HAL
open science

Un résultat de convergence des algorithmes parallèles asynchrones. Application aux opérateurs maximaux fortement monotones

Abdenasser Benahmed

To cite this version:

Abdenasser Benahmed. Un résultat de convergence des algorithmes parallèles asynchrones. Application aux opérateurs maximaux fortement monotones. Mathématiques [math]. Faculté des Sciences d'Oujda, 2005. Français. NNT : . tel-00134642v2

HAL Id: tel-00134642

https://theses.hal.science/tel-00134642v2

Submitted on 5 Mar 2007

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

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

UNIVERSITÉ MOHAMED PREMIER
FACULTÉ DES SCIENCES
OUJDA

N^o d'ordre : 80/05

THÈSE

Pour l'obtention du
Doctorat en Sciences
Discipline : Mathématiques
Spécialité : Analyse Numérique et Informatique

Un résultat de convergence des algorithmes parallèles asynchrones. Application aux opérateurs maximaux fortement monotones.

Par

ABDENASSER BENAHMED

Soutenue publiquement le 28 juillet 2005 devant le jury :

Président :

M. E. M. DAOUDI	Faculté des sciences Oujda

Examinateurs :

M. A. ADDOU	Faculté des sciences Oujda
M. O. CHAKRONE	Faculté des sciences Oujda
M. A. LIDOUH	Faculté des sciences Oujda
M. A. ROUBI	Faculté des sciences et techniques Settat

À toute ma famille...

Remerciements

Tout au long de ces années de thèse, certaines personnes m'ont soutenu et aidé. Je tiens à les remercier tout particulièrement.

Je voudrais tout dabord te remercier Ahmed ADDOU, pour avoir dirigé cette thèse. Merci pour ta disponibilité, ton soutien permanent et ton aide qui m'ont permis de mener à bien ces travaux. Aussi, je tiens à t'assurer ma profonde gratitude pour m'avoir initié à la recherche.

Je tiens également à remercier El Mostapha DAOUDI et Omar CHAKRONE de la faculté des sciences d'Oujda et Ahmed ROUBI de la faculté des sciences et techniques de Settat de m'avoir fait l'honneur d'accepter d'être rapporteurs de cette thèse, ainsi que Abdelouahab LIDOUH de la faculté des sciences d'oujda pour avoir accepté d'être membre de mon jury.

Merci à tous mes proches, famille et amis, qui m'ont soutenu et encouragé tout au long de ces années de thèse.

Résumé

Dans ce travail, nous avons considéré des algorithmes parallèles asynchrones associés à des applications non linéaires non-expansives définies sur $\mathbb{R}^n$. Nous avons montré le résultat de convergence concernant ces algorithmes vers un point fixe de ces applications relativement à la norme uniforme sur $\mathbb{R}^n$. Ce théorème étend le résultat donné par Bahi[3] au cas non linéaire. Nous avons ensuite montré comment ces algorithmes sont bien adaptés au calcul de la solution d'un opérateur maximal fortement monotone défini sur $\mathbb{R}^n$ et au calcul des solutions d'un opérateur maximal monotone dans le cas de l'algorithme parallèle synchrone de Jacobi. Ensuite, nous avons appliqué ces résultats aux calculs du minimum de fonctionnelles, du point selle, de la solution des programmes convexes et enfin de la solution du problème de l'inégalité variationnelle.

Abstract

A convergence result for parallel asynchronous algorithms. Application to maximal strongly monotone operators

In this work, we have considered a parallel asynchronous algorithms associated with nonlinear and nonexpansive mapping defined in $\mathbb{R}^n$. We have shown the convergence result for this algorithms to a fixed point of such mapping with respect to the uniform norm. This result extends the result published by Bahi[3] to the nonlinear case. We have then proved how this algorithms can be used to calculate the solution of a maximal strongly monotone operator in $\mathbb{R}^n$ and calculate the solutions of a maximal monotone operator in the special case of parallel synchronous algorithm of Jacobi. Finally, this results are applied to calculate a minimum of functionals, a saddle point, a solution of convex programming and a solution of variational inequality problem.

Table des matières

Remerciements **2**

Résumé **3**

Introduction **6**

1 Notions et résultats préliminaires **14**

1.1 Notions d'analyse convexe 14

1.2 Notions d'opérateurs maximaux monotones 19

1.3 Algorithmes asynchrones 23

2 Convergence de l'algorithme général **29**

2.1 Introduction 29

2.2 Le cas asynchrone 29

2.3 Algorithme synchrone de Jacobi 35

2.4 Solution des opérateurs maximaux monotones 35

3 Applications **40**

3.1 Minimum d'une fonctionnelle 40

3.2 Point selle 42

3.3 Programme convexe 45

3.4 Inégalité variationnelle 46

Conclusion **49**

Références **50**

Introduction

Lors de la modélisation d'un grand nombre de problèmes mathématiques, physiques, économiques ou sociaux, nous sommes confronté à la résolution d'une équation de la forme :

$$f(x) = 0 \tag{0.1}$$

où $x \in E$ espace vectoriel, et f une application de E vers E. Cette équation peut être écrite sous la forme :

$$F(x) = x \tag{0.2}$$

si nous posons $F(x) = x - f(x)$, équation connue sous le nom de problème de point fixe, étant donné que les solutions de l'équation (0.2) sont appelées points fixes de F.

Pour résoudre ce type de problème, nous pouvons utiliser deux principales méthodes numériques :

– Les méthodes directes qui se définissent sous forme d'équations de récurrence et conduisent à la solution en un nombre fini d'étapes. Ces méthodes ont l'inconvénient de ne pas prendre en considération les particularités de l'application F.

– Les méthodes itératives qui évaluent la solution du problème par approximations successives en engendrant une suite de vecteurs qui doit tendre

vers la solution. Elles ont l'avantage d'utiliser les particularités de F, mais la convergence de ces méthodes vers la solution exige des conditions plus ou moins sévères.

De nombreux algorithmes itératifs ont été développés pour le calcul de la solution du problème (0.1). On peut citer par exemple, les méthodes de substitutions successives (méthode de Jacobi, méthode de Gauss-Seidel), les méthodes de Newton et ses dérivées, les méthodes de minimisation de fonctionnelles (gradient, gradient conjugué, Levenberg-Marquardt, Powell...) et bien d'autres plus au moins anciennes.

Ces algorithmes rentrent dans le cadre de ce qu'on appelle algorithmes séquentiels, c'est-à-dire des algorithmes qui pourront être implémentés sur des machines mono-processeur qui les exécutera séquentiellement (instruction après instruction). La parallélisation de ces algorithmes consiste à les réécrire en les divisant en des tâches indépendantes, chacune exécutée par un processeur. Ces algorithmes parallèles s'implémentent sur des machines multi-processeurs appelées machines parallèles.

Si les processeurs commencent chaque itération en même temps, et attendent d'avoir reçu les données nécessaires mises à jour à l'itération précédente pour commencer l'itération suivante, on parle alors d'algorithmes synchrones. Avec les algorithmes asynchrones, les processeurs effectuent leurs iétrations sans tenir compte de l'avancement des autres. Il n'y a plus d'attente des données venant d'autres processeurs pour commencer une itération.

L'idée des itérations chaotiques a été introduite par Chazan et Miranker dans [10] en 1969 pour résoudre le système linéaire

$$Ax = b$$

où A est une matrice symétrique définie positive dans $\mathbb{R}^{n\times n}$, x et b sont des vecteurs de $\mathbb{R}^n$.
En décomposant la matrice A sous la forme $A = D - E$, on obtient l'équation

$$x = F(x) \tag{0.3}$$

où $F(x) = Bx + C$, $B = D^{-1}E$ et $C = D^{-1}b$.
Ils ont proposé le modèle suivant

$$\begin{cases} x^0 \in \mathbb{R}^n \\ x_i^{p+1} = \begin{cases} x_i^p & si \quad i \neq l(p) \\ F_i(x_1^{p-r_1(p)}, ..., x_n^{p-r_n(p)}) & si \quad i = l(p) \end{cases} \\ i = 1, ..., n \\ p = 0, 1, .. \end{cases} \tag{0.4}$$

où pour tout $p \in \mathbb{N}$, x^p dénote le vecteur itéré à l'étape p, $l(p) \in \{1, ..., n\}$ et $R = \{(r_1(p), ..., r_n(p))\}_{p\in\mathbb{N}}$ est une suite de $\mathbb{N}^n$ vérifiant :
Pour $s \in \mathbb{N}^*$,

(a) $0 \leq r_i(p) < s \ \forall i \in \{1, ..., n\}, \ \forall p \in \mathbb{N}$.

(b) $\forall i \in \{1, ..., n\}$ *l'ensemble* $\{p \in \mathbb{N}, \ i = l(p)\}$ *est infini.*

Ce modèle doit être interprété comme suit : A chaque itération p+1, seulement la composante $l(p)$ qui est mise à jour, les autres restent inchangées. La condition (a) affirme que les retards $r_i(p)$ $(1 \leq i \leq n)$ dus aux communications entre processeurs et aux différents calculs sont bornés, et la condition (b) signifie qu'aucun bloc de composante ou aucune composante n'est abandonnée définitivement lors du processus itératif ; on doit réactualiser chaque composante une infinité de fois et donc toutes les composantes sont mises à jour de manière équitable. Notons que les méthodes de Gauss-Seidel et de Jacobi sont des cas particuliers du modèle (0.4), en effet :
– dans le cas où $r_1(p) = r_2(p) = ... = r_n(p) = 0$ et $l(p) \equiv (p \ mod \ n) + 1$, on tombe sur les relaxations de Gauss-Seidel.
– et si nous prenons $r_1(p) = r_2(p) = ... = r_n(p) = (l(p) - 1)$ et $l(p) \equiv (p \ mod \ n) + 1$ alors on aura les relaxations de Jacobi.
Dans [10] Chazan et Miranker ont établi le résultat suivant :

Théorème 0.1
Les itération $\{x^p\}_{p\in\mathbb{N}}$ *définies par (0.4) convergent vers la solution de (0.3) ssi* $\rho(|B|) < 1$.

où $|B|$ est la matrice dont les éléments sont les valeurs absolues des éléments de B, $\rho(A)$ est le rayon spectrale de la matrice A.

Les travaux de Chazan et Miranker ont été étendu dans le cas de systèmes non linéaires par F. Robert et al. [21],[22], dans une situation où les processeurs communiquent entre eux de manière synchrone. Dans un même temps, en 1974 (et 1975) Miellou dans [16] et [17] a étendu le modèle (0.4) dans deux sens : le premier est qu'à chaque itération, la mise à jour peut toucher plus qu'une composante (un ensemble de composantes), le deuxième est que l'application F peut être non linéaire. Il a alors proposé le modèle suivant à retards bornés :

$$\begin{cases} x^0 \in \mathbb{R}^n \\ x_i^{p+1} = \begin{cases} x_i^p & si \quad i \notin J(p) \\ F_i(x_1^{p-r_1(p)}, ..., x_n^{p-r_n(p)}) & si \quad i \in J(p) \end{cases} \\ i = 1, ..., n \\ p = 0, 1, .. \end{cases} \tag{0.5}$$

où $\{J(p)\}_{p\in\mathbb{N}}$ est une suite de sous-ensembles non vides de $\{1, ..., n\}$ appelée stratégie et $R = \{(r_1(p), ..., r_n(p))\}_{p\in\mathbb{N}}$ est une suite de $\mathbb{N}^n$ définissant la suite des retards vérifiant :

(a) $\forall i \in \{1, ..., n\}\,,\ 0 \leq r_i(p) \leq p.$

(b) $\forall i \in \{1, ..., n\}$ l'ensemble $\{p \in \mathbb{N},\ i \in J(p)\}$ est infini.

La notion de stratégie correspond aux numéros des composantes sur lesquelles on travaille et rend bien compte du parallélisme ; à l'itération p on traitera en prallèle les composantes i telles que $i \in J(p)$. La notion de retards rend compte de l'asynchronisme avec lequel est traitée chacune des composantes.

Dans la démonstration de son résultat, Miellou a utilisé une technique de contraction en norme vectorielle définie comme suit :

Définition 0.1
Une application F de $\mathbb{R}^n$ vers lui-même est dite contractive par rapport à $x^* \in \mathbb{R}^n$ s'il existe une matrice T dans $\mathbb{R}^{n\times n}$ non-négative telle que :

(i) $\rho(T) < 1$

(ii) $|F(x) - F(x^*)| \leq T|x - x^*| \ \forall x \in \mathbb{R}^n$

où $|x|$ est le vecteur à composantes les valeurs absolues des composantes du vecteur x, et l'inégalité dans (ii) est vectorielle, c'est-à-dire vérifée composante par composante.
Il a supposé l'hypothèse suivante traduisant les retards bornés :
(c) Toute composante est mise à jour au moins une fois après chaque s itérations consécutives ($s \in \mathbb{N}^*$).
Il a alors montré le résultat suivant :

Théorème 0.2
Si F admet un point fixe x^* et si elle est contractive par rapport à x^*, alors les itérations (0.5) convergent vers le point fixe x^* de F.

Il est facile de vérifier que les conditions de Chazan-Miranker entraînent celles de Miellou et donc ce modèle est plus général que celui proposé par Chazan-Miranker. En effet, on a la proposition suivante :

Proposition 0.3
Soit l'application $F(x) = Bx + C$ de $\mathbb{R}^n$ vers $\mathbb{R}^n$ telle que $\rho(|B|) < 1$. Alors

1. *F admet un point fixe unique x^*.*
2. *F est contractive par rapport à x^* au sens de la définition 0.1.*

En 1978 Baudet dans [5] a généralisé les itérations chaotiques de Chazan-Miranker et de Miellou par des itérations qu'il a nommées itérations asynchrones. Il a proposé le modèle suivant où les retards considérés peuvent être infinis :

$$\begin{cases} x^0 \in \mathbb{R}^n \\ x_i^{p+1} = \begin{cases} x_i^p & si \quad i \notin J(p) \\ F_i(x_1^{s_1(p)}, ..., x_n^{s_n(p)}) & si \quad i \in J(p) \end{cases} \\ i = 1, ..., n \\ p = 0, 1, .. \end{cases} \tag{0.6}$$

où $\{J(p)\}_{p\in\mathbb{N}}$ est une suite de sous-ensembles non vides de $\{1, ..., n\}$ et $S = \{(s_1(p), ..., s_n(p))\}$ est une suite de $\mathbb{N}^n$ vérifiant pour tout $i \in \{1, ..., n\}$:

(a) $s_i(p) \leq p, \ \forall p \in \mathbb{N}$.

(b) l'ensemble $\{p \in \mathbb{N}, \ i \in J(p)\}$ *est infini.*

(c) $\lim\limits_{p\to\infty} s_i(p) = \infty$.

$J(p)$ est l'ensemble des composantes relaxées à l'itération p.
$p - s_i(p)$ correspond au retard éventuel dû au $i^{ème}$ processeur lors du calcul du $i^{ème}$ bloc à l'itération p.
La condition (a) affirme que les valeurs du vecteur itéré utilisées à l'itération $p + 1$ proviennent aux maximum de l'itération p. La condition (c) indique que les valeurs trop anciennes sont écartées au fur et à mesure que les calculs progressent.

Baudet a supposé que F est contractive sur $D \subset \mathbb{R}^n$ c'est-à-dire contractive par rapport à chaque $x \in D$ condition plus forte que celle proposée par Miellou et a montré le résultat de convergence suivant :

Théorème 0.4
Si F est contractive sur un ensemble fermé $D \subset \mathbb{R}^n$ et si $F(D) \subset D$ alors tout algorithme asynchrone (0.6) converge vers l'unique point fixe de F dans D.

Citons également les travaux de Bertsekas et Tsitsiklis [7],[8] et [9], où les itérés successifs appartiennent à des espaces emboités ce qui assure la convergence, ce dernier travail ayant été repris par Miellou, Cortey-Dumont et Boulbrachêne dans [19] en prenant en compte la propagation d'erreur d'arrondi. Dans un context différent, El Tarazi[13] a également établi un résultat

de convergence des algorithmes asynchrones par des techniques de contraction selon une norme scalaire appropriée.

La quasi-totalité des résultats de convergence concernant ces algorithmes utilisent des techniques de contraction [5], [10], [13], [17], [21] ou des techniques d'ordre partiel [12], [18]. Bahi dans [3] a donné un résultat de convergence concernant les algorithmes parallèles asynchrones pour des problèmes linéaires de point fixe utilisant des opérateurs non-expansifs relativement à une norme uniforme avec poids. Dans son travail, il a utilisé un processus itératif non stationnaire, c'est-à-dire qu'il a pris une matrice T^p qui varie à chaque itération p, utilisant les hypothèses :

$$\begin{cases} \exists K \in \mathbb{N}^*,\ \forall p \in \mathbb{N} : p - K + 1 \leq s_i(p) \leq p \\ \textit{(retards bornés)} \end{cases} \tag{0.7}$$

$$\begin{cases} \textit{La suite de matrices } \{T^p\}_{p\in\mathbb{N}} \textit{ converge vers} \\ \textit{une matrice } Q \textit{ paracontractante} \\ \textit{par rapport à la norme uniforme pondérée, i.e.} \\ x \neq Qx \Leftrightarrow \|Qx\|_{\infty,\gamma} < \|x\|_{\infty,\gamma},\ \forall x \in \mathbb{R}^n \end{cases} \tag{0.8}$$

Notre travail (voir Addou et Benahmed [1] et [2]), consiste à étudier la convergence des algorithmes parallèles asynchrones associés à des opérateurs non linéaires non-expansifs définis sur $\mathbb{R}^n$. Dans notre approche, nous avons suivi les étapes traitées par Miellou dans [17] et par Bahi dans [3] et avons donc supposé que les retards dus aux communications et aux différents calculs sont bornés. Cependant, à la place de la condition (0.8) nous avons supposé que l'application F vérifie l'hypothèse suivante :

$$\forall x, x' \in \mathbb{R}^n,\ \|F(x) - F(x')\|^2 \leq \langle F(x) - F(x'), x - x' \rangle \tag{0.9}$$

où $\langle ., . \rangle$ est le produit scalaire euclidien sur $\mathbb{R}^n$ et $\|..\|$ désigne la norme euclidienne (associée), hypothèse vérifée par une large classe d'opérateurs comme nous allons le voir dans la section 2.2.

Nous montrerons dans un premier temps qu'étant donnée une application F non linéaire de $\mathbb{R}^n$ vers $\mathbb{R}^n$ et sous certaines conditions portées sur

F, l'algorithme parallèle asynchrone à retards bornés associé à l'application F converge vers un point fixe de F. Ensuite, dans un deuxième temps, nous montrerons comment ces algorithmes sont bien adaptés au calcul des solutions d'un opérateur T maximal fortement monotone défini sur $\mathbb{R}^n$ et au calcul des solutions d'un opérateur T maximal monotone dans le cas de l'algorithme parallèle synchrone (sans retards) de Jacobi utilisant dans les deux cas l'application $F = (I + cT)^{-1}$ où c est un réel strictement positif bien choisi. Nous appliquerons enfin ces derniers résultats aux calculs de minimum de fonctionnelles, des points selle, des solutions des programmes convexes et enfin des solutions du problème de l'inégalité variationnelle.

Le reste du travail est organisé comme suit :

Dans le chapitre 1, nous présenterons les objets et résultats que nous allons utiliser dans la suite. Nous citerons dans un premier temps des notions d'analyse convexe, notamment concernant les fonctions convexes, fonctions concaves et fonctions selle comme elles ont été définies dans Rockafellar[23]. Ensuite dans un deuxième temps, nous parlerons des opérateurs maximaux monotones et leurs propriétés notamment celles qui nous intéressent dans ce travail, et nous terminerons le chapitre par une description plus ou moins détaillée des algorithmes parallèles asynchrones associés à un opérateur F de $\mathbb{R}^n$ vers $\mathbb{R}^n$ dans son cadre général.

Dans le chapitre 2, nous démontrerons le résultat principal concernant la convergence de l'algorithme général vers un point fixe d'un opérateur non linéaire F de $\mathbb{R}^n$ vers $\mathbb{R}^n$. Nous montrerons ensuite la convergence de cet algorithme vers la solution d'un opérateur maximal fortement monotone défini sur $\mathbb{R}^n$ et la convergence de l'algorithme parallèle synchrone de Jacobi vers la solution d'un opérateur maximal monotone défini sur $\mathbb{R}^n$.

Dans le chapitre 3, nous appliquerons les résultats du chapitre 2 aux calculs du minimum d'une fonctionnelle, du point selle d'une fonction selle, de la solution des programmes convexes et enfin de la solution du problème de l'inégalité variationnelle.

Chapitre 1

Notions et résultats préliminaires

Dans tout le chapitre, H désigne un espace de Hilbert réel muni du produit scalaire noté $\langle .,. \rangle$ et de la norme associée notée $\|..\|$.
L'ensemble des parties de H sera noté 2^H.

1.1 Notions d'analyse convexe

Toutes les définitions et propriétés citées dans cette section peuvent être consultées dans Rockafellar[23].

(a) **Fonctions convexes**

Considérons une fonction f définie sur H à valeurs dans $]-\infty, +\infty]$. Donc f peut prendre la valeur $+\infty$.

(a1) Le domaine de f est l'ensemble

$$dom(f) = \{x \in H,\ f(x) < +\infty\}$$

(a2) L'épigraphe de f est l'ensemble

$$epif = \{(x, c) \in H \times \mathbb{R},\ f(x) \leq c\}$$

(a3) La fonction f est dite
– convexe si :

$$f(tx + (1-t)x') \leq tf(x) + (1-t)f(x') \ \forall x, x' \in H, \ \forall t \in [0,1]$$

– a-fortement convexe ($a > 0$) si :

$$\begin{cases} f(tx + (1-t)x') \leq tf(x) + (1-t)f(x') - \frac{1}{2}at(1-t)\|x - x'\|^2 \\ \forall x, x' \in H, \ \forall t \in [0,1] \end{cases}$$

(a4) La fonction convexe f est dite propre si son domaine $dom(f)$ est un ensemble non vide. Nous utiliserons aussi l'écriture $f \not\equiv +\infty$ pour dire que f est propre.

(a5) La fonction f est dite semi-continue inférieurement (s.c.i.) si pour tout $c \in \mathbb{R}$, l'ensemble

$$\{x \in H, \ f(x) \leq c\}$$

est un ensemble fermé.

Des exemples et propriétés des fonctions convexes et s.c.i. peuvent être trouvés dans les cours de topologie générale.

(a6) Soit f une fonction convexe et propre. Si f est s.c.i. alors on dit que f est fermée.

(a7) Un élément x de H est dit sous-gradient de f au point $x_0 \in H$ si

$$f(x') \geq f(x_0) + \langle x, x' - x_0 \rangle, \quad \forall x' \in H$$

L'ensemble des sous-gradients de f en x_0 est appelé sous-différentiel de f au point x_0 et est noté par $\partial f(x_0)$. L'opérateur multivoque[1] $\partial f : x_0 \to \partial f(x_0)$ est appelé sous-différentiel de f. L'ensemble $\partial f(x_0)$ est un convexe fermé qui peut être vide, et si $\partial f(x_0) \neq \phi$, alors f est dite sous-différentiable en x_0.

[1]Voir la définition des opérateurs multivoques dans la section 1.2.

(b) **Fonctions concaves**

Soit maintenant une fonction $g : H \to [-\infty, +\infty[$.
La valeur $+\infty$ est exclue alors que la valeur $-\infty$ est possible.

(b1) Le domaine de g est l'ensemble

$$dom(g) = \{x \in H,\ g(x) > -\infty\}$$

(b2) L'épigraphe de g est l'ensemble

$$epig = \{(x, c) \in H \times \mathbb{R},\ g(x) \geq c\}$$

(b3) La fonction g est dite
– concave si la fonction $-g : H \to]-\infty, +\infty]$ est convexe.
– a-fortement concave ($a > 0$) si la fonction $-g$ est a-fortement convexe.

(b4) La fonction concave g est dite propre si la fonction convexe $-g$ est propre, c'est-à-dire s'il existe $x \in H$ tel que $g(x) > -\infty$ ($dom(g) \neq \phi$).

(b5) La fonction g est dite semi-continue supérieurement (s.c.s.) si $-g$ est s.c.i., autrement dit, si pour tout $c \in \mathbb{R}$, l'ensemble

$$\{x \in H,\ g(x) \geq c\}$$

est un ensemble fermé.

(b6) Soit f une fonction concave et propre. Si f est s.c.s. alors on dit que f est fermée.

(b7) Pour chaque $y_0 \in H$, l'ensemble $\partial g(y_0)$ est défini comme étant l'ensemble des $y \in H$ tels que :

$$g(y') \leq g(y_0) + \langle y, y' - y_0 \rangle\,, \quad \forall y' \in H$$

Les éléments y de $\partial g(y_0)$ sont appelés sous-gradients de g au point y_0, et la multifonction $y_0 \to \partial g(y_0)$ est appelée sous-différentiel de g, bien qu'on aurait pu utiliser les termes super-gradients et super-différentiel.
Remarquons que $\quad \partial g(y) = -\partial(-g)(y)$

(c) **Fonctions selle**

Considérons deux espaces de Hilbert H_1 et H_2, chacun muni d'un produit scalaire que nous notons $\langle , \rangle$ pour simplifier et L une fonction de $H_1 \times H_2$ à valeurs dans $[-\infty, +\infty]$.

(c1) Un point selle de L est un point (x_0, y_0) de $H_1 \times H_2$ caractérisé par

$$L(x_0, y) \leq L(x_0, y_0) \leq L(x, y_0), \ \forall (x, y) \in H_1 \times H_2$$

c'est-à-dire :

$$L(x_0, y_0) = \inf_{x \in H_1} L(x, y_0) = \sup_{y \in H_2} L(x_0, y)$$

(c2) On dit que L est une fonction selle si elle est convexe-concave, c'est-à-dire si,
la fonction $L_y : x \to L(x, y)$ est convexe $\forall y \in H_2$ et
la fonction $L_x : y \to L(x, y)$ est concave $\forall x \in H_1$.

(c3) On dit que L est une fonction a-fortement convexe-concave ($a > 0$), si $L(x, y)$ est a-fortement convexe en x et a-fortement concave en y.

(c4) Soit L une fonction selle sur $H_1 \times H_2$. Définisons

$$dom_1(L) = \{x \in H_1 : L(x, y) < +\infty, \ \forall y \in H_2\} = \bigcap_{y \in H_2} dom(L_y)$$

$$dom_2(L) = \{y \in H_2 : L(x, y) > -\infty, \ \forall x \in H_1\} = \bigcap_{x \in H_1} dom(L_x)$$

$dom_1(L)$ est une partie convexe de H_1.
$dom_2(L)$ est une partie convexe de H_2.

Le produit cartésien (convexe)

$$dom(L) = dom_1(L) \times dom_2(L)$$

est appelé domaine effectif de L, puisque

$$\forall x \in dom_1(L),\ \forall y \in dom_2(L),\ -\infty < L(x,y) < +\infty$$

et donc L est finie partout sur $dom(L)$.

(c5) La fonction L est dite propre si $dom(L) \neq \phi$.

(c6) Soit L une fonction selle propre sur $H_1 \times H_2$. On dit que L est fermée si,
la fonction $x \to L(x,y)$ est s.c.i. sur H_1 $\forall y \in H_2$,
et la fonction $y \to L(x,y)$ est s.c.s. sur H_2 $\forall x \in H_1$.

(c7) Soit L une fonction selle sur $H_1 \times H_2$ et (x_0, y_0) un élément de $H_1 \times H_2$. Définissons

$$\partial_1 L(x_0, y_0) = \partial_x L(x_0, y_0)$$

comme étant l'ensemble des sous-gradients de la fonction convexe $L(., y_0)$ en x, i.e. l'ensemble des vecteurs x de H_1 tels que

$$L(x', y_0) \geq L(x_0, y_0) + \langle x, x' - x_0 \rangle\,, \quad \forall x' \in H_1$$

De façons similaire, nous définissons

$$\begin{aligned} \partial_2 L(x_0, y_0) &= \partial_y L(x_0, y_0) \\ &= \{y \in H_2 : L(x_0, y') \leq L(x_0, y_0) + \langle y, y' - y_0 \rangle\,,\ \forall y' \in H_2\} \end{aligned}$$

comme étant l'ensemble des sous-gradients de la fonction concave $L(x_0, .)$ en y.
Les éléments de l'ensemble

$$\partial L(x_0, y_0) = \partial_1 L(x_0, y_0) \times \partial_2 L(x_0, y_0)$$

sont alors définis comme étant les sous-gradients de L en (x_0, y_0). Et l'application multivoque $\partial L : (x_0, y_0) \to \partial L(x_0, y_0)$ est appelée sous-différentiel de L. Les éléments de $\partial L(x_0, y_0)$ sont les vecteurs $(x, y) \in H_1 \times H_2$ vérifiants :

$$\begin{cases} L(x', y_0) - \langle x, x' - x_0 \rangle \geq L(x_0, y_0) \geq L(x_0, y') - \langle y, y' - y_0 \rangle \\ \forall (x', y') \in H_1 \times H_2 \end{cases}$$

Notons que pour tout $(x_0, y_0) \in H_1 \times H_2$, $\partial L(x_0, y_0)$ est un ensemble convexe fermé de $H_1 \times H_2$ qui peut être vide.

1.2 Notions d'opérateurs maximaux monotones

Beaucoup de problèmes (problèmes de minimisation de fonctionnelles, problèmes de minimax, problèmes des inégalités variationnelles...) peuvent se ramener à la détermination des solutions (ou zéros) d'un opérateur multivoque maximal monotone A, c'est-à-dire des éléments x de H satisfaisant $0 \in Ax$, d'où l'utilité de ces opérateurs.

Les définitions et propriétés que nous allons citer dans cette section, le lecteur pourra les retrouver dans Brézis[6] et Deimling[11].

(a) Un opérateur multivoque sur H est une application $A : H \to 2^H$.
Donc, pour chaque $x \in H$, Ax représente une partie de H qui peut être vide.

Le domaine de A est l'ensemble

$$D(A) = \{x \in H : Ax \neq \phi\}$$

L'image de A est l'ensemble

$$R(A) = \{Ax,\ x \in D(A)\} = \bigcup_{x \in D(A)} Ax$$

Le graphe de A est l'ensemble

$$G(A) = \{(x, y) : x \in D(A) \ et \ y \in Ax\} \subset H \times H$$

qui sera identifié à l'opérateur A.

Si pour tout $x \in H$, l'ensemble Ax contient au plus un élément, on dira que A est univoque.

L'inverse de A est l'opérateur A^{-1} dont le graphe est le symétrique du graphe de A, i.e.

$$(x, y) \in G(A^{-1}) \iff (y, x) \in G(A)$$

et on a

$$D(A^{-1}) = R(A).$$

Si A et B sont deux opérateurs de H, α et β deux réels, alors on définit $\alpha A + \beta B$ comme étant l'opérateur

$$x \to \alpha Ax + \beta By = \{\alpha u + \beta v : u \in Ax \ et \ v \in By\}$$

(b) Un opérateur univoque A sur H est dit k-lipschitzien ou lipschitzien de rapport k ($k > 0$) par rapport à une norme $\|..\|$ si

$$\forall x, y \in D(A), \ \|Ax - Ay\| \leq k\|x - y\|$$

– si $k = 1$, alors A est dit non-expansif.
– si $k < 1$, on dit que A est contractif ou contraction de rapport k.

(c) Un opérateur linéaire A défini sur H est dit :
– symétrique si

$$\forall x, y \in H, \ \langle y, Ax\rangle = \langle Ay, x\rangle .$$

– semi-défini positive ou tout simplement positif si

$$\forall x \in H, \ \langle Ax, x\rangle \geq 0.$$

(d) Un opérateur multivoque A de H est dit :
– monotone si

$$\forall x, x' \in D(A),\ \langle y - y', x - x' \rangle \geq 0\ \forall y \in Ax,\ \forall y' \in Ax'$$

– strictement monotone si

$$\forall x, x' \in D(A),\ \langle y - y', x - x' \rangle > 0\ \forall y \in Ax,\ \forall y' \in Ax'$$

– a-fortement monotone ($a > 0$) si

$$\forall x, x' \in D(A),\ \langle y - y', x - x' \rangle \geq a\|x - x'\|^2\ \forall y \in Ax,\ \forall y' \in Ax'$$

Exemples d'opérateurs monotones :

(d1) Si A est un opérateur monotone, alors les opérateurs λA ($\lambda > 0$) et A^{-1} sont monotones.

(d2) Si F est une application non-expansive sur H, alors l'opérateur $I - F$ est monotone (I étant l'identité de H).

(d3) Si A et B sont deux opérateurs monotones alors la somme $A + B$ est un opérateur monotone.

(d4) Si $f : H \to]-\infty, +\infty]$ est une fonction convexe propre, alors le sous-différentiel ∂f est monotone dans H.

(d5) T est a-fortement monotone ($a > 0$) $\iff$ l'opérateur $T' = T - aI$ est monotone.

(e) Soient A et B deux opérateurs monotones de H. Alors

$$G(A) \subset G(B) \iff \begin{cases} D(A) \subset D(B) \\ \forall x \in D(A),\ Ax \subset Bx \end{cases}$$

(f) Un opérateur A monotone est dit maximal monotone s'il est maximal dans l'ensemble des graphes monotones par l'inclusion, c'est-à-dire si le graphe $G(A)$ n'est contenu dans le graphe d'aucun autre opérateur monotone $B : H \to 2^H$.

En général, pour montrer qu'un opérateur monotone A est maximal, nous prenons un $(x, y) \in H \times H$ tel que $\langle y - \beta, x - \alpha \rangle \geq 0,\ \forall (\alpha, \beta) \in G(A)$ et nous montrons que $x \in D(A)$ et $y \in Ax$.

Exemples d'opérateurs maximaux monotones :

(f1) Si A est un opérateur maximal monotone, alors les opérateurs A^{-1} et λA $(\lambda > 0)$ sont maximaux monotones.

(f2) La somme de deux opérateurs A et B maximaux monotones n'est pas forcément maximal monotone.

(f3) Le sous-différentiel ∂f d'une fonction convexe propre et s.c.i. sur H est un opérateur maximal monotone (Rockafellar[23]).

(f4) Un opérateur univoque A monotone et hémi-continue[2] est maximal monotone.

La proposition suivante donne une caractérisation des opérateurs maximaux monotones.

Proposition 1.1
Soit A un opérateur de H. Alors les trois propriétés suivantes sont équivalentes :

(i) A est maximal monotone.

(ii) A est monotone et $R(I + A) = H$.

(iii) $\forall \lambda > 0$, l'application $A_\lambda = (I + \lambda A)^{-1}$ est non-expansive définie sur H tout entier.

Preuve.
Voir Brézis[6].

Nous verrons dans le chapitre 3 d'autres exemples fondamentaux d'opérateurs maximaux monotones que nous allons utiliser comme applications. Ces opérateurs sont construits pour résoudre les problèmes de minimisation

[2] C'est-à-dire vérifiant $\lim_{t \to 0^+} \langle A(x + ty), h \rangle = \langle Ax, h \rangle\ ,\ \forall x, y, h \in H$.

de fonctionnelle et bien d'autres problèmes liés à l'analyse convexe. Le lecteur intéréssé trouvera ces exemples dans Rockafellar[24], [25] et [26].

1.3 Algorithmes asynchrones

Les algorithmes asynchrones sont utilisés pour traiter de façon parallèle des problèmes qui prennent en considération l'interaction de plusieurs processeurs. Ces algorithmes apparaissent comme une extension naturelle de la méthode des approximations successives :

$$x^{p+1} = g(x^p),\ x^0 \in \mathbb{R}^n,\ p = 0, 1, ...$$

où chaque x^p est un vecteur à n composantes et g est une fonction de $\mathbb{R}^n$ vers lui-même.

Dans le calcul itératif parallèle, au cours d'une itération chaque processeur execute la tache qui lui a été confiée (phase de calcul ou d'activité) pour ensuite communiquer ses résultats aux autres processeurs afin de les utiliser dans le calcul de l'itération suivante (phase de communication). Selon le mode utilisé pour effectuer ses deux phases, on distingue trois classes d'algorithmes itératifs parallèles (voir Bahi et al.[4]) :

1. Algorithmes avec itérations synchrones et communications synchrones (ISCS) :
   Pour cette classe d'algorithmes, les processeurs commencent leurs itérations en même temps et les échanges de données sont réalisés à la fin d'une itération. Le principal inconvénient de ces algorithmes vient des communications synchrones qui pénalisent fortement les performances. Comme on peut le voir sur la figure 1.1, il peut y avoir beaucoup de temps morts entre les itérations selon les performances des processeurs et la vitesse des communications. Après parallélisation, les itérations de ces algorithmes sont identiques à celles réalisées par les algorithmes séquentiels. La convergence est également la même que dans le cas séquentiel. Par contre les communications synchrones représentées par des flèches sur la figure précédente peuvent être fortement pénalisantes.

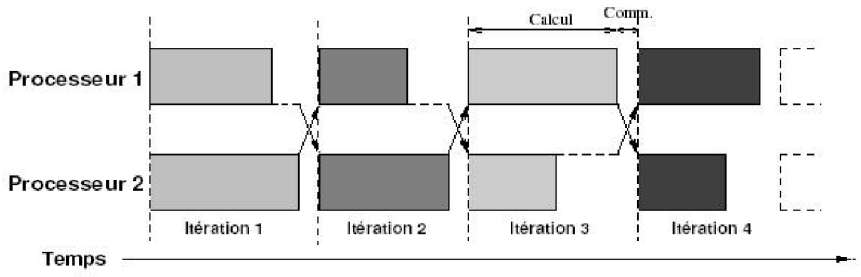

FIG. 1.1: Exemple d'algorithme ISCS.

2. Les algorithmes avec itérations synchrones et communications asynchrones (ISCA) :
   Pour cette classes d'algorithme, les processeurs attendent d'avoir reçu les données nécessaires mises à jour à l'itération précédente pour commencer la suivante. Par contre, les données (ou groupes de données) nécessaires à un autre processeur sont envoyées dès que possible de manière asynchrone pour obtenir un recouvrement des communications par les calculs restants à faire dans l'itération courante. Ce schéma compte sur une bonne probabilité que les données soient reçues sur le processeur destination avant qu'il ait fini son itération courante. Ainsi, il n'aura pas de délai d'attente pour commencer son itération suivante. Ce recouvrement partiel des communications par du calcul implique moins de temps morts entre les itérations et donc de meilleures performances globales. Ce système de communication implique aussi que les processeurs peuvent ne pas commencer leurs itérations tous en même temps. Néanmoins, la notion de synchronisme reste valable en termes d'itérations puisqu'à tout instant t, il n'est pas possible d'avoir deux processeurs effectuant des itérations différentes. Ils sont soient en train de calculer la même itération, soit en train d'attendre les données pour la suivante. Malgré tout, ce schéma n'élimine pas tous les temps morts (voir figure 1.2) puisque le recouvrement n'est pas possible pour les dernières données calculées sur le processeur le plus en retard. Enfin, comme la précédente, cette classe d'algorithmes a aussi le même comportement que l'équivalent synchrone en ce qui concerne les calculs effectués et donc les mêmes propriétés de convergence.

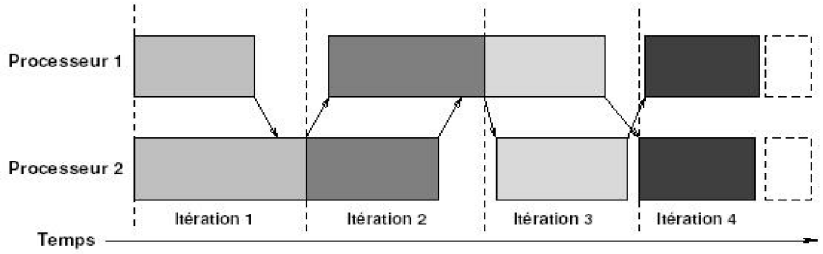

FIG. 1.2: Exemple d'algorithme ISCA.

3. Les algorithmes avec itérations asynchrones et communications asynchrones (IACA) :
   Dans cette troisième et dernière classes d'algorithme, qui est la plus générale, les processeurs effectuent leurs itérations sans tenir compte de l'avancement des autres. Il n'y a plus d'attente des données venant d'autres processeurs pour commencer une itération, chaque processeur effectue ses calculs en utilisant les dernières valeurs connues de chaque donnée. Comme il n'y a plus d'attente dues aux communications, il n'y a plus non plus de temps morts entre les itérations, comme on peut le voir sur la figure 1.3. Contrairement aux deux classes précédentes, les itérations effectuées ne sont plus forcément les mêmes que dans l'équivalent séquentiel puisque celles-ci vont dépendre des instants où sont reçues et prises en compte les données. La figure 1.3 montre bien ce phénomène : il n'y a plus aucune période d'inactivité en terme de calcul et on remarque également qu'à un même instant t, les deux processeurs peuvent être en train de calculer une itération différente.
   Certains problèmes spécifiques apparaissent alors tels que la détection de convergence et la procédure d'arrêt de l'algorithme. Il y a aussi des problèmes liés à l'implantation qui implique l'utilisation d'un environnement de programmation adéquat. Néanmoins, ces algorithmes restent assez pratiques à mettre en oeuvre et se révèlent surtout les plus performants particulièrement dans un contexte de calcul sur la grille.

Notons par α le nombre de processeurs utilisés ($\alpha \in \mathbb{N}^*$), et décomposons l'espace $\mathbb{R}^n$ sous forme de produit $\prod_{i=1}^{\alpha} \mathbb{R}^{n_i}$ où n_i sont des entiers naturels

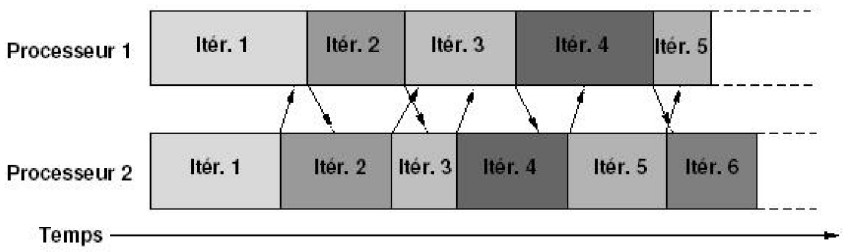

FIG. 1.3: Exemple d'algorithme IACA

non nuls et $n = \sum_{i=1}^{\alpha} n_i$. Tous les vecteurs $x \in \mathbb{R}^n$ considérés seront alors décomposés sous la forme

$$x = (x_1, ..., x_\alpha) \text{ avec } x_i \in \mathbb{R}^{n_i}$$

Comme exemple, supposons qu'on travaille avec des vecteurs de $\mathbb{R}^{10}$ à dix composantes sur une machine comportant seulement quatre processeurs. Alors une décomposition possible de $\mathbb{R}^{10}$ est $\mathbb{R}^{10} = \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{R}^2$. Et donc chaque vecteur x de $\mathbb{R}^{10}$ s'écrira comme $x = (x_1, x_2, x_3, x_4)$ où $x_1 \in \mathbb{R}^3,\ x_2 \in \mathbb{R}^3,\ x_3 \in \mathbb{R}^2$ et $x_4 \in \mathbb{R}^2$.

Munissons chaque $\mathbb{R}^{n_i}$ du produit scalaire euclidien $\langle ., . \rangle_i$ et de la norme associée $\|..\|_i = \langle ., . \rangle_i^{1/2}$ de telle façon que l'espace $\mathbb{R}^n$ soit muni du produit scalaire

$$\langle x, y \rangle = \sum_{i=1}^{\alpha} \langle x_i, y_i \rangle_i \text{ où } x, y \in \mathbb{R}^n$$

et de la norme associée

$$\|x\| = \langle x, x \rangle^{1/2} = (\sum_{i=1}^{\alpha} \|x_i\|_i^2)^{1/2},\ x \in \mathbb{R}^n$$

Nous munissons aussi l'espace $\mathbb{R}^n$ de la norme uniforme définie par

$$\|x\|_\infty = \max_{1 \le i \le \alpha} \|x_i\|_i\ ,\ x \in \mathbb{R}^n$$

Considérons une application F de $\mathbb{R}^n$ vers $\mathbb{R}^n$.
Pour chaque $x \in \mathbb{R}^n$, $F(x)$ est donc un vecteur de $\mathbb{R}^n$ qui s'écrit sous la forme

$$F(x) = (F_1(x), F_2(x), ..., F_\alpha(x)),$$

$$\text{avec } F_i(x) \in \mathbb{R}^{n_i} \text{ pour } i = 1, 2, ..., \alpha$$

Les itérations asynchrones telles qu'elles sont proposées par Baudet[5] sont alors définies de la façon suivante :
Soient, $J = \{J(p)\}_{p\in\mathbb{N}}$ une suite de sous ensembles non vides de $\{1, ..., \alpha\}$ appelée stratégie et $S = \{(s_1(p), ..., s_\alpha(p))\}_{p\in\mathbb{N}}$ une suite de $\mathbb{N}^\alpha$ définissant les retards.

Pour un vecteur initial $x^0 = (x_1^0, ..., x_\alpha^0) \in \mathbb{R}^n$, la suite $\{x^p\}_{p\in\mathbb{N}}$ est défini par,

$$\begin{cases} x_i^{p+1} = \begin{cases} F_i(x_1^{s_1(p)}, ..., x_\alpha^{s_\alpha(p)}) & si \quad i \in J(p) \\ x_i^p & si \quad i \notin J(p) \end{cases} \\ i = 1, ..., \alpha \\ p = 0, 1, .. \end{cases} \tag{1.1}$$

(Voir Baudet[5], El Tarazi[13] et introduction).

On l'appelle algorithme asynchrone associé à F et il sera noté (F, x^0, J, S).

(a) Cet algorithme décrit le comportement d'un processus itératif exécuté de façon asynchrone sur une machine parallèle comportant α processeurs. A chaque itération $p+1$, le $i^{ème}$ processeur calcule x_i^{p+1} en utilisant (1.1).

$J(p)$ est l'ensemble des composantes relaxées à l'itération p.

$p - s_i(p)$ correspond au retard éventuel dû au $i^{ème}$ processeur lors du calcul du $i^{ème}$ bloc à l'itération p.

(b) Dans le cas des retards nuls :

$$s_i(p) = p, \ \forall p \in \mathbb{N}, \ \forall i \in \{1, ..., \alpha\}$$

(1.1) décrit le comportement d'un algorithme synchrone (sans retard). Pendant chaque itération $p + 1$, chaque processeur exécute un nombre

de calculs qui dépend des résultats des calculs des autres processeurs à l'itération p. A l'intérieur de chaque itération, chaque processeur n'entre pas en interaction avec les autres processeurs, toutes les interactions auront lieu à la fin des itérations.
Si de plus, $\forall p \in \mathbb{N}$:

(b1) $J(p) = \{1, ..., \alpha\}$ alors (1.1) modélise l'algorithme de Jacobi par blocs.

(b2) $J(p) = \{p \ mod \ \alpha + 1\}$ alors (1.1) modélise l'algorithme de Gauss-Seidel par blocs.

Pour plus de details sur les algorithmes asynchrones voir [3], [5], [18], [13] et [9]. Le lecteur trouvera dans [19] une collection d'un grand nombre de papiers traitant des itérations asynchrones (résultats théoriques et applications).

Chapitre 2

Convergence de l'algorithme général

2.1 Introduction

Ce chapitre est consacré à l'étude de la convergence des algorithmes itératifs de relaxation asynchrones pour résoudre un problème général de point fixe. La modélisation de ces algorithmes asynchrones a été développée successivement par Chazan et Miranker [10] dans le cadre linéaire, Miellou [17], Baudet [5], Bertsekas et Tsitsiklis [8] dans le cadre non linéaire ; citons également les travaux de F. Robert ([21],[22]) dans le cas synchrone. La modélisation de ces algorithmes est effectuée en introduisant une stratégie de choix des composantes pour rendre compte du prallélisme ainsi qu'une notion de retards pour rendre compte de l'asynchronisme entre les processeurs (voir introduction).

2.2 Le cas asynchrone

Dans cette section, nous allons démontrer la convergence de l'algorithme parallèle asynchrone (1.1) avec des retards bornés vers un point fixe d'un opérateur F non linéaire non-expansif de $\mathbb{R}^n$ vers $\mathbb{R}^n$ relativement à la norme uniforme $\|..\|_\infty$ sur $\mathbb{R}^n$.

Théorème 2.1
Supposons
(h_0) $\exists$ *une sous-suite* $\{p_k\}_{k\in\mathbb{N}}$ *telle que,* $J(p_k) = \{1, ..., \alpha\}$ *et* $\forall i \in \{1, ..., \alpha\}$ $s_i(p_k) = p_k$
(h_1) $\exists s \in \mathbb{N}$, *tel que,* $\forall i \in \{1, ..., \alpha\}$, $\forall p \in \mathbb{N}$, $p - s \leq s_i(p) \leq p$
(h_2) $\exists u \in \mathbb{R}^n$, $F(u) = u$
(h_3) $\forall x, x' \in \mathbb{R}^n$, $\|F(x) - F(x')\|_\infty \leq \|x - x'\|_\infty$
(h_4) $\forall x, x' \in \mathbb{R}^n, \|F(x) - F(x')\|^2 \leq \langle F(x) - F(x'), x - x'\rangle$
Alors pour tout $x^0 \in \mathbb{R}^n$, *les itérations (1.1) convergent vers* x^* *point fixe de* F.

Preuve.
Nous procédons en trois étapes :

(i) Nous montrons en premier lieu que la suite $\{\|x^p - u\|_\infty\}_{p\in\mathbb{N}}$ est convergente. Pour cela, pour $p \in \mathbb{N}$, considérons les $(s+1)$ itérés $x^p, x^{p-1}, ..., x^{p-s}$ du processus et posons

$$z^p = \max_{0\leq l\leq s} \|x^{p-l} - u\|_\infty = \max_{p-s\leq l\leq p} \|x^l - u\|_\infty$$

Alors, $\forall i \in \{1, ..., \alpha\}$ on a :
ou bien $i \notin J(p)$ et donc

$$\begin{aligned} \|x_i^{p+1} - u_i\|_i &= \|x_i^p - u_i\|_i \\ &\leq \|x^p - u\|_\infty \\ &\leq \max_{0\leq l\leq s} \|x^{p-l} - u\|_\infty \\ &= z^p \end{aligned}$$

ou bien $i \in J(p)$ et donc

$$\begin{aligned}
\|x_i^{p+1} - u_i\|_i &= \|F_i(x_1^{s_1(p)}, ..., x_\alpha^{s_\alpha(p)}) - F_i(u)\|_i \\
&\leq \|F(x_1^{s_1(p)}, ..., x_\alpha^{s_\alpha(p)}) - F(u)\|_\infty \\
&\leq \|(x_1^{s_1(p)}, ..., x_\alpha^{s_\alpha(p)}) - u\|_\infty \text{ (grâce à } (h_3)) \\
&= \|x_j^{s_j(p)} - u_j\|_j \quad (1 \leq j \leq \alpha) \\
&\leq \|x^{s_j(p)} - u\|_\infty \\
&\leq \max_{p-s \leq l \leq p} \|x^l - u\|_\infty \quad (car\ p - s \leq s_j(p) \leq p) \\
&= z^p
\end{aligned}$$

donc

$$\forall i \in \{1, ..., \alpha\}\,,\ \|x_i^{p+1} - u_i\|_i \leq z^p$$

c'est-à-dire

$$\|x^{p+1} - u\|_\infty \leq z^p$$

par conséquent

$$\begin{aligned}
z^{p+1} &= \max_{0 \leq l \leq s} \|x^{p+1-l} - u\|_\infty \\
&= Max \left\{ \max_{0 \leq l \leq s-1} \|x^{p-l} - u\|_\infty \ ,\ \|x^{p+1} - u\|_\infty \right\} \\
&\leq z^p
\end{aligned}$$

ce qui prouve que la suite $\{z^p\}_{p \in \mathbb{N}}$ est décroissante (positive) donc convergente. Sa limite est

$$\begin{aligned}
\lim_{p \to \infty} z^p &= \lim_{p \to \infty} \max_{0 \leq l \leq s} \|x^{p-l} - u\|_\infty \\
&= \lim_{p \to \infty} \|x^{p-j(p)} - u\|_\infty \quad (0 \leq j(p) \leq s) \\
&= \lim_{p \to \infty} \|x^p - u\|_\infty
\end{aligned}$$

ceci prouve que la suite $\{\|x^p - u\|_\infty\}_{p\in\mathbb{N}}$ est convergente.

(ii) Dans *(i)* nous avons montré en particulier que la suite $\{x^p\}_{p\in\mathbb{N}}$ est bornée. Prenons alors la sous-suite $\{p_k\}_{k\in\mathbb{N}}$ définie par (h_0). Comme la suite $\{x^{p_k}\}_{k\in\mathbb{N}}$ est bornée, elle admet une sous-suite notée aussi $\{x^{p_k}\}_{k\in\mathbb{N}}$ qui converge vers un x^* de $\mathbb{R}^n$. Montrons alors que x^* est un point fixe de F. Pour cela considérons la suite

$$y^p = x^p - F(x^p),\ p \in \mathbb{N}$$

et montrons que $\lim\limits_{k\to\infty} y^{p_k} = 0$.

$$\begin{aligned}
\|x^{p_k} - u\|^2 &= \|y^{p_k} + F(x^{p_k}) - u\|^2 \\
&= \|y^{p_k}\|^2 + \|F(x^{p_k}) - u\|^2 + 2\left\langle F(x^{p_k}) - u, y^{p_k}\right\rangle
\end{aligned}$$

d'où

$$\|y^{p_k}\|^2 = \|x^{p_k} - u\|^2 - \|F(x^{p_k}) - u\|^2 - 2\left\langle F(x^{p_k}) - u, y^{p_k}\right\rangle$$

or

$$\begin{aligned}
\left\langle F(x^{p_k}) - u, y^{p_k}\right\rangle &= \left\langle F(x^{p_k}) - F(u), x^{p_k} - F(x^{p_k})\right\rangle \\
&= \left\langle F(x^{p_k}) - F(u), [x^{p_k} - F(u)] - [F(x^{p_k}) - F(u)]\right\rangle \\
&= \left\langle F(x^{p_k}) - F(u), x^{p_k} - u\right\rangle - \|F(x^{p_k}) - F(u)\|^2 \\
&\geq 0 \text{ (grâce à } (h_4))
\end{aligned}$$

d'où,

$$\begin{aligned}
\|y^{p_k}\|^2 &\leq \|x^{p_k} - u\|^2 - \|F(x^{p_k}) - u\|^2 \\
&= \|x^{p_k} - u\|^2 - \|x^{p_k+1} - u\|^2 \text{ (grâce à } (h_0))
\end{aligned}$$

Or d'après (i) la suite $\{\|x^p - u\|_\infty\}_{p\in\mathbb{N}}$ est convergente, donc la suite $\{\|x^p - u\|\}_{p\in\mathbb{N}}$ est aussi convergente (normes équivalentes). Par consequent, toute sous-suite de $\{\|x^p - u\|\}_{p\in\mathbb{N}}$ est convergente et vers la même limite. D'où,

$$\begin{aligned}
\lim_{p\to\infty} \|x^p - u\| &= \lim_{k\to\infty} \|x^{p_k+1} - u\| \\
&= \lim_{k\to\infty} \|x^{p_k} - u\| \\
&= \|x^* - u\|
\end{aligned}$$

d'où on tire

$$\lim_{k\to\infty} \|y^{p_k}\| = 0$$

c'est-à-dire

$$\lim_{k\to\infty} y^{p_k} = 0$$

par conséquent

$$x^* - F(x^*) = 0$$

et donc x^* est bien un point fixe de F.

(iii) On démontre comme dans (i) que la suite $\{\|x^p - x^*\|_\infty\}_{p\in\mathbb{N}}$ est convergente, sa limite est

$$\lim_{p\to\infty} \|x^p - x^*\|_\infty = \lim_{k\to\infty} \|x^{p_k} - x^*\|_\infty = 0$$

Ce qui prouve bien que $x^p \to x^*$ pour la norme $\|..\|_\infty$. ■

Remarque 2.1
L'hypothèse (h_0) signifie que de temps à autre (après quelques itérations) les processeurs se synchronisent et mettent à jours leurs données en les échangeant. Cette sous-suite peut très bien être programmée par l'utilisateur.

Remarque 2.2
L'hypothèse (h_1) signifie que les retards dus aux communications entre processeurs et aux différents temps de calcul sont bornés, ce qui revient à supposer qu'au bout d'au plus $(s+1)$ itérations, tous les processeurs finissent par mettre à jour leurs données.

Remarque 2.3
L'hypothèse (h_4) est vérifiée par une large classe d'opérateurs, nous citons par exemple la résolvante $F_\lambda = (I+\lambda T)^{-1}$ (où $\lambda > 0$) associée à un opérateur T maximal monotone (voir lemme 2.3 ci-dessous). Aussi, la projection p_c d'un espace de Hilbert réel H sur un convexe fermé non vide C, c'est-à-dire pour $x \in H$, $p_c(x)$ est l'unique élément de C satisfaisant :

$$\|x - p_c(x)\| = \inf_{y\in C} \|x - y\|$$

Pour la preuve, voir Phelps[20], Examples 1.2.(f) et Deimling[11], proposition 9.2. Dans le cas linéaire, nous pouvons citer par exemple les opérateurs linéaires symétriques positifs et non-expansifs, comme le montre la proposition suivante :

Proposition 2.2
Soit A un opérateur linéaire symétrique positif et non-expansif défini sur un espace de Hilbert réel H. Alors A vérifie l'hypothèse (h_4).

Preuve.

(i) L'opérateur $B = I - A$ est symétrique. En effet, $\forall x, y \in H$

$$\begin{aligned} \langle Bx, y\rangle &= \langle x - Ax, y\rangle = \langle x, y\rangle - \langle Ax, y\rangle \\ &= \langle x, y\rangle - \langle x, Ay\rangle = \langle x, y - Ay\rangle \\ &= \langle x, By\rangle \end{aligned}$$

(ii) L'opérateur B est positif. En effet, $\forall x \in H$

$$\langle Bx, x\rangle = \langle x - Ax, x\rangle = \|x\|^2 - \langle Ax, x\rangle \geq 0$$

car $\quad \langle Ax, x\rangle \leq \|Ax\|\|x\| \leq \|x\|^2.$

(iii) Les opérateurs A et B commutent. En effet,

$$AB = A(I - A) = A - A^2 = (I - A)A = BA$$

(iv) L'opérateur AB est symétrique. En effet, $\forall x, y \in H$

$$\langle ABx, y\rangle = \langle Bx, Ay\rangle = \langle x, BAy\rangle = \langle x, ABy\rangle$$

(v) L'opérateur AB est positif (voir preuve dans [15], théorème 10.7).

(vi) L'opérateur A vérifie l'hypothèse (h_4). En effet, $\forall x \in H$

$$\langle Ax, x\rangle - \|Ax\|^2 = \langle Ax, x - Ax\rangle = \langle Ax, Bx\rangle = \langle ABx, x\rangle \geq 0$$

■

2.3 Algorithme synchrone de Jacobi

Dans ce cas, nous supposons que pour tout $p \in \mathbb{N}$

$$\begin{cases} s_i(p) = p, \ \forall i \in \{1, ..., \alpha\} \\ J(p) = \{1, ..., \alpha\} \end{cases}$$

qui décrit un algorithme parallèle synchrone sans retards. Nous pouvons alors nous passer de l'hypothèse (h_3) comme le montre le théorème suivant :

Théorème 2.3
Sous les hypothèses (h_2), (h_4) *et*
$(h_0^{'})$ $\forall p \in \mathbb{N}, \ J(p) = \{1, ..., \alpha\}$ *et* $s_i(p) = p, \ \forall i \in \{1, ..., \alpha\}$
l'algorithme parallèle synchrone de Jacobi défini par

$$\begin{cases} x^0 = (x_1^0, ..., x_\alpha^0) \in \mathbb{R}^n \\ x_i^{p+1} = F_i(x_1^p, ..., x_\alpha^p) \\ i = 1, ..., \alpha \\ p = 1, 2... \end{cases} \tag{2.1}$$

converge dans $\mathbb{R}^n$ *vers* x^* *point fixe de* F.

Preuve.
(i) Grâce à (h_4), l'application F est non-expansive par rapport à la norme euclidienne. En utilisant (2.1), nous obtenons

$$\|x^{p+1} - u\| = \|F(x^p) - F(u)\| \leq \|x^p - u\|$$

Ce qui prouve que la suite $\{\|x^p - u\|\}_{p\in\mathbb{N}}$ est décroissante (positive) donc convergente, donc la suite $\{\|x^p - u\|_\infty\}_{p\in\mathbb{N}}$ est convergente. La suite (bornée) $\{x^p\}_{p\in\mathbb{N}}$ admet alors uns sous-suite notée $\{x^{p_k}\}_{k\in\mathbb{N}}$ qui converge vers x^* de $\mathbb{R}^n$.
Les étapes *(ii)* et *(iii)* sont similaires à celles du théorème 2.1. ■

2.4 Solution des opérateurs maximaux monotones

Les opérateurs maximaux monotones ont été étudiés de manière extensive pour leurs rôles dans l'analyse convexe (minimisation de fonctionnelles,

problèmes de minimax, ...) et dans quelques équations aux dérivées partièlles.

Dans cette section, nous appliquerons l'algorithme parallèle asynchrone au calcul des solutions des opérateurs maximaux fortement monotones et l'algorithme parallèle synchrone de Jacobi au calcul des solutions des opérateurs maximaux monotones. Ces résultats généraux concernant ces opérateurs, vont être appliqués dans le chapitre suivant lors de la résolution de quelques problèmes liés à l'analyse convexe.

Théorème 2.4
Soit T un opérateur multivoque maximal a-fortement monotone sur $\mathbb{R}^n$ $(a > 0)$. Alors

1. *T admet une unique solution x^*.*
2. *Tout algorithme parallèle asynchrone (1.1) à retards bornés associé à l'application univoque $F = (I + cT)^{-1}$ où $c \geq \frac{\sqrt{a}-1}{a}$ converge dans $\mathbb{R}^n$ vers la solution x^* de l'opérateur T.*

Preuve.
Elle est donnée sous forme de lemmes :

Lemme 2.1
Soit T un opérateur maximal monotone sur $\mathbb{R}^n$ et $F = (I + cT)^{-1}$, $(c > 0)$. Alors les solutions de T sont exactement les points fixes de F sur $\mathbb{R}^n$.

Preuve du lemme 2.1.

$$\begin{aligned} 0 \in Tx &\iff x \in (I + cT)x \\ &\iff x = (I + cT)^{-1}x \\ &\iff x = Fx \end{aligned}$$

■

Lemme 2.2
Soit T un opérateur maximal a-fortement monotone sur $\mathbb{R}^n$ $(a > 0)$ et $F = (I + cT)^{-1}$ $(c > 0)$. Alors

(a) F admet un point fixe unique x^*.

(b) *Pour* $c \geq \frac{\sqrt{\alpha}-1}{a}$, *l'application* F *est non-expansive par rapport à la norme* $\|..\|_\infty$ *sur* $\mathbb{R}^n$.

Preuve du lemme 2.2.

(a) Considérons $T' = T - aI$, $\beta = \frac{1}{1+ac}$ et $F' = (I + \beta c T')^{-1}$. Pour $x, y \in \mathbb{R}^n$, nous avons

$$\begin{aligned} y = F(x) &\iff x \in y + cTy \\ &\iff x \in (1+ac)y + c(T - aI)y \\ &\iff \beta x \in \beta(1+ac)y + \beta c T' y \\ &\iff \beta x \in (I + \beta c T')y \\ &\iff y = F'(\beta x) \end{aligned}$$

c'est-à-dire , $\forall x \in \mathbb{R}^n$, $F(x) = F'(\beta x)$.
Comme T est a-fortement monotone (maximal), l'opérateur T' est maximal monotone et donc l'application $F' = (I + \beta c T')^{-1}$ est non-expansive par rapport à la norme euclidienne $\|..\|$ sur $\mathbb{R}^n$ (proposition 1.1) . D'où $\forall x, x' \in \mathbb{R}^n$

$$\|F(x) - F(x')\| = \|F'(\beta x) - F'(\beta x')\| \leq \beta \|x - x'\|$$

Comme $\beta = \frac{1}{1+ac} < 1$, l'application F est une contraction sur $\mathbb{R}^n$ et donc admet un point fixe unique x^* (théorème du point fixe de Banach) qui sera la solution de l'opérateur T d'après le lemme 2.1.

(b) La norme euclidienne et la norme uniforme sur $\mathbb{R}^n$ sont équivalentes suivant la relation :

$$\|x\|_\infty \leq \|x\| \leq \sqrt{\alpha}\|x\|_\infty \ , \ \forall x \in \mathbb{R}^n$$

Alors, $\forall x, x' \in \mathbb{R}^n$

$$\|F(x) - F(x')\| \leq \beta \|x - x'\|$$

entraîne

$$\begin{aligned} \|F(x) - F(x')\|_\infty &\leq \beta\sqrt{\alpha}\|x - x'\|_\infty \\ &= \tfrac{\sqrt{\alpha}}{1+ac}\|x - x'\|_\infty \end{aligned}$$

Il suffit donc de choisir c tel que

$$\frac{\sqrt{\alpha}}{1+ac} \leq 1$$

c'est-à-dire

$$c \geq \frac{\sqrt{\alpha}-1}{a} \tag{2.2}$$

■

Lemme 2.3
Soit T un opérateur maximal monotone et $F = (I + cT)^{-1}$ $(c > 0)$. Alors F vérifie l'hypothèse (h_4).

Preuve du lemme 2.3.
Prenons x, x' dans $\mathbb{R}^n$ et montrons que

$$\|F(x) - F(x')\|^2 \leq \langle F(x) - F(x'), x - x' \rangle$$

Soient $y = F(x)$ et $y' = F(x')$ donc,

$$\begin{cases} x \in y + cTy \\ x' \in y' + cTy' \end{cases}$$

c'est-à-dire

$$\begin{cases} x - y \in cTy \\ x' - y' \in cTy' \end{cases}$$

Comme $c > 0$, l'opérateur cT est monotone, et donc

$$\langle (x - y) - (x' - y'), y - y' \rangle \geq 0$$

d'où

$$\langle x - x', y - y' \rangle - \|y - y'\|^2 \geq 0$$

c'est-à-dire

$$\|F(x) - F(x')\|^2 \leq \langle F(x) - F(x'), x - x' \rangle$$

et le théorème est entièrement démontré. ■

Remarque 2.4
Pour $x \in \mathbb{R}^n$, l'image $F(x) = (I + cT)^{-1}x$ doit être vue comme étant l'unique élément y de $\mathbb{R}^n$ vérifiant $x \in y + cTy$, c'est-à-dire vérifiant $c^{-1}(x - y) \in Ty$.

Remarque 2.5
La condition de monotonie forte sur l'opérateur T a assuré l'existence d'un point fixe de F et la non-expansivité de l'application $F = (I + cT)^{-1}$ $(c > 0)$ par rapport à la norme $\|..\|_\infty$. Dans le cas de l'algorithme parallèle de Jacobi, on peut se passer de cette hypothèse et ajouter l'hypothèse (h_2) d'existence de solutions. Nous pouvons alors énoncer :

Théorème 2.5
Soit T un opérateur multivoque maximal monotone sur $\mathbb{R}^n$ tel que $T^{-1}0 \neq \phi$. Alors tout algorithme parallèle synchrone de Jacobi (2.1) associé à l'application univoque $F = (I + cT)^{-1}$ où $c > 0$ quelconque, converge dans $\mathbb{R}^n$ vers x^* solution du problème $0 \in Tx$.

Chapitre 3

Applications

Dans ce chapitre, nous appliquerons les résultats du chapitre précedent, notamment les théorèmes 2.4 et 2.5 à quelques problèmes liés à l'analyse convexe.

3.1 Minimum d'une fonctionnelle

Commençons par cette proposition qui fournit une caractérisation des fonctions fortement convexes.

Proposition 3.1
Soit $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ convexe propre et s.c.i. Alors les conditions suivantes sont équivalentes :

(a) f est a-fortement convexe.

(b) ∂f est a-fortement monotone ($a > 0$).

(c) pour $y \in \partial f(x)$ on a pour tout $x' \in \mathbb{R}^n$:

$$f(x') \geq f(x) + \langle y, x' - x \rangle + \frac{1}{2} a \|x' - x\|^2$$

Preuve.
Voir Rockafellar[26], proposition 6. ■
Nous pouvons alors énoncer :

Corollaire 3.2
Soit $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ a-fortement convexe propre et s.c.i.[1] *Alors*

1. *f admet un minimum unique x^*.*
2. *Tout algorithme parallèle asynchrone (1.1) à retards bornés associé à l'application $F = (I + c\partial f)^{-1}$ où $c \geq \frac{\sqrt{\alpha}-1}{a}$ converge vers x^* le point de minimum de f sur $\mathbb{R}^n$.*

Preuve.
En remarquant que

$$0 \in \partial f(x_0) \iff f(x) \geq f(x_0)\ \forall x \in \mathbb{R}^n$$

$$\iff f(x_0) = \min_{x \in \mathbb{R}^n} f(x)$$

nous déduisons que les solutions de l'opérateur ∂f sont exactement les points de minimum globaux de f.
Le sous-différentiel ∂f est maximal (voir exemple (f3) page 22) a-fortement monotone (proposition 3.1). On applique alors le théorème 2.4 à l'opérateur ∂f pour conclure. ■

Remarque 3.1
Dans le cas où f est convexe (non nécessairement fortement convexe) propre et s.c.i., le sous-différentiel ∂f est maximal monotone. On peut alors appliquer le théorème 2.5 à l'application $F = (I + c\partial f)^{-1}$ ($c > 0$ quelconque), nous obtenons alors,

Corollaire 3.3
Soit $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ convexe propre et s.c.i. telle que le problème de minimisation $\min_{\mathbb{R}^n} f(x)$ admette une solution. Alors tout algorithme parallèle synchrone de Jacobi (2.1) associé à l'application $F = (I + c\partial f)^{-1}$ (où $c > 0$ quleconque) converge vers x^* point de minimum de f sur $\mathbb{R}^n$.

[1] Voir les définitions (a3), (a4) et (a5) dans la section 1.1

3.2 Point selle

Dans ce paragraphe, nous allons appliquer le théorème 2.4 au calcul de point selle d'une fonction selle $L : \mathbb{R}^n \times \mathbb{R}^m \to [-\infty, +\infty]$. Nous rappelons qu'un point selle de L est un point (x^*, y^*) de $\mathbb{R}^n \times \mathbb{R}^m$ caractérisé par :

$$L(x^*, y) \leq L(x^*, y^*) \leq L(x, y^*), \ \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^m$$

c'est-à-dire :

$$L(x^*, y^*) = \min_{x \in \mathbb{R}^n} L(x, y^*) = \max_{y \in \mathbb{R}^m} L(x^*, y)$$

Nous associons à la fonctionnelle L l'opérateur multivoque T_L défini sur $\mathbb{R}^n \times \mathbb{R}^m$ par,

$$(z, t) \in T_L(x, y) \iff \begin{cases} L(x, y') + \langle t, y' - y \rangle \leq L(x, y) \leq L(x', y) - \langle z, x' - x \rangle \\ \forall (x', y') \in \mathbb{R}^n \times \mathbb{R}^m \end{cases} \tag{3.1}$$

Autrement dit

$$T_L(x, y) = \{(z, t) \in \mathbb{R}^n \times \mathbb{R}^m : (z, -t) \in \partial L(x, y)\}$$

Et donc

$$\begin{aligned} (0, 0) \in T_L(x, y) \quad &\iff \begin{cases} L(x, y') \leq L(x, y) \leq L(x', y) \\ \forall (x', y') \in \mathbb{R}^n \times \mathbb{R}^m \end{cases} \\ &\iff (x, y) \text{ est un point selle de } L \end{aligned} \tag{3.2}$$

Ce qui signifie que les points selle globaux de L (par rapport à la minimisation en x et la maximisation en y) sont les éléments (x^*, y^*) solutions du problème $(0, 0) \in T_L(x^*, y^*)$. Autrement écrit :

$$(0, 0) \in T_L(x^*, y^*) \iff (x^*, y^*) = arg \min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} L(x, y)$$

Le résultat suivant du à Rockafellar[24] fournit des conditions suffisantes pour la maximal monotonie de l'opérateur T_L.

Proposition 3.4
Soit L une fonction selle propre et fermée[2] *sur $\mathbb{R}^n \times \mathbb{R}^m$. Alors T_L est un opérateur maximal monotone.*

Lemme 3.1
Si la fonction L est a-fortement convexe-concave[3]*, alors T_L est un opérateur a-fortement monotone.*

Preuve.
Définissons le produit scalaire et la norme sur $\mathbb{R}^n \times \mathbb{R}^m$ comme suit : Soit $(x, y), (x', y') \in \mathbb{R}^n \times \mathbb{R}^m$:

$$\begin{cases} \langle (x,y),(x',y')\rangle_{\mathbb{R}^n\times\mathbb{R}^m} = \langle x, x'\rangle_{\mathbb{R}^n} + \langle y, y'\rangle_{\mathbb{R}^m} \\ \|(x,y)\|_{\mathbb{R}^n\times\mathbb{R}^m} = \sqrt{\|x\|^2_{\mathbb{R}^n} + \|y\|^2_{\mathbb{R}^m}} \end{cases}$$

ce que nous écrivons simplement comme

$$\begin{cases} \langle (x,y),(x',y')\rangle = \langle x, x'\rangle + \langle y, y'\rangle \\ \|(x,y)\| = \sqrt{\|x\|^2 + \|y\|^2} \end{cases}$$

pour ne pas alourdir le texte.

Prenons alors (x, y) et (x', y') dans $\mathbb{R}^n \times \mathbb{R}^m$, $(z, t) \in T_L(x, y)$ et $(z', t') \in T_L(x', y')$ et montrons que

$$\langle (x,y)-(x',y'),(z,t)-(z',t')\rangle \geq a\|(x,y)-(x',y')\|^2.$$

Nous avons $(z, -t) \in \partial L(x, y)$ et $(z', -t') \in \partial L(x', y')$. La fonction $L(x, y)$ est a-fortement convexe par rapport à x, donc en utilisant la proposition 3.1, on conclut que l'opérateur $\partial_1 L$ (voir la définition (c7) à la page 18) est a-fortement monotone par rapport à x, et comme $z \in \partial_1 L(x, y)$ et $z' \in \partial_1 L(x', y')$, on a :

$$\langle z - z', x - x'\rangle \geq a\|x - x'\|^2$$

de même, $-t \in \partial_2 L(x, y)$ et $-t' \in \partial_2 L(x', y')$, donc $t \in \partial_2(-L(x, y))$ et $t' \in \partial_2(-L(x', y'))$ et comme $\partial_2(-L)$ est a-fortement monotone par rapport

[2] Voir les définitions à la page 18.
[3] Voir définition (c3) à la page 17.

à y (utiliser la proposition 3.1 avec $f(y) = -L(x, y)$), on a :

$$\langle t - t', y - y' \rangle \geq a\|y - y'\|^2$$

d'où

$$\langle z - z', x - x' \rangle + \langle t - t', y - y' \rangle \geq a(\|x - x'\|^2 + \|y - y'\|^2)$$

c'est-à-dire

$$\langle (z, t) - (z', t'), (x, y) - (x', y') \rangle \geq a\|(x, y) - (x', y')\|^2$$

ce qui prouve que T_L est a-fortement monotone sur $\mathbb{R}^n \times \mathbb{R}^m$. ■

Si L est une fonction a-fortement convexe-concave propre et fermée, alors l'opérateur T_L est maximal (proposition 3.4) a-fortement monotone (lemme 3.1). En appliquant (3.2) et le théorème 2.4 à l'opérateur T_L, nous pouvons alors énoncer :

Corollaire 3.5
Soit L une fonction a-fortement convexe-concave propre et fermée de $\mathbb{R}^n \times \mathbb{R}^m$ vers $[-\infty, +\infty]$. Alors

1. *L admet un point selle unique (x^*, y^*).*
2. *Tout algorithme parallèle asynchrone (1.1) à retards bornés associé à l'application univoque $F = (I + cT_L)^{-1}$ de $\mathbb{R}^n \times \mathbb{R}^m$ vers $\mathbb{R}^n \times \mathbb{R}^m$ où $c \geq \frac{\sqrt{\alpha}-1}{a}$ converge vers le point selle (x^*, y^*) de L.*

En se basant sur la remarque 2.5, nous pouvons énoncer le résultat suivant concernant l'algorithme parallèle de Jacobi par application du théorème 2.5.

Corollaire 3.6
Soit L une fonction selle fermée et propre de $\mathbb{R}^n \times \mathbb{R}^m$ vers $[-\infty, +\infty]$ admettant un point selle. Alors tout algorithme parallèle synchrone de Jacobi (2.1) associé à l'application univoque $F = (I + cT_L)^{-1}$ de $\mathbb{R}^n \times \mathbb{R}^m$ vers $\mathbb{R}^n \times \mathbb{R}^m$ où $c > 0$ quelconque converge vers un point selle de L.

3.3 Programme convexe

Considérons le programme convexe :

$$(P)\begin{cases} Min\ f_0(x) \\ x \in C \\ f_i(x) \leq 0,\ (1 \leq i \leq m) \end{cases} \tag{3.3}$$

où C un convexe fermé non vide de $\mathbb{R}^n$ et $f_i : C \to \mathbb{R}$ sont des fonctions (finies) convexes s.c.i. pour $0 \leq i \leq m$.

Le problème (P) est dit à contraintes qualifiées si,

(CQ) $\quad \exists x_0 \in C$ tel que $f_i(x_0) < 0,\ \forall i \in \{1, ..., m\}$.

Le Lagrangien associé au problème (P) dans sa forme étendue est la fonctionnelle L définie sur $\mathbb{R}^n \times \mathbb{R}^m$ par,

$$L(x,y) = \begin{cases} f_0(x) + \sum\limits_{i=1}^{m} y_i f_i(x) & si\ x \in C\ et\ y \in \mathbb{R}^m_+ \\ -\infty & si\ x \in C\ et\ y \notin \mathbb{R}^m_+ \\ +\infty & si\ x \notin C \end{cases}$$

Voir Rockafellar[27].

Le problème dual associé à (P) est,

$$(D)\begin{cases} Max\ g_0(y) \\ y \in \mathbb{R}^m_+ \end{cases} \tag{3.4}$$

où $g_0 : \mathbb{R}^m \to \mathbb{R} \cup \{-\infty\}$ est la fonction concave définie par,

$$g_0(y) = \inf_{x \in C} L(x,y)$$

Si (x^*, y^*) est un point selle du lagrangien L sur $\mathbb{R}^n \times \mathbb{R}^m$, alors x^* est une solution optimale du problème primal (P) et y^* est une solution optimale du problème dual (D). Inversement, si le problème *(P)* admet une solution x^* et si les contraintes sont qualifiées (condition (CQ)) alors il existe $y^* \in \mathbb{R}^m_+$ tel que (x^*, y^*) soit point selle de la fonctionelle L (voir Faure[14]

et Rockafellar[23] Cor. 28.3.1).

Supposons que le programme (P) admette une solution. Comme les fonctions f_i, $(0 \leq i \leq m)$ sont convexes et s.c.i., la fonctionnelle L est une fonction selle propre et fermée. Nous pouvons alors appliquer le corollaire 3.6 à la fonctionnelle L. Nous obtenons,

Corollaire 3.7
Supposons que le programme convexe (P) défini par (3.3) soit à contraintes qualifiées et admette une solution. Alors tout algorithme parallèle synchrone de Jacobi (2.1) associé à l'application univoque $F = (I + cT_L)^{-1}$ (où $c > 0$ quelconque) de $\mathbb{R}^n \times \mathbb{R}^m$ vers $\mathbb{R}^n \times \mathbb{R}^m$ converge vers (x^*, y^*) point selle de L, et donc x^* est une solution du primal (P) et y^* solution du dual (D).

3.4 Inégalité variationnelle

Soit C un convexe fermé non vide de $\mathbb{R}^n$, et A un opérateur multivoque maximal monotone sur $\mathbb{R}^n$ tel que $D(A) = C$.
Le problème de l'inégalité variationnelle dans sa forme générale consiste à trouver un élément x^* de C satisfaisant

$$\exists\ y^* \in Ax^*,\ \langle y^*, x - x^* \rangle \geq 0,\ \forall x \in C. \tag{3.5}$$

Pour $x \in \mathbb{R}^n$, notons par $N_c(x)$ le cône normal de C en x défini par

$$N_c(x) = \{y \in \mathbb{R}^n : \langle y, x - z \rangle \geq 0,\ \forall z \in C\}.$$

L'opérateur multivoque T défini sur $\mathbb{R}^n$ par,

$$Tx = \begin{cases} Ax + N_c(x) & si\ x \in C \\ \phi & si\ x \notin C \end{cases} \tag{3.6}$$

est un opérateur maximal monotone (Rockafellar[25]).

Lemme 3.2
Les solutions de l'opérateur T sont exactement les solutions du problème de l'inégalité variationnelle (3.5).

Preuve.

$$\begin{aligned}
0 \in Tx^* &\iff 0 \in Ax^* + N_c(x^*) \\
&\iff \exists y^* \in Ax^* : 0 \in y^* + N_c(x^*) \\
&\iff \exists y^* \in Ax^* : -y^* \in N_c(x^*) \\
&\iff \exists y^* \in Ax^* : \langle -y^*, x^* - z\rangle \geq 0 \ \forall z \in C \\
&\iff \exists y^* \in Ax^* : \langle y^*, z - x^*\rangle \geq 0 \ \forall z \in C \\
&\iff x^* \textit{ est solution de } (3.5)
\end{aligned}$$

■

Lemme 3.3
Si A est a-fortement monotone alors T est un opérateur a-fortement monotone.

Preuve.
Soient $x, x' \in D(T) = C$, $y \in Tx$ et $y' \in Tx'$
donc

$$\begin{cases} y = y_1 + y_2, \ y_1 \in Ax, \ y_2 \in N_c(x) \\ y' = y'_1 + y'_2, \ y'_1 \in Ax', \ y'_2 \in N_c(x') \end{cases}$$

or

$$\begin{cases} y_2 \in N_c(x) \Rightarrow \langle y_2, x - z\rangle \geq 0, \ \forall z \in C \\ y'_2 \in N_c(x') \Rightarrow \langle y'_2, x' - z\rangle \geq 0, \ \forall z \in C \end{cases}$$

d'où

$$\begin{aligned}
\langle y - y', x - x'\rangle &= \langle y_1 - y'_1, x - x'\rangle + \langle y_2 - y'_2, x - x'\rangle \\
&= \underbrace{\langle y_1 - y'_1, x - x'\rangle}_{\geq a\|x-x'\|^2} + \underbrace{\langle y_2, x - x'\rangle}_{\geq 0} + \underbrace{\langle y'_2, x' - x\rangle}_{\geq 0} \\
&\geq a\|x - x'\|^2
\end{aligned}$$

■

En utilisant les lemmes 3.2, 3.3 et le théorème 2.4 nous pouvons alors énoncer :

Corollaire 3.8
Soit C un convexe fermé non vide de $\mathbb{R}^n$ et A un opérateur multivoque maximal a-fortement monotone défini sur C. Alors

1. *Le problème de l'inégalité variationnelle (3.5) admet une solution unique* x^*.
2. *Tout algorithme parallèle asynchrone (1.1) à retards bornés associé à l'application* $F = (I + cT)^{-1}$ *où* T *est défini par (3.6) et* $c \geq \frac{\sqrt{\alpha}-1}{a}$ *converge vers la solution* x^* *du problème (3.5).*

Nous pouvons aussi énoncer son équivalent synchrone pour A maximal monotone (non nécessairement fortement monotone) et $c > 0$ quelconque obtenu en utilisant le théorème 2.5.

Corollaire 3.9
Soit C *un convexe fermé non vide de* $\mathbb{R}^n$ *et* A *un opérateur multivoque maximal monotone défini sur* C *tel que le problème (3.5) admette une solution. Alors tout algorithme parallèle synchrone de Jacobi (2.1) associé à l'application* $F = (I + cT)^{-1}$ *où* T *est défini par (3.6) et* $c > 0$ *quelconque converge vers* x^* *solution du problème (3.5).*

Conclusion

Dans ce travail, nous avons considéré des algorithmes parallèles asynchrones associés à des opérateurs F non linéaires non-expansifs définis sur $\mathbb{R}^n$. Nous avons donné le résultat de convergence concernant ces algorithmes vers un point fixe de F relativement à la norme uniforme sur $\mathbb{R}^n$, résultat qui étend celui de Bahi[3] au cas non linéaire. Nous avons montré comment ces algorithmes sont bien adaptés au calcul des solutions d'un opérateur maximal fortement monotone défini sur $\mathbb{R}^n$ et d'un opérateur maximal monotone dans le cas de l'algorithme parallèle de Jacobi. Ensuite, nous avons appliqué ces résultats pour les calculs du minimum d'une fonctionnelle, du point selle d'une fonction selle, de la solution des programmes convexes et enfin de la solution du problème de l'inégalité variationnelle.

Références

[1] A. Addou, A. Benahmed, *Parallel synchronous algorithm for nonlinear fixed point problems*, Proyecciones, vol. 23, no. 3, pp. 241-252 (2004).

[2] A. Addou, A. Benahmed, *Parallel asynchronous iterations for nonlinear fixed point problems*, (to appear).

[3] J. Bahi, *Asynchronous iterative algorithms for nonexpansive linear systems*, Parallel And Distributed Computing, vol. 60, no. 1, pp. 92-112 (2000).

[4] J. Bahi, S. Contrassot-Vivier, R. Couturier, *Coupling dynamic load balancing with asynchronism in iterative algorithms on the computational grid*, In 17th IEEE and ACM int. conf. on International Parallel and Distributed Processing Symposium, IPDPS03, pages 40a, 9 pages (2003).

[5] G. M. Baudet, *Asynchronous iterative methods for multiprocessors*, J. ACM, 25, pp. 226-244 (1978).

[6] H. Brézis, *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*, North Holland (1973).

[7] D. P. Bertsekas, *Distributed asynchronous computation of fixed points*, Math. Prog., vol. 27, pp. 107-120 (1983).

[8] D. P. Bertsekas, J. Tsitsiklis, *Parallel and Distributed Computation : Numerical Methods*, Prentice-Hall (1989).

[9] D. P. Bertsekas, J. Tsitsiklis, *Some aspects of parallel and distributed iterative algorihms-A survey*, Automatica, vol. 27, no.1, pp. 3-21 (1991).

[10] D. Chazan, W. L. Miranker, *Chaotic relaxation*, Lin. Alg. Appl. 2, pp. 199-222 (1969).

[11] K. Deimling, *Nonlinear Functional Analysis*, Springer-Verlag Berlin (1985).

[12] M.N. El Tarazi, *Contraction et ordre partiel pour l'étude d'algorithmes synchrones et asynchrones en analyse numérique*, Thesis. Univ. Franche-comt. (1981).

[13] M.N. El Tarazi, *Somme convergence results for asynchronous algorithms*, Numer. Math. 39, pp. 325-340 (1982).

[14] P. Faure, *Analyse numérique. Notes d'optimisation*, (1988).

[15] D. H. Griffel, *Applied Functional Analysis*, Wiley (1981).

[16] J. C. Miellou, *Itérations chaotique à retard*, C.R.A.S. Paris, 278 pp. 957-960 (1974).

[17] J. C. Miellou, *Algorithmes de relaxation chaotiques à retard*, RAIRO (R1), pp. 55-82 (1975).

[18] J. C. Miellou, *Itérations chaotique à retard, étude de la convergence dans le cas d'espaces partiellement ordonnés*, C.R.A.S. Paris, série A, 280 pp. 233-236 (1975).

[19] J. C. Miellou, P. Cortey-Dumont, M. Boulbrachêne, *Perturbation of fixed-point iterative methods*, Advances in Parallel Computing, Vol. 1, pp. 81-122 (1990).

[20] R. R. Phelps, *Lectures on maximal monotone operators*, arXiv :math.FA/9302209 v1 (4 Feb 1993).

[21] F. Robert, M. Charnay, F. Musy, *Itérations chaotiques série-parallèle pour des équations non-linéaires de point fixe*, Apl. Mat., 20, pp. 1-38 (1975).

[22] F. Robert, *Convergence locale d'itération chaotique non-linéaire*, C.R.A.S. série A, 284 pp. 679-682 (1977).

[23] R.T. Rockafellar, *Convex Analysis*, Princeton University Press. (1970).

[24] R. T. Rockafellar, *Monotone operators associated with saddle-functions and minimax problems*, Proc. Symp. Pure Math. 18, 241-250 (1970).

[25] R.T. Rockafellar, *On the maximality of sums of nonlinear monotone operators*, Trans. Amer. Math. Soc., 149, pp. 75-88 (1970).

[26] R.T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control And Optimization, Vol. 14, No. 5, pp. 877-898 (1976).

[27] R.T. Rockafellar, *Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming*, Math. Oper. Res., 1, pp. 97-116 (1976).