrainte* c_1 *n'est pas convexe puisque* $\Pi_x^{c_1}(B) = [-2, -1] \cup [1, 2]$. *Inversement,* $f_2 : (x, y) \longrightarrow E(x - y)$, *où* $E(z)$ *est la partie entière de z, est discontinue sur* $B = \mathbf{x} \times \mathbf{y} = [-2, 2] \times [-2, 2]$ *alors qu'en posant* $c_2 : f_2(x, y) = 0$, *la contrainte* c_2 *est convexe puisque* $\Pi_x^{c_2}(B) = \Pi_y^{c_2}(B) = [-2, 2]$.

La notion de convexité sera utile plus loin à travers la proposition suivante :

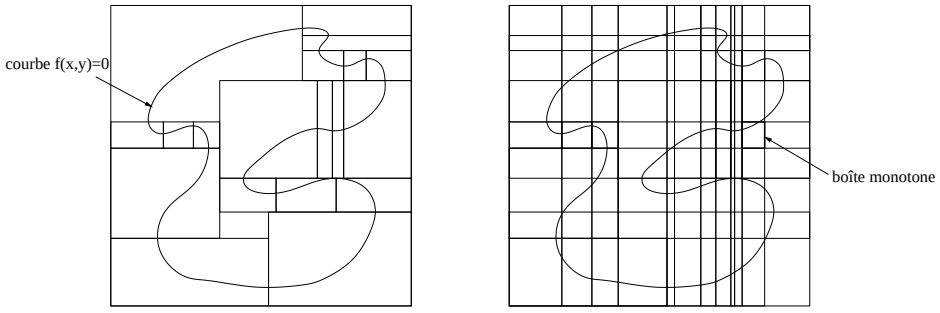
Proposition 7.1
Si chaque contrainte est convexe sur une boîte B alors B est 2B-cohérente ssi B est arc–cohérente.

Preuve. B est 2B-cohérente ssi $\forall (c,x) \in \mathcal{C} \times \mathcal{X}$, $D_x = \Box\Pi_x^c(B)$. Chaque contrainte c étant convexe, $\Box\Pi_x^c(B) = \Pi_x^c(B)$ donc B est 2B-cohérente ssi $\forall (c,x) \in \mathcal{C} \times \mathcal{X}$, $D_x = \Pi_x^c(B)$, c'est à dire, ssi B est arc–cohérente. □

La notion de monotonie nous permet d'énoncer des conditions sous lesquelles la box–set cohérence peut être calculée : les contraintes doivent être *monotones par morceaux* :

Définition 7.3 (Monotonie par morceaux) *Soit $(\mathcal{X}, \mathcal{C}, B)$ un CSP-boîte, $c \in \mathcal{C}$ une contrainte.*
On dit que c est monotone par morceaux (sur B) s'il existe une partition[1] finie de B telle que c soit monotone sur toute boîte appartenant à cette partition.

La figure 7.2 représente une contrainte binaire monotone par morceaux. Il est possible de construire à partir de la partition de B une *grille de monotonie*, en projetant les boîtes de cette partition sur les axes (voir figure). Il est clair que si le domaine d'une variable x est restreint à l'un des intervalles formés sur cet axe, la projection de c sur x avec une boîte ne peut jamais créer de trou.



(a) Partition (b) Grille de monotonie

FIG. *7.2: Contrainte monotone par morceaux*

Définition 7.4 (Degré de monotonie) *Soit $\mathcal{P} = (\mathcal{C}, \mathcal{X}, B)$ un CSP-boîte, c une contrainte monotone par morceaux. On appelle degré de monotonie de c le nombre minimum d'intervalles qu'il faut sur un axe pour construire une grille de monotonie de la contrainte c (c.a.d. un quadrillage de B où c est monotone sur chaque case).*

Dans l'exemple de la figure 7.2 , le degré de monotonie est 12. Les contraintes de la forme $f(x_1, \ldots, x_n) = 0$, où f est une fonction arithmétique sont, à l'exception de cas marginaux (cf. exemple 7.3), monotones par morceaux. De plus, dans le cas de contraintes sans occurrence multiple de variables, le degré de monotonie peut être borné en utilisant les projections théoriques. En effet, tout d'abord, dans le cas d'une contrainte primitive le degré est connu. Par exemple, avec $y = x^2$, on a le degré $p = 2$ et cette valeur correspond au nombre de fonctions implicites ($-\sqrt{y}$ et $+\sqrt{y}$). Pour une fonction arithmétique quelconque, il suffit alors de composer les projections théoriques des contraintes primitives, en tenant compte des domaines des variables implicites. Ceci est illustré dans l'exemple suivant.

[1] Pour être rigoureux, il faudrait plutôt parler de recouvrement puisque les boîtes ont des faces en commun.

Exemple 7.2 *Considérons la contrainte* $\sin(5\pi/8 \times (x-y)^2) = 0.75$*, et la boîte* $\mathbf{x} \times \mathbf{y} = [-1,1] \times [-1,1]$*, le degré p de la contrainte est majorée (pour x ou y) par* $1 \times 2 \times 1 \times 3 \times 1 = 6$*, car :*
- *La contrainte c se décompose en 5 contraintes* $(c_1, \ldots, c_5)$ *avec*
  $(c_1)\ w_1 = x - y \quad (c_2)\ w_2 = w_1^2 \quad (c_3)\ w_3 = 5\pi/8 \times w_2 \quad (c_4)\ w_4 = \sin(w_3) \quad (c_5)\ w_5 = w_4 - 0/75.$
- *Le domaine de* w_1 *est évalué à* $\mathbf{x} - \mathbf{y} = [-2, 2]$ *et la contrainte* c_1 *est toujours de degré 1.*
- *Le domaine de* w_2 *est évalué à* $\mathbf{w}_1^2 = [0, 4]$ *et la contrainte* c_2 *est de degré 2 sur* $\mathbf{w}_1 \times \mathbf{w}_2 = [-2, 2] \times [0, 4]$.
- *Le domaine de* w_3 *est évalué à* $5\pi/8 \times [0, 4] = [0, 5\pi/2]$*. Le degré est* 1.
- *Le domaine de* w_4 *est évalué à* $\sin([0, 5\pi/2]) = [-1, 1]$ *et comme* $[0, 5\pi/2]$ *traverse trois demi-périodes de la fonction sinus, le degré de la contrainte est* 3.
- *Le degré de* c_5 *est* 1.

La borne calculée peut être ajustée moyennant quelques manipulations symboliques. Par exemple, si (c) est la contrainte $(((x^2)^2)^2)^2 = y$, alors $p \leq 1 \times 2 \times 2 \times 2 \times 2 = 8$. En réécrivant cette contrainte $x^8 = y$ on trouve $p \leq 2$. Par contre, aucun résultat ne semble facile à établir dans le cas général, c.a.d., lorsque les variables ont des occurrences multiples. L'estimation précédente basée sur la décomposition est clairement fausse (ex : la contrainte $y = x^3 - x$ est de degré $p = 3$ alors que la décomposition fait intervenir les contraintes $w = x^3$ et $y = w - x$ de degré 1).

Exemple 7.3 *La contrainte* $y = \sin(1/x)$ *n'est pas monotone par morceaux sur* $[0, 1] \times [-1, 1]$ *: un nombre infini de périodes « s'écrasent » en* $x = 0$ *:*

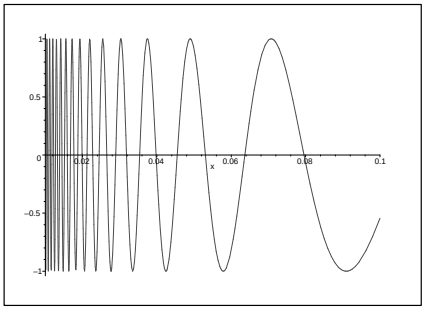


FIG. *7.3: Fonction* $\sin(1/x)$.

Si toutes les contraintes sont monotones par morceaux, en superposant les « grilles de monotonie » obtenues pour chaque contrainte on obtient finalement une unique grille globale[2] dont chaque case représente une boîte où le système entier est monotone :

Définition 7.5 (Boîte monotone) *Soit* $\mathcal{P}$ *un CSP. On appelle boîte monotone (relativement à* $\mathcal{P}$*) toute boîte sur laquelle les contraintes de* $\mathcal{P}$ *sont monotones.*

Proposition 7.2 (Complexité en espace) *La clôture par box–set cohérence d'un CSP-boîte* $(\mathcal{C}, \mathcal{X}, B)$ *où chaque contrainte est monotone par morceaux sur B est un ensemble fini de boîtes dont la taille est bornée par* $(p \times a)^n$*, où p désigne le degré de monotonie maximal des contraintes, a l'arité maximale des variables (c.a.d. le nombre maximal de contraintes impliquant une variable), et n le nombre de variables.*

Preuve. Tout d'abord, en considérant la grille globale décrite ci-dessus, il est possible de partitionner B avec des boîtes monotones dont le nombre est borné par $(p \times a)^n$ (l'axe de chaque variable étant au plus découpé a fois en p intervalles).

[2]Dans la terminologie d'Hyvönen [Hyv92], cette grille porte le nom de « global application space ».

Montrons alors qu'il n'y a au plus qu'une boîte dans la clôture par box–set cohérence qui intersecte une boîte monotone. Par l'absurde, supposons que ce ne soit pas le cas, et considérons deux boîtes disjointes B_1 et B_2 intersectant une boîte monotone. Clairement, $\{B_1, B_2\}$ est la clôture par box–set cohérence de $B_1 \cup B_2$. Ceci implique qu'un filtrage par arc–cohérence de $B_1 \cup B_2$ fait apparaître à une itération une union, sinon le point fixe de ce filtrage serait une unique boîte arc–cohérente incluant B_1 et B_2 ce qui est impossible puisque ces boîtes sont maximales. Considérons la première itération où apparaît une union. L'espace de recherche à l'itération précédente est une boîte, ce qui signifie qu'une projection avec cette boîte a produit une union. Or, dans une boîte monotone les contraintes sont toutes monotones, contradiction. □

Remarque 7.2 *La borne $O(p \times a)^n$ est exponentielle, mais elle est en général fortement réduite par les dépendances entre variables. Si on ajoute dans l'exemple 7.2, la contrainte $x + y = 0$, la clôture par box–set cohérence inclut 6 boîtes, au lieu de $6^2 = 36$.*

Proposition 7.3 (Complexité en temps) *Le nombre de projections nécessaires pour obtenir la clôture par box–set cohérence d'un CSP-boîte est borné par $O((p \times a)^n \times m|\sigma|^3)$, où m désigne le nombre de contraintes et $|\sigma|$ le nombre d'éléments (flottants) de la subdivision des réels compris dans la boîte initiale.*

Preuve. Dans le pire des cas, les $(p \times a)^n$ feuilles sont déployées, et chacune nécessite une propagation où chaque projection ne retire qu'un flottant avant l'épuisement total des flottants qu'elle contient. On montre [Lho94] que la clôture par arc–cohérence dans ce cas est en $O(m|\sigma|^3)$ (on retrouve une complexité similaire à celle obtenue en domaine discret). □

7.1.2 Découpage naturel

Dans ce paragraphe et le suivant, nous décrivons un algorithme de filtrage par box–set cohérence.

Tout d'abord nous ne pouvons plus nous placer dans le modèle théorique (cf. 6.3.1) ; les problèmes de convergence nous empêcheraient de définir convenablement un algorithme. D'autre part, nous ne souhaitons pas non plus spécifier explicitement quelle approximation on utilise. En effet, pour être rigoureux, il faudrait remplacer la box–set cohérence définie ci-dessus par la σ-box set cohérence (ou w-box set cohérence), introduire σ dans chacune des définitions relatives à la box–set cohérence, et énoncer un certain nombre d'hypothèses sur les calculs avec flottants, comme nous l'avons fait au chapitre précédent. Le texte s'en trouverait inutilement surchargé. Nous passerons donc outre ces détails en conservant la seule propriété qui nous intéresse : les points fixes sont atteints en un nombre fini d'itérations. Il faut garder à l'esprit que cette propriété se traduit en terme de σ-cohérence.

L'idée de la méthode de base [Hyv92] consiste simplement à calculer successivement les projections des contraintes et à introduire un point de choix chaque fois qu'une projection produit plusieurs intervalles, c'est à dire lorsqu'un « trou » apparaît dans le domaine de la variable sur laquelle on projette. Chaque intervalle de l'union remplace le domaine courant. Nous appelons un tel point de choix un découpage *naturel* par opposition à la bissection. Notons que dans la version *en profondeur d'abord*, le découpage naturel a comme avantage sur la bissection de supprimer du domaine de la variable les parties correspondantes aux trous, et qu'il évite également d'avoir à dupliquer un point de découpage (comme le point-milieu dans le cas de la bissection). En revanche, le découpage naturel peut conduire à des arbres déséquilibrés, car il n'y a aucune raison que les intervalles d'une union soient de rayon homogène. Une étude comparative approfondie des performances du découpage naturel est effectuée dans [BMR05].

L'algorithme de base mélange projection et découpage naturel : il construit un arbre de recherche. Une boîte cesse d'être coupée lorsque le point fixe des projections est atteint ; elle devient une feuille de cet arbre. Un espace de recherche $\mathcal{D}$ est désormais représenté par deux ensembles de boîtes $\mathcal{L}$ (*left*) et $\mathcal{R}$ (*right*). L'ensemble $\mathcal{L}$ représente la partie déjà parcourue de l'arbre de recherche (voir figure 7.4). Il contient les feuilles déjà calculées à gauche du nœud courant. L'ensemble $\mathcal{R}$ représente la pile des nœuds à traiter (en "attente") ainsi que le nœud courant.

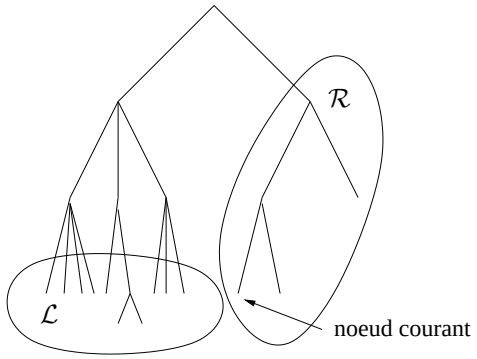


FIG. *7.4: Espace de recherche sous forme d'arbre*

La méthode repose sur l'opérateur suivant :

Définition 7.6 (Opérateur de projection-découpage) *Soit $\mathcal{C}$ un ensemble de contraintes, $\mathcal{X} = \{x_1, \ldots, x_n\}$ un ensemble de variables et $(c, x_i) \in \mathcal{C} \times \mathcal{X}$. On appelle opérateur de projection-découpage et on note $\Xi^c_{x_i}$ la fonction qui à une boîte B associe un ensemble de boîtes $\Xi^c_{x_i}(B)$ tel que*
- *$\Xi^c_{x_i}(B) := \emptyset$ si $\Pi^c_{x_i}(B) = \emptyset$;*
- *$\Xi^c_{x_i}(B) := \{B^{(1)}, \ldots, B^{(p)}\}$ sinon, avec $B^{(j)} = B_{x_1} \times B_{x_{i-1}} \times \mathbf{x}_i^{(j)} \times B_{x_{i+1}} \times B_{x_n}$ où $\mathbf{x}_i^{(1)} \cup \ldots \cup \mathbf{x}_i^{(p)}$ est l'union représentant le domaine de x_i dans la projection $\Pi^c_{x_i}(B)$.*

On remarque qu'une projection-découpage équivaut à une simple projection dans le cas d'une contrainte convexe (à ceci près qu'elle retourne un singleton contenant une boîte au lieu d'une simple boîte). Nous pouvons maintenant définir l'algorithme de base, appelé `basic-box-set`. Cet algorithme correspond à celui d'Hyvönen [Hyv92].

Algorithme `basic-box-set` (paramètre : CSP-boîte $(\mathcal{C}, \mathcal{X}, B_{init})$, retour : ensemble de boîtes)

```
1  L ← ∅
2  R ← {B_init}
3  tant que R ≠ ∅ faire
4    soit B ∈ R, R' ← R\{B}
5    si Π(B) = B alors L ← L ∪ {B}, R ← R'      // rappel : Π enchaîne toutes les projections (cf. p.150)
6    sinon
7      soit (c, x) un couple tel que Π^c_x(B) ≠ B
8      R ← Ξ^c_x(B) ∪ R
9    fin si
10 fin tant que
11 retourner L
```

L'algorithme termine car nous avons supposé ci-dessus que le point fixe d'un filtrage par arc–cohérence était atteint en un nombre fini d'itérations. Un des invariants de `basic-box-set` est que $\forall B \in \mathcal{L}$, $\Pi(B) = B$; ce qui signifie que les boîtes de la partie gauche de l'arbre sont arc–cohérentes. La proposition suivante énonce que cet algorithme permet bien d'effectuer un filtrage par box–set cohérence :

Proposition 7.4
L'algorithme `basic-box-set` *calcule la clôture par box–set cohérence d'un CSP.*

Preuve. Clairement, `basic-box-set` renvoie un ensemble de boîtes disjointes (par définition de la projection-découpage) et arc–cohérentes (en vertu de la ligne 5 de l'algorithme). Il reste à prouver que ces boîtes sont

maximales, c.a.d. qu'il n'existe pas de boîtes arc–cohérentes plus grandes.

Soient B la boîte initiale et B^* la boîte obtenue au dernier passage dans la boucle où $\mathcal{L} = \emptyset$ (donc le passage qui précède celui du premier découpage naturel). Notons $\{B^{(1)}, ..., B^{(p)}\} := \Xi_x^c(B^*)$ les boîtes-filles obtenues au passage suivant. Il est clair que la clôture par box–set cohérence de B est aussi la clôture par box–set cohérence de B^*. Prouvons que cette dernière est l'union des clôtures par box–set cohérence de $B^{(1)}$,...,$B^{(p)}$. Supposons que ce ne soit pas le cas. Il existe une boîte B' dans la clôture par box–set cohérence de l'une des boîtes filles, par exemple $B^{(1)}$, qui n'est pas maximale. Par conséquent B' peut être élargie, c.a.d., une valeur adhérente à B' peut être ajoutée au domaine d'une variable. Comme B' est maximale pour $B^{(1)}$ et que B^* et $B^{(1)}$ diffèrent uniquement par le domaine de la variable bissectée (disons x), alors la valeur ajoutée possède un support supplémentaire dans $B_x^* \cap B_x^{(1)}$. Mais le domaine $B_x^{(1)}$ ne peut être élargi, puisqu'une nouvelle valeur serait forcément à l'intérieur d'un trou ou en dehors des bornes de B^*, et dans tous les cas, incohérente. Par une induction immédiate, la clôture par box–set cohérence de la boîte initiale est l'union des clôtures par box–set cohérence des feuilles calculées par l'algorithme. Les feuilles sont donc bien maximales. □

L'algorithme `basic-box-set` est inefficace. La raison est que le découpage est prioritaire sur le filtrage : dès que la projection d'une contrainte fait apparaître un trou le découpage est appliqué, dupliquant un espace de recherche où le filtrage n'a pas été mené à terme. Une explosion combinatoire se produit presque systématiquement, à l'image de l'exemple 7.4.

Exemple 7.4 *Soit le CSP* $(\{c_1, \ldots, c_{50}, c'\}, \{x_1, \ldots, x_{50}\}, [-1, 1]^{50})$ *avec*

$$\begin{array}{lll} \forall i \in [1..50] & (c_i) & x_i^2 = 1 \\ & (c') & x_1 = 0. \end{array}$$

Si les contraintes sont traitées dans l'ordre $c_1 \prec \ldots \prec c_{50} \prec c'$*, alors l'algorithme d'Hyvönen déploie un arbre de taille* 2^{50} *(chaque contrainte* c_i *introduisant un découpage naturel) avant de s'apercevoir au niveau de chaque feuille que la contrainte* c' *rend le système insatisfiable, et de supprimer la boîte.*

7.1.3 Méthode `Lazy-box-set`

Il est possible d'améliorer l'algorithme d'Hyvönen en filtrant une boîte par 2B-cohérence avant de procéder à un découpage naturel. Ce filtrage anticipe les projections calculées pour les sous–boîtes et permet donc de limiter la combinatoire. Il suffit d'insérer dans l'algorithme `basic-box-set` les deux lignes suivantes :

```
    4    soit B tel que R = B ∪ R'
⟶  4'   B ← Φ2B(B)
⟶  4"   si B = ∅ alors R ← R' sinon
    5      si Π(B) = B alors L ← L ∪ {B}, R ← R'
```

Ces lignes permettent de repousser le plus possible le découpage naturel en filtrant les domaines. On prouve que les feuilles sont de nouveaux arc–cohérentes et maximales, en reprenant la preuve de la proposition 7.4 et en s'appuyant sur la proposition 7.1. L'algorithme obtenu calcule bien la clôture par box–set cohérence du CSP.

Une seconde amélioration substantielle peut être apportée à l'opérateur `basic-box-set`. Les lignes 5 et 7 de l'algorithme nécessitent a priori un recalcul des projections (en plus de la ligne 8). Il est possible d'éviter cela en mémorisant au cours du filtrage par 2B-cohérence l'ensemble des contraintes potentiellement non convexes, c'est à dire celles dont la dernière projection sur une des variables a produit une union d'intervalles. Plus précisément, nous remplaçons l'appel à $\Phi_{\texttt{2B}}$ par un filtrage par 2B-cohérence mémorisant également les contraintes potentiellement non convexes. Ce filtrage, noté $\Phi_{\texttt{2B+}}$ retourne, en plus d'une boîte, un ensemble S de contraintes candidates.

L'algorithme `Lazy-box-set` s'écrit finalement :

Algorithme `lazy-box-set` (paramètre : CSP-boîte $(\mathcal{C}, \mathcal{X}, B)$, retour : ensemble de boîtes)

```
1  L ← ∅
2  R ← {B}
3  tant que R ≠ ∅ faire
4   soit B ∈ R, R' ← R\{B}
5   (B, S) ← Φ2B+(B)
6   si B = ∅ alors R ← R'
7   sinon
8    trou←faux          // sert à mémoriser la présence d'une contrainte non convexe
9    pour tout c ∈ S    // parcours des contraintes "candidates"
10     pour tout x ∈ Var(c)
11      E ← Ξ^c_x(B)
12      si |E| > 1 alors R ← E ∪ R,   trou←vrai    // contrainte non convexe
13     fin pour
14    fin pour
15    si (trou=faux) alors L ← L ∪ {B}
16   fin si
17  fin tant que
18  retourner L
```

Le filtrage $\Phi_{\texttt{2B+}}$ est facile à programmer : il suffit, en plus du filtrage à proprement parler, de stocker dans une table indicée par les couples (c, x) un drapeau signalant si la dernière projection de c sur x a fait apparaître un trou. La détection de convexité représente donc un temps négligeable.

7.1.4 Intégration de `HC4Revise`

Les projections dans `lazy-box-set` (que ce soit pour le filtrage $\Phi_{\texttt{2B}}$ ou l'opérateur Ξ) peuvent naturellement être calculées à l'aide d'`HC4Revise` (cf. §6.8 p.164), mais ce bien sûr, au prix d'une approximation.

Avant de voir ce que nous calculons en intégrant `HC4Revise`, nous pouvons apporter une dernière amélioration. Elle consiste à utiliser la version "tout intervalle" d'`HC4Revise`, notée $\texttt{HC4Revise}_{\square}$ (cf. §6.8) lors du filtrage par 2B-cohérence. Le recours à $\texttt{HC4Revise}_{\square}$ permet de limiter la projection avec unions uniquement à la ligne 11 de l'algorithme `lazy-box-set`. Cette ligne est exécutée un nombre négligeable de fois en comparaison au nombre d'appels à l'opérateur $\texttt{HC4Revise}_{\square}$ engendré par la propagation du filtrage 2B. L'algorithme `lazy-box-set` applique donc une cohérence sur les unions d'intervalles en réduisant les calculs avec unions au strict minimum (d'où son nom), ce qui le rend en pratique comparable dans sa version *en profondeur d'abord* à une recherche « classique » alternant filtrage 2B et bissection.

Tout d'abord, montrons que cette version permet la détection de convexité dans le cas de contrainte sans occurrence multiple de variable. En effet, toute contrainte non convexe de ce type provoque nécessairement l'apparition d'un trou au niveau d'une projection sur une variable implicite. Le drapeau peut donc être levé le cas échéant. En revanche, la détection est rendue plus faible qu'avec `HC4Revise` (version "unions"), car l'apparition d'un trou au niveau d'une variable implicite ne se répercute pas nécessairement sur les variables de la contrainte, ce qui signifie que la contrainte n'est pas forcément non convexe. Par exemple, $c : (x-y)^2 = 1$ est convexe sur $B = [-1,1] \times [-1,1]$ puisque $\Pi^c_x(B) = \Pi^c_y(B) = B$ alors que la projection sur $(x-y)$ produit l'union $[-1,-1] \cup [1,1]$.

De plus, nous verrons que la projection d'une contrainte détectée non-convexe peut réduire les bornes d'une variable malgré le point fixe d'$\texttt{HC4Revise}_{\square}$. En conséquence, l'algorithme doit être légèrement modifié. La ligne 12 doit être remplacée par la suivante :

```
12    si |E| > 1 ou E ≠ {B} alors R ← E ∪ R,   trou←vrai.
```

On ajoute la condition $\mathcal{E} \neq \{B\}$ car même si aucune union n'apparaît à ce point du programme, le point fixe n'est pas atteint. Si la projection d'une contrainte marquée comme non convexe réduit le domaine d'une variable, la boîte B doit subir un nouveau tour de boucle, qu'il y ait eu un trou ou non : elle n'est pas considérée comme une feuille. Sans cette modification, la proposition suivante ne pourrait pas être établie.

Proposition 7.5 *Soit un CSP-boîte tel qu'aucune contrainte ne contienne d'occurrence multiple de variable. La combinaison de* `lazy-box-set` *et* `HC4Revise`$_\square$ *calcule la clôture par box–set cohérence de ce CSP.*

Preuve. Montrons qu'en utilisant `HC4Revise`$_\square$ dans `lazy-box-set`, les boîtes contenues dans $\mathcal{L}$ sont arc–cohérentes. Tout d'abord, elles sont un point fixe pour `HC4Revise`$_\square$. Considérons une boîte B de $\mathcal{L}$ et une contrainte c.

- Si aucun trou n'est apparu lors des dernières projections sur les variables (y compris implicites) de c avec B, cela signifie qu'aucune enveloppe n'a été effectuée, et dans ce cas `HC4Revise`$_\square$ a calculé la même chose que `HC4Revise`.
- Si un trou est apparu lors d'une des dernières projections de c, elle a donc été marquée comme potentiellement non convexe. La ligne 11 a donc été exécutée pour c, puisque dans le cas d'une feuille (une boîte de $\mathcal{L}$) toutes les contraintes de $\mathcal{S}$ sont passées en revue. Or, la projection à la ligne 11 est effectuée cette fois via `HC4Revise` et cette projection n'a provoqué aucune réduction puisqu'une réduction aurait empêché, grâce à la nouvelle condition $\mathcal{E} \neq \{B\}$, que B soit ajoutée à $\mathcal{L}$.

On a donc atteint le point fixe de `HC4Revise`. Or, d'après la proposition 6.6 p.166, `HC4Revise` calcule des projections exactes pour des contraintes sans occurrence multiple de variable. On a donc bien atteint le point fixe des projections, c'est à dire l'arc–cohérence.

La maximalité des feuilles se prouve de la même façon que pour la proposition 7.4. □

Une erreur consiste à déduire de la proposition précédente que dans le cas d'un CSP-boîte quelconque (avec ou sans occurrence multiple de variable) la combinaison de `lazy-box-set` et `HC4Revise`$_\square$ calcule la restriction de la box–set cohérence de la décomposition de ce CSP. L'exemple suivant montre que cela est impossible.

Exemple 7.5 *Soit $\mathcal{P}$ le CSP $(\{c\}, \{x, y\}, \mathbf{x} \times \mathbf{y})$ avec $\mathbf{x} = [0, 4]$, $\mathbf{y} = [0, 4]$ et*

$$(c) \quad (x - y)^2 = 4.$$

La décomposition de $\mathcal{P}$ est le CSP $\mathcal{P}' = (\{c_1, c_2\}, \{x, y, w\}, \mathbf{x} \times \mathbf{y} \times \mathbb{R})$ avec

$$\begin{array}{ll} (c_1) & x - y = w, \\ (c_2) & w^2 = 4. \end{array}$$

$\mathcal{P}$ est arc–cohérent, donc box–set cohérent, alors que la clôture par box–set cohérence de $\mathcal{P}'$ est la paire de boîtes $\{[0, 2] \times [2, 4] \times [-2, -2], [2, 4] \times [0, 2] \times [2, 2]\}$.

Ce phénomène n'est pas très surprenant puisqu'en décomposant le système nous faisons apparaître de nouvelles variables pour lesquelles il devient possible d'effectuer un découpage naturel. Il peut en résulter au final des boîtes plus petites. On peut montrer que l'arc–cohérence d'un CSP sans occurrence multiple de variables est une propriété plus faible que l'arc–cohérence de sa décomposition. Il en résulte une situation hybride, résumée dans le tableau suivant :

2B-cohérence	$\Longrightarrow$	2B-cohérence de la décomposition,
arc–cohérence	$\Longleftarrow$	arc–cohérence de la décomposition.

TAB. *7.1: Comparaison des filtrages en l'absence d'occurrence multiple de variables*

En revanche, dans le cas général (en présence d'occurrence multiple), il n'y a pas de comparaison possible. Dans le cas général, le résultat obtenu par `lazy-box-set` et $\texttt{HC4Revise}_{\square}$ est la clôture par box–set cohérence du *renommage* du CSP, où une variable implicite est substituée pour chaque nouvelle occurrence d'une même variable[3].

Exemple 7.6 (Renommage) *La contrainte* $c : (x-y)^2 = x$ *est équivalente au système sans occurrence multiple de variable formé des deux contraintes suivantes :*

$$\begin{array}{ll}(c_1) & (x-y)^2 = w,\\ (c_2) & w = x.\end{array}$$

Les variables x *et* w *dans le nouveau système sont appelées les* **alias** *de* x*.*

Remarquons que le renommage produit un CSP intermédiaire entre le CSP original et sa décomposition, en terme de nombre de variables. En particulier, un CSP produit par renommage peut contenir des contraintes portant sur plus de trois variables.

Proposition 7.6 *Pour tout CSP-boîte, la combinaison de* `lazy-box-set` *et* $\texttt{HC4Revise}_{\square}$ *calcule la clôture par box–set cohérence du renommage de ce CSP.*

Preuve. En présence d'occurrences multiples de variables, `HC4Revise` et $\texttt{HC4Revise}_{\square}$ se comportent exactement comme s'ils étaient appliqués sur le renommage du système : par exemple, la première opération effectuée par ces algorithmes avec une contrainte est d'affecter aux alias d'une même variable x le domaine de x. Si nous notons ces alias $w_1, ..., w_n$, la première étape de ces opérateurs est bien équivalente à une projection des $w_i = x$ sur les alias w_i dans le renommage. Voir §6.8 pour plus de détails.

Rappelons toutefois qu'une précaution doit être prise avec des occurrences multiples : la contrainte courante doit être réintroduite dans l'agenda de propagation après un calcul de projection (cf. remarque 6.2 p.167).

Ainsi, `lazy-box-set` fournit le même résultat que s'il était appliqué directement sur le renommage du problème. Le renommage n'ayant pas d'occurrence multiple de variables, il suffit pour conclure d'appliquer la proposition 7.5. □

7.2 IGC (*I-cohérence globale*)

(*Cette section a fait l'objet de la publication* [CTN05b].)

Nous proposons dans cette section une nouvelle cohérence partielle, notée IGC, intermédiaire entre l'arc–cohérence et la box–set cohérence (c.a.d., plus forte que la première et plus faible que la seconde). Cette cohérence peut être appliquée par un algorithme de filtrage pur, c'est à dire sans point de choix. Nous nous limitons à un jeu très simple de contraintes : les opérations arithmétiques de base, les fonctions monotones (comme exp) et la fonction puissance. L'ajout des autres contraintes primitives (sin, etc...) ne pose pas de problème en théorie mais entraînerait l'utilisation de structures de données plus complexes et rendrait l'analyse de cas plus fastidieuse (nous y reviendrons).

7.2.1 Introduction

Notre but consiste à améliorer l'algorithme de filtrage par arc–cohérence (ou plutôt σ–arc cohérence), grâce au découpage de l'espace de recherche en boîtes monotones que nous avons introduit au §7.1.1. L'idée de base est

[3] Il ne semble pas qu'il y ait de termes dédiés dans la littérature pour distinguer cette transformation de la décomposition. Suivant les articles, la *décomposition* peut désigner l'une ou l'autre de ces transformations.

la suivante : lorsqu'une projection fait apparaître une union, les intervalles de cette union "intersectent" des boîtes monotones différentes. Si on enrichit la structure d'union d'intervalles en ajoutant à chaque intervalle une information indiquant sur quelles boîtes monotones il prend support et que l'on propage cette information au fil des projections, il est alors possible par recoupement d'éviter une explosion des unions telle que nous l'avons observée au §6.3.2 p.151.

Pour décrire informellement cette idée, considérons par exemple deux variables x et y liées par une contrainte $y = x^2$. Supposons que cette contrainte ne soit pas convexe sur la boîte initiale. Un intervalle de y peut se projeter alors en deux intervalles de x, étiquetés par un "+" et un "−" sur la figure 7.5.

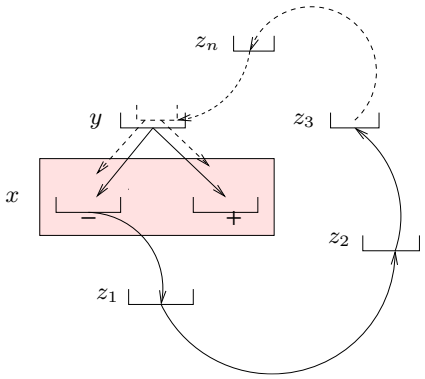


FIG. *7.5: Exemple d'incompatibilité conduisant à une explosion des unions*

Considérons par exemple l'intervalle "−" . La projection sur z_1 peut faire apparaître un intervalle ayant support uniquement sur l'intervalle "−" de x. Les projections sur les variables suivantes peuvent également faire apparaître des intervalles ayant support indirectement sur l'intervalle "−", et uniquement celui-ci. Supposons que ce soit le cas pour la variable y elle-même. La projection sur x produit de nouveau deux intervalles, or la partie positive de cette projection est forcément incohérente puisque l'intervalle projeté n'est compatible qu'avec l'intervalle "−" de x. C'est ce genre d'incompatibilité qui est à l'origine de l'explosion : chaque projection de y sur x peut doubler le nombre d'intervalles.

Cette observation suggère une vision plus "macroscopique" de notre gruyère. Plutôt que de limiter notre vision aux supports entre valeurs réelles, on peut regarder globalement sur quels intervalles un intervalle possède des supports. On obtient alors un graphe très simplifié, qui permet de détecter qu'un intervalle entier est incohérent. La définition suivante se place directement dans la cas ternaire (mais les exemples de cette section sont donnés dans le cas binaire) :

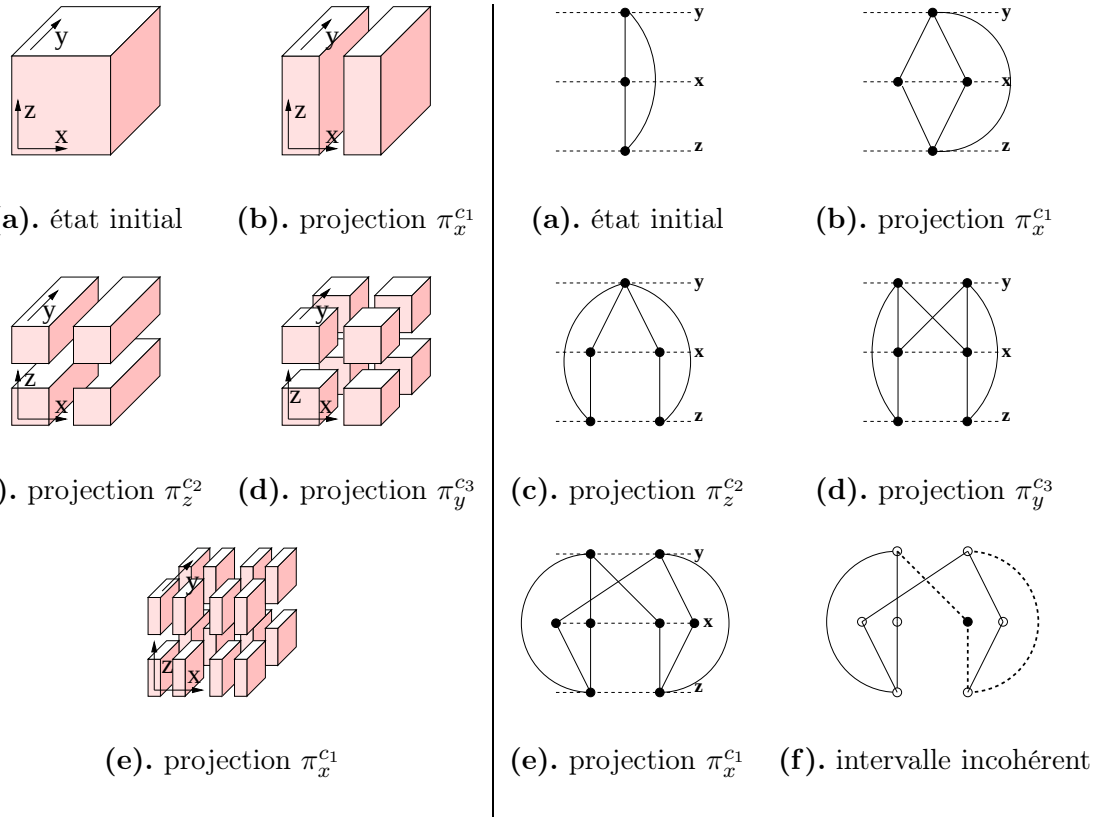
Définition 7.7 (I-Support) *Soient x, y et z, trois variables reliées par une contrainte c. Soit $\mathbf{x}$ (resp. $\mathbf{y}$, $\mathbf{z}$) un intervalle du domaine de x (resp. y, z). On dit que $\mathbf{x}$, $\mathbf{y}$ et $\mathbf{z}$ sont I-supports entre eux (ou compatibles) s'il existe $v_i \in \mathbf{x}$, $v_j \in \mathbf{y}$ et $v_k \in \mathbf{z}$ tel que $c(v_i, v_j, v_k)$.*

Afin d'illustrer la notion d'I-support, nous reprenons l'exemple-type 6.5 p.152, mais dans une version à trois variables qui facilite la représentation graphique.

Exemple 7.7 *Soit $\mathcal{P} = (\{c_1, c_2, c_3\}, \{x, y, z\}, \mathcal{D}_x \times \mathcal{D}_y)$ le CSP suivant :*
$\mathcal{D}_x = \mathcal{D}_y = \mathcal{D}_z = [1, 9]$ et

$$\begin{array}{ll} (c_1) & (\frac{3}{4}(x-5))^2 = y, \\ (c_2) & x = z, \\ (c_3) & z = y. \end{array}$$

La figure 7.6 illustre dans l'exemple 7.7, à gauche, l'évolution du gruyère et à droite, son évolution du point de vue macroscopique, c'est à dire du graphe dont les sommets représentent des intervalles et dont les arêtes représentent des I-supports.



(a). état initial **(b).** projection $\pi_x^{c_1}$ **(a).** état initial **(b).** projection $\pi_x^{c_1}$

(c). projection $\pi_z^{c_2}$ **(d).** projection $\pi_y^{c_3}$ **(c).** projection $\pi_z^{c_2}$ **(d).** projection $\pi_y^{c_3}$

(e). projection $\pi_x^{c_1}$ **(e).** projection $\pi_x^{c_1}$ **(f).** intervalle incohérent

FIG. *7.6: Filtrage AC (vue "valeurs" et vue "intervalles")*

La figure 7.6.(f) souligne le fait que l'intervalle correspondant au point noirci ne peut contenir de solution. Ses supports, apparaissant en pointillés, montrent en effet une incompatibilité. Notre but consiste à éliminer les intervalles présentant ce genre d'incompatibilité. Pour cela, nous prouverons que propager une information sur les boîtes monotones tel que nous l'avons fait ci-dessus (avec simplement une étiquette "+" et "−") suffit. Les intervalles compatibles entrent eux forment une *I–clique* (rappelons qu'une clique est un sous-graphe complet) :

Définition 7.8 (I–clique) *Soit $\mathcal{P}$ un CSP-gruyère.*
Une **I–clique** *de $\mathcal{P}$ est un n-uplet constitué d'un intervalle du domaine de chaque variable de $\mathcal{P}$, tel que les n intervalles soient tous compatibles trois par trois (ou deux par deux pour les contraintes binaires).*

Ainsi, dans la figure 7.6.(f), il n'est pas possible de former une I–clique avec l'intervalle noirci.

Remarque 7.3 *Pour être plus précis, en présence de contraintes ternaires, une I-clique n'est pas une clique dans un graphe mais dans un hypergraphe. Une manière de ramener cet hypergraphe à un simple graphe consiste à ajouter des sommets pour chaque couple de variables impliqué dans une contrainte ternaire. Un exemple est donné à la figure 7.10 plus loin.*

Nous pouvons maintenant définir la propriété que nous cherchons à calculer.

7.2.2 Définition d'IGC

Définition 7.9 (IGC) *Un CSP-gruyère $(\mathcal{C}, \mathcal{X}, G)$ est* **globalement I-cohérent** *(IGC) si tout intervalle du domaine de n'importe quelle variable appartient à une I–clique de G.*

L'intérêt de cette cohérence au niveau "intervalles" est de la combiner avec une cohérence au niveau "valeurs" :

Définition 7.10 (AC-IGC) *Un CSP-gruyère $(\mathcal{C}, \mathcal{X}, G)$ est* **AC-IGC** *s'il est à la fois arc–cohérent et globalement I-cohérent.*

La cohérence AC–IGC est plus faible que la box–set cohérence. En effet, dans un problème AC-IGC, une I–clique n'est pas forcément une boîte arc–cohérente, et n'en contient pas forcément non plus :

Exemple 7.8 *Soit $\mathcal{P} = (\{c_1, c_2, c_3\}, \{x_1, x_2, y\}, \mathcal{D}_{x_1} \times \mathcal{D}_{x_2} \times \mathcal{D}_y)$ un CSP-gruyère avec $\mathcal{D}_{x_1} = [-2, 2]$, $\mathcal{D}_{x_2} = [-2, 2]$, $\mathcal{D}_y = [-2, -1] \cup [1, 2]$, et :*

$$\begin{array}{ll} (c_1) & (y - x_1)^2 \in [0, 1], \\ (c_2) & (y - x_2)^2 \in [0, 1], \\ (c_3) & x_1 = -x_2. \end{array}$$

$\mathcal{P}$ est bien AC-IGC, et on peut voir par exemple sur la figure 7.7 ci-dessous que $([-2, 2], [-2, 2], [1, 2])$ est une I–clique. Cependant, si on suit les supports de chaque valeur, on s'aperçoit que la boîte $[-2, 2] \times [-2, 2] \times [1, 2]$ n'est pas arc–cohérente. Dans ce cas, elle contient une sous–boîte arc–cohérente, en l'occurrence $[0, 0] \times [0, 0] \times [1, 2]$. Mais il suffit d'ajouter une contrainte, par exemple, $(x_1 - x_2)^2 = 1$, pour que ce ne soit plus le cas (l'ajout de cette contrainte ne change pas le fait que $\mathcal{P}$ soit AC-IGC).

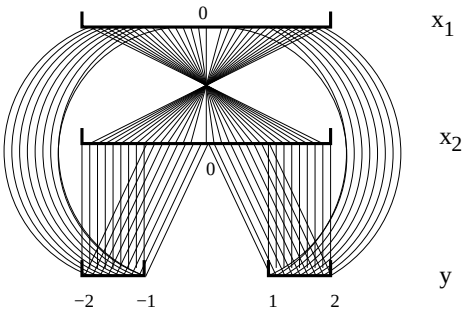


FIG. *7.7: Exemple de I–cliques*

Ainsi, contrairement à ce qu'il pourrait sembler, éviter les points de choix de la box–set cohérence en propageant les informations sur les boîtes monotones, ne revient pas à simuler un algorithme non-déterministe par un algorithme déterministe (ce qui n'aurait, bien sûr, aucun intérêt).

7.3 Algorithme naïf de filtrage AC–IGC

Nous décrivons maintenant notre approche pour calculer la clôture AC–IGC d'un CSP. Notre stratégie consiste à éliminer à chaque étape du filtrage par arc–cohérence les intervalles n'appartenant pas à une I–clique. Une première approche consiste à greffer un algorithme générique de graphes pour effectuer la recherche de clique maximale. Nous proposons une autre approche, plus efficace. Elle consiste à définir un nouvel opérateur de

projection qui, en plus de la projection à proprement parler, vérifie de façon incrémentale les compatibilités (au sens de la définition 7.7) entre les intervalles des domaines. L'efficacité tient donc à son imbrication peu coûteuse dans la boucle de propagation.

Nous commençons par décrire un premier algorithme naïf qui, lui, est inefficace mais permet d'expliquer plus clairement le principe de la projection étiquetée. Cet algorithme sera ensuite remplacé par un autre (`Etiq-AC`), basé sur le même principe mais utilisant une structure de données plus élaborée.

7.3.1 Description informelle de la projection étiquetée (naïve)

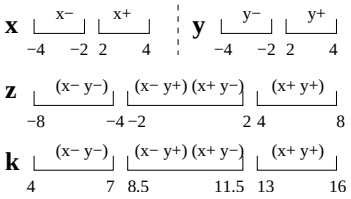
L'algorithme consiste à associer à chaque intervalle du gruyère, une étiquette mémorisant les boîtes monotones dont ses valeurs sont issues depuis le début de la propagation. Pour toute contrainte non convexe (puissance ou division), une projection de cette contrainte sur x crée au plus deux intervalles de part et d'autre de la valeur 0 (nous avons écarté les fonctions trigonométriques justement pour éviter d'avoir à prendre en compte d'autres types de séparation). Il suffit donc de savoir si un intervalle prend support sur la partie négative x (ce qu'on note x^-) ou sur la partie x^+, pour garder une trace de cette projection.

Lorsqu'on intersecte un intervalle étiqueté x^- avec un intervalle étiqueté x^+, il s'agit donc d'une incompatibilité, et le résultat de cette intersection peut être effacé du domaine. Nous allons dérouler l'algorithme sur un exemple :

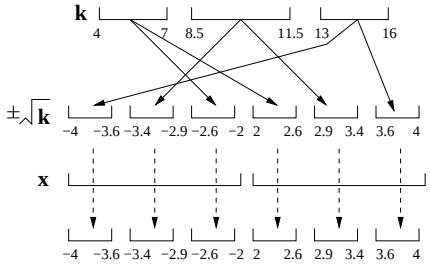
Exemple 7.9
Soit le CSP $(\{c_1, c_2, c_3, c_4\}, \{x, y, z, k\}, \mathcal{D}_x \times \mathcal{D}_y \times \mathcal{D}_z \times \mathcal{D}_k)$ *avec* $\mathcal{D}_x = [-4, 4]$, $\mathcal{D}_y = [-4, 4]$, $\mathcal{D}_z = [-8, 8]$, $\mathcal{D}_k = [4, 16]$ *et*

$$\begin{array}{ll} (c_1) & x^2 = k, \\ (c_2) & y^2 = k, \\ (c_3) & z = x + y, \\ (c_4) & k = 0.75 \times z + 10. \end{array}$$

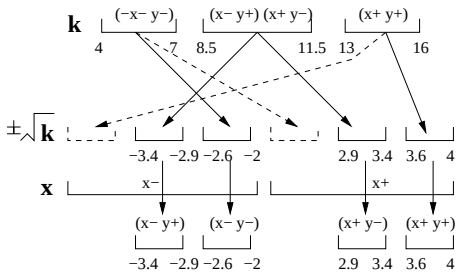
Tout d'abord, la projection de c_1 sur x donnera deux intervalles, $[-4, -2]$ (étiqueté x^-) et $[2, 4]$ (étiqueté x^+). La même observation vaut pour la projection de c_2 sur y. La figure suivante montre le résultat de la projection de c_3 sur z et de c_4 sur k.



L'intervalle $[-2, 2]$ de z est étiqueté $\{(x^-, y^+), (x^+, y^-)\}$ ce qui signifie qu'une I-clique formée avec cet intervalle prend support sur la partie positive de x et négative de y, ou inversement. La figure suivante montre ce qu'une projection classique de c_1 sur x effectuerait à ce stade :



La figure suivante montre que la projection étiquetée permet, elle, d'éliminer deux intervalles.



Les quatre intervalles restants contiennent chacun une solution du problème, et leurs étiquettes indiquent l'unique boîte monotone à laquelle appartient cette solution. Cet exemple montre également qu'il ne suffit pas de mémoriser une information par variable (exemple : x^-), mais bien toutes les combinaisons possibles (comme $\{x^-y^+, x^+y^-\}$), sans quoi la propriété IGC n'est pas obtenue, et surtout, l'explosion observée avec l'arc–cohérence se produit de nouveau (l'intervalle $[8.5, 11.5]$ de k perdrait en effet toute information, et un déroulement de l'exemple montrerait alors de nouveau un dédoublement de cet intervalle à chaque étape).

7.3.2 Définitions des étiquettes

Soient $\mathcal{P} = (\mathcal{C}, \mathcal{X}, B)$ un CSP-boîte, $\mathcal{W} = \{w_1, \ldots, w_{n'}\}$ l'ensemble des variables impliquées dans les contraintes non monotones de $\mathcal{P}$. Quitte à réordonner les variables, on peut supposer

$$\mathcal{X} = \{w_1, \ldots, w_{n'}, x_{n'+1}, \ldots, x_n\}.$$

Rappelons que chaque élément de $\Omega = \{[-\infty, 0], [0, +\infty]\}^{n'} \times \{[-\infty, +\infty]\}^{n-n'}$ est alors une boîte monotone. On appelle **étiquette** un sous-ensemble de Ω. Une étiquette peut donc prendre $2^{(2^{n'})}$ valeurs possibles. C'est pour cette raison que l'algorithme naïf présenté ici n'est pas applicable. Supposons par exemple $\mathcal{X} = \{x, y, z\}$ et $\mathcal{W} = \{x, y\}$. Un exemple d'étiquette est

$$\{([-\infty, 0], [0, +\infty], [-\infty, +\infty]), ([0, +\infty], [-\infty, 0], [-\infty, +\infty])\},$$

ce qu'on notera de façon plus succincte : $\{(x^-, y^+), (x^+, y^-)\}$. Avec cette dernière notation, on pourra s'autoriser à permuter les variables (et donc, par exemple, à écrire $\{(y^+, x^-), (y^-, x^+)\}$). L'important est de pouvoir associer sans ambiguïté à toute boîte monotone B contenue dans une étiquette un domaine pour chaque variable. On notera d'ailleurs B_x le domaine associé à x dans B. Ainsi, avec le même exemple, si $B = (x^-, y^+)$, alors $B_x = [-\infty, 0]$, $B_y = [0, +\infty]$ et $B_z = [-\infty, +\infty]$. Il est possible de racourcir davantage les notations : l'étiquette $\{(x^-, y^+), (x^-, y^-)\}$ peut être notée simplement $\{x^-\}$. Lorsqu'une variable w de $\mathcal{W}$ n'apparaît pas dans une étiquette, cela signifie que toutes les combinaisons sont possibles avec w^- et w^+. En particulier, $\{x^-\}$ représente l'étiquette $\{x^-\} \times \prod_{w \in W, w \neq x} \{w^-, w^+\}$.

7.3.3 Calcul de la projection étiquetée (naïve)

Tout d'abord, rappelons que le calcul des projections se ramène pour des contraintes primitives à deux étapes simples : évaluation de la contrainte et intersection avec le domaine courant (cf. §6.10.3), l'étape de sélection étant inutile en l'absence de fonctions périodiques. Par exemple, la projection de la contrainte $y = \exp(x)$ sur y (resp. x) consiste à évaluer $\exp(x)$ sur le domaine de x (resp. $\ln(y)$ sur le domaine de y). Dans le cas d'une contrainte non monotone, par exemple $y = x^2$, la projection sur x se ramène à deux évaluations, celle de $\sqrt{y}$ et de $-\sqrt{y}$. Ceci nous permet, dans le schéma suivant, de nous limiter à quatre cas.

Rappelons également que *la projection d'une union est l'union des projections* (cf. prop 6.9 p.173) ; on peut donc se ramener au calcul de projections avec des intervalles uniquement. Les projections avec intervalles forment des

résultats intermédiaires, dont il faut ensuite calculer l'union. Nous montrons tout d'abord comment procéder pour un calcul intermédiaire, c.a.d. avec des opérandes intervalles. Nous noterons donc $\mathbf{x}$, $\mathbf{y}$ et $\mathbf{z}$ les intervalles des domaines de x, y et z avec lesquels se calcule la projection. Le résultat pour x est noté $\mathbf{x}$, s'il s'agit d'un intervalle, ou $\mathbf{x}^{(1)} \cup \mathbf{x}^{(2)}$ s'il s'agit d'une union. Enfin, si I est un intervalle, on désignera par I^* son étiquette.

A chaque projection, on calcule, en plus des domaines, les étiquettes grâce au schéma suivant :

Base :
Au départ, chaque intervalle possède pour étiquette Ω lui-même (c.a.d. l'ensemble des boîtes monotones).

Induction :

1. Projection sur x de $x = f(y)$ (avec f fonction élémentaire) ou sur x de $y = f^{-1}(x)$ (si f est monotone) :
   $\mathbf{x} \leftarrow \mathbf{x} \cap f(\mathbf{y})$.
   $\mathbf{x}^* \leftarrow \mathbf{y}^* \cap \mathbf{x}^*$.
2. Projection sur x de $y = x^2$ (la généralisation à $y = x^p$, avec p entier positif, est immédiate) :
   $\mathbf{x}^{(1)} \leftarrow -\sqrt{\mathbf{y}} \cap \mathbf{x}, \quad \mathbf{x}^{(2)} \leftarrow +\sqrt{\mathbf{y}} \cap \mathbf{x}$.
   ${\mathbf{x}^{(1)}}^* \leftarrow \mathbf{y}^* \cap \mathbf{x}^* \cap \{x^-\}$,
   ${\mathbf{x}^{(2)}}^* \leftarrow \mathbf{y}^* \cap \mathbf{x}^* \cap \{x^+\}$.
3. Projection sur x de $x = y \star z$ (ou $y = x \star^{-1} z$), avec $\star \in \{+, -, \times, /\}$, et $0 \notin \mathbf{z}$ si $\star = /$:
   $\mathbf{x} \leftarrow \mathbf{y} \star \mathbf{z}$,
   $\mathbf{x}^* \leftarrow \mathbf{y}^* \cap \mathbf{z}^* \cap \mathbf{x}^*$.
4. Projection sur x de $x = y/z$, avec $0 \in \mathbf{z}$. Elle est basée sur l'arithmétique d'intervalles étendue présentée au §6.9.5 p.170. Considérons le cas où la division $\mathbf{y}$ par $\mathbf{z}$ peut produire deux intervalles $[-\infty, \alpha]$ et $[\beta, +\infty]$ (les autres cas se déduisent facilement) :
   $\mathbf{x}^{(1)} \leftarrow [-\infty, \alpha] \cap \mathbf{x}, \quad \mathbf{x}^{(2)} \leftarrow [\beta, +\infty] \cap \mathbf{x}$,
   ${\mathbf{x}^{(1)}}^* \leftarrow \mathbf{y}^* \cap \mathbf{z}^* \cap \mathbf{x}^* \cap \{x^-\}$,
   ${\mathbf{x}^{(2)}}^* \leftarrow \mathbf{y}^* \cap \mathbf{z}^* \cap \mathbf{x}^* \cap \{x^+\}$.

La phase d'union consiste ensuite à fusionner les intervalles qui se chevauchent, c'est à dire à effectuer à la fois l'union de leurs domaines et de leurs étiquettes. Remarquons que dans le cas **1**, une fusion ne peut être due qu'à un problème d'arrondi[4].

7.3.4 Boucle principale

La boucle principale de l'algorithme est exactement celle d'`AC3` (ou plutôt son équivalent en domaine continu, c'est à dire σ–`AC3`), sauf que l'opérateur de projection effectue un travail plus complexe qu'une simple réduction des domaines : il calcule en plus l'étiquette de chaque intervalle (nous verrons dans la section suivante comment le faire de façon efficace) et supprime au fur et à mesure les intervalles ayant des étiquettes vides. Cette suppression est donc synchronisée avec la projection.

Dans la section 7.5, on prouve que cet algorithme est correct, i.e, qu'il calcule bien la clôture AC-IGC. La preuve s'articule autour de deux propositions :
Correction : La proposition 7.7 énonce qu'un intervalle dont l'étiquette est l'ensemble vide n'appartient à aucune I–clique (et peut donc être éliminé).
Complétude : La proposition 7.9 énonce qu'un intervalle ayant une étiquette autre que l'ensemble vide appartient bien à une I–clique.
La proposition intermédiaire 7.8 sert pour la preuve de la proposition 7.9, et constitue aussi la clef de voûte de l'autre version de cet algorithme, `Etiq-AC`.

[4]Puisqu'on effectue en réalité une projection approximative en élargissant les bornes des intervalles aux valeurs de la subdivision σ choisie, si deux intervalles deviennent contigus, on doit également les fusionner.

7.4 Algorithme `Etiq-AC`

L'algorithme proposé consiste à calculer les étiquettes d'une façon optimisée, grâce à une structure de données adaptée.

7.4.1 Structure de données

Supposons que notre système comporte trois variables impliquées dans des contraintes non monotones, w_1, w_2 et w_3. Avec les notations du §7.3.2, nous aurons donc

$$\Omega = \{(w_1^-, w_2^-, w_3^-), (w_1^-, w_2^-, w_3^+), \ldots, (w_1^+, w_2^+, w_3^+)\}$$

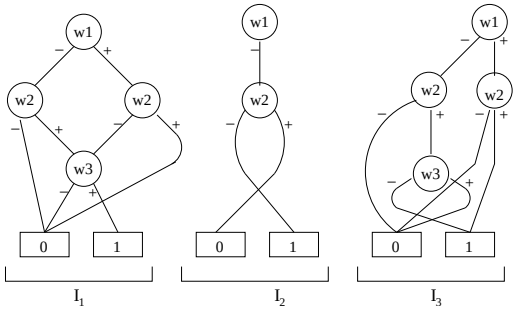
et rappelons qu'une étiquette est un sous-ensemble de Ω.

Une manière efficace de représenter un sous-ensemble du produit cartésien de n paires est le ROBDD [Bry92] (*reduced ordered binary decision diagram*), que nous nommerons simplement BDD.

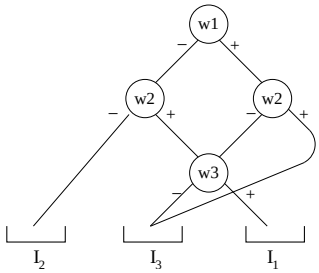
Mettons par exemple que le domaine d'une variable possède trois intervalles, I_1, I_2 et I_3, avec les étiquettes suivantes :

$$\begin{array}{lcl} {I_1}^* & = & \{(w_1^-, w_2^+, w_3^+), (w_1^+, w_2^-, w_3^+)\}, \\ {I_2}^* & = & \{(w_1^-, w_2^-, w_3^-), (w_1^-, w_2^-, w_3^+)\}, \\ {I_3}^* & = & \{(w_1^-, w_2^+, w_3^-), (w_1^+, w_2^+, w_3^-), (w_1^+, w_2^+, w_3^+)\}. \end{array}$$

Avec un ordre fixé entre les variables (par exemple $w_1 \prec w_2 \prec w_3$), chaque étiquette peut être représentée par un BDD unique, comme indiqué sur cette figure :



Cette représentation peut être facilement optimisée. Tout d'abord, on peut omettre de stocker pour chaque étiquette les chemins menant à l'état "0", qui s'obtiennent automatiquement par complémentarité. Ensuite, en s'appuyant sur la proposition 7.8, on voit que si un chemin dans le BDD d'un intervalle mène à l'état "1", il mène forcement à l'état "0" dans le BDD des autres intervalles. On peut donc utiliser un seul BDD par variable et stocker aux feuilles du BDD l'intervalle correspondant, comme illustré ci-dessous :



7.4.2 Nouvelle projection étiquetée

Un BDD peut se voir comme la représentation d'une fonction f de $\{0,1\}^n$ dans $\{0,1\}$. Une opération logique (mettons $\vee$) entre deux BDDs f_1 et f_2 peut être réalisée grâce à une fonction générique `APPLY` [Bry92] qui produit le BDD représentant la fonction $f_1 \vee f_2$.

La fonction `APPLY` est basée sur le principe suivant :
- Phase descendante : parcourir de façon synchronisée le DAG (*directed acyclic graph* - graphe orienté acyclique) de f_1 et f_2 en générant les combinaisons de valeurs possibles pouvant mener à des feuilles de $f_1 \vee f_2$.
- Calcul des feuilles : lorsqu'une combinaison notée $w_1, \ldots, w_n$ aboutit à une feuille de $f_1 \vee f_2$, la valeur de cette feuille (0 ou 1) se calcule en appliquant l'opération logique entre les feuilles correspondantes de f_1 et f_2, c.a.d.
$$(f_1 \vee f_2)(w_1, \ldots, w_n) := f_1(w_1, \ldots, w_n) \vee f_2(w_1, \ldots, w_n).$$
- Phase ascendante : le DAG est réduit de telle sorte que chaque nœud ait des sous-DAG différents (*non-redundant test*), et que pour une variable donnée, deux nœuds étiquetés par cette variable n'aient pas à la fois même sous-DAG gauche et même sous-DAG droit (*uniqueness*).

Par souci de clarté, nous scindons le nouveau calcul d'une projection étiquetée en deux : évaluation et projection. Comme dans le cas de l'algorithme naïf, ces deux phases sont en fait simultanées.

Phase d'évaluation (cas monotone)

Plaçons-nous d'abord dans le cas **3** (cf. l'algorithme du §7.3.3), celui de la projection sur x de $x = y \star z$. La fonction `APPLY` peut être facilement étendue pour des fonctions de $\{0,1\}^n$ dans n'importe quel ensemble, et ce pour n'importe quel type d'opération. Notre BDD servant à représenter le domaine d'une variable, dans sa version optimisée, est une fonction de $\{0,1\}^n$ dans $\mathbb{IR}$ (l'ensemble des intervalles), et on peut appliquer une opération arithmétique entre deux de ces fonctions. Le calcul des feuilles devient simplement :

$$(f_1 \star f_2)(w_1, \ldots, w_n) := f_1(w_1, \ldots, w_n) \star f_2(w_1, \ldots, w_n).$$

L'algorithme se déroule de la même manière, avec toutefois la nuance suivante : dans la version originale d'`APPLY`, pour réduire le DAG dans la phase ascendante, on est amené à comparer des nœuds et des feuilles. La comparaison entre nœuds utilise une égalité de pointeurs, et la comparaison entre feuilles une égalité de valeurs (0 ou 1). Dans notre version étendue d'`APPLY`, la comparaison entre deux feuilles I et I' utilise la relation $I \cap I' \neq \emptyset$. Lorsque deux feuilles sont "égales" (lorsqu'elles vérifient la relation), elles sont donc fusionnées, et l'opération de fusion consiste à calculer l'enveloppe $I \cup I'$. Par exemple, dans la figure 7.8, la combinaison w_1^-, w_2^+, w_3^- donne l'intervalle $[1,2]+[-3,-2]$, soit $[-2,0]$. La combinaison w_1^+, w_2^+, w_3^+ donne l'intervalle $[-2,-1]+[2,3]$, soit $[0,2]$. Ces deux feuilles sont fusionnées en un seul intervalle $[-2,2]$.

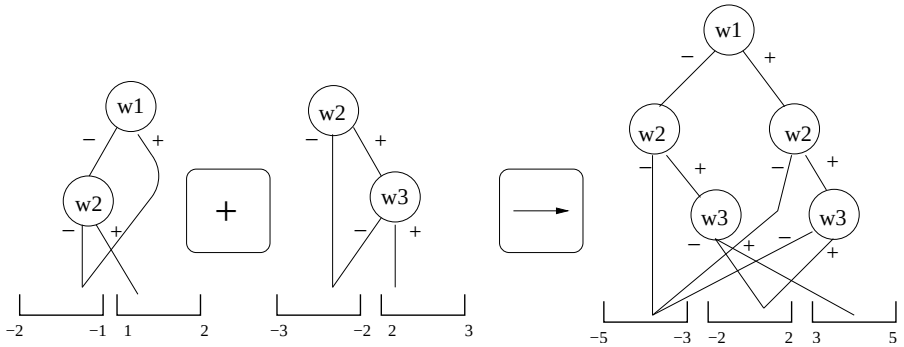


FIG. *7.8: Nouvel opérateur Somme*

Ce n'est donc plus seulement une somme que l'on calcule, mais bien la phase d'évaluation de la projection étiquetée décrite au §7.3.3.

Le cas **1** de la projection, celui de $x = f(y)$, se déduit facilement de notre discussion. Il correspond à une opération unaire effectuée sur le BDD de x, obtenue comme pour un `not` sur un BDD ordinaire.

Phase d'évaluation (cas non-monotone)

Étudions maintenant les cas **2** et **4** de la phase d'évaluation du calcul de projection. Nous continuons à noter x la variable sur laquelle la contrainte est projetée.

Lors du parcours du DAG de y (et de z dans le cas **4**), l'idée est d'introduire artificiellement la variable x dans les combinaisons en entrée, sauf si celle-ci est déjà présente.

On crée donc une sous-branche x^- et une sous-branche x^+ dans chaque branche du DAG. Dans la sous-branche x^- (resp. x^+), seules les projections négatives, $-\sqrt{\cdot}$ ou $[-\infty, \alpha]$ (resp. $+\sqrt{\cdot}$ ou $[\beta, +\infty]$) seront autorisées aux feuilles. La figure 7.9 montre un exemple pour la contrainte $y = x^2$. On suppose dans l'ordre des variables que $w_2 \prec x$. Le BDD de gauche est celui de y, celui de droite le résultat de la phase d'évaluation. On remarque que x n'apparaît pas dans le sous-DAG w_1^-, w_2^- car les feuilles $[-1, 0]$ (pour $-\sqrt{[0,1]}$) et $[0, 1]$ (pour $+\sqrt{[0,1]}$) ont été fusionnées.

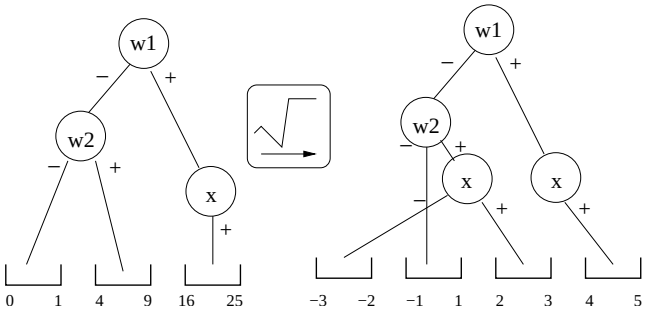


FIG. *7.9: Projection sur x de $y = x^2$*

On peut vérifier une fois de plus que cela coïncide avec la projection étiquetée du §7.3.3.

Phase d'intersection

La phase d'intersection nécessite d'appliquer une opération d'intersection entre le BDD issu de la phase d'évaluation (mettons $y \star z$) et le BDD de la variable courante, x. Mais plutôt que de procéder en deux étapes (évaluation *puis* intersection), on peut encore étendre la fonction `APPLY` à des opérations ternaires. Cette fois, on parcourt de façon synchronisée à la fois le BDD des variables-paramètres et le BDD de x, pour calculer aux feuilles l'opération :

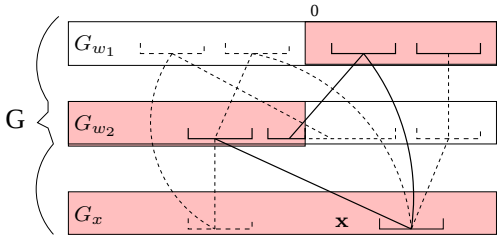
$$(f_x \cap (f_y \star f_z))(w_1, \ldots, w_n) := f_x(w_1, \ldots, w_n) \cap \Big(f_y(w_1, \ldots, w_n) \star f_z(w_1, \ldots, w_n)\Big).$$

On anticipe ainsi sur le fait que les intersections des étiquettes seront calculées au final, en combinant uniquement des intervalles pour lesquels le résultat de cette intersection sera non vide.

Ce triple parcours récursif nous mène donc à effectuer une projection d'intervalles cette fois (et non plus d'unions), avec une feuille pour x, y et z. Si l'intervalle obtenu s'avère être vide, le BDD-résultat sera réduit en conséquence à la remontée, de telle sorte qu'il sera bien minimal une fois la projection terminée.
Il remplace ensuite simplement le BDD courant de x.

7.5 Résultats théoriques

Revenons à un CSP-gruyère obtenu à une étape quelconque de l'algorithme `EtiqAc` (c'est à dire σ–`AC3` enrichi du calcul des étiquettes à l'aide des BDD). Soit x une variable, $\mathbf{x}$ un intervalle de son domaine. Nous allons montrer que si une boîte monotone B n'appartient pas à l'étiquette de $\mathbf{x}$, alors le gruyère G' obtenu en remplaçant dans $G \cap B$ le domaine de x par $B_x \cap \mathbf{x}$ (on écrira $G' := (G \cap B)_{x \leftarrow \mathbf{x}}$) ne contient pas de I–clique. Ceci est illustré sur la figure ci-dessous. On suppose que B est la boîte monotone (w_1^+, w_2^-), et que $\mathbf{x}^*$ ne contient pas B. La figure montre qu'on ne peut pas former de I–clique avec les arêtes en gras, c.a.d. celles dont les extrémités sont dans $G \cap B$ (en gris).



Proposition 7.7 *Soient $(\mathcal{C}, \mathcal{X}, G)$ un CSP-gruyère étiqueté, $x \in \mathcal{X}$, $\mathbf{x}$ un intervalle de G_x, et $B \in \Omega$.*

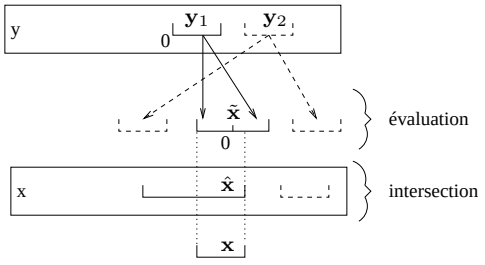
$$B \notin \mathbf{x}^* \Longrightarrow (G \cap B)_{x \leftarrow \mathbf{x}} \textit{ ne contient pas de I–clique.}$$

Preuve. *Base :* L'implication est vraie au départ puisque les étiquettes contiennent toutes les boîtes monotones.

Induction : Il suffit de montrer que l'opération de projection conserve cette propriété. Soit $\mathbf{x}$ un intervalle obtenu pour x, la variable sur laquelle on projette. $\mathbf{x}$ résulte de l'intersection d'un intervalle $\hat{\mathbf{x}}$ de l'ancien domaine de x, et de l'union $\tilde{\mathbf{x}}$ de plusieurs intervalles (mettons deux) issus de la phase d'évaluation.

Considérons $B \in \Omega$ tel que $B \notin \mathbf{x}^*$. Soit $B \notin \hat{\mathbf{x}}^*$, soit $B \notin \tilde{\mathbf{x}}^*$. Si $B \notin \hat{\mathbf{x}}^*$, alors la boîte $(G \cap B)_{x \leftarrow \hat{\mathbf{x}}}$ ne contient pas de I–clique (par hypothèse), et il en est de même pour toute sous–boîte. En particulier, $(G \cap B)_{x \leftarrow \mathbf{x}}$ ne contient pas de I–clique. Supposons $B \notin \tilde{\mathbf{x}}^*$. Par l'absurde, supposons qu'il existe une I–clique dans $(G \cap B)_{x \leftarrow \mathbf{x}}$, et observons les quatre cas de projection possibles :

1. $\tilde{\mathbf{x}} = f(\mathbf{y}_1) \cup f(\mathbf{y}_2)$. Les seuls I-supports de $\mathbf{x}$ sont $\mathbf{y}_1$ et $\mathbf{y}_2$ donc la I–clique prend ses valeurs pour y soit dans $\mathbf{y}_1$, soit dans $\mathbf{y}_2$, et est donc incluse soit dans $(G \cap B)_{y \leftarrow \mathbf{y}_1}$, soit dans $(G \cap B)_{y \leftarrow \mathbf{y}_2}$. Donc $B \in {\mathbf{y}_1}^*$ ou $B \in {\mathbf{y}_2}^*$, ce qui implique $B \in \tilde{\mathbf{x}}^*$, contradiction.
2. i) Soit $\mathbf{y}_1$ tel que $-\sqrt{\mathbf{y}_1} \subseteq \tilde{\mathbf{x}}$ et $+\sqrt{\mathbf{y}_1} \subseteq \tilde{\mathbf{x}}$. Si la I–clique prenait ses valeurs dans $\mathbf{y}_1$ alors $B \in {\mathbf{y}_1}^* \Rightarrow B \in \tilde{\mathbf{x}}^*$, contradiction (voir figure ci-dessous). ii) Soit $\mathbf{y}_2$ tel que seul $+\sqrt{\mathbf{y}_2} \subseteq \tilde{\mathbf{x}}$. Si $B_x = [-\infty, 0]$, la I–clique ne peut pas prendre ses valeurs dans $\mathbf{y}_2$ (puisque $(G \cap B)_{x \leftarrow \mathbf{x}} \cap -\sqrt{\mathbf{y}_2} = \emptyset$). Si $B_x = [0, +\infty]$ alors $B \in {\mathbf{y}_2}^* \Rightarrow B \in \tilde{\mathbf{x}}^*$, contradiction. Même raisonnement dans le cas négatif.
3. Si la I–clique prend ses valeurs dans $\mathbf{y}_1$ pour y et dans $\mathbf{z}_1$ pour z, alors $B \in {\mathbf{y}_1}^*$ et $B \in {\mathbf{z}_1}^*$, donc $B \in {\mathbf{y}_1}^* \cap {\mathbf{z}_1}^*$, ce qui implique $B \in \tilde{\mathbf{x}}^*$, contradiction.
4. Même idée en combinant les cas 2 et 3.



□

Proposition 7.8 *Soient $(\mathcal{C}, \mathcal{X}, G)$ un CSP-gruyère étiqueté, $x \in \mathcal{X}$.*

$$\big(\forall \mathbf{x}_1 \in G_x\big)\big(\forall \mathbf{x}_2 \in G_x\big) \quad \mathbf{x}_1 \neq \mathbf{x}_2 \Longleftrightarrow {\mathbf{x}_1}^* \cap {\mathbf{x}_2}^* = \emptyset,$$

c.a.d., les étiquettes des intervalles d'une variable sont deux à deux disjointes.

Preuve. Toujours par induction.
Base : La propriété est vraie au départ puisque le domaine de chaque variable ne comporte qu'un seul intervalle.

Induction : Considérons deux intervalles disjoints $\mathbf{x}_1$ et $\mathbf{x}_2$ dans le nouveau domaine de x. On suppose $\mathbf{x}_1 < \mathbf{x}_2$. S'il existait auparavant dans le domaine de x deux intervalles disjoints $\hat{\mathbf{x}}_1$ et $\hat{\mathbf{x}}_2$ tels que $\mathbf{x}_1 \subseteq \hat{\mathbf{x}}_1$ et $\mathbf{x}_2 \subseteq \hat{\mathbf{x}}_2$ alors $({\mathbf{x}_1}^* \cap {\mathbf{x}_2}^*) \subseteq (\hat{\mathbf{x}}_1^* \cap \hat{\mathbf{x}}_2^*) = \emptyset$. Sinon, observons de nouveau les quatre cas de projection possibles :

1. $\mathbf{x}_1$ et $\mathbf{x}_2$ proviennent forcément de la projection d'intervalles différents pour y, formant dans la phase d'évaluation deux intervalles disjoints $\mathbf{y}_1$ et $\mathbf{y}_2$. Les intervalles de y pris deux à deux ont tous au moins une étiquette disjointe, par hypothèse d'induction, donc $({\mathbf{x}_1}^* \cap {\mathbf{x}_2}^*) \subseteq ({\mathbf{y}_1}^* \cap {\mathbf{y}_2}^*) = \emptyset$
2. Si $\mathbf{x}_1 < 0 < \mathbf{x}_2$, alors ${\mathbf{x}_1}^*$ ne contient aucune boîte monotone B telle que $B_x = x^-$ et ${\mathbf{x}_2}^*$ ne contient aucune boîte B telle que $B_x = x^+$ donc ${\mathbf{x}_1}^* \cap {\mathbf{x}_2}^* = \emptyset$. Sinon, $\mathbf{x}_1$ et $\mathbf{x}_2$ proviennent d'intervalles différents de y et on raisonne comme au cas **1**.
3. Idem que pour 1. Il suffit d'isoler deux intervalles disjoints pour y ou pour z, et les étiquettes de $\mathbf{x}_1$ et $\mathbf{x}_2$ hériteront de la propriété voulue.
4. Idem en combinant les cas **2** et **3**.

□

Nous pouvons maintenant montrer facilement que, pour un CSP arc–cohérent, la proposition 7.7 admet une réciproque. Il s'agit toutefois de prendre la précaution suivante : comme il est possible qu'une projection ne modifie pas le domaine d'une variable, mais modifie son étiquette, lorsque le point fixe de la boucle principale d'`Etiq-AC` (cf. §7.3.4) est atteint, il n'est pas garanti que les étiquettes soient "stables". On prolonge donc artificiellement les projections (sans incidence sur le gruyère) jusqu'à ce que les étiquettes également aient atteint leur point fixe (ce qui arrive forcément après un nombre fini d'itérations, puisqu'elles ne font que décroître).

Exemple 7.10 *Soit $(\{c_1, \ldots, c_5\}, \{w_1, w_2, x, y\}, \{G_{w_1} \times G_{w_2} \times G_x \times G_y\})$ un CSP, avec $G_{w_1} = [-2, 2]$, $G_{w_2} = [-2, 2]$, $G_x = [-1, 1]$, $G_y = [-1, 1]$, et*

$$\begin{array}{ll} (c_1) & w_1^2 \in [1, 4], \\ (c_2) & w_2^2 \in [1, 4], \\ (c_3) & x = y, \\ (c_4) & x = w_1 - w_2, \\ (c_5) & y = w_1 + w_2. \end{array}$$

Si les contraintes sont empilées dans leur ordre de déclaration, les deux premières réduisent G_{w_1} et G_{w_2} à $[-2, -1] \cup [1, 2]$. Les trois contraintes suivantes ne modifient pas les domaines. Si le point fixe s'arrête là, IGC n'est pas atteint (il n'y a pas de I–clique sur la figure 7.10).

Proposition 7.9 *Soient G la clôture par arc–cohérence étiquetée (et "stable") d'un CSP-boîte, x une variable et $\mathbf{x}$ un intervalle de G_x.*

$$B \in \mathbf{x}^* \Longleftrightarrow (G \cap B)_{x \leftarrow \mathbf{x}} \text{ contient une I–clique.}$$

Preuve. Il reste à montrer ($\Longrightarrow$). Soit B une boîte monotone. Si $B \in \mathbf{x}^*$, il existe un intervalle compatible avec $\mathbf{x}$ dans le domaine de chaque variable (car l'arc cohérence est atteinte) dont l'étiquette contienne B (puisque les

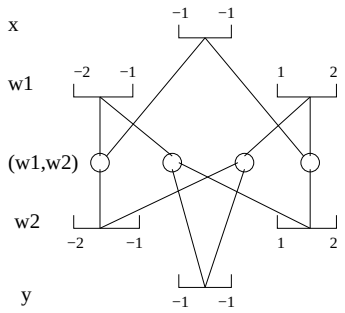


FIG. *7.10: Problème du point fixe*

étiquettes sont stables), et un seul (d'après la proposition 7.8). On obtient un n-uplet d'intervalles compatibles avec $\mathbf{x}$, mais ce même n-uplet est aussi compatible avec ses autres intervalles, en répétant le même argument. Il s'agit donc bien d'une I–clique, et cette I–clique s'intersecte avec B (ce qu'on montre facilement par induction). □

De part l'unicité de la clôture par (σ-)arc–cohérence d'un CSP, et la convergence vers ce gruyère de l'algorithme `AC3`, on peut donc associer à tout CSP-boîte un gruyère représentant la clôture par arc–cohérence de ce CSP où les étiquettes sont calculées pour chaque intervalle du gruyère, et ce, de telle sorte que ces étiquettes soient "stables". Nous pouvons alors énoncer la proposition suivante, qui est une conséquence immédiate des propositions 7.8 et 7.9 :

Proposition 7.10 *Soient $\mathcal{P} = (\mathcal{C}, \mathcal{X}, B)$ un CSP-boîte, $\mathcal{P}' = (\mathcal{C}, \mathcal{X}, G)$ la clôture par arc–cohérence de $\mathcal{P}$.*

$$\mathcal{P}' \text{ est AC-IGC} \iff \text{tout intervalle de } G \text{ possède une étiquette non vide.}$$

7.5.1 Complexité

La proposition 7.10 admet le corollaire suivant :

Corollaire 7.1 (Complexité en espace) *Soit $\mathcal{P}$ un CSP-boîte ayant n' variables impliquées dans des contraintes non-monotones. Soit σ un ensemble de flottants et $|\sigma|$ le nombre maximal de flottants nécessaire pour recouvrir la boîte initiale de $\mathcal{P}$ (sur chaque dimension). Si G est la clôture σ–AC–IGC de $\mathcal{P}$, alors le nombre d'intervalles pour chaque union de G est borné par $2^{n'}$ et par $|\sigma|$.*

Très souvent en pratique, la clôture AC–IGC est un gruyère nettement plus petit que la clôture AC. Il existe des contre-exemples :

Exemple 7.11 *Soit le CSP $(\{c_1, \ldots, c_m\}, \{x_1, \ldots, x_m\}, [-2, 2]^m)$ avec*

$$\begin{array}{ll} (c_1) & x_1^2 = 1, \\ (c_2) & (x_2 - x_1)^2 = 1/2, \\ (c_3) & (x_3 - x_2)^2 = 1/4, \\ \vdots & \vdots \\ (c_m) & (x_m - x_{m-1})^2 = 1/(2^{m-1}). \end{array}$$

Supposons que σ soit une subdivision uniforme de pas $1/2^m$ de $[-2,2]$. Dans ce problème, on voit alors que la clôture σ–AC-IGC se projette en $2^m = 2^{n'}$ intervalles disjoints sur x_m. Ainsi, dans la clôture σ–AC–IGC, le nombre d'intervalles peut bien atteindre d'une part la borne $2^{n'}$ et, d'autre part, la borne $|\sigma|$.

Il apparaît, au travers de l'exemple précédent, qu'un algorithme qui calcule la clôture σ–AC–IGC est, en temps, au pire exponentiel (en n'). Néanmoins, nous avons pris dans cet exemple σ de telle sorte que $|\sigma| \geq 2^{n'}$. Or, puisque la complexité en espace est aussi bornée par $|\sigma|$, on pourrait espérer trouver un algorithme calculant la clôture σ–AC–IGC qui soit également borné, en temps, par un polynôme en $|\sigma|$. En utilisant un découpage assez "grossier", on éviterait ainsi l'explosion combinatoire qui se produit dans le cas où $2^{n'} \gg \sigma$. Autrement dit, nous disposerions d'un algorithme qui vérifierait deux critères d'efficacité :
- Théorique : avoir une complexité en temps et en espace polynomiale en $|\sigma|$ (et donc ne jamais présenter de performances moins bonnes que le filtrage par σ–arc–cohérence).
- Pratique : ne produire qu'un petit nombre d'intervalles en calculant la partie σ–AC–IGC.

La proposition suivante montre que ce n'est pas possible. Cela signifie que la taille d'un BDD peut être exponentielle en n' même si $2^{n'} > |\sigma|$; seul le nombre de feuilles est borné par $|\sigma|$.

Proposition 7.11 *Soit un CSP-boîte B et σ une subdivision des réels. En admettant que $P \neq NP$, il n'existe pas d'algorithme polynomial pour calculer la clôture σ–AC–IGC de B.*

Preuve. Le preuve repose sur une relation entre le problème 3–SAT [GJ79] et le problème suivant :

P_1 : *Langage :* soit $\mathcal{P}$ un CSP-boîte et σ une subdivision ;
Question : existe-t-il une I-clique dans la clôture σ–AC de $\mathcal{P}$?

Nous allons prouver que le problème P_1 est NP-complet. On en déduit alors immédiatement la proposition : en effet, par contradiction, s'il existait un algorithme polynomial pour calculer la clôture σ-AC-IG, alors cet algorithme fournirait également une réponse à P_1 en temps polynomial, ce qui est impossible, puisque ce problème est NP–complet.

Pour prouver que P_1 est NP–complet, nous montrons que 3-SAT peut se réduire polynomialement vers le problème P_1. La réduction polynomiale est la suivante :

Soit $F = (\mathcal{B}, \mathcal{C})$ une instance de 3-SAT ; $\mathcal{B}$ est l'ensemble des variables booléennes et $\mathcal{C}$ est l'ensemble des clauses. On note $\mathcal{P} = (\mathcal{C}', \mathcal{X}, B)$ le CSP-boîte issu de la transformation polynomiale, et σ la subdivision des réels associée à $\mathcal{P}$ (également issue de cette transformation).

Pour chaque variable booléenne $b \in \mathcal{B}$, on crée une variable $x_b \in \mathcal{X}$ avec pour domaine $[-3,3]$. On choisit la subdivision σ égale à $\{-3,-2,-1,0,1,2,3\}$. On crée également une contrainte $x_b^2 = 5$ dans $\mathcal{C}$ (de telle sorte que seulement deux intervalles soient possibles pour x_b dans la clôture AC de $\mathcal{P}$: $[-3,-2]$ et $[2,3]$).

Chaque clause c est une disjonction de 3 littéraux l_1, l_2 et l_3, impliquant resp. les variables b_1, b_2 et b_3. Elle produit une contrainte $\pm x_{b_1} + \pm x_{b_2} \pm x_{b_3} \geq -5$ dans $\mathcal{C}'$. Le signe de x_{b_i} est positif (resp. négatif) si le littéral l_i est b_i (resp. sa négation). L'intuition derrière cette réduction est la suivante : si la valuation d'une variable booléenne explique la satisfiabilité d'une clause donnée, on sélectionne l'intervalle de même "signe" pour la variable correspondante dans la contrainte ternaire associée : $[2,3]$ si la variable est à *vrai* et $[-3,-2]$ si elle est à *faux*. Cela assure qu'un des termes de la contrainte ternaire est $[2,3]$, si bien qu'elle est satisfaite (et σ–AC). En effet, si l'un des trois termes est $[2,3]$, alors la contrainte est nécessairement satisfaite, indépendamment du domaine des deux autres variables ($[-3,-2]$, $[2,3]$ ou $[-3,-2] \cup [2,3]$), le total dépassant -5. De plus, un filtrage σ–AC de cette contrainte ne réduit (a fortiori supprime) aucun des intervalles des domaines (l'inégalité ne filtre pas).

$P_1 \in NP$.
Calculer la clôture σ–AC de $\mathcal{P}$ est en $O(m)$ (m est le nombre de contraintes $x_b^2 = 5$). Il suffit en effet d'effectuer

toutes les projections $x_b^2 = 5$. Une fois l'union $[-3,-2] \cup [2,3]$ associée à chaque variable, le gruyère est arc-cohérent. Vérifier alors qu'un ensemble de n intervalles donnés forment une I–clique se fait en $O(n)$. La réponse à la question de P_1 se teste en temps polynomial, montrant l'appartenance de P_1 à la classe NP.

Il nous reste à démontrer l'équivalence entre (i) une solution S pour n'importe quelle instance F de 3-SAT et (ii) une solution S' pour l'instance $\mathcal{P}$ du problème P_1 obtenue par la réduction présentée ci-dessus.

$(i) \rightarrow (ii)$. Pour construire la solution S' de $\mathcal{P}$, on sélectionne, dans la contrainte ternaire correspondante, l'intervalle (positif ou négatif) selon la valuation des variables booléennes. L'intuition de la réduction montre directement le résultat. La contrainte $l_x^2 = 5$ de $\mathcal{P}$ restreint le domaine de l_x à $[-3,-2] \cup [2,3]$. Les contraintes ternaires sont σ–AC car l'intervalle sélectionné produit un terme $[2,3]$ dans la contrainte et l'inégalité fait que l'on ne peut pas réduire ou supprimer des intervalles : G est donc σ–AC. On voit que l'ensemble des intervalles ainsi sélectionnés forment une I–clique : cette I–clique passe par un intervalle de chaque variable et les intervalles sont compatibles entre eux. Finalement, le CSP est σ–AC–IGC.

$(ii) \rightarrow (i)$. Pour construire la solution S de F, on considère les intervalles sélectionnés dans la I–clique de S'. On effectue l'opération duale de l'autre sens de la démonstration : si l'intervalle est $[-3,-2]$ (resp. $[2,3]$), on value la variable booléenne correspondante à *faux* (resp. *vrai*) dans S. Comme la I–clique "satisfait" chaque contrainte ternaire, les valuations correspondantes dans F satisfont chacune des clauses, par construction. □

7.5.2 Autres I-cohérences

Au vu des résultats sur la complexité du paragraphe précédent, nous sommes confrontés à deux choix possibles :
- Soit calculer σ–AC comme nous l'avons fait au chapitre précédent, avoir une complexité polynomiale (en $O(|\sigma|)$), mais en pratique avoir à gérer un nombre d'intervalles de l'ordre de $|\sigma|$, ce qui est inenvisageable avec $|\sigma|$ très grand.
- Soit calculer σ–AC–IGC, avec un petit nombre d'intervalles en pratique, mais une complexité au pire exponentielle (en $O(2^{n'})$).

Dans cette section, nous avons montré une façon de calculer la clôture σ-AC-IGC d'un CSP. Mais il serait de bon aloi de s'assurer qu'il n'existe pas un filtrage intermédiaire entre σ–AC et σ–AC–IGC qui cumulerait l'avantage des deux critères. Ce filtrage pourrait par exemple consister à maintenir la présence de I-cliques, mais uniquement pour des sous-graphes de contraintes de taille k. Plutôt que de résoudre un problème NP-difficile pour effectuer un filtrage, on resterait alors sur le plan théorique dans un cadre polynomial, tout en évitant sur le plan pratique d' "atteindre les flottants".

Malheureusement, cela semble peu probable. Des contre-exemples ont pu être isolés, lorsqu'on applique un filtrage au niveau "intervalles" assez fort tel que PIC ou Max–RPC [DB97a].

7.6 Conclusion

Dans ce chapitre deux nouvelles cohérences ont été présentées pour pouvoir approcher l'arc-cohérence avec des CSP continus.

La première est la box–set cohérence, qui permet d'isoler les sous-boîtes arc-cohérentes d'un CSP. Appliquer la box-set cohérence nécessite l'introduction de points de choix au cours du filtrage, mais l'aspect combinatoire de ce calcul est compensé par la possibilité de l'intégrer dans une recherche de solutions sans en impacter les performances (les découpages naturels s'ajoutent aux bissections). Dans ce contexte, l'obtention de la box-set cohérence représente en effet un faible coût grâce au filtrage par 2B-cohérence qui permet de minimiser le nombre d'appels à la seule opération véritablement coûteuse, c'est à dire, la projection avec unions d'intervalles. Notre contribution principale est d'avoir détaillé les propriétés de cette méthode "paresseuse", notamment lorsque le

calcul de projections est basé sur l'algorithme `HC4Revise`. Nous avons montré qu'en dépit des approximations liées à cet algorithme (projections avec $\texttt{HC4Revise}_{\square}$), la box-set cohérence était bien calculée pour des contraintes ne comportant pas d'occurrence multiple de variable.

La seconde est la I-cohérence globale qui, contrairement à la box-set cohérence, permet d'approcher l'arc-cohérence d'un CSP sans faire de point de choix. L'idée principale est de maintenir l'existence de cliques dans le graphe représentant les compatibilités entre les différents intervalles des unions servant à représenter les domaines. Nous avons donné un algorithme, `Etiq-AC`, qui utilise une structure de données sophistiquée pour pouvoir en pratique appliquer cette cohérence. A ce jour, un premier prototype d'`Etiq-AC` a été implanté, avec des algorithmes naïfs (notamment pour la fonction `APPLY`). Elle a permis de vérifier ce qui est avancé dans ce chapitre : le filtrage AC–IGC s'obtient en pratique avec très peu d'intervalles, notamment là où l'arc–cohérence échoue. Les cas pathologiques semblent très artificiels. Notons que nous avons limité notre étude aux contraintes primitives, mais notre programme s'applique à un CSP quelconque (toutefois sans fonction trigonométrique) en prenant en compte les variables implicites dans les étiquettes. Il semble a priori que la clôture AC–IGC du renommage du CSP est obtenue.

Si l'analyse théorique est bien avancée, une analyse expérimentale approfondie reste à mener pour comparer la box-set cohérence, AC-IGC et les cohérences classiques. Nous ne reportons pas ici les résultats expérimentaux obtenus qui ne sont pas aboutis. Cependant, ceux-ci nous permettent de dégager quelques enseignements préliminaires. Les points positifs sont les suivants :
- `Lazy-box-set` en version *profondeur d'abord* introduit un surcoût négligeable par rapport à un schéma de résolution classique 2B+bissection. Les résultats sont souvent comparables et parfois meilleurs.
- `Etiq-AC` perd un facteur constant par rapport à la 2B–cohérence. Ce facteur pourrait être réduit en s'appuyant sur une bibliothèque de gestion de BDD optimisée. Nous disposerions ainsi d'un filtrage par arc-cohérence comme alternative possible à la 2B–cohérence.

En contrepartie :
- La stratégie de recherche `lazy-box-set` est rarement meilleure que 2B+bissection.
- Malgré son absence de point de choix, `Etiq-AC` n'est pas compétitif avec `Lazy-box-set`. Le manque d'optimisation est une première raison. La deuxième raison est finalement la faible explosion combinatoire des bissections naturelles avec `lazy-box-set`. En effet, sur les benchmarks que nous avons testés, le nombre de bissections naturelles reste assez réduit. Il resterait l'espoir de voir `Etiq-AC` compétitif sur des instances de taille plus importante, mais nous croyons peu à cette éventualité (nous y reviendrons dans la conclusion).

Nous exposerons dans la conclusion de cette thèse notre interprétation de ces résultats pratiques mitigés.

Chapitre 8

Conclusion

Nous avons présenté dans cette thèse quelques contributions sur les techniques d'intervalles, avec une volonté sous-jacente de rendre ce document autonome.

En analyse par intervalles, nous nous sommes penché sur les *AE-solution sets*, c'est à dire, les systèmes avec paramètres quantifiés. La résolution de tels systèmes implique trois types d'opérations : filtrage, test de boîte intérieure et bissection. Notre contribution porte essentiellement sur le filtrage. Nous avons proposé notamment une extension de l'algorithme de Hansen-Bliek pour l'approximation extérieure d'AE-solution sets linéaires quantifiés « à droite ». Bien que d'apparence restrictif, cet algorithme sert de brique pour l'opérateur de *Newton généralisé* qui permet de filtrer l'espace de recherche pour n'importe quel AE-solution set non linéaire.

Le développement de tests de boîte intérieure et de techniques de bissection fait partie de notre travail en cours ou à venir. Dans notre état de l'art, le test le plus efficace que nous avons exposé est sans doute celui de permutation variable-paramètre (cf. §2.9.5), dans la mesure où il autorise les occurrences multiples de paramètres. En revanche, ce test n'est applicable qu'à des systèmes où le nombre d'équations coïncide avec le nombre de paramètres existentiels. L'un de nos objectifs futurs est de nous affranchir de cette condition en exploitant d'autres types de tests d'existence de solutions que celui de Newton, limité aux systèmes carrés. De tels tests existent déjà dans le cas linéaire. Que ce soit pour le filtrage ou la détection de boîtes intérieures, nous avons également vu au chapitre 3 que la résolution d'AE-solution sets nécessitait de faire des points de choix à la fois sur les variables et sur les paramètres pour permettre de conclure. Or, une implémentation naïve sous forme de boucles de bissection imbriquées conduit à de nombreuses redondances de calculs. Nous sommes en train de mener certains développements pour éviter des calculs par le biais d'optimisations systématiques ou heuristiques, basées sur une gestion d'historiques de tests. Des premières expérimentations sur une application réelle en robotique ont permis d'observer des gains substantiels.

Enfin, les travaux menés n'ont pu être testés que sur des problèmes de faible dimension car la représentation des continuums de solutions par des boîtes reste la pierre d'achoppement de toute technique d'intervalles. Une alternative possible est l'extension de domaines, c'est à dire, l'obtention d'une unique boîte intérieure maximale à partir d'une solution approximative. Des premiers résultats ont déjà été publiés [GCN07]. L'autre voie consiste à développer de nouveaux types de représentation des continuums de solutions : tracé de frontières, boîtes obliques, pavages à raffinement dynamique, etc.

En programmation par contraintes, nous avons proposé une synthèse des différents filtrages existants et un travail nouveau sur des cohérences partielles utilisant la structure d'unions d'intervalles.

De notre point de vue, gérer des unions d'intervalles s'est révélé avoir peu d'intérêt en pratique. Les unions reposent sur la non-monotonie des contraintes, et cette propriété des contraintes signifie que plusieurs supports peuvent exister pour une même valeur, autrement dit qu'il y a intrinsèquement une combinatoire. L'idée d'exploiter les unions pour mieux filtrer les domaines se justifie donc à partir de cette combinatoire. Mais la véritable

difficulté des CSP continus réside beaucoup plus dans la taille des domaines (le nombre de flottants qu'il faut passer en revue) que dans leur nature combinatoire. Hormis en CAO ou en chimie moléculaire, où les problèmes géométriques comportent des symétries qui peuvent effectivement entraîner une combinatoire, les problèmes numériques sont en général dépourvus de cette caractéristique. En pratique, l'expression des contraintes se limite à des polynômes de faible degré ou des fonctions trigonométriques simples et quelques bissections suffisent à isoler des boîtes où elles sont monotones. Il s'ensuit que les consistances d'unions n'ont de sens que pour les toutes premières boîtes traitées dans l'arbre de recherche ; mais même dans ce cas, ce qu'elles grignotent est en général rattrapé par une ou deux bissections supplémentaires dans une recherche banale (sans union) si bien que les gains en temps de calcul plafonnent avec `lazy-box-set` autour de 20% (`etiq-AC` n'étant pas compétitif). Les cas où `lazy-box-set` produit effectivement de bien meilleurs temps sont ceux où il amène à choisir un point de bissection « gagnant » qui simplifie brusquement le problème. Mais ceci est plus un effet de bord qu'un argument en faveur des unions. De plus, si les unions ne servent qu'à définir une heuristique de point de choix, il s'agirait alors de comparer celle-ci avec d'autres heuristiques (comme par exemple celle de la fonction *smear* [Mer06]).

Avec le recul, ce constat pessimiste n'est pas étonnant : la connaissance des périodes ou des branches monotones d'une fonction élémentaire (par exemple, savoir que la fonction carrée ou inverse est monotone à droite de 0) ne capte qu'une infime partie de la sémantique des contraintes. Cette connaissance ne peut être utilisée pour un découpage naturel (`lazy-box-set`) ou un filtrage (`etiq-AC`) que pour des boîtes englobant un des points délimitant les branches monotones. A l'inverse, le recours aux dérivées, par exemple, est clairement une manière d'exploiter beaucoup plus finement notre connaissance des contraintes.

En revanche, nous pensons que l'étude théorique effectuée sur les cohérences d'unions conserve d'une part un intérêt d'un point de vue académique et, d'autre part, la possibilité de trouver un champ d'application ailleurs que dans les CSP continus. L'idée de départ d'`etiq-AC`, qui consiste à appliquer une cohérence à deux niveaux (c'est à dire d'appliquer simultanément une cohérence faible (comme l'arc–cohérence) sur une représentation fine des domaines et une cohérence forte (comme la cohérence globale) sur une représentation grossière) est une idée nouvelle qui pourrait en effet dépasser le cadre des systèmes d'équations de variables réelles.

En programmation par contraintes, nous nous tournons désormais davantage vers des algorithmes de filtrage dédié à des contraintes ayant une structure particulière (par exemple, des polynômes homogènes) en nous appuyant sur des outils mathématiques adaptés (par exemple, les bases de Gröbner [Gra98]). Idéalement, ces filtrages pourront amener à définir des contraintes globales comme en domaine discret. La gestion de paramètres quantifiés est également un défi où l'approche « contraintes » a certainement beaucoup à apporter, ne serait-ce que pour le développement d'algorithmes de « double bissection », comme nous l'avons mentionné ci-dessus.

Ajoutons que tout au long de cette thèse, nous avons implanté une bibliothèque en C++ (environ 10000 lignes). Elle comprend certains des principaux opérateurs d'analyse par intervalles et de programmation par contraintes, mais son développement est davantage mû par un souci de modularité et de robustesse que par une volonté d'être exhaustif. L'objectif semble atteint puisque cette bibliothèque nous a permis de mettre au point les algorithmes développés dans cette thèse (notamment `etiq-AC`) et qu'elle a également été liée avec succès à d'autres travaux de recherche [DACP06, NCT06, TC06]. Cette bibliothèque pourrait servir de base à un projet plus ambitieux : une plate-forme ouverte pour la résolution de CSP continus avec un langage de haut niveau pour le pilotage d'algorithmes.

Bibliographie

[Abe97] O. Aberth. The Solution of Linear Interval Equations by a Linear Programming Method. *Linear Algebra and its Applications*, 259 :271–279, 1997.

[AFHM05] G. Alefeld, A. Frommer, G. Heindl, and J. Mayer. On the Existence Theorems of Kantorovich, Miranda and Borsuk. Technical report, Bergische Universität Wuppertal, 2005.

[AH83] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.

[AM00] G. Alefeld and G. Mayer. Interval Analysis : Theory and Applications. *J. Comput. Appl. Math.*, 121(1-2) :421–464, 2000.

[BE04] R. Barták and R. Erben. A new Algorithm for Singleton Arc Consistency. In *FLAIRS'04*, 2004.

[Bea97] O. Beaumont. *Algorithmique pour les Intervalles*. PhD Thesis, Université de Rennes, 1997.

[BGGP99] F. Benhamou, F. Goualard, L. Granvilliers, and J-F. Puget. Revising Hull and Box Consistency. In *ICLP*, pages 230–244, 1999.

[Bli92] C. Bliek. *Computer Methods for Design Automation*. PhD Thesis, Massachusetts Institute of Technology, 1992.

[BMR05] H. Batnini, C. Michel, and M. Rueher. Mind The Gaps : A New Splitting Strategy for Consistency Technique. In *CP*, pages 77–91. Springer, 2005.

[BMVH94] F. Benhamou, D. McAllester, and P. Van Hentenryck. CLP(intervals) revisited. In *International Symposium on Logic programming*, pages 124–138. MIT Press, 1994.

[BO97] F. Benhamou and W.J. Older. Applying Interval Arithmetic to Real, Integer and Boolean Constraints. *Journal of Logic Programming*, 32 :1–24, 1997.

[BR05] C. Bessière and Debruyne R. Optimal and Suboptimal Singleton Arc Consistency Algorithms. In *IJCAI*, pages 54–59, 2005.

[Bry92] R.E. Bryant. Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams. *ACM Comput. Surv.*, 24(3) :293–318, 1992.

[CDR99] H. Collavizza, F. Delobel, and M. Rueher. Extending Consistent Domains of Numeric CSP. In *IJCAI*, pages 406–413, 1999.

[CG06] G. Chabert and A. Goldsztejn. Extension of the Hansen-Bliek Method to Right-Quantified Linear Systems. *Reliable Computing*, (accepted for publication), 2006.

[Cle87] J.G. Cleary. Logical Arithmetic. *Future Computing Systems*, 2(2) :125–149, 1987.

[Col75] G.E. Collins. Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. *Automata Theory and Formal Languages*, pages 134–183, 1975.

[Col98] H. Collavizza. A Note on Partial Consistencies over Continuous Domains Solving Techniques. In *CP*, pages 147–161, 1998.

[CTN05a] G. Chabert, G. Trombettoni, and B. Neveu. Box-set Consistency for Interval-Based Constraint Problems. In *ACM SAC*, pages 1439–1443, 2005.

[CTN05b] G. Chabert, G. Trombettoni, and B. Neveu. IGC : Une Nouvelle Consistance Partielle pour les CSPs Continus. In *JFPC - Journées Francophones de Programmation par Contraintes*, 2005.

[DACP06] D. Daney, N. Andreff, G. Chabert, and Y. Papegay. Interval Method for Calibration of Parallel Robots : A Vision-based Experimentation. *Mechanism and Machine Theory*, pages 929–944, 2006.

[Dan63] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[DB97a] R. Debruyne and C. Bessière. From Restricted Path Consistency to Max-Restricted Path Consistency. In *CP*, pages 312–326, 1997.

[DB97b] R. Debruyne and C. Bessière. Some Practicable Filtering Techniques for the Constraint Satisfaction Problem. In *IJCAI*, pages 412–417, 1997.

[DH88] J.H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *J. Symb. Comput.*, 5(1-2) :29–35, 1988.

[DM73] B. Demidovitch and I. Maron. *Éléments de Calcul Numérique*. Editions Mir, Moscou, 1973.

[DP87] R. Dechter and J. Pearl. Network-Based Heuristics for Constraint Satisfaction Problems. *Artificial Intelligence*, 34 :1–38, 1987.

[Fal94] B. Faltings. Arc-consistency for continuous variables. *Artificial Intelligence*, 65, 1994.

[FHL04] A. Frommer, F. Hoxha, and B. Lang. Proving Existence of Zeros Using the Topological Degree and Interval Arithmetic. Technical report, Bergische Universität Wuppertal, 2004.

[FL04] A. Frommer and B. Lang. Existence Tests for Solutions of Nonlinear Equations Using Borsuk's Theorem. Technical report, Bergische Universität Wuppertal, 2004.

[FLS04] A. Frommer, B. Lang, and M. Schnurr. A Comparison of the Moore and Miranda Existence Tests. *Computing*, 72(3–4) :349–354, 2004.

[Fre82] E. Freuder. A Sufficient Condition for Backtrack-Free Search. *Journal of the ACM*, 29(1) :24–32, 1982.

[GB01] L. Granvilliers and F. Benhamou. Progress in the Solving of a Circuit Design Problem. *Journal of Global Optimization*, 20(2) :155–168, 2001.

[GC06a] A. Goldsztejn and G. Chabert. A Generalized Interval LU Decomposition for the Solution of Interval Linear Systems. In *NMA - 6th International Conference on Numerical Methods and Applications*. Springer, 2006.

[GC06b] A. Goldsztejn and G. Chabert. On the Approximation of Linear AE-Solution Sets. In *SCAN - 12th International Symposion on Scientific Computing, Computer Arithmetic and Validated Numerics*, 2006.

[GCN07] C. Grandón, G. Chabert, and B. Neveu. Generalized Interval Projection : A New Technique for Consistent Domain Extension. In *IJCAI*, 2007.

[GDRT05] A. Goldsztejn, D. Daney, M. Rueher, and P. Taillibert. Modal intervals revisited : a mean-value extension to generalized intervals. In *QCP*, 2005.

[GGB99] L. Granvilliers, F. Goualard, and F. Benhamou. Box Consistency through Weak Box Consistency. In *IEEE Conference on Tools with Artificial Intelligence*, pages 373–380, 1999.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability (A Guide to the Theory of NP-Completeness)*. W.H. Freeman and Company, 1979.

[GJ06] A. Goldsztejn and L. Jaulin. Inner and Outer Approximations of Existentially Quantified Equality Constraints. In *CP*. Springer, 2006.

[GN06] C. Grandón and B. Neveu. A Specific Quantifier Elimination for Inner Box Test in Distance Constraints with Uncertainties. Technical Report 5883, INRIA, 2006.

[GnMA85] E. Gardeñes, H. Mielgo, and Trepat A. Modal Intervals : Reason and Ground Semantics. *Interval Mathematics*, 212 :27–35, 1985.

[GnSJ+01] E. Gardeñes, M. Á Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo, and A. Trepat. Modal Intervals. *Reliable Computing*, 7(2) :77–111, 2001.

[Gol05a] A. Goldsztejn. A Right-Preconditioning Process for the Formal-Algebraic Approach to Inner and Outer Estimation of AE-solution Sets. *Reliable Computing*, 11(6) :443–478, 2005.

[Gol05b] A. Goldsztejn. *Définition et Applications des Extensions des Fonctions Réelles aux Intervalles Généralisés*. PhD Thesis, Université de Nice-Sophia Antipolis, 2005.

[Gol06] A. Goldsztejn. A Branch and Prune Algorithm for the Approximation of Non-Linear AE-solution Sets. In *ACM SAC*, pages 1650–1654, 2006.

[Gol07a] A. Goldsztejn. Comparison of the Hansen-Sengupta and the Frommer-Lang-Schnurr Existence Tests. *Computing (Accepted for publication)*, 2007.

[Gol07b] A. Goldsztejn. Modal Intervals Revisited, Part 1 : A Generalized Interval Natural Extension. *Reliable Computing*, (under submission), 2007.

[Gol07c] A. Goldsztejn. Modal Intervals Revisited, Part 2 : A Generalized Interval Mean-Value Extension. *Reliable Computing*, (under submission), 2007.

[Gou05] F. Goualard. On Considering an Interval Constraint Solving Algorithm as a Free-Steering Nonlinear Gauss-Seidel Procedure. In *ACM SAC*, pages 1434–1438, 2005.

[Gra98] L. Granvilliers. A Symbolic-Numerical Branch-and-Prune Algorithm for Solving Nonlinear Polynomial Systems. *Journal of Universal Computer Science*, 4(2) :125–146, 1998.

[Gra01] L. Granvilliers. On the Combination of Interval Constraint Solvers. *Reliable Computing*, 7(6) :467–483, 2001.

[Gri00] A. Griewank. *Evaluating Derivatives : Principles and Techniques of Algorithmic Differentiation*. SIAM, 2000.

[Han65] E.R. Hansen. Interval Arithmetic in Matrix Computations, Part 1. *SIAM J. Numer. Anal.*, 2(2) :308–320, 1965.

[Han78] E.R. Hansen. Interval Forms of Newton's Method. *Computing*, 20 :153–163, 1978.

[Han92a] E.R. Hansen. Bounding the Solution of Interval Linear Equations. *SIAM J. Numer. Anal.*, 29(5) :1493–1503, 1992.

[Han92b] E.R. Hansen. *Global Optimization using Interval Analysis*. Marcel Dekker, 1992.

[HM05] F. Hao and J-P. Merlet. Multi-criteria Optimal Design of Parallel Manipulators based on Interval Analysis. *Mechanism and Machine Theory*, 40(2) :151–171, 2005.

[HS80] E.R. Hansen and S. Sengupta. Bounding Solutions of Systems of Equations Using Interval Analysis. *BIT Numerical Mathematics*, 21(2) :203–211, 1980.

[HSVJ05] P. Herrero, M.A. Sainz, J. Vehí, and L. Jaulin. Quantified Set Inversion Algorithm with Applications to Control. *Reliable Computing*, 11(5) :369–382, 2005.

[Hyv92] E. Hyvönen. Constraint Reasoning Based on Interval Arithmetic—The Tolerance Propagation Approach. *Artificial Intelligence*, 58 :71–112, 1992.

[Jan04] C. Jansson. Rigorous Lower and Upper Bounds in Linear Programming. *SIAM Jour. of Optimization*, 14(3) :914–935, 2004.

[Jau02] L. Jaulin. Nonlinear bounded-error state estimation of continuous-time systems. *Automatica*, pages 1079–1082, 2002.

[Jau06] L. Jaulin. Localization of an Underwater Robot using Interval Constraint Propagation. In *CP*. Springer, 2006.

[JKDW01] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, 2001.

[Kau80] E. Kaucher. Interval Analysis in the Extended Interval Space. *Computing, Suppl.*, 2 :33–49, 1980.

[Kea96] R.B. Kearfott. *Rigorous Global Search : Continuous Problems*. Springer, 1996.

[Kea02] R.B. Kearfott. Symbolic Preconditioning with Taylor Models : Some Examples. *Reliable Computing*, 8(6) :453–468, 2002.

[KLRK97] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl. *Computational complexity and feasibility of data processing and interval computations*. Kluwer, 1997.

[Lak96] A.V. Lakeyev. On the Computational Complexity of the Solution of Linear Systems with Moduli. *Reliable Computing*, 2(2) :125–132, 1996.

[Leb99] Y. Lebbah. *Contribution à la Résolution de Contraintes par Consistance Forte*. PhD Thesis, Université de Nantes, 1999.

[Lho93] O. Lhomme. Consistency Techniques for Numeric CSPs. In *IJCAI*, pages 232–238, 1993.

[Lho94] O. Lhomme. *Contribution à la résolution de contraintes sur les réels par propagation d'intervalles*. PhD Thesis, Université de Nice-Sophia Antipolis, 1994.

[Mac77] A.K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8 :99–118, 1977.

[MD06] J-P. Merlet and P. Donelan. On the Regularity of the Inverse Jacobian of Parallel Robots. *ARK*, pages 41–48, 2006.

[Mer04] J-P. Merlet. Solving the Forward Kinematics of a Gough-type Parallel Manipulator with Interval Analysis. *Int. J. of Robotics Research*, 23(3) :221–236, 2004.

[Mer06] J-P. Merlet. *Alias-C++*. http ://www-sop.inria.fr/coprin/logiciels/ALIAS, 2006.

[Moo66] R. Moore. *Interval Analysis*. Prentice-Hall, 1966.

[NCT06] B. Neveu, G. Chabert, and G. Trombettoni. When Interval Analysis Helps Inter-Block Backtracking. In *CP*. Springer, 2006.

[Neu90] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, 1990.

[Neu99] A. Neumaier. A simple Derivation of the Hansen-Bliek-Rohn-Ning-Kearfott Enclosure for Linear Interval Equations. *Reliable Computing*, 5(2) :131–136, 1999.

[NK97] S. Ning and R.B. Kearfott. A Comparison of Some Methods for Solving Linear Interval Equations. *SIAM J. Numer. Anal.*, 34(1) :1289–1305, 1997.

[Oet65] W. Oettli. On the Solution Set of a Linear System with Inaccurate Coefficients. *SIAM J. Numer. Anal.*, 2(1) :115–118, 1965.

[OR70] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.

[Ort69] H-J. Ortolf. Eine Verallgemeinerung der Intervallarithmetik. *Geselschaft fuer Mathematik und Datenverarbeitung*, 11 :1–71, 1969.

[Rég05] J-C. Régin. AC- : A Configurable, Generic and Adaptive Arc Consistency Algorithm. In *CP*, pages 505–519. Springer, 2005.

[Roh80] J. Rohn. An Existence Theorem for Systems of Nonlinear Equations. *Zeitschrift fuer Angewandte Mathematik und Mechanik*, 60 :345, 1980.

[Roh89] J. Rohn. Systems of Linear Interval Equations. *Linear Algebra and its Applications*, 126 :39–78, 1989.

[Roh93] J. Rohn. Cheap and Tight Bounds : The Recent Result by E. Hansen Can Be Made More Efficient. *Interval Computations*, 1(4) :13–21, 1993.

[Roh02] J. Rohn. Systems of Interval Linear Equations and Inequalities (Rectangular Case). Technical Report 875, Academy of Science of the Czech Republic, 2002.

[Roh05] J. Rohn. How Strong is Strong Regularity ? *Reliable Computing*, 11(6) :491–493, 2005.

[RP93] J. Rohn and S. Poljak. Checking Robust NonSingularity is NP-hard. *Mathematics of Control, Signals, and Systems*, 6(1) :1–9, 1993.

[Rum83] S.M. Rump. Validated Solutions of Linear Equations. *A New Approach to Scientific Computation*, pages 51–120, 1983.

[Rum88] S.M. Rump. Algorithms for Verified Inclusions-Theory and Practice. *Reliability in computing : the role of interval methods in scientific computing*, pages 109–126, 1988.

[Sha95] S.P. Shary. On Optimal Solution of Interval Linear Equations. *SIAM J. Numer. Anal.*, 32(2) :610–630, 1995.

[Sha99] S.P. Shary. Outer Estimation of Generalized Solution Sets to Interval Linear Systems. *Reliable Computing*, 5(3) :323–335, 1999.

[Sha01] I.A. Sharaya. On Maximal Inner Estimation of the Solution Sets of Linear Systems with Interval Parameters. *Reliable Computing*, 7(5) :409–424, 2001.

[Sha02] S.P. Shary. A New Technique in Systems Analysis Under Interval Uncertainty and Ambiguity. *Reliable Computing*, 8(5) :321–418, 2002.

[Tap71] R.A. Tapia. The Kantorovitch Theorem for Newton's Method. *American Mathematic Monthly*, 78(1.ea) :389–392, 1971.

[TC06] G. Trombettoni and G. Chabert. Constructive Interval Disjunction. In *IntCP*, 2006.

[VHMD97] P. Van Hentenryck, L. Michel, and Y. Deville. *Numerica : A Modeling Language for Global Optimization*. MIT Press, Cambridge, 1997.

[VHMK97] P. Van Hentenryck, D. McAllester, and D. Kapur. Solving Polynomial Systems Using a Branch and Prune Approach. *SIAM J. Numer. Anal.*, 34(2) :797–827, 1997.

[Wal75] D.L. Waltz. Understanding Line Drawings of Scenes with Shadows. *The Psychology of Computer Vision*, pages 19–91, 1975.

Résumé

Cette thèse porte sur la résolution numérique de systèmes d'équations non-linéaires. Elle présente des contributions dans trois sous-domaines utilisant le calcul par intervalles : l'analyse par intervalles, les intervalles modaux et la programmation par contraintes. Le traitement des systèmes linéaires est au centre de plusieurs des travaux. Il sert notamment de base à la résolution dans le cas non-linéaire.

En analyse par intervalles, nous proposons une extension de la méthode de Hansen-Bliek pour l'approximation extérieure optimale de l'ensemble des solutions d'un système linéaire dont les coefficients varient dans des intervalles. L'extension proposée prend en compte la possibilité de choisir le quantificateur (existentiel ou universel) associé à certains coefficients du système. Cette liberté permet de modéliser un plus large éventail de problèmes linéaires, notamment ceux obtenus itérativement à partir de l'opérateur de Newton (intervalle) généralisé. Une généralisation de la décomposition LU exploitant l'arithmétique de Kaucher est également proposée.

Sur les intervalles modaux, nous proposons une construction originale de la théorie qui s'articule autour de la notion d'*image quantifiée*, généralisation naturelle de la notion d'image d'une fonction. La construction proposée présente certains avantages, comme celui de pouvoir donner un sens plus concret à l'arithmétique de Kaucher.

En programmation par contraintes, nous étudions de nouvelles cohérences partielles reposant sur la structure d'unions d'intervalles. Cette structure peut être utilisée pour représenter plus finement le domaine des variables dans des systèmes de contraintes numériques. Nous montrons notamment dans quelle mesure, et à quel coût, la propriété d'arc-cohérence peut ainsi être obtenue grâce à cette nouvelle représentation.

Abstract

This dissertation is devoted to solving systems of nonlinear equations. It presents a survey of various theories based on interval computations (interval analysis, modal intervals and constraint programming over the reals) as well as some new results in each of these areas. A special care is brought to the linear case, one of our main field of study. This case is of considerable significance since all nonlinear methods are based upon it.

In interval analysis, we give an extension of the Hansen-Bliek method which computes an optimal outer approximation of the solution set of interval linear systems. This extension allows more freedom in the choice of the quantifiers (existential or universal) associated to the coefficients, thus handling a wider variety of problems. A generalization of the LU decomposition based on Kaucher's interval arithmetic is also given.

We also propose a new formulation of the modal intervals theory, with the underlying concept of *quantified range* – a natural generalization of the range of a function. This new approach allows us to introduce Kaucher's arithmetic with a vivid meaning, and not only as an abstract algebraic extension of the classical interval arithmetic.

In constraint programming, we study local consistencies with domains represented by unions of intervals. This structure is more adapted to store refinements of domains throughout propagation. We show to which extent arc-consistency can be achieved with this structure.