



HAL
open science

Méthodes de Points Intérieurs et de quasi-Newton

Philippe Segalat

► **To cite this version:**

Philippe Segalat. Méthodes de Points Intérieurs et de quasi-Newton. Mathématiques [math]. Université de Limoges, 2002. Français. NNT: . tel-00005478

HAL Id: tel-00005478

<https://theses.hal.science/tel-00005478>

Submitted on 29 Mar 2004

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.

UNIVERSITE DE LIMOGES
U.F.R. DE SCIENCES ET TECHNIQUES

THÈSE

pour obtenir le grade de
DOCTEUR DE L'UNIVERSITE DE LIMOGES

Discipline : **Mathématiques et ses applications**

présentée et soutenue publiquement

par

Philippe SEGALAT

le 20 Décembre 2002

Méthodes de points intérieurs et de quasi-Newton

JURY

Président	M. Michel THERA, Professeur à l'université de Limoges
Rapporteurs	M. Jean-Pierre DUSSAULT, Professeur à l'université de Sherbrooke (Canada) M. Pierre MARECHAL, Maître de conférences à l'université de Montpellier
Examineur	M. Mounir HADDOU, Maître de conférences à l'université d'Orléans
Directeurs de thèse	M. Paul ARMAND, Maître de conférences à l'université de Limoges M. Christian MALIVERT, Professeur à l'université de Limoges.

Table des matières

Remerciements	5
Introduction	7
1 Mise en œuvre d'une méthode de points intérieurs et de quasi-Newton.	15
1.1 Introduction	15
1.2 Présentation de la méthode	15
1.2.1 Algorithme \mathcal{A}_μ	18
1.2.2 Algorithme \mathcal{A}	22
1.2.3 Convergence des algorithmes	23
1.3 Mise en œuvre	25
1.3.1 Données initiales	25
1.3.2 Tests d'arrêt	28
1.3.3 Calcul de la direction	31
1.3.4 Recherche linéaire	32
1.3.5 Formules de quasi-Newton	37
1.3.6 Décroissance de μ	41
1.4 Résultats numériques	42
1.4.1 Codes testés : LANCELOT et NOPTIQ	42
1.4.2 Résultats	43
1.5 Conclusion	52
2 Utilisation des formules de quasi-Newton à mémoire limitée.	55
2.1 Introduction	55
2.2 Représentation compacte des formules de BFGS à mémoire limitée	57
2.2.1 Formule récursive	57
2.2.2 Représentation compacte	58
2.3 Une méthode de points intérieurs à mémoire limitée	60
2.3.1 Utilisation des formules de quasi-Newton à mémoire limitée	60
2.3.2 Problèmes avec des contraintes de bornes	65
2.3.3 Problèmes avec des contraintes générales	68
2.4 Algorithme et convergence	73
2.4.1 Algorithme	73

2.4.2	Analyse de convergence	74
2.5	Résultats numériques	78
2.5.1	Le code 1-BFGS-B	78
2.5.2	Comparaisons des différents codes	79
2.6	Conclusion	87
3	Analyse de convergence d'un algorithme non réalisable de points intérieurs.	89
3.1	Introduction	89
3.2	Rappels	91
3.3	Hypothèse	93
3.4	Le problème pénalisé	94
3.5	Le problème barrière	97
3.6	Le pas de Newton	101
3.7	Le problème barrière primal-dual	105
3.8	Algorithme $\mathcal{A}_\mu^{\sigma,\omega}$	116
3.9	Convergence r-linéaire de l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$	117
3.10	Convergence q-superlinéaire de l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$	125
3.11	Algorithme global \mathcal{A}	131
3.12	Convergence de l'algorithme \mathcal{A}	132
3.13	Convergence des itérés vers le centre analytique primal-dual . . .	137
3.14	Conclusion	143
	Conclusion	145
	Annexes	147
	Bibliographie	167

Remerciements

Je remercie mon directeur de thèse Paul Armand de m'avoir proposé ce sujet, de son aide aux moments opportuns et de m'avoir fait partager son expérience.

Je remercie Christian Malivert d'avoir été mon directeur de thèse et de l'attention qu'il a portée à mon travail.

Je remercie mes deux rapporteurs Jean-Pierre Dussault et Pierre Maréchal de la lecture attentive de mon mémoire et de leurs conseils pour l'améliorer.

Je remercie Mounir Haddou de sa participation à mon jury et de ses remarques constructives.

Je remercie Michel Théra d'avoir été mon président de jury et tout l'équipe d'Analyse non linéaire et optimisation de son soutien pendant ces années.

Je remercie les membres du LACO, Martine Guerletin, Nadine Tchefranoff, Patricia Vareille et tout particulièrement Yolande Pinol de leur aide, leur disponibilité et leur gentillesse.

Je remercie l'ensemble des doctorants (par ordre d'apparition) : Cyril Brunie, Cyril Vervoux, Delphine Boucher, Aude Maignan, Christina Stamate, Meriem Heraoua, Ayoub Otmani, Jean-Yves Enjalbert, Thomas Cluzeau, Matthieu Le Floc'h, Michaël Lescop, Carmen Nedeloaia et Laurent Dubreuil.

Je remercie tous mes amis de m'avoir distrait de mon travail pendant ces années : Bertrand Cholet, Raphaël Matuszewicz, Benjamin Roche, Christophe et Liliane Abel, Sylvain Fayette, Laurent Thouron, Stéphanie Picaud, Florent Pinard, Eric Périgault, Isabelle Thouron, Corinne Guillon, Marc Thévenot, Aurélie Varachaud, Marjorie Palomarès, Céline et Thierry Périnet, Marianne et Jérôme Perrier, ainsi que tous les autres...

Je remercie tout particulièrement mes parents, mon frère, mes grands-mères et ma tante de leur soutien.

Je remercie finalement Carine Picaud de son aide, de son soutien et de son amour.

Introduction

Les méthodes de points intérieurs sont apparues dans les années cinquante. Ces méthodes reposent sur la fonction barrière logarithmique définie, en 1955, par Frisch ([23]). Cette fonction barrière logarithmique sera largement étudiée dans les années soixante, notamment dans le livre de Fiacco et McCormick ([19]). C'est dans ce livre que le terme de points intérieurs sera introduit. Les itérés générés par les méthodes basées sur cette fonction sont strictement réalisables : ils restent à l'intérieur du domaine réalisable, d'où la terminologie de ces méthodes. L'engouement pour ces méthodes dans les années soixante est dû au développement de méthodes performantes en optimisation sans contrainte, citons notamment le code **SUMT** développé par Fiacco et McCormick ([19]). Le problème du mauvais conditionnement des matrices hessiennes des fonctions barrières d'abord étudié par Murray ([46, 45]) puis par Lootsma ([39, 40, 41]) ralentit le développement de ces méthodes au début des années soixante-dix. On redécouvrira ces méthodes plus tard grâce à l'optimisation linéaire.

Avant 1984, tout problème d'optimisation linéaire se résolvait par la méthode du simplexe développé par Dantzig ([17]) ou par une variante de celle-ci. Des recherches ont été menées pour mettre au point une autre méthode mais aucune de celles proposées n'améliorait celle du simplexe. Aussi, pendant une quarantaine d'années, cette méthode domina l'optimisation linéaire. Puis, dans les années 70, la théorie de la complexité devint une part intégrante de l'optimisation linéaire, si bien qu'on demanda aux méthodes développées de converger en un temps polynômial, c'est à dire de résoudre le problème en un nombre d'opérations qui doit être borné par un polynôme en la taille du problème. Mais la méthode du simplexe n'a pas cette propriété, comme l'ont montré Klee et Minty ([36]). On se demanda alors si un algorithme d'optimisation linéaire avait cette propriété. En 1979, Khachian proposa un algorithme de programmation linéaire appelé méthode des ellipses de Khachian ([35]). Bien que convergeant polynômialement en théorie, cet algorithme convergeait en pratique moins vite que le simplexe. Toutefois, Khachian montra théoriquement l'existence d'algorithmes à convergence polynômiale. Il restait maintenant à en trouver qui soient efficaces en pratique. Ce que fit Karmarkar en 1984 ([34]). Il proposa en effet un algorithme de points intérieurs à convergence polynômiale pour résoudre des problèmes d'optimisation linéaire. Cela provoqua un regain d'intérêt pour les méthodes de points intérieurs, aussi bien en optimisation linéaire qu'en optimisation non linéaire.

Décrivons maintenant ces méthodes de points intérieurs. Considérons le

problème suivant

$$(\mathcal{P}) \quad \begin{cases} \min f(x) \\ c(x) \leq 0 \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est la fonction à minimiser, $c(x) \leq 0$ signifie que les fonctions $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ($1 \leq i \leq m$), composantes de c , doivent être non positives à la solution. On suppose que les fonctions f et c_i ($1 \leq i \leq m$) sont au moins deux fois différentiables et convexes. Le problème barrière associé au problème (\mathcal{P}) est le suivant

$$(\mathcal{P}_\mu) \quad \min \varphi_\mu(x) := f(x) - \mu \log(-c(x)),$$

où φ_μ est la fonction barrière définie par Frisch ([23]) et où μ , appelé paramètre barrière, est strictement positif. Comme le logarithme n'est défini que pour des arguments strictement positifs, la fonction barrière logarithmique n'est définie que sur l'ensemble noté \mathcal{F}^0 des points strictement réalisables (c'est à dire vérifiant $c(x) < 0$). Par convexité de la fonction f , des fonctions c_i ($1 \leq i \leq m$), par stricte concavité et croissance de la fonction \log , la fonction φ_μ est une fonction convexe.

Le théorème suivant établit des propriétés de la fonction barrière. Notons $\text{opt}(\mathcal{P})$ l'ensemble des solutions de (\mathcal{P}) et f^* la valeur optimale de f sur cet ensemble.

Théorème *Supposons que les fonctions f et c du problème (\mathcal{P}) soient convexes et que l'ensemble des points strictement réalisables noté \mathcal{F}^0 soit non vide. Soit $\{\mu_k\}$ une suite décroissante de paramètres positifs convergeant vers 0. Supposons que l'ensemble des solutions de (\mathcal{P}) soit non vide et borné. On a alors les assertions suivantes :*

- (i) *Pour tout $\mu > 0$, φ_μ est convexe dans \mathcal{F}^0 et possède un minimum noté x_μ (non nécessairement unique) sur \mathcal{F}^0 . Tout minimum local x_μ est aussi un minimum global de φ_μ .*
- (ii) *Toute suite de minima x_{μ_k} de φ_μ a une sous suite convergente, et tout point limite possible de telles suites est dans $\text{opt}(\mathcal{P})$.*
- (iii) *$f(x_{\mu_k}) \rightarrow f^*$ et $\varphi_\mu(x_{\mu_k}) \rightarrow f^*$, pour toute suite convergente de minima $\{x_{\mu_k}\}$.*

Une preuve de ce résultat est donnée dans [60, Théorème 5].

Les méthodes de points intérieurs se déroulent comme suit. A partir d'une valeur initiale de μ , on résout de manière approchée le problème barrière (\mathcal{P}_μ) . On obtient alors un itéré dans un voisinage d'une solution du problème (\mathcal{P}_μ) . On calcule ensuite un nouveau paramètre de perturbation $\mu > 0$ inférieur au précédent. On obtient alors un nouveau problème (\mathcal{P}_μ) à résoudre. On le résout et on fait à nouveau décroître μ de telle manière que la suite de ces paramètres tende vers 0. On va donc résoudre une suite de problèmes barrières à μ fixé de manière approchée jusqu'à l'obtention d'une solution du problème initial (\mathcal{P}) . Le théorème suivant nous assure, sous certaines conditions, la convergence de ces solutions approchées vers une solution du problème (\mathcal{P}) .

Théorème *Supposons que l'ensemble \mathcal{F}^0 soit non vide et que x^* soit une solution de (\mathcal{P}) pour laquelle les conditions d'optimalité de Karush-Kuhn-Tucker soient satisfaites pour un multiplicateur λ^* . Supposons aussi que les gradients des contraintes actives sont linéairement indépendants et que les conditions de complémentarité stricte*

et du second ordre soient satisfaites en (x^*, λ^*) . Considérons la suite des problèmes barrières \mathcal{P}_{μ_k} où $\{\mu_k\}$ est une suite de valeurs positives convergent de façon monotone vers zéro quand $k \rightarrow \infty$. Alors

- (i) il existe une sous-suite de minima des fonctions barrières $\phi_{\mu_k}(x)$ convergent vers x^* ;
- (ii) pour k suffisamment grand, le hessien $\nabla^2 \phi_{\mu_k}(x_k)$ est défini positif ;
- (iii) il existe une unique fonction continûment différentiable $x(\mu)$ des minima de $\phi_{\mu}(x)$ dans un voisinage de $\mu = 0$;
- (iv) $\lim_{\mu \rightarrow 0} x(\mu) = x^*$.

Pour une preuve de ce résultat, voir [19, Théorème 12] ou [60, Théorème 8].

La figure 1 donne une idée du déroulement de la résolution du problème (\mathcal{P}) en utilisant les méthodes de points intérieurs. Les cercles représentent les voisinages des solutions des problèmes barrière dans lesquels on estime avoir résolu le problème barrière correspondant. Ces voisinages deviennent de plus en plus petit lorsqu'on s'approche de la solution du problème (\mathcal{P}) . Les itérés calculés par la méthode se rapprochent, au fur et à mesure que μ tend vers 0, de la projection dans l'espace primal du chemin central. Cette projection du chemin central est définie comme étant l'ensemble des points strictement réalisables paramétré par μ , telle que $x(\mu)$ est solution du problème (\mathcal{P}_{μ}) . Pour plus de détails sur les méthodes de points intérieurs, se référer à [61].

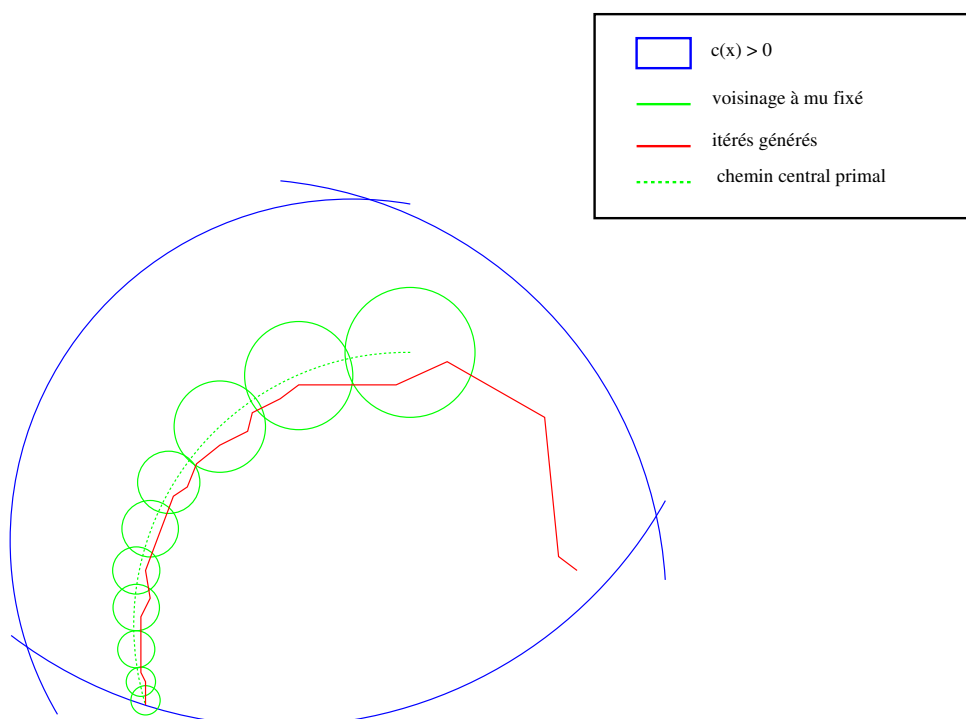


FIG. 1 – Représentation graphique d'une méthode de points intérieurs dans l'espace primal.

Dans le chapitre 1 de ce mémoire, on étudie la mise en œuvre d'une mé-

thode de points intérieurs utilisant les formules de quasi-Newton présentée dans [4]. Ces formules permettent d'approcher l'information du second ordre, lorsque celle-ci n'est pas disponible. On montrera que le code `NOPTIQ` ainsi obtenu est robuste et numériquement comparable au code `LANCELOT` qui est un code de référence. Le code `NOPTIQ` présenté dans ce chapitre est en cours de dépôt de logiciel à l'Agence pour la Protection des Programmes ([5]).

Dans le chapitre 2, on présente une version du code `NOPTIQ` employant une méthode de points intérieurs et de quasi-Newton à mémoire limitée pour résoudre les problèmes de minimisation non linéaire avec des contraintes d'inégalité non linéaire et des contraintes de bornes sur les variables. L'originalité de cette approche est d'employer dans le cadre des points intérieurs les formules de BFGS à mémoire limitée ([49]) pour résoudre des problèmes de grande taille. Cette technique évite de stocker une approximation du hessien du lagrangien. Seul un petit nombre de vecteurs est conservé en mémoire au cours des itérations. Le produit hessien-vecteur se calcule en $O(n)$ opérations. L'efficacité des formules de BFGS à mémoire limitée a été montrée dans le cas sans contrainte ([25],[38]) ou avec seulement des contraintes de bornes ([11]). On étend l'utilisation de ces méthodes au cas de contraintes d'inégalité non linéaire. On montre que l'approche par points intérieurs se prête bien à ces méthodes. On montre aussi que l'emploi des formules de BFGS à mémoire limitée diminue à la fois l'espace mémoire nécessaire au calcul de la direction de recherche et le coût en opérations de ce calcul. Notons n le nombre de variables et p le nombre de contraintes d'inégalité (autres que les contraintes de bornes) du problème à résoudre. Au lieu de stocker en mémoire $O(n^2)$ informations, on va stocker $O(nm_\ell)$ informations, où m_ℓ est un nombre généralement petit choisi par l'utilisateur suivant l'espace mémoire disponible. Le calcul de la direction primale se fera alors en $O(nm_\ell^2)$ opérations dans le cas de problèmes avec uniquement des contraintes de bornes, et en $O(n(2p^2+7m_\ell^2/2))$ opérations dans le cas de problèmes avec contraintes d'inégalité générale. Le code `NOPTIQ` suit l'algorithme présenté au chapitre 1. Des tests numériques sont présentés et un résultat de convergence r -linéaire dans le cas d'un problème convexe est montré. L'extension du code `NOPTIQ` présentée dans ce chapitre est en cours de dépôt de logiciel à l'Agence pour la Protection des Programmes ([5]).

Dans le chapitre 3, on présente un algorithme employant une méthode de points intérieurs et de quasi-Newton pour résoudre un problème de minimisation non linéaire avec contraintes d'inégalité et d'égalité linéaire. L'intérêt de notre approche réside dans la possibilité de pouvoir prendre comme point initial un point non réalisable. On étend la méthode proposée dans [2] pour prendre en compte des contraintes d'égalité linéaire. L'idée est de pénaliser d'abord le problème initial afin d'obtenir un point initial strictement réalisable. Cette pénalisation se fait à l'aide de variables de décalage et de la méthode *big M* utilisée en programmation linéaire ([17]). On montre que cette approche permet une convergence q -superlinéaire des itérés vers une solution approchée des problèmes barrière. Elle nous permet aussi d'avoir une convergence globale des itérés vers la solution du problème initial.

Notations

On note $u^\top v$ le produit scalaire des deux vecteurs u et v de \mathbb{R}^n défini par $u^\top v = \sum_{i=1}^n u_i v_i$ et $\|\cdot\|$, la norme associée. Etant donnée une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$, le gradient de f sera noté $\nabla f(x)$ et son hessien par $\nabla^2 f(x)$. Etant donnée une fonction $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$, on note $\nabla c(x)$ la matrice jacobienne de c de taille $n \times m$.

Définition Une fonction est dite de classe C^1 si elle est continûment différentiable, de classe $C^{1,1}$ et si en plus sa dérivée est Lipschitzienne.

Définition (Proposition 1.1.2, [33]) Une fonction $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ est fortement convexe de module $\kappa > 0$, si la fonction $\phi(\cdot) - \frac{\kappa}{2} \|\cdot\|^2$ est convexe.

Si ϕ est différentiable, une propriété équivalente est la forte monotonie de son gradient, c'est à dire :

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, (\nabla \phi(x) - \nabla \phi(y))^\top (x - y) \geq \kappa \|x - y\|^2.$$

Définition ([4]) Une fonction f deux fois différentiable dans un voisinage d'un point $x \in \mathbb{R}^n$ a un hessien localement radialement Lipschitzien en x , s'il existe une constante positive L telle que pour tout x' proche de x , on a

$$\|\nabla^2 f(x) - \nabla^2 f(x')\| \leq L \|x - x'\|.$$

Définition Soient $\{\alpha_k\}_{k \geq 1}$ et $\{\beta_k\}_{k \geq 1}$ deux suites de réels positifs. On dit que α_k est un grand O de β_k et on note $\alpha_k = O(\beta_k)$ s'il existe une constante C positive telle que $\alpha_k \leq C\beta_k$ pour k suffisamment grand.

Définition Soient $\{\alpha_k\}_{k \geq 1}$ et $\{\beta_k\}_{k \geq 1}$ deux suites de réels avec $\beta_k > 0$ pour tout $k \geq 1$. On dit que α_k est un petit o de β_k et on note $\alpha_k = o(\beta_k)$ s'il existe une suite $\{\epsilon_k\}_{k \geq 1}$ de réels tendant vers 0 telle que $\alpha_k \leq \epsilon_k \beta_k$ pour tout $k \geq 1$.

Définition

Soient $\{\alpha_k\}_{k \geq 1}$ une suite de vecteurs de \mathbb{R}^n et $\{\beta_k\}_{k \geq 1}$ une suite de réels avec $\beta_k > 0$ pour tout $k \geq 1$. On note $\alpha_k = o(\beta_k)$ s'il existe une suite $\{\epsilon_k\}_{k \geq 1}$ de réels tendant vers 0 telle que $\|\alpha_k\| \leq \epsilon_k \beta_k$ pour tout $k \geq 1$.

Définition Soient $\{\alpha_k\}_{k \geq 1}$ et $\{\beta_k\}_{k \geq 1}$ deux suites de réels avec $\beta_k > 0$ pour tout $k \geq 1$. On note $\alpha_k \geq o(\beta_k)$ s'il existe une suite $\{\epsilon_k\}_{k \geq 1}$ de réels tendant vers 0 telle que $\alpha_k \geq \epsilon_k \beta_k$ pour tout $k \geq 1$.

La proposition suivante établit une équivalence entre $\alpha_k \geq o(\beta_k)$ et

$$\liminf_{k \rightarrow \infty} \left(\frac{\alpha_k}{\beta_k} \right) \geq 0.$$

Proposition Soient $\{\alpha_k\}_{k \geq 1}$ et $\{\beta_k\}_{k \geq 1}$ deux suites de réels avec $\beta_k > 0$ pour tout $k \geq 1$. On a :

$$\alpha_k \geq o(\beta_k) \Leftrightarrow \liminf_{k \rightarrow \infty} \left(\frac{\alpha_k}{\beta_k} \right) \geq 0$$

Preuve. Supposons que $\alpha_k \geq o(\beta_k)$. Alors, par définition, il existe une suite $\{\epsilon_k\}_{k \geq 1}$ de réels tendant vers 0 telle que $\alpha_k \geq \epsilon_k \beta_k$ pour tout $k \geq 1$. Donc, pour tout $k \geq 1$, on a

$$\liminf_{k \rightarrow \infty} \left(\frac{\alpha_k}{\beta_k} \right) \geq \liminf_{k \rightarrow \infty} \epsilon_k = 0,$$

ce qui montre la première implication.

Supposons maintenant que la limite inférieure de la suite $\left\{ \frac{\alpha_k}{\beta_k} \right\}_{k \geq 1}$ soit positive. Notons ℓ cette limite. Par définition et par positivité de cette limite, on a

$$\frac{\alpha_k}{\beta_k} \geq \inf_{n \geq k} \left(\frac{\alpha_n}{\beta_n} \right) - \ell.$$

Il suffit, pour montrer la définition de $\alpha_k \geq o(\beta_k)$, de prendre la suite $\{\epsilon_k\}_{k \geq 1}$ où

$$\epsilon_k = \inf_{n \geq k} \left(\frac{\alpha_n}{\beta_n} \right) - \ell.$$

On montre facilement que $\lim_{k \rightarrow \infty} \epsilon_k = 0$. □

Définition Soient $\{\alpha_k\}_{k \geq 1}$ et $\{\beta_k\}_{k \geq 1}$ deux suites de réels avec $\beta_k > 0$ pour tout $k \geq 1$. On note $\alpha_k \leq o(\beta_k)$ s'il existe une suite $\{\epsilon_k\}_{k \geq 1}$ de réels tendant vers 0 telle que $\alpha_k \leq \epsilon_k \beta_k$ pour tout $k \geq 1$.

La proposition suivante établit une équivalence entre $\alpha_k \leq o(\beta_k)$ et $\limsup_{k \rightarrow \infty} \left(\frac{\alpha_k}{\beta_k} \right) \leq 0$.

Proposition Soient $\{\alpha_k\}_{k \geq 1}$ et $\{\beta_k\}_{k \geq 1}$ deux suites de réels avec $\beta_k > 0$ pour tout $k \geq 1$. On a :

$$\alpha_k \leq o(\beta_k) \Leftrightarrow \limsup_{k \rightarrow \infty} \left(\frac{\alpha_k}{\beta_k} \right) \leq 0.$$

Preuve. Supposons que $\alpha_k \leq o(\beta_k)$. Alors, par définition, il existe une suite $\{\epsilon_k\}_{k \geq 1}$ de réels tendant vers 0 telle que $\alpha_k \leq \epsilon_k \beta_k$ pour tout $k \geq 1$. Donc, pour tout $k \geq 1$, on a

$$\limsup_{k \rightarrow \infty} \left(\frac{\alpha_k}{\beta_k} \right) \leq \limsup_{k \rightarrow \infty} \epsilon_k = 0,$$

ce qui montre la première implication.

Supposons maintenant que la limite supérieure de la suite $\left\{ \frac{\alpha_k}{\beta_k} \right\}_{k \geq 1}$ soit négative. Notons ℓ cette limite. Par définition et par négativité de cette limite, on a

$$\frac{\alpha_k}{\beta_k} \leq \sup_{n \geq k} \left(\frac{\alpha_n}{\beta_n} \right) - \ell.$$

Il suffit, pour montrer la définition de $\alpha_k \leq o(\beta_k)$, de prendre la suite $\{\epsilon_k\}_{k \geq 1}$ où

$$\epsilon_k = \sup_{n \geq k} \left(\frac{\alpha_n}{\beta_n} \right) - \ell.$$

On montre facilement que $\lim_{k \rightarrow \infty} \epsilon_k = 0$.

□

Chapitre 1

Mise en œuvre d'une méthode de points intérieurs et de quasi-Newton.

1.1 Introduction

La première partie de mon étude est la mise en œuvre de la méthode d'optimisation proposée dans [4] utilisant les méthodes de points intérieurs et celles de quasi-Newton. Les méthodes de points intérieurs ont de bonnes propriétés aussi bien théoriques que numériques en optimisation linéaire ([42],[30]). De nombreux logiciels employant ces méthodes sont disponibles ([29]). Ces méthodes se développent en optimisation non linéaire ([58], [24]). La méthode de points intérieurs présentée dans [4] propose d'utiliser les méthodes de quasi-Newton pour approcher l'information du second ordre. Ces méthodes permettent en effet de résoudre des problèmes dont les dérivées secondes ne sont pas disponibles ou difficilement calculables.

On va tout d'abord rappeler la méthode proposée dans [4], puis on va détailler sa mise en œuvre en fortran pour finalement présenter les résultats numériques obtenus. Le code nommé NOPTIQ ainsi obtenu fait l'objet d'un dépôt de logiciel à l'Agence pour la Protection des Programmes ([5]) et est disponible sur simple demande auprès de Paul Armand ou de moi-même.

1.2 Présentation de la méthode

On rappelle dans ce paragraphe la méthode proposée dans [4] pour résoudre le problème suivant

$$(\mathcal{P}) \begin{cases} \min f(x) \\ c(x) \leq 0, \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est la fonction à minimiser, $c(x) \leq 0$ signifie que les fonctions $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ($1 \leq i \leq m$), composantes de c , doivent être non positives à la solution. On suppose que les fonctions f et c sont au moins deux fois différentiables et

convexes. Les contraintes d'égalité non linéaire ne sont pas prises en compte pour plusieurs raisons. On suppose tout d'abord que le problème à résoudre est convexe et la présence de ce type de contraintes rendrait le problème non convexe. On suppose ensuite qu'on dispose d'un point initial x^0 strictement réalisable pour initialiser la méthode. Obtenir un tel point avec des contraintes d'inégalité et des contraintes d'égalité non linéaire peut être difficile. Par contre, la prise en compte de contraintes d'égalité linéaire est plus aisée. Considérons le problème suivant

$$\begin{cases} \min f(x) \\ Ax = b, \\ c(x) \leq 0, \end{cases}$$

obtenu en ajoutant des contraintes d'égalité linéaire au problème (\mathcal{P}) . Supposons que A soit une matrice de taille $p \times n$ avec $p < n$ et que $b \in \mathbb{R}^p$. On suppose aussi que la matrice A est de rang plein (si ce n'est pas le cas, soit le problème est inconsistant, soit certaines contraintes sont redondantes et on élimine alors ces redondances). Notons Z la matrice de taille $n \times q$ dont les colonnes forment une base de l'espace tangent $N(A) := \{x \in \mathbb{R}^n : Ax = 0\}$. Par définition de Z , on a la propriété suivante

$$AZ = 0.$$

Soit x^0 un point initial strictement réalisable, c'est à dire tel que

$$Ax^0 = b \quad \text{et} \quad c(x^0) < 0.$$

Tout point x satisfaisant les contraintes linéaire d'égalité peut alors s'écrire

$$x = x^0 + Zu \quad \text{où} \quad u \in \mathbb{R}^q.$$

Le problème avec les contraintes d'égalité linéaire et les contraintes d'inégalité peut alors se réduire à un problème n'ayant que des contraintes d'inégalité de la même forme que (\mathcal{P}) :

$$\begin{cases} \min f(x^0 + Zu) \\ c(x^0 + Zu) \leq 0. \end{cases}$$

Notons maintenant $\ell : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ le lagrangien du problème (\mathcal{P}) défini par

$$\ell(x, \lambda) = f(x) + \lambda^\top c(x).$$

Le gradient et le hessien de ℓ par rapport à x sont donnés par les formules suivantes

$$\nabla_x \ell(x, \lambda) = \nabla f(x) + \nabla c(x) \lambda \quad \text{et} \quad \nabla_{xx}^2 \ell(x, \lambda) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 c_i(x).$$

Les conditions d'optimalité de Karush-Kuhn-Tucker (KKT) du problème (\mathcal{P}) peuvent s'écrire comme suit : si x est solution de (\mathcal{P}) , il existe un vecteur de multiplicateurs $\lambda \in \mathbb{R}^m$ tel que

$$\begin{cases} \nabla f(x) + \nabla c(x) \lambda = 0, \\ C(x) \lambda = 0, \\ c(x) \leq 0, \\ 0 \leq \lambda, \end{cases} \quad (1.1)$$

où $C(x) = \text{diag}(c_1(x), \dots, c_m(x))$ est la matrice diagonale dont les éléments diagonaux sont les composantes de $c(x)$. La première de ces conditions d'optimalité signifie que le gradient du lagrangien par rapport à x doit être nul à la solution. La seconde condition appelée condition de complémentarité signifie que si une contrainte n'est pas saturée à la solution, le multiplicateur correspondant doit être nul, c'est à dire pour tout $i \in \{1, \dots, m\}$, si $c_i(x) < 0$ alors $\lambda_i = 0$. C'est cette condition de complémentarité qui est la plus difficile à satisfaire car on ne connaît pas a priori l'ensemble des contraintes actives à la solution. Si tel était le cas, on éliminerait les contraintes non saturées à la solution pour résoudre alors un problème avec uniquement des contraintes d'égalité. Des méthodes efficaces existent pour résoudre ce type de problèmes (notamment la programmation quadratique successive). Pour contourner cette difficulté, l'idée des méthodes de points intérieurs est de perturber cette condition de complémentarité par un paramètre μ strictement positif tout en maintenant $-c(x)$ et λ strictement positifs. Cette propriété de maintenir les itérés réalisables est à l'origine de la terminologie de ces méthodes. On résout le système ainsi obtenu. Puis, par décroissance du paramètre de perturbation μ vers 0, les solutions calculées convergent vers la solution du problème initial (\mathcal{P}) (voir [60]).

La perturbation des conditions d'optimalité (1.1) par ce paramètre μ strictement positif définit le système suivant des conditions d'optimalité perturbées

$$\begin{cases} \nabla f(x) + \nabla c(x)\lambda = 0, \\ -C(x)\lambda = \mu e, \\ c(x) < 0, \\ 0 < \lambda, \end{cases} \quad (1.2)$$

où e est le vecteur de \mathbb{R}^m dont toutes les composantes sont égales à 1.

En substituant la valeur des multiplicateurs λ dans la première équation à l'aide de la seconde, on obtient le système suivant

$$\begin{cases} \nabla f(x) - \mu \nabla c(x) C^{-1}(x) e = 0, \\ c(x) < 0. \end{cases} \quad (1.3)$$

On peut alors interpréter le système (1.2) comme les conditions d'optimalité de Karush-Kuhn-Tucker du problème sans contrainte suivant

$$(\mathcal{P}_\mu) \quad \min \phi_\mu(x)$$

où $\phi_\mu : \mathbb{R}^n \rightarrow \mathbb{R}$ est la fonction définie par

$$\phi_\mu(x) = f(x) - \mu \sum_{i=1}^m \log(-c_i(x)).$$

Cette fonction ϕ_μ est appelée fonction barrière logarithmique et le problème (\mathcal{P}_μ) , le problème barrière. On emploie le terme de barrière car on remarque que si la variable x prend des valeurs telles qu'une contrainte c_i tende vers 0, le terme correspondant $\log(-c_i(x))$ dans la fonction ϕ_μ tendra vers l'infini. étant

donné qu'on souhaite minimiser la fonction ϕ_μ , les itérés x vont donc rester loin du bord, à l'intérieur de l'ensemble défini par les contraintes.

L'algorithme proposé dans [4] se déroule de la manière suivante : à partir d'une valeur initiale de μ , on résout de manière approchée les conditions d'optimalité perturbées (1.2). On obtient alors un itéré dans un voisinage d'une solution du problème (\mathcal{P}_μ). On désigne par \mathcal{A}_μ l'algorithme nécessaire à cette résolution et on appelle les itérés calculés par cet algorithme les itérés internes. On calcule ensuite un nouveau paramètre de perturbation $\mu > 0$ inférieur au précédent. On obtient alors un nouveau problème (\mathcal{P}_μ) et de nouvelles conditions d'optimalité perturbées (1.2). On les résout à l'aide de l'algorithme \mathcal{A}_μ et on fait à nouveau décroître μ de telle manière que la suite des ces paramètres tende vers 0. On va donc résoudre une suite de problèmes barrières à μ fixé de manière approchée jusqu'à l'obtention d'une solution du problème initial (\mathcal{P}). On appelle itérés externes les solutions approchées calculées par l'algorithme \mathcal{A}_μ et on désigne par \mathcal{A} l'algorithme qui, à partir des données initiales et d'un paramètre de perturbation μ initial, calcule une solution du problème donné (voir la figure 1).

1.2.1 Algorithme \mathcal{A}_μ

On va étudier dans cette partie la résolution des problèmes barrières (\mathcal{P}_μ) à μ fixé. On doit donc résoudre le système suivant

$$\begin{cases} \nabla f(x) + \nabla c(x)\lambda = 0, \\ -C(x)\lambda = \mu e, \\ c(x) < 0, \\ 0 < \lambda. \end{cases} \quad (1.4)$$

On va résoudre les égalités du système ci-dessus avec la méthode de Newton tout en maintenant satisfaites les inégalités $c(x) < 0$ et $\lambda > 0$. Notons $F : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{n+m}$ la fonction définie par

$$F(x, \lambda) = \begin{pmatrix} \nabla f(x) + \nabla c(x)\lambda \\ -C(x)\lambda - \mu e \end{pmatrix}.$$

On va appliquer la méthode de Newton pour résoudre le système suivant

$$F(x, \lambda) = 0.$$

Cette résolution va nous fournir un pas de Newton noté $d = (d^x, d^\lambda)$. La méthode de Newton étant une méthode locale qui converge quadratiquement dans un voisinage de la solution, on va la globaliser en faisant une recherche linéaire le long de la direction d calculée. On obtiendra alors un nouvel itéré à l'aide duquel on mettra à jour l'information du second ordre nécessaire au calcul du prochain pas de Newton.

Trois étapes sont donc nécessaires à la résolution du système (1.4) : le calcul d'une direction de recherche, le calcul d'un pas le long de cette direction et une mise à jour de l'information du second ordre.

Calcul du pas de Newton

On applique la méthode de Newton au système $F(x, \lambda) = 0$. On remplace donc F par son approximation linéaire et on cherche la direction notée $d = (d^x, d^\lambda) \in \mathbb{R}^n \times \mathbb{R}^m$ solution de

$$\nabla F(x, \lambda)^\top d + F(x, \lambda) = 0.$$

Cette direction est donc la solution du système suivant

$$\begin{pmatrix} M & \nabla c(x) \\ -\Lambda \nabla c(x)^\top & -C(x) \end{pmatrix} \begin{pmatrix} d^x \\ d^\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(x) + \nabla c(x) \lambda \\ -C(x) \lambda - \mu e \end{pmatrix}, \quad (1.5)$$

où Λ est une matrice diagonale définie par $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$ et où M est une approximation définie positive obtenue par les formules de quasi-Newton du hessien $\nabla_{xx}^2 \ell(x, \lambda)$ du lagrangien en x .

Il faut remarquer qu'on n'applique pas directement la méthode de Newton pour résoudre les conditions d'optimalité du problème (\mathcal{P}_μ) . La raison est la suivante. Appliquer la méthode de Newton au système (1.3) revient à calculer le pas $d^x \in \mathbb{R}^n$ solution de

$$\nabla^2 \phi_\mu(x)^\top d^x + \nabla \phi_\mu(x) = 0$$

tout en maintenant $c(x) < 0$. Le calcul du gradient et du hessien de ϕ_μ en x nous donne

$$\nabla \phi_\mu(x) = \nabla f(x) - \mu \nabla c(x) C^{-1}(x) e \quad (1.6)$$

et

$$\nabla^2 \phi_\mu(x) = \nabla^2 f(x) - \mu \sum_{i=1}^m \frac{\nabla^2 c_i(x)}{c_i(x)} + \mu \nabla c(x) C(x)^{-2} \nabla c(x)^\top.$$

On peut réécrire le hessien de ϕ_μ en fonction de celui du lagrangien ℓ :

$$\nabla^2 \phi_\mu(x) = \nabla_{xx}^2 \ell(x, -\mu C(x)^{-1} e) + \mu \nabla c(x) C(x)^{-2} \nabla c(x)^\top. \quad (1.7)$$

On remarque que, lorsque les itérés convergent vers la solution du système (1.4), le hessien $\nabla^2 \phi_\mu(x)$ de la fonction barrière devient instable numériquement. En effet, le second terme du membre droit de l'égalité peut devenir mal conditionné si une des contraintes est saturée à la solution (le produit $\mu/c_i(x)$ correspondant tend vers une valeur finie en raison de la condition de complémentarité perturbée alors que le terme $1/c_i(x)$ tend vers l'infini). C'est pour cette raison que les méthodes de points intérieurs introduites par Fiacco et McCormick ([19]) dans les années soixante ont été abandonnées peu après dans les années soixante-dix ([39],[46]).

Le calcul de la direction (d^x, d^λ) se déroule comme suit. En substituant la valeur de d^λ dans la première ligne du système (1.5) et en considérant (1.6), on obtient

$$\begin{cases} (M + \nabla c(x) \Lambda (-C(x))^{-1} \nabla c(x)^\top) d^x & = & -\nabla \phi_\mu(x) \\ d^\lambda & = & (-C(x))^{-1} (\mu e + \Lambda \nabla c(x)^\top d^x) - \lambda. \end{cases}$$

La définie positivité de M et la semi-définie positivité de la matrice $\nabla c(x)\Lambda(-C(x))^{-1}\nabla c(x)^\top$ entraînent l'existence d'une solution unique (d^x, d^λ) pour ce système. On remarque en utilisant la première équation du système précédent que la direction primale d^x est une direction de descente de la fonction barrière ϕ_μ :

$$\nabla\phi_\mu(x)^\top d^x = -(d^x)^\top(M + \nabla c(x)\Lambda(-C(x))^{-1}\nabla c(x)^\top)d^x < 0.$$

Recherche linéaire

La méthode de Newton étant une méthode locale, on la globalise avec une recherche linéaire de type Armijo en utilisant la fonction de mérite $\psi_\mu : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ définie par

$$\psi_\mu(x, \lambda) = \phi_\mu(x) + \mathcal{V}_\mu(x, \lambda) \quad (1.8)$$

où $\mathcal{V}_\mu(x, \lambda) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ est la fonction

$$\mathcal{V}_\mu(x, \lambda) = \lambda^\top(-c(x)) - \mu \sum_{i=1}^m \log(-c_i(x)\lambda_i)$$

Cette fonction primale-duale \mathcal{V}_μ est ajoutée à la fonction primale ϕ_μ pour la recherche linéaire afin de prendre en compte les déplacements duaux d^λ . Le lemme suivant montre que la direction $d = (d^x, d^\lambda)$ est une direction de descente de la fonction de mérite $\psi_\mu(x, \lambda)$. Notons $(\hat{x}_\mu, \hat{\lambda}_\mu)$ la solution du problème barrière (\mathcal{P}_μ) .

Lemme 1.2.1 [4, lemme 2.6] *Supposons que $(x, \lambda) \in \{(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m : (-c(x), \lambda) > 0\}$ et que M soit une matrice symétrique définie positive. Soit $d = (d^x, d^\lambda)$ la solution du système (1.5). Alors*

$$\begin{aligned} \nabla\psi_\mu(x, \lambda)^\top d &= -(d^x)^\top(M + \nabla c(x)\Lambda(-C(x))^{-1}\nabla c(x)^\top)d^x \\ &\quad - \|(-C(x))^{-1/2}\Lambda^{-1/2}(-C(x)\lambda - \mu e)\|^2. \end{aligned}$$

Ainsi, d est une direction de descente de ψ_μ au point $(x, \lambda) \neq (\hat{x}_\mu, \hat{\lambda}_\mu)$ ce qui signifie que $\nabla\psi_\mu(x, \lambda)^\top d < 0$.

La recherche linéaire consiste alors à calculer un pas $\alpha \in]0, 1]$ par rebroussement tel que le nouvel itéré obtenu soit toujours strictement réalisable et que la condition de décroissance suffisante ci-dessous soit satisfaite :

$$\psi_\mu(x + \alpha d^x, \lambda + \alpha d^\lambda) \leq \psi_\mu(x, \lambda) + \omega \alpha \nabla\psi_\mu(x, \lambda)^\top d,$$

où $\omega \in (0, \frac{1}{2})$. Un tel pas est calculable car, d'après le lemme 1.2.1, d est une direction de descente de la fonction ψ_μ .

Formules de quasi-Newton

Au lieu de calculer explicitement le hessien $\nabla_{xx}^2 \ell$ ou son inverse, on va les approcher par une matrice M qu'on va calculer à chaque itération à partir d'une matrice initiale M_0 .

D'après l'hypothèse de forte convexité du lagrangien, la matrice qu'on souhaite approcher est une matrice symétrique définie positive. On exige donc que la matrice M qu'on calcule soit symétrique et définie positive.

La première méthode de quasi-Newton a été suggérée par Davidon ([18]) en 1959, puis améliorée par Fletcher et Powell ([22]) en 1963 et est connue sous la dénomination de formule DFP. La formule de quasi-Newton la plus robuste est celle de BFGS, indépendamment suggérée par Broyden ([8]), Fletcher ([20]), Goldfarb ([28]) et Shanno ([57]) en 1970. Cette formule de mise à jour permet d'approcher le hessien du lagrangien $\nabla_{xx}^2 \ell$: étant donnée une matrice M , on obtient une nouvelle approximation notée M_+ par

$$M_+ = M - \frac{M s s^\top M}{s^\top M s} + \frac{y^\top y}{s^\top y}. \quad (1.9)$$

où s et y sont les deux vecteurs de \mathbb{R}^n définis par

$$s = x_+ - x \quad \text{et} \quad y = \nabla_x \ell(x_+, \lambda_+) - \nabla_x \ell(x, \lambda_+),$$

avec (x_+, λ_+) le nouvel itéré obtenu à la fin de la recherche linéaire à partir du point (x, λ) .

Le théorème suivant montre que les formules de BFGS maintiennent la définie positivité des matrices à condition que le produit scalaire $y^\top s$ soit strictement positif :

Théorème 1.2.2 [7, Théorème 3.5] *Si M est une matrice définie positive et si $y^\top s > 0$, alors la formule (1.9) donne une matrice M_+ définie positive.*

Cette définie positivité de la matrice M est essentielle par la suite car elle assure l'unicité de la solution du système (1.5).

Détail de l'algorithme \mathcal{A}_μ

On peut maintenant établir une itération de l'algorithme résolvant le système des conditions d'optimalité perturbées (1.2) par un paramètre de perturbation $\mu > 0$. Les constantes $\omega \in (0, \frac{1}{2})$ (test d'Armijo), $0 < \beta \leq \beta' < 1$ (coefficients de réduction pour le rebroussement) sont données indépendamment de l'indice de l'itération. Au début de l'itération, on suppose disponibles l'itéré courant $(x, \lambda) \in \mathcal{Z} := \{(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m : (-c(x), \lambda) > 0\}$, une matrice M définie positive approchant soit le hessien du lagrangien $\nabla_{xx}^2 \ell(x, \lambda)$ soit son inverse suivant le nombre de variables et de contraintes présentes dans le problème.

ALGORITHME \mathcal{A}_μ pour résoudre (1.2)

(une itération, de (x, λ, M) vers (x_+, λ_+, M_+))

1. Calcul de $d = (d^x, d^\lambda)$ solution unique du système linéaire (1.5).
Si $d = 0$, stop $((x, \lambda)$ résout le système (1.2)).
 2. Calcul d'un pas $\alpha > 0$ par rebroussement :
 - 2.0. Soit $\alpha = 1$.
 - 2.1. Tant que $(x + \alpha d^x, \lambda + \alpha d^\lambda) \notin \mathcal{Z}$, choisir un nouveau pas α dans $[\beta\alpha, \beta'\alpha]$.
 - 2.2. Tant que la condition d'Armijo

$$\psi_\mu(x + \alpha d^x, \lambda + \alpha d^\lambda) \leq \psi_\mu(x, \lambda) + \omega\alpha \nabla\psi_\mu(x, \lambda)^\top d$$
 n'est pas satisfaite, choisir un nouveau pas α dans $[\beta\alpha, \beta'\alpha]$.
 - 2.3. Soit $(x_+, \lambda_+) = (x, \lambda) + \alpha(d^x, d^\lambda)$.
 3. Mise à jour de M par les formules de quasi-Newton.
-

1.2.2 Algorithme \mathcal{A}

Une fois le problème (\mathcal{P}_μ) résolu, on fait décroître le paramètre μ vers 0 et on résout ensuite un nouveau problème barrière. L'algorithme \mathcal{A} contrôle ces itérations et diminue le paramètre de perturbation. établissons maintenant une itération de cet algorithme résolvant le problème (\mathcal{P}) . Au début de la j -ème itération externe, une approximation $z_1^j := (x_1^j, \lambda_1^j) \in \mathcal{Z}$ de la solution $\hat{z} := (\hat{x}, \hat{\lambda})$ de (1.1) et une matrice définie positive M_1^j approchant le hessien du lagrangien en x_1^j sont supposées disponibles. On suppose connues les valeurs du paramètre de perturbation $\mu^j > 0$ et celles des seuils de précision $\epsilon^j := (\epsilon_\ell^j, \epsilon_c^j) > 0$.

ALGORITHME \mathcal{A} pour résoudre le problème (\mathcal{P})

(une itération externe)

1. A partir du point initial z_1^j , utiliser l'algorithme \mathcal{A}_μ jusqu'à l'obtention d'un point $z^j := (x^j, \lambda^j)$ satisfaisant

$$\begin{cases} \|\nabla f(x^j) + \nabla c(x^j)\lambda^j\| \leq \epsilon_\ell^j, \\ \|\| - C(x^j)\lambda^j - \mu^j e\| \leq \epsilon_c^j. \end{cases} \quad (1.10)$$

2. Définir un nouveau paramètre barrière $\mu^{j+1} > 0$, de nouveaux seuils de précision $\epsilon^{j+1} := (\epsilon_\ell^{j+1}, \epsilon_c^{j+1}) > 0$ tels que $\{\mu^j\}$ et $\{\epsilon^j\}$ convergent vers zéro quand $j \rightarrow \infty$. Choisir un nouvel itéré de départ $z_1^{j+1} \in \mathcal{Z}$ et une matrice définie positive M_1^{j+1} pour l'itération externe suivante.
-

1.2.3 Convergence des algorithmes

Dans [4], les auteurs font l'hypothèse de forte convexité d'une des fonctions intervenant dans le problème pour démontrer les résultats de convergence. Une hypothèse moins forte est considérée dans [3] pour obtenir les mêmes résultats de convergence.

Hypothèse 1.2.3 [3, Hypothèse 3.1] *Les fonctions f et c_i ($1 \leq i \leq m$) sont convexes et différentiables de \mathbb{R}^n dans \mathbb{R} ; et il existe au moins un multiplicateur $\tilde{\lambda} \in \mathbb{R}^m$ tel que le lagrangien $\ell(\cdot, \tilde{\lambda})$ soit fortement convexe de module $\tilde{\kappa} > 0$.*

Hypothèse 1.2.4 [4, Hypothèse 2.1 (ii)] *L'ensemble des points strictement réalisables pour le problème (\mathcal{P}) est non vide, c'est à dire il existe un point $x \in \mathbb{R}^n$ tel que $c(x) < 0$.*

La première hypothèse permet de garantir la forte convexité du lagrangien en x pour tout multiplicateur $\lambda \in \mathbb{R}^m$. Cela a pour conséquence de maintenir strictement positif le produit $y^\top s$ dans la mise à jour de la matrice M et donc de maintenir la définie positivité de cette matrice. La seconde hypothèse, aussi appelée condition de Slater, garantit le fait que le problème barrière est bien posé.

Convergence de l'algorithme \mathcal{A}_μ

Le théorème suivant montre la convergence r -linéaire de la suite des itérés vers la solution du problème (\mathcal{P}) . On dit qu'une suite $\{x_k\}_{k \geq 1}$ converge r -linéairement vers x^* si, en notant le taux $r_k = |x_k - x^*|^{1/k}$, on a

$$\limsup_{k \rightarrow \infty} r_k < 1.$$

Notons $\hat{z}_\mu := (\hat{x}_\mu, \hat{\lambda}_\mu)$ la solution du problème (\mathcal{P}_μ) et $z^k := (x^k, \lambda^k)$ l'itéré courant généré par l'algorithme \mathcal{A}_μ .

Théorème 1.2.5 [4, Théorème 3.4] *Supposons l'hypothèse 1.2.3 satisfaite et que les fonctions f et c soient $C^{1,1}$. L'algorithme \mathcal{A}_μ génère alors une suite $\{z^k\}$ convergeant vers \hat{z}_μ r -linéairement, c'est à dire*

$$\limsup_{k \rightarrow \infty} \|z^k - \hat{z}_\mu\|^{1/k} < 1.$$

En particulier,

$$\sum_{k \geq 1} \|z^k - \hat{z}_\mu\| < \infty.$$

On montre à partir de cette convergence r -linéaire la convergence q -superlinéaire avec le théorème ci-dessous. On rappelle qu'une suite $\{x^k\}$ converge q -superlinéairement vers le point x^* si

$$x^{k+1} - x^* = o(\|x^k - x^*\|).$$

Cela revient à dire que

$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0$$

en supposant que $x^k \neq x^*$.

Théorème 1.2.6 [4, Théorème 4.4] *Supposons l'hypothèse 1.2.3 satisfaite et que les fonctions f et c soient $C^{1,1}$, deux fois continûment différentiables dans un voisinage de \hat{x}_μ avec un hessien localement radialement Lipschitzien en \hat{x}_μ . Supposons aussi que la recherche linéaire de l'algorithme \mathcal{A}_μ utilise une constante $\omega < 1/2$. Alors la suite $\{z^k\}$ générées par l'algorithme \mathcal{A}_μ converge vers \hat{z}_μ q -superlinéairement et, pour des valeurs de k suffisamment grandes, le pas unité $\alpha^k = 1$ est accepté par la recherche linéaire.*

Convergence de l'algorithme \mathcal{A}

Le théorème suivant montre la convergence globale des itérés externes générés par l'algorithme \mathcal{A} vers la solution du problème (\mathcal{P}) .

Théorème 1.2.7 [4, Théorème 5.1] *Supposons l'hypothèse 1.2.3 satisfaite et que les fonctions f et c soient $C^{1,1}$. L'algorithme \mathcal{A} génère alors une suite $\{z^k\}$ bornée et tout point limite de cette suite $\{z^k\}$ est une solution primale-duale de problème (\mathcal{P}) .*

On montre maintenant que, sous certaines conditions, les solutions approchées convergent vers un point particulier appelé centre analytique de l'ensemble des solutions, ensemble noté $\text{opt}(\mathcal{P})$, du problème initial (\mathcal{P}) . On note $\text{val}(\mathcal{P})$ la valeur optimale atteinte sur $\text{opt}(\mathcal{P})$. Par convexité des fonctions f , c_i pour $i \in \{1, \dots, m\}$, $\text{opt}(\mathcal{P})$ est un ensemble convexe. Si $\text{opt}(\mathcal{P})$ est réduit à un point, on définit le centre analytique comme étant ce point. Si tel n'est pas le cas, on définit l'ensemble

$$B = \{i : \exists x \in \text{opt}(\mathcal{P}), c_i(x) < 0\}. \quad (1.11)$$

B est un ensemble non vide d'après l'hypothèse 1.2.3. Le centre analytique de $\text{opt}(\mathcal{P})$ est alors la solution du problème

$$\begin{cases} \max \sum_{i \in B} \log(-c_i(x)), \\ c_B(x) < 0, \\ x \in \text{opt}(\mathcal{P}). \end{cases} \quad (1.12)$$

De façon similaire, notons $\text{opt}(D)$ l'ensemble des solutions duales associées au problème (\mathcal{P}) . Si $\text{opt}(D)$ est réduit à un point, le centre analytique de l'ensemble $\text{opt}(D)$ est ce point. Dans le cas contraire, on définit l'ensemble suivant

$$N = \{i : \exists \lambda \in \text{opt}(D) \text{ tel que } \lambda_i > 0\}.$$

N est un ensemble non vide (s'il l'était, l'ensemble $\text{opt}(D)$ serait réduit à $\{0\}$). On définit ensuite le centre analytique de l'ensemble $\text{opt}(D)$ comme l'unique

solution du problème

$$\begin{cases} \max \sum_{i \in N} \log(\lambda_i), \\ \lambda_N > 0, \\ \lambda \in \text{opt}(D). \end{cases} \quad (1.13)$$

On dit que le problème (\mathcal{P}) a la propriété de complémentarité stricte si $B \cap N = \{1, \dots, m\}$. Le théorème suivant montre la convergence des solutions approchées vers le centre analytique de l'ensemble des solutions du problème initial (\mathcal{P}) .

Théorème 1.2.8 [4, Théorème 5.3] *Supposons les hypothèses 1.2.3 et 1.2.4 satisfaites et que les fonctions f et c soient $C^{1,1}$. Supposons aussi que le problème (\mathcal{P}) ait la propriété de complémentarité stricte et que les tolérances des critères d'arrêt $\epsilon^j = (\epsilon_b^j, \epsilon_c^j)$ utilisées dans l'algorithme \mathcal{A} satisfassent l'estimation $\epsilon^j = o(\mu^j)$. Alors, l'algorithme \mathcal{A} génère une suite d'itérés $\{z^j\}$ convergeant vers un point $z^* = (x^*, \lambda^*)$ où x^* est le centre analytique de l'ensemble de solutions primales et λ^* est le centre analytique de l'ensemble de solutions duales.*

Dans le chapitre 3, on montre une généralisation de ces résultats de convergence pour des problèmes avec des contraintes d'inégalité et des contraintes d'égalité linéaire dont le point de départ n'est pas forcément réalisable.

1.3 Mise en œuvre

On présente maintenant une mise en œuvre de cette méthode. Cette mise en œuvre s'est faite en fortran 77. Le nom du code est NOPTIQ (Nonlinear Optimization with Interior points and Quasi-Newton techniques).

On va détailler dans cette partie les différentes étapes rencontrées dans le code : initialisation des données, choix de tests d'arrêt, calcul d'une direction de recherche linéaire, calcul d'un pas dans cette direction, mise à jour de l'information du second ordre, décroissance du paramètre de perturbation.

1.3.1 Données initiales

Considérons à nouveau le problème (\mathcal{P}) . L'utilisateur doit fournir un itéré x^0 strictement réalisable au début de l'algorithme, ce qui signifie que pour tout $i \in \{1, \dots, m\}$, on a $c_i(x^0) < 0$. Les valeurs initiales du multiplicateur noté λ^0 et du paramètre de perturbation noté μ^0 ne sont par contre pas fournies. NOPTIQ doit donc les déterminer.

Une idée consiste à les choisir de façon à l'itéré (x^0, λ^0) soit proche de la solution des conditions d'optimalité du premier problème barrière à résoudre avec un paramètre de perturbation égal à μ^0 . Connaissant déjà la valeur de x^0 , on va donc chercher les solutions notées $(\lambda^0, \mu^0) \in \{(\lambda, \mu) \in \mathbb{R}^m \times \mathbb{R} : (\lambda, \mu) > 0\}$

du système suivant

$$\begin{cases} \nabla f(x^0) + \nabla c(x^0)\lambda = 0, \\ -C(x^0)\lambda = \mu e \\ (\lambda, \mu) > 0. \end{cases} \quad (1.14)$$

Le choix de ces paramètres restent un problème ouvert. Aussi plusieurs méthodes ont été codées.

Méthode 1

Une première méthode consiste à déterminer λ^0 en premier avant de calculer une valeur de μ^0 . On résout pour cela le problème de moindre carré suivant en $\lambda \in \mathbb{R}^m$

$$\min \|\nabla f(x^0) + \nabla c(x^0)\lambda\|_2,$$

de façon à satisfaire au mieux la première égalité du système précédent. Notons $\hat{\lambda}$ la solution du problème ci-dessus. On définit alors le multiplicateur initial comme suit, pour tout $i \in \{1, \dots, m\}$,

$$\lambda_i^0 = \begin{cases} |\hat{\lambda}_i| & \text{si } \hat{\lambda}_i \neq 0, \\ \epsilon & \text{sinon} \end{cases}$$

où ϵ est une constante positive proche de 0.

D'après la condition de complémentarité perturbée, on a

$$\lambda^0 = -\mu C(x^0)^{-1}e.$$

En considérant cette égalité, on a donc la valeur suivante pour le paramètre de perturbation initial :

$$\mu^0 = \frac{\|\lambda^0\|}{\| -C(x^0)^{-1}e \|}.$$

Un autre choix possible inspiré des méthodes de points intérieurs en optimisation linéaire pour calculer le paramètre de perturbation μ^0 est de multiplier la condition de complémentarité perturbée par le vecteur e^\top . On obtient alors

$$\mu^0 = -\frac{c(x^0)^\top \lambda^0}{m}.$$

Ces deux méthodes du calcul de μ^0 ont été codées. Les résultats obtenus étaient similaires.

L'inconvénient de cette méthode est le suivant : la solution $\hat{\lambda}$ du problème de moindre carré obtenue n'étant pas forcément strictement positive, le multiplicateur λ^0 calculé ne se trouve alors plus dans un voisinage de ce multiplicateur et la norme du gradient du lagrangien peut devenir assez grande.

Méthode 2

Une autre méthode, déjà employée dans [24], consiste à calculer d'abord le paramètre de perturbation initial et ensuite le multiplicateur. A l'aide de la condition de complémentarité perturbée, on élimine le multiplicateur λ dans la première équation des conditions d'optimalité perturbées. On obtient alors

$$\nabla f(x^0) - \mu \nabla c(x^0) C(x^0)^{-1} e = 0.$$

On détermine ensuite μ^0 comme la solution du problème quadratique sans contrainte ci-dessous

$$\min \|\nabla f(x^0) - \mu \nabla c(x^0) C(x^0)^{-1} e\|_2^2.$$

On obtient alors facilement la valeur de μ^0 :

$$\mu^0 = \frac{\nabla f(x^0)^\top \nabla c(x^0) C(x^0)^{-1} e}{\|\nabla c(x^0) C(x^0)^{-1} e\|_2^2}.$$

On en déduit ensuite une valeur pour le multiplicateur initial avec la formule suivante : pour tout $i \in \{1, \dots, m\}$,

$$\lambda_i^0 = -\frac{\mu^0}{c_i(x^0)}.$$

Cette seconde méthode donne de meilleurs résultats que la première.

Méthode 3

Une méthode qui marche très bien en pratique est de fixer le paramètre de perturbation μ^0 à 1 et d'en déduire le multiplicateur correspondant à l'aide de la condition de complémentarité perturbée : pour tout $i \in \{1, \dots, m\}$, on pose

$$\lambda_i^0 = -1/c_i(x^0) \text{ et } \mu^0 = 1.$$

Par défaut, le code NOPTIQ utilise cette dernière méthode pour démarrer l'algorithme. On précise tout de même que, pour les méthodes 1 et 2, le paramètre de perturbation initial μ^0 est recentré de manière à ne pas avoir une valeur trop petite ou trop grande. Le code NOPTIQ maintient donc ce paramètre dans l'intervalle $[\mu^{\min}, \mu^{\max}]$, où μ^{\min} et μ^{\max} sont deux valeurs réelles strictement positives avec

$$0 < \mu^{\min} \leq \mu^{\max} < 10^{20}.$$

Par défaut ces valeurs sont fixées à $\mu^{\min} = 1$ et à $\mu^{\max} = 1$ car on utilise la méthode 3 par défaut, mais l'utilisateur peut, s'il le souhaite, facilement les modifier.

1.3.2 Tests d'arrêt

On va maintenant établir les tests d'arrêts des algorithmes \mathcal{A} et \mathcal{A}_μ résolvant respectivement le problème (\mathcal{P}) et les problèmes (\mathcal{P}_μ) . On va tout d'abord montrer l'équivalence entre différents tests d'arrêts avant de déterminer ceux utilisés par NOPTIQ.

Considérons le problème avec contraintes de bornes suivant

$$(\mathcal{P}_{l,u}) \begin{cases} \min f(x) \\ l \leq x \leq u, \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction deux fois différentiable et où l et u sont les vecteurs de \mathbb{R}^n représentant respectivement les bornes inférieures et supérieures sur les variables, avec pour tout $i \in \{1, \dots, n\}$,

$$-\infty \leq l_i \leq u_i \leq +\infty.$$

Les conditions d'optimalité de Karush-Kuhn-Tucker du problème ci-dessus peuvent s'écrire : si x est solution du problème, il existe deux multiplicateurs $(v, w) \in \mathbb{R}^n \times \mathbb{R}^n$ tels que

$$\begin{cases} \nabla f(x) - v + w = 0, \\ (x_i - l_i)v_i = 0 \text{ pour tout } i \in \{1, \dots, n\} \\ (u_i - x_i)w_i = 0 \text{ pour tout } i \in \{1, \dots, n\} \\ (x - l, u - x, v, w) \geq 0. \end{cases} \quad (1.15)$$

Les solutions primales-duales du problème $(\mathcal{P}_{l,u})$ sont les solutions (x^*, v^*, w^*) de $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ vérifiant ces conditions d'optimalité (1.15). Ce système est équivalent au système ci-dessous

$$\begin{cases} l \leq x \leq u, \\ \text{pour } i \text{ variant de } 1 \text{ à } n : \\ \quad [\nabla f(x)]_i \geq 0 \text{ si } x_i = l_i, \\ \quad [\nabla f(x)]_i = 0 \text{ si } l_i < x_i < u_i, \\ \quad [\nabla f(x)]_i \leq 0 \text{ si } x_i = u_i, \end{cases} \quad (1.16)$$

où $[\nabla f(x)]_i$ est la i -ème composante du gradient de f en x . L'équivalence des systèmes (1.16) et (1.15) se démontre facilement. Les conditions (1.16) traduisent le fait que le gradient de la fonction f en la solution doit appartenir au cône normal des contraintes actives en la solution.

Considérons maintenant la projection d'un point arbitraire $y \in \mathbb{R}^n$ sur l'ensemble délimité par les contraintes de bornes défini par

$$\mathcal{B} = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, \forall i \in \{1, \dots, n\}\}.$$

Notons $P(y, l, u)$ cette projection. La i -ème composante de $P(y, l, u)$ est donnée par la formule suivante

$$[P(y, l, u)]_i = \begin{cases} l_i & \text{si } y_i < l_i \\ y_i & \text{si } l_i \leq y_i \leq u_i \\ u_i & \text{si } u_i < y_i. \end{cases} \quad (1.17)$$

La proposition ci-dessous montre que les systèmes

$$x - P(x - \nabla f(x), l, u) = 0 \quad (1.18)$$

et (1.16) sont équivalents. La démonstration de cette proposition se fait à l'aide du lemme suivant :

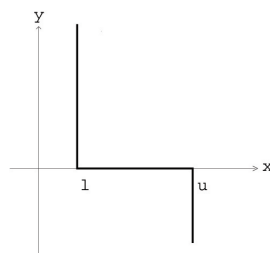
Lemme 1.3.1 Soit (x, y) un couple de nombres réels. Soient l et u deux nombres réels tels que $l \leq u$. Les conditions suivantes sont équivalentes :

$$(i) \quad \begin{cases} l \leq x \leq u, \\ y \geq 0 \quad si \quad x = l, \\ y \leq 0 \quad si \quad x = u, \\ y = 0 \quad si \quad l < x < u, \end{cases} \quad (ii) \quad x - P(x - y, l, u) = 0,$$

où P est l'opérateur défini par

$$P(x - y, l, u) = \begin{cases} l & si \quad x - y < l, \\ u & si \quad u < x - y, \\ x - y & si \quad l \leq x - y \leq u. \end{cases}$$

Preuve. Les couples (x, y) qui vérifient ces deux conditions sont de la forme



(l, y) avec $y > 0$, $(x, 0)$ avec $l \leq x \leq u$ et (u, y) avec $y < 0$. □

Proposition 1.3.2 Les systèmes (1.18) et (1.16) sont équivalents.

Preuve. On applique le lemme 1.3.1 aux couples $(x_i, [\nabla f(x)]_i)$ pour i variant de 1 à n . □

Considérons maintenant le problème avec contraintes suivant

$$\begin{cases} \min f(x) \\ c(x) \leq 0, \\ l \leq x \leq u, \end{cases} \quad (1.19)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ sont deux fonctions deux fois différentiables et où l et u sont respectivement les bornes inférieures et supérieures sur les variables (vecteurs de \mathbb{R}^n). Les conditions d'optimalité de Karush-Kuhn-Tucker

du problème peuvent s'écrire : si x est une solution de (1.19), il existe des multiplicateurs $(\lambda, v, w) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$ tels que

$$\begin{cases} \nabla f(x) + \nabla c(x)\lambda - v + w = 0, \\ C(x)\lambda = 0 \\ (x_i - l_i)v_i = 0 \text{ pour tout } i \in \{1, \dots, n\} \\ (u_i - x_i)w_i = 0 \text{ pour tout } i \in \{1, \dots, n\} \\ (-c(x), \lambda, x - l, u - x, v, w) \geq 0. \end{cases} \quad (1.20)$$

Soit $\ell : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ le lagrangien associé au problème (1.19) défini par

$$\ell(x, \lambda) = f(x) + c(x)^\top \lambda.$$

Remarquons qu'on ne dualise pas les contraintes de bornes dans cette définition du lagrangien. Les conditions d'optimalité (1.20) peuvent alors s'écrire

$$\begin{cases} \nabla_x \ell(x, \lambda) - v + w = 0, \\ C(x)\lambda = 0 \\ (x_i - l_i)v_i = 0 \text{ pour tout } i \in \{1, \dots, n\} \\ (u_i - x_i)w_i = 0 \text{ pour tout } i \in \{1, \dots, n\} \\ (-c(x), \lambda, x - l, u - x, v, w) \geq 0. \end{cases} \quad (1.21)$$

Les solutions primales-duales du problème (1.19) sont les solutions $(x^*, \lambda^*, v^*, w^*)$ de $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^n$ vérifiant les conditions d'optimalité (1.21). Le système (1.21) est équivalent, en éliminant les multiplicateurs, au système suivant

$$\begin{cases} l \leq x \leq u, \\ C(x)\lambda = 0, \\ (-c(x), \lambda) \geq 0, \\ \text{pour } i \text{ variant de } 1 \text{ à } n : \\ \quad [\nabla \ell(x, \lambda)]_i \geq 0 \text{ si } x_i = l_i, \\ \quad [\nabla \ell(x, \lambda)]_i = 0 \text{ si } l_i < x_i < u_i, \\ \quad [\nabla \ell(x, \lambda)]_i \leq 0 \text{ si } x_i = u_i. \end{cases} \quad (1.22)$$

La proposition ci-dessous étend la proposition 1.3.2 précédente au cas où des contraintes d'inégalité plus générale que des contraintes de bornes sont présentes dans le problème. Elle montre l'équivalence du système (1.22) avec le système suivant :

$$\begin{cases} x - P(x - \nabla_x \ell(x, \lambda), l, u) = 0, \\ C(x)\lambda = 0, \\ (-c(x), \lambda) \geq 0, \end{cases} \quad (1.23)$$

où P est l'opérateur de projection défini par (1.17).

Proposition 1.3.3 *Pour des problèmes avec des contraintes d'inégalité générale, les systèmes (1.22) et (1.23) sont équivalents.*

Preuve. On applique le lemme 1.3.1 aux couples $(x_i, [\nabla\ell(x, \lambda)]_i)$ pour i variant de 1 à n . \square

Le code NOPTIQ propose les deux tests d'arrêt ci-dessus pour stopper les itérations externes (et donc l'algorithme \mathcal{A}). Il considère qu'une solution du problème (1.19) est obtenue si

$$\begin{cases} \|x - P(x - \nabla_x \ell(x, \lambda), l, u)\|_\infty \leq \text{tol}, \\ \|C(x)\lambda\|_\infty \leq \text{tol}, \end{cases} \quad (1.24)$$

ou si

$$\begin{cases} \|\nabla f(x) + \nabla c(x)\lambda - v + w\|_\infty \leq \text{tol}, \\ \|C(x)\lambda\|_\infty \leq \text{tol} \\ \max(|(x_i - l_i)v_i| \text{ pour tout } i \in \{1, \dots, n\}) \leq \text{tol} \\ \max(|(u_i - x_i)w_i| \text{ pour tout } i \in \{1, \dots, n\}) \leq \text{tol}, \end{cases} \quad (1.25)$$

où tol est la précision souhaitée fournie au code par l'utilisateur.

Pour les itérations internes, le code considère qu'une solution approchée du problème barrière (\mathcal{P}_μ) est trouvée si l'itéré courant (x, λ) vérifie les conditions suivantes :

$$\begin{cases} \|\nabla f(x) + \nabla c(x)\lambda - v + w\|_\infty \leq \epsilon_\ell, \\ \|-C(x)\lambda - \mu e\|_\infty \leq \epsilon_c \\ \max(|(x_i - l_i)v_i - \mu| \text{ pour tout } i \in \{1, \dots, n\}) \leq \epsilon_c \\ \max(|(u_i - x_i)w_i - \mu| \text{ pour tout } i \in \{1, \dots, n\}) \leq \epsilon_c, \end{cases} \quad (1.26)$$

où ϵ_ℓ et ϵ_c sont des tolérances qui varient en fonction de la valeur du paramètre de perturbation μ courant :

$$\epsilon_\ell = \frac{\mu}{10\mu^0} \quad \text{et} \quad \epsilon_c = 0.999\mu.$$

Ces tolérances sont choisies de façon à ne pas demander une précision trop importante sur la résolution du premier problème barrière. Elles deviennent ensuite de plus en plus exigeantes sur la précision des solutions approchées calculées au fur et à mesure que μ décroît vers 0.

Par défaut, le code NOPTIQ utilise le test d'arrêt (1.26) pour les itérations internes (algorithme \mathcal{A}_μ) et le test d'arrêt (1.24) pour les itérations externes (algorithme \mathcal{A}).

1.3.3 Calcul de la direction

Considérons maintenant le calcul de la direction de recherche linéaire. Cette direction $d = (d^x, d^\lambda)$ est solution du système

$$\begin{pmatrix} M & \nabla c(x) \\ -\Lambda \nabla c(x)^\top & -C(x) \end{pmatrix} \begin{pmatrix} d^x \\ d^\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(x) + \nabla c(x)\lambda \\ -C(x)\lambda - \mu e \end{pmatrix}. \quad (1.27)$$

En développant l'égalité (1.27), on obtient le système

$$\begin{cases} Md^x + \nabla c(x)d^\lambda & = -\nabla f(x) - \nabla c(x)\lambda \\ -\Lambda \nabla c(x)^\top d^x - C(x)d^\lambda & = \mu e + C(x)\lambda. \end{cases} \quad (1.28)$$

Le calcul de d peut se faire suivant deux méthodes : soit on calcule d'abord la direction primale d^x puis la direction duale d^λ , soit on calcule d'abord la direction d^λ puis d^x . Calculons tout d'abord d^x . En éliminant d^λ dans la première équation à l'aide de la seconde, le système s'écrit

$$\begin{cases} (M + \nabla c(x)\Lambda(-C(x))^{-1}\nabla c(x)^\top)d^x & = -\nabla f(x) + \mu\nabla c(x)C(x)^{-1}e \\ d^\lambda & = (-C(x))^{-1}(\mu e + \Lambda\nabla c(x)^\top d^x) - \lambda \end{cases} \quad (1.29)$$

On doit alors résoudre un système de taille n^2 pour déterminer la valeur de d^x . La valeur de la direction d^λ est obtenue à l'aide de la seconde égalité. On désigne cette méthode comme étant la méthode primale.

Une autre alternative est de calculer en premier la direction duale d^λ . On élimine alors la direction d^x dans la seconde équation de (1.28) à l'aide de la première équation. On obtient

$$\begin{cases} d^x & = M^{-1}(-\nabla f(x) - \nabla c(x)(\lambda + d^\lambda)) \\ (\Lambda\nabla c(x)^\top M^{-1}\nabla c(x) - C(x))(\lambda + d^\lambda) & = \mu e - \Lambda\nabla c(x)^\top M^{-1}\nabla f(x). \end{cases} \quad (1.30)$$

On doit dans ce cas résoudre un système de taille m^2 pour déterminer la valeur de d^λ . La valeur de la direction d^x est obtenue à l'aide de la seconde égalité. On désigne cette méthode comme étant la méthode duale.

Deux modes de calcul de la direction d se présentent donc suivant le nombre n de variables et celui m de contraintes présentes dans le problème :

- si $n \leq m$ alors on calcule d avec la méthode primale (1.29) et M est mise à jour avec les formules de quasi-Newton de façon à approcher le hessien du lagrangien $\nabla_{xx}^2 \ell(x, \lambda)$ en x .
- si $m < n$ alors on calcule d avec la méthode duale (1.30) et M^{-1} est mise à jour avec les formules de quasi-Newton de façon à approcher l'inverse du hessien du lagrangien $\nabla_{xx}^2 \ell(x, \lambda)$ en x .

On remarque que cette partie de l'algorithme est la plus coûteuse en opérations et donc en temps de calcul. La résolution de ces systèmes se fait en utilisant des routines de LAPACK ([1]).

1.3.4 Recherche linéaire

NOPTIQ fait une recherche linéaire le long de la direction de descente d précédemment calculée. La fonction de mérite employée est la fonction $\psi_\mu : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ définie par

$$\psi_\mu(x, \lambda) = \phi_\mu(x) + \mathcal{V}_\mu(x, \lambda). \quad (1.31)$$

Le code calcule, par rebroussement, un pas $\alpha \in]0, 1]$ tel que la condition d'Armijo ci-dessous soit satisfaite :

$$\psi_\mu(x + \alpha d^x, \lambda + \alpha d^\lambda) \leq \psi_\mu(x, \lambda) + \omega \alpha \nabla \psi_\mu(x, \lambda)^\top d. \quad (1.32)$$

L'algorithme de recherche linéaire est donc le suivant. On suppose disponibles en entrée un itéré (x, λ) et une direction de recherche linéaire $d = (d^x, d^\lambda)$, solution du système (1.27). On rappelle que \mathcal{Z} est l'ensemble des points strictement

réalisables

$$\mathcal{Z} := \{(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m : (-c(x), \lambda) > 0\}.$$

ALGORITHME DE RECHERCHE LINÉAIRE \mathcal{A}_α
(Calcul d'un pas $\alpha > 0$ par rebroussement)

1. Soit $\alpha = 1$.
 2. Si $(x + \alpha d^x, \lambda + \alpha d^\lambda) \notin \mathcal{Z}$, choisir un nouveau pas α .
 3. Test de la condition d'Armijo (1.32).
 4. Tant que cette condition n'est pas satisfaite, choisir un nouveau pas α et aller en 3.
 5. Stop.
-

L'étape 2 de cet algorithme permet de calculer le plus grand pas α réalisable le long de la direction d dans l'espace \mathcal{Z} à partir de l'itéré (x, λ) . Afin de rester strictement réalisable, on choisit un pas α_λ maximal dans la direction duale avec la formule

$$\alpha_\lambda = \min(\alpha : \lambda_i + \alpha d_i^\lambda > 0, \forall i \in \{1, \dots, m\}),$$

et un pas maximal α_x le long de la direction primale tel que la tangente en x à la contrainte au point $x + \alpha d^x$ reste strictement négative :

$$\alpha_x = \min(\alpha : c_i(x) + \alpha \nabla_i c(x)^\top d^x < 0, \forall i \in \{1, \dots, m\}).$$

Le pas initial choisi est alors le plus grand possible dans $]0, 1]$. Il est déterminé par la formule suivante :

$$\alpha = \min(0.95 \min(\alpha_\lambda, \alpha_x), \alpha).$$

Si le pas calculé est trop petit (par exemple $\alpha \leq 10^{-6}$), NOPTIQ s'arrête. Si ce n'est pas le cas, on teste la condition d'Armijo (1.32). Si elle est vérifiée, on a trouvé un pas α satisfaisant et la recherche linéaire s'arrête. Par contre, si cette condition n'est pas vérifiée, on calcule un nouveau pas par des méthodes d'interpolation (étape 4 de l'algorithme \mathcal{A}_α). On va sélectionner les différentes valeurs du pas déjà testées et interpoler une fonction (généralement un polynôme) passant par les valeurs correspondantes de la fonction q définie par

$$q(\alpha) = \psi_\mu((x, \lambda) + \alpha(d^x, d^\lambda)).$$

A l'étape 4 de l'algorithme de recherche linéaire, on a les valeurs suivantes de cette fonction stockées en mémoire

$$\begin{cases} q(0) = \psi_\mu(x, \lambda), \\ q(\alpha_1) = \psi_\mu((x, \lambda) + \alpha_1(d^x, d^\lambda)) \end{cases} \quad \text{et} \quad q'(0) = \nabla \psi_\mu(x, \lambda)^\top d,$$

où α_1 est la pas précédemment calculé par la recherche linéaire. Ces trois données nous permettent de faire une interpolation de la fonction de q :

- interpolation quadratique : $P(\alpha) = a\alpha^2 + b\alpha + c$,
- interpolation logarithmique : $\ell(\alpha) = a + b\alpha - \mu \log(d - \alpha)$

Si le pas noté α_2 , calculé comme étant le minimum de ces fonctions, ne satisfait pas le test d'Armijo, on a alors les données suivantes

$$\begin{cases} q(0) = \psi_\mu(x, \lambda), \\ q(\alpha_1) = \psi_\mu((x, \lambda) + \alpha_1(d^x, d^\lambda)) \\ q(\alpha_2) = \psi_\mu((x, \lambda) + \alpha_2(d^x, d^\lambda)) \end{cases} \quad \text{et} \quad q'(0) = \nabla \psi_\mu(x, \lambda)^\top d.$$

Ces quatre données nous permettent de faire une interpolation de la fonction de q :

- interpolation cubique : $P(\alpha) = a\alpha^3 + b\alpha^2 + c\alpha + d$,
- interpolation logarithmique : $\ell(\alpha) = a + b\alpha - \mu \log(d - \alpha)$

On teste alors le nouveau pas obtenu. S'il ne convient pas, on fait à nouveau une des interpolations précédentes avec les dernières données calculées pour obtenir un nouveau pas.

Interpolation logarithmique

L'interpolation logarithmique, introduite dans [47], remplace l'interpolation polynômiale définie sur tout l'espace par une interpolation prenant en compte des singularités (comme c'est le cas avec la fonction barrière).

Considérons la fonction d'interpolation logarithmique $\ell : [0, d[\rightarrow \mathbb{R}$ suivante définie de

$$\ell(\alpha) = a + b\alpha - \mu \log(d - \alpha).$$

où a , b et d sont des constantes réelles à déterminer et où μ est le paramètre de perturbation définissant la fonction barrière ϕ_μ . Les dérivées première et seconde de cette fonction sont données par les formules suivantes

$$\ell'(\alpha) = b + \frac{\mu}{d - \alpha} \quad \text{et} \quad \ell''(\alpha) = \frac{\mu}{(d - \alpha)^2}.$$

Par stricte positivité de sa dérivée seconde en tout point de $[0, d[$, la fonction ℓ est strictement convexe sur cet intervalle. Elle y admet donc un minimum unique. On va calculer ce minimum, ce qui nous donnera alors une estimation du prochain pas à tester dans la recherche linéaire.

Pour déterminer ce minimum, on va tout d'abord calculer les valeurs des constantes a , b et d . Au début de l'étape 4 de l'algorithme de recherche linéaire, les données suivantes sont disponibles

$$\begin{cases} \ell(0) = q(0) = a - \mu \log(d), \\ \ell'(0) = q'(0) = b + \mu/d, \\ \ell(\alpha_1) = q(\alpha_1) = a + b\alpha_1 - \mu \log(d - \alpha_1). \end{cases}$$

A l'aide des deux premières équations, on obtient les valeurs de a et de b en fonction de celle de d :

$$\begin{cases} a = q(0) + \mu \log(d) \\ b = q'(0) - \mu/d. \end{cases}$$

Il nous reste donc à déterminer la valeur de d . d est la solution de l'équation suivante obtenue en injectant les valeurs de a et de b dans la troisième équation du système précédent :

$$\left(\frac{q(\alpha_1) - q(0) - q'(0)\alpha_1}{\mu} \right) + \log\left(1 - \frac{\alpha_1}{d}\right) + \frac{\alpha_1}{d} = 0.$$

Notons $y = \frac{\alpha_1}{d}$ et F la fonction de $[0, 1[\rightarrow \mathbb{R}$ définie par

$$F(y) = y + \log(1 - y) + k_1,$$

où k_1 est la constante réelle égale à $k_1 = (q(\alpha_1) - q(0) - q'(0)\alpha_1)/\mu$. Notons tout d'abord que $y \in [0, 1[$ (par définition du \log , on doit avoir $d - \alpha_1 > 0$). Il faut maintenant trouver la valeur de y (et donc celle de d) pour laquelle la fonction F s'annule :

$$F(y) = 0. \tag{1.33}$$

Les calculs des dérivées première et seconde de F ,

$$F'(y) = 1 - \frac{1}{1-y} \quad \text{et} \quad F''(y) = -\frac{1}{(1-y)^2},$$

montrent que cette fonction est décroissante et strictement concave sur $[0, 1[$. Elle aura donc un zéro unique si la valeur de la constante k_1 est strictement positive. Si ce n'est pas le cas, on ne fait pas d'interpolation logarithmique mais de l'interpolation polynômiale. Supposons donc $k_1 > 0$. Par une exponentiation et en réarrangeant l'équation (1.33), on obtient

$$1 - y - e^{-y}e^{-k_1} = 0.$$

On applique alors la méthode de Newton pour trouver un point y vérifiant cette équation. Notons y^* la solution trouvée. On en déduit alors la valeur du paramètre d :

$$d = \frac{\alpha_1}{y^*}.$$

Le nouveau pas noté α_2 pour la recherche linéaire est alors le minimum de la fonction d'interpolation. On le détermine par la nullité de la dérivée première de cette fonction en ce point :

$$\alpha_2 = d + \frac{\mu}{b}.$$

On teste le pas α_2 . S'il ne satisfait pas le test d'Armijo, on fait une interpolation logarithmique comme suit. On a alors les données suivantes en mémoire :

$$\begin{cases} \ell(0) = q(0) = a - \mu \log(d), \\ \ell(\alpha_1) = q(\alpha_1) = a + b\alpha_1 - \mu \log(d - \alpha_1), \\ \ell(\alpha_2) = q(\alpha_2) = a + b\alpha_2 - \mu \log(d - \alpha_2), \end{cases}$$

où α_1 et α_2 sont les deux derniers pas calculés (on a donc $\alpha_1 > \alpha_2$). Les paramètres a et b peuvent de nouveau s'écrire en fonction de d :

$$\begin{cases} a = q(0) + \mu \log(d) \\ b = \frac{1}{\alpha_1} \left(\frac{q(\alpha_1) - q(0)}{\mu} + \log\left(1 - \frac{\alpha_1}{d}\right) \right). \end{cases}$$

En arrangeant ces égalités, on doit trouver une valeur de d telle que l'équation suivante soit satisfaite

$$\log(1 - y) - \nu \log\left(1 - \frac{y}{\nu}\right) + k_2 = 0$$

où k_2 est la constante réelle définie par

$$k_2 = \frac{\alpha_1}{\mu} \left(\frac{q(\alpha_1) - q(0)}{\mu} - \frac{q(\alpha_2) - q(0)}{\mu} \right).$$

Notons $y = \frac{\alpha_1}{d}$ (on alors $y \in [0, 1[$) et $\nu = \frac{\alpha_1}{\alpha_2}$ (on a $\nu > 1$). Notons aussi F la fonction de $[0, 1[\rightarrow \mathbb{R}$ définie par

$$F(y) = \log(1 - y) - \nu \log\left(1 - \frac{y}{\nu}\right) + k_2.$$

Il faut maintenant trouver la valeur de y (et donc celle de d) pour laquelle cette fonction F s'annule

$$F(y) = 0. \tag{1.34}$$

Les valeurs des dérivées première et seconde de F sont données par les formules suivantes

$$F'(y) = -\frac{1}{1-y} + \frac{1}{1-\frac{y}{\nu}} \quad \text{et} \quad F''(y) = -\frac{1}{(1-y)^2} + \frac{1}{\nu(1-\frac{y}{\nu})^2}.$$

Un simple calcul montre alors que la fonction F est décroissante et strictement concave sur $[0, 1[$. Elle aura donc un zéro unique sur cet intervalle si la valeur de k_2 est strictement positive. Si ce n'est pas le cas, NOPTIQ fait une interpolation polynômiale. Supposons donc que $k_2 > 0$. Par une exponentiation de (1.34) et en réarrangeant les termes obtenus, on a

$$1 - y - \beta \left(1 - \frac{y}{\nu}\right)^\nu = 0.$$

On applique alors la méthode de Newton pour trouver un point y vérifiant cette équation. Notons y^* la solution trouvée. On en déduit ensuite la valeur du paramètre d :

$$d = \frac{\alpha_1}{y^*}.$$

Le nouveau pas noté α_ℓ pour la recherche linéaire est alors le minimum de la fonction d'interpolation. On le détermine par la nullité de la dérivée première de cette fonction en ce point :

$$\alpha_\ell = d + \frac{\mu}{b}.$$

On teste le nouveau pas pour la condition d'Armijo. S'il vérifie cette condition, la recherche linéaire est terminée, sinon on fait une nouvelle interpolation avec $\alpha_1 := \alpha_2$ et $\alpha_2 := \alpha_\ell$.

Interpolation polynômiale

Dans le cas où les constantes k_1 et k_2 sont négatives ou nulles, on fait une interpolation polynômiale. A partir des données connues, on calcule les coefficients du polynôme P où

$$P(x) = ax^2 + bx + c \quad \text{ou} \quad P(x) = ax^3 + bx^2 + cx + d$$

suivant le nombre de données calculées. On cherche ensuite en quel point le minimum de ce polynôme est atteint. On détermine ce point comme celui où le gradient de la fonction s'annule. Cela nous donne alors une estimation du prochain pas. Pour plus de détails sur ces méthodes, voir [7].

Des essais numériques ont été réalisés pour tester les différentes formules d'interpolation. Dans la majorité des cas, le meilleur pas (c'est à dire celui qui donnait la plus grande décroissance de la fonction de mérite) était celui donné par l'interpolation logarithmique. Aussi, NOPTIQ emploie cette méthode d'interpolation par défaut. Si elle n'est pas possible, l'interpolation polynômiale est alors utilisée.

1.3.5 Formules de quasi-Newton

NOPTIQ propose plusieurs mises à jour de la matrice M . Si le nombre de variables n est inférieur à celui du nombre de contraintes m , M est mise à jour de façon à approcher le hessien du lagrangien en x avec la formule suivante :

$$M_+ = M - \frac{Mss^T M}{s^T M s} + \frac{yy^T}{y^T s} \quad (1.35)$$

où s et y sont les vecteurs de \mathbb{R}^n définis par

$$s = (x + \alpha d^x) - x \quad \text{et} \quad y = \nabla_x \ell(x + \alpha d^x, \lambda + \alpha d^\lambda) - \nabla_x \ell(x, \lambda + \alpha d^\lambda).$$

Dans le cas contraire, M approche l'inverse du hessien du lagrangien et on considère alors la mise à jour suivante :

$$M_+ = M - \frac{Msy^T + ys^T M}{y^T s} + \left(1 + \frac{s^T M s}{y^T s}\right) \frac{yy^T}{y^T s}. \quad (1.36)$$

On rappelle le théorème suivant sur la conservation de la définie positivité des matrices M après leurs mises à jour par les formules (1.35) ou (1.36).

Théorème 1.3.4 *Si M est définie positive et si $y^T s > 0$ alors les formules (1.35) et (1.36) donnent une matrice M_+ définie positive.*

Correction de Powell et autres

Dans le cas sans contrainte, la condition de positivité du produit $y^T s$ est garantie en utilisant la recherche linéaire de Wolfe ([7]). Dans [4], l'hypothèse sur la forte convexité du lagrangien garantit la stricte positivité du produit $y^T s$.

Mais, en pratique, rien ne la garantit. Si $\nabla_{xx}^2 \ell$ est définie positif, les approximations de quasi-Newton M refléteront la courbure du problème, le produit $y^\top s$ sera alors strictement positif et les itérés convergeront rapidement, comme dans le cas sans contrainte. Si, par contre, le hessien $\nabla_{xx}^2 \ell$ a des valeurs propres négatives, approcher cette matrice par des approximations définies positives peut se révéler inadéquat et le produit $y^\top s$ peut alors prendre des valeurs négatives ou nulles. Pour contourner ce problème, Powell, [56], a proposé de calculer un vecteur y_p en modifiant le vecteur y vérifiant la condition de courbure du lagrangien $y_p^\top s > 0$. Cette modification est faite dès que le produit scalaire $y^\top s$ n'est pas suffisamment positif. Powell calcule y_p comme une combinaison convexe des vecteurs y et Ms :

$$y_p = \theta y + (1 - \theta)Ms,$$

où θ est un nombre réel dans $[0, 1]$. Le choix de Ms dans cette combinaison convexe vient du fait que, si $\theta = 0$ alors $y_p = Ms$, $M_+ = M$ et $s^\top Ms > 0$ par définie positivité de M . De manière à modifier le moins possible le vecteur y , on choisit θ le plus grand possible dans $[0, 1]$ en s'assurant de réaliser l'inégalité

$$y_p^\top s > \tau(s^\top Ms),$$

où τ est une constante positive dans $]0, 1[$. Un simple calcul nous donne alors la valeur de θ :

$$\theta = \begin{cases} 1 & \text{si } y^\top s > \tau(s^\top Ms), \\ (1 - \tau) \frac{s^\top Ms}{s^\top Ms - y^\top s} & \text{sinon.} \end{cases}$$

Une autre règle a été testée pour garantir la positivité du produit $y^\top s$ (voir [14]). On calcule un vecteur y_b en modifiant le vecteur y comme suit

$$y_b = y + \theta \nabla c(x) \wedge C(x)^{-1} \nabla c(x)^\top s,$$

où θ est un scalaire réel positif. Il est déterminé par les tests suivants :

$$\begin{aligned} \text{si } y^\top s > \nu \|\Lambda^{1/2} C^{-1/2}(x) \nabla c(x)^\top s\|^2 & \text{ alors } \theta = 0 \text{ et } y_b = y \\ \text{sinon } y_b = y + \rho v, & \end{aligned}$$

où v est un vecteur de \mathbb{R}^n déterminé par

$$\begin{aligned} \text{si } \|\Lambda^{1/2} C^{-1/2}(x) \nabla c(x)^\top s\|^2 & \geq \min(\beta_1, \|s\|) \|s\| \text{ alors} \\ v = \nabla c(x) \wedge C(x)^{-1} \nabla c(x)^\top s, & \\ \text{sinon } v = s, & \end{aligned}$$

et où ρ est un réel donné par la formule

$$\rho = \frac{\max(|y^\top s|, \nu \|\Lambda^{1/2} C^{-1/2}(x) \nabla c(x)^\top s\|^2) - y^\top s}{\|v\|^2}.$$

Une autre méthode, dite méthode du skipping, consiste à ne pas mettre à jour la matrice M par les formules de quasi-Newton si le produit $y^\top s$ n'est pas suffisamment positif. On définit alors la matrice M_+ par $M_+ = M$.

Des essais numériques ont été faits pour tester ces différentes corrections. La correction de Powell s'est montrée la plus performante. La méthode du skipping a donné des résultats nettement inférieurs. La seconde correction présentée ci-dessus donne des résultats un peu moins bons que ceux donnés par la correction de Powell. NOPTIQ utilise donc, par défaut, la correction de Powell avec la valeur de τ égale à 0.2.

Mise à l'échelle

Une mise à l'échelle de la matrice M est faite à chaque itération de l'algorithme avant sa mise à jour avec les formules de quasi-Newton (voir [51]). NOPTIQ prend pour matrice M initiale la matrice $M = \sigma I$ où I est la matrice identité de taille n^2 et σ un scalaire réel strictement positif. Suivant les formules de BFGS utilisées, le code a deux façons de calculer σ . Si n est inférieur ou égal à m , on choisit σ de façon à satisfaire l'équation de la sécante :

$$(\sigma M)s = y.$$

En multipliant cette égalité par s^\top , on obtient la valeur suivante pour le paramètre σ :

$$\sigma = \frac{y^\top s}{s^\top M s}.$$

Dans le cas contraire, M approche l'inverse du hessien du lagrangien. On choisit alors σ de façon à satisfaire l'égalité

$$(\sigma M)y = s.$$

En multipliant cette égalité par y^\top , la valeur de σ est donnée par

$$\sigma = \frac{y^\top s}{y^\top M y}.$$

NOPTIQ fait une mise à l'échelle de la matrice M à chaque itération avant la mise à jour par les formules de BFGS en multipliant la matrice M par le scalaire σ calculé par les formules précédentes. Des essais numériques ont montré que cette mise à l'échelle améliore les résultats. En effet, ne pas faire de mise à l'échelle ou n'en faire qu'une au départ de l'algorithme est moins performant. NOPTIQ, par défaut, réalise donc une mise à l'échelle de la matrice M à chaque itération.

Mise à jour des facteurs de Cholesky de M

Au lieu de mettre à jour directement la matrice M à l'aide des formules (1.35) ou (1.36), on peut mettre à jour les facteurs de Cholesky de cette matrice

([27]). On peut ainsi modifier une de ses valeurs propres si des erreurs d'arrondis perturbent les calculs. Les mises à jour de BFGS de la matrice M peuvent se voir comme deux corrections de rang 1 :

$$M_+ = M + vv^T - ww^T,$$

où v et w sont des vecteurs dépendant de la mise à jour considérée. Notons \bar{M} la matrice définie par

$$\bar{M} = M + vv^T.$$

Notons D (matrice diagonale) et L (matrice triangulaire inférieure) les facteurs de Cholesky de la matrice M . On peut alors écrire \bar{M} sous la forme

$$\bar{M} = L(D + pp^T)L^T,$$

où p est la solution du système triangulaire $Lp = v$. Par la nature même de la matrice $D + pp^T$, ses facteurs de Cholesky notés \hat{D} et \hat{L} , sont calculables directement. On a alors

$$\bar{M} = L\hat{L}\hat{D}\hat{L}^T L^T = \bar{L}\bar{D}\bar{L}^T,$$

où $\bar{D} = \hat{D}$ et $\bar{L} = L\hat{L}$ sont les facteurs de Cholesky de la matrice \bar{M} . Les composants des matrices \bar{D} et \bar{L} sont calculés récursivement avec les relations suivantes

```

t0 = 1 et v(1) = v
Pour i = 1, n
  pi = vi(i)
  ti = ti-1 + pi2/di
  d̄i = diti/ti-1
  βi = pi/(diti)
  Pour j = (i + 1), n
    vj(i+1) = vj(i) - pilji
    l̄ji = lji + βivj(i+1)
  fin (pour)
fin (pour)

```

La seconde correction de rang 1, $M_+ = \bar{M} - ww^T$ se fait de la même manière. Une attention particulière doit tout de même est prise dans ce cas car des erreurs d'arrondis peuvent perturber la définie positivité de la matrice M_+ . Cela est pris en compte dans les formules suivantes avec le paramètre ϵ (précision relative de la machine).

```

Résoudre le système  $Lp = v$ .
 $t_{n+1} = 1 - p^T D^{-1} p$ .
Si  $t_{n+1} \leq \epsilon$ 
     $t_{n+1} = \epsilon$ 
fin (si)
Pour  $i = 1, 2, \dots, n$ 
     $t_i = t_{i+1} + p_i^2 / d_i$ 
     $\bar{d}_i = d_i t_{i+1} / t_i$ 
     $\beta_i = -p_i / (d_i t_{i+1})$ 
     $v_i^{(i)} = p_i$ 
    Pour  $j = i + 1, \dots, n$ 
         $\bar{l}_{ji} = l_{ji} + \beta_i v_j^{(i+1)}$ 
         $v_j^{(i)} = v_j^{(i+1)} - p_i \bar{l}_{ji}$ 
    fin (pour)
fin (pour)

```

Des essais numériques ont été réalisés en utilisant cette méthode. Les résultats obtenus sont similaires à ceux obtenus en utilisant la formule de BFGS pour la mise à jour de la matrice M .

1.3.6 Décroissance de μ

NOPTIQ réduit le paramètre de perturbation μ en le divisant par une constante positive que l'utilisateur peut facilement modifier :

$$\mu \leftarrow \mu / \beta$$

où β est un réel strictement positif.

Il est souhaitable pour que les itérés internes restent dans un voisinage du chemin central de ne pas faire décroître trop rapidement le paramètre de perturbation. Or, en pratique, les résultats numériques ont montré que initialiser la constante β à 1000 (et donc faire décroître rapidement μ) donne des meilleurs résultats que si on affecte la valeur 10 à β .

Une autre méthode de décroissance du paramètre μ , inspirée de celle proposée dans [24], a été testée. Dans cette méthode, la décroissance de μ ne se fait plus de façon géométrique mais suivant la précision des solutions approchées de (\mathcal{P}_μ) calculées. Si l'itéré est près de la solution, μ décroît rapidement. Si l'itéré est loin de la solution, μ est multiplié par 0.9. Cette proximité de l'itéré et de la solution est mesurée par la norme infinie des conditions d'optimalité

perturbées.

$$\begin{aligned} \text{Si } & \left\| \begin{pmatrix} \nabla f(x) + \nabla c(x)\lambda \\ -C(x)\lambda - \mu e \end{pmatrix} \right\|_{\infty} < 10\mu \text{ alors} \\ & \text{Si } \mu < 10^{-4} \text{ alors } \mu \leftarrow 10\mu^2, \text{ sinon } \mu \leftarrow \mu/10, \\ \text{sinon} & \\ & \mu \leftarrow 0.9\mu. \end{aligned}$$

En pratique, cette seconde méthode n'améliore pas les résultats obtenus avec la première (c'est à dire avec $\beta = 1000$). Elle donne des résultats similaires en prenant la méthode précédente avec β égal à 10. Par défaut, NOPTIQ utilise la première méthode de décroissance de μ avec $\beta = 1000$.

1.4 Résultats numériques

Les essais numériques du code NOPTIQ ont été réalisés sous l'environnement CUTER (Constrained and Unconstrained Testing Environment revisited, [6] et [16]). Cet environnement regroupe une collection de problèmes test. Cette collection contient un grand nombre de problèmes d'optimisation non linéaire de tailles et de difficultés variables, représentant des problèmes académiques et des applications industrielles. Les essais numériques ont été réalisés sur un PC sous linux ayant un processeur cadencé à 1 GHz et possédant une mémoire de 128 Mo.

1.4.1 Codes testés : LANCELOT et NOPTIQ

LANCELOT ([15]) est un ensemble de routines pour résoudre un problème d'optimisation avec ou sans contraintes. Différents types de contraintes peuvent être pris en compte par LANCELOT : simples bornes sur les variables, contraintes d'égalité (linéaire ou non), contraintes d'inégalité.

LANCELOT est un algorithme utilisant la méthode du lagrangien augmenté. Il transforme les contraintes d'inégalité en contraintes d'égalité à l'aide de variables d'écart et résout les problèmes de la forme suivante

$$\begin{cases} \min f(x) \\ c_i(x) = 0, \text{ pour } i = 1, \dots, m, \\ l \leq x \leq u. \end{cases}$$

Les variables d'écart sont une partie du vecteur x et les vecteurs l et u représentent les bornes inférieures et supérieures sur les variables. LANCELOT prend en compte la structure de séparabilité de la fonction objectif et des contraintes pour résoudre le problème ci-dessus. Le lagrangien augmenté utilisé dans LANCELOT est la fonction suivante

$$\ell(x, \lambda, \mu) = f(x) - \sum_{i=1}^m \lambda_i c_i(x) + \frac{1}{2\mu} \sum_{i=1}^m c_i(x)^2,$$

où λ est le vecteur des multiplicateurs associé aux contraintes c et μ , le paramètre de pénalité.

LANCELOT résout une suite de problèmes de la forme suivante pour obtenir une solution du problème initial :

$$\begin{cases} \min \ell(x, \lambda, \mu) \\ l \leq x \leq u. \end{cases}$$

où les multiplicateurs λ et le paramètre de pénalité μ sont fixés. Une solution approchée de ce problème est obtenue en formant un modèle quadratique de $\ell(x, \lambda, \mu)$ et en utilisant la méthode de projection sur le gradient pour obtenir un nouvel itéré. Le hessien du modèle quadratique utilise les dérivées secondes exactes du problème ou une estimation à l'aide des formules de quasi-Newton de celles-ci. Pour plus de détails, on peut se référer à [15].

1.4.2 Résultats

On a testé les résultats obtenus par NOPTIQ sur des problèmes de petite et moyenne taille avec des contraintes de bornes et d'inégalité plus générale. Les tableaux 1.1, 1.2 et 1.3 suivants listent ces problèmes où

- nom représente le nom du problème testé,
- n représente le nombre total de variables,
- m représente le nombre total de contraintes,
- l représente le nombre total de variables bornées inférieurement,
- u représente le nombre total de variables bornées supérieurement,
- nfix représente le nombre total de variables fixées,
- ineg représente le nombre total de contraintes d'inégalité du type $c(x) \leq 0$.

Cela représente 97 problèmes de la collection de CUTEr. La sélection comporte 65 problèmes n'ayant que des contraintes de bornes et 32 problèmes avec des contraintes d'inégalité plus générale. On a retenu parmi les problèmes avec des contraintes d'inégalité générale ceux dont CUTEr fournit un point initial x^0 strictement réalisable. Dans le cas de problèmes de bornes, NOPTIQ calcule un point initial strictement réalisable si celui fourni par CUTEr ne l'est pas.

Dans les tableaux de résultats, on note n le nombre de variables et m le nombre de contraintes du problème considéré. nf est le nombre d'appels cumulés au simulateur pour calculer la valeur de la fonction objectif et celle des contraintes. La valeur de variable info indique le mode d'arrêt de NOPTIQ :

- 0 NOPTIQ stoppe normalement,
- 2 NOPTIQ stoppe car la matrice M devient singulière,
- 5 NOPTIQ stoppe car le nombre maximal d'appels au simulateur est atteint (par défaut ce nombre est égal à 10000),
- 6 NOPTIQ stoppe car le pas initial dans la recherche linéaire est trop petit,
- 7 NOPTIQ stoppe car la direction dans l'espace primal est trop petite. L'utilisateur doit fournir au code un vecteur dx_{\min} sur la précision souhaitée sur chaque composante de x . Notons x^{k+1} et x^k deux itérés succes-

nom	n	m	bornes			contraintes
			l	u	nfix	ineg
3PK	30	30	30	0	0	0
ALLINIT	4	3	2	1	1	0
BQP1VAR	1	2	1	1	0	0
BQPGABIM	50	92	46	46	4	0
BQPGASIM	50	100	50	50	0	0
CAMEL6	2	4	2	2	0	0
CHEBYQAD	100	200	100	100	0	0
DECONVB	61	72	61	11	0	0
DIPIGRI	7	4	0	0	0	4
EG1	3	4	2	2	0	0
HART6	6	12	6	6	0	0
HATFLDA	4	4	4	0	0	0
HATFLDB	4	5	4	1	0	0
HATFLDC	25	48	24	24	0	0
HIMMELPI	2	4	2	2	0	0
HS100	7	4	0	0	0	4
HS100MOD	7	4	0	0	0	4
HS110	200	400	200	200	0	0
HS113	10	8	0	0	0	8
HS117	15	20	15	0	0	5
HS12	2	1	0	0	0	1
HS1	2	1	1	0	0	0
HS24	2	5	2	0	0	3
HS25	3	6	3	3	0	0
HS29	3	1	0	0	0	1
HS2	2	1	1	0	0	0
HS35	3	4	3	0	0	1
HS35I	3	7	3	3	0	1
HS35MOD	3	3	2	0	1	1

TAB. 1.1 – Problèmes testés de CUTER.

sifs générés par le code. NOPTIQ s'arrête avec $\text{info} = 7$ si

$$|x_i^{k+1} - x_i^k| \leq dx \min_i, \text{ pour tout } i \in \{1, \dots, n\}.$$

Si les conditions d'optimalité sont pratiquement satisfaites, cette sortie n'est pas un échec pour le code.

Comparaison des formules de quasi-Newton

On a tout d'abord comparé les différentes formules des quasi-Newton utilisées dans NOPTIQ pour mettre à jour l'information du second ordre. Les tableaux 1.4 et 1.5 représentent ces résultats. On note bfgs si la mise à jour de la matrice M utilise les formules de BFGS; inverse si on utilise les formules de BFGS pour mettre à jour l'inverse du hessien du lagrangien et cholesky si NOPTIQ utilise la mise à jour des facteurs de Cholesky de la matrice M . La précision demandée est de 10^{-8} sur les conditions d'optimalité.

En annexes se trouvent les fichiers détaillés de ces résultats. On remarque que les formules utilisant la mise à jour du hessien du lagrangien sont beaucoup plus performantes que celles mettant à jour l'inverse de ce hessien. En effet, ces formules donnent les meilleurs résultats sur 66 problèmes tandis que l'autre formule n'est meilleure que sur 19 problèmes. Les formules sont équivalentes sur 7 problèmes. Les méthodes utilisant les formules de BFGS ou celles mettant à jour les facteurs de Cholesky de la matrice M sont équivalentes. Les formules de BFGS sont meilleures sur 24 problèmes, sont moins performantes sur 26 problèmes et donnent des résultats équivalents sur 16 problèmes.

nom	n	m	bornes			contraintes
			l	u	nfix	ineg
HS36	3	7	3	3	0	1
HS37	3	8	3	3	0	2
HS38	4	8	4	4	0	0
HS3	2	1	1	0	0	0
HS3MOD	2	1	1	0	0	0
HS43	4	3	0	0	0	3
HS44NEW	4	10	4	0	0	6
HS45	5	10	5	5	0	0
HS4	2	2	2	0	0	0
HS57	2	3	2	0	0	1
HS5	2	4	2	2	0	0
HS67	3	20	3	3	0	14
HS70	4	9	4	4	0	1
HS76	4	7	4	0	0	3
HS76I	4	11	4	4	0	3
HS84	5	16	5	5	0	6
HS85	5	48	5	5	0	38
HS93	6	8	6	0	0	2
KSIP	20	1001	0	0	0	1001
LOGROS	2	2	2	0	0	0
MAXLIKA	8	16	8	8	0	0
MDHOLE	2	1	1	0	0	0
MIFFLIN1	3	2	0	0	0	2
MINMAXBD	5	20	0	0	0	20
OSLBQP	8	11	8	3	0	0
PALMER1A	6	2	2	0	0	0
PALMER1B	4	2	2	0	0	0
PALMER1	4	3	3	0	0	0
PALMER1E	8	1	1	0	0	0
PALMER2A	6	2	2	0	0	0
PALMER2B	4	2	2	0	0	0
PALMER2	4	3	3	0	0	0
PALMER2E	8	1	1	0	0	0
PALMER3A	6	2	2	0	0	0
PALMER3B	4	2	2	0	0	0
PALMER3	4	3	3	0	0	0
PALMER3E	8	1	1	0	0	0
PALMER4A	6	2	2	0	0	0
PALMER4B	4	2	2	0	0	0
PALMER4	4	3	3	0	0	0
PALMER4E	8	1	1	0	0	0
PALMER5A	8	2	2	0	0	0
PALMER5B	9	2	2	0	0	0
PALMER5D	8	4	4	0	0	0
PALMER5E	8	1	1	0	0	0
PALMER6A	6	2	2	0	0	0
PALMER6E	8	1	1	0	0	0
PALMER7A	6	2	2	0	0	0
PALMER7E	8	1	1	0	0	0
PALMER8A	6	2	2	0	0	0
PALMER8E	8	1	1	0	0	0
PROBPENL	500	1000	500	500	0	0
PSPDOC	4	1	0	1	0	0

TAB. 1.2 – Problèmes testés de CUTeR.

nom	n	m	bornes			contraintes
			l	u	nfix	ineg
S368	8	16	8	8	0	0
SIM2BQP	2	2	1	1	1	0
SIMBQP	2	2	1	1	0	0
SIPOW1	2	2000	0	0	0	2000
SIPOW1M	2	2000	0	0	0	2000
SIPOW2	2	2000	0	0	0	2000
SIPOW2M	2	2000	0	0	0	2000
SIPOW3	4	2000	0	0	0	2000
SIPOW4	4	2000	0	0	0	2000
SPECAN	9	18	9	9	0	0
SPIRAL	3	2	0	0	0	2
WEEDS	3	4	3	1	0	0
WOMFLET	3	3	0	0	0	3
YFIT	3	1	1	0	0	0
ZY2	3	6	3	1	0	2

TAB. 1.3 – Problèmes testés de CUTeR.

Probleme			bfgs		inverse		cholesky	
nom	n	m	nf	info	nf	info	nf	info
3PK	30	30	328	7	519	0	246	7
ALLINIT	4	3	1	6	1	6	1	6
BQP1VAR	1	2	13	7	13	7	18	7
BQPGABIM	50	92	41	7	48	7	65	7
BQPGASIM	50	100	47	7	51	7	72	7
CAMEL6	2	4	27	7	18	7	32	7
CHEBYQAD	100	200	2054	7	456	7	1856	7
DECONVB	61	72	62	2	227	7	1898	2
DIPIGRI	7	4	24	7	27	7	30	7
EG1	3	4	54	7	75	7	46	7
HART6	6	12	38	7	27	7	75	7
HATFLDA	4	4	41	0	51	7	49	0
HATFLDB	4	5	35	0	38	7	28	0
HATFLDC	25	48	44	7	40	7	77	7
HIMMELP1	2	4	19	0	21	0	19	0
HS100MOD	7	4	28	7	29	7	33	7
HS100	7	4	24	7	27	7	29	7
HS110	200	400	1	6	1	6	1	6
HS113	10	8	21	0	24	7	25	7
HS117	15	20	31	2	39	4	30	2
HS12	2	1	19	7	19	7	27	7
HS1	2	1	69	0	83	7	78	7
HS24	2	5	18	7	47	7	18	7
HS25	3	6	135	7	9	7	143	7
HS29	3	1	36	7	30	7	40	7
HS2	2	1	29	7	40	7	26	7
HS351	3	7	21	0	21	7	27	7
HS35MOD	3	3	24	0	24	0	60	7
HS35	3	4	20	7	23	7	28	7
HS36	3	7	32	7	38	7	22	7
HS37	3	8	24	7	27	7	30	7
HS38	4	8	80	7	102	7	93	7
HS3MOD	2	1	20	0	17	0	19	0
HS3	2	1	23	7	6	4	23	7
HS43	4	3	39	7	39	7	35	7

TAB. 1.4 – Résultats de NOPTIQ obtenus en variant les formules de quasi-Newton.

Correction de Powell

On compare maintenant la prise en compte de la convexité ou du manque de convexité du problème en utilisant la correction de Powell ou pas. Ne pas l'utiliser revient à faire du *skipping* : on ne met pas à jour la matrice M si le produit $y^T s$ n'est pas positif. Les résultats sont donnés dans les tableaux 1.6 et 1.7 et une précision de 10^{-8} sur les conditions d'optimalité est demandée.

On remarque que la correction de Powell améliore les résultats sur de nombreux problèmes. Elle est plus performante sur 54 problèmes, équivalente sur 22 problèmes et moins bonne sur 24 problèmes. Cela peut se justifier par le fait que faire du *skipping* revient à ne pas modifier la matrice approchant le hessien du lagrangien pendant une ou plusieurs itérations. L'information du second ordre conservée en mémoire peut alors provenir d'itérés distants de l'itéré courant. Tandis qu'avec la méthode la de Powell, cette information est modifiée à chaque correction.

Comparaison des codes NOPTIQ et LANCELOT

Les tests suivants comparent les résultats donnés par NOPTIQ et ceux donnés par le code LANCELOT. On a utilisé le code LANCELOT avec ses données par défaut (représenté par default dans les tableaux 1.8 et 1.9). LANCELOT emploie par défaut les dérivées secondes exactes des fonctions intervenant dans le problème. On a modifié cette option pour qu'il utilise les formules de BFGS pour approcher cette information du second ordre. Cette option est notée bfgs

Probleme			bfgs		inverse		cholesky	
nom	n	m	nf	info	nf	info	nf	info
HS44NEW	4	10	23	7	92	7	23	7
HS45	5	10	28	2	333	4	23	2
HS4	2	2	14	7	16	7	15	7
HS57	2	3	71	2	568	0	160	0
HS5	2	4	21	7	19	7	51	7
HS67	3	20	31	0	38	0	31	0
HS70	4	9	75	0	82	0	50	0
HS76I	4	11	21	7	20	7	22	0
HS76	4	7	20	7	22	7	20	7
HS84	5	16	35	7	41	7	36	7
HS85	5	48	31	7	89	4	31	7
HS93	6	8	197	7	2310	7	208	7
KSIP	20	1001	53	0	61	0	53	0
LOGROS	2	2	194	7	173	7	180	7
MAXLIKA	8	16	79	7	106	7	67	7
MDHOLE	2	1	164	0	150	7	147	7
MIFFLINI	3	2	18	7	17	0	20	7
MINMAXBD	5	20	7	6	394	0	7	6
OSLBQP	8	11	23	0	23	0	29	0
PALMER1A	6	2	230	0	478	7	141	7
PALMER1B	4	2	64	7	74	7	89	7
PALMER1E	8	1	250	7	314	7	201	7
PALMER1	4	3	53	7	75	7	894	7
PALMER2A	6	2	182	0	264	0	233	0
PALMER2B	4	2	33	2	87	7	65	7
PALMER2E	8	1	271	0	607	7	275	7
PALMER2	4	3	36	2	50	7	188	0
PALMER3A	6	2	448	0	202	7	362	0
PALMER3B	4	2	72	7	81	7	126	7
PALMER3E	8	1	261	7	540	7	172	0
PALMER3	4	3	51	7	73	7	593	7
PALMER4A	6	2	52	2	273	7	252	0
PALMER4B	4	2	103	7	85	7	149	7
PALMER4E	8	1	171	0	442	7	139	7
PALMER4	4	3	64	2	215	7	37	6
PALMER5A	8	2	26	2	29	7	10000	5
PALMER5B	9	2	3368	7	477	4	1235	0
PALMER5D	8	4	43	7	42	7	38	7
PALMER5E	8	1	10000	5	1245	4	10000	5
PALMER6A	6	2	50	2	552	7	445	7
PALMER6E	8	1	223	0	485	0	187	0
PALMER7A	6	2	10000	5	1531	7	10000	5
PALMER7E	8	1	3055	7	1123	7	1121	7
PALMER8A	6	2	129	7	239	7	221	7
PALMER8E	8	1	117	7	247	0	96	0
PROBPENL	500	1000	5	7	85	7	10	7
PSPDOC	4	1	27	7	28	7	27	7
S368	8	16	66	7	30	7	74	7
SIM2BQP	2	2	13	7	13	7	20	7
SIMBQP	2	2	14	7	14	7	23	7
SIPOW1M	2	2000	22	0	3	0	22	0
SIPOW1	2	2000	23	0	3	0	23	0
SIPOW2M	2	2000	52	0	3	0	52	0
SIPOW2	2	2000	51	0	3	0	51	0
SIPOW3	4	2000	30	0	3	0	30	0
SIPOW4	4	2000	29	0	3	0	29	0
SPECAN	9	18	50	7	62	7	69	7
SPIRAL	3	2	112	6	388	0	236	7
WEEDS	3	4	152	0	86	7	77	7
WOMFLET	3	3	213	2	2410	7	116	2
YFIT	3	1	160	0	207	0	158	0
ZY2	3	6	19	7	126	4	19	7

TAB. 1.5 – Résultats de NOPTIQ obtenus en variant les formules de quasi-Newton.

Probleme			powell		skip	
nom	n	m	nf	info	nf	info
3PK	30	30	246	7	10000	5
ALLINIT	4	3	1	6	1	6
BQP1VAR	1	2	18	7	18	7
BQPGABIM	50	92	65	7	66	7
BQPGASIM	50	100	72	7	72	7
CAMEL6	2	4	32	7	35	7
CHEBYQAD	100	200	1856	7	3568	7
DECONVB	61	72	1898	2	893	0
DIPIGRI	7	4	30	7	30	7
EG1	3	4	46	7	26	7
HART6	6	12	75	7	75	7
HATFLDA	4	4	49	0	38	0
HATFLDB	4	5	28	0	36	0
HATFLDC	25	48	77	7	77	7
HIMMELPI	2	4	19	0	175	7
HS100MOD	7	4	33	7	33	7
HS100	7	4	29	7	29	7
HS110	200	400	1	6	1	6
HS113	10	8	25	7	25	7
HS117	15	20	30	2	55	7
HS12	2	1	27	7	27	7
HS1	2	1	78	7	926	7
HS24	2	5	18	7	20	7
HS25	3	6	143	7	10000	5
HS29	3	1	40	7	31	7
HS2	2	1	26	7	30	7
HS351	3	7	27	7	27	7
HS35MOD	3	3	60	7	60	7
HS35	3	4	28	7	28	7
HS36	3	7	22	7	18	7
HS37	3	8	30	7	26	7
HS38	4	8	93	7	10000	5
HS3MOD	2	1	19	0	18	0
HS3	2	1	23	7	10000	5
HS43	4	3	35	7	35	7
HS44NEW	4	10	23	7	20	7
HS45	5	10	23	2	149	7
HS4	2	2	15	7	16	7
HS57	2	3	160	0	10000	5
HS5	2	4	51	7	51	7
HS67	3	20	31	0	50	0
HS70	4	9	50	0	209	0
HS76I	4	11	22	0	22	0
HS76	4	7	20	7	20	7
HS84	5	16	36	7	4171	7
HS85	5	48	31	7	10000	5
HS93	6	8	208	7	10000	5
KSIP	20	1001	53	0	53	0
LOGROS	2	2	180	7	3416	7
MAXLIKA	8	16	67	7	1519	7
MDHOLE	2	1	147	7	1822	7
MIFFLINI	3	2	20	7	19	7
MINMAXBD	5	20	7	6	130	0

TAB. 1.6 – Résultats de NOPTIQ obtenus en utilisant la correction de Powell ou le skipping.

Probleme			powell		skip	
nom	n	m	nf	info	nf	info
OSLBQP	8	11	29	0	29	0
PALMER1A	6	2	141	7	10000	5
PALMER1B	4	2	89	7	10000	5
PALMER1E	8	1	201	7	10000	5
PALMER1	4	3	894	7	10000	5
PALMER2A	6	2	233	0	10000	5
PALMER2B	4	2	65	7	10000	5
PALMER2E	8	1	275	7	10000	5
PALMER2	4	3	188	0	4630	7
PALMER3A	6	2	362	0	1300	0
PALMER3B	4	2	126	7	10000	5
PALMER3E	8	1	172	0	10000	5
PALMER3	4	3	593	7	1429	7
PALMER4A	6	2	252	0	10000	5
PALMER4B	4	2	149	7	4827	7
PALMER4E	8	1	139	7	10000	5
PALMER4	4	3	37	6	10000	5
PALMER5A	8	2	10000	5	10000	5
PALMER5B	9	2	1235	0	10000	5
PALMER5D	8	4	38	7	10000	5
PALMER5E	8	1	10000	5	10000	5
PALMER6A	6	2	445	7	10000	5
PALMER6E	8	1	187	0	10000	5
PALMER7A	6	2	10000	5	10000	5
PALMER7E	8	1	1121	7	10000	5
PALMER8A	6	2	221	7	10000	5
PALMER8E	8	1	96	0	10000	5
PROBPENL	500	1000	10	7	10	7
PSPDOC	4	1	27	7	27	7
S368	8	16	74	7	28	7
SIM2BQP	2	2	20	7	20	7
SIMBQP	2	2	23	7	23	7
SIPOW1M	2	2000	22	0	27	7
SIPOW1	2	2000	23	0	32	7
SIPOW2M	2	2000	52	0	47	7
SIPOW2	2	2000	51	0	63	7
SIPOW3	4	2000	30	0	44	7
SIPOW4	4	2000	29	0	70	7
SPECAN	9	18	69	7	2724	7
SPIRAL	3	2	236	7	10000	5
WEEDS	3	4	77	7	81	7
WOMFLET	3	3	116	2	8947	6
YFIT	3	1	158	0	6752	7
ZY2	3	6	19	7	19	7

TAB. 1.7 – Résultats de NOPTIQ obtenus en utilisant la correction de Powell ou le skipping.

dans les tableaux 1.8 et 1.9. Le nombre maximal d'appels au simulateur autorisé est initialisé à 10000 pour les deux codes et NOPTIQ utilise la mise à jour des facteurs de Cholesky pour mettre à jour l'information du second ordre. La précision sur les tests d'arrêt des deux codes est égale à 10^{-8} . Dans les tableaux 1.8 et 1.9, *nf* représente le nombre d'appels cumulés au simulateur pour évaluer la fonction objectif et les contraintes et *info* indique le mode d'arrêt des deux codes. La valeur de la variable *info* indique le mode d'arrêt de LANCELOT :

- 0 LANCELOT stoppe normalement,
- 1 LANCELOT stoppe car le nombre maximal d'appels au simulateur est atteint,
- 3 LANCELOT stoppe car le pas initial dans la recherche linéaire est trop petit,
- 8 LANCELOT stoppe car il n'arrive pas à satisfaire les contraintes.

probleme			lancelot				noptiq	
			default		bfgs		cholesky	
nom	n	m	info	nf	nf	info	nf	info
3PK	30	30	0	17	17	0	246	7
ALLINIT	4	3	0	6	25	0	1	6
BQP1VAR	1	2	0	1	1	0	18	7
BQPGABIM	50	92	0	4	10000	1	65	7
BQPGASIM	50	100	0	4	10000	1	72	7
CAMEL6	2	4	0	5	17	0	32	7
CHEBYQAD	100	200	0	140	10000	1	1856	7
DECONVB	61	72	0	110	163	0	1898	2
DPIGRI	7	4	0	59	1728	0	30	7
EG1	3	4	0	6	18	0	46	7
HART6	6	12	0	9	9	0	75	7
HATFLDA	4	4	0	24	39	0	49	0
HATFLDB	4	5	0	21	31	0	28	0
HATFLDC	25	48	0	5	4	0	77	7
HIMMELPI	2	4	0	11	60	3	19	0
HS1	2	1	0	33	36	0	78	7
HS100	7	4	0	56	1540	0	29	7
HS100MOD	7	4	3	131	1268	3	33	7
HS110	200	400	0	1	1	0	1	6
HS113	10	8	3	94	184	8	25	7
HS117	15	20	0	69	324	0	30	2
HS12	2	1	0	24	71	0	27	7
H@S2	2	1	0	7	6	0	26	7
HS24	2	5	0	8	23	0	18	7
HS25	3	6	0	29	10000	1	143	7
HS29	3	1	0	31	48	0	40	7
HS3	2	1	0	4	4	0	23	7
HS35	3	4	0	9	67	0	28	7
HS351	3	7	0	9	67	0	27	7
HS35MOD	3	3	0	1	40	0	60	7
HS36	3	7	0	11	19	0	22	7
HS37	3	8	3	18	77	0	30	7
HS38	4	8	0	58	1900	0	93	7
HS3MOD	2	1	0	4	4	0	19	0
HS4	2	2	0	1	1	0	15	7
HS43	4	3	0	25	10000	1	35	7
HS44NEW	4	10	0	7	16	0	23	7
HS45	5	10	0	2	2	0	23	2
HS5	2	4	0	4	4	0	51	7
HS57	2	3	0	15	10000	1	160	0
HS67	3	20	0	68	10000	1	31	0
HS70	4	9	0	35	10000	1	50	0
HS76	4	7	0	8	61	0	20	0
HS761	4	11	0	8	61	0	22	7
HS84	5	16	3	74	91	3	36	7
HS85	5	48	8	5134	4445	3	31	7
HS93	6	8	8	5	5	8	208	7
KSIP	20	1001	3	102	99	3	53	0
LOGROS	2	2	0	70	67	0	180	7
MAXLIKA	8	16	0	10	10000	1	67	7
MDHOLE	2	1	0	63	67	0	147	7
MIFFLINI	3	2	0	13	15	0	20	7
MINMAXBD	5	20	0	596	601	0	7	6
OSLBQP	8	11	0	1	2	0	29	0

TAB. 1.8 – Résultats de LANCELOT et de NOPTIQ.

probleme			lancelot				noptiq	
			default		bfgs		cholesky	
nom	n	m	info	nf	nf	info	nf	info
PALMER1	4	3	0	30	38	0	894	7
PALMER1A	6	2	0	67	211	3	141	7
PALMER1B	4	2	0	39	86	3	89	7
PALMER1E	8	1	0	230	10000	1	201	7
PALMER2	4	3	0	21	131	0	188	0
PALMER2A	6	2	0	159	503	0	233	0
PALMER2B	4	2	0	100	106	3	65	7
PALMER2E	8	1	0	134	10000	1	275	7
PALMER3	4	3	0	38	6144	0	593	7
PALMER3A	6	2	0	148	10000	1	362	0
PALMER3B	4	2	0	65	57	3	126	7
PALMER3E	8	1	0	77	10000	1	172	0
PALMER4	4	3	0	40	10000	1	37	6
PALMER4A	6	2	0	47	130	0	252	0
PALMER4B	4	2	0	34	34	3	149	7
PALMER4E	8	1	0	56	10000	1	139	7
PALMER5A	8	2	3	6472	6854	3	10000	5
PALMER5B	9	2	0	5309	10000	1	1235	0
PALMER5D	8	4	0	9	9	0	38	7
PALMER5E	8	1	0	9	10000	1	10000	5
PALMER6A	6	2	0	300	630	0	445	7
PALMER6E	8	1	0	10	10000	1	187	0
PALMER7A	6	2	3	9736	10000	1	10000	5
PALMER7E	8	1	3	3662	29	0	1121	7
PALMER8A	6	2	0	82	471	0	221	7
PALMER8E	8	1	0	73	10000	1	96	0
PROBPENL	500	1000	0	2517	10000	1	10	7
PSPDOC	4	1	0	6	2	0	27	7
S368	8	16	0	4	10000	1	74	7
SIM2BQP	2	2	0	1	1	0	20	7
SIMBQP	2	2	0	4	4	0	23	7
SIPOW1	2	2000	0	13	13	0	23	0
SIPOW1M	2	2000	3	49	49	3	22	0
SIPOW2	2	2000	0	48	48	0	51	0
SIPOW2M	2	2000	0	39	39	0	52	0
SIPOW3	4	2000	0	38	38	0	30	0
SIPOW4	4	2000	3	54	54	3	29	0
SPECAN	9	18	0	9	14	0	69	7
SPIRAL	3	2	0	86	185	0	236	7
WEEDS	3	4	3	56	10000	1	77	7
WOMFLET	3	3	0	46	55	0	116	2
YFIT	3	1	0	96	6100	0	158	0
ZY2	3	6	0	10	11	0	19	7

TAB. 1.9 – Résultats de LANCELOT et de NOPTIQ.

On remarque que LANCELOT utilisant les dérivées secondes exactes des fonctions est plus performant que les autres codes sur 62 problèmes, équivalentes au code LANCELOT utilisant les formules de BFGS sur 15 problèmes. Ce dernier code est plus performant que les autres codes sur 5 problèmes. Le fait que LANCELOT soit plus performant que NOPTIQ sur de nombreux problèmes est normal car il utilise les dérivées secondes exactes des fonctions tandis que NOPTIQ utilise une approximation de celles-ci. Ce qui est plus surprenant c'est que NOPTIQ est plus performant que LANCELOT sur 15 problèmes.

Si on compare plus particulièrement les versions utilisant des approximations des dérivées secondes des fonctions des problèmes, on remarque que NOPTIQ est plus performant que LANCELOT sur 49 problèmes et moins performant sur 45 problèmes. Le graphique 1.1 donne une interprétation de ces résultats.

Notons q_i la valeur suivante

$$q_i = -\log_2 \left(\frac{nf_A}{nf_B} \right),$$

où nf_A est le nombre d'appels cumulés au simulateur avec le code NOPTIQ, nf_B le nombre d'appels cumulés au simulateur avec le code LANCELOT, et i est le $i^{\text{ème}}$ problème testé. Le signe de q_i indique le code nécessitant le moins d'appels au simulateur. Si ce signe est positif, NOPTIQ est plus performante et s'il est négatif, c'est le code LANCELOT qui l'est. Plus la valeur de $|q_i|$ est importante, plus le code est performant par rapport à l'autre. Le graphique 1.1 représente les valeurs de q_i pour les problèmes testés. Les problèmes ont été remplacés selon l'ordre décroissant de la valeur de $|q_i|$ ([44]). On précise qu'on ne compare sur ce graphique que les problèmes pour lesquels les deux codes ont donné la même solution.

1.5 Conclusion

Le code NOPTIQ est donc performant théoriquement et numériquement comme le montrent les résultats numériques présentés ci-dessus. Il est robuste car c'est une méthode simple : à chaque itération, on résout un système linéaire et on calcule un pas. Il y a peu de paramètres à régler. La comparaison de NOPTIQ avec le code LANCELOT qui est un code de référence, est prometteuse. Ces constatations nous ont motivé pour étudier l'utilisation de cet algorithme pour la résolution de problèmes de grande taille. Pour ce type de problèmes, les formules de BFGS décrites dans ce chapitre ne sont plus adaptées. On va plutôt utiliser les formules de BFGS à mémoire limitée.

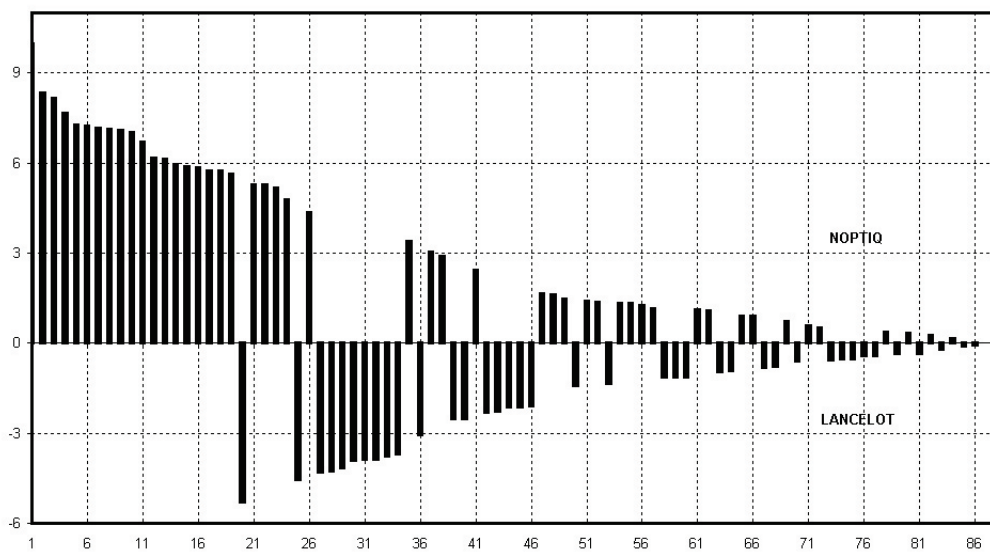


FIG. 1.1 – *Comparaison de NOPTIQ et LANCELOT.*

Chapitre 2

Utilisation des formules de quasi-Newton à mémoire limitée.

2.1 Introduction

On présente dans ce chapitre un algorithme de quasi-Newton à mémoire limitée pour résoudre un problème de minimisation de grande taille avec des contraintes d'inégalité. Le problème est le suivant :

$$(\mathcal{P}) \begin{cases} \min f(x) \\ c(x) \leq 0, \\ l \leq x \leq u, \end{cases} \quad (2.1)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $c : \mathbb{R}^n \rightarrow \mathbb{R}^p$ sont des fonctions deux fois continûment différentiables dans \mathbb{R}^n , où $-\infty \leq l < u \leq +\infty$ représentent les bornes sur les variables. et où le nombre de variables n est supposé grand. On suppose que les fonctions f et c sont convexes.

L'algorithme présenté dans ce chapitre est une mise en œuvre de la méthode de points intérieurs décrite dans le chapitre précédent utilisant non plus les formules de quasi-Newton pour approcher les dérivées secondes du problème mais celles de quasi-Newton à mémoire limitée. En effet, on ne peut pas appliquer directement les formules de quasi-Newton à la résolution de problèmes de grande taille car l'espace mémoire requis pour stocker les approximations du hessien ou de son inverse devient vite important lorsque n est grand. Des modifications et des extensions de ces formules ont donc été développées pour les problèmes de grande taille.

En 1980, Nocedal ([49]) a introduit les formules de quasi-Newton à mémoire limitée en optimisation sans contrainte. Ces formules modifient les techniques de quasi-Newton pour obtenir des approximations du hessien ou de son inverse stockables de manière compacte en utilisant un petit nombre de vecteurs. De nombreuses études numériques sur des problèmes de grande taille ont montré que des méthodes basées sur ces formules sont très efficaces, robustes et peu coûteuses en espace mémoire et en temps de calculs ([25, 48]).

Une autre méthode pour utiliser les formules de quasi-Newton dans le cas de problèmes de grande taille est basée sur la structure du hessien de la fonc-

tion objectif de ces problèmes ([31, 32]). En effet, la fonction objectif des problèmes de grande taille possède parfois la propriété dite de séparabilité : elle peut se décomposer comme somme de fonctions faisant intervenir chacune un petit nombre de variables. On utilise alors les formules de quasi-Newton pour approcher le hessien de ces fonctions.

En 1989, le code 1-BFGS développé par J. Nocedal et D. C. Liu ([38]) et le code N1QN3 développé par J.-Ch. Gilbert et C. Lemaréchal ([25]) résolvent des problèmes d'optimisation sans contrainte en utilisant les formules de quasi-Newton à mémoire limitée. En 1994, le code 1-BFGS-B est une extension de l'algorithme 1-BFGS qui permet de prendre en compte les contraintes de bornes sur les variables ([11]). Les codes LOQO ([58]), KNITRO ([9, 10]) SNOPT ([26]) et LANCELOT ([15]) utilisent la propriété de séparabilité de la fonction objectif pour résoudre des problèmes de grande taille avec des contraintes d'inégalité. Cette technique demande des renseignements sur les données du problème qui sont parfois difficiles à obtenir. Aussi, nous présentons dans ce chapitre le code NOPTIQ qui utilise la représentation compacte des formules de quasi-Newton introduites dans [13] pour résoudre des problèmes de grande taille avec des contraintes d'inégalité sans information particulière sur la structure de la fonction objectif. La prise en compte de ce type de contraintes est un apport important dans l'utilisation des formules de BFGS à mémoire limitée. En effet, les problèmes industriels sont souvent des problèmes de grande taille avec des contraintes de bornes sur les variables et quelques contraintes d'inégalité. On montre aussi que l'emploi des formules de BFGS à mémoire limitée diminue à la fois l'espace mémoire nécessaire au calcul de la direction de recherche et le coût en opérations de ce calcul. Notons n le nombre de variables et p le nombre de contraintes d'inégalité (autres que les contraintes de bornes) du problème à résoudre. Au lieu de stocker en mémoire $O(n^2)$ informations, on va stocker $O(nm_\ell)$ informations, où m_ℓ est un nombre généralement petit choisi par l'utilisateur suivant l'espace mémoire disponible. Le calcul de la direction primale se fera alors en $O(nm_\ell^2)$ opérations dans le cas de problèmes avec uniquement des contraintes de bornes, et en $O(n(2p^2 + 7m_\ell^2/2))$ opérations dans le cas de problèmes avec contraintes d'inégalité générale. Le code NOPTIQ utilisant les formules de BFGS à mémoire limitée fait l'objet d'un dépôt de logiciel à l'Agence pour la Protection des Programmes ([5]) et est disponible sur simple demande auprès de Paul Armand ou de moi-même.

On va tout d'abord rappeler ce que sont les formules de BFGS à mémoire limitée ainsi que leur représentation compacte. On va ensuite détailler l'emploi de ces formules au calcul de la direction de recherche en distinguant deux cas : soit les problèmes n'a que des contraintes de bornes, soit il a aussi des contraintes d'inégalité générale. Une étude de convergence montrera que l'algorithme A_μ du chapitre précédent utilisant les formules de quasi-Newton à mémoire limitée converge r -linéairement. Des résultats numériques du code NOPTIQ utilisant la mémoire limitée concluront cette étude.

2.2 Représentation compacte des formules de BFGS à mémoire limitée

La formule de BFGS directe (ou à mémoire pleine par opposition avec celles à mémoire limitée) est la suivante

$$M_{k+1} = M_k - \frac{M_k s_k s_k^\top M_k}{s_k^\top M_k s_k} + \frac{y_k y_k^\top}{y_k^\top s_k}, \quad (2.2)$$

où s_k et y_k sont deux vecteurs de \mathbb{R}^n . Cette formule nécessite le stockage d'une matrice pleine de taille n^2 . Le coût en calculs et en espace mémoire engendré par l'emploi de cette matrice devient trop important dans le cas de problème de grande taille. L'idée des méthodes de BFGS à mémoire limitée introduite par Nocedal ([49]) en 1980 est donc de diminuer ces coûts. On remarque que, dans (2.2), la matrice M_k est corrigée à l'aide d'un couple de vecteurs (s_k, y_k) . La méthode de BFGS à mémoire limitée consiste à ne conserver en mémoire qu'un petit nombre de ces couples pour reconstruire rapidement la matrice M_{k+1} . Quand la capacité de stockage est atteinte, on élimine alors la paire de vecteurs la plus ancienne pour la remplacer par la dernière calculée. Les formules pour reconstruire la matrice M_{k+1} sont dérivées de celles des méthodes de BFGS, d'où leur nom de formules de BFGS à mémoire limitée. Notons m_ℓ le nombre de paires de vecteurs conservées. Pendant les m_ℓ premières itérations, la méthode est identique à celle de BFGS. Pour $k > m_\ell$, on applique m_ℓ mises à jour de BFGS à une matrice M_0 (de mise à l'échelle) en utilisant les m_ℓ paires de vecteurs stockées.

2.2.1 Formule récursive

On peut écrire la formule (2.2) sous la forme

$$M_{k+1} = M_k - a_k a_k^\top + b_k b_k^\top,$$

où les vecteurs a_k et b_k sont définis par

$$a_k = \frac{M_k s_k}{(s_k^\top M_k s_k)^{1/2}} \quad \text{et} \quad b_k = \frac{y_k}{(y_k^\top s_k)^{1/2}}.$$

Une méthode de BFGS à mémoire limitée serait de définir une matrice initiale M_k^0 à chaque itération et la mise à jour de la matrice M_k se ferait alors avec la formule suivante

$$M_k = M_k^0 + \sum_{i=k-m_\ell}^{k-1} (b_i b_i^\top - a_i a_i^\top).$$

Les paires de vecteurs $\{a_i, b_i\}$ pour i variant de $k - m_\ell$ à $k - 1$ seraient alors calculées à partir des vecteurs $\{s_i, y_i\}$ ($i = k - m_\ell, \dots, k - 1$) à l'aide de la procédure suivante

Pour $i = k - m_\ell, \dots, k - 1$
 $b_i \leftarrow y_i / (y_i^\top s_i)^{1/2}$
 $a_i \leftarrow M_k^0 s_i + \sum_{j=k-m_\ell}^{i-1} [(b_j^\top s_i) b_j - (a_j^\top s_i) a_j]$
 $a_i \leftarrow a_i / (s_i^\top a_i)^{1/2}$
fin (pour)

On remarque que les vecteurs a_i doivent être recalculés à chaque itération alors que le vecteur b_i et la valeur $b_i^\top s_i$ peuvent être conservés en mémoire au cours des itérations. Si on suppose que la matrice initiale M_k^0 est égale à la matrice identité, le nombre d'opérations nécessaires pour déterminer la matrice M_k (c'est à dire les vecteurs (a_i, b_i)) est approximativement de l'ordre de $\frac{3}{2} m_\ell^2 n$. Le produit de la matrice M_k par un vecteur arbitraire v de \mathbb{R}^n est alors de $4m_\ell n$ opérations. Le coût total de mise à jour et de calcul est donc de $\frac{3}{2} m_\ell^2 n + 4m_\ell n$ opérations. Lorsque le nombre de variables n est important, cette approche est donc plus efficace que les méthodes de BFGS à mémoire pleine qui nécessitent $3n^2 + 3n$ opérations. On va maintenant voir une représentation compacte des formules de BFGS à mémoire limitée. Cette représentation va diminuer le coût de calcul. On va en effet passer de $\frac{3}{2} m_\ell^2 n$ à $2m_\ell n$ opérations.

2.2.2 Représentation compacte

La représentation compacte de ces formules introduite dans [13, théorème 2.3] est la suivante.

Théorème 2.2.1 *Soit M_0 une matrice symétrique définie positive. Supposons que les $2k$ vecteurs $\{s_i, y_i\}_{i=0}^{k-1}$ de \mathbb{R}^n satisfassent $s_i^\top y_i > 0$ pour tout i dans $\{0, \dots, k-1\}$. Soit M_k la matrice obtenue après k mises à jour de la matrice M_0 avec la formule de BFGS (2.2) en utilisant ces vecteurs. On a alors*

$$M_k = M_0 - [M_0 S_k \quad Y_k] \begin{bmatrix} S_k^\top M_0 S_k & L_k \\ L_k^\top & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^\top M_0 \\ Y_k^\top \end{bmatrix}, \quad (2.3)$$

où S_k et Y_k sont les deux matrices de taille $n \times k$ définies par

$$S_k = [s_0, \dots, s_{k-1}] \text{ et } Y_k = [y_0, \dots, y_{k-1}],$$

où L_k est la matrice triangulaire inférieure de taille $k \times k$ définie par

$$(L_k)_{ij} = \begin{cases} s_{i-1}^\top y_{j-1} & \text{si } i > j \\ 0 & \text{sinon} \end{cases}$$

et où D est la matrice diagonale de taille $k \times k$ définie par

$$D_k = \text{diag}[s_0^\top y_0, \dots, s_{k-1}^\top y_{k-1}].$$

2.2. Représentation compacte des formules de BFGS à mémoire limitée

La condition sur la stricte positivité des produits $s_i^\top y_i$ pour tout i dans $\{0, \dots, k-1\}$ assure que la matrice à inverser dans la formule (2.3) est non singulière.

Les méthodes de BFGS à mémoire limitée utilisant cette représentation compacte consistent à garder en mémoire les m_ℓ dernières paires de vecteurs de corrections (s_i, y_i) . Cet ensemble de paires est mis à jour à chaque itération en effaçant la plus vieille paire conservée et en la remplaçant par la nouvelle paire calculée. Durant les m_ℓ premières itérations, la procédure de mise à jour décrite dans le théorème 2.2.1 est identique à celle de mise à jour des formules de BFGS à mémoire pleine. Puis, ayant atteint le nombre maximal de paires à sauvegarder, la procédure de mise à jour doit être légèrement modifiée pour prendre en compte la nature des paires de vecteurs sauvegardées $\{s_i, y_i\}$ pour i variant de $k - m_\ell$ à $k - 1$. On définit alors les matrices S_k et Y_k de taille $n \times m_\ell$ par

$$S_k = [s_{k-m_\ell}, \dots, s_{k-1}] \text{ et } Y_k = [y_{k-m_\ell}, \dots, y_{k-1}].$$

On obtient la matrice M_k après m_ℓ mises à jour de la matrice M_0 :

$$M_k = M_0 - [M_0 S_k \quad Y_k] \begin{bmatrix} S_k^\top M_0 S_k & L_k \\ L_k^\top & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^\top M_0 \\ Y_k^\top \end{bmatrix}, \quad (2.4)$$

où L_k est la matrice triangulaire inférieure de taille $m_\ell \times m_\ell$ définie par

$$(L_k)_{ij} = \begin{cases} s_{k-m_\ell-1+i}^\top y_{k-m_\ell-1+j} & \text{si } i > j \\ 0 & \text{sinon} \end{cases}$$

et où D est la matrice diagonale de taille $m_\ell \times m_\ell$ définie par

$$D_k = \text{diag}[s_{k-m_\ell}^\top y_{k-m_\ell}, \dots, s_{k-1}^\top y_{k-1}].$$

Une fois le nouvel itéré x_{k+1} calculé, on obtient la matrice S_{k+1} en effaçant le vecteur s_{k-m_ℓ} de S_k et en le remplaçant par le vecteur s_k . On met à jour la matrice Y_{k+1} de manière similaire. La matrice diagonale D_{k+1} est mise à jour en effaçant le produit $s_{k-m_\ell}^\top y_{k-m_\ell}$ et en le remplaçant par le produit $s_k^\top y_k$. On passe finalement de la matrice L_k à la matrice L_{k+1} en effaçant la première colonne de la matrice L_k , en décalant la matrice restante et en calculant les $n - 1$ produits scalaires $s_k^\top y_{k-m_\ell+i}$ pour i variant de 1 à m_ℓ . La figure 2.1 illustre graphiquement ces mise à jours pour les matrices L_{k+1} et D_{k+1} : les zones hachurées représentent les parties sauvegardées ; celles rayées, les parties à calculer et celles blanches, les parties effacées.

Comme la matrice du milieu dans la formule de mise à jour (2.4) est petite, de dimension $2m_\ell$, sa factorisation nécessite peu de calculs. L'idée principale de la représentation compacte (2.4) est le fait que les corrections faites d'une matrice initiale M_0 peuvent s'écrire sous la forme d'un produit de deux matrices allongées, $[M_0 S_k \quad Y_k]$ et sa transposée, dans lequel intervient une multiplication par une matrice de petite taille $m_\ell \times m_\ell$ (voir la figure 2.2).

La procédure de mise à jour de la matrice M_k en utilisant les formules de BFGS à mémoire limitée dans leur représentation compacte nécessite approximativement $2m_\ell n + O(m_\ell^3)$ opérations si la matrice initiale M_0 est de la forme

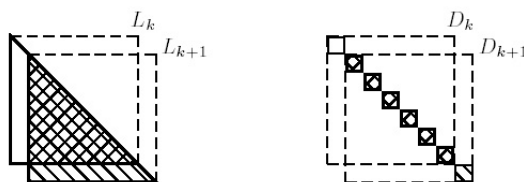


FIG. 2.1 – Mise à jour des matrices L_{k+1} et D_{k+1} .

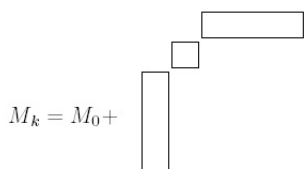


FIG. 2.2 – Représentation compacte de M_k .

$\sigma_k I$ ($\sigma_k > 0$). Le produit matrice vecteur de la forme $M_k v$ où v est un vecteur arbitraire de \mathbb{R}^n se fait alors en $(4m_\ell + 1)n + O(m_\ell^2)$ multiplications. Ce détail des opérations indique donc que l'emploi de ces formules est économique lorsque le nombre de paires sauvegardées m_ℓ est petit.

2.3 Une méthode de points intérieurs à mémoire limitée

On va présenter dans cette partie un algorithme de points intérieurs basé sur la méthode vue au chapitre 1, utilisant la représentation compacte des formules de BFGS à mémoire limitée.

2.3.1 Utilisation des formules de quasi-Newton à mémoire limitée

On va utiliser les formules de BFGS à mémoire limitée et leur représentation compacte pour résoudre la première équation du système de Newton (1.29). Dans cette équation, la matrice M_k est maintenant mise à jour avec les formules de BFGS à mémoire limitée et approche le hessien du lagrangien $\nabla_{xx}^2 \ell(x, \lambda)$ en x . Notons N la matrice symétrique semi-définie positive définie par

$$N = \nabla c(x) \Lambda (-C(x))^{-1} \nabla c(x)^T \tag{2.5}$$

et b le vecteur de \mathbb{R}^n défini par

$$b = -\nabla f(x) + \mu \nabla c(x) C(x)^{-1} e. \tag{2.6}$$

Pour obtenir la direction de recherche primale d^x , on doit donc résoudre l'équation suivante

$$(M_k + N)d^x = b. \quad (2.7)$$

On va tout d'abord rappeler la notion de complément de Schur d'une matrice ([50]). Soit D une matrice non singulière et X une matrice de taille n^2 de la forme

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

On peut factoriser X de la manière suivante

$$X = \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I & 0 \\ D^{-1}C & I \end{bmatrix}.$$

où $S = A - BD^{-1}C$ est appelé le complément de Schur de D dans la matrice X . Par hypothèse de non singularité de la matrice D , cette matrice est bien définie. L'inverse de la matrice X est alors donné par la formule suivante

$$X^{-1} = \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} \cdot \begin{bmatrix} S^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix}.$$

Rappelons la formule de Sherman-Morrison-Woodbury ([50]) pour calculer l'inverse d'une matrice \bar{A} définie par

$$\bar{A} = A - XDY^T$$

où les matrices A , D et $(D^{-1} - Y^T A^{-1} X)$ sont non singulières. L'inverse de \bar{A} est alors donné par la formule

$$\bar{A}^{-1} = A^{-1} + A^{-1} X (D^{-1} - Y^T A^{-1} X)^{-1} Y^T A^{-1}. \quad (2.8)$$

La proposition suivante donne une représentation compacte de l'inverse de $M_k + N$ après k itérations où M_k est mise à jour en utilisant la représentation compacte des formules de BFGS à mémoire limitée.

Proposition 2.3.1 *Soit M_0 une matrice symétrique définie positive. Soit N une matrice symétrique semi-définie positive. Supposons que les k paires de vecteurs $\{s_i, y_i\}_{i=0}^{k-1}$ satisfassent $s_i^T y_i > 0$ pour tout i dans $\{0, \dots, k-1\}$. Soit M_k la matrice obtenue après k mises à jour de la matrice M_0 avec la formule de BFGS (2.2) en utilisant ces vecteurs. L'inverse de $M_k + N$ est alors donné par*

$$(M_k + N)^{-1} = (M_0 + N)^{-1} + (M_0 + N)^{-1} U_k E_k^{-1} U_k^T (M_0 + N)^{-1},$$

où S_k et Y_k sont les deux matrices de $n \times k$ définies par

$$S_k = [s_0 \ \dots \ s_{k-1}] \quad \text{et} \quad Y_k = [y_0 \ \dots \ y_{k-1}],$$

où U_k est la matrice de $n \times (2k)$ définie par

$$U_k = [M_0 S_k \ Y_k],$$

où E_k , \tilde{L}_k et \tilde{D}_k sont les matrices de taille $k \times k$ définies par

$$\begin{aligned} E_k &= \begin{bmatrix} S_k^\top \tilde{M}_0 S_k & \tilde{L}_k \\ \tilde{L}_k^\top & -\tilde{D}_k \end{bmatrix} \\ \tilde{L}_k &= L_k - S_k^\top M_0 (M_0 + N)^{-1} Y_k, \\ \tilde{D}_k &= D_k + Y_k^\top (M_0 + N)^{-1} Y_k. \end{aligned}$$

et où \tilde{M}_0 est la matrice de taille $n \times n$ définie par

$$\tilde{M}_0 = M_0 - M_0 (M_0 + N)^{-1} M_0.$$

De plus, le complément de Schur de $-\tilde{D}_k$ de la matrice E_k , c'est à dire la matrice de taille $k \times k$

$$S_k^\top \tilde{M}_0 S_k + \tilde{L}_k \tilde{D}_k^{-1} \tilde{L}_k^\top,$$

est définie positive.

Preuve. En utilisant la représentation compacte (2.3) de M_k et en notant $U_k = [M_0 S_k \ Y_k]$, on a

$$M_k + N = (M_0 + N) - U_k \begin{bmatrix} S_k^\top \tilde{M}_0 S_k & L_k \\ L_k^\top & -D_k \end{bmatrix}^{-1} U_k^\top.$$

Puis en appliquant la formule de Sherman-Morrison-Woodbury (2.8) à la matrice $M_k + N$, on obtient

$$(M_k + N)^{-1} = (M_0 + N)^{-1} + (M_0 + N)^{-1} U_k E_k^{-1} U_k^\top (M_0 + N)^{-1},$$

où

$$E_k = \begin{bmatrix} S_k^\top \tilde{M}_0 S_k & L_k \\ L_k^\top & -D_k \end{bmatrix} - U_k^\top (M_0 + N)^{-1} U_k.$$

On en déduit ensuite facilement les formules données pour les matrices E_k , \tilde{L}_k et \tilde{D}_k . La première partie de la proposition est alors démontrée.

Montrons maintenant que la matrice E_k est non singulière. Remarquons tout d'abord qu'étant donné que les k paires $\{s_i, y_i\}_{i=0}^{k-1}$ satisfont l'inégalité

$$s_i^\top y_i > 0$$

pour tout i dans $\{0, \dots, k-1\}$, la matrice D_k est définie positive. Par construction, la matrice \tilde{D}_k l'est aussi.

On vérifie par un simple calcul que E_k a une factorisation de la forme

$$E_k = \begin{bmatrix} I & -\tilde{L}_k \tilde{D}_k^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} S^\top \tilde{M}_0 S + \tilde{L}_k \tilde{D}_k^{-1} \tilde{L}_k^\top & 0 \\ 0 & -\tilde{D}_k \end{bmatrix} \begin{bmatrix} I & 0 \\ -\tilde{D}_k^{-1} \tilde{L}_k^\top & I \end{bmatrix}.$$

Puis en notant $K = S^\top \tilde{M}_0 S + \tilde{L}_k \tilde{D}_k^{-1} \tilde{L}_k^\top$ le complément de Schur de $-\tilde{D}_k$ dans E_k , on obtient facilement l'inverse de E_k . Il est donné par la formule suivante

$$\begin{aligned} E_k^{-1} &= \begin{bmatrix} I & 0 \\ \tilde{D}_k^{-1} \tilde{L}_k^\top & I \end{bmatrix} \begin{bmatrix} K^{-1} & 0 \\ 0 & -\tilde{D}_k^{-1} \end{bmatrix} \begin{bmatrix} I & \tilde{L}_k \tilde{D}_k^{-1} \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} K^{-1} & K^{-1} \tilde{L}_k \tilde{D}_k^{-1} \\ \tilde{D}_k^{-1} \tilde{L}_k^\top K^{-1} & \tilde{D}_k^{-1} \tilde{L}_k^\top K^{-1} \tilde{L}_k \tilde{D}_k^{-1} - \tilde{D}_k^{-1} \end{bmatrix}. \end{aligned}$$

Il nous reste à montrer, pour finir la preuve, que le complément de Schur K est non singulier. Montrons donc que le noyau de la matrice K est réduit à $\{0\}$. En notant la matrice \tilde{M}_0 sous la forme

$$\tilde{M}_0 = M_0 (M_0^{-1} - (M_0 + N)^{-1}) M_0,$$

on vérifie facilement que cette matrice est semi-définie positive. La matrice $\tilde{L}_k \tilde{D}_k^{-1} \tilde{L}_k^T$ est aussi une matrice semi-définie positive. Par définition de la semi-définie positivité de ces matrices, on a les inégalités suivantes pour tout vecteur $u \in \mathbb{R}^k$

$$u^T S_k^T \tilde{M}_0 S_k u \leq 0, \quad (2.9)$$

$$u^T \tilde{L}_k \tilde{D}_k^{-1} \tilde{L}_k^T u \leq 0. \quad (2.10)$$

Le noyau noté $\text{Ker}(K)$ de la matrice K est l'ensemble des vecteurs u de \mathbb{R}^k tels que

$$u^T (S_k^T \tilde{M}_0 S_k + \tilde{L}_k \tilde{D}_k^{-1} \tilde{L}_k^T) u = 0. \quad (2.11)$$

Par définie positivité de la matrice \tilde{D}_k et l'inégalité (2.10), on a pour tout vecteur $u \in \text{Ker}(K)$

$$\tilde{L}_k^T u = 0. \quad (2.12)$$

Montrons maintenant que, pour tout vecteur $u \in \text{Ker}(K)$, on a

$$\tilde{M}_0 S_k u = 0. \quad (2.13)$$

Si $u \in \text{Ker}(K)$ et, en tenant compte de l'inégalité (2.9), on a

$$u^T S_k^T \tilde{M}_0 S_k u = 0. \quad (2.14)$$

Notons v le vecteur défini par $v = S_k u$. D'après l'inégalité ci-dessus, ce vecteur v est orthogonal au vecteur $\tilde{M}_0 v$. Donc $v \in I(\tilde{M}_0^T)^\perp$. Or, $I(\tilde{M}_0^T)^\perp = \text{Ker}(\tilde{M}_0)$. Donc $v \in \text{Ker}(\tilde{M}_0)$ et donc, en réécrivant la valeur de v , on a

$$\tilde{M}_0 S_k u = 0. \quad (2.15)$$

Par définition de \tilde{M}_0 , on a donc

$$S_k u = (M_0 + N)^{-1} M_0 S_k u,$$

et par définition de \tilde{L}_k , on a aussi

$$L_k^T u = Y_k^T (M_0 + N)^{-1} M_0 S_k u.$$

On obtient finalement que

$$(S_k^T Y_k - L_k)^T u = 0.$$

Par définition de L_k , la matrice $S_k^T Y_k - L_k$ est une matrice triangulaire supérieure avec des éléments strictement positifs sur la diagonale. On obtient donc que $u = 0$ et la non singularité du complément de Schur. \square

Lorsque le nombre d'itérations k devient supérieur au nombre total de paires stockées, cette proposition doit être légèrement modifiée pour prendre en compte la nature des paires de vecteurs sauvegardées $\{s_i, y_i\}$ pour i variant de $k - m_\ell$ à $k - 1$. On définit alors les matrices S_k et Y_k de taille $n \times m_\ell$ par

$$S_k = [s_{k-m_\ell}, \dots, s_{k-1}] \text{ et } Y_k = [y_{k-m_\ell}, \dots, y_{k-1}].$$

On obtient l'inverse de la matrice $(M_k + N)^{-1}$ après m_ℓ mises à jour de la matrice M_0 :

$$(M_k + N)^{-1} = (M_0 + N)^{-1} + (M_0 + N)^{-1} U_k E_k^{-1} U_k^T (M_0 + N)^{-1}, \quad (2.16)$$

où U_k est la matrice de $n \times (2m_\ell)$ définie par

$$U_k = [M_0 S_k \quad Y_k],$$

où E_k , \tilde{L}_k et \tilde{D}_k sont les matrice de taille $k \times k$ définies par

$$\begin{aligned} E_k &= \begin{bmatrix} S_k^T \tilde{M}_0 S_k & \tilde{L}_k \\ \tilde{L}_k^T & -\tilde{D}_k \end{bmatrix} \\ \tilde{L}_k &= L_k - S_k^T M_0 (M_0 + N)^{-1} Y_k, \\ \tilde{D}_k &= D_k + Y_k^T (M_0 + N)^{-1} Y_k, \end{aligned}$$

où \tilde{M}_0 est la matrice de taille $n \times n$ définie par

$$\tilde{M}_0 = M_0 - M_0 (M_0 + N)^{-1} M_0,$$

où L_k est la matrice triangulaire inférieure de taille $m_\ell \times m_\ell$ définie par

$$(L_k)_{ij} = \begin{cases} s_{k-m_\ell-1+i}^T y_{k-m_\ell-1+j} & \text{si } i > j \\ 0 & \text{sinon} \end{cases}$$

et où D est la matrice diagonale de taille $m_\ell \times m_\ell$ définie par

$$D_k = \text{diag}[s_{k-m_\ell}^T y_{k-m_\ell}, \dots, s_{k-1}^T y_{k-1}].$$

Une fois le nouvel itéré x_{k+1} calculé, on obtient les matrices S_{k+1} , Y_{k+1} , L_{k+1} et D_{k+1} en utilisant les formules de mises à jour de BFGS à mémoire limitée. Les matrices \tilde{M}_0 , \tilde{L}_k et \tilde{D}_k sont, elles, recalculées à chaque itération (la matrice N définie par (2.5) changeant à chaque itération).

On remarque que si $N = 0$ dans la proposition 2.3.1, on retrouve la représentation compacte de l'inverse de M_k , voir [13, Théorème 2.2]. En effet, si $N = 0$, alors

$$\begin{cases} \tilde{M}_0 = 0, \\ \tilde{L}_k = L_k - S_k^T Y_k = -R_k, \\ \tilde{D}_k = D_k + Y_k^T M_0 Y_k, \end{cases}$$

où R_k est une matrice triangulaire supérieure non singulière. On en déduit que

$$K = R_k \tilde{D}_k^{-1} R_k^T, \quad \tilde{D}_k^{-1} \tilde{L}_k^T K^{-1} = -R_k^{-1} \text{ et } \tilde{D}_k^{-1} \tilde{L}_k^T K^{-1} \tilde{L}_k \tilde{D}_k^{-1} - \tilde{D}_k^{-1} = 0.$$

On obtient alors

$$E_k^{-1} = \begin{bmatrix} R_k^{-T}(D_k + Y_k^T M_0 Y_k) R_k^{-1} & -R_k^{-1} \\ -R_k^{-1} & 0 \end{bmatrix},$$

et ainsi

$$M_k^{-1} = M_0^{-1} + [S_k \quad M_0^{-1} Y_k] E_k^{-1} [S_k \quad M_0^{-1} Y_k]^T.$$

On va maintenant utiliser la proposition 2.3.1 pour calculer la direction primale d^x . Cette direction est solution de l'équation

$$d^x = (M_k + N)^{-1} b \quad (2.17)$$

où N est la matrice de dimension $n \times n$ définie par (2.5) et b le vecteur de \mathbb{R}^n défini par (2.6). On va distinguer deux méthodes pour le calcul de d^x suivant le fait que la matrice N soit diagonale ou pas. Par construction de N , cette distinction va porter sur le type des contraintes rencontrées. Dans le cas de problèmes n'ayant que des contraintes de bornes, on va montrer que N est une matrice diagonale. Dans le cas de problèmes avec des contraintes de type plus général, on utilisera la structure de la jacobienne des contraintes pour diminuer le coût en calculs.

On supposera par la suite qu'on stocke k paires de vecteurs (s_i, y_i) tels que, pour tout $i \in \{0, \dots, k-1\}$, $s_i^T y_i > 0$. On notera S_k et Y_k les deux matrices de taille $n \times k$

$$S_k = [s_0 \quad \dots \quad s_{k-1}] \quad \text{et} \quad Y_k = [y_0 \quad \dots \quad y_{k-1}]$$

ainsi que les matrices L_k et D_k de taille $k \times k$ définies dans la proposition 2.3.1. On supposera aussi que $M_0 = \sigma_k I$ est une matrice diagonale, où

$$\sigma_k = \frac{y_{k-1}^T s_{k-1}}{s_{k-1}^T s_{k-1}}$$

(cette initialisation étant celle qui donne les meilleurs résultats en pratique, [25] et [38]).

2.3.2 Problèmes avec des contraintes de bornes

On considère dans cette partie le problème suivant

$$(\mathcal{P}) \begin{cases} \min f(x) \\ l \leq x \leq u \end{cases} \quad (2.18)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction continûment différentiable dans \mathbb{R}^n et où $-\infty \leq l < u \leq +\infty$ sont des vecteurs de \mathbb{R}^n représentant les bornes sur les variables. On note m^l le nombre de contraintes de bornes inférieures, m^u , celui des contraintes de bornes supérieures, $m = m^l + m^u$ le nombre total de contraintes de bornes et c le vecteur de \mathbb{R}^m des contraintes

$$c(x) = \begin{pmatrix} l - x \\ x - u \end{pmatrix}.$$

La jacobienne des contraintes peut donc s'écrire

$$\nabla c(x) = (\nabla c_1(x) \cdots \nabla c_m(x))$$

où, pour $i \in \{1, \dots, m^l\}$, le vecteur $\nabla c_i(x)$ est un vecteur de la base canonique de \mathbb{R}^n multiplié par (-1) , et où, pour $i \in \{1, \dots, m^u\}$, le vecteur $\nabla c_i(x)$ est un vecteur de la base canonique de \mathbb{R}^n . On remarque donc que la matrice jacobienne $\nabla c(x)$ est une matrice dont les colonnes sont au signe près une partition des colonnes de la matrice identité.

La formule (2.5) nous donne la matrice N :

$$N = (\nabla c_1(x) \cdots \nabla c_m(x)) \Lambda (-C(x))^{-1} \begin{pmatrix} \nabla c_1^\top(x) \\ \vdots \\ \nabla c_m^\top(x) \end{pmatrix}.$$

Et, donc en développant cette matrice, on a

$$N = (\nabla c_1(x) \cdots \nabla c_m(x)) \begin{pmatrix} \frac{\lambda_1}{-c_1(x)} \nabla c_1^\top(x) \\ \vdots \\ \frac{\lambda_m}{-c_m(x)} \nabla c_m^\top(x) \end{pmatrix}.$$

Finalement, en effectuant les produits, on obtient

$$N = \sum_{i=1}^m \frac{\lambda_i}{-c_i(x)} \nabla c_i(x) \nabla c_i^\top(x)$$

où, pour chaque indice i , le terme correspondant dans cette somme est une matrice de taille $n \times n$ dont toutes les composantes sont nulles sauf la composante (i, i) sur la diagonale égale à $\lambda_i / -c_i(x)$. La matrice N est donc une matrice diagonale dont chaque terme sur la diagonale est positif ou nul. Elle est de plus définie positive car les éléments de sa diagonale sont strictement positifs.

Avec la proposition 2.3.1, les égalités (1.29), (2.5) et (2.6), on obtient la formule suivante pour le calcul de d^x :

$$d^x = [(M_0 + N)^{-1} + (M_0 + N)^{-1} U_k E_k^{-1} U_k^\top (M_0 + N)^{-1}] b. \quad (2.19)$$

Proposition 2.3.2 *Soit M_0 une matrice symétrique définie positive. Soit N la matrice symétrique semi-définie positive définie par (2.5). Supposons que les k paires de vecteurs $\{s_i, y_i\}_{i=0}^{k-1}$ satisfassent $s_i^\top y_i > 0$ pour tout i dans $\{0, \dots, k-1\}$. Soit M_k la matrice obtenue après k mises à jour de la matrice M_0 avec la formule de BFGS (2.2) en utilisant ces vecteurs. Le coût du calcul de d^x est alors de $n(4k^2 + 7k + 4) + m + O(k^3) + 5/2k^2 + 3/2k$ opérations.*

Preuve. Par définie positivité des matrices diagonale M_0 et N , la matrice $(M_0 + N)$ est une matrice diagonale inversible donc l'inverse se calcule rapidement (en n opérations). On décompose alors le calcul de d^x avec l'algorithme \mathcal{A}_b suivant. On suppose disponibles au début du calcul les matrices S_{k-1} , Y_{k-1} , D_{k-1} , L_{k-1} , les vecteurs s_k et y_k vérifiant $s_k^\top y_k > 0$ et le vecteur b défini par (2.6).

ALGORITHME \mathcal{A}_b pour calculer d^x avec (2.19)

0. Calcul de N et de $(M_0 + N)^{-1}$.
1. Mise à jour de S_k, Y_k, D_k et L_k .
2. Calcul de \tilde{D}_k, \tilde{L}_k et $S_k^T \tilde{M}_0 S_k$.
3. Calcul de $d_1^x = (M_0 + N)^{-1} b$.
4. Calcul de $d_2^x = U_k^T d_1^x$.
5. Calcul de $d_3^x = E_k^{-1} d_2^x$.
6. Calcul de $d_4^x = U_k d_3^x$.
7. Calcul de $d^x = d_1^x + (M_0 + N)^{-1} d_4^x$.

L'étape 0 coûte m opérations pour le calcul de N et n opérations pour le calcul de l'inverse de $(M_0 + N)$. Le coût des mises à jour à l'étape 1 est de kn opérations (k produits scalaires de taille n). Dans l'étape 2, on calcule les matrices \tilde{D}_k, \tilde{L}_k et $S_k^T \tilde{M}_0 S_k$ à l'aide de l'algorithme \mathcal{A}_b^1 (voir tableau ci-dessous) en $nk(4k + 2) + (k/2)(5k + 1)$ opérations.

La troisième étape coûte n multiplications. L'étape 4 se fait en décomposant le vecteur d_2^x en deux vecteurs $d_{2,1}^x$ et $d_{2,2}^x$ de taille k avec

$$\begin{cases} d_{2,1}^x = S_k^T M_0 d_1^x \\ d_{2,2}^x = Y_k^T d_1^x. \end{cases}$$

Cela nécessite $k(n + 1) + nk = 2nk + k$ opérations. Pour la cinquième étape de l'algorithme, on résout le système $E_k d_3^x = d_2^x$. On obtient alors l'équation suivante en décomposant le vecteur d_3^x en deux vecteurs $d_{3,1}^x$ et $d_{3,2}^x$ de taille k et en écrivant le complément de Schur de E_k :

$$\begin{bmatrix} I & -\tilde{L}_k \tilde{D}_k^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} S^T \tilde{M}_0 S + \tilde{L}_k \tilde{D}_k^{-1} \tilde{L}_k^T & 0 \\ 0 & -\tilde{D}_k \end{bmatrix} \begin{bmatrix} I & 0 \\ -\tilde{D}_k^{-1} \tilde{L}_k^T & I \end{bmatrix} \begin{pmatrix} d_{3,1}^x \\ d_{3,2}^x \end{pmatrix} = \begin{pmatrix} d_{2,1}^x \\ d_{2,2}^x \end{pmatrix}$$

La résolution de ce système nécessite l'inversion de deux matrices D_k et $S^T \tilde{M}_0 S + \tilde{L}_k \tilde{D}_k^{-1} \tilde{L}_k^T$ de dimension $k \times k$. La proposition 2.3.1 nous assure que ces deux matrices sont non singulières. Au lieu de calculer ces inverses, on calcule les facteurs de Cholesky de ces matrices. La multiplication de l'inverse par une autre matrice revient alors à résoudre deux systèmes triangulaires. Cette étape coûte finalement $O(k^3)$ opérations et se réalise à l'aide des routines de LAPACK ([1]). L'étape 6 se fait en décomposant le vecteur d_4^x en deux vecteurs $d_{4,1}^x$ et $d_{4,2}^x$ de taille n avec

$$\begin{cases} d_{4,1}^x = M_0 S_k d_3^x \\ d_{4,2}^x = Y_k d_3^x. \end{cases}$$

Elle coûte $n(k + 1) + nk = n(2k + 1)$ opérations. La dernière étape coûte finalement n divisions.

Le coût total du calcul de d^x dans le cas où ne sont présentes que des contraintes de bornes est alors de $n(4k^2 + 7k + 4) + m + O(k^3) + 5/2k^2 + 3/2k$ opérations. \square

ALGORITHME \mathcal{A}_b^1 pour le calcul de \tilde{D}_k , de \tilde{L}_k et de $S_k^\top \tilde{M}_0 S_k$.

Entrée : $N, S_k, Y_k, L_k, D_k, \sigma_k$.

Pour $i = 1, k$

 Pour $j = i, k$

$v_1 = v_2 = v_3 = 0$

 Pour $l = 1, n$

$v = \frac{(y_{j-1})_l}{\sigma_k + N_{l,l}}$

$v_1 = v_1 + (s_{i-1})_l v$

$v_2 = v_2 + (y_{i-1})_l v$

$v_3 = v_3 + ((s_{i-1})_l (s_{j-1})_l \frac{N_{ll}}{\sigma_k + N_{ll}})$

 fin (pour)

$(\tilde{D}_k)_{i,j} = v_2$

$(\tilde{L}_k)_{i,j} = -\sigma_k v_1$

$(S_k^\top \tilde{M}_0 S_k)_{i,j} = \sigma_k v_3$

$(\tilde{D}_k)_{i,j} = (D_k)_{ij} + (\tilde{D}_k)_{i,j}$

$(S_k^\top \tilde{M}_0 S_k)_{j,i} = (S_k^\top \tilde{M}_0 S_k)_{i,j}$

 fin (pour)

 Si $i > j$

 Pour $j = 1, (i-1)$

$v_1 = 0$

 Pour $l = 1, n$

$v = \frac{(y_{j-1})_l}{\sigma_k + N_{l,l}}$

$v_1 = v_1 + (s_{i-1})_l v$

 fin (pour)

$(\tilde{L}_k)_{i,j} = (L_k)_{ij} - \sigma_k v_1$

 fin (pour)

 fin (pour)

 fin (si)

Sortie : $\tilde{D}_k, \tilde{L}_k, S_k^\top \tilde{M}_0 S_k$.

2.3.3 Problèmes avec des contraintes générales

On considère dans cette partie le problème suivant

$$(\mathcal{P}) \begin{cases} \min f(x) \\ c_g(x) \leq 0, \\ l \leq x \leq u, \end{cases} \quad (2.20)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $c_g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ sont deux fonctions continûment différentiables dans \mathbb{R}^n et où $-\infty \leq l < u \leq +\infty$ sont des vecteurs de \mathbb{R}^n représentant les bornes sur les variables. On note m^l le nombre de contraintes de bornes inférieures, m^u , celui des contraintes de bornes supérieures, $m = m^l + m^u$ le nombre total de contraintes de bornes et c le vecteur de \mathbb{R}^{m+p} des contraintes

$$c(x) = \begin{pmatrix} c_b(x) \\ c_g(x) \end{pmatrix} \quad \text{avec} \quad c_b(x) = \begin{pmatrix} l - x \\ x - u \end{pmatrix}. \quad (2.21)$$

La jacobienne des contraintes peut donc s'écrire

$$\nabla c(x) = (\nabla c_b(x) | \nabla c_g(x))$$

où $\nabla c_b(x)$ et $\nabla c_g(x)$ sont respectivement la jacobienne des contraintes de bornes et celle des contraintes d'inégalité générale. Dans le cas de problèmes avec des contraintes générales, la matrice N définie par (2.5) n'est plus, à priori, une matrice diagonale. Le calcul de $(M_0 + N)^{-1}$ demande alors plus d'attention. Séparons les multiplicateurs associés aux contraintes de bornes notés λ_b et ceux associés aux contraintes d'inégalité λ_g . Notons maintenant Λ_g , Λ_b , C_g et C_b les matrices diagonales ayant respectivement sur leur diagonale les composantes des vecteurs λ_g , λ_b , c_g et c_b . La matrice N définie par (2.5) peut alors s'écrire

$$N = \nabla c_b(x) \Lambda_b (-C_b)^{-1} \nabla c_b(x)^\top + \nabla c_g(x) \Lambda_g (-C_g)^{-1} \nabla c_g(x)^\top. \quad (2.22)$$

Avec les mêmes arguments que dans le cas où ne sont présentes que des contraintes de bornes, on montre que la matrice $\nabla c_b(x)^\top \Lambda_b (-C_b)^{-1} \nabla c_b(x)$ est une matrice diagonale définie positive. étant la somme d'une matrice définie positive et d'une matrice semi-définie positive, la matrice N est une matrice définie positive.

Avec la proposition 2.3.1, les égalités (1.29) et (2.22), on obtient la formule suivante pour le calcul de d^x

$$d^x = [(M_0 + N)^{-1} + (M_0 + N)^{-1} U_k E_k^{-1} U_k^\top (M_0 + N)^{-1}] b.$$

Proposition 2.3.3 *Soit M_0 une matrice symétrique définie positive. Soit N la matrice symétrique semi-définie positive définie par (2.5). Supposons que les k paires de vecteurs $\{s_i, y_i\}_{i=0}^{k-1}$ satisfassent $s_i^\top y_i > 0$ pour tout i dans $\{0, \dots, k-1\}$. Soit M_k la matrice obtenue après k mises à jour de la matrice M_0 avec la formule de BFGS (2.2) en utilisant ces vecteurs. Le coût du calcul de d^x est alors de $n(2p^2 + (7k + 17)k/2 + 3pk + 4p + 6) + m + p(p(k + 2) + 3k(k + 1)/2 + 8) + O(p^3) + (3k^2 + k)/2 + O(k^3)$ opérations.*

Preuve. La matrice $(M_0 + N)$ n'étant plus diagonale, son inverse est plus coûteux à calculer. On ne va pas calculer explicitement son inverse mais donner une factorisation avantageuse de son inverse sans avoir à le stocker. Notons D_0 la matrice diagonale de taille $n \times n$ définie par

$$D_0 = M_0 + \nabla c_b(x) \Lambda_b (-C_b)^{-1} \nabla c_b(x)^\top.$$

En utilisant la formule de Sherman-Morrison-Woodbury [52] pour calculer l'inverse de $(M_0 + N)$ et avec l'égalité (2.22), on obtient la formule suivante

$$(M_0 + N)^{-1} = D_0^{-1} - D_0^{-1} \nabla c_g(x)^\top (\Lambda_g^{-1} (-C_g) + \nabla c_g(x) D_0^{-1} \nabla c_g(x)^\top)^{-1} \nabla c_g(x) D_0^{-1}. \quad (2.23)$$

On ne calcule pas explicitement cet inverse dans l'algorithme, mais on calcule la matrice diagonale D_0^{-1} et les facteurs de Cholesky L_g de la matrice de taille $p \times p$ $(\Lambda_g^{-1} (-C_g) + \nabla c_g(x)^\top D_0^{-1} \nabla c_g(x))$. Ainsi, si le nombre p de contraintes c_g

d'inégalité générale est peu important, ces facteurs sont rapidement calculés et nécessitent peu d'espace mémoire.

L'algorithme pour le calcul de d^x est similaire à celui dans le cas de problèmes avec des contraintes de bornes. Seule l'étape 0 est différente. On suppose disponibles au début du calcul les matrices $S_{k-1}, Y_{k-1}, D_{k-1}, L_{k-1}, \nabla c_g, \Lambda_g, C_g, \nabla c_b, \Lambda_b, C_b$, les vecteurs s_k et y_k vérifiant $s_k^T y_k > 0$ et le vecteur b défini par (2.5).

ALGORITHME \mathcal{A}_g pour calculer d^x avec (2.19)

0. Calcul de D_0^{-1}, L_g .
 1. Mise à jour de S_k, Y_k, D_k, L_k .
 2. Calcul de \tilde{D}_k, \tilde{L}_k et $S_k^T \tilde{M}_0 S_k$.
 3. Calcul de $d_1^x = (M_0 + N)^{-1} b$.
 4. Calcul de $d_2^x = U_k^T d_1^x$.
 5. Calcul de $d_3^x = E_k^{-1} d_2^x$.
 6. Calcul de $d_4^x = U_k d_3^x$.
 7. Calcul de $d^x = d_1^x + (M_0 + N)^{-1} d_4^x$.
-

L'étape 0 nécessite $(n + m)$ opérations pour le calcul de D_0^{-1} . Le calcul de $\Lambda_g^{-1} C_g$ nécessite p opérations et celui de $\nabla c_g(x) D_0^{-1} \nabla c_g(x)^T$, $2np^2$ opérations. En ajoutant le coût de la factorisation de Cholesky, on a un coût total de $(n + m) + p + 2np^2 + O(p^3)$ opérations. Le coût des mises à jour à l'étape 1 est de kn opérations. Le calcul de \tilde{D}_k, \tilde{L}_k et $S_k^T \tilde{M}_0 S_k$ de l'étape 2 se fait à l'aide des deux algorithmes suivants. Soit X une matrice de dimension $n \times k$. En utilisant l'égalité (2.23), le produit $X^T (M_0 + N)^{-1} X$ s'écrit

$$X^T (M_0 + N)^{-1} X = X^T D_0^{-1} X - X^T D_0^{-1} A_g^T (L_g L_g^T)^{-1} \nabla c_g(x) D_0^{-1} X.$$

Notons X_1 et X_2 les matrices de taille $p \times n$ définies par $X_1 = \nabla c_g(x)^T D_0^{-1} X$ et $X_2 = L_g^{-1} X_1$ (c'est à dire $L_g X_2 = X_1$). L'égalité précédente devient

$$X^T (M_0 + N)^{-1} X = X^T D_0^{-1} X - X_2^T X_2 \quad (2.24)$$

On utilise cette formule pour calculer rapidement les matrices suivantes

$$\begin{cases} S_k^T \tilde{M}_0 S_k &= S_k^T (M_0 - M_0 (M_0 + N)^{-1} M_0) S_k, \\ \tilde{L}_k &= L_k - S_k^T M_0 (M_0 + N)^{-1} Y_k, \\ \tilde{D}_k &= D_k + Y_k^T (M_0 + N)^{-1} Y_k. \end{cases}$$

Notons X_1^i et X_2^i (pour $i = 1, 2$) les matrices de taille $p \times n$ définies par

$$\begin{aligned} X_1^1 &= \nabla c_g(x)^T D_0^{-1} Y_k \\ X_1^2 &= \nabla c_g(x)^T D_0^{-1} M_0 S_k, \\ X_2^1 &= L_g^{-1} X_1^1, \\ X_2^2 &= L_g^{-1} X_1^2. \end{aligned}$$

On obtient alors

$$\begin{cases} S_k^T \tilde{M}_0 S_k &= S_k^T M_0 S_k - S_k^T M_0 D_0^{-1} M_0 S_k + (X_2^2)^T X_2^2, \\ \tilde{L}_k &= L_k - S_k^T M_0 D_0^{-1} Y_k + (X_2^2)^T X_2^1, \\ \tilde{D}_k &= D_k + Y_k^T D_0^{-1} Y_k - (X_2^1)^T X_2^1. \end{cases}$$

On va utiliser ces factorisations pour les calculs de l'étape 2. On a l'algorithme \mathcal{A}_g^1 (voir tableau ci-dessous) pour le calcul de X_1^1 et X_1^2 . Il nécessite $(3n + 1)pk$ opérations.

ALGORITHME \mathcal{A}_g^1 Calcul X_1^1 et X_1^2 nécessaires au calcul de \tilde{D}_k , \tilde{L}_k et $S_k^T \tilde{M}_0 S_k$.

```

Pour i = 1, p
  Pour j = 1, k
     $(X_1^1)_{ij} = (X_1^2)_{ij} = 0$ 
    Pour h = 1, n
       $(\nabla c_g(x))_{ih} = (\nabla c_g(x))_{ih} (D_0^{-1})_{hh}$ 
       $(X_1^1)_{ij} = (X_1^1)_{ij} + (\nabla c_g(x))_{ih} Y_{hj}$ 
       $(X_1^2)_{ij} = (X_1^2)_{ij} + (\nabla c_g(x))_{ih} S_{hj}$ 
    fin (pour)
     $(X_1^2)_{ij} = \sigma_k (X_1^2)_{ij}$ 
  fin (pour)
fin (pour)

```

On calcule X_2^1 et X_2^2 en résolvant les deux systèmes triangulaires suivants

$$L_g X_2^i = X_1^i \text{ (pour } i = 1, 2).$$

On utilise l'algorithme \mathcal{A}_g^2 suivant (tableau ci-dessous), avec $l = 1, 2$, pour résoudre ces systèmes.

ALGORITHME \mathcal{A}_g^2 calculant X_2^1 et X_2^2 nécessaires au calcul de \tilde{D}_k , \tilde{L}_k et $S_k^T \tilde{M}_0 S_k$.

```

Pour j = 1, k
  Pour i=1,p
     $(X_2^l)_{ij} = ((X_1^l)_{ij} - \sum_{h=1}^{i-1} (L_g)_{ih} (X_2^l)_{hj}) / (L_g)_{ii}$ 
  fin (pour)
fin (pour)

```

Le décompte des opérations dans l'algorithme \mathcal{A}_g^2 est de

$$k \left(\sum_{i=1}^{p+1} i - 2 \right) = k \frac{p^2 + 3p - 2}{2} \text{ opérations.}$$

Les calculs de $\{X_2^i, i = 1, 2\}$ coûtent donc $k(p^2 + 3p - 2)$ opérations. On récupère finalement les matrices de l'étape 2 avec l'algorithme A_g^3 (voir tableau ci-dessous), dont le coût est de $(7n + 3p + 3)(k(k + 1)/2)$ opérations.

ALGORITHME A_g^3 pour le calcul de \tilde{D}_k, \tilde{L}_k et $S_k^\top \tilde{M}_0 S_k$ dans le cas de problèmes avec des contraintes générales.

Pour $i = 1, k$
 Pour $j = i, k$
 $v_1 = v_2 = v_{\tilde{M}_0} = 0$
 $v_{\tilde{D}_k} = v_{\tilde{L}_k} = 0$
 $(S_k^\top \tilde{M}_0 S_k)_{ij} = 0$
 Pour $h = 1, n$
 $v_1 = \sigma_k (S_k)_{hi}$
 $v_2 = (D_0^{-1})_{hh} (Y_k)_{hj}$
 $v_{\tilde{M}_0} = v_{\tilde{M}_0} + v_1 (D_0^{-1})_{hh} (S_k)_{hj}$
 $v_{\tilde{D}_k} = v_{\tilde{D}_k} + (Y_k)_{hi} v_2$
 $v_{\tilde{L}_k} = v_{\tilde{L}_k} + v_1 v_2$
 $(S_k^\top \tilde{M}_0 S_k)_{ij} = (S_k^\top \tilde{M}_0 S_k)_{ij} + v_1 (S_k)_{hj}$
 fin (pour)
 $(S_k^\top \tilde{M}_0 S_k)_{ij} = (S_k^\top \tilde{M}_0 S_k)_{ij} - v_{\tilde{M}_0}$
 $(\tilde{D}_k)_{ij} = (D_k)_{ij} + v_{\tilde{D}_k}$
 $(\tilde{L}_k)_{ij} = -v_{\tilde{L}_k}$
 $v_{\tilde{D}_k} = 0$
 Pour $h = 1, p$
 $(\tilde{L}_k)_{ij} = (\tilde{L}_k)_{ij} + (X_2^2)_{hi} (X_2^1)_{hj}$
 $(S_k^\top \tilde{M}_0 S_k)_{ij} = (S_k^\top \tilde{M}_0 S_k)_{ij} + (X_2^2)_{hi} (X_2^2)_{hj}$
 $v_{\tilde{D}_k} = v_{\tilde{D}_k} + (X_2^1)_{hi} (X_2^1)_{hj}$
 fin (pour)
 $(\tilde{D}_k)_{ij} = (\tilde{D}_k)_{ij} - v_{\tilde{D}_k}$
 $(S_k^\top \tilde{M}_0 S_k)_{ji} = (S_k^\top \tilde{M}_0 S_k)_{ij}$
 $(\tilde{D}_k)_{ji} = (\tilde{D}_k)_{ij}$
 $(\tilde{L}_k)_{ji} = (L_k)_{ji} + (\tilde{L}_k)_{ij}$
 fin (pour)
 fin (pour)

Finalement, le coût total du calcul de \tilde{D}_k, \tilde{L}_k et $S_k^\top \tilde{M}_0 S_k$ est de $n(3pk + (7k/2)(k + 1)) + pk(p + 3k(k + 1)/2 + 4) + (3k^2 - k)/2$ opérations. La troisième étape coûte $2n(p + 1) + (p^2 + 3p - 2)$ opérations en conservant le produit $\nabla c_g(\tilde{x})_g^\top = \nabla c_g(\tilde{x})^\top D_0^{-1}$ (calculé dans l'algorithme A_g^1) et en utilisant la factorisation (2.23). Le coût en opérations des étapes 4, 5, 6 est identique au cas où ne sont présentes que des contraintes de bornes. La dernière étape coûte $2n(p + 1) + (p^2 + 3p - 2)$ opérations.

Finalement, le coût total du calcul de d^x est de $n(2p^2 + (7k + 17)k/2 + 3pk + 4p + 6) + m + p(p(k + 2) + 3k(k + 1)/2 + 8) + O(p^3) + (3k^2 + k)/2 + O(k^3)$

opérations. □

2.4 Algorithme et convergence

2.4.1 Algorithme

On peut maintenant établir une itération de l'algorithme résolvant le système des conditions d'optimalité perturbées (1.2) en utilisant les formules de BFGS à mémoire limitée. Les constantes $\omega \in (0, \frac{1}{2})$ (test d'Armijo), $0 < \beta \leq \beta' < 1$ (coefficients de réduction pour le rebroussement) sont données indépendamment de l'indice de l'itération. Au début de l'itération, on suppose disponibles l'itéré courant $(x, \lambda) \in \mathcal{Z} = \{(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m : (-c(x), \lambda) > 0\}$.

ALGORITHME \mathcal{A}_μ^l pour résoudre (1.2)

(une itération, de (x, λ) vers (x_+, λ_+))

1. Calcul de $d = (d^x, d^\lambda)$, unique solution du système linéaire (1.5).
Si $p > 0$, on utilise l'algorithme \mathcal{A}_g sinon on utilise l'algorithme \mathcal{A}_b . Si $d = 0$, stop ((x, λ) résout le système (1.2)).

2. Calcul d'un pas $\alpha > 0$ par rebroussement :

- 2.0. Soit $\alpha = 1$.

- 2.1. Tant que $(x + \alpha d^x, \lambda + \alpha d^\lambda) \notin \mathcal{Z}$, choisir un nouveau pas α dans $[\beta\alpha, \beta'\alpha]$.

- 2.2. Tant que la condition d'Armijo

$$\psi_\mu(x + \alpha d^x, \lambda + \alpha d^\lambda) \leq \psi_\mu(x, \lambda) + \omega \alpha \nabla \psi_\mu(x, \lambda)^\top d$$

n'est pas satisfaite, choisir un nouveau pas α dans $[\beta\alpha, \beta'\alpha]$.

- 2.3. Soit $(x_+, \lambda_+) = (x, \lambda) + \alpha(d^x, d^\lambda)$.

3. Calcul de (s, y) et de σ_+ avec

$$\begin{aligned} s &= x_+ - x \\ y &= \nabla \ell(x_+, \lambda_+) - \nabla \ell(x, \lambda_+) \\ \sigma_+ &= \frac{y^\top s}{s^\top s}. \end{aligned}$$

La recherche linéaire de l'étape 2 de l'algorithme \mathcal{A}_μ^l est identique à celle de l'algorithme \mathcal{A}_μ présentée dans le chapitre 1. La seule différence entre ces deux algorithmes est la formule de mise à jour de la matrice M approchant le hessien du lagrangien. L'algorithme global contrôlant les itérations externes et la décroissance du paramètre de perturbation μ est le même qu'au chapitre 1. Au début de la $j^{\text{ième}}$ itération externe, une approximation $z_1^j := (x_1^j, \lambda_1^j) \in \mathcal{Z}$ de la solution $\hat{z} := (\hat{x}, \hat{\lambda})$ de (1.1) est supposée disponible. On suppose connues

les valeurs du paramètre de perturbation $\mu^j > 0$ et celles des seuils de précision $\epsilon^j := (\epsilon_\ell^j, \epsilon_c^j) > 0$.

ALGORITHME \mathcal{A} pour résoudre le problème (\mathcal{P})
(une itération externe)

1. A partir du point initial z_1^j , utiliser l'algorithme \mathcal{A}_μ jusqu'à l'obtention d'un point $z^j := (x^j, \lambda^j)$ satisfaisant

$$\begin{cases} \|\nabla f(x^j) + \nabla c(x^j)\lambda^j\| \leq \epsilon_\ell^j, \\ \|C(x^j)\lambda^j - \mu^j e\| \leq \epsilon_c^j. \end{cases} \quad (2.25)$$

2. Définir un nouveau paramètre barrière $\mu^{j+1} > 0$, de nouveaux seuils de précision $\epsilon^{j+1} := (\epsilon_\ell^{j+1}, \epsilon_c^{j+1}) > 0$ tels que $\{\mu^j\}$ et $\{\epsilon^j\}$ convergent vers zéro quand $j \rightarrow \infty$. Choisir un nouvel itéré de départ $z_1^{j+1} \in \mathcal{Z}$ pour l'itération externe suivante.
-

2.4.2 Analyse de convergence

On va montrer dans cette partie que l'algorithme \mathcal{A}_μ^l converge r-linéairement. Ce résultat repose sur l'hypothèse suivante

Hypothèse 2.4.1 *Les fonctions f et $-c_i$ ($1 \leq i \leq m$) sont convexes différentiables de \mathbb{R}^n dans \mathbb{R} ; il existe $\check{\lambda} \in \mathbb{R}^m$, tel que le lagrangien $\ell(\cdot, \check{\lambda})$ soit fortement convexe de module $\check{\kappa} > 0$.*

Il est montré, sous cette hypothèse, voir [4, théorème 3.4], que si M_k est mise à jour en utilisant les formules de BFGS, l'algorithme \mathcal{A}_μ génère une suite d'itérés qui convergent r-linéairement vers la solution de (1.2) (le résultat est prouvé avec une hypothèse plus forte mais il est encore valide avec l'hypothèse 2.4.1, voir [3, théorème 4.10]). Ce résultat repose sur une propriété des formules de BFGS ([12, théorème 2.1]) :

Lemme 2.4.2 *Soit $\{M_k\}$ les matrices obtenues par les formules de BFGS avec les paires de vecteurs $\{(s_k, y_k)\}_{k \geq 1}$, satisfaisant pour tout $k \geq 1$,*

$$s_k^\top y_k \geq \alpha_1 \|s_k\|^2 \text{ et } s_k^\top y_k \geq \alpha_1 \|y_k\|^2,$$

où $\alpha_1 > 0$ et $\alpha_2 > 0$ ne dépendent pas de k . Alors, il existe des constantes positives b_1 , b_2 , b_3 telles que, pour tout $k \geq 1$,

$$\cos \theta_k = \frac{s_k^\top M_k s_k}{\|M_k s_k\| \|s_k\|} \geq b_1 \quad \text{et} \quad b_2 \leq \frac{\|M_k s_k\|}{\|s_k\|} \leq b_3$$

pour au moins $\lceil \frac{k}{2} \rceil$ indices j dans $\{1, \dots, k\}$.

L'algorithme à mémoire limitée présenté dans ce chapitre et celui proposé dans le chapitre 1 ne diffèrent que par la manière de mettre à jour la matrice M_k . La convergence r -linéaire sera donc montrée si l'algorithme génère une suite d'itérés satisfaisant les inégalités ci-dessus. Ce résultat est démontré dans le cadre d'algorithmes à mémoire limitée pour résoudre des problèmes sans contraintes ([38, Théorème 7.1]). On va généraliser ce résultat avec le lemme suivant.

On dit que la matrice M est générée par les formules de BFGS à mémoire limitée à partir d'une matrice initiale $M^{(0)}$ avec les q paires de vecteurs $\{(s_i, y_i)\}_{i=0}^{q-1}$, si M est calculée récursivement comme suit :

Pour $i = 0, \dots, q - 1$

$$M^{(i+1)} = M^{(i)} - \frac{M^{(i)} s_i s_i^T M^{(i)}}{s_i^T M^{(i)} s_i} + \frac{y_i y_i^T}{y_i^T s_i}, \quad (2.26)$$

fin (pour)
 $M = M^{(q)}$.

Les deux lemmes suivants nous permettent d'obtenir des bornes pour le déterminant de la matrice M mise à jour avec les formules de BFGS à mémoire limitée.

Lemme 2.4.3 *Soient u et v deux vecteurs arbitraires de \mathbb{R}^n . La matrice $(I + uv^T)$ a $n - 1$ valeurs propres égales à 1 et une valeur propre égale à $(1 + v^T u)$.*

Preuve. Calculons le produit de la matrice $(I + uv^T)$ par un vecteur x de \mathbb{R}^n :

$$(I + uv^T)x = x + (v^T x)u.$$

Si $x \in \{y \in \mathbb{R}^n : v^T y = 0\}$, alors $(I + uv^T)x = x$ et la matrice $(I + uv^T)$ a $n - 1$ valeurs propres égales à 1. Si $x \in \{y \in \mathbb{R}^n : y = \alpha u, \alpha \in \mathbb{R}\}$, alors $(I + uv^T)x = (1 + v^T u)x$ et la matrice $(I + uv^T)$ a 1 valeur propre égale à $(1 + v^T u)$. D'après la définition du déterminant d'une matrice, on obtient le résultat du lemme. \square

On va maintenant appliquer ce lemme pour calculer le déterminant d'une matrice M mise à jour avec les formules de BFGS à mémoire limitée (voir [53])

Lemme 2.4.4 *Soit M la matrice mise à jour par les formules de BFGS à mémoire limitée à partir de la matrice initiale $M^{(0)}$ avec les q paires de vecteurs $\{(s_i, y_i)\}_{i=0}^{q-1}$. Alors*

$$\det(M) = \det(M^{(0)}) \prod_{i=0}^{q-1} \frac{y_i^T s_i}{s_i^T M^{(i)} s_i}. \quad (2.27)$$

Preuve. Notons tout d'abord la matrice M_+ mise à jour par la formule de BFGS à mémoire limitée (2.26) (une étape i). La matrice M_+ peut s'écrire

$$M_+ = \left(M + \frac{yy^\top}{y^\top s} \right) - \frac{Mss^\top M}{s^\top Ms}.$$

Notons $M' = M + \frac{yy^\top}{y^\top s}$. La matrice M_+ peut alors s'écrire

$$M_+ = (M')^{1/2} \left(I - \frac{(M')^{-1/2} Mss^\top M (M')^{-1/2}}{s^\top Ms} \right) (M')^{1/2}.$$

Donc, en rappelant le fait que le déterminant d'un produit de deux matrices est le produit des déterminants de ces matrices et en appliquant le lemme précédent, on obtient :

$$\begin{aligned} \det(M_+) &= \det(M') \det\left(I - \frac{(M')^{-1/2} Mss^\top M (M')^{-1/2}}{s^\top Ms}\right) \\ &= \det(M') \left(1 - \frac{s^\top M (M')^{-1} Ms}{s^\top Ms}\right). \end{aligned} \quad (2.28)$$

Calculons tout d'abord le terme $\det(M')$. Remarquons que la matrice M' peut se factoriser comme suit :

$$M' = M^{1/2} \left(I + \frac{(M)^{-1/2} yy^\top M^{-1/2}}{y^\top s} \right) M^{1/2}.$$

En appliquant le lemme précédent, on obtient une formule pour le déterminant de M' :

$$\begin{aligned} \det(M') &= \det(M) \det\left(I + \frac{(M)^{-1/2} yy^\top M^{-1/2}}{y^\top s}\right) \\ &= \det(M) \left(1 + \frac{y^\top M^{-1} y}{y^\top s}\right) \\ &= \det(M) \left(\frac{y^\top s + y^\top M^{-1} y}{y^\top s}\right). \end{aligned} \quad (2.29)$$

Calculons maintenant le second terme du produit du membre de droite dans (2.28). Remarquons que l'inverse de la matrice M' peut s'écrire, en utilisant la formule de Sherman-Morrison-Woodbury,

$$(M')^{-1} = M^{-1} - \left(\frac{M^{-1} yy^\top M^{-1}}{y^\top s + y^\top M^{-1} y} \right).$$

Notons D le déterminant à calculer :

$$D = \left(1 - \frac{s^\top M (M')^{-1} Ms}{s^\top Ms}\right).$$

En utilisant la formule ci-dessus de $(M')^{-1}$ et le lemme précédent, on a

$$\begin{aligned} D &= \left(1 - \frac{s^\top M (M')^{-1} M s}{s^\top M s}\right) \\ &= \left(1 - \left[s^\top M \left(M^{-1} - \frac{M^{-1} y y^\top M^{-1}}{y^\top s + y^\top M^{-1} y} \right) M s \right] / (s^\top M s)\right) \\ &= \frac{(y^\top s)^2}{(y^\top s + y^\top M^{-1} y)(s^\top M s)}. \end{aligned} \quad (2.30)$$

Finalement, on a, en considérant les équations (2.28), (2.29) et (2.30),

$$\det(M_+) = \det(M) \frac{y^\top s}{s^\top M s}. \quad (2.31)$$

On a le résultat en appliquant de manière récursive la formule (2.31) à la matrice $M^{(0)}$ avec les vecteurs $\{s_i, y_i\}_{i=0}^{q-1}$. \square

Lemme 2.4.5 Soit $\{M_k^{(0)}\}$ une suite bornée de matrices définies positives avec des inverses bornés. Soit $\{(s_k, y_k)\}_{k \geq 1}$ une suite de paires de vecteurs tels que, pour tout $k \geq 1$

$$y_k^\top s_k \geq \alpha_1 \|s_k\|^2 \quad \text{et} \quad y_k^\top s_k \geq \alpha_2 \|y_k\|^2, \quad (2.32)$$

avec $\alpha_1 > 0$ et $\alpha_2 > 0$ des constantes. Soit q un entier positif et soit $\{M_k\}$ la suite de matrices définies positives telles que chaque matrice M_k est calculée à l'aide des formules de BFGS à mémoire limitée à partir de la matrice initiale $M_k^{(0)}$ avec les paires $\{(s_{ij}, y_{ij})\}_{j=0}^{q_k-1}$ où $1 \leq q_k \leq q$. Il existe alors des constantes positives b_1, b_2 , et b_3 , telles que pour tout indice, $k \geq 1$,

$$b_1 \leq \cos \theta_k \quad \text{and} \quad b_2 \leq \frac{\|M_k s_k\|}{\|s_k\|} \leq b_3. \quad (2.33)$$

Preuve. Avec (2.26), la seconde inégalité de (2.32), $q_k \leq q$ et la bornitude de la suite $\{M_k^{(0)}\}$, la trace de la matrice M_k peut être bornée supérieurement comme suit :

$$\text{tr}(M_k) \leq \text{tr}(M_k^{(0)}) + \sum_{j=0}^{q_k-1} \frac{\|y_{ij}\|^2}{y_{ij}^\top s_{ij}} \leq \text{tr}(M_k^{(0)}) + q \alpha_2^{-1} \leq K_1, \quad (2.34)$$

où K_1 est une constante positive.

La formule du déterminant de M_k est donnée par le lemme 2.4.4 :

$$\det(M_k) = \det(M_k^{(0)}) \prod_{i=0}^{q_k-1} \frac{y_i^\top s_i}{s_i^\top M_k^{(i)} s_i} = \det(M_k^{(0)}) \prod_{j=0}^{q_k-1} \frac{y_j^\top s_j}{s_j^\top s_j} \frac{s_j^\top s_j}{s_j^\top M_k^{(j)} s_j}.$$

En utilisant la première inégalité de (2.32), (2.34) et la bornitude des inverses de $M_k^{(0)}$, on a

$$\det(M_k) \geq \det(M_k^{(0)}) (\alpha_1 K_1^{-1})^{q_k} \geq \det(M_k^{(0)}) \min(1, (\alpha_1 K_1^{-1})^q) \geq K_2, \quad (2.35)$$

pour une constante positive K_2 .

On déduit des inégalités (2.34) et (2.35) que les matrices M_k et leurs inverses M_k^{-1} sont uniformément bornés. On en conclut alors que les inégalités (2.33) sont satisfaites. \square

Théorème 2.4.6 *Supposons l'hypothèse 2.4.1 satisfaite et que f et c sont des fonctions $C^{1,1}$. Alors, l'algorithme \mathcal{A}_μ^l génère une suite d'itérés $\{z_k\}$ convergeant vers \hat{z}_μ r -linéairement, c'est à dire $\limsup_{k \rightarrow \infty} \|z_k - \hat{z}_\mu\|^{1/k} < 1$.*

Preuve. D'après une remarque précédente, il suffit de montrer qu'on a les deux inégalités (2.32). Elles sont une conséquence directe de la forte convexité du lagrangien et des dérivées Lipschitz continues des fonctions f et c (voir [3, lemme 4.5]). La fin de la démonstration est identique à celle du théorème 3.4 dans [4]. \square

2.5 Résultats numériques

On présente dans cette partie l'extension du code NOPTIQ utilisant les formules de BFGS à mémoire limitée pour résoudre des problèmes de grande taille avec des contraintes d'inégalité générale. La mise en œuvre du code suit les algorithmes présentés dans ce chapitre. On précise que la paire de vecteurs (s, y) est conservée en mémoire pour les formules de BFGS à mémoire limitée si on a l'inégalité suivante :

$$y^T s \geq \epsilon \|s\|^2,$$

où ϵ est une petite constante positive.

2.5.1 Le code 1-BFGS-B

Le code 1-BFGS-B ([11]) est une extension du code 1-BFGS ([38]) utilisé en optimisation sans contrainte. Cette extension permet de résoudre des problèmes de grande taille avec des contraintes de bornes. A chaque itération, une approximation M_k du hessien est calculée à l'aide des formules de BFGS à mémoire limitée. Cette approximation est utilisée pour construire un modèle quadratique de la fonction objectif

$$m_k(x) = f(x_k) + \nabla f(x_k)^T (x - x_k) + (x - x_k)^T M_k (x - x_k).$$

Une direction de recherche est alors calculée en deux étapes : la méthode de la projection sur le gradient permet d'identifier les contraintes actives (c'est à dire l'ensemble des variables se trouvant sur les faces du polyèdre défini par les contraintes de bornes) : on minimise m_k sur la projection de la direction de plus grande pente ; on calcule x^c comme suit

$$x^c = x(t_c) \quad \text{avec} \quad t^c = \min_t \{m_k(x(t))\} \quad \text{où} \quad x(t) = P(x_k - t \nabla f(x_k), l, u)$$

où P est l'opérateur de projection défini pour $i = 1, \dots, n$ par

$$P(\mathbf{y}, \mathbf{l}, \mathbf{u})_i = \begin{cases} l_i & \text{si } y_i < l_i \\ y_i & \text{si } l_i \leq y_i \leq u_i \\ u_i & \text{si } u_i < y_i. \end{cases}$$

Le modèle quadratique créé est alors minimisé suivant les variables libres (c'est à dire non actives). On calcule une solution approchée notée \bar{x}_{k+1} du problème suivant

$$\begin{cases} \min m_k(x) \\ l_i \leq x_i \leq u_i, & \text{si } i \notin A(x^c), \\ x_i = x_i^c, & \text{si } i \in A(x^c). \end{cases}$$

où $A(x^c) = \{i : x_i^c < l_i \text{ ou } u_i < x_i^c\}$. La direction de recherche est alors définie comme étant le vecteur joignant l'itéré courant et cette approximation du minimum du modèle quadratique $d_k = \bar{x}_{k+1} - x_k$. Finalement, une recherche linéaire est faite pour obtenir un pas le long de cette direction. Ce pas est calculé de façon à satisfaire la condition de décroissance

$$f(x_{k+1}) \leq f(x_k) + \omega \alpha \nabla f(x_k)^\top d_k$$

et la condition de courbure

$$|\nabla f(x_{k+1})^\top d_k| \leq \beta |f(x_k)^\top d_k|,$$

où ω et β sont des paramètres fixés respectivement à 10^{-4} et à 0.9. 1-BFGS-B considère qu'une solution du problème initial est trouvée si

$$\|P(x_k - \nabla f(x_k), \mathbf{l}, \mathbf{u}) - x_k\|_\infty \leq \epsilon.$$

2.5.2 Comparaisons des différents codes

Les essais numériques du code NOPTIQ ont été réalisés sous l'environnement CUTER (Constrained and Unconstrained Testing Environment revisited, [6] et [16]) sur un PC sous linux ayant un processeur cadencé à 1 GHz et possédant une mémoire de 128 Mo.

Comparaison des formules de BFGS et formules de BFGS à mémoire limitée

On a tout d'abord testé le code NOPTIQ utilisant les formules de BFGS à mémoire limitée sur des problèmes de petite et moyenne taille (voir les tableaux 1.1, 1.2 et 1.3). On a comparé les résultats obtenus avec ceux obtenus avec le code NOPTIQ utilisant les formules de BFGS à mémoire pleine. Une précision de 10^{-8} sur les conditions d'optimalité est demandée. Le graphique 2.3 représente le nombre d'appels au simulateur faits par NOPTIQ pour résoudre les problèmes considérés en utilisant soit les formules de BFGS à mémoire pleine (notées bfgs) soit celles à mémoire limitée (notées lbfgs). On a éliminé les problèmes non résolus par NOPTIQ et ceux dont les solutions calculées par les

deux versions de NOPTIQ sont différentes. Ce graphique montre que les résultats obtenus en utilisant les formules à mémoire limitée sont comparables avec ceux obtenus en utilisant les formules à mémoire pleine. Sur certains problèmes, NOPTIQ fait beaucoup plus d'itérations en utilisant les formules de BFGS à mémoire limitée qu'en utilisant les formules à mémoire pleine. Cela s'explique par l'emploi de la correction de Powell dans les secondes formules et par le fait qu'on fasse du skipping quand on utilise les formules à mémoire limitée. On a testé l'emploi de la correction de Powell dans les formules de BFGS à mémoire limitée afin de conserver à chaque itération la paire de vecteurs calculée mais cela n'améliorait pas les résultats. On remarque aussi, sur ce graphique, que l'emploi des formules à mémoire limitée est parfois meilleur que l'emploi des formules à mémoire pleine sur certains problèmes. Cela s'explique par le fait qu'en utilisant ces formules, on conserve une information sur les données du second ordre qui est locale. Les méthodes de BFGS à mémoire pleine conservent, elles, toutes l'information du second ordre depuis le début de l'algorithme. Si cette information est initialement très éloignée de la véritable information du second ordre du problème, les formules de BFGS à mémoire pleine devront alors la corriger, ce qui peut nécessiter des itérations supplémentaires.

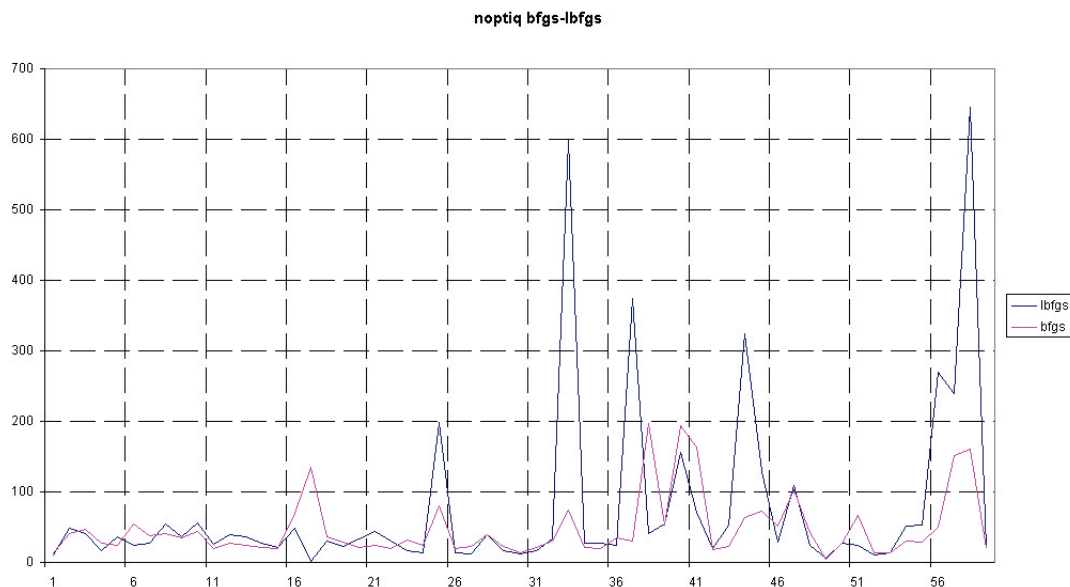


FIG. 2.3 – Nombre d'appels au simulateur de NOPTIQ avec les formules de BFGS à mémoire pleine et à mémoire limitée.

Résultats numériques avec différentes valeurs de m_ℓ

On a réalisé des tests numériques sur des problèmes de grande taille avec uniquement des contraintes de bornes sur les variables dont la liste est donnée dans le tableau 2.1.

nom	n	m	bornes		
			l	u	nfix
BDEXP	5000	5000	5000	0	0
BDEXP	1000	1000	1000	0	0
BQPGABIM	50	92	46	46	4
BQPGASIM	50	100	50	50	0
CVXBQP1	1000	2000	1000	1000	0
EXPLIN2	1200	2400	1200	1200	0
EXPLIN	1200	2400	1200	1200	0
JNLBRNG1	10000	9604	9604	0	396
JNLBRNG2	10000	9604	9604	0	396
JNLBRNGA	10000	9604	9604	0	396
JNLBRNGB	10000	9604	9604	0	396
LINVERSE	1999	1000	1000	0	0
LINVERSE	999	500	500	0	0
MCCORMCK	10000	20000	10000	10000	0
MCCORMCK	5000	10000	5000	5000	0
MCCORMCK	1000	2000	1000	1000	0
NCVXBQP1	1000	2000	1000	1000	0
NCVXBQP2	1000	2000	1000	1000	0
NCVXBQP3	1000	2000	1000	1000	0
NOBNDTOR	5476	10368	5184	5184	292
NONSCOMP	1000	2000	1000	1000	0
OBSTCLAE	5625	10658	5329	5329	296
OBSTCLAL	10000	19208	9604	9604	396
OBSTCLBL	10000	19208	9604	9604	396
OBSTCLBM	10000	19208	9604	9604	396
OBSTCLBU	10000	19208	9604	9604	396
PENTDI	5000	5000	5000	0	0
PROBPENL	500	1000	500	500	0
TORSION1	5476	10368	5184	5184	292
TORSION2	5476	10368	5184	5184	292
TORSION3	5476	10368	5184	5184	292
TORSION4	5476	10368	5184	5184	292
TORSION5	5476	10368	5184	5184	292
TORSION6	5476	10368	5184	5184	292
TORSTONA	5476	10368	5184	5184	292
TORSTIONB	5476	10368	5184	5184	292
TORSTIONC	5476	10368	5184	5184	292
TORSTIOND	5476	10368	5184	5184	292
TORSTONE	5476	10368	5184	5184	292
TORSTIONF	5476	10368	5184	5184	292

TAB. 2.1 – Problèmes testés avec contraintes de bornes de CUTEr.

Les résultats suivants sont obtenus avec le code NOPTIQ en faisant varier le nombre m_ℓ de paires de vecteurs conservées en mémoire : $m_\ell = 3$, $m_\ell = 5$, $m_\ell = 17$ et $m_\ell = 29$. La précision souhaitée sur les tests d'arrêt est de 10^{-8} . Les deux graphiques 2.4 et 2.5 montrent respectivement le nombre d'itérations calculées et le temps écoulé pour résoudre ces problèmes avec le code NOPTIQ avec les différentes valeurs de m_ℓ .

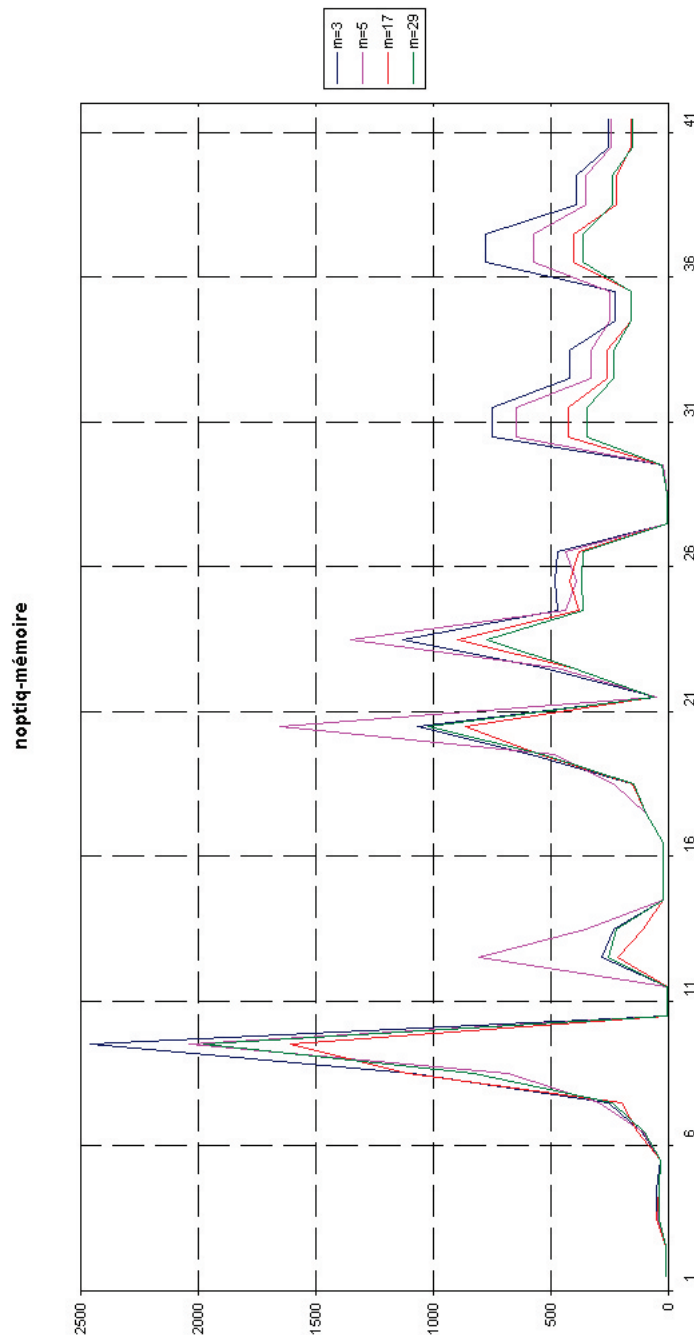


FIG. 2.4 – Nombre d'appels au simulateur de NOPTIQ obtenus en variant le nombre de paire de vecteurs conservées en mémoire.

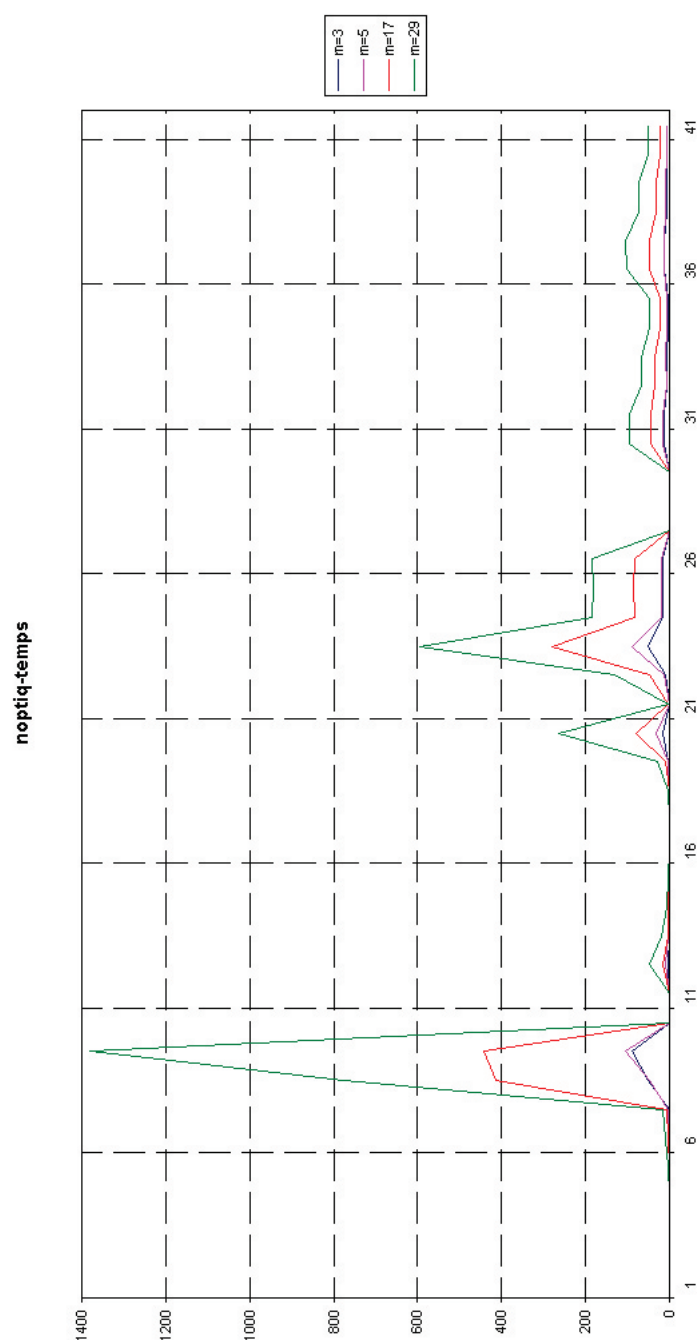


FIG. 2.5 – Temps d'exécution (en seconde) de NOPTIQ obtenus en variant le nombre de paire de vecteurs conservées en mémoire.

On remarque, sur le graphique 2.4, que plus on conserve de vecteurs en mémoire, meilleurs sont les résultats. On remarque aussi sur le graphique 2.5, que plus on conserve de vecteurs en mémoire, plus le temps de résolution augmente. On note toutefois que conserver en mémoire peu de paires de vecteurs est fortement conseillé : le nombre d'itérations calculées est peu élevé et le temps de résolution est petit. Par défaut, NOPTIQ conserve 5 paires de vecteurs

en mémoire.

Tests comparatifs de NOPTIQ et de l-BFGS-B.

On compare dans ce paragraphe les résultats obtenus avec le code NOPTIQ utilisant les formules de BFGS à mémoire limitée et le code l-BFGS-B pour résoudre les problèmes avec contraintes de bornes du tableau 2.1. l-BFGS-B est utilisé avec ses valeurs par défaut et le nombre de paires conservées pour les formules de BFGS à mémoire limitée est de 5 pour les deux codes. La précision souhaitée sur les tests d'arrêt est de 10^{-8} . On compare les deux codes pour la résolution de problèmes de grande taille avec des contraintes de bornes. Notons q_i la valeur suivante

$$q_i = -\log_2 \left(\frac{nf_A}{nf_B} \right),$$

où nf_A est le nombre d'appels cumulés au simulateur réalisés par le code NOPTIQ, où nf_B le nombre d'appels cumulés au simulateur réalisés par le code l-BFGS-B et où i est le $i^{\text{ème}}$ problème testé. Le signe de q_i indique la méthode nécessitant le moins d'appels au simulateur. Si ce signe est positif, le code NOPTIQ est plus performant et s'il est négatif, c'est le code l-BFGS-B qui l'est. Plus la valeur de $|q_i|$ est importante, plus la méthode gagnante est performante par rapport à l'autre. Le graphique 2.6 représente les valeurs de q_i pour les problèmes testés dont les solutions calculées par les deux codes sont identiques. Les problèmes ont été replacés selon l'ordre décroissant de la valeur de $|q_i|$.

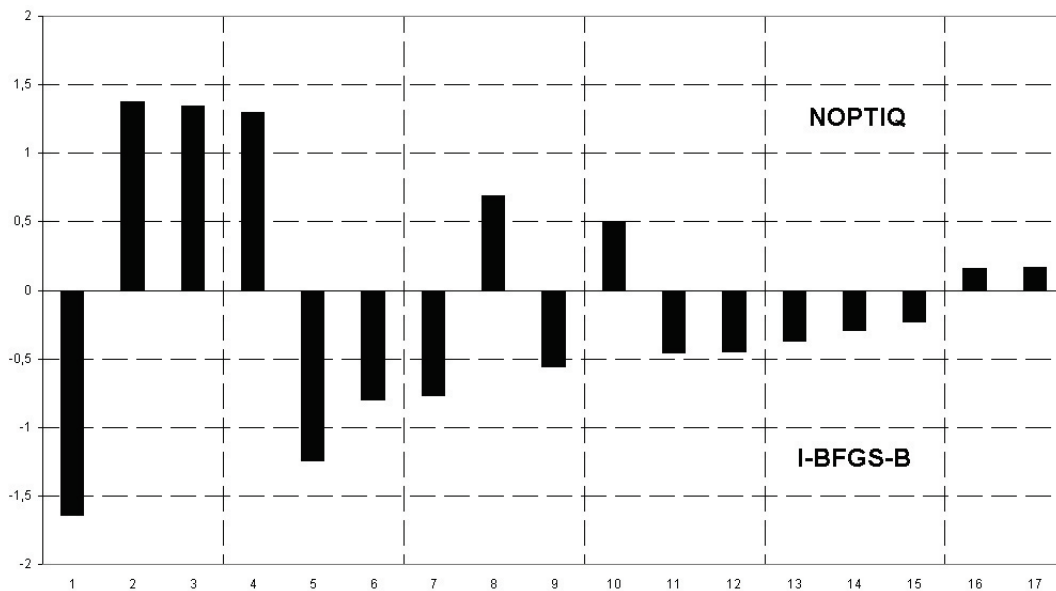


FIG. 2.6 – Résultats numériques de NOPTIQ et de l-BFGS-B.

On remarque que les deux codes sont globalement équivalents. Les codes sont aussi équivalents pour le temps de résolution de ces problèmes. l-BFGS-B

étant un code de référence largement utilisé dans l'industrie, il est encourageant pour notre code NOPTIQ d'obtenir des résultats similaires.

Tests du code NOPTIQ sur des problèmes avec contraintes générales de grande taille.

On a testé le code NOPTIQ utilisant les formules de BFGS à mémoire limitée sur des problèmes avec des contraintes d'inégalité générale listés dans les tableaux 2.2 et 2.3.

nom	n	m	bornes			contraintes
			l	u	nfix	ineg
3PK	30	30	30	0	0	0
ALLINIT	4	3	2	1	1	0
BQP1VAR	1	2	1	1	0	0
BQPGABIM	50	92	46	46	4	0
BQPGASIM	50	100	50	50	0	0
CAMEL6	2	4	2	2	0	0
CHEBYQAD	100	200	100	100	0	0
DECONVB	61	72	61	11	0	0
DIPIGRI	7	4	0	0	0	4
EG1	3	4	2	2	0	0
HART6	6	12	6	6	0	0
HATFLDA	4	4	4	0	0	0
HATFLDB	4	5	4	1	0	0
HATFLDC	25	48	24	24	0	0
HIMMELP1	2	4	2	2	0	0
HS100	7	4	0	0	0	4
HS100MOD	7	4	0	0	0	4
HS110	200	400	200	200	0	0
HS113	10	8	0	0	0	8
HS117	15	20	15	0	0	5
HS12	2	1	0	0	0	1
HS1	2	1	1	0	0	0
HS24	2	5	2	0	0	3
HS25	3	6	3	3	0	0
HS29	3	1	0	0	0	1
HS2	2	1	1	0	0	0
HS35	3	4	3	0	0	1
HS351	3	7	3	3	0	1
HS35MOD	3	3	2	0	1	1

TAB. 2.2 – Problèmes testés de CUTER.

L'initialisation du code NOPTIQ nécessite un point x^0 strictement réalisable. Peu de problèmes de CUTER fournissent un tel point. D'où le fait qu'il y ait peu de problèmes de grande taille dans les tableaux 2.2 et 2.3. Dans le cas où ne sont présentes que des contraintes de bornes, un point strictement réalisable est aisément calculable. Dans le cas où sont présentes des contraintes d'inégalité non linéaire, cela est plus difficile.

La figure 2.7 compare les résultats obtenus par les codes NOPTIQ et LANCELOT pour résoudre ces problèmes. Les paramètres de LANCELOT sont tels que la mise à jour de l'information du second ordre des problèmes est approchée par les formules de BFGS. Bien que ces deux codes ne soient pas conçus pour résoudre le même type de problèmes, on remarque que NOPTIQ donne souvent de meilleurs résultats que LANCELOT. LANCELOT utilise la propriété de séparabilité des fonctions des problèmes pour résoudre des problèmes de grande taille. Il utilise donc la creusité des matrices hessiennes. NOPTIQ lui, est plutôt écrit pour résoudre des problèmes de grande taille donc les matrices hessiennes sont denses.

nom	n	m	bornes			contraintes
			l	u	nfix	ineg
HS36	3	7	3	3	0	1
HS37	3	8	3	3	0	2
HS38	4	8	4	4	0	0
HS3	2	1	1	0	0	0
HS3MOD	2	1	1	0	0	0
HS43	4	3	0	0	0	3
HS44NEW	4	10	4	0	0	6
HS45	5	10	5	5	0	0
HS4	2	2	2	0	0	0
HS57	2	3	2	0	0	1
HS5	2	4	2	2	0	0
HS67	3	20	3	3	0	14
HS70	4	9	4	4	0	1
HS76	4	7	4	0	0	3
HS76I	4	11	4	4	0	3
HS84	5	16	5	5	0	6
HS85	5	48	5	5	0	38
HS93	6	8	6	0	0	2
KSIP	20	1001	0	0	0	1001
LOGROS	2	2	2	0	0	0
MAXLIKA	8	16	8	8	0	0
MDHOLE	2	1	1	0	0	0
MIFFLIN1	3	2	0	0	0	2
MINMAXBD	5	20	0	0	0	20
MOSARQP1	900	930	900	0	0	30
MOSARQP2	900	930	900	0	0	30
OSLBQP	8	11	8	3	0	0
PALMER1A	6	2	2	0	0	0
PALMER1B	4	2	2	0	0	0
PALMER1	4	3	3	0	0	0
PALMER1E	8	1	1	0	0	0
PALMER2A	6	2	2	0	0	0
PALMER2B	4	2	2	0	0	0
PALMER2	4	3	3	0	0	0
PALMER2E	8	1	1	0	0	0
PALMER3A	6	2	2	0	0	0
PALMER3B	4	2	2	0	0	0
PALMER3	4	3	3	0	0	0
PALMER3E	8	1	1	0	0	0
PALMER4A	6	2	2	0	0	0
PALMER4B	4	2	2	0	0	0
PALMER4	4	3	3	0	0	0
PALMER4E	8	1	1	0	0	0
PALMER5A	8	2	2	0	0	0
PALMER5B	9	2	2	0	0	0
PALMER5D	8	4	4	0	0	0
PALMER5E	8	1	1	0	0	0
PALMER6A	6	2	2	0	0	0
PALMER6E	8	1	1	0	0	0
PALMER7A	6	2	2	0	0	0
PALMER7E	8	1	1	0	0	0
PALMER8A	6	2	2	0	0	0
PALMER8E	8	1	1	0	0	0
PROBPENL	500	1000	500	500	0	0
PSPDOC	4	1	0	1	0	0
S368	8	16	8	8	0	0
SIM2BQP	2	2	1	1	1	0
SIMBQP	2	2	1	1	0	0
SIPOW1	2	2000	0	0	0	2000
SIPOW1M	2	2000	0	0	0	2000
SIPOW2	2	2000	0	0	0	2000
SIPOW2M	2	2000	0	0	0	2000
SIPOW3	4	2000	0	0	0	2000
SIPOW4	4	2000	0	0	0	2000
SPECAN	9	18	9	9	0	0
SPIRAL	3	2	0	0	0	2
WEEDS	3	4	3	1	0	0
WOMFLET	3	3	0	0	0	3
YFIT	3	1	1	0	0	0
ZY2	3	6	3	1	0	2

TAB. 2.3 – Problèmes testés de CUTER.

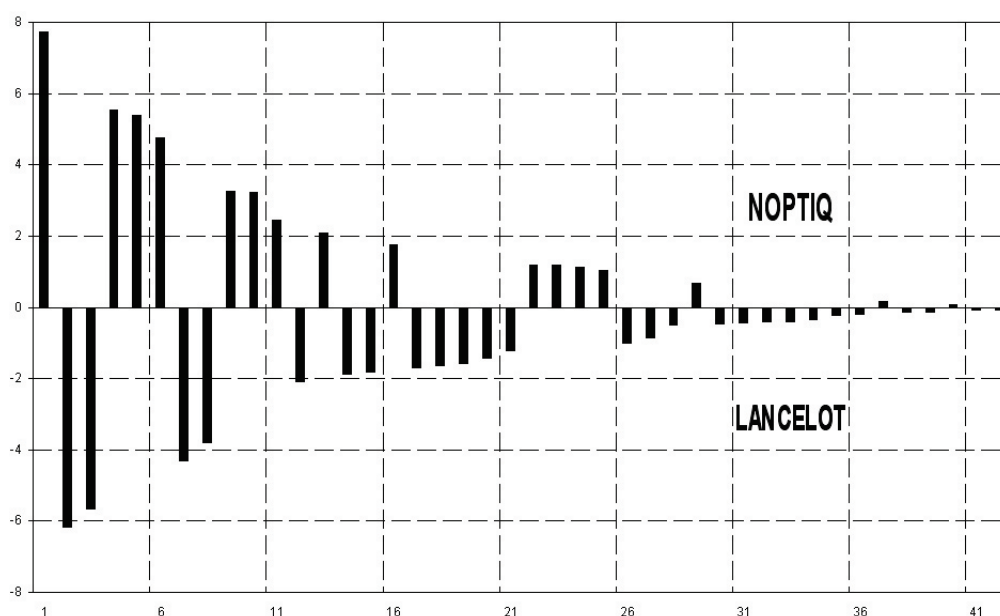


FIG. 2.7 – Résultats numériques de NOPTIQ et de LANCELOT.

2.6 Conclusion

Le code NOPTIQ est performant théoriquement et numériquement pour résoudre des problèmes de grande taille avec contraintes de bornes et plus spécialement pour résoudre des problèmes avec des contraintes d'inégalité plus générale. Les formules de BFGS à mémoire limitée s'adaptent bien aux méthodes de points intérieurs. Les comparaisons du code NOPTIQ avec les codes 1-BFGS-B et LANCELOT qui sont des codes de référence, sont très prometteuses. NOPTIQ est un code réalisable. Cela signifie qu'à l'initialisation, un point x^0 strictement réalisable doit lui être fourni. La dernière partie de mon travail a été de pallier à ce problème. L'objet du chapitre suivant est l'étude de convergence d'une méthode de points intérieurs non réalisable pour résoudre des problèmes avec contraintes d'inégalité générale et d'égalité linéaire.

Chapitre 3

Analyse de convergence d'un algorithme non réalisable de points intérieurs.

3.1 Introduction

On considère le problème de minimiser une fonction convexe sur un ensemble convexe défini par des contraintes d'inégalité et d'égalité linéaire. Le problème est le suivant

$$(\mathcal{P}) \begin{cases} \min f(x), \\ c(x) \leq 0, \\ Ax = b, \end{cases} \quad (3.1)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est la fonction à minimiser, $c(x) \leq 0$ signifie que les fonctions $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ($1 \leq i \leq m$), composantes de c , doivent être non positives à la solution et où A est une matrice de dimension $p \times n$ et b un vecteur de \mathbb{R}^p . Notons a_i , pour i variant de 1 à p , les colonnes de A^T . On suppose $p < n$ et que les fonctions f et c sont convexes. On va présenter dans ce chapitre un algorithme de points intérieurs et de quasi-Newton pour résoudre le problème (\mathcal{P}) à partir d'un point initial non réalisable. Cet algorithme est basé sur la méthode développée dans [4] dont la mise en œuvre est présentée au chapitre 1.

Dans [3], une méthode non réalisable pour résoudre le problème suivant est proposée :

$$\begin{cases} \min f(x), \\ c(x) \leq 0. \end{cases}$$

On ajoute des variables de décalage s aux contraintes pour obtenir un point (x, s) réalisable. Le problème devient alors

$$\begin{cases} \min f(x), \\ s - c(x) \geq 0, \\ s = 0. \end{cases}$$

On pénalise ensuite le problème barrière par un paramètre de pénalité σ . Cette méthode conserve la convexité du problème mais nécessite une hypothèse sur la bornitude du paramètre de pénalité pour obtenir la convergence des itérés. Dans [2], un algorithme pour résoudre le problème précédent à partir d'un point initial non réalisable est présenté. Des variables de décalage s sont ajoutées aux contraintes pour obtenir un point initial (x, s) réalisable. On pénalise le problème ainsi obtenu. On résout alors le problème

$$\begin{cases} \min f(x) + \sigma^\top s, \\ s - c(x) \geq 0, \\ s \geq 0, \end{cases}$$

en utilisant les méthodes de points intérieurs. Dans cette approche, la convergence des itérés est démontrée sans hypothèse sur la bornitude des paramètres de pénalité. On va, dans ce chapitre, étendre la méthode proposée dans [2] à la résolution de problèmes ayant des contraintes d'inégalité et des contraintes d'égalité linéaire. L'extension présentée dans ce chapitre de la méthode proposée dans [2] n'est pas directe car les points qu'on va générer vont satisfaire des contraintes d'égalité linéaire à chaque itération. On va en effet ajouter des variables d'écart s et y . Il est aisé de calculer un vecteur s_1 tel que le couple (x_1, s_1) satisfasse les contraintes $c(x) \leq s$, où x_1 et le point initial fourni par l'utilisateur. Les variables d'écart y vont perturber la contrainte $Ax = b$ en la contrainte $Ax + Dy = b$, où D est une matrice diagonale définie à l'aide du point x_1 . Il sera alors aisé de trouver un point y_1 tel que le couple (x_1, y_1) satisfasse cette contrainte d'égalité.

A condition que les contraintes satisfassent des hypothèses de qualification, les conditions d'optimalité de Karush-Kuhn-Tucker (KKT) du problème (\mathcal{P}) peuvent s'écrire comme suit (voir [50]) : si x est solution de (\mathcal{P}) , il existe un vecteur de multiplicateurs $(\lambda, \xi) \in \mathbb{R}^m \times \mathbb{R}^p$ tel que

$$\begin{cases} \nabla f(x) + \nabla c(x)\lambda - A^\top \xi = 0, \\ c(x) \leq 0, \\ Ax = b, \\ C(x)\lambda = 0, \\ \lambda \geq 0. \end{cases} \quad (3.2)$$

Le lagrangien associé au problème (\mathcal{P}) est défini pour $(x, \lambda, \xi) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$ par

$$\ell(x, \lambda, \xi) = f(x) + \lambda^\top c(x) + \xi^\top (b - Ax).$$

Quand f et c sont deux fois différentiables, le gradient du lagrangien par rapport à x est donné par

$$\nabla_x \ell(x, \lambda, \xi) = \nabla f(x) + \nabla c(x)\lambda - A^\top \xi$$

et son hessien par rapport à x par

$$\nabla_{xx}^2 \ell(x, \lambda, \xi) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 c_i(x).$$

Dans la section suivante, on va faire des rappels. On va ensuite introduire le problème pénalisé avant d'étudier le problème barrière. Des résultats de convergence r-linéaire et q-superlinéaire de l'algorithme résolvant ce problème barrière seront présentés. On présentera finalement l'algorithme global pour résoudre le problème (\mathcal{P}) et on étudiera la convergence des itérés qu'il génère.

3.2 Rappels

Soit f une fonction semi continue inférieurement (sci), propre de \mathbb{R}^n dans $\mathbb{R} \cup \{+\infty\}$. L'épigraphe noté $\text{epi } f$ de la fonction f est l'ensemble convexe fermé défini par

$$\text{epi } f := \{(x, r) \in \mathbb{R}^n \times \mathbb{R} : f(x) \leq r\}.$$

Soit C un ensemble convexe fermé de \mathbb{R}^n . On appelle cône asymptotique de C , noté C^∞ , l'ensemble convexe fermé suivant

$$C^\infty = \{d \in \mathbb{R}^n : x + td \in C, \forall t > 0, \forall x \in C\}.$$

Proposition 3.2.1 [33, Proposition IV.3.2.2] Soit f une fonction convexe fermée de \mathbb{R}^n dans $\mathbb{R} \cup \{+\infty\}$. Le cône asymptotique de $\text{epi } f$ est l'épigraphe de la fonction f^∞ définie par

$$d \mapsto f^\infty(d) = \lim_{t \rightarrow +\infty} \frac{f(x + td) - f(x)}{t} = \sup_{t > 0} \frac{f(x + td) - f(x)}{t}$$

où x est un point arbitraire du domaine de f .

Définition 3.2.2 [33, Définition IV.3.2.3] La fonction f^∞ de la proposition 3.2.1 est appelée dérivée asymptotique ou fonction de récession de f .

La dérivée asymptotique d'une fonction convexe f est une fonction convexe. La dérivée asymptotique d'une fonction f est positivement homogène : $f^\infty(\alpha d) = \alpha f^\infty(d)$, pour tout réel $\alpha > 0$ et $f^\infty(0) = 0$ (son épigraphe est un cône convexe). On montre par un simple calcul que la dérivée asymptotique d'une fonction affine $h : x \rightarrow Ax - b$ dans la direction d est égale à $h^\infty(d) = Ad$.

Lemme 3.2.3 Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction fortement convexe de module κ . Alors $f^\infty(d) = +\infty$ pour toute direction d non nulle.

Preuve. Soit d une direction non nulle de \mathbb{R}^n . Alors,

$$\begin{aligned} f^\infty(d) &= \lim_{t \rightarrow +\infty} \frac{f(x + td) - f(x)}{t} \geq \lim_{t \rightarrow +\infty} (\nabla f(x)^\top d + \frac{\kappa}{2} \|td\|^2) \\ &\geq +\infty. \end{aligned}$$

□

Considérons la fonction logarithmique η de \mathbb{R} dans $\mathbb{R} \cup \{+\infty\}$ suivante

$$\eta(x) = \begin{cases} -\log(x) & \text{si } x > 0 \\ +\infty & \text{si } x \leq 0. \end{cases} \quad (3.3)$$

On calcule les dérivées asymptotiques de cette fonction car nous en aurons besoin plus tard dans le chapitre. La dérivée asymptotique dans une direction d d'une fonction à variable réelle peut être calculée à partir des valeurs $f^\infty(1)$ et $f^\infty(-1)$ suivant le signe de la direction d . Calculons donc les dérivées asymptotiques de la fonction η en 1 et -1 .

$$\begin{aligned}\eta^\infty(1) &= \lim_{t \rightarrow +\infty} \frac{\eta(1+t) - \eta(1)}{t} \\ &= \lim_{t \rightarrow +\infty} \frac{\eta(1+t)}{t} \\ &= 0\end{aligned}$$

et

$$\begin{aligned}\eta^\infty(-1) &= \lim_{t \rightarrow +\infty} \frac{\eta(1-t) - \eta(1)}{t} \\ &= \lim_{t \rightarrow +\infty} \frac{\eta(1-t)}{t} \\ &= +\infty.\end{aligned}$$

On a donc les valeurs suivantes pour la dérivée asymptotique de la fonction η

$$\eta^\infty(d) = \begin{cases} 0 & \text{si } d \geq 0 \\ +\infty & \text{si } d < 0. \end{cases} \quad (3.4)$$

Notons $\mathcal{L}_r(f)$ l'ensemble de niveau de la fonction f au niveau r . On a

$$\mathcal{L}_r(f) = \{x \in \mathbb{R}^n : f(x) \leq r\}.$$

La proposition suivante établit une relation entre les ensembles de niveau d'une fonction f et la dérivée asymptotique de cette fonction.

Proposition 3.2.4 [33, Proposition IV.3.2.5] *Soit f une fonction convexe de \mathbb{R}^n dans $\mathbb{R} \cup \{+\infty\}$. Tous les ensembles de niveau de f ont le même cône asymptotique qui est l'ensemble de niveau de f^∞ au niveau 0 :*

$$\forall r \in \mathbb{R} \text{ tel que } \mathcal{L}_r(f) \neq \emptyset, (\mathcal{L}_r(f))^\infty = \mathcal{L}_0(f^\infty) = \{d \in \mathbb{R}^n : f^\infty(d) \leq 0\}.$$

En particulier, les assertions suivantes sont équivalentes :

- (i) il existe r pour lequel $\mathcal{L}_r(f)$ est non vide et compact ;
- (ii) tous les ensembles de niveau de f sont compacts ;
- (iii) $f^\infty(d) > 0$ pour toute direction d de \mathbb{R}^n non nulle.

Le corollaire ci-dessous établit une caractérisation de la compacité de l'ensemble des solutions du problème (\mathcal{P}) et des dérivées asymptotiques des fonctions intervenant dans le problème.

Corollaire 3.2.5 *Si le problème (\mathcal{P}) est réalisable, alors son ensemble de solution est non vide et compact si et seulement si il n'existe pas de vecteur $d \in \mathbb{R}^n$ non nul tel que $f^\infty(d) \leq 0$, $Ad = 0$ et $c_i^\infty(d) \leq 0$, pour tout $i = 1, \dots, m$.*

Preuve. Notons X^* l'ensemble convexe fermé des solutions du problème (\mathcal{P}). On peut écrire X^* comme l'intersection de plusieurs ensembles :

$$X^* = \mathcal{L}_\alpha(f) \cap \{x : Ax = b\} \cap (\cap_{i=1}^m \mathcal{L}_0(c_i))$$

où α est la valeur optimale de la fonction f dans X^* . Montrer que X^* est un ensemble non vide et compact revient à montrer que $(X^*)^\infty$ est réduit à $\{0\}$ (Proposition 2.2.3, [33]). On sait que l'intersection des cônes asymptotiques d'une famille finie d'ensembles ayant au moins un point en commun est égale au cône asymptotique de l'intersection des ensembles de cette famille (Proposition 2.2.5, [33]). On obtient donc les équivalences suivantes, en utilisant

$$\{x \in \mathbb{R}^n : Ax = b\}^\infty = \{d \in \mathbb{R}^n : Ad = 0\}$$

et la proposition 3.2.4 :

$$\begin{aligned} & X^* \text{ non vide et compact} \\ \Leftrightarrow & (\mathcal{L}_\alpha(f))^\infty \cap \{x : Ax = b\}^\infty \cap (\cap_{i=1}^m (\mathcal{L}_0(c_i))^\infty) = \{0\}. \\ \Leftrightarrow & \mathcal{L}_0(f^\infty) \cap \{x : Ax = b\}^\infty \cap (\cap_{i=1}^m \mathcal{L}_0(c_i^\infty)) = \{0\}. \\ \Leftrightarrow & \mathcal{L}_0(f^\infty) \cap \{d : Ad = 0\} \cap (\cap_{i=1}^m \mathcal{L}_0(c_i^\infty)) = \{0\} \\ \Leftrightarrow & \{d : f^\infty(d) \leq 0\} \cap \{d : Ad = 0\} \cap (\cap_{i=1}^m \{d : c_i^\infty(d) \leq 0\}) = \{0\}. \end{aligned}$$

□

3.3 Hypothèse

Comme dans les chapitres 1 et 2, on fait l'hypothèse suivante sur la convexité du problème (\mathcal{P}). Cette hypothèse est faite dans \mathbb{R}^n tout entier et pas seulement sur la variété affine $\{x \in \mathbb{R}^n : Ax = b\}$ car les itérés qu'on va calculer ne seront pas forcément sur cette variété.

Hypothèse 3.3.1 (H_1) Les fonctions f et c_i ($1 \leq i \leq m$) sont convexes et différentiables de \mathbb{R}^n dans \mathbb{R} ; et il existe $\check{\lambda} \in \mathbb{R}^m$, tel que la fonction $x \rightarrow f(x) + \check{\lambda}^\top c(x)$ soit fortement convexe de module $\check{\kappa} > 0$.

Une première conséquence de l'hypothèse H_1 est la forte convexité du lagrangien pour tout multiplicateur $\lambda > 0$ et pour tout ξ , dans \mathbb{R}^p . Pour $t \in \mathbb{R}$, on définit $t^+ := \max(0, t)$.

Lemme 3.3.2 [3, Lemme 3.2] *Supposons l'hypothèse H_1 satisfaite. Alors, pour tout $\lambda > 0$ et pour tout $\xi \in \mathbb{R}^p$, le lagrangien $\ell(\cdot, \lambda, \xi)$ est fortement convexe de module*

$$\kappa = \check{\kappa} \min(1, \lambda_1/\check{\lambda}_1^+, \dots, \lambda_m/\check{\lambda}_m^+) > 0.$$

Preuve. La preuve est identique à celle dans [3], le fait d'ajouter un terme linéaire au lagrangien ne modifiant en rien sa forte convexité. □

Une seconde conséquence de l'hypothèse H_1 est la compacité de l'ensemble des solutions du problème (\mathcal{P}).

Proposition 3.3.3 [3, Proposition 3.3] *Supposons l'hypothèse H_1 satisfaite. Alors, il n'existe pas de vecteur $d \in \mathbb{R}^n$ non nul tel que $f^\infty(d) < \infty$ et $c_i^\infty(d) < \infty$ pour tout $i \in \{1, \dots, m\}$. En particulier, si le problème (\mathcal{P}) est réalisable, son ensemble de solutions est non vide et compact.*

3.4 Le problème pénalisé

Il n'est pas toujours possible d'avoir initialement un point x satisfaisant les contraintes du problème. Notre motivation est donc de pénaliser le problème (\mathcal{P}) de manière à pouvoir facilement calculer un point initial réalisable pour ce problème pénalisé. étant donné un point initial x_1 quelconque, on modifie la contrainte d'égalité linéaire du problème (\mathcal{P}) afin d'obtenir facilement un point initial réalisable pour cette nouvelle contrainte d'égalité linéaire. On utilise pour cela une méthode dérivée de la méthode dite *big M* employée en programmation linéaire ([17]). étant donné un point x_1 , on calcule la matrice diagonale D dont les éléments sur la diagonale sont les composantes du vecteur $b - Ax_1$. On pénalise alors le problème initial avec un paramètre de pénalisation $\omega \in \mathbb{R}$:

$$(\mathcal{P}^\omega) \begin{cases} \min f(x) + \omega e_p^\top y \\ c(x) \leq 0, \\ Ax + Dy = b, \\ y \geq 0, \end{cases} \quad (3.5)$$

où e_p est le vecteur de \mathbb{R}^p dont toutes les composantes sont égales à 1.

étant donné un point x_1 quelconque, on remarque que le couple $(x_1, y_1) = (x_1, e_p)$ est alors réalisable pour la contrainte d'égalité linéaire. Remarquons que notre méthode permet la variation du paramètre de pénalisation ω au cours des itérations alors que la méthode *big M* fixe initialement le paramètre de pénalisation noté M à une valeur assez grande pour obtenir la convergence des itérés vers la solution du problème initial.

Une remarque importante est la suivante. Pour résoudre le problème (\mathcal{P}^ω) , on applique le pas de Newton à son système des conditions d'optimalité. On va donc linéariser l'ensemble de ces conditions. Comme la condition d'égalité est déjà linéaire et comme l'itéré initial satisfait cette contrainte, l'ensemble des itérés que notre algorithme va générer seront réalisables pour la contrainte $Ax + Dy = b$.

On pénalise maintenant le problème (\mathcal{P}^ω) en ajoutant des variables de décalage afin d'obtenir facilement un point strictement réalisable pour les contraintes d'inégalité en utilisant la méthode étudiée dans [2]. On a alors le problème suivant

$$(\mathcal{P}^{\sigma, \omega}) \begin{cases} \min \Theta_{\sigma, \omega}(x, s, y) := f(x) + \sigma e_m^\top s + \omega e_p^\top y \\ c(x) \leq s, \\ Ax + Dy = b, \\ s \geq 0, \\ y \geq 0, \end{cases} \quad (3.6)$$

où la fonction de pénalité $\Theta_{\sigma,\omega}$ est paramétrée par deux paramètres de pénalité $(\sigma, \omega) \in \mathbb{R}^2$ et où e_m est le vecteur de \mathbb{R}^m dont toutes les composantes sont égales à 1.

Notre motivation pour introduire ce problème est la suivante. étant donné un point de départ (x_1, y_1) , pas nécessairement réalisable pour les contraintes d'inégalité, il est facile de trouver un triplet (x_1, s_1, y_1) strictement réalisable pour le problème $(\mathcal{P}^{\sigma,\omega})$. On a déjà vu que le point $(x_1, y_1) = (x_1, e_p)$ est réalisable pour la contrainte d'égalité linéaire. Il suffit ensuite de choisir s_1 de manière à satisfaire l'inégalité $c(x_1) < s_1$. De plus, à condition que les paramètres de pénalité ω et σ soient suffisamment grandes, un algorithme de points intérieurs appliqué au problème $(\mathcal{P}^{\sigma,\omega})$ génère une suite d'itérés qui convergent vers la solution du problème (\mathcal{P}) initial.

Les conditions d'optimalité du problème $(\mathcal{P}^{\sigma,\omega})$ peuvent s'écrire

$$\left\{ \begin{array}{l} \nabla f(x) + \nabla c(x)\lambda - A^T \xi = 0, \\ \sigma e_m - \lambda - \tau = 0 \\ \omega e_p - D\xi - \nu = 0 \\ (S - C(x))\lambda = 0 \\ S\tau = 0 \\ Y\nu = 0 \\ Ax + Dy = b \\ (s - c(x), s, y, \lambda, \tau, \nu) \geq 0, \end{array} \right.$$

où $S = \text{diag}(s_1, \dots, s_m)$ et $Y = \text{diag}(y_1, \dots, y_p)$ sont deux matrices diagonales. On note λ le vecteur de \mathbb{R}^m des multiplicateurs associé aux contraintes $c(x) \leq s$, τ celui de \mathbb{R}^m associé aux contraintes $s \geq 0$, ξ celui de \mathbb{R}^p associé aux contraintes linéaires d'égalité et ν celui de \mathbb{R}^p associé aux contraintes $y \geq 0$. Avec les deuxième et troisième équations, on élimine les multiplicateurs $\tau = \sigma e_m - \lambda$ et $\nu = \omega e_p - D\xi$. On obtient

$$\left\{ \begin{array}{l} \nabla f(x) + \nabla c(x)\lambda - A^T \xi = 0, \\ (S - C(x))\lambda = 0 \\ S(\sigma e_m - \lambda) = 0 \\ Y(\omega e_p - D\xi) = 0 \\ Ax + Dy = b \\ (s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) \geq 0, \end{array} \right. \quad (3.7)$$

Notons que $(\sigma e_m - \lambda)$ est alors le vecteur des multiplicateurs associé aux contraintes $s \geq 0$ et que $(\omega e_p - D\xi)$ est celui associé aux contraintes $y \geq 0$.

Proposition 3.4.1 *Supposons l'hypothèse H_1 satisfaite. Alors, pour tout vecteur de pénalité $(\sigma, \omega) \in \mathbb{R}^2$, tels que $(\sigma, \omega) > 0$, le problème $(\mathcal{P}^{\sigma,\omega})$ est strictement réalisable et son ensemble de solution est non vide et compact.*

De plus, si $(\hat{x}, \hat{\lambda}, \hat{\xi})$ est une solution primale-duale de (\mathcal{P}) , alors pour tout $(\sigma, \omega) \in \mathbb{R}^2$, tels que $\sigma \geq \|\hat{\lambda}\|_\infty$ et $\omega \geq \|D\hat{\xi}\|_\infty$, $(\hat{x}, \hat{s}, \hat{y}, \hat{\lambda}, \hat{\xi})$ est une solution primale duale de $(\mathcal{P}^{\sigma,\omega})$, avec $\hat{s} = 0$ et $\hat{y} = 0$. Inversement, soit $(\hat{x}, \hat{s}, \hat{y}, \hat{\lambda}, \hat{\xi})$ une solution de $(\mathcal{P}^{\sigma,\omega})$. Si $\sigma > \|\hat{\lambda}\|_\infty$ et si $\omega > \|D\hat{\xi}\|_\infty$ alors $\hat{s} = 0$, $\hat{y} = 0$, $A\hat{x} = b$ et $(\hat{x}, \hat{\lambda}, \hat{\xi})$ est une solution primale-duale de (\mathcal{P}) .

Preuve. Montrons tout d'abord que le problème $(\mathcal{P}^{\sigma, \omega})$ est réalisable. Le point $(x, y) = (x_1, e_p)$ est réalisable pour la contrainte d'égalité du problème $(\mathcal{P}^{\sigma, \omega})$. Le choix de la variable s_1 se fait alors dans l'espace $(\mathbb{R}^m)^+$ avec $(s_1)_i > c_i(x_1)$ pour $i \in \{1, \dots, m\}$.

D'après le corollaire 3.2.5, l'ensemble des solutions du problème $(\mathcal{P}^{\sigma, \omega})$ est non vide et compact si et seulement si il n'existe pas de direction $(d^x, d^s, d^y) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$ non nulle telle que

$$\begin{cases} \Theta_{\sigma, \omega}^{\infty}(d^x, d^s, d^y) = f^{\infty}(d^x) + \sigma e_m^{\top} d^s + \omega e_p^{\top} d^y \leq 0 \\ Ad^x + Dd^y = 0 \\ d^s \geq 0 \\ d^y \geq 0 \\ c_i^{\infty}(d^x) \leq d_i^s, \forall i \in \{1, \dots, m\}. \end{cases}$$

On sait, d'après la proposition 3.3.3, qu'il n'existe pas une direction d^x non nulle telle que $f^{\infty}(d^x) < \infty$ et $c_i^{\infty}(d^x) < \infty$ pour $i \in \{1, \dots, m\}$. On a donc $d^x = 0$. Par positivité des paramètres σ et ω et des directions d^s et d^y , on a $d^s = 0$ et $d^y = 0$.

Montrons maintenant la deuxième partie de la proposition. Soit $(\hat{x}, \hat{\lambda}, \hat{\xi})$ une solution primale-duale de (\mathcal{P}) . $(\hat{x}, \hat{\lambda}, \hat{\xi})$ vérifie donc les conditions d'optimalité (3.2). En prenant $(\sigma, \omega) \in \mathbb{R}^2$, tels que $\sigma \geq \|\hat{\lambda}\|_{\infty}$ et $\omega \geq \|D\hat{\xi}\|_{\infty}$, et $(\hat{s}, \hat{y}) \in \mathbb{R}^m \times \mathbb{R}^p$ avec $\hat{s} = 0$ et $\hat{y} = 0$, le vecteur $(\hat{x}, \hat{s}, \hat{y}, \hat{\lambda}, \hat{\xi})$ est une solution de $(\mathcal{P}^{\sigma, \omega})$.

Prenons maintenant $(\hat{x}, \hat{s}, \hat{y}, \hat{\lambda}, \hat{\xi})$ une solution de $(\mathcal{P}^{\sigma, \omega})$ et $(\sigma, \omega) \in \mathbb{R}^2$, tels que $\sigma > \|\hat{\lambda}\|_{\infty}$ et $\omega > \|D\hat{\xi}\|_{\infty}$. D'après les conditions de complémentarité du système (3.7), cela entraîne que $\hat{s} = 0$ et que $\hat{y} = 0$. En éliminant ensuite les variables nulles \hat{s} et \hat{y} , on vérifie facilement que $(\hat{x}, \hat{\lambda}, \hat{\xi})$ satisfait les conditions d'optimalité (3.2). \square

Une autre pénalisation possible est la suivante :

$$(\mathcal{P}^{\omega}) \begin{cases} \min f(x) + \omega y \\ c(x) \leq 0, \\ Ax + yd = b, \\ y \geq 0, \end{cases}$$

où y est un réel et d le vecteur de \mathbb{R}^p dont toutes les composantes sont les éléments de la diagonale de D . Cette approche diminue le nombre de variables y ajoutées au problème initial et la contrainte linéaire d'égalité est pénalisée de manière identique sur chacune de ses composantes. Les conditions d'optimalité de ce problème sont

$$\begin{cases} \nabla f(x) + \nabla c(x)\lambda - A^{\top}\xi = 0, \\ \sigma e_m - \lambda - \tau = 0 \\ \omega - d^{\top}\xi - \nu = 0 \\ (S - C(x))\lambda = 0 \\ S\tau = 0 \\ y\nu = 0 \\ Ax + yd = b \\ (s - c(x), s, y, \lambda, \tau, \nu) \geq 0. \end{cases}$$

On note λ le vecteur de \mathbb{R}^m des multiplicateurs associé aux contraintes $c(x) \leq s$, τ celui de \mathbb{R}^m associé aux contraintes $s \geq 0$, ξ celui de \mathbb{R}^p associé aux contraintes linéaires d'égalité et ν celui de \mathbb{R} associé à la contrainte $y \geq 0$. Avec les deuxième et troisième équations, on élimine les multiplicateurs $\tau = \sigma e_m - \lambda$ et $\nu = \omega - d^T \xi$. On obtient

$$\begin{cases} \nabla f(x) + \nabla c(x)\lambda - A^T \xi = 0, \\ (S - C(x))\lambda = 0 \\ S(\sigma e_m - \lambda) = 0 \\ y(\omega - d^T \xi) = 0 \\ Ax + yd = b \\ (s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega - d^T \xi) \geq 0, \end{cases}$$

Notons que $(\sigma e_m - \lambda)$ est alors le vecteur des multiplicateurs associé aux contraintes $s \geq 0$ et que $(\omega - d^T \xi)$ est celui associé aux contraintes $y \geq 0$. On remarque que le changement de ν en $(\omega - d^T \xi)$ n'est pas bijectif. En effet, en ajoutant à ξ n'importe quel vecteur orthogonal à d , on obtient le même ν . Cette non bijectivité entraîne que la solution notée $(\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu, \hat{\lambda}_\mu, \hat{\xi}_\mu)$ du problème $(\mathcal{PD})_\mu^{\sigma, \omega}$ (voir section 3.7 de ce chapitre) n'est plus unique. La solution n'étant plus unique, la convergence de l'algorithme vers la solution cherchée n'est plus assurée. On a donc abandonné cette technique au profit de celle qui ajoute autant de variables y qu'il y a de contrainte d'égalité. La solution du problème $(\mathcal{PD})_\mu^{\sigma, \omega}$ est alors unique et l'algorithme présenté converge vers la bonne solution.

3.5 Le problème barrière

On perturbe les conditions de complémentarité du système (3.7) avec un paramètre réel μ strictement positif. Le système des conditions d'optimalité perturbées s'écrit alors

$$\begin{cases} \nabla f(x) + \nabla c(x)\lambda - A^T \xi = 0, \\ (S - C(x))\lambda = \mu e_m, \\ S(\sigma e_m - \lambda) = \mu e_m, \\ Y(\omega e_p - D\xi) = \mu e_p, \\ Ax + Dy = b \\ (s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) > 0, \end{cases} \quad (3.8)$$

Les deuxième, troisième et quatrième égalités du système précédent nous donnent

$$\begin{cases} \lambda = \mu (S - C(x))^{-1} e_m, \\ (\sigma e_m - \lambda) = \mu S^{-1} e_m, \\ (\omega e_p - D\xi) = \mu Y^{-1} e_p. \end{cases}$$

On obtient finalement en éliminant les multiplicateurs λ , $\sigma e_m - \lambda$ et $\omega e_p - D\xi$ du système (3.8)

$$\begin{cases} \nabla f(x) - \mu \nabla c(x)(S - C(x))^{-1} e_m - A^\top \xi = 0, \\ \sigma e_m - \mu(S - C(x))^{-1} e_m - \mu S^{-1} e_m = 0 \\ (\omega e_p - D\xi) - \mu Y^{-1} e_p = 0 \\ Ax + Dy = b \\ (s - c(x), s, y) > 0. \end{cases}$$

On peut interpréter ce système comme celui des conditions d'optimalité du problème barrière associé au problème $(\mathcal{P}^{\sigma, \omega})$:

$$\mathcal{P}_\mu^{\sigma, \omega} \begin{cases} \min \varphi_\mu^{\sigma, \omega}(x, s, y) \\ Ax + Dy = b, \end{cases} \quad (3.9)$$

où $(\sigma, \omega) \in \mathbb{R}^2$ et où $\varphi_\mu^{\sigma, \omega}$ est la fonction barrière paramétrée par $\mu > 0$ définie par

$$\begin{aligned} \varphi_\mu^{\sigma, \omega}(x, s, y) &= f(x) + \sigma e_m^\top s + \omega e_p^\top y - \mu \sum_{i=1}^m \log(s_i - c_i(x)) \\ &\quad - \mu \sum_{i=1}^m \log(s_i) - \mu \sum_{i=1}^p \log(y_i). \end{aligned}$$

L'idée des méthodes de points intérieurs est de résoudre de manière approchée une suite de problèmes barrière en faisant tendre le paramètre de perturbation μ vers 0. La proposition suivante montre que la forte convexité du lagrangien (lemme 3.3.2) entraîne celle de la fonction barrière.

Proposition 3.5.1 *Supposons l'hypothèse H_1 satisfaite. Alors, pour tout vecteur de pénalité $(\sigma, \omega) > 0$, et pour tout paramètre barrière $\mu > 0$, la fonction barrière est strictement convexe sur son domaine et ses ensembles de niveau sont compacts.*

Preuve. Le gradient et le hessien de la fonction barrière sont donnés par (en omettant certaines dépendances en x) par les formules suivantes, où $\xi \in \mathbb{R}^p$,

$$\nabla \varphi_\mu^{\sigma, \omega}(x, s, y) = \begin{pmatrix} \nabla f + \mu \nabla c(S - C)^{-1} e_m \\ \sigma e_m - \mu(S - C)^{-1} e_m - \mu S^{-1} e_m \\ \omega e_p - \mu Y^{-1} e_p \end{pmatrix} \quad (3.10)$$

et

$$\nabla^2 \varphi_\mu^{\sigma, \omega}(x, s, y) = \begin{pmatrix} \nabla_{xx}^2 \varphi_\mu^{\sigma, \omega}(x, s, y) & -\mu \nabla c(S - C)^{-2} & 0 \\ -\mu(S - C)^{-2} \nabla c^\top & \mu(S - C)^{-2} + \mu S^{-2} & 0 \\ 0 & 0 & \mu Y^{-2} \end{pmatrix}, \quad (3.11)$$

où

$$\begin{aligned} \nabla_{xx}^2 \varphi_\mu^{\sigma, \omega}(x, s, y) &= \nabla_{xx}^2 f(x) + \mu \sum_{i=1}^m (s_i - c_i(x))^{-1} \nabla_{xx}^2 c_i(x) + \mu \nabla c(S - C)^{-2} \nabla c^\top \\ &= \nabla_{xx}^2 \ell(x, \mu(S - C)^{-1} e_m, \xi) + \mu \nabla c(S - C)^{-2} \nabla c^\top. \end{aligned} \quad (3.12)$$

On peut donc écrire le hessien de $\varphi_\mu^{\sigma,\omega}(x, s, t)$ de la manière suivante

$$\begin{aligned} \nabla^2 \varphi_\mu^{\sigma,\omega}(x, s, y) &= \begin{pmatrix} \nabla_{xx}^2 \ell(x, \mu(S-C)^{-1} e_m, \xi) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ &+ \mu \begin{pmatrix} \nabla c(S-C)^{-2} \nabla c^\top & -\nabla c(S-C)^{-2} & 0 \\ (S-C)^{-2} \nabla c^\top & (S-C)^{-2} & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ &+ \mu \begin{pmatrix} 0 & 0 & 0 \\ 0 & S^{-2} & 0 \\ 0 & 0 & 0 \end{pmatrix} + \mu \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & Y^{-2} \end{pmatrix}. \end{aligned}$$

Ainsi, pour des vecteurs (u, v, t) de $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$, on a

$$\begin{aligned} &\begin{pmatrix} u \\ v \\ t \end{pmatrix}^\top \nabla^2 \varphi_\mu^{\sigma,\omega}(x, s, y) \begin{pmatrix} u \\ v \\ t \end{pmatrix} \\ &= u^\top \nabla_{xx}^2 \ell(x, \mu(S-C)^{-1} e_m, \xi) u + \mu t^\top Y^{-2} t + \mu v^\top S^{-2} v \\ &\quad + \mu \begin{pmatrix} u \\ v \end{pmatrix}^\top \begin{pmatrix} \nabla c(S-C)^{-2} \nabla c^\top & -\nabla c(S-C)^{-2} \\ -(S-C)^{-2} \nabla c^\top & (S-C)^{-2} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \\ &= u^\top \nabla_{xx}^2 \ell(x, \mu(S-C)^{-1} e_m, \xi) u + \mu t^\top Y^{-2} t + \mu v^\top S^{-2} v \\ &\quad + \mu [(u^\top \nabla c(S-C)^{-2} \nabla c^\top u - u^\top \nabla c(S-C)^{-2} v \\ &\quad - v^\top (S-C)^{-2} \nabla c^\top u + v^\top (S-C)^{-2} v] \\ &= u^\top \nabla_{xx}^2 \ell(x, \mu(S-C)^{-1} e_m, \xi) u + \mu t^\top Y^{-2} t + \mu v^\top S^{-2} v \\ &\quad + \mu [(\nabla c^\top u)^\top (S-C)^{-2} (\nabla c^\top u) - (\nabla c^\top u) (S-C)^{-2} v \\ &\quad - v^\top (S-C)^{-2} (\nabla c^\top u) + v^\top (S-C)^{-2} v] \\ &= u^\top \nabla_{xx}^2 \ell(x, \mu(S-C)^{-1} e_m, \xi) u + \mu \|Y^{-1} t\|^2 + \mu \|S^{-1} v\|^2 \\ &\quad + \mu \|(S-C)^{-1} (\nabla c^\top u - v)\|^2. \end{aligned} \tag{3.13}$$

On sait, d'après le lemme 3.3.2, que le lagrangien est fortement convexe. Son hessien est donc une matrice définie positive. Par non négativité des termes de (3.13), si le produit

$$\begin{pmatrix} u \\ v \\ t \end{pmatrix}^\top \nabla^2 \varphi_\mu^{\sigma,\omega}(x, s, y) \begin{pmatrix} u \\ v \\ t \end{pmatrix} = 0$$

alors

$$(u, v, t) = 0.$$

Cela montre la stricte convexité de la fonction $\varphi_\mu^{\sigma,\omega}(x, s, y)$.

Montrons maintenant la compacité des ensembles de niveau de la fonction barrière. On utilise pour cela la dérivée asymptotique de la fonction barrière. La dérivée asymptotique de la fonction barrière dans la direction $d = (d^x, d^s, d^y)$

peut s'écrire, en rappelant (3.4),

$$(\varphi_\mu^{\sigma,\omega})^\infty(d) = \begin{cases} f^\infty(d^x) + \sigma e_m^\top d^s + \omega e_p^\top d^y & \text{si } (d^s - c^\infty(d^x), d^s, d^y) > 0 \\ +\infty & \text{si } (d^s - c^\infty(d^x), d^s, d^y) \leq 0. \end{cases} \quad (3.14)$$

D'après la proposition 3.2.4, la compacité des ensembles de niveau de la fonction barrière revient à montrer que $(\varphi_\mu^{\sigma,\omega})^\infty(d) > 0$ pour toute direction d non nulle. Supposons le contraire : il existe une direction $d = (d^x, d^s, d^y)$ non nulle telle que $(\varphi_\mu^{\sigma,\omega})^\infty(d) \leq 0$. On a donc une direction d non nulle telle que

$$(\varphi_\mu^{\sigma,\omega})^\infty(d) \leq 0 \text{ et } (d^s - c^\infty(d^x), d^s, d^y) \geq 0.$$

Donc, d'après (3.14), on a une direction d non nulle

$$f^\infty(d^x) + \sigma e_m^\top d^s + \omega e_p^\top d^y \leq 0 \text{ et } (d^s - c^\infty(d^x), d^s, d^y) \geq 0.$$

Or, on a vu dans la démonstration de la proposition 3.4.1 qu'il n'existe pas de direction d non nulle telle que

$$\begin{cases} f^\infty(d^x) + \sigma e_m^\top d^s + \omega e_p^\top d^y \leq 0 \\ Ad^x + Dd^s = 0 \\ d^s \geq 0 \\ d^y \geq 0 \\ d_i^s - c_i^\infty(x) \geq 0, \forall i \in \{1, \dots, m\}. \end{cases}$$

Un telle direction d ne peut exister et on en déduit la compacité des ensembles de niveau de la fonction barrière. \square

La proposition suivante établit un lien entre la solution du système perturbé (3.8) et celle du problème barrière $\mathcal{P}_\mu^{\sigma,\omega}$.

Proposition 3.5.2 *Supposons l'hypothèse H_1 satisfaite. Alors, pour tout vecteur de pénalité $(\sigma, \omega) > 0$, et pour tout paramètre barrière $\mu > 0$, le problème barrière $\mathcal{P}_\mu^{\sigma,\omega}$ a une solution unique, notée $(\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu)$. Elle est caractérisée par l'existence de $\hat{\lambda}_\mu \in \mathbb{R}^m$ et de $\hat{\xi}_\mu \in \mathbb{R}^p$ tels que le vecteur $(\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu, \hat{\lambda}_\mu, \hat{\xi}_\mu)$ soit une solution du système (3.8).*

Preuve. Il est clair que le domaine de $\varphi_\mu^{\sigma,\omega}$ est non vide (en effet $(x_1, s_1, t_1) \in \text{dom}(\varphi_\mu^{\sigma,\omega})$). Par stricte convexité de $\varphi_\mu^{\sigma,\omega}$ (proposition 3.5.1), le problème barrière a une solution unique sur l'ensemble

$$L = \{(x, s, y) : \varphi_\mu^{\sigma,\omega}(x, s, y) \leq \varphi_\mu^{\sigma,\omega}(x_1, s_1, y_1)\} \cap \{(x, s, y) : Ax + Dy = b\}$$

On a donc les égalités suivantes

$$\begin{cases} \nabla f(\hat{x}_\mu) + \mu \nabla c(\hat{x}_\mu) (\hat{S}_\mu - C(\hat{x}_\mu))^{-1} e_m - A^\top \hat{\xi}_\mu = 0, \\ \sigma e_m - \mu (\hat{S}_\mu - C(\hat{x}_\mu))^{-1} e_m - \mu \hat{S}_\mu^{-1} e_m = 0, \\ \omega e_p - D \hat{\xi}_\mu - \mu (Y_\mu)^{-1} e_p = 0 \\ A \hat{x}_\mu + D \hat{y}_\mu = b \\ (\hat{s}_\mu - c(\hat{x}_\mu), \hat{s}_\mu, \hat{y}_\mu) > 0. \end{cases}$$

En définissant $\hat{\lambda}_\mu$ par $(\hat{\lambda}_\mu)_i := \mu / ((\hat{s}_\mu)_i - c_i(\hat{x}_\mu))$ pour $i \in \{1, \dots, m\}$, on obtient les conditions (3.8). \square

3.6 Le pas de Newton

Pour résoudre le système des conditions d'optimalité perturbées (3.8), on utilise la méthode de Newton. Le pas de Newton $(d^x, d^s, d^y, d^\lambda, d^\xi) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^p$ de (3.8) est la solution de

$$\begin{pmatrix} M & 0 & 0 & \nabla c & -A^\top \\ -\Lambda \nabla c^\top & \Lambda & 0 & S-C & 0 \\ 0 & \sigma I - \Lambda & 0 & -S & 0 \\ 0 & 0 & \omega I - D \Xi & 0 & -YD \\ A & 0 & D & 0 & 0 \end{pmatrix} \begin{pmatrix} d^x \\ d^s \\ d^y \\ d^\lambda \\ d^\xi \end{pmatrix} = - \begin{pmatrix} \nabla_x \ell(x, \lambda, \xi) \\ (S-C)\lambda - \mu e_m \\ S(\sigma e_m - \lambda) - \mu e_m \\ Y(\omega e_p - D\xi) - \mu e_p \\ 0 \end{pmatrix} \quad (3.15)$$

où Λ, Ξ sont deux matrices diagonales définies par $\Lambda := \text{diag}(\lambda_1, \dots, \lambda_m)$ et $\Xi := \text{diag}(\xi_1, \dots, \xi_p)$, et où M est une approximation définie positive du hessien du lagrangien obtenue par les formules de BFGS ou par celles de BFGS à mémoire limitée.

On peut interpréter ce système comme les conditions d'optimalité d'un problème quadratique. On va tout d'abord intervertir certaines lignes de ce système

$$\begin{pmatrix} M & 0 & 0 & \nabla c & -A^\top \\ 0 & \sigma I - \Lambda & 0 & -S & 0 \\ 0 & 0 & \omega I - D \Xi & 0 & -YD \\ -\Lambda \nabla c^\top & \Lambda & 0 & S-C & 0 \\ A & 0 & D & 0 & 0 \end{pmatrix} \begin{pmatrix} d^x \\ d^s \\ d^y \\ \lambda^+ \\ \xi^+ \end{pmatrix} = - \begin{pmatrix} \nabla f(x) \\ \sigma s - \mu e_m \\ \omega y - \mu e_p \\ -\mu e_m \\ 0 \end{pmatrix}$$

où $\lambda^+ = \lambda + d^\lambda$ et $\xi^+ = \xi + d^\xi$. On factorise maintenant certains termes pour symétriser le système :

$$\begin{pmatrix} M & 0 & 0 & \nabla c & | & -A^\top \\ 0 & S^{-1}(\sigma I - \Lambda) & 0 & -I & | & 0 \\ 0 & 0 & Y^{-1}(\omega I - D \Xi) & 0 & | & -D \\ -\nabla c^\top & I & 0 & \Lambda^{-1}(S-C) & | & 0 \\ \hline A & 0 & D & 0 & | & 0 \end{pmatrix} \begin{pmatrix} d^x \\ d^s \\ d^y \\ \lambda^+ \\ \xi^+ \end{pmatrix} = - \begin{pmatrix} \nabla f(x) \\ (\sigma I - \mu S^{-1})e_m \\ (\omega I - \mu Y^{-1})e_p \\ -\mu \Lambda^{-1}e_m \\ 0 \end{pmatrix}.$$

Notons Q la matrice de taille $(n + 2m + p) \times (n + 2m + p)$ définie par

$$Q := \begin{pmatrix} M & 0 & 0 & \nabla c \\ 0 & S^{-1}(\sigma I - \Lambda) & 0 & -I \\ 0 & 0 & Y^{-1}(\omega I - D \Xi) & 0 \\ -\nabla c^\top & I & 0 & \Lambda^{-1}(S-C) \end{pmatrix}.$$

Notons \bar{A} la matrice de taille $p \times (n + 2m + p)$ définie par

$$\bar{A} = \begin{pmatrix} A & 0 & D & 0 \end{pmatrix}$$

Notons q le vecteur de taille $(n + 2m + p)$ défini par

$$q := - \begin{pmatrix} \nabla f(x) \\ (\sigma I - \mu S^{-1})e_m \\ (\omega I - \mu Y^{-1})e_p \\ -\mu \Lambda^{-1}e_m \end{pmatrix}.$$

Le système précédent peut alors s'interpréter comme la solution du problème suivant, où $d = (d^x, d^s, d^y, \lambda^+)$,

$$\begin{cases} \min \frac{1}{2} d^T Q d + q^T d, \\ \bar{A} d = 0. \end{cases} \quad (3.16)$$

Ce problème est un problème quadratique (ce n'est pas le problème quadratique tangent associé au problème $\mathcal{P}_\mu^{\sigma, \omega}$ car ce n'est pas le hessien de $\varphi_\mu^{\sigma, \omega}$ qui intervient). Il peut être résolu en employant divers algorithmes (gradients conjugués, Newton tronqué).

On remarque que résoudre le système (3.15) est équivalent à résoudre le problème (3.16). La proposition suivante montre que le système (3.15) a une unique solution sous certaines hypothèses.

Proposition 3.6.1 *Supposons que M soit définie positive et que $(s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) > 0$, alors le problème (3.16) a une solution unique, ce qui est équivalent à dire que le système (3.15) a une unique solution.*

Preuve. Le problème (3.16) est un problème avec une contrainte linéaire d'égalité. On va montrer que la matrice Q est définie positive d'après les hypothèses de la proposition. Le problème (3.16) est donc un problème quadratique avec une contrainte linéaire d'égalité. Ce problème aura donc une unique solution. Soit (u, v, t, r) un vecteur de $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^p$:

$$\begin{aligned} (u, v, t, r)^T Q \begin{pmatrix} u \\ v \\ t \\ r \end{pmatrix} &= u^T M u + u^T \nabla c r + v^T S^{-1} (\sigma I - \Lambda) v - v^T r \\ &\quad + t^T Y^{-1} (\omega I - D\xi) t - r^T \nabla c^T u + r^T v \\ &\quad + r^T \Lambda^{-1} (S - C) r \\ &= u^T M u + v^T S^{-1} (\sigma I - \Lambda) v + t^T Y^{-1} (\omega I - D\xi) t \\ &\quad + r^T \Lambda^{-1} (S - C) r. \end{aligned}$$

D'après l'hypothèse de définie positivité de la matrice M et comme $(s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) > 0$, si le produit ci-dessus s'annule, alors $u = 0, v = r = 0$ et $t = 0$. La matrice Q est donc définie positive.

L'unicité de la solution du système (3.15) s'obtient avec la remarque précédant cette proposition. \square

La proposition suivante montre que la solution du système (3.15) est une direction de descente de la fonction barrière.

Proposition 3.6.2 *Supposons que (x, s, y, λ, ξ) satisfait $(s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) > 0$ et que M soit symétrique définie positive. Soit $(d^x, d^s, d^y, d^\lambda, d^\xi)$ l'unique solution de (3.15), alors*

$$\begin{aligned} \nabla \varphi_\mu^{\sigma, \omega}(x, s, y)^\top \begin{pmatrix} d^x \\ d^s \\ d^y \end{pmatrix} &= -(d^x)^\top M d^x - \|(\sigma I - \Lambda)^{1/2} S^{-1/2} d^s\|^2 \\ &\quad - \|\Upsilon^{1/2} (\omega I - D\xi)^{-1/2} d^y\|^2 \\ &\quad - \|\Lambda^{1/2} (S - C)^{-1/2} (\nabla c^\top d^x - d^s)\|^2 \\ &\leq 0, \end{aligned} \tag{3.17}$$

et donc (d^x, d^s, d^y) est une direction de descente de $\varphi_\mu^{\sigma, \omega}$ au point $(x, s, y) \neq (\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu)$.

Preuve. En développant le système (3.15), on obtient

$$\begin{cases} M d^x + \nabla c \lambda^+ - A^\top \xi^+ &= -\nabla f(x) \\ -\Lambda \nabla c^\top d^x + \Lambda d^s + (S - C) \lambda^+ &= \mu e_m \\ (\sigma I - \Lambda) d^s - S \lambda^+ &= \mu e_m - \sigma s \\ (\omega I - D\xi) d^y - Y D \xi^+ &= \mu e_p - \omega y \\ A d^x + D d^y &= 0, \end{cases}$$

en notant $\lambda^+ := \lambda + d^\lambda$, $\xi^+ := \xi + d^\xi$. En isolant λ^+ dans les seconde et troisième égalités de ce système, on obtient

$$\begin{aligned} \lambda^+ &= \sigma e_m - \mu S^{-1} e_m + S^{-1} (\sigma I - \Lambda) d^s \\ &= \mu (S - C)^{-1} e_m + (S - C)^{-1} \Lambda \nabla c^\top d^x - (S - C)^{-1} \Lambda d^s. \end{aligned}$$

Ainsi,

$$\sigma e_m - \mu (S - C)^{-1} e_m - \mu S^{-1} e_m = -(\sigma I - \Lambda) S^{-1} d^s - (S - C)^{-1} \Lambda (\nabla c d^x - d^s) \tag{3.18}$$

puis, en regroupant certains termes et en rappelant la valeur de $\nabla_s \varphi_\mu^{\sigma, \omega}$,

$$\begin{aligned} (\nabla_s \varphi_\mu^{\sigma, \omega})^\top d^s &= (\sigma e_m - \mu (S - C)^{-1} e_m - \mu S^{-1} e_m)^\top d^s \\ &= -(d^s)^\top ((\sigma I - \Lambda) S^{-1} + \Lambda (S - C)^{-1}) d^s + (d^x)^\top \nabla c \Lambda (S - C)^{-1} d^s \end{aligned}$$

La quatrième égalité du système nous donne

$$\omega e_p - \mu Y^{-1} e_p = -Y^{-1} (\omega I - D\xi) d^y + D \xi^+, \tag{3.19}$$

et en rappelant la définition de $\nabla_y \varphi_\mu^{\sigma, \omega}$, on a

$$\begin{aligned} (\nabla_y \varphi_\mu^{\sigma, \omega})^\top d^y &= (\omega e_p - \mu Y^{-1} e_p)^\top d^y \\ &= -(d^y)^\top Y^{-1} (\omega I - D\xi) d^y + (D \xi^+)^\top d^y \end{aligned}$$

On va maintenant évaluer le produit $(\nabla_x \varphi_\mu^{\sigma, \omega})^\top d^x$. La première égalité du système ci-dessus et la valeur de λ^+ nous donne

$$\nabla f + \mu \nabla c (S - C)^{-1} e_m = -(M d^x + \nabla c (S - C)^{-1} \Lambda (\nabla c^\top d^x - d^s) - A^\top \xi^+) \tag{3.20}$$

puis en rappelant la définition de $\nabla_x \varphi_\mu^{\sigma, \omega}$, on obtient

$$\begin{aligned} (\nabla_x \varphi_\mu^{\sigma, \omega})^\top d^x &= (\nabla f + \mu \nabla c (S - C)^{-1} e_m)^\top d^x & (3.21) \\ &= -(d^x)^\top (M + \nabla c (S - C)^{-1} \Lambda \nabla c^\top) d^x + (d^s)^\top \Lambda (S - C)^{-1} \nabla c^\top d^x \\ &\quad + (\xi^+)^\top \Lambda d^x. \end{aligned}$$

Calculons maintenant le produit $(\nabla \varphi_\mu^{\sigma, \omega})^\top d$:

$$\begin{aligned} (\nabla \varphi_\mu^{\sigma, \omega})^\top \begin{pmatrix} d^x \\ d^s \\ d^y \end{pmatrix} &= (\nabla_x \varphi_\mu^{\sigma, \omega})^\top d^x + (\nabla_s \varphi_\mu^{\sigma, \omega})^\top d^s + \nabla_y \varphi_\mu^{\sigma, \omega} d^y \\ &= -(d^x)^\top (M + \nabla c (S - C)^{-1} \Lambda \nabla c^\top) d^x \\ &\quad - (d^s)^\top ((\sigma I - \Lambda) S^{-1} + \Lambda (S - C)^{-1}) d^s \\ &\quad + 2(d^s)^\top \Lambda (S - C)^{-1} \nabla c^\top d^x \\ &\quad - (d^y)^\top Y^{-1} (\omega I - D \Xi) d^y + (D \xi^+)^\top d^y + (\xi^+)^\top \Lambda d^x. \end{aligned}$$

On remarque que

$$(D^\top \xi^+) d^y + (\xi^+)^\top \Lambda d^x = (\xi^+)^\top (\Lambda d^x + D d^y) = 0.$$

Ainsi

$$\begin{aligned} (\nabla \varphi_\mu^{\sigma, \omega})^\top \begin{pmatrix} d^x \\ d^s \\ d^y \end{pmatrix} &= -(d^x)^\top (M + \nabla c (S - C)^{-1} \Lambda \nabla c^\top) d^x \\ &\quad - (d^s)^\top ((\sigma I - \Lambda) S^{-1} + \Lambda (S - C)^{-1}) d^s \\ &\quad + 2(d^s)^\top \Lambda (S - C)^{-1} \nabla c^\top d^x \\ &\quad - (d^y)^\top Y^{-1} (\omega I - D \Xi) d^y. \end{aligned}$$

On a donc, en regroupant les termes en d^x et en d^s

$$\begin{aligned} (\nabla \varphi_\mu^{\sigma, \omega})^\top \begin{pmatrix} d^x \\ d^s \\ d^y \end{pmatrix} &= -(d^x)^\top (M + \nabla c (S - C)^{-1} \Lambda \nabla c^\top) d^x \\ &\quad - (d^s)^\top ((\sigma I - \Lambda) S^{-1} + \Lambda (S - C)^{-1}) d^s \\ &\quad + 2(d^s)^\top \Lambda (S - C)^{-1} \nabla c^\top d^x \\ &\quad - (d^y)^\top Y^{-1} (\omega I - D \Xi) d^y \\ &= - \begin{pmatrix} d^x \\ d^s \\ d^y \end{pmatrix}^\top \begin{pmatrix} M & 0 & 0 \\ 0 & S^{-1}(\sigma I - \Lambda) & 0 \\ 0 & 0 & Y^{-1}(\omega I - D \Xi) \end{pmatrix} \begin{pmatrix} d^x \\ d^s \\ d^y \end{pmatrix} \\ &\quad - \begin{pmatrix} d^x \\ d^s \end{pmatrix} \begin{pmatrix} \nabla c (S - C)^{-1} \Lambda \nabla c^\top & -\nabla c (S - C)^{-1} \Lambda \\ -\Lambda (S - C)^{-1} \nabla c^\top & (S - C)^{-1} \Lambda \end{pmatrix} \begin{pmatrix} d^x \\ d^s \end{pmatrix} \\ &= -(d^x)^\top M d^x - \|(\sigma I - \Lambda)^{1/2} S^{-1/2} d^s\|^2 \\ &\quad - \|Y^{-1/2} (\omega I - D \Xi)^{1/2} d^y\|^2 \\ &\quad - \|\Lambda^{1/2} (S - C)^{-1/2} (\nabla c^\top d^x - d^s)\|^2. \end{aligned}$$

Avec les hypothèses $(s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) > 0$, M symétrique définie positive et l'égalité précédente, on obtient le fait que (d^x, d^s, d^y) est une direction de descente de la fonction $\varphi_\mu^{\sigma, \omega}$.

Si le produit (3.17) s'annule, alors par définie positivité de M et par positivité des variables $(s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi)$, $(d^x, d^s, d^y) = 0$. Comme $(d^x, d^s, d^y, d^\lambda, d^\xi)$ est la solution de (3.15), on en déduit que $(x, s, y, \lambda + d^\lambda, \xi + d^\xi)$ est alors une solution de (3.8). D'après la proposition 3.5.2, on en déduit que $(x, s, y) = (\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu)$. \square

3.7 Le problème barrière primal-dual

On ajoute à la fonction barrière, qui est une fonction primale, un terme primal-dual, afin de pouvoir contrôler les déplacements des variables duales. On va donc résoudre de manière approchée le problème barrière suivant

$$(\mathcal{PD})_\mu^{\sigma, \omega} \begin{cases} \min \psi_\mu^{\sigma, \omega}(x, s, y, \lambda, \xi) \\ Ax + Dy = b, \end{cases} \quad (3.22)$$

avec

$$\psi_\mu^{\sigma, \omega}(x, s, y, \lambda, \xi) = \varphi_\mu^{\sigma, \omega}(x, s, y) + \tau \mathcal{V}_\mu^{\sigma, \omega}(x, s, y, \lambda, \xi), \quad (3.23)$$

où

$$\begin{aligned} \mathcal{V}_\mu^{\sigma, \omega}(x, s, y, \lambda, \xi) &= \lambda^\top (s - c(x)) - \mu \sum_{i=1}^m \log(\lambda_i (s_i - c_i(x))) \\ &\quad + s^\top (\sigma e_m - \lambda) - \mu \sum_{i=1}^m \log(s_i (\sigma - \lambda_i)) \\ &\quad + y^\top (\omega e_p - D\xi) - \mu \sum_{i=1}^p \log(y_i (\omega - (D\xi)_i)) \end{aligned}$$

et où $\tau > 0$ est une constante positive permettant une mise à l'échelle des deux fonctions $\varphi_\mu^{\sigma, \omega}$ et $\mathcal{V}_\mu^{\sigma, \omega}$. Pour simplifier les notations, on note par z le quintuplet (x, s, y, λ, ξ) et par \mathcal{Z} le domaine de $\psi_\mu^{\sigma, \omega}$:

$$\begin{aligned} \mathcal{Z} := \{ &z = (x, s, y, \lambda, \xi) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^p : \\ &(s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) > 0 \text{ et } Ax + Dy = b \}. \end{aligned}$$

Les dérivées suivantes de $\mathcal{V}_\mu^{\sigma,\omega}$ seront utiles par la suite :

$$\begin{aligned} \nabla \mathcal{V}_\mu^{\sigma,\omega}(z) = & \begin{pmatrix} -\nabla c (\lambda - \mu(S-C)^{-1} e_m) \\ \lambda - \mu(S-C)^{-1} e_m \\ 0 \\ (s-c) - \mu\Lambda^{-1} e_m \\ 0 \end{pmatrix} \\ & + \begin{pmatrix} 0 \\ (\sigma e_m - \lambda) - \mu S^{-1} e_m \\ 0 \\ -s + \mu(\sigma I - \Lambda)^{-1} e_m \\ 0 \end{pmatrix} \\ & + \begin{pmatrix} 0 \\ 0 \\ (\omega e_p - D\xi) - \mu Y^{-1} e_p \\ 0 \\ -Dy + \mu D(\omega I - D\xi)^{-1} e_p \end{pmatrix} \end{aligned} \quad (3.24)$$

et

$$\begin{aligned} \nabla^2 \mathcal{V}_\mu^{\sigma,\omega}(z) = & \begin{pmatrix} \nabla_{xx}^2 \mathcal{V}_\mu^{\sigma,\omega}(z) & -\mu \nabla c (S-C)^{-2} & 0 & -\nabla c & 0 \\ -\mu (S-C)^{-2} \nabla c^\top & \mu (S-C)^{-2} & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\nabla c^\top & I & 0 & \mu \Lambda^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ & + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu S^{-2} & 0 & -I & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -I & 0 & \mu(\sigma I - \Lambda)^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ & + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu Y^{-2} & 0 & -D \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -D & 0 & \mu D^2 (\omega I - D\xi)^{-2} \end{pmatrix}, \end{aligned} \quad (3.25)$$

$$\text{où } \nabla_{xx}^2 \mathcal{V}_\mu^{\sigma,\omega}(z) = \sum_{i=1}^m \left(\frac{\mu}{s_i - c_i} - \lambda_i \right) \nabla_{xx}^2 c_i + \mu \nabla c (S-C)^{-2} \nabla c^\top.$$

Ainsi, en multipliant des deux côtés par $(u, v, t, q, r) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^p$, on obtient

$$\begin{aligned}
 & \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{t} \\ \mathbf{q} \\ \mathbf{r} \end{pmatrix}^\top \nabla^2 \mathcal{V}_\mu^{\sigma, \omega}(\mathbf{z}) \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{t} \\ \mathbf{q} \\ \mathbf{r} \end{pmatrix} \\
 &= \mathbf{u}^\top \left(\sum_{i=1}^m \left(\frac{\mu}{s_i - c_i} - \lambda_i \right) \nabla_{xx}^2 c_i \right) \mathbf{u} \\
 & \quad + \mu \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}^\top \begin{pmatrix} \nabla c(S-C)^{-2} \nabla c^\top & -\nabla c(S-C)^{-2} \\ -(S-C)^{-2} \nabla c^\top & (S-C)^{-2} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \\
 & \quad + \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{q} \end{pmatrix}^\top \begin{pmatrix} 0 & 0 & -\nabla c \\ 0 & 0 & \mathbf{I} \\ -\nabla c^\top & \mathbf{I} & \mu \Lambda^2 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{q} \end{pmatrix} \\
 & \quad + \begin{pmatrix} \mathbf{v} \\ \mathbf{q} \end{pmatrix}^\top \begin{pmatrix} \mu S^{-2} & -\mathbf{I} \\ -\mathbf{I} & \mu(\sigma \mathbf{I} - \Lambda)^2 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{q} \end{pmatrix} \\
 & \quad + \begin{pmatrix} \mathbf{t} \\ \mathbf{r} \end{pmatrix}^\top \begin{pmatrix} \mu Y^{-2} & -\mathbf{D} \\ -\mathbf{D} & \mu D^2(\omega \mathbf{I} - D\Xi)^{-2} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{t} \\ \mathbf{r} \end{pmatrix} \\
 &= \mathbf{u}^\top \left(\sum_{i=1}^m \left(\frac{\mu}{s_i - c_i} - \lambda_i \right) \nabla_{xx}^2 c_i \right) \mathbf{u} \\
 & \quad + \mu \| (S-C)^{-1} (\nabla c^\top \mathbf{u} - \mathbf{v}) \|^2 \\
 & \quad + \mu \| \Lambda^{-1} \mathbf{q} \|^2 - 2 \mathbf{q}^\top (\nabla c^\top \mathbf{u} - \mathbf{v}) \\
 & \quad + \mu \| S^{-1} \mathbf{v} \|^2 + \mu \| (\sigma \mathbf{I} - \Lambda)^{-1} \mathbf{q} \|^2 - 2 \mathbf{v}^\top \mathbf{q} \\
 & \quad + \mu \| Y^{-1} \mathbf{t} \|^2 + \mu \| D(\omega \mathbf{I} - D\Xi)^{-1} \mathbf{r} \|^2 - 2 \mathbf{t}^\top D \mathbf{r}.
 \end{aligned} \tag{3.26}$$

La proposition suivante montre que la solution $(\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu)$ de $\mathcal{P}_\mu^{\sigma, \omega}$ et les multiplicateurs $(\hat{\lambda}_\mu, \hat{\xi}_\mu)$ de la proposition 3.5.2 est la solution unique du problème $(\mathcal{PD})_\mu^{\sigma, \omega}$.

Proposition 3.7.1 *Supposons l'hypothèse H_1 satisfaite. Alors, $(\mathcal{PD})_\mu^{\sigma, \omega}$ a pour unique solution le quintuplet $\hat{\mathbf{z}}_\mu := (\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu, \hat{\lambda}_\mu, \hat{\xi}_\mu)$ donné par la proposition 3.5.2.*

Preuve. Montrons tout d'abord que le point $\hat{\mathbf{z}}_\mu$ est une solution du problème $(\mathcal{PD})_\mu^{\sigma, \omega}$. La problème barrière $\varphi_\mu^{\sigma, \omega}$ a pour unique solution le triplet $(\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu)$. On a donc l'inégalité suivante

$$\varphi_\mu^{\sigma, \omega}(\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu) \leq \varphi_\mu^{\sigma, \omega}(x, s, y) \tag{3.27}$$

pour tout triplet (x, s, y) tel que $(s - c(x), s, y) > 0$ et $Ax + Dy = b$. Il est facile de vérifier que la fonction $y \rightarrow y - \mu \log y$ atteint son minimum au point $y = \mu$. On sait d'après la proposition 3.5.2 que le quintuplet $(\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu, \hat{\lambda}_\mu, \hat{\xi}_\mu)$ vérifie les égalités suivantes

$$\begin{cases} (\hat{S}_\mu - C(\hat{x}_\mu)) \hat{\lambda}_\mu = \mu e_m, \\ \hat{S}_\mu (\sigma e_m - \hat{\lambda}_\mu) = \mu e_m, \\ \hat{Y}_\mu (\omega e_p - D \hat{\xi}_\mu) = \mu e_p. \end{cases},$$

où $\hat{S}_\mu := \text{diag}((\hat{s}_\mu)_1, \dots, (\hat{s}_\mu)_m)$ et $\hat{Y}_\mu := \text{diag}((\hat{y}_\mu)_1, \dots, (\hat{y}_\mu)_p)$. On a donc

$$\mathcal{V}_\mu^{\sigma, \omega}(\hat{z}_\mu) \leq \mathcal{V}_\mu^{\sigma, \omega}(z), \text{ pour tout } z \in Z. \quad (3.28)$$

En sommant les deux inégalités (3.27) et (3.28) et par positivité du scalaire τ , on obtient

$$\psi_\mu^{\sigma, \omega}(\hat{z}_\mu) \leq \psi_\mu^{\sigma, \omega}(z), \text{ pour tout } z \in Z.$$

On a donc montré que \hat{z}_μ est une solution de $(\mathcal{PD})_\mu^{\sigma, \omega}$.

Montrons maintenant que $(\mathcal{PD})_\mu^{\sigma, \omega}$ a une solution unique. D'après la définition (3.23) de $\psi_\mu^{\sigma, \omega}$, les dérivées (3.10) et (3.24) et le fait que $\tau > 0$, tout point z solution de $(\mathcal{PD})_\mu^{\sigma, \omega}$ doit vérifier les conditions d'optimalité suivantes obtenues après simplifications et regroupement de certains termes,

$$\nabla f + \mu \nabla c(S-C)^{-1}e_m + \tau \nabla c \left(\mu(S-C)^{-1}e_m - \lambda \right) - A^\top \alpha = 0 \quad (3.29)$$

$$\sigma e_m - \mu(S-C)^{-1}e_m - \mu S^{-1}e_m = 0 \quad (3.30)$$

$$(\omega e_p - D\alpha) - \mu Y^{-1}e_p + \tau(\omega e_p - D\xi - \mu Y^{-1}e_p) = 0 \quad (3.31)$$

$$(s - c) - \mu \Lambda^{-1}e_m - s + \mu(\sigma I - \Lambda)^{-1}e_m = 0 \quad (3.32)$$

$$-Dy + \mu D(\omega I - D\xi)^{-1}e_p = 0 \quad (3.33)$$

$$Ax + Dy = b, \quad (3.34)$$

où α est le multiplicateur associé à la contrainte d'égalité linéaire. L'égalité (3.30) peut s'écrire

$$\begin{aligned} & \sigma e_m - \mu(S-C)^{-1}e_m - \mu S^{-1}e_m \\ &= \sigma e_m - \lambda - \mu S^{-1}e_m + \lambda - \mu(S-C)^{-1}e_m \\ &= S^{-1}(S(\sigma e_m - \lambda) - \mu e_m) + (S-C)^{-1}(\Lambda(s - c) - \mu e_m) \\ &= S^{-1}(\sigma I - \Lambda)(s - \mu(\sigma I - \Lambda)^{-1}e_m) \\ & \quad + (S-C)^{-1}\Lambda((s - c) - \mu \Lambda^{-1}e_m). \end{aligned} \quad (3.35)$$

Ainsi en notant $u = s - \mu(\sigma I - \Lambda)^{-1}e_m$, $v = (s - c) - \mu \Lambda^{-1}e_m$ et en considérant les égalités ci-dessus et (3.32), on obtient le système suivant

$$\begin{cases} S^{-1}(\sigma I - \Lambda)u + (S-C)^{-1}\Lambda v = 0 \\ v - u = 0 \end{cases}$$

Par positivité des variables $(s, s - c, \lambda, \sigma e_m - \lambda)$, la solution du système ci-dessus est donc $(u, v) = 0$. Cela entraîne donc, par définition de u et de v , qu'on a

$$\begin{cases} S(\sigma e_m - \lambda) = \mu e_m \\ (S - C)\lambda = \mu e_m. \end{cases}$$

La dernière égalité (3.33) peut s'écrire sous la forme suivante

$$D(\omega I - D\xi)^{-1}(Y(\omega e_p - D\xi) - \mu e_p) = 0.$$

On sait par construction de D que, pour tout $i \in \{1, \dots, p\}$, D_{ii} est non nul. Cela entraîne donc que

$$Y(\omega e_p - D\xi) = \mu e_p.$$

En multipliant (3.31) par Y et en simplifiant à l'aide de l'égalité ci-dessus, on obtient

$$Y(\omega e_p - D\alpha) = \mu e_p.$$

Avec les deux égalités ci-dessus, on obtient

$$D(\alpha - \xi) = 0,$$

et par non nullité des composantes de la diagonale de D on en déduit que $\alpha = \xi$. Enfin, en substituant $\mu(S-C)^{-1}e_m$ par λ et α par ξ dans (3.29), on obtient $\nabla f + \nabla c\lambda - A^T\xi = 0$. Cela montre que le quintuplet (x, s, y, λ, ξ) est solution du système (3.8). Par la proposition 3.5.2, on obtient $(x, s, y, \lambda, \xi) = (\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu, \hat{\lambda}_\mu, \hat{\xi}_\mu)$. \square

La proposition suivante vérifie que la direction calculée en résolvant le système de Newton est toujours une direction de descente de la fonction $\psi_\mu^{\sigma, \omega}$.

Proposition 3.7.2 *Supposons que M soit une matrice symétrique définie positive. Soit $d = (d^x, d^s, d^y, d^\lambda, d^\xi)$ l'unique solution de (3.15). Alors, pour tout $z \in \mathcal{Z}$,*

$$\begin{aligned} \nabla \psi_\mu^{\sigma, \omega}(z)^T d &= -(d^x)^T M d^x - \|(\sigma I - \Lambda)^{1/2} S^{-1/2} d^s\|^2 \\ &\quad - \|\Lambda^{1/2} (S - C)^{-1/2} (\nabla c^T d^x - d^s)\|^2 \\ &\quad - \|Y^{-1/2} (\omega I - D\xi)^{1/2} d^y\|^2 \\ &\quad - \tau \|(S - C)^{-1/2} \Lambda^{-1/2} ((S - C)\lambda - \mu e_m)\|^2 \\ &\quad - \tau \|S^{-1/2} (\sigma I - \Lambda)^{-1/2} (S(\sigma e_m - \lambda) - \mu e_m)\|^2 \\ &\quad - \tau \|Y^{-1/2} (\omega I - D\xi)^{-1/2} (Y(\omega e_p - D\xi) - \mu e_p)\|^2, \end{aligned} \quad (3.36)$$

et donc, si $z \neq \hat{z}_\mu$, d est une direction de descente de $\psi_\mu^{\sigma, \omega}$.

Preuve. Par la définition (3.23) de $\psi_\mu^{\sigma, \omega}$, on a

$$\nabla \psi_\mu^{\sigma, \omega}(z)^T d = \nabla \varphi_\mu^{\sigma, \omega}(x, s, y)^T \begin{pmatrix} d^x \\ d^s \\ d^y \end{pmatrix} + \tau \nabla \mathcal{V}_\mu^{\sigma, \omega}(z)^T d.$$

Les quatre premiers termes de (3.7.2) proviennent de la dérivée du terme barrière et sont donnés par (3.17). Calculons les trois derniers termes en utilisant

les formules (3.24) et les trois dernières équations de (3.15) :

$$\begin{aligned}
& \nabla \mathcal{V}_\mu^{\sigma, \omega}(z)^\top d \\
&= (-\lambda + \mu(S-C)^{-1}e_m)^\top \nabla c^\top d^x \\
&\quad + (\lambda - \mu(S-C)^{-1}e_m)^\top d^s + (s - c - \mu\Lambda^{-1}e_m)^\top d^\lambda \\
&\quad + (\sigma e_m - \lambda - \mu S^{-1}e_m)^\top d^s + (-s + \mu(\sigma I - \Lambda)^{-1}e_m)^\top d^\lambda \\
&\quad + ((\omega e_p - D\xi) - \mu Y^{-1}e_p)^\top d^y + (-Dy + \mu D(\omega I - D\xi)^{-1}e_p)^\top d^\xi \\
&= (\lambda - \mu(S-C)^{-1}e_m)^\top (-\nabla c^\top d^x + d^s) + (s - c - \mu\Lambda^{-1}e_m)^\top d^\lambda \\
&\quad + (\sigma e_m - \lambda - \mu S^{-1}e_m)^\top d^s + (-s + \mu(\sigma I - \Lambda)^{-1}e_m)^\top d^\lambda \\
&\quad + ((\omega e_p - D\xi) - \mu Y^{-1}e_p)^\top d^y + (-Dy + \mu D(\omega I - D\xi)^{-1}e_p)^\top d^\xi \\
&= (e_m - \mu(S-C)^{-1}\Lambda^{-1}e_m)^\top (-\Lambda \nabla c^\top d^x + \Lambda d^s) \\
&\quad + (e_m - \mu\Lambda^{-1}(S-C)^{-1}e_m)^\top ((S-C)^{-1}d^\lambda) \\
&\quad + (e_m - \mu S^{-1}(\sigma I - \Lambda)^{-1}e_m)^\top ((\sigma I - \Lambda)^{-1}d^s) \\
&\quad - (e_m - \mu(\sigma I - \Lambda)^{-1}S^{-1}e_m)^\top (S^{-1}d^\lambda) \\
&\quad + (e_p - \mu(\omega I - D\xi)^{-1}Y^{-1}e_p)^\top (\omega I - D\xi)^{-1}d^y \\
&\quad + (-e_p + \mu Y^{-1}(\omega I - D\xi)^{-1}e_p)^\top Y D d^\xi \\
&= (e_m - \mu(S-C)^{-1}\Lambda^{-1}e_m)^\top (-\Lambda \nabla c^\top d^x + \Lambda d^s + (S-C)d^\lambda) \\
&\quad + (e_m - \mu S^{-1}(\sigma I - \Lambda)^{-1}e_m)^\top ((\sigma I - \Lambda)d^s - Sd^\lambda) \\
&\quad + (e_p - \mu(\omega I - D\xi)^{-1}Y^{-1}e_p)^\top ((\omega I - D\xi)^{-1}d^y - Y D d^\xi)
\end{aligned}$$

et donc,

$$\begin{aligned}
& \nabla \mathcal{V}_\mu^{\sigma, \omega}(z)^\top d \\
&= -(e_m - \mu(S-C)^{-1}\Lambda^{-1}e_m)^\top ((S-C)\lambda - \mu e_m) \\
&\quad - (e_m - \mu S^{-1}(\sigma I - \Lambda)^{-1}e_m)^\top (S(\sigma e_m - \lambda) - \mu e_m) \\
&\quad - (e_p - \mu(\omega I - D\xi)^{-1}Y^{-1}e_p)^\top (Y(\omega e_p - D\xi) - \mu e_p) \\
&= -((S-C)\lambda - \mu e_m)^\top (S-C)^{-1}\Lambda^{-1}((S-C)\lambda - \mu e_m) \\
&\quad - (S(\sigma e_m - \lambda) - \mu e_m)^\top S^{-1}(\sigma I - \Lambda)^{-1}(S(\sigma e_m - \lambda) - \mu e_m) \\
&\quad - (Y(\omega e_p - D\xi) - \mu e_p)^\top (\omega I - D\xi)^{-1}Y^{-1}(Y(\omega e_p - D\xi) - \mu e_p) \\
&= -\|(S-C)^{-1/2}\Lambda^{-1/2}((S-C)\lambda - \mu e_m)\|^2 \\
&\quad - \|S^{-1/2}(\sigma I - \Lambda)^{-1/2}(S(\sigma e_m - \lambda) - \mu e_m)\|^2 \\
&\quad - \|Y^{-1/2}(\omega I - D\xi)^{-1/2}(Y(\omega e_p - D\xi) - \mu e_p)\|^2
\end{aligned}$$

Et donc $\nabla \psi_\mu^{\sigma, \omega}(z)^\top d \leq 0$. Si ce produit scalaire s'annule, alors

$$\nabla \varphi_\mu^{\sigma, \omega}(x, s, t)^\top \begin{pmatrix} d^x \\ d^s \\ d^y \end{pmatrix} = 0 \quad \text{et} \quad \nabla \mathcal{V}_\mu^{\sigma, \omega}(z)^\top d = 0.$$

On obtient avec la première égalité et la proposition 3.6.2 que

$$(x, s, y) = (\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu).$$

La seconde égalité implique alors que

$$\begin{cases} \hat{S}_\mu(\sigma e_m - \lambda) = \mu e_m \\ (\hat{S}_\mu - C(\hat{x}_\mu))\lambda = \mu e_m \\ \hat{Y}_\mu(\omega e_p - D\xi) = \mu e_p. \end{cases}$$

et donc que $\lambda = \hat{\lambda}_\mu$ et $\xi = \hat{\xi}_\mu$. \square

Le résultat suivant montre que les ensembles de niveau de la fonction de mérite $\psi_\mu^{\sigma, \omega}$ sont compacts et que les variables $(s, y, \lambda, s - c, \sigma e_m - \lambda, \omega e_p - D\xi)$ sont bornés par une constante positive sur tout ensemble de niveau de $\psi_\mu^{\sigma, \omega}$.

Proposition 3.7.3 *Supposons l'hypothèse H_1 satisfaite. Alors, pour tout $r \in \mathbb{R}$, l'ensemble de niveau $\mathcal{L}_r := \{z \in \mathcal{Z} : \psi_\mu^{\sigma, \omega}(z) \leq r\}$ est compact et il existe deux nombres positifs K_1 et K_2 tels que*

$$K_1 \leq (s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) \leq K_2 \quad \text{pour tout } z \in \mathcal{L}_r.$$

Preuve. Montrons tout d'abord que les variables $(s - c(x), s, y)$ sont bornées pour tout $z = (x, s, y, \lambda, \xi) \in \mathcal{L}_r$. On remarque que la fonction $\mathcal{V}_\mu^{\sigma, \omega}$ est bornée inférieurement par $(2m + p)\mu(1 - \log \mu)$ sur son domaine. On a donc, pour tout $z = (x, s, y, \lambda, \xi) \in \mathcal{L}_r$,

$$\varphi_\mu^{\sigma, \omega}(x, s, y) \leq r - (2m + p)\mu(1 - \log \mu).$$

Notons $K'_1 := r - (2m + p)\mu(1 - \log \mu)$ cette constante et \mathcal{L}' l'ensemble de niveau défini par

$$\mathcal{L}' = \{(x, s, y) \in \text{dom}(\varphi_\mu^{\sigma, \omega}) : \varphi_\mu^{\sigma, \omega}(x, s, y) \leq K'_1\}$$

On a l'inclusion suivante

$$\mathcal{L}'_r = \{(x, s, y) : z = (x, s, y, \lambda, \xi) \in \mathcal{L}_r\} \subset \mathcal{L}'. \quad (3.37)$$

D'après l'hypothèse H_1 et la proposition 3.5.1, l'ensemble \mathcal{L}' est compact sur le domaine de $\varphi_\mu^{\sigma, \omega}$. Par continuité de l'application $(x, s) \rightarrow s - c(x)$, l'ensemble $\{(s - c(x), s, y) : (x, s, y) \in \mathcal{L}'\}$ est un ensemble compact et donc borné. Il existe donc une constante K'_2 telle que

$$0 \leq \|(s - c(x), s, y)\| \leq K'_2, \quad \text{pour tout } z \in \mathcal{L}',$$

et, avec l'inclusion (3.37),

$$0 \leq \|(s - c(x), s, y)\| \leq K'_2, \quad \text{pour tout } z \in \mathcal{L}'_r.$$

Par bornitude de l'ensemble \mathcal{L}' et avec (3.37), l'ensemble \mathcal{L}'_r est borné. Par continuité de la fonction f , on a le fait que la fonction $(x, s, y) \rightarrow f(x) + \sigma e^\top s +$

$\omega e^\top y$ est bornée inférieurement pour tout $(x, s, y) \in \mathcal{L}'_r$. Notons K'_3 cette borne inférieure. Comme $\varphi_\mu^{\sigma, \omega}(x, s, y) \leq K'_1$ pour tout $(x, y, y) \in \mathcal{L}'_r$, on a

$$-\sum_{i=1}^m \log(s_i - c_i(x)) - \sum_{i=1}^m \log(s_i) - \sum_{i=1}^p \log(y_i) \leq (K'_1 - K'_3)/\mu.$$

On en déduit alors, par majoration des termes logarithmiques de $\varphi_\mu^{\sigma, \omega}$, le fait que les variables $(s - c(x), s, y)$ ne peuvent tendre vers 0. Il existe donc une constante K'_4 strictement positive telle que

$$K'_4 \leq \|(s - c(x), s, y)\| \leq K'_2, \text{ pour tout } z \in \mathcal{L}_r. \quad (3.38)$$

Montrons maintenant que les variables $(\lambda, \sigma e_m - \lambda, \omega e_p - D\xi)$ sont bornées pour tout $z \in \mathcal{L}_r$. Notons $(\varphi_\mu^{\sigma, \omega})^*$ la valeur minimale de $\varphi_\mu^{\sigma, \omega}$ sur l'ensemble compact \mathcal{L}' . Alors,

$$\mathcal{V}_\mu^{\sigma, \omega}(z) \leq r - (\varphi_\mu^{\sigma, \omega})^* \quad \text{pour tout } z \in \mathcal{L}_r.$$

Comme le terme $t - \mu \log t$ tend vers $+\infty$ quand t tend vers 0 ou $+\infty$, on en déduit qu'il existe deux nombres positifs K'_4 et K'_5 tels que

$$K'_4 \leq (\lambda(s - c(x)), s(\sigma e_m - \lambda), y(\omega e_p - D\xi)) \leq K'_5 \quad \forall z \in \mathcal{L}_r.$$

Donc qu'il existe deux constantes strictement positives K'_6 et K'_7 telles que

$$K'_6 \leq (\lambda, (\sigma e_m - \lambda), \omega e_p - D\xi) \leq K'_7 \quad \forall z \in \mathcal{L}_r. \quad (3.39)$$

En considérant (3.38) et (3.39), on a montré qu'il existe deux constantes positives non nulles K_1 et K_2 telles que

$$K_1 \leq (s - c(x), s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi) \leq K_2 \quad \text{pour tout } z \in \mathcal{L}_r.$$

Par bornitude des variables $(\omega e_p - D\xi)$, on en déduit celle des variables ξ . Ainsi \mathcal{L}_r est inclus dans un ensemble borné. Par continuité de la fonction $\psi_\mu^{\sigma, \omega}$, ses ensembles de niveau (et donc \mathcal{L}_r) sont fermés. \mathcal{L}_r étant un ensemble fermé et borné est compact. \square

Bien que la fonction $\psi_\mu^{\sigma, \omega}$ ne soit pas convexe, elle est fortement convexe dans un voisinage de \hat{z}_μ , et plus généralement dans un voisinage de tout point satisfaisant les conditions de centralité.

Proposition 3.7.4 *Supposons l'hypothèse H_1 satisfaite. Si $z \in \mathcal{Z}$ satisfait les conditions de centralité $(S - C)\lambda = \mu e_m$, $S(\sigma e_m - \lambda) = \mu e_m$ et $Y(\omega e_p - D\xi) = \mu e_p$, alors la fonction de mérite $\psi_\mu^{\sigma, \omega}$ est localement fortement convexe en z .*

Preuve. Soit $z \in \mathcal{Z}$ un point satisfaisant les conditions de centralité. Montrons que la matrice hessienne de $\psi_\mu^{\sigma, \omega}$ est définie positive en z . Par (3.11), (3.25) et

comme z satisfait $(S-C)\lambda = \mu e_m$, $S(\sigma e_m - \lambda) = \mu e_m$ et $Y(\omega e_p - D\xi) = \mu e_p$, le hessien de $\psi_\mu^{\sigma, \omega}$ est égal à

$$\begin{aligned} \nabla^2 \psi_\mu^{\sigma, \omega}(z) = & \begin{pmatrix} \nabla^2 \varphi_\mu^{\sigma, \omega}(x, s, t) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ & + \tau \begin{pmatrix} \mu \nabla c (S-C)^{-2} \nabla c^\top & -\mu \nabla c (S-C)^{-2} & 0 & -\nabla c & 0 \\ -\mu (S-C)^{-2} \nabla c^\top & \mu (S-C)^{-2} & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\nabla c^\top & I & 0 & \frac{1}{\mu} (S-C)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\ & + \tau \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mu S^{-2} & 0 & -I & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -I & 0 & \frac{1}{\mu} S^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \tau \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu Y^{-2} & 0 & -D \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -D & 0 & \frac{1}{\mu} D^2 Y^2 \end{pmatrix}. \end{aligned}$$

En multipliant des deux côtés par $(u, v, t, q, r) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^p$ et avec (3.13), on obtient, pour tout $\xi \in \mathbb{R}^p$,

$$\begin{aligned} & \begin{pmatrix} u \\ v \\ t \\ q \\ r \end{pmatrix}^\top \nabla^2 \psi_\mu^{\sigma, \omega}(z) \begin{pmatrix} u \\ v \\ t \\ q \\ r \end{pmatrix} \\ & = u^\top \nabla_{xx}^2 \ell(x, \lambda, \xi) u + \mu \| (S-C)^{-1} (\nabla c^\top u - v) \|^2 \\ & \quad + \|\mu^{1/2} S^{-1} v\|^2 + \|\mu^{1/2} Y^{-1} t\|^2 \\ & \quad + \tau \begin{pmatrix} u \\ v \\ q \end{pmatrix}^\top \begin{pmatrix} \mu \nabla c (S-C)^{-2} \nabla c^\top & -\mu \nabla c (S-C)^{-2} & -\nabla c \\ -\mu (S-C)^{-2} \nabla c^\top & \mu (S-C)^{-2} & I \\ -\nabla c^\top & I & \frac{1}{\mu} (S-C)^2 \end{pmatrix} \begin{pmatrix} u \\ v \\ q \end{pmatrix} \\ & \quad + \tau \begin{pmatrix} v \\ q \end{pmatrix}^\top \begin{pmatrix} \mu S^{-2} & -I \\ -I & \frac{1}{\mu} S^2 \end{pmatrix} \begin{pmatrix} v \\ q \end{pmatrix} \\ & \quad + \tau \begin{pmatrix} t \\ r \end{pmatrix}^\top \begin{pmatrix} \mu Y^{-2} & -D \\ -D & \frac{1}{\mu} D^2 Y^2 \end{pmatrix} \begin{pmatrix} t \\ r \end{pmatrix} \\ & = u^\top \nabla_{xx}^2 \ell(x, \lambda, \xi) u + \|\mu^{1/2} (S-C)^{-1} (\nabla c^\top u - v)\|^2 \\ & \quad + \|\mu^{1/2} S^{-1} v\|^2 + \|\mu^{1/2} Y^{-1} t\|^2 \\ & \quad + \tau \|\mu^{1/2} (S-C)^{-1} (\nabla c^\top u - v) - \mu^{-1/2} (S-C) q\|^2 \\ & \quad + \tau \|\mu^{1/2} S^{-1} v - \mu^{-1/2} S q\|^2 \\ & \quad + \tau \|\mu^{1/2} Y^{-1} t - \mu^{-1/2} D Y r\|^2, \end{aligned}$$

ce qui est non négatif. Si cette somme s'annule, la définie positivité du hessien du lagrangien entraîne que $u = 0$ puis par la suite que $v = q = 0$ et $y = r = 0$. \square

Le résultat suivant montre que, sur un ensemble de niveau de $\psi_\mu^{\sigma,\omega}$, la distance séparant un point z de la solution \hat{z}_μ est bornée par la différence de valeurs prises par la fonction de mérite en ces points. Cette différence est elle-même majorée par la norme du gradient du lagrangien en ce point z sommée avec les normes des conditions de complémentarité perturbées. Ce résultat est une conséquence de la forte convexité de $\psi_\mu^{\sigma,\omega}$ en \hat{z}_μ . Cet encadrement est différent de celui démontré dans [2, Lemme A.4] car le problème barrière n'est plus sans contrainte.

Proposition 3.7.5 *Supposons l'hypothèse H_1 satisfaite. Alors, pour tout $r \in \mathbb{R}$ il existe un nombre $\alpha > 0$ et un voisinage ouvert $\mathcal{N} \subset \mathcal{L}_r$ de \hat{z}_μ tel que pour tout $z \in \mathcal{N}$*

$$\begin{aligned} \alpha \|z - \hat{z}_\mu\|^2 &\leq \psi_\mu^{\sigma,\omega}(z) - \psi_\mu^{\sigma,\omega}(\hat{z}_\mu) \\ &\leq \frac{1}{\alpha} (\|\nabla \ell(x, s, y, \lambda, \xi)\|^2 \\ &\quad + \|(S - C)\lambda - \mu e_m\|^2 \\ &\quad + \|S(\sigma e_m - \lambda) - \mu e_m\|^2 \\ &\quad + \|Y(\omega e_p - D\xi) - \mu e_p\|^2). \end{aligned} \quad (3.40)$$

Preuve. Montrons tout d'abord l'inégalité de gauche dans (3.40). Comme le point \hat{z}_μ est dans \mathcal{L}_r et satisfait les conditions de centralité, on sait, d'après la proposition précédente, que la fonction $\psi_\mu^{\sigma,\omega}$ est localement fortement convexe en ce point. On a donc, pour tout $z \in \mathcal{N}$, $\mathcal{N} \subset \mathcal{L}_r$ voisinage ouvert de \hat{z}_μ ,

$$\psi_\mu^{\sigma,\omega}(z) - \psi_\mu^{\sigma,\omega}(\hat{z}_\mu) \geq \nabla \psi_\mu^{\sigma,\omega}(\hat{z}_\mu)^\top (z - \hat{z}_\mu) + \frac{\kappa}{2} \|z - \hat{z}_\mu\|^2, \quad (3.41)$$

où κ est le module de forte convexité de la fonction $\psi_\mu^{\sigma,\omega}$. D'après les définitions des gradients de $\varphi_\mu^{\sigma,\omega}$ (3.10) et de $\mathcal{V}_\mu^{\sigma,\omega}$ (3.24), le système (3.8) dont \hat{z}_μ est solution, et les conditions de centralité satisfaites en \hat{z}_μ , on a aussi

$$\begin{aligned} &\nabla \psi_\mu^{\sigma,\omega}(\hat{z}_\mu)^\top (z - \hat{z}_\mu) \\ &= \nabla \varphi_\mu^{\sigma,\omega}(\hat{x}_\mu, \hat{s}_\mu, \hat{y}_\mu)^\top (x - \hat{x}_\mu, s - \hat{s}_\mu, y - \hat{y}_\mu) + \tau \nabla \mathcal{V}_\mu^{\sigma,\omega}(\hat{z}_\mu)^\top (z - \hat{z}_\mu) \\ &= \begin{pmatrix} A^\top \hat{\xi}_\mu \\ 0 \\ D \hat{\xi}_\mu \end{pmatrix}^\top (x - \hat{x}_\mu, s - \hat{s}_\mu, y - \hat{y}_\mu) + \tau \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^\top (z - \hat{z}_\mu) \\ &= (\hat{\xi}_\mu)^\top ((Ax + Dy) - (A\hat{x}_\mu + D\hat{y}_\mu)) \\ &= (\hat{\xi}_\mu)^\top (b - b) \\ &= 0 \end{aligned}$$

Avec (3.41), on obtient l'inégalité de gauche dans (3.40) en posant $\alpha = \frac{\kappa}{2}$.

Montrons maintenant l'inégalité de droite dans (3.40). Remarquons tout d'abord que

$$\begin{aligned}
 \nabla \varphi_{\mu}^{\sigma, \omega}(x, s, y) &= \begin{pmatrix} \nabla f(x) + \mu \nabla c(S - C)^{-1} e_m \\ \sigma e_m - \mu(S - C)^{-1} e_m - \mu S^{-1} e_m \\ \omega e_p - \mu Y^{-1} e_p \end{pmatrix} \\
 &= \begin{pmatrix} \nabla f(x) + \nabla c \lambda - A^{\top} \xi + \nabla c(S - C)^{-1}((S - C)\lambda - \mu e_m) + A^{\top} \xi \\ (S - C)^{-1}((S - C)\lambda - \mu e_m) + S^{-1}(S(\sigma e_m - \lambda) - \mu e_m) \\ Y^{-1}(Y(\omega e_p - D\xi) - \mu e_p) + D\xi \end{pmatrix} \\
 &= \begin{pmatrix} \nabla \ell(x, s, y, \lambda, \xi) + \nabla c(S - C)^{-1}((S - C)\lambda - \mu e_m) \\ (S - C)^{-1}((S - C)\lambda - \mu e_m) + S^{-1}(S(\sigma e_m - \lambda) - \mu e_m) \\ Y^{-1}(Y(\omega e_p - D\xi) - \mu e_p) \end{pmatrix} + \begin{pmatrix} A^{\top} \xi \\ 0 \\ D\xi \end{pmatrix}.
 \end{aligned}$$

On peut écrire le gradient de $\psi_{\mu}^{\sigma, \omega}$ comme suit

$$\nabla \psi_{\mu}^{\sigma, \omega}(z) = \begin{pmatrix} \nabla \varphi_{\mu}^{\sigma, \omega}(x, s, y) \\ 0 \\ 0 \end{pmatrix} + \tau \nabla \mathcal{V}_{\mu}^{\sigma, \omega}(z) = g + \begin{pmatrix} A^{\top} \xi \\ 0 \\ D\xi \\ 0 \\ 0 \end{pmatrix}, \quad (3.42)$$

en notant

$$g = \begin{pmatrix} \nabla \ell(x, s, y, \lambda, \xi) + (1 + \tau) \nabla c(S - C)^{-1}((S - C)\lambda - \mu e_m) \\ (1 + \tau)(S - C)^{-1}((S - C)\lambda - \mu e_m) + S^{-1}(S(\sigma e_m - \lambda) - \mu e_m) \\ (1 + \tau)Y^{-1}(Y(\omega e_p - D\xi) - \mu e_p) \\ \tau(\Lambda^{-1}((S - C)\lambda - \mu e_m) - (\sigma I - \Lambda)^{-1}(S(\sigma e_m - \lambda) - \mu e_m)) \\ -\tau D(\omega I - D\xi)^{-1}(Y(\omega e_p - D\xi) - \mu e_p) \end{pmatrix}.$$

D'après la proposition 3.7.4, on a, par convexité de $\psi_{\mu}^{\sigma, \omega}$ sur \mathcal{N} , où $\mathcal{N} \subset \mathcal{L}_{\tau}$ est un voisinage ouvert de \hat{z}_{μ} ,

$$\psi_{\mu}^{\sigma, \omega}(\hat{z}_{\mu}) \geq \psi_{\mu}^{\sigma, \omega}(z) + \nabla \psi_{\mu}^{\sigma, \omega}(z)^{\top}(\hat{z}_{\mu} - z), \quad \forall z \in \mathcal{N}.$$

On a alors en utilisant successivement l'égalité (3.42), l'inégalité de Cauchy-Schwarz et l'inégalité de gauche de (3.40),

$$\begin{aligned}
 \psi_{\mu}^{\sigma, \omega}(z) - \psi_{\mu}^{\sigma, \omega}(\hat{z}_{\mu}) &\leq \nabla \psi_{\mu}^{\sigma, \omega}(z)^{\top}(z - \hat{z}_{\mu}) \\
 &\leq g^{\top}(z - \hat{z}_{\mu}) + \xi^{\top}(A(x - \hat{x}_{\mu}) + D(y - \hat{y}_{\mu})) \\
 &\leq g^{\top}(z - \hat{z}_{\mu}) + \xi^{\top}(\mathbf{b} - \mathbf{b}) \\
 &\leq g^{\top}(z - \hat{z}_{\mu}) \\
 &\leq \|g\| \|z - \hat{z}_{\mu}\| \\
 &\leq \|g\| \left(\frac{\psi_{\mu}^{\sigma, \omega}(z) - \psi_{\mu}^{\sigma, \omega}(\hat{z}_{\mu})}{\alpha} \right)^{1/2}.
 \end{aligned}$$

En élevant au carré cette inégalité et en simplifiant, on obtient

$$\psi_{\mu}^{\sigma, \omega}(z) - \psi_{\mu}^{\sigma, \omega}(\hat{z}_{\mu}) \leq \frac{1}{\alpha} \|g\|^2 \quad (3.43)$$

Majorons pour finir la démonstration la norme de g . En utilisant les bornes de la proposition (3.7.3), en notant K_c une borne de $\|\nabla c\|$ sur \mathcal{L}_τ et K_D une borne de $\|D\|$, on a

$$\begin{aligned}
 \|g\|^2 &\leq \|\nabla\ell(x, s, y, \lambda, \xi) + (1 + \tau)\nabla c(S - C)^{-1}((S - C)\lambda - \mu e_m)\|^2 \\
 &\quad + \|(1 + \tau)(S - C)^{-1}((S - C)\lambda - \mu e_m) + S^{-1}(S(\sigma e_m - \lambda) - \mu e_m)\|^2 \\
 &\quad + \|(1 + \tau)Y^{-1}(Y(\omega e_p - D\xi) - \mu e_p)\|^2 \\
 &\quad + \|\tau(\Lambda^{-1}((S - C)\lambda - \mu e_m) - (\sigma I - \Lambda)^{-1}(S(\sigma e_m - \lambda) - \mu e_m))\|^2 \\
 &\quad + \|\tau D(\omega I - D\xi)^{-1}(Y(\omega e_p - D\xi) - \mu e_p)\|^2 \\
 &\leq \|\nabla\ell(x, s, y, \lambda, \xi)\|^2 + (1 + \tau)^2 K_c^2 K_1^{-2} \|(S - C)\lambda - \mu e_m\|^2 \\
 &\quad + (1 + \tau)^2 K_1^{-2} (\|(S - C)\lambda - \mu e_m\|^2 + \|S(\sigma e_m - \lambda) - \mu e_m\|^2) \\
 &\quad + (1 + \tau)^2 K_1^{-2} \|Y(\omega e_p - D\xi) - \mu e_p\|^2 \\
 &\quad + \tau^2 K_1^{-2} (\|(S - C)\lambda - \mu e_m\| + \|S(\sigma e_m - \lambda) - \mu e_m\|)^2 \\
 &\quad + \tau^2 K_D^2 K_1^{-2} \|Y(\omega e_p - D\xi) - \mu e_p\|^2 \\
 &\leq K'1 (\|\nabla\ell(x, s, y, \lambda, \xi)\|^2 + \|(S - C)\lambda - \mu e_m\|^2 \\
 &\quad + \|S(\sigma e_m - \lambda) - \mu e_m\|^2 + \|Y(\omega e_p - D\xi) - \mu e_p\|^2)
 \end{aligned}$$

En utilisant cette majoration et l'inégalité (3.43), on obtient l'inégalité de droite dans (3.40). \square

3.8 Algorithme $\mathcal{A}_\mu^{\sigma, \omega}$

On peut maintenant établir une itération de l'algorithme résolvant le système perturbé KKT (3.8), avec des paramètres de pénalité $\sigma > 0$ et $\omega > 0$ fixés et avec un paramètre barrière $\mu > 0$. Les constantes $\omega_a \in (0, \frac{1}{2})$ (test d'Armijo), $0 < \beta \leq \beta' < 1$ (coefficients de réduction pour le rebroussement), $\tau > 0$ (facteur de centralisation) sont donnés indépendamment de l'indice de l'itération. Au début de l'itération, on suppose disponibles l'itéré courant $z_k = (x_k, s_k, y_k, \lambda_k, \xi_k) \in \mathcal{Z}$, une matrice définie positive M_k approchant le hessien du lagrangien $\nabla_{xx}^2 \ell(x_k, \lambda_k, \xi_k)$.

ALGORITHME $\mathcal{A}_\mu^{\sigma, \omega}$ pour résoudre (3.8) (une itération, de (z_k, M_k) vers (z_{k+1}, M_{k+1}))

1. Calcul de $d_k := (d_k^x, d_k^s, d_k^y, d_k^\lambda, d_k^\xi)$, unique solution du système linéaire (3.15). Si $d_k = 0$, stop (z_k résout le système (3.8)).
2. Calcul d'un pas $\alpha_k > 0$ par rebroussement :
 - 2.0. Soit $\alpha_k = 1$.
 - 2.1. Tant que $z_k + \alpha_k d_k \notin \mathcal{Z}$, choisir un nouveau pas α_k dans $[\beta \alpha_k, \beta' \alpha_k]$.

2.2. Tant que la condition de décroissance suffisante (ou condition d'Armijo)

$$\psi_\mu^{\sigma,\omega}(z_k + \alpha_k d_k) \leq \psi_\mu^{\sigma,\omega}(z_k) + \omega_\alpha \alpha_k \nabla \psi_\mu^{\sigma,\omega}(z_k)^\top d_k \quad (3.44)$$

n'est pas satisfaite, choisir un nouveau pas α_k dans $[\beta \alpha_k, \beta' \alpha_k]$.

2.3. Soit $z_{k+1} := z_k + \alpha_k d_k$.

3. Mise à jour de M_k par les formules de BFGS

$$M_{k+1} := M_k - \frac{M_k \delta_k \delta_k^\top M_k}{\delta_k^\top M_k \delta_k} + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k},$$

où γ_k et δ_k sont donnés par

$$\delta_k := x_{k+1} - x_k$$

et

$$\gamma_k := \nabla_x \ell(x_{k+1}, \lambda_{k+1}, \xi_{k+1}) - \nabla_x \ell(x_k, \lambda_{k+1}, \xi_{k+1}).$$

3.9 Convergence r-linéaire de l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$

Soit $z_1 = (x_1, s_1, y_1, \lambda_1, \xi_1)$ l'itéré initial de l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$. On définit l'ensemble suivant

$$\mathcal{L}_1 := \{z \in \mathcal{Z} : \psi_\mu^{\sigma,\omega}(z) \leq \psi_\mu^{\sigma,\omega}(z_1)\}.$$

étant donné que la fonction de mérite décroît au cours des itérations, tous les itérés restent dans cet ensemble.

L'analyse de convergence des itérés internes repose sur les deux lemmes suivants. Le premier est une conséquence de la condition de décroissance suffisante (3.44).

Lemme 3.9.1 *Supposons que $\psi_\mu^{\sigma,\omega}$ est $C^{1,1}$ dans un voisinage ouvert convexe \mathcal{N} de l'ensemble de niveau \mathcal{L}_1 . Il existe une constante $K_3 > 0$ telle que pour tout $z \in \mathcal{L}_1$ et pour toute direction de descente d de $\psi_\mu^{\sigma,\omega}$, si le pas α est déterminé par la recherche linéaire de l'étape 2 de l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$, alors une des inégalités suivantes est vérifiée :*

$$\psi_\mu^{\sigma,\omega}(z + \alpha d) \leq \psi_\mu^{\sigma,\omega}(z) - K_3 |\nabla \psi_\mu^{\sigma,\omega}(z)^\top d|, \quad (3.45)$$

$$\psi_\mu^{\sigma,\omega}(z + \alpha d) \leq \psi_\mu^{\sigma,\omega}(z) - K_3 \frac{|\nabla \psi_\mu^{\sigma,\omega}(z)^\top d|^2}{\|d\|^2}. \quad (3.46)$$

Preuve. Si la recherche linéaire est satisfaite avec le pas unité $\alpha = 1$, il suffit de prendre $K_3 = \omega_\alpha$ et on a la première inégalité.

Supposons que le pas α accepté par la recherche linéaire ne soit pas le pas unité. Il existe alors un pas α' non accepté par la recherche linéaire tel que

$$\beta\alpha' \leq \alpha \leq \beta'\alpha'.$$

La non acceptation de ce pas α' a deux raisons possibles : soit $z + \alpha'd \notin \mathcal{Z}$, soit $z + \alpha'd \in \mathcal{Z}$ mais la condition d'Armijo (3.44) n'est pas vérifiée. Supposons que α' vérifie $z + \alpha'd \notin \mathcal{Z}$. Comme \mathcal{Z} est un ensemble ouvert qui contient un voisinage ouvert \mathcal{N} convexe de \mathcal{L}_1 , on a

$$\exists \tilde{\alpha} \in]\alpha, \alpha'[: z + \tilde{\alpha}d \in \mathcal{N} \setminus \mathcal{L}_1.$$

On a alors les inégalités suivantes

$$\psi_{\mu}^{\sigma,\omega}(z + \tilde{\alpha}d) > \psi_{\mu}^{\sigma,\omega}(z_1) > \psi_{\mu}^{\sigma,\omega}(z) > \psi_{\mu}^{\sigma,\omega}(z) + \omega_a \tilde{\alpha} \nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d.$$

La première inégalité vient du fait que $z + \tilde{\alpha}d \notin \mathcal{L}_1$, la seconde, du fait que $z \in \mathcal{L}_1$ et la troisième du fait que $\nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d \leq 0$. Ainsi en notant $\alpha' = \tilde{\alpha}$, on est ramené à la seconde raison du refus du pas. Supposons donc maintenant que le pas α soit accepté pour la recherche linéaire et qu'il existe un pas α' tel que : $\alpha \in [\beta\alpha', \beta'\alpha]$, $z + \alpha'd \in \mathcal{Z}$ et

$$\psi_{\mu}^{\sigma,\omega}(z + \alpha'd) > \psi_{\mu}^{\sigma,\omega}(z) + \omega_a \alpha' \nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d.$$

On a alors

$$\psi_{\mu}^{\sigma,\omega}(z + \alpha'd) - \psi_{\mu}^{\sigma,\omega}(z) > \omega_a \alpha' \nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d.$$

Donc,

$$\left(\int_0^1 \nabla \psi_{\mu}^{\sigma,\omega}(z + t\alpha'd) dt \right)^{\top} \alpha'd > \omega_a \alpha' \nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d.$$

Donc, en simplifiant par α' et en retranchant $\nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d$,

$$\left(\int_0^1 (\nabla \psi_{\mu}^{\sigma,\omega}(z + t\alpha'd) - \nabla \psi_{\mu}^{\sigma,\omega}(z)) dt \right)^{\top} d > (\omega_a - 1) \nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d.$$

Donc, en utilisant l'inégalité de Cauchy-Schwarz,

$$\left(\int_0^1 \|\nabla \psi_{\mu}^{\sigma,\omega}(z + t\alpha'd) - \nabla \psi_{\mu}^{\sigma,\omega}(z)\| dt \right)^{\top} \|d\| > (\omega_a - 1) \nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d.$$

Puis, en notant L la constante de Lipschitz du gradient de $\psi_{\mu}^{\sigma,\omega}$,

$$\left(\int_0^1 L \|(z + t\alpha'd) - z\| dt \right)^{\top} \|d\| > (\omega_a - 1) \nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d.$$

En simplifiant, on a

$$\alpha' > \frac{(\omega_a - 1) \nabla \psi_{\mu}^{\sigma,\omega}(z)^{\top} d}{L \|d\|^2}.$$

On a $\beta\alpha' < \alpha$. On obtient donc la minoration

$$\alpha > \frac{\beta'(\omega_\alpha - 1)}{L} \frac{\nabla\psi_\mu^{\sigma,\omega}(z)^\top d}{\|d\|^2}.$$

En rappelant que $\nabla\psi_\mu^{\sigma,\omega}(z)^\top d \leq 0$, on en déduit que

$$\omega_\alpha \alpha \nabla\psi_\mu^{\sigma,\omega}(z)^\top d \leq \frac{\beta'(\omega_\alpha - 1)\omega_\alpha}{L} \frac{(\nabla\psi_\mu^{\sigma,\omega}(z)^\top d)^2}{\|d\|^2}.$$

Comme le pas α vérifie la condition d'Armijo (3.44), on a

$$\psi_\mu^{\sigma,\omega}(z + \alpha d) \leq \psi_\mu^{\sigma,\omega}(z) - \frac{\beta'(1 - \omega_\alpha)\omega_\alpha}{L} \frac{|\nabla\psi_\mu^{\sigma,\omega}(z)^\top d|^2}{\|d\|^2}.$$

On obtient donc la seconde inégalité (3.46) et, en posant

$$K_3 = \omega_\alpha \min\left(\frac{\beta'(1 - \omega_\alpha)}{L}, 1\right),$$

on a le résultat cherché. \square

Le second lemme est une propriété des formules de BFGS ([12]). Notons par $[\cdot]$ l'opérateur de partie entière défini par $[x] = i$, quand $i - 1 < x \leq i$ et $i \in \mathbb{N}$. Pour une démonstration du second lemme, se référer à [12].

Lemme 3.9.2 Soit $\{M_k\}$ les matrices définies positives obtenues par les formules de BFGS avec les paires de vecteurs $\{(\gamma_k, \delta_k)\}_{k \geq 1}$, satisfaisant pour tout $k \geq 1$

$$\gamma_k^\top \delta_k \geq \alpha_1 \|\delta_k\|^2 \quad \text{et} \quad \gamma_k^\top \delta_k \geq \alpha_2 \|\gamma_k\|^2, \quad (3.47)$$

où $\alpha_1 > 0$ et $\alpha_2 > 0$ ne dépendent pas de k . Alors, pour tout $r \in]0, 1[$, il existe des constantes positives b_1, b_2 , et b_3 telles que pour tout $k \geq 1$,

$$\cos \theta_j := \frac{\delta_j^\top M_j \delta_j}{\|M_k \delta_j\| \|\delta_j\|} \geq b_1 \quad \text{et} \quad b_2 \leq \frac{\|M_j \delta_j\|}{\|\delta_j\|} \leq b_3, \quad (3.48)$$

pour au moins $[rk]$ indices j dans $\{1, \dots, k\}$.

Montrons qu'on a effectivement les hypothèses du lemme précédent satisfaites dans notre cas.

Lemme 3.9.3 Supposons l'hypothèse H_1 satisfaite et f et c de classe $C^{1,1}$. Il existe alors deux constantes $\alpha_1 > 0$ et $\alpha_2 > 0$ telles que, pour tout $k \geq 1$

$$\gamma_k^\top \delta_k \geq \alpha_1 \|\delta_k\|^2 \quad \text{et} \quad \gamma_k^\top \delta_k \geq \alpha_2 \|\gamma_k\|^2.$$

Preuve. On sait que la fonction $\psi_\mu^{\sigma,\omega}$ décroît à chaque itération. Les itérés générés par l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$ restent donc dans l'ensemble de niveau \mathcal{L}_1 . D'après

le lemme 3.3.2 et les bornes de la proposition 3.7.3, le lagrangien est fortement convexe sur \mathcal{L}_1 de module $\kappa > 0$. Ainsi

$$\gamma_k^\top \delta_k = (\nabla_x \ell(x_{k+1}, \lambda_{k+1}, \xi_{k+1}) - \nabla_x \ell(x_k, \lambda_{k+1}, \xi_{k+1}))^\top (x_{k+1} - x_k) \geq \kappa \|\delta_k\|^2.$$

Il suffit alors de prendre $\alpha_1 = \kappa$ pour obtenir la première inégalité.

Pour montrer la seconde inégalité, on commence par majorer la norme de γ_k . Par hypothèse, les gradients des fonctions f et c sont Lipschitz de module respectif κ_f et κ_c . On sait aussi d'après la proposition 3.7.3 que λ_{k+1} est majorée par une constante positive non nulle K_2 . On a donc

$$\begin{aligned} \|\gamma_k\| &\leq \|\nabla_x \ell(x_{k+1}, \lambda_{k+1}, \xi_{k+1}) - \nabla_x \ell(x_k, \lambda_{k+1}, \xi_{k+1})\| \\ &\leq \|(\nabla f(x_{k+1}) - \nabla f(x_k)) + (\nabla c(x_{k+1}) - \nabla c(x_k))^\top \lambda_{k+1}\| \\ &\leq (\kappa_f + \kappa_c K_2) \|x_{k+1} - x_k\| \\ &\leq K' \|\delta_k\|. \end{aligned}$$

On en déduit que $\|\delta_k\|^2 \geq (1/K')^2 \|\gamma_k\|^2$, puis en utilisant l'inégalité faisant intervenir α_1 , qu'il suffit de prendre $\alpha_2 = \alpha_1 / (K')^2$ pour obtenir le résultat. \square

Proposition 3.9.4 *Supposons l'hypothèse H_1 satisfaite et f et c de classe $C^{1,1}$ dans un voisinage de l'ensemble de niveau \mathcal{L}_1 . Alors l'algorithme $\mathcal{A}_\mu^{\sigma, \omega}$ génère une suite $\{z_k\}$ telle que, pour tout $r \in]0, 1[$, on a pour tout $k \geq 1$,*

$$\begin{aligned} &\psi_\mu^{\sigma, \omega}(z_j + \alpha_j d_j) - \psi_\mu^{\sigma, \omega}(z_j) \\ &\leq -K(\|M_j d_j^x\|^2 + \|\nabla c_j d_j^x - d_j^s\|^2 + \|d_j^s\|^2 + \|d_j^y\|^2 \\ &\quad + \|(S_j - C_j)\lambda_j - \mu e_m\|^2 + \|S_j(\sigma_j e_m - \lambda_j) - \mu e_m\|^2 \\ &\quad + \|Y_j(\omega_j e_p - D\xi_j) - \mu e_p\|^2) \end{aligned} \tag{3.48}$$

pour au moins $\lceil rk \rceil$ indices j dans $\{1, \dots, k\}$, où K est une constante positive et où $d_j = (d_j^x, d_j^s, d_j^y, d_j^\lambda, d_j^\xi)$ est la direction calculée par l'algorithme à l'étape 1.

Preuve. Pour simplifier les notations, on omet les indices j des itérations. On note z l'itéré courant et K'_1, K'_2, \dots des constantes positives (indépendantes de l'indice de l'itération).

Les bornes sur $(s - c, s, y, \lambda, \sigma e_m - \lambda, \omega e_p - D\xi)$ obtenues par le lemme 3.7.3 et le fait que f et c soient de classe $C^{1,1}$ entraînent que $\psi_\mu^{\sigma, \omega}$ est aussi de classe $C^{1,1}$ sur un voisinage ouvert convexe de l'ensemble de niveau \mathcal{L}_1 . La recherche linéaire employée et le lemme 3.9.1 nous permettent d'obtenir une des deux inégalités (3.45) ou (3.46).

Pour appliquer le lemme 3.9.2, on fixe $r \in]0, 1[$ et on note J l'ensemble des indices j pour lesquels on a les inégalités (3.48).

On va tout d'abord minorer le terme $|\nabla \psi_\mu^{\sigma, \omega}(z)^\top d|$ présent dans les inégalités (3.45) et (3.46). On va ensuite majorer le terme $\|d\|^2$ de l'inégalité (3.46).

En utilisant (3.36), les bornes de la proposition 3.7.3 et (3.48), on a

$$\begin{aligned}
 & |\nabla\psi_\mu^{\sigma,\omega}(z)^\top d| \\
 &= (d^x)^\top M d^x + \|\Upsilon^{1/2}(\omega I - D\xi)^{-1/2} d^y\|^2 \\
 &\quad + \|\Lambda^{1/2}(S-C)^{-1/2}(\nabla c^\top d^x - d^s)\|^2 + \|(\sigma I - \Lambda)^{1/2} S^{-1/2} d^s\|^2 \\
 &\quad + \tau \|(S-C)^{-1/2} \Lambda^{-1/2} ((S-C)\lambda - \mu e_m)\|^2 \\
 &\quad + \tau \|S^{-1/2}(\sigma I - \Lambda)^{-1/2} (S(\sigma e_m - \lambda) - \mu e_m)\|^2 \\
 &\quad + \tau \|\Upsilon^{-1/2}(\omega I - D\xi)^{-1/2} (Y(\omega e_p - D\xi) - \mu e_p)\|^2 \\
 &\geq \frac{b_1}{b_3} \|M d^x\|^2 + K_1 K_2^{-1} \left(\|\nabla c^\top d^x - d^s\|^2 + \|d^s\|^2 + \|d^y\|^2 \right) \\
 &\quad + \tau K_2^{-2} (\|(S-C)\lambda - \mu e_m\|^2 + \|S(\sigma e_m - \lambda) - \mu e_m\|^2) \\
 &\quad + \tau K_2^{-2} \|Y(\omega e_p - D\xi) - \mu e_p\|^2 \\
 &\geq K'_1 (\|M d^x\|^2 + \|\nabla c^\top d^x - d^s\|^2 + \|d^s\|^2 + \|d^y\|^2 + \|(S-C)\lambda - \mu e_m\|^2 \\
 &\quad + \|S(\sigma e_m - \lambda) - \mu e_m\|^2 + \|Y(\omega e_p - D\xi) - \mu e_p\|^2) \tag{3.49}
 \end{aligned}$$

Majorons maintenant la norme de la direction d . D'après le système de Newton (3.15), on remarque que

$$d^\xi = (YD)^{-1}((\omega I - D\xi)d^y + (Y(\omega e_p - D\xi) - \mu e_p)) \tag{3.50}$$

$$\begin{aligned}
 d^\lambda &= S^{-1}((\sigma I - \Lambda)d^s - (S(\sigma e_m - \lambda) - \mu e_m)) \\
 &= (S-C)^{-1}(\Lambda(\nabla c d^x - d^s) - ((S-C)\lambda - \mu e_m)) \tag{3.51}
 \end{aligned}$$

Ainsi, à partir du système de Newton (3.15), de l'égalité précédente, des inégalités (3.48) et des bornes de la proposition 3.7.3, on obtient

$$\begin{aligned}
 \|d\|^2 &= \|d^x\|^2 + \|d^s\|^2 + \|d^y\|^2 + \|d^\lambda\|^2 + \|d^\xi\|^2 \\
 &\leq \|d^x\|^2 + \|d^s\|^2 + \|d^y\|^2 \\
 &\quad + \|S^{-1}((\sigma I - \Lambda)d^s - (S(\sigma e_m - \lambda) - \mu e_m))\|^2 \\
 &\quad + \|(YD)^{-1}((\omega I - D\xi)d^y + (Y(\omega e_p - D\xi) - \mu e_p))\|^2 \\
 &\leq \|d^x\|^2 + \|d^s\|^2 + \|d^y\|^2 \\
 &\quad + \|S^{-1}(\sigma I - \Lambda)d^s\|^2 + \|S^{-1}S(\sigma e_m - \lambda) - \mu e_m\|^2 \\
 &\quad + \|(YD)^{-1}(\omega I - D\xi)d^y\|^2 + \|(YD)^{-1}(Y(\omega e_p - D\xi) - \mu e_p)\|^2 \\
 &\leq b_2^{-2} \|M d^x\|^2 + (1 + (K_1^{-1} K_2)^2) \|d^s\|^2 + K_1^{-2} \|S(\sigma e_m - \lambda) - \mu e_m\|^2 \\
 &\quad + (1 + (K_1^{-1} K_2 K'_D)^2) \|d^y\|^2 + (K'_D K_1^{-1})^2 \|Y(\omega e_p - D\xi) - \mu e_p\|^2 \\
 &\leq K'_3 (\|M d^x\|^2 + \|d^s\|^2 + \|d^y\|^2 + \|S(\sigma e_m - \lambda) - \mu e_m\|^2 \\
 &\quad + \|Y(\omega e_p - D\xi) - \mu e_p\|^2) \tag{3.52} \\
 &\leq K'_3 (\|M d^x\|^2 + \|\nabla c d^x - d^s\|^2 + \|d^s\|^2 + \|d^y\|^2 + \|(S-C)\lambda - \mu e_m\|^2 \\
 &\quad + \|S(\sigma e_m - \lambda) - \mu e_m\|^2 + \|Y(\omega e_p - D\xi) - \mu e_p\|^2)
 \end{aligned}$$

en notant K'_D une majoration de la norme de $\|D\|$.

En utilisant les inégalités (3.45), (3.46), (3.49),(3.52), on obtient la majoration suivante

$$\begin{aligned} \psi_{\mu}^{\sigma,\omega}(z + \alpha d) &\leq \psi_{\mu}^{\sigma,\omega}(z) - K(\|Md^x\|^2 + \|\nabla c d^x - d^s\|^2 + \|d^s\|^2 + \|d^y\|^2 \\ &\quad + \|(S-C)\lambda - \mu e_m\|^2 + \|S(\sigma e_m - \lambda) - \mu e_m\|^2 \\ &\quad + \|Y(\omega e_p - D\xi) - \mu e_p\|^2) \end{aligned}$$

où $K = K_3 K'_1 \max(1, K'_1/K'_3)$ est une constante positive. \square

On ne montre pas directement la convergence r-linéaire des itérés internes vers la solution du problème barrière comme dans [2]. On montre, comme dans [3], d'abord la convergence de ces itérés vers cette solution, puis la convergence r-linéaire.

Proposition 3.9.5 *Supposons l'hypothèse H_1 satisfaite et f et c de classe $C^{1,1}$ dans un voisinage de l'ensemble de niveau \mathcal{L}_1 . Supposons que l'algorithme $\mathcal{A}_{\mu}^{\sigma,\omega}$ ne stoppe pas à l'étape 1. Alors l'algorithme $\mathcal{A}_{\mu}^{\sigma,\omega}$ génère une suite $\{z_k\}$ convergent vers \hat{z}_{μ} .*

Preuve. Fixons $r \in]0, 1[$ et notons J l'ensemble des indices j pour lesquels on a les inégalités (3.48). On sait d'après la recherche linéaire que la suite $\{\psi_{\mu}^{\sigma,\omega}(z_k)\}$ est une suite décroissante minorée par $\psi_{\mu}^{\sigma,\omega}(\hat{z}_{\mu})$. C'est donc une suite convergente. D'après l'inégalité (3.48) de la proposition (3.9.4), on en déduit que, pour $j \rightarrow \infty$, on a

$$\begin{cases} M_j d_j^x & \rightarrow 0 \\ \nabla c_j d_j^x - d_j^s & \rightarrow 0 \\ d_j^s & \rightarrow 0 \\ d_j^y & \rightarrow 0 \\ (S_j - C_j)\lambda_j - \mu e_m & \rightarrow 0 \\ S_j(\sigma_j e_m - \lambda_j) - \mu e_m & \rightarrow 0 \\ Y_j(\omega_j e_p - D\xi_j) - \mu e_p & \rightarrow 0. \end{cases}$$

Donc, en considérant le système de Newton (3.15) et les limites précédentes, on a, pour $j \rightarrow \infty$,

$$\begin{cases} \nabla \ell(x_j, s_j, y_j, \lambda_j + d_j^{\lambda}, \xi_j + d_j^{\xi}) & \rightarrow 0 \\ d_j^s & \rightarrow 0 \\ d_j^y & \rightarrow 0 \\ (S_j - C_j)d_j^{\lambda} & \rightarrow 0 \\ -Y_j D d_j^{\xi} & \rightarrow 0. \end{cases}$$

En utilisant les bornes strictement positives de la proposition 3.7.3, et comme $\{x_j\}_{j \in J}$ est dans le compact \mathcal{L}_1 (proposition 3.7.3), on a aussi

$$d_j^{\lambda} \rightarrow 0 \text{ et } d_j^{\xi} \rightarrow 0 \text{ et } \nabla c_j^{\top} d_j^{\lambda} \rightarrow 0.$$

On obtient donc que, pour $j \rightarrow \infty$,

$$\begin{cases} \nabla \ell(x_j, s_j, y_j, \lambda_j, \xi_j) & \rightarrow 0 \\ (S_j - C_j)\lambda_j - \mu e_m & \rightarrow 0 \\ S_j(\sigma_j e_m - \lambda_j) - \mu e_m & \rightarrow 0 \\ Y_j(\omega_j e_p - D\xi_j) - \mu e_p & \rightarrow 0 \\ Ax_j + Dy_j = b. & \end{cases}$$

Ainsi tout point limite \bar{z} de la suite $\{z_j\}_{j \in J}$ vérifie le système (3.8) et donc $\bar{z} = \hat{z}_\mu$. Comme la suite $\{z_j\}_{j \in J}$ reste dans le compact \mathcal{L}_1 , toute la suite converge vers \hat{z}_μ . On sait que la suite $\{\psi_\mu^{\sigma,\omega}(z_k)\}$ converge vers $\psi_\mu^{\sigma,\omega}(\hat{z}_\mu)$. Comme la fonction $\psi_\mu^{\sigma,\omega}$ a un minimum unique (proposition 3.7.1) et que tous les itérés z_k restent dans \mathcal{L}_1 , on en déduit que toute la suite $\{z_k\}$ converge vers \hat{z}_μ . \square

Théorème 3.9.6 *Supposons l'hypothèse H_1 satisfaite, f et c de classe $C^{1,1}$ dans un voisinage ouvert de l'ensemble de niveau \mathcal{L}_1 . Alors, l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$ génère une suite $\{z_k\}$ convergente vers \hat{z}_μ r-linéairement. On a, en particulier,*

$$\sum_{k \geq 1} \|z_k - \hat{z}_\mu\| < \infty.$$

Preuve. On sait d'après la proposition précédente que la suite $\{z_k\}$ converge vers \hat{z}_μ . Fixons $r \in]0, 1[$ et notons J l'ensemble des indices j pour lesquels les inégalités (3.48) sont vérifiées. On sait d'après la proposition 3.9.4 que, pour tout $j \in J$

$$\begin{aligned} & \psi_\mu^{\sigma,\omega}(z_j + \alpha_j d_j) - \psi_\mu^{\sigma,\omega}(z_j) \\ & \leq -K(\|M_j d_j^x\|^2 + \|\nabla c_j d_j^x - d_j^s\|^2 + \|d_j^s\|^2 + \|d_j^y\|^2 \\ & \quad + \|(S_j - C_j)\lambda_j - \mu e_m\|^2 + \|S_j(\sigma_j e_m - \lambda_j) - \mu e_m\|^2 \\ & \quad + \|Y_j(\omega_j e_p - D\xi_j) - \mu e_p\|^2) \end{aligned}$$

et, d'après la proposition 3.7.5 que, pour z_j dans un voisinage \mathcal{N} de \hat{z}_μ ($j \in J$),

$$\begin{aligned} \psi_\mu^{\sigma,\omega}(z_j) - \psi_\mu^{\sigma,\omega}(\hat{z}_\mu) & \leq \frac{1}{\alpha} (\|\nabla \ell(x_j, s_j, y_j, \lambda_j, \xi_j)\|^2 \\ & \quad + \|(S_j - C_j)\lambda_j - \mu e_m\|^2 \\ & \quad + \|S_j(\sigma e_m - \lambda_j) - \mu e_m\|^2 \\ & \quad + \|Y_j(\omega e_p - D\xi_j) - \mu e_p\|^2). \end{aligned}$$

Majorons le terme $\|\nabla \ell(x_j, s_j, y_j, \lambda_j, \xi_j)\|^2$ dans l'inégalité précédente. Avec le pas de Newton (3.15), les inégalités (3.50) et (3.51), en notant K_A une borne de la norme $\|A\|$, on a

$$\begin{aligned} & \|\nabla \ell(x_j, s_j, y_j, \lambda_j, \xi_j)\|^2 \\ & = \|\nabla f(x_j) + \nabla c_j^\top \lambda_j - A^\top \xi_j\|^2 \\ & = \|M_j d_j^x + \nabla c_j^\top d_j^\lambda - A^\top d_j^\xi\|^2 \\ & \leq 4(\|M_j d_j^x\|^2 + K_2(K_c/K_1)^2 \|\nabla c_j^\top d_j^x - d_j^s\|^2 \\ & \quad + (K_c/K_1)^2 \|(S_j - C_j)\lambda_j - \mu e_m\|^2 + K_2(K_D/K_1)^2 \|d_j^y\|^2 \\ & \quad + (K_c/K_1)^2 \|Y_j(\omega e_p - D\xi_j) - \mu e_p\|^2). \end{aligned}$$

Il existe donc une constante K' strictement positive telle que

$$\begin{aligned}
& \|\nabla\ell(x_j, s_j, y_j, \lambda_j, \xi_j)\|^2 + \|(S_j - C_j)\lambda_j - \mu e_m\|^2 + \|S_j(\sigma e_m - \lambda_j) - \mu e_m\|^2 \\
& \quad + \|Y_j(\omega e_p - D\xi_j) - \mu e_p\|^2 \\
& \leq K'(\|M_j d_j^x\|^2 + \|\nabla c_j^\top d_j^x - d_j^s\|^2 + \|d_j^s\|^2 + \|d_j^y\|^2 \\
& \quad + \|(S_j - C_j)\lambda_j - \mu e_m\|^2 + \|S_j(\sigma e_m - \lambda_j) - \mu e_m\|^2 \\
& \quad + \|Y_j(\omega e_p - D\xi_j) - \mu e_p\|^2).
\end{aligned}$$

Finalement, en posant $K'' = K/K'$, on obtient la majoration suivante

$$\begin{aligned}
& \psi_\mu^{\sigma, \omega}(z_j + \alpha_j d_j) - \psi_\mu^{\sigma, \omega}(z_j) \\
& \leq -K''(\|\nabla\ell(x_j, s_j, y_j, \lambda_j, \xi_j)\|^2 + \|(S_j - C_j)\lambda_j - \mu e_m\|^2 + \|S_j(\sigma e_m - \lambda_j) - \mu e_m\|^2 \\
& \quad + \|Y_j(\omega e_p - D\xi_j) - \mu e_p\|^2).
\end{aligned}$$

Par convergence de la suite $\{z_k\}$ vers \hat{z}_μ , il existe un indice $j_0 \in J$ tel que, pour tout $j \in J$ avec $j \geq j_0$, z_j soit dans le voisinage \mathcal{N} de la proposition 3.7.5. On a donc, pour tout $j \in J$ tel que $j \geq j_0$,

$$\begin{aligned}
& \psi_\mu^{\sigma, \omega}(z_{j+1}) - \psi_\mu^{\sigma, \omega}(\hat{z}_\mu) \\
& \leq (\psi_\mu^{\sigma, \omega}(z_{j+1}) - \psi_\mu^{\sigma, \omega}(z_j)) + (\psi_\mu^{\sigma, \omega}(z_j) - \psi_\mu^{\sigma, \omega}(\hat{z}_\mu)) \\
& \leq \tau (\psi_\mu^{\sigma, \omega}(z_j) - \psi_\mu^{\sigma, \omega}(\hat{z}_\mu)), \tag{3.53}
\end{aligned}$$

avec $\tau = (1 - \alpha K'')$. Par positivité des constantes α et K'' , par positivité du terme de gauche de l'inégalité ci-dessus (\hat{z}_μ est le minimum unique de $(\mathcal{PD})_\mu^{\sigma, \omega}$, proposition 3.7.1) et comme $\psi_\mu^{\sigma, \omega}(z_{j+1}) \leq \psi_\mu^{\sigma, \omega}(z_j)$, $\tau \in [0, 1[$.

D'après le lemme (3.9.2), on a au moins $\lceil rk \rceil$ indices qui vérifient l'inégalité (3.53). On a donc la relation suivante sur les indices

$$|[1, k] \cap J| \geq \lceil rk \rceil.$$

Considérons les indices k plus grands que l'indice j_0 . On a la relation

$$|[j_0, k] \cap J| \geq \lceil rk \rceil - j_0 + 1 \geq rk - j_0 + 1.$$

Posons $k_0 := \lceil j_0/r \rceil$, on a alors, pour tout $k \geq k_0$,

$$|[j_0, k] \cap J| \geq 0.$$

Cela signifie que, pour tout indice $k \geq k_0$, l'itéré z_{k+1} vérifie l'inégalité (3.53). On a aussi d'après la recherche linéaire (3.44),

$$\psi_\mu^{\sigma, \omega}(z_{k+1}) - \psi_\mu^{\sigma, \omega}(\hat{z}_\mu) \leq \psi_\mu^{\sigma, \omega}(z_k) - \psi_\mu^{\sigma, \omega}(\hat{z}_\mu).$$

On en déduit récursivement que

$$\psi_\mu^{\sigma, \omega}(z_{k+1}) - \psi_\mu^{\sigma, \omega}(\hat{z}_\mu) \leq \tau^{k-k_0+1} (\psi_\mu^{\sigma, \omega}(z_{k_0}) - \psi_\mu^{\sigma, \omega}(\hat{z}_\mu)),$$

pour tout $k \geq k_0$. Finalement, avec l'inégalité de gauche dans (3.40) on a, pour $k \geq k_0$

$$\|z_{k+1} - \hat{z}_\mu\| \leq K'_7 \tau^{k/2},$$

en notant $K'_7 = \alpha^{-1/2} \left(\frac{\psi_\mu^{\sigma, \omega}(z_{k_0}) - \psi_\mu^{\sigma, \omega}(\hat{z}_\mu)}{(1 - \alpha K'')^{k_0 - 1}} \right)^{1/2}$. \square

3.10 Convergence q-superlinéaire de l'algorithme

$\mathcal{A}_\mu^{\sigma,\omega}$

Dans cette partie, on montre que la suite $\{z_k\}$ générée par l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$ converge q-superlinéairement vers la solution \hat{z}_μ du problème barrière, c'est à dire que

$$z_{k+1} - \hat{z}_\mu = o(\|z_k - \hat{z}_\mu\|),$$

ou encore que $\|z_{k+1} - \hat{z}_\mu\|/\|z_k - \hat{z}_\mu\| \rightarrow 0$ (en supposant $z_k \neq \hat{z}_\mu$). Pour montrer cette convergence, on montre tout d'abord que sous l'hypothèse (3.54), le pas unité est accepté pour la recherche linéaire. Cette hypothèse signifie que le quotient de Rayleigh

$$\frac{(d_k^x)^\top M_k d_k^x}{\|d_k^x\|^2}$$

ne devient pas trop petit, donc que la direction d^x obtenue en résolvant le système (3.15) ne devient pas trop grande pour pouvoir obtenir l'acceptation du pas unité pour la recherche linéaire.

Proposition 3.10.1 *Supposons l'hypothèse H_1 satisfaite et que f et c soient deux fois continûment différentiables au voisinage de \hat{x}_μ . Supposons aussi que la suite $\{z_k\}$ générée par l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$ converge vers \hat{z}_μ et que les matrices définies positives M_k satisfont l'estimation*

$$(d_k^x)^\top (M_k - \hat{M}_\mu^{\sigma,\omega}) d_k^x \geq o(\|d_k^x\|^2), \quad (3.54)$$

où

$$\hat{M}_\mu^{\sigma,\omega} := \nabla_{xx}^2 \ell(\hat{x}_\mu, \hat{\lambda}_\mu, \hat{\xi}_\mu).$$

Alors, pour k suffisamment grand, le pas unité est accepté pour la condition de recherche linéaire, c'est à dire

$$\psi_\mu^{\sigma,\omega}(z_k + d_k) \leq \psi_\mu^{\sigma,\omega}(z_k) + \omega_a \nabla \psi_\mu^{\sigma,\omega}(z_k)^\top d_k. \quad (3.55)$$

Preuve. Pour simplifier les notations, on omet l'indice de l'itération k et on note z l'itéré courant. Comme on prouve une propriété asymptotique, il est implicitement supposé que k est suffisamment grand pour satisfaire les diverses propriétés mentionnées et que les limites sont calculées quand $k \rightarrow \infty$. On note K'_1, K'_2, \dots des constantes positives (indépendantes de l'indice de l'itération).

Observons tout d'abord que la définie positivité de $\hat{M}_\mu^{\sigma,\omega}$ et (3.54) entraînent que

$$(d^x)^\top M d^x \geq K'_1 \|d^x\|^2, \quad (3.56)$$

pour une constante positive K'_1 . Observons aussi que $d \rightarrow 0$ (pour $d^x \rightarrow 0$, utiliser (3.15), (3.56) et $\nabla \varphi_\mu^{\sigma,\omega}(x, s, y) \rightarrow 0$). On a aussi le fait que z et $z+d$ soient proches de \hat{z}_μ après un certain nombre d'itérations. On peut donc développer à l'ordre 2 la fonction $\psi_\mu^{\sigma,\omega}(z+d)$ en z . Un développement de Taylor du second ordre nous donne

$$\psi_\mu^{\sigma,\omega}(z+d) = \psi_\mu^{\sigma,\omega}(z) + \nabla \psi_\mu^{\sigma,\omega}(z)^\top d + \frac{1}{2} d^\top \nabla^2 \psi_\mu^{\sigma,\omega}(z) d + o(\|d\|^2).$$

En ajoutant le terme $\omega_a \nabla \psi_\mu^{\sigma, \omega}(z)^\top d$ à cette égalité et en regroupant certains termes, on obtient

$$\begin{aligned}
 & \psi_\mu^{\sigma, \omega}(z + d) - \psi_\mu^{\sigma, \omega}(z) - \omega_a \nabla \psi_\mu^{\sigma, \omega}(z)^\top d \\
 &= (1 - \omega_a) \nabla \psi_\mu^{\sigma, \omega}(z)^\top d + \frac{1}{2} d^\top \nabla^2 \psi_\mu^{\sigma, \omega}(z) d + o(\|d\|^2) \\
 &= \left(\frac{1}{2} - \omega_a \right) \nabla \psi_\mu^{\sigma, \omega}(z)^\top d \\
 & \quad + \frac{1}{2} (\nabla \psi_\mu^{\sigma, \omega}(z)^\top d + d^\top \nabla^2 \psi_\mu^{\sigma, \omega}(z) d) + o(\|d\|^2).
 \end{aligned} \tag{3.57}$$

On va maintenant montrer que le membre de droite de l'égalité ci-dessus est négatif pour des valeurs de k suffisamment grandes (ce qui montrera que le pas unité est alors accepté). Montrons tout d'abord que le terme $(\nabla \psi_\mu^{\sigma, \omega}(z)^\top d + d^\top \nabla^2 \psi_\mu^{\sigma, \omega}(z) d)$ est plus petit qu'une quantité de l'ordre de $o(\|d\|^2)$. Par (3.36) (d direction de descente de $\psi_\mu^{\sigma, \omega}$), on a

$$\begin{aligned}
 \nabla \psi_\mu^{\sigma, \omega}(z)^\top d &= -(d^x)^\top M d^x - \|(\sigma I - \Lambda)^{1/2} S^{-1/2} d^s\|^2 \\
 & \quad - \|\Lambda^{1/2} (S - C)^{-1/2} (\nabla c^\top d^x - d^s)\|^2 \\
 & \quad - \|Y^{-1/2} (\omega I - D\xi)^{1/2} d^y\|^2 \\
 & \quad - \tau \|(S - C)^{-1/2} \Lambda^{-1/2} ((S - C)\lambda - \mu e_m)\|^2 \\
 & \quad - \tau \|S^{-1/2} (\sigma I - \Lambda)^{-1/2} (S(\sigma e_m - \lambda) - \mu e_l)\|^2 \\
 & \quad - \tau \|Y^{-1/2} (\omega I - D\xi)^{-1/2} (Y(\omega e_p - D\xi) - \mu e_p)\|^2,
 \end{aligned} \tag{3.58}$$

Avec le système de Newton (3.15), on obtient les égalités suivantes

$$\begin{aligned}
 & \|(S - C)^{-1/2} \Lambda^{-1/2} ((S - C)\lambda - \mu e)\|^2 \\
 &= \|(S - C)^{-1/2} \Lambda^{-1/2} (\Lambda (\nabla c^\top d^x - d^s) - (S - C) d^\lambda)\|^2 \\
 &= \|(S - C)^{-1/2} \Lambda^{1/2} (\nabla c^\top d^x - d^s)\|^2 + \|(S - C)^{1/2} \Lambda^{-1/2} d^\lambda\|^2 \\
 & \quad - 2(\nabla c d^x - d^s)^\top d^\lambda,
 \end{aligned} \tag{3.59}$$

$$\begin{aligned}
 & \|S^{-1/2} (\sigma I - \Lambda)^{-1/2} (S(\sigma - \lambda) - \mu e)\|^2 \\
 &= \|S^{-1/2} (\sigma I - \Lambda)^{-1/2} (-(\sigma I - \Lambda) d^\lambda + S d^\lambda)\|^2 \\
 &= \|S^{-1/2} (\sigma I - \Lambda)^{1/2} d^s\|^2 + \|S^{1/2} (\sigma I - \Lambda)^{-1/2} d^\lambda\|^2 \\
 & \quad - 2(d^\lambda)^\top d^s
 \end{aligned} \tag{3.60}$$

$$\begin{aligned}
 & \|Y^{-1/2} (\omega I - D\xi)^{-1/2} (Y(\omega e_p - D\xi) - \mu e_p)\|^2 \\
 &= \|Y^{-1/2} (\omega I - D\xi)^{-1/2} (Y D d^\xi - (\omega I - D\xi) d^y)\|^2 \\
 &= \|Y^{-1/2} (\omega I - D\xi)^{1/2} d^y\|^2 + \|Y^{1/2} (\omega I - D\xi)^{-1/2} D d^\xi\|^2 \\
 & \quad - 2(d^\xi)^\top D d^y.
 \end{aligned} \tag{3.61}$$

A l'aide de ces égalités, on réécrit l'équation (3.58)

$$\begin{aligned}
 \nabla\psi_\mu^{\sigma,\omega}(z)^\top \mathbf{d} &= -(\mathbf{d}^x)^\top \mathbf{M} \mathbf{d}^x \\
 &\quad - (1 + \tau) \|(\sigma \mathbf{I} - \Lambda)^{1/2} \mathbf{S}^{-1/2} \mathbf{d}^s\|^2 \\
 &\quad - (1 + \tau) \|\Lambda^{1/2} (\mathbf{S} - \mathbf{C})^{-1/2} (\nabla \mathbf{c}^\top \mathbf{d}^x - \mathbf{d}^s)\|^2 \\
 &\quad - (1 + \tau) \|\mathbf{Y}^{-1/2} (\omega \mathbf{I} - \mathbf{D} \Xi)^{1/2} \mathbf{d}^y\|^2 \\
 &\quad - \tau \|(\mathbf{S} - \mathbf{C})^{1/2} \Lambda^{-1/2} \mathbf{d}^\lambda\|^2 \\
 &\quad - \tau \|\mathbf{S}^{1/2} (\sigma \mathbf{I} - \Lambda)^{-1/2} \mathbf{d}^\lambda\|^2 \\
 &\quad - \tau \|\mathbf{Y}^{1/2} (\omega \mathbf{I} - \mathbf{D} \Xi)^{-1/2} \mathbf{D} \mathbf{d}^\xi\|^2 \\
 &\quad + 2\tau (\nabla \mathbf{c} \mathbf{d}^x - \mathbf{d}^s)^\top \mathbf{d}^\lambda + 2\tau (\mathbf{d}^\lambda)^\top \mathbf{d}^s + 2\tau (\mathbf{d}^\xi)^\top \mathbf{D} \mathbf{d}^y.
 \end{aligned} \tag{3.62}$$

On a aussi, avec les équations (3.13) et (3.26) faisant intervenir les hessiens de $\varphi_\mu^{\sigma,\omega}$ et de $\mathcal{V}_\mu^{\sigma,\omega}$, avec $\xi \in \mathbb{R}^p$ quelconque,

$$\begin{aligned}
 &\begin{pmatrix} \mathbf{d}^x \\ \mathbf{d}^s \\ \mathbf{d}^y \\ \mathbf{d}^\lambda \\ \mathbf{d}^\xi \end{pmatrix}^\top \nabla^2 \psi_\mu^{\sigma,\omega}(z) \begin{pmatrix} \mathbf{d}^x \\ \mathbf{d}^s \\ \mathbf{d}^y \\ \mathbf{d}^\lambda \\ \mathbf{d}^\xi \end{pmatrix} \\
 &= (\mathbf{d}^x)^\top \nabla_{xx}^2 \ell(x, (1 + \tau)\mu(\mathbf{S} - \mathbf{C})^{-1} \mathbf{e} - \tau\lambda, \xi) \mathbf{d}^x \\
 &\quad + (1 + \tau)\mu \|\mathbf{Y}^{-1} \mathbf{d}^y\|^2 \\
 &\quad + (1 + \tau)\mu \|\mathbf{S}^{-1} \mathbf{d}^s\|^2 \\
 &\quad + (1 + \tau)\mu \|(\mathbf{S} - \mathbf{C})^{-1} (\nabla \mathbf{c}^\top \mathbf{d}^x - \mathbf{d}^s)\|^2 \\
 &\quad + \tau\mu \|\Lambda^{-1} \mathbf{d}^\lambda\|^2 \\
 &\quad + \tau\mu \|(\sigma \mathbf{I} - \Lambda)^{-1} \mathbf{d}^\lambda\|^2 \\
 &\quad + \tau\mu \|(\omega \mathbf{I} - \mathbf{D} \Xi)^{-1} \mathbf{d}^\xi\|^2 \\
 &\quad - 2\tau (\mathbf{d}^s)^\top \mathbf{d}^\lambda - 2\tau (\nabla \mathbf{c}^\top \mathbf{d}^x - \mathbf{d}^s)^\top \mathbf{d}^\lambda - 2\tau (\mathbf{d}^\xi)^\top \mathbf{D} \mathbf{d}^y.
 \end{aligned} \tag{3.63}$$

On obtient finalement en additionnant (3.62) et (3.63), en regroupant certains termes et en notant $\tilde{\lambda} = (1 + \tau)\mu(S-C)^{-1}e - \tau\lambda$, avec $\xi \in \mathbb{R}^p$ quelconque,

$$\begin{aligned}
& \nabla\psi_{\mu}^{\sigma,\omega}(z)^{\top}d + d^{\top}\nabla^2\psi_{\mu}^{\sigma,\omega}(z)d \\
&= (d^x)^{\top}(\nabla_{xx}^2\ell(x, \tilde{\lambda}, \xi) - M)d^x \\
&\quad + (1 + \tau)(\nabla c d^x - d^s)^{\top}(\mu(S - C)^{-2} - (S - C)^{-1}\Lambda)(\nabla c d^x - d^s) \\
&\quad + (1 + \tau)(d^s)^{\top}(\mu S^{-2} - (\sigma I - \Lambda)S^{-1})d^s \\
&\quad + (1 + \tau)d^y(\mu Y^{-2} - Y^{-1}(\omega I - D\xi))d^y \\
&\quad + \tau(d^\lambda)^{\top}(\mu\Lambda^{-2} - (S - C)\Lambda^{-1})d^\lambda \\
&\quad + \tau(d^\lambda)^{\top}(\mu(\sigma I - \Lambda)^{-2} - (\sigma I - \Lambda)^{-1}S)d^\lambda \\
&\quad + \tau(d^\xi)^{\top}((\omega I - D\xi)^{-2} - Y(\omega I - D\xi)^{-1})d^\xi \\
&= (d^x)^{\top}(\nabla_{xx}^2\ell(x, \tilde{\lambda}, \xi) - \hat{M}_{\mu}^{\sigma,\omega})d^x \\
&\quad + (d^x)^{\top}(\hat{M}_{\mu}^{\sigma,\omega} - M)d^x \\
&\quad + (1 + \tau)(\nabla c d^x - d^s)^{\top}(\mu(S - C)^{-2} - (S - C)^{-1}\Lambda)(\nabla c d^x - d^s) \\
&\quad + (1 + \tau)(d^s)^{\top}(\mu S^{-2} - (\sigma I - \Lambda)S^{-1})d^s \\
&\quad + (1 + \tau)d^y(\mu Y^{-2} - Y^{-1}(\omega I - D\xi))d^y \\
&\quad + \tau(d^\lambda)^{\top}(\mu\Lambda^{-2} - (S - C)\Lambda^{-1})d^\lambda \\
&\quad + \tau(d^\lambda)^{\top}(\mu(\sigma I - \Lambda)^{-2} - (\sigma I - \Lambda)^{-1}S)d^\lambda \\
&\quad + \tau(d^\xi)^{\top}((\omega I - D\xi)^{-2} - Y(\omega I - D\xi)^{-1})d^\xi.
\end{aligned}$$

Remarquons qu'on a, d'après l'inégalité de Cauchy-Schwarz et l'estimation (3.54)

$$(d^x)^{\top}(\hat{M}_{\mu}^{\sigma,\omega} - M)d^x \leq o(\|d^x\|^2) \leq o(\|d\|^2).$$

Notons aussi que si $z \rightarrow \hat{z}_{\mu}$ alors $\nabla_{xx}^2\ell(x, \tilde{\lambda}, \xi) \rightarrow \hat{M}_{\mu}^{\sigma,\omega}$. Ainsi, avec l'estimation (3.54) et le fait que $z \rightarrow \hat{z}_{\mu}$, on obtient

$$\nabla\psi_{\mu}^{\sigma,\omega}(z)^{\top}d + d^{\top}\nabla^2\psi_{\mu}^{\sigma,\omega}(z)d \leq o(\|d\|^2), \quad (3.64)$$

et donc

$$\psi_{\mu}^{\sigma,\omega}(z + d) - \psi_{\mu}^{\sigma,\omega}(z) - \omega_a \nabla\psi_{\mu}^{\sigma,\omega}(z)^{\top}d \leq \left(\frac{1}{2} - \omega_a\right) \nabla\psi_{\mu}^{\sigma,\omega}(z)^{\top}d + o(\|d\|^2).$$

Montrons maintenant qu'il existe une constante positive K' telle que

$$\nabla\psi_{\mu}^{\sigma,\omega}(z)^{\top}d \leq -K'\|d\|^2.$$

Avec les égalités (3.59), (3.60) et (3.61), on obtient les inégalités suivantes

$$\begin{aligned}
2\tau(d^\lambda)^{\top}(\nabla c^{\top}d^x - d^s) &\leq \frac{1 + 2\tau}{2} \|\Lambda^{1/2}(S-C)^{-1/2}(\nabla c^{\top}d^x - d^s)\|^2 \\
&\quad + \frac{2\tau^2}{1 + 2\tau} \|(S-C)^{1/2}\Lambda^{-1/2}d^\lambda\|^2,
\end{aligned}$$

et

$$2\tau(d^\lambda)^{\top}d^s \leq \frac{1 + 2\tau}{2} \|S^{-1/2}(\sigma I - \Lambda)^{1/2}d^s\|^2 + \frac{2\tau^2}{1 + 2\tau} \|S^{1/2}(\sigma I - \Lambda)^{-1/2}d^\lambda\|^2,$$

et

$$\begin{aligned} 2\tau(\mathbf{d}^\xi)^\top \mathbf{D} \mathbf{d}^y &\leq \frac{1+2\tau}{2} \|\mathbf{Y}^{-1/2}(\omega \mathbf{I} - \mathbf{D}\Xi)^{1/2} \mathbf{d}^y\|^2 \\ &\quad + \frac{2\tau^2}{1+2\tau} \|\mathbf{Y}^{1/2}(\omega \mathbf{I} - \mathbf{D}\Xi)^{-1/2} \mathbf{D} \mathbf{d}^\xi\|^2. \end{aligned} \quad (3.65)$$

En combinant ces trois inégalités avec (3.62) et en utilisant (3.56) on obtient

$$\begin{aligned} \nabla \psi_\mu^{\sigma,\omega}(\mathbf{z})^\top \mathbf{d} &= -(\mathbf{d}^x)^\top \mathbf{M} \mathbf{d}^x - \frac{1}{2} \|\mathbf{Y}^{-1/2}(\omega \mathbf{I} - \mathbf{D}\Xi)^{1/2} \mathbf{d}^y\|^2 \\ &\quad - \frac{1}{2} \|\Lambda^{1/2}(\mathbf{S}-\mathbf{C})^{-1/2}(\nabla \mathbf{c}^\top \mathbf{d}^x - \mathbf{d}^s)\|^2 - \frac{1}{2} \|(\sigma \mathbf{I} - \Lambda)^{1/2} \mathbf{S}^{-1/2} \mathbf{d}^s\|^2 \\ &\quad - \frac{\tau}{1+2\tau} \|(\mathbf{S}-\mathbf{C})^{1/2} \Lambda^{-1/2} \mathbf{d}^\lambda\|^2 - \frac{\tau}{1+2\tau} \|\mathbf{S}^{1/2}(\sigma \mathbf{I} - \Lambda)^{-1/2} \mathbf{d}^\lambda\|^2 \\ &\quad - \frac{\tau}{1+2\tau} \|\mathbf{Y}^{1/2}(\omega \mathbf{I} - \mathbf{D}\Xi)^{-1/2} \mathbf{D} \mathbf{d}^\xi\|^2 \\ &\leq -\mathbf{K}'_1 \|\mathbf{d}^x\|^2 - \mathbf{K}'_2 \|\nabla \mathbf{c}^\top \mathbf{d}^x - \mathbf{d}^s\|^2 - \mathbf{K}'_3 \|\mathbf{d}^s\|^2 - \mathbf{K}'_4 \|\mathbf{d}^y\|^2 \\ &\quad - \mathbf{K}'_5 \|\mathbf{d}^\lambda\|^2 - \mathbf{K}'_6 \|\mathbf{d}^\xi\|^2. \end{aligned}$$

Minorons le terme $\|\nabla \mathbf{c}^\top \mathbf{d}^x - \mathbf{d}^s\|^2$. Pour tout $\epsilon > 0$, on

$$\begin{aligned} \|\nabla \mathbf{c}^\top \mathbf{d}^x - \mathbf{d}^s\|^2 &= \|\nabla \mathbf{c}^\top \mathbf{d}^x\|^2 - 2(\nabla \mathbf{c}^\top \mathbf{d}^x)^\top \mathbf{d}^s + \|\mathbf{d}^s\|^2 \\ &\geq \|\nabla \mathbf{c}^\top \mathbf{d}^x\|^2 - (1+\epsilon) \|\nabla \mathbf{c}^\top \mathbf{d}^x\|^2 - \frac{1}{1+\epsilon} \|\mathbf{d}^s\|^2 + \|\mathbf{d}^s\|^2 \\ &\geq -\epsilon \|\nabla \mathbf{c}^\top\|^2 \|\mathbf{d}^x\|^2 + \frac{\epsilon}{1+\epsilon} \|\mathbf{d}^s\|^2. \end{aligned}$$

Posons $\epsilon := \mathbf{K}'_1 / (2\mathbf{K}'_2 \|\nabla \mathbf{c}\|^2)$. On a alors la majoration

$$\begin{aligned} \nabla \psi_\mu^{\sigma,\omega}(\mathbf{z})^\top \mathbf{d} &\leq -\frac{\mathbf{K}'_1}{2} \|\mathbf{d}^x\|^2 - \left(\mathbf{K}'_3 + \frac{\epsilon}{1+\epsilon} \mathbf{K}'_2 \right) \|\mathbf{d}^s\|^2 - \mathbf{K}'_4 \|\mathbf{d}^y\|^2 \\ &\quad - \mathbf{K}'_5 \|\mathbf{d}^\lambda\|^2 - \mathbf{K}'_6 \|\mathbf{d}^\xi\|^2 \\ &\leq -\mathbf{K}' \|\mathbf{d}\|^2. \end{aligned}$$

On obtient donc finalement que, pour des valeurs de k assez grandes,

$$\psi_\mu^{\sigma,\omega}(\mathbf{z} + \mathbf{d}) - \psi_\mu^{\sigma,\omega}(\mathbf{z}) - \omega_a \nabla \psi_\mu^{\sigma,\omega}(\mathbf{z})^\top \mathbf{d} \leq -\mathbf{K}' \|\mathbf{d}\|^2 + o(\|\mathbf{d}\|^2).$$

Le pas unité est donc asymptotiquement accepté dans la recherche linéaire. \square

On donne dans la proposition suivante une condition nécessaire et suffisante pour la convergence q-superlinéaire de l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$.

Proposition 3.10.2 *Supposons l'hypothèse H_1 satisfaite et que f et c soient deux fois différentiables en \hat{x}_μ . Supposons que la suite $\{z_k\}$ générée par l'algorithme $\mathcal{A}_\mu^{\sigma,\omega}$ converge vers \hat{z}_μ et que, pour k suffisamment grand, le pas unité $\alpha_k = 1$ est accepté par la recherche linéaire. Alors $\{z_k\}$ converge q-superlinéairement vers \hat{z}_μ si et seulement si*

$$(\mathbf{M}_k - \hat{\mathbf{M}}_\mu^{\sigma,\omega}) \mathbf{d}_k^x = o(\|\mathbf{d}_k\|). \quad (3.66)$$

Preuve. Notons $\mathcal{M}(z)$ la matrice jacobienne du système de Karush-Kuhn-Tucker perturbé (3.8) en z :

$$\mathcal{M}(z) = \begin{pmatrix} M & 0 & 0 & \nabla c & -A^\top \\ -\Lambda \nabla c^\top & \Lambda & 0 & S-C & 0 \\ 0 & \sigma I - \Lambda & 0 & -S & 0 \\ 0 & 0 & (\omega I - D\Xi) & 0 & -YD \\ A & 0 & D & 0 & 0 \end{pmatrix}$$

Un développement de rang un du membre droit de (3.15) en la solution \hat{z}_μ donne

$$\mathcal{M}(z_k) d_k = -\mathcal{M}(\hat{z}_\mu)(z_k - \hat{z}_\mu) + o(\|z_k - \hat{z}_\mu\|).$$

En soustrayant $\mathcal{M}(\hat{z}_\mu) d_k$ des deux membres et en supposant le pas unité accepté, on obtient

$$(\mathcal{M}(z_k) - \mathcal{M}(\hat{z}_\mu)) d_k = -\mathcal{M}(\hat{z}_\mu)(z_{k+1} - \hat{z}_\mu) + o(\|z_k - \hat{z}_\mu\|). \quad (3.67)$$

Supposons maintenant que $\{z_k\}$ converge q -superlinéairement. Alors, le membre droit de (3.67) est d'ordre $o(\|z_k - \hat{z}_\mu\|)$ et donc d'ordre $o(\|d_k\|)$. En évaluant la quantité $(\mathcal{M}(z_k) - \mathcal{M}(\hat{z}_\mu)) d_k$, on obtient

$$(\mathcal{M}_k - \hat{\mathcal{M}}_\mu^{\sigma, \omega}) d_k^x + o(\|d_k^s\|) + o(\|d_k^y\|) + o(\|d_k^\lambda\|) + o(\|d_k^\xi\|) = o(\|d_k\|),$$

et on en déduit alors (3.66).

Montrons maintenant l'implication inverse. D'après (3.66) et la convergence de $\{z_k\}$ vers \hat{z}_μ , le membre gauche de (3.67) est en $o(\|d_k\|)$. On a donc

$$-\mathcal{M}(\hat{z}_\mu)(z_{k+1} - \hat{z}_\mu) + o(\|z_k - \hat{z}_\mu\|) = o(\|d_k\|).$$

La non-singularité de $\mathcal{M}(\hat{z}_\mu)$ nous donne

$$z_{k+1} - \hat{z}_\mu = o(\|z_k - \hat{z}_\mu\|) + o(\|d_k\|).$$

Le pas unité étant accepté, on a $d_k = (z_{k+1} - \hat{z}_\mu) - (z_k - \hat{z}_\mu)$. On obtient finalement

$$z_{k+1} - \hat{z}_\mu = o(\|z_k - \hat{z}_\mu\|).$$

□

Théorème 3.10.3 *Supposons l'hypothèse H_1 satisfaite et que f et c soient des fonctions $C^{1,1}$ deux fois continûment différentiables dans un voisinage de \hat{x}_μ avec un hessien localement radialement Lipschitzien en \hat{x}_μ . Supposons que l'algorithme $\mathcal{A}_\mu^{\sigma, \omega}$ ne s'arrête pas à l'étape 1. Alors, la suite $\{z_k\}$ générée par l'algorithme converge vers \hat{z}_μ q -superlinéairement et, pour k suffisamment grand, le pas unité $\alpha_k = 1$ est accepté par la recherche linéaire.*

Preuve. On déduit de la preuve du théorème 4.4 dans [4] que la convergence de la série $\sum_{k \geq 1} \|z_k - \hat{z}_\mu\|$ (théorème 3.9.6), la continuité locale radialement

Lipschitzienne des hessiens de f et de c , les propriétés des formules de BFGS (voir [55, 12]) entraînent

$$(M_k - \hat{M}_\mu^{\sigma, \omega})d_k^x = o(\|d_k^x\|).$$

Cette estimation entraîne l'égalité $(d_k^x)^\top (M_k - \hat{M}_\mu^{\sigma, \omega})d_k^x = o(\|d_k^x\|^2)$. En effet,

$$(d_k^x)^\top (M_k - \hat{M}_\mu^{\sigma, \omega})d_k^x \leq \|d_k^x\| \| (M_k - \hat{M}_\mu^{\sigma, \omega})d_k^x \| \leq \epsilon_k (\|d_k^x\|^2),$$

où $\{\epsilon_k\}$ est une suite de réels tendant vers 0. Cette égalité entraîne donc l'inégalité (3.54) et l'acceptation du pas unitaire (proposition 3.10.1). Cela implique aussi qu'on a l'égalité (3.66), et donc la convergence q -superlinéaire (proposition 3.10.2). \square

3.11 Algorithme global \mathcal{A}

On peut maintenant établir une itération de l'algorithme résolvant le problème (\mathcal{P}) . Les constantes $\underline{\sigma} > 0$ (mise à jour de σ) et $\underline{\omega} > 0$ (mise à jour de ω) sont données indépendamment de l'indice de l'itération. Ces valeurs sont choisies telles que

$$\underline{\sigma} > 0 \text{ et } \underline{\omega} > 0. \quad (3.68)$$

Au début de la $j^{\text{ième}}$ itération externe, une approximation $z_1^j := (x_1^j, s_1^j, y_1^j, \lambda_1^j, \xi_1^j) \in \mathcal{Z}$ de la solution $\hat{z}_{\sigma, \omega} := (\hat{x}_{\sigma, \omega}, \hat{s}_{\sigma, \omega}, \hat{y}_{\sigma, \omega}, \hat{\lambda}_{\sigma, \omega}, \hat{\xi}_{\sigma, \omega})$ de (3.7) et une matrice définie positive M_1^j approchant le hessien du lagrangien sont supposées disponibles. On suppose connues les valeurs du paramètre de perturbation $\mu^j > 0$, celles des seuils de précision $\epsilon^j := (\epsilon_1^j, \epsilon_c^j, \epsilon_s^j, \epsilon_y^j) > 0$ et celles des paramètres de pénalité σ^{j-1} et ω^{j-1} utilisés à l'itération précédente.

ALGORITHME \mathcal{A} pour résoudre le problème (\mathcal{P}) (une itération externe)

1. A partir du point initial z_1^j , utiliser l'algorithme $\mathcal{A}_\mu^{\sigma, \omega}$ jusqu'à l'obtention d'un point $z^j := (x^j, s^j, y^j, \lambda^j, \xi^j) \in \mathcal{Z}$ satisfaisant

$$\begin{cases} \|\nabla f(x^j) + \nabla c(x^j)\lambda^j - A^\top \xi^j\| \leq \epsilon_1^j, \\ \|(S^j - C(x^j))\lambda^j - \mu^j e_m\| \leq \epsilon_c^j, \\ \|S^j(\sigma^j e_m - \lambda^j) - \mu^j e_m\| \leq \epsilon_s^j, \\ \|Y_j(\omega^j e_p - D\xi^j) - \mu^j e_p\| \leq \epsilon_y^j, \\ Ax^j + Dy^j = b. \end{cases} \quad (3.69)$$

2. Mise à jour des vecteurs de pénalité :

- 2.0. si $\sigma^j \geq \|\lambda^j\|_\infty + \underline{\sigma}$, alors $\sigma^{j+1} := \sigma^j$,
 sinon $\sigma^{j+1} := \max(1.1\sigma^j, \|\lambda^j\|_\infty + \underline{\sigma})$.
- 2.1. si $\omega^j \geq \|D\|_\infty \|\xi^j\|_\infty + \underline{\omega}$, alors $\omega^{j+1} := \omega^j$,
 sinon $\omega^{j+1} := \max(1.1\omega^j, \|D\|_\infty \|\xi^j\|_\infty + \underline{\omega})$.

2. Définir un nouveau paramètre barrière $\mu^{j+1} > 0$, de nouveaux seuils de précision $\epsilon^{j+1} := (\epsilon_1^{j+1}, \epsilon_c^{j+1}, \epsilon_s^{j+1}, \epsilon_y^{j+1}) > 0$, tels que $\{\mu^j\}$ et $\{\epsilon^j\}$ convergent vers zéro quand $j \rightarrow \infty$. Choisir un nouvel itéré de départ $z_1^{j+1} \in \mathcal{Z}$ pour l'itération externe suivante. Choisir aussi une matrice définie positive M_1^{j+1} .

On remarque que la mise à jour des paramètres de pénalité σ^j et ω^j (étape 1) entraîne l'équivalence de la bornitude de la suite $\{\sigma^j\}$ avec celle de la suite $\{\lambda^j\}$ et l'équivalence de la bornitude de la suite $\{\omega^j\}$ avec celle de la suite $\{\xi^j\}$. Ainsi, pour j suffisamment grand, les suites $\{\sigma^j\}$ et $\{\omega^j\}$ sont stationnaires.

Lemme 3.11.1 *On a les équivalences suivantes :*

- (i) *La suite croissante des paramètres de pénalité $\{\sigma^j\}$ est bornée si et seulement si la suite des multiplicateurs $\{\lambda^j\}$ est bornée. De plus, si pour un indice $i \in \{1, \dots, m\}$, la suite $\{\lambda_i^j\}$ est bornée, alors la suite $\{s_i^j\}$ converge vers 0.*
- (ii) *La suite croissante des paramètres de pénalité $\{\omega^j\}$ est bornée si et seulement si la suite des multiplicateurs $\{\xi^j\}$ est bornée. De plus, si pour un indice $i \in \{1, \dots, p\}$, la suite $\{\xi_i^j\}$ est bornée, alors la suite $\{y_i^j\}$ converge vers 0.*

Preuve. Montrons le (i). L'équivalence sur la bornitude des suites $\{\sigma^j\}$ et $\{\lambda^j\}$ découle directement de la mise à jour à l'étape 1 des paramètres de pénalité dans l'algorithme \mathcal{A} . Supposons maintenant qu'il existe un indice i tel que la suite $\{\lambda_i^j\}$ soit bornée. Si $\{\sigma^j\}$ est bornée, alors on a, d'après la règle de mise à jour de σ^j , pour j suffisamment grand : $\sigma^j - \lambda_i^j \geq \bar{\sigma} > 0$. Donc, d'après le troisième critère d'arrêt (3.69), on a $s_i^j \rightarrow 0$ quand $j \rightarrow \infty$. Si $\{\sigma^j\}$ est non bornée, alors, d'après la règle de mise à jour de σ^j , $\sigma^j - \lambda_i^j \rightarrow \infty$ quand $j \rightarrow \infty$. Donc, d'après le troisième critère d'arrêt (3.69), on a $s_i^j \rightarrow 0$ quand $j \rightarrow \infty$.

Montrons le (ii). L'équivalence sur la bornitude des suites $\{\omega^j\}$ et $\{\xi^j\}$ découle directement de la mise à jour à l'étape 1 des paramètres de pénalité dans l'algorithme \mathcal{A} . Supposons maintenant qu'il existe un indice i tel que la suite $\{\xi_i^j\}$ soit bornée. Si $\{\omega^j\}$ est bornée, alors on a, d'après la règle de mise à jour de ω^j , pour j suffisamment grand : $\omega^j - d_i \xi_i^j \geq \bar{\omega} > 0$. Donc, d'après le quatrième critère d'arrêt (3.69), on a $y_i^j \rightarrow 0$ quand $j \rightarrow \infty$. Si $\{\omega^j\}$ est non bornée, alors, d'après la règle de mise à jour de ω^j , $\omega^j - d_i \xi_i^j \rightarrow \infty$ quand $j \rightarrow \infty$. Donc, d'après le troisième critère d'arrêt (3.69), on a $y_i^j \rightarrow 0$ quand $j \rightarrow \infty$. \square

3.12 Convergence de l'algorithme \mathcal{A}

On va montrer dans cette section que les itérés générés par l'algorithme \mathcal{A} convergent vers la solution du problème initial (\mathcal{P}). Le lemme suivant donne une estimation de la valeur de la fonction objectif pondérée à la fin d'une itération externe.

Lemme 3.12.1 *Supposons l'hypothèse H_1 satisfaite. Si z^j satisfait (3.69), alors pour tout $x \in \mathbb{R}^n$ on a*

$$\ell(x^j, \lambda^j, \xi^j) \leq f(x^j) + \sigma^j e_m^\top s^j + \omega^j e_p^\top y^j \quad (3.70)$$

$$\begin{aligned} &\leq \ell(x, \lambda^j, \xi^j) + m^{\frac{1}{2}}(\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}}\epsilon_y^j \\ &\quad + (2m + p)\mu^j + \|x^j - x\|\epsilon_l^j \end{aligned} \quad (3.71)$$

Preuve. La convexité du lagrangien nous donne, pour tout $x \in \mathbb{R}^n$

$$\ell(x^j, \lambda^j, \xi^j) + \nabla_x \ell(x^j, \lambda^j, \xi^j)^\top (x - x^j) \leq \ell(x, \lambda^j, \xi^j).$$

Ainsi,

$$f(x^j) + c(x^j)^\top \lambda^j + (\xi^j)^\top (b - Ax^j) \leq \ell(x, \lambda^j, \xi^j) + \nabla_x \ell(x^j, \lambda^j, \xi^j)^\top (x^j - x).$$

Par la première inégalité de (3.69), on obtient

$$f(x^j) \leq -c(x^j)^\top \lambda^j - (\xi^j)^\top (b - Ax^j) + \ell(x, \lambda^j, \xi^j) + \|x^j - x\|\epsilon_l^j.$$

On a en utilisant le second critère de (3.69)

$$\begin{aligned} -c(x^j)^\top \lambda^j &= (s^j - c(x^j))^\top \lambda^j - (\lambda^j)^\top s^j \\ &= e_m^\top ((S^j - C(x^j))\lambda^j - \mu^j e_m) + m\mu^j - (\lambda^j)^\top s^j \\ &\leq m^{\frac{1}{2}}\epsilon_c^j + m\mu^j - (\lambda^j)^\top s^j. \end{aligned}$$

De même, on a avec le troisième critère de (3.69)

$$\begin{aligned} -(\lambda^j)^\top s^j &= (\lambda^j)^\top (\sigma^j e_m - s^j) - \sigma^j e_m^\top s^j \\ &= e_m^\top (S^j (\sigma^j e_m - \lambda^j) - \mu^j e_m) + m\mu^j - \sigma^j e_m^\top s^j \\ &\leq m^{\frac{1}{2}}\epsilon_s^j + m\mu^j - \sigma^j e_m^\top s^j. \end{aligned}$$

De même, avec le quatrième critère de (3.69) et la contrainte linéaire d'égalité,

$$\begin{aligned} -(\xi^j)^\top (b - Ax^j) &= -(\xi^j)^\top Dy^j \\ &= Y^j (\omega^j e_p - D\xi^j) - \omega^j e_p^\top y^j \\ &= e_p^\top (Y^j (\omega^j e_p - D\xi^j) - \mu^j e_p) + p\mu^j - \omega^j e_p^\top y^j \\ &\leq p^{\frac{1}{2}}\epsilon_y^j + p\mu^j - \omega^j e_p^\top y^j. \end{aligned}$$

En combinant ces inégalités, on obtient

$$\begin{aligned} f(x^j) &\leq -\sigma^j e_m^\top s^j - \omega^j e_p^\top y^j \\ &\quad + \ell(x, \lambda^j, \xi^j) + m^{\frac{1}{2}}(\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}}\epsilon_y^j + (2m + p)\mu^j + \|x^j - x\|\epsilon_l^j \end{aligned}$$

ce qui montre l'inégalité (3.71). L'inégalité (3.70) se déduit des inégalités suivantes

$$(\omega^j e_p - D\xi^j, y^j, s^j - c(x^j), \lambda^j, \sigma^j e_m - \lambda^j, s^j) > 0.$$

□

On rappelle la condition de Slater pour le problème (\mathcal{P}) :

$$\begin{cases} \exists x \in \mathbb{R}^n : Ax = b \text{ et } c(x) < 0 \\ A \text{ est de rang } p. \end{cases} \quad (3.72)$$

La deuxième condition signifie que les lignes de la matrice A sont linéairement indépendantes. Dans le cas de problème convexe, la condition de Slater est équivalente à celle de Mangasarian-Fromovitz :

$$\forall \alpha \in \mathbb{R}^p \text{ et } \beta \in \mathbb{R}_+^m, \sum_{i \in I} \beta_i \nabla c_i(\bar{x}) - A^\top \alpha = 0 \Rightarrow \alpha = 0 \text{ et } \beta = 0, \quad (3.73)$$

où I est l'ensemble des indices i pour lesquels les contraintes c_i sont actives en \bar{x} . Le théorème suivant établit un lien entre la bornitude des $\{(x_j, \lambda_j, \xi_j)\}$ et la réalisabilité du problème.

Théorème 3.12.2 *Supposons l'hypothèse H_1 satisfaite et que f et c soient des fonctions $C^{1,1}$. L'algorithme A génère alors une suite $\{z^j\}$ satisfaisant les critères (3.69) et une des situations suivantes est rencontrée :*

- (i) *la suite $\{x^j\}$ est non bornée. Alors la suite des multiplicateurs $\{(\lambda^j, \xi^j)\}$ est non bornée et le problème (\mathcal{P}) n'est pas réalisable.*
- (ii) *la suite $\{x^j\}$ est bornée et la suite $\{(\lambda^j, \xi^j)\}$ est non bornée. Alors les suites $\{s^j\}$ et $\{y^j\}$ sont bornées. Dans ce cas, la condition de Slater n'est pas satisfaite.*
- (iii) *les deux suites $\{x^j\}$ et $\{(\lambda^j, \xi^j)\}$ sont bornées. Alors, $\{s^j\}$ tend vers zéro, $\{y^j\}$ tend vers zéro et tout point d'accumulation de $\{(x^j, \lambda^j, \xi^j)\}$ est une solution primale-duale du problème (\mathcal{P}).*

Preuve. Montrons tout d'abord le résultat du (i). Par le lemme précédent, on a

$$\begin{aligned} f(x^j) - f(x) &\leq -\sigma^j e_m^\top s^j - \omega^j e_p^\top y^j \\ &\quad + (\lambda^j)^\top c(x) + (\xi^j)^\top (b - Ax) \\ &\quad + m^{\frac{1}{2}}(\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}}\epsilon_y^j + (2m + p)\mu^j + \|x^j - x\|\epsilon_l^j. \end{aligned}$$

Ainsi, en additionnant $\underline{\sigma} e_m^\top (c(x^j) - c(x)) + \underline{\omega} e_p^\top D^{-1}A(x - x^j)$, on a

$$\begin{aligned} f(x^j) - f(x) + \underline{\sigma} e_m^\top (c(x^j) - c(x)) + \underline{\omega} e_p^\top D^{-1}A(x - x^j) \\ \leq \underline{\sigma} e_m^\top (c(x^j) - c(x)) \\ + \underline{\omega} e_p^\top D^{-1}A(x - x^j) \\ - \sigma^j e_m^\top s^j - \omega^j e_p^\top y^j \\ + (\lambda^j)^\top c(x) + (\xi^j)^\top (b - Ax) \\ + m^{\frac{1}{2}}(\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}}\epsilon_y^j \\ + (2m + p)\mu^j + \|x^j - x\|\epsilon_l^j, \end{aligned}$$

puis, en réorganisant les termes,

$$\begin{aligned}
 & f(x^j) - f(x) + \underline{\sigma} e_m^\top (c(x^j) - c(x)) + \underline{\omega} e_p^\top D^{-1} A(x - x^j) \\
 & \leq (\lambda^j - \underline{\sigma} e_m)^\top c(x) \\
 & \quad + (\xi^j - \underline{\omega} D^{-1} e_p)^\top (b - Ax) \\
 & \quad - \sigma^j e_m^\top s^j + \underline{\sigma} e_m^\top c(x^j) \\
 & \quad + \underline{\omega} e_p^\top D^{-1} (b - Ax^j) - \omega^j e_p^\top y^j \\
 & \quad + m^{\frac{1}{2}} (\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}} \epsilon_y^j \\
 & \quad + (2m + p) \mu^j + \|x^j - x\| \epsilon_l^j
 \end{aligned}$$

puis, en ajoutant et retranchant le terme $\underline{\sigma} e_m^\top s^j$ et en remarquant que

$$D^{-1}(b - Ax^j) = y^j,$$

on obtient

$$\begin{aligned}
 & f(x^j) - f(x) + \underline{\sigma} e_m^\top (c(x^j) - c(x)) + \underline{\omega} e_p^\top D^{-1} A(x - x^j) \\
 & \leq (\lambda^j - \underline{\sigma} e_m)^\top c(x) \\
 & \quad + (\xi^j - \underline{\omega} D^{-1} e_p)^\top (b - Ax) \\
 & \quad + (\underline{\sigma} - \sigma^j) e_m^\top s^j + \underline{\sigma} e_m^\top (c(x^j) - s^j) \\
 & \quad + (\underline{\omega} - \omega) e_p^\top y^j \\
 & \quad + m^{\frac{1}{2}} (\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}} \epsilon_y^j \\
 & \quad + (2m + p) \mu^j + \|x^j - x\| \epsilon_l^j.
 \end{aligned}$$

Comme $(\omega^j - \underline{\omega}, s^j - c(x^j), \underline{\sigma}, \sigma^j - \underline{\sigma}, y^j) > 0$, on peut éliminer les termes négatifs de l'inégalité précédente :

$$\begin{aligned}
 & f(x^j) - f(x) + \underline{\sigma} e_m^\top (c(x^j) - c(x)) + \underline{\omega} e_p^\top D^{-1} A(x - x^j) \\
 & \leq (\lambda^j - \underline{\sigma} e_m)^\top c(x) \\
 & \quad + (\xi^j - \underline{\omega} D^{-1} e_p)^\top (b - Ax) \\
 & \quad + m^{\frac{1}{2}} (\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}} \epsilon_y^j \\
 & \quad + (2m + p) \mu^j + \|x^j - x\| \epsilon_l^j.
 \end{aligned} \tag{3.74}$$

La suite $\{x^j\}$ étant non bornée, il existe un sous-ensemble J d'indices tels que $\|x^j\| \rightarrow \infty$ quand $j \rightarrow \infty$ dans J . On définit les deux suites $t^j := \|x^j - x\|$ et $d^j := (x^j - x)/t^j$. On remarque que par construction $x^j = x + t^j d^j$, $t^j \rightarrow \infty$ quand $j \rightarrow \infty$ dans J et $d^j \rightarrow d \neq 0$ pour une sous-suite $J' \subset J$. En divisant les deux

côtés de (3.74) par t^j , on obtient

$$\begin{aligned} & \frac{f(x^j) - f(x)}{t^j} + \sum_{i=1}^m \underline{\sigma} \frac{(c_i(x^j) - c_i(x))}{t^j} + \sum_{i=1}^p \frac{\underline{\omega}}{d_i} \frac{a_i^\top (x^j - x)}{t^j} \\ & \leq \sum_{i=1}^m (\lambda_i^j - \underline{\sigma}) \frac{c_i(x)}{t^j} \\ & \quad + \sum_{i=1}^p \left(\xi_i^j - \frac{\underline{\omega}}{d_i} \right) \frac{(b_i - a_i^\top x)}{t^j} \\ & \quad + \left(1 + \frac{1}{t^j}\right) \epsilon^j. \end{aligned}$$

avec $\epsilon^j \rightarrow 0$ quand $j \rightarrow \infty$. Ainsi en passant à la limite $j \rightarrow \infty$ et par forte convexité du lagrangien, on a

$$\lim_{j \rightarrow \infty} \left(\sum_{i=1}^m \lambda_i^j \frac{c_i(x)}{t^j} + \sum_{i=1}^p \xi_i^j \frac{(b_i - a_i^\top x)}{t^j} \right) = +\infty. \quad (3.75)$$

Montrons le résultat du (i) par contradiction. Supposons que la suite $\{(\lambda^j, \xi^j)\}$ soit bornée. Alors la limite (3.75) serait différente de l'infini ce qui est faux. La suite des multiplicateurs est donc non bornée. Supposons maintenant qu'il existe un point x réalisable, c'est à dire tel que $c(x) \leq 0$ et $Ax = b$. Alors, comme $\lambda^j \geq 0$, la limite ne pourrait être l'infini. Le problème (\mathcal{P}) est donc non réalisable dans ce cas.

Montrons le (ii). Supposons que la suite $\{x^j\}$ soit bornée. En considérant l'égalité linéaire, on obtient la bornitude de la suite $\{y^j\}$. Montrons par contradiction que la suite $\{s_i^j\}$ est bornée. Supposons donc que cette suite ne soit pas bornée. Il existe donc i dans $\{1, \dots, m\}$ et un sous-ensemble J tel que $s_i^j \rightarrow \infty$ quand $j \rightarrow \infty$ dans J . En utilisant le troisième critère d'arrêt de (3.69) on a $s_i^j(\sigma^j - \lambda_i^j) \rightarrow 0$ quand $j \rightarrow \infty$ dans J . Donc $\sigma^j - \lambda_i^j \rightarrow 0$ quand $j \rightarrow \infty$ dans J . Ainsi, pour j assez grand dans J , on a $\sigma^j - \lambda_i^j < \underline{\sigma}$. La suite $\{\lambda_i^j\}_{j \in J}$ est donc non bornée. En considérant le second critère d'arrêt, on a $s_i^j - c_i(x^j)\lambda_i^j \rightarrow 0$ quand $j \rightarrow \infty$ dans J . La suite $\{x^j\}$ étant bornée et les fonctions c continues, la suite $\{s_i^j\}_{j \in J}$ est donc bornée, ce qui contredit l'hypothèse sur la non bornitude de la suite $\{s_i^j\}_{j \in J}$. On a montré la première partie du (ii).

Montrons la seconde partie du (ii). On suppose que les suites $\{x^j\}$, $\{s^j\}$ et $\{y^j\}$ sont bornées et que la suite $\{(\lambda^j, \xi^j)\}$ est non bornée. Il existe un sous-ensemble J tel que $\|(\lambda^j, \xi^j)\| \rightarrow \infty$ quand $j \rightarrow \infty$ dans J . Il existe alors une sous suite d'indices J' dans J telle que la suite $\left\{ \frac{(\lambda^j, \xi^j)}{\|(\lambda^j, \xi^j)\|} \right\}_{j \in J'}$ tende vers $(\bar{\lambda}, \bar{\xi})$ avec $\|(\bar{\lambda}, \bar{\xi})\| = 1$. En utilisant le premier critère (3.69), réécrit de la manière suivante

$$\|\nabla f(x^j) + (\nabla c(x^j), -A^\top) \begin{pmatrix} \lambda^j \\ \xi^j \end{pmatrix}\| \leq \epsilon^j,$$

et en le divisant par $\|(\lambda^j, \xi^j)\|$, on obtient l'égalité suivante après passage à la limite dans J' :

$$\nabla c(\bar{x})\bar{\lambda} - A^\top \bar{\xi} = 0. \quad (3.76)$$

En utilisant le second critère, en le divisant par $\|(\lambda^j, \xi^j)\|$ et après passage à la limite dans J' , on obtient

$$(\bar{S} - C(\bar{x}))\bar{\lambda} = 0.$$

Comme $(\bar{s}, \bar{s} - c(\bar{x}), \bar{\lambda}) \geq 0$ et d'après l'inégalité précédente, si $\bar{s}_i - c_i(\bar{x}) > 0$ alors $\bar{\lambda}_i = 0$. Notons $I = \{i : c_i(\bar{x}) = \bar{s}_i\}$. L'égalité (3.76) peut s'écrire

$$\sum_{i \in I} \nabla c_i(\bar{x}) \bar{\lambda}_i - A^T \bar{\xi} = 0.$$

Cela montre que la condition de Mangasarian Fromovitz n'est pas satisfaite. Par équivalence dans le cas convexe de cette condition avec celle de Slater, cela montre que l'ensemble $\{x : c(x) \leq \bar{s} \text{ et } Ax = b\}$ n'est pas strictement réalisable. Par positivité de \bar{s} et comme $\{x : c(x) \leq 0 \text{ et } Ax = b\}$ est inclus dans $\{x : c(x) \leq \bar{s} \text{ et } Ax = b\}$, la condition de Slater n'est pas vérifiée, ce qui montre le (ii).

Montrons maintenant le (iii). Supposons que les suites $\{x^j\}$ et $\{(\lambda^j, \xi^j)\}$ soient bornées. Le lemme 3.11.1 entraîne que les suites $\{\sigma^j\}$ et $\{\omega^j\}$ sont constantes à partir d'un certain rang et que $s^j \rightarrow 0$ et $y^j \rightarrow 0$. Notons $(\bar{x}, \bar{s}, \bar{y}, \bar{\lambda}, \bar{\xi})$ un point limite de la suite $\{(x^j, s^j, y^j, \lambda^j, \xi^j)\}$. On a $\bar{s} = 0$ et $\bar{y} = 0$. Par passage à la limite dans les critères (3.69) et comme $z^j \in \mathcal{Z}$, on obtient

$$\begin{cases} \nabla f(\bar{x}) + \nabla c(\bar{x})^T \bar{\lambda} - A^T \bar{\xi} = 0 \\ c(\bar{x}) \leq 0 \\ A\bar{x} = b \\ C(\bar{x})\bar{\lambda} = 0 \\ \bar{\lambda} \geq 0 \end{cases} \quad (3.77)$$

Ainsi $(\bar{x}, \bar{\lambda}, \bar{\xi})$ est une solution primale-duale du problème (\mathcal{P}) . □

Le théorème suivant montre que les itérés générés par l'algorithme convergent vers une solution primale-duale du problème (\mathcal{P}) , sous l'hypothèse que la condition de Slater soit vérifiée.

Théorème 3.12.3 *Supposons l'hypothèse H_1 satisfaite et que f et c soient des fonctions $C^{1,1}$. Soit $\{z^j\}$ la suite générée par l'algorithme \mathcal{A} . Si le problème (\mathcal{P}) est strictement réalisable alors la suite $\{(x^j, \lambda^j, \xi^j)\}$ est bornée, $\{s^j\}$ tend vers zéro, $\{y^j\}$ tend vers zéro et tout point d'accumulation de $\{(x^j, \lambda^j, \xi^j)\}$ est une solution primale-duale du problème (\mathcal{P}) .*

Preuve. Si le problème est strictement réalisable alors il existe $x \in \mathbb{R}^n$ tel que $c(x) < 0$ et $Ax = b$. La condition de Slater est donc vérifiée. Le (i) du théorème 3.12.2 entraîne que la suite $\{x^j\}$ est bornée et le (ii) de ce même théorème entraîne que la suite des multiplicateurs $\{(\lambda^j, \xi^j)\}$ est bornée. En concluant avec le (iii), on obtient le résultat souhaité. □

3.13 Convergence des itérés vers le centre analytique primal-dual

On montre maintenant que le chemin central converge vers un point particulier appelé centre analytique de l'ensemble des solutions, ensemble noté

$\text{opt}(\mathcal{P})$, du problème initial (\mathcal{P}). On a associé au problème initial (\mathcal{P}) le problème pénalisé ($\mathcal{P}^{\sigma, \omega}$). Il serait donc logique de démontrer que les itérés générés convergent vers le centre analytique du problème pénalisé. Mais on a montré que si la condition de Slater est vérifiée, les itérés convergent vers la solution du problème initial (\mathcal{P}). Les variables s^j et y^j tendent alors vers 0. Ainsi, les variables primales convergent donc vers un point particulier de l'ensemble $\text{opt}(\mathcal{P})$ des solutions du problème initial (\mathcal{P}). Les variables duales ne convergent pas vers un point particulier de l'ensemble des solutions du problème dual associé à (\mathcal{P}). On doit en effet considérer les variables duales associées aux contraintes de positivité des variables s^j et y^j . On va donc montrer que les variables duales $(\lambda^j, \sigma e_m - \lambda^j, \omega e_p - D\xi^j)$ convergent vers un point particulier de l'ensemble des solutions du problème dual associé au problème pénalisé ($\mathcal{P}^{\sigma, \omega}$).

On note $\text{val}(\mathcal{P})$ la valeur optimale atteinte sur $\text{opt}(\mathcal{P})$. Par convexité des fonctions f, c_i pour $i \in \{1, \dots, m\}$ et par linéarité de la contrainte d'égalité, $\text{opt}(\mathcal{P})$ est un ensemble convexe. Si $\text{opt}(\mathcal{P})$ est réduit à un point, on définit le centre analytique comme étant ce point. Si tel n'est pas le cas, on définit l'ensemble

$$B = \{i : \exists x \in \text{opt}(\mathcal{P}), c_i(x) < 0\}. \quad (3.78)$$

Le centre analytique de $\text{opt}(\mathcal{P})$ est alors la solution du problème

$$\begin{cases} \max \sum_{i \in B} \log(-c_i(x)), \\ c_B(x) < 0, \\ x \in \text{opt}(\mathcal{P}). \end{cases} \quad (3.79)$$

Notons O_B l'ensemble réalisable du problème (3.79).

Lemme 3.13.1 *Supposons l'hypothèse H_1 satisfaite. L'ensemble B défini par (3.78) est non vide.*

Preuve. Supposons le contraire : $B = \emptyset$. Alors pour tout x dans $\text{opt}(\mathcal{P})$, $c_i(x) = 0$ pour tout i dans $\{1, \dots, m\}$. Prenons deux points x_1 et x_2 de $\text{opt}(\mathcal{P})$. Par convexité de $\text{opt}(\mathcal{P})$, le segment $[x_1, x_2]$ est contenu dans $\text{opt}(\mathcal{P})$. Ainsi, le lagrangien $\ell(x, \lambda, \xi)$ serait constant à $\text{val}(\mathcal{P})$ pour tout x dans $[x_1, x_2]$. Cela est impossible d'après l'hypothèse H_1 sur la forte convexité du lagrangien. Donc B est un ensemble non vide. \square

Lemme 3.13.2 *L'ensemble réalisable O_B du problème (3.79) est non vide.*

Preuve. On sait que, pour tout i dans B , il existe un point noté x^i dans $\text{opt}(\mathcal{P})$ tel que $c_i(x^i) < 0$. On définit le point suivant

$$\bar{x} := \frac{1}{|B|} \sum_{i \in B} x^i,$$

où $|B|$ est le cardinal de l'ensemble B . Par convexité de $\text{opt}(\mathcal{P})$ et comme \bar{x} est une combinaison convexe des points x^i de $\text{opt}(\mathcal{P})$, ce point est dans $\text{opt}(\mathcal{P})$. De

plus, pour tout l dans B ,

$$c_l(\bar{x}) \leq \frac{1}{|B|} \left(\sum_{j \in B, j \neq l} c_l(x^j) + c_l(x^l) \right) < 0.$$

Ainsi, \bar{x} est un point de O_B . Cet ensemble n'est donc pas vide. \square

Supposons que le problème (3.79) vérifie la condition de Slater .

Hypothèse 3.13.3 (H_2) $\exists x \in \mathbb{R}^n$ tel que $Ax = b$ et $c(x) < 0$.

Montrons maintenant que le problème (3.79) a une solution unique.

Lemme 3.13.4 *Supposons les hypothèses H_1 et H_2 satisfaites. Si $\text{opt}(\mathcal{P})$ n'est pas réduit à un seul point, le problème (3.79) a une unique solution.*

Preuve. L'hypothèse H_2 et le théorème 3.12.3 montrent que les itérés convergent vers un point de $\text{opt}(\mathcal{P})$. Les variables primales s^j et y^j tendent donc vers 0. Le problème (3.79) est réalisable d'après le lemme 3.13.2. Prenons x_0 dans O_B . On définit l'ensemble

$$\{x : x \in \text{opt}(\mathcal{P}), c_B(x) < 0, \sum_{i \in B} \log(-c_i(x)) \geq \sum_{i \in B} \log(-c_i(x_0))\}.$$

Cet ensemble est non vide (x_0 est dedans). Par compacité de l'ensemble $\text{opt}(\mathcal{P})$ (lemme 3.3.2 et hypothèse H_1) et par continuité des fonctions \log et $-c_i$ pour $i \in B$, cet ensemble est aussi compact. Donc d'après la proposition 3.2.4 et le corollaire 3.2.5, l'ensemble de solutions du problème (3.79) est non vide et compact. Montrons maintenant que cet ensemble est réduit à un point. Supposons le contraire. Soient x_1 et x_2 deux points solutions de (3.79). Par convexité de $\text{opt}(\mathcal{P})$ et des fonctions c_i , tout point de $[x_1, x_2]$ est dans O_B . Par stricte concavité du \log , les fonctions c_B sont constantes sur ce segment. Les fonctions $(c_i)_{i \notin B}$ le sont aussi sur ce segment (elles sont égales à 0). Comme $[x_1, x_2] \subset \text{opt}(\mathcal{P})$ et que les contraintes c sont constantes sur ce segment, il en est de même pour la fonction f . Tout cela entraîne donc que le lagrangien est constant sur le segment $[x_1, x_2]$. Cela est impossible d'après l'hypothèse H_1 . \square

Considérons maintenant le problème dual associé au problème pénalisé perturbé

$$\begin{cases} \max L(\lambda, \xi, \sigma e_m - \lambda, \omega e_p - D\xi) + \mu \sum_{i=1}^m \log(\lambda_i) + \mu \sum_{i=1}^m \log(\sigma - \lambda_i) \\ \quad \quad \quad + \mu \sum_{i=1}^p \log(\omega - D\xi_i), \\ (\lambda, \sigma e_m - \lambda, \omega e_p - D\xi) > 0, \\ L(\lambda, \sigma e_m - \lambda, \omega e_p - D\xi) > -\infty. \end{cases} \quad (3.80)$$

avec

$$L(\lambda, \xi, \sigma e_m - \lambda, \omega e_p - D\xi) = \min\{\ell(x, s, y, \lambda, \xi, \sigma e_m - \lambda, \omega e_p - D\xi), \\ (x, s, y) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p\}$$

où $\ell : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ est le lagrangien associé au problème pénalisé perturbé. Notons $\text{opt}(D)$ l'ensemble des solutions

duales associées au problème (\mathcal{P}) . On peut définir le (σ, ω) -centre analytique de l'ensemble des solutions du problème dual associé au problème (\mathcal{P}) . Soient $(\sigma, \omega) \in \mathbb{R} \times \mathbb{R}$ tel que $\sigma e_m > \lambda$ et $\omega e_p > D\xi$ pour au moins un couple $(\lambda, \xi) \in \text{opt}(D)$. Si $\text{opt}(D)$ est réduit à un point, le (σ, ω) -centre analytique de l'ensemble $\text{opt}(D)$ est ce point. Dans le cas contraire, on définit l'ensemble suivant

$$N = \{i : \exists(\lambda, \xi) \in \text{opt}(D) \text{ tel que } \lambda_i > 0\}.$$

Cet ensemble est non vide (si ce n'était pas le cas, $\text{opt}(D)$ serait réduit à $\{0\}$). On définit ensuite le (σ, ω) -centre analytique de l'ensemble $\text{opt}(D)$ comme l'unique solution du problème

$$\left\{ \begin{array}{l} \max \sum_{i \in N} \log(\lambda_i) + \sum_{i=1}^m \log(\sigma - \lambda_i) + \sum_{i=1}^p \log(\omega - d_i \xi_i), \\ (\lambda_N, \sigma e_m - \lambda, \omega e_p - D\xi) > 0, \\ (\lambda, \xi) \in \text{opt}(D). \end{array} \right. \quad (3.81)$$

Lemme 3.13.5 *Supposons les hypothèses H_1 et H_2 satisfaites. Alors, pour tout $(\sigma, \omega) \in \mathbb{R} \times \mathbb{R}$ tel que $\sigma e_m > \lambda$ et $\omega e_p > D\xi$ pour au moins un couple $(\lambda, \xi) \in \text{opt}(D)$, le problème (3.81) a une solution unique.*

Preuve. La démonstration est similaire à celle du lemme (3.13.4) en remarquant que la fonction objectif du problème (3.81) est strictement concave. \square

D'après la condition de complémentarité, l'intersection des ensembles B et N est vide, mais il se peut qu'il y ait des indices qui ne soient ni dans B , ni dans N . On dit que le problème (\mathcal{P}) a la propriété de stricte complémentarité si $B \cup N = \{1, \dots, m\}$.

Théorème 3.13.6 *Supposons les hypothèses H_1 et H_2 satisfaites. Supposons aussi que les fonctions f et c soient $C^{1,1}$. Supposons encore que le problème (\mathcal{P}) ait la propriété de stricte complémentarité et que les tolérances des critères d'arrêt $\epsilon^j := (\epsilon_\ell^j, \epsilon_c^j, \epsilon_s^j, \epsilon_y^j)$ utilisées dans l'algorithme \mathcal{A} satisfassent l'estimation $\epsilon^j = o(\mu^j)$. Alors, l'algorithme \mathcal{A} génère une suite de paramètres de pénalité $\{(\sigma^j, \omega^j)\}$ et une suite d'itérés $\{z^j\}$ tels que, pour j assez grand, $\sigma^j = \sigma$, $\omega^j = \omega$ et $\{z^j\}$ converge vers un point $z^* := (x^*, s^*, y^*, \lambda^*, \xi^*)$ où x^* est le centre analytique de l'ensemble de solutions primales, $s^* = 0$, $y^* = 0$ et (λ^*, ξ^*) est le (σ, ω) -centre analytique de l'ensemble de solutions duales.*

Preuve. La condition de Slater étant vérifiée, le problème (\mathcal{P}) est strictement réalisable. Donc, d'après le (iii) du théorème 3.12.3, la suite $\{(x^j, \lambda^j, \xi^j)\}$ est bornée, $\{s^j\}$ tend vers zéro, $\{y^j\}$ tend vers zéro et tout point d'accumulation de $\{(x^j, \lambda^j, \xi^j)\}$ est une solution primale-duale du problème (\mathcal{P}) . La suite des multiplicateurs $\{(\lambda^j, \xi^j)\}$ étant bornée, cela entraîne d'après la mise à jour des paramètres de pénalité que ces derniers sont bornés. Ainsi, pour j assez grand, $\sigma^j = \sigma$, $\omega^j = \omega$. On a aussi, pour j assez grand, $\sigma e_m - \lambda^j \geq \underline{\sigma} e_m > 0$ et $\omega e_p - D\xi^j \geq \underline{\omega} e_p > 0$. Cela montre que tout point limite (λ, ξ) de la suite

3.13. Convergence des itérés vers le centre analytique primal-dual

$\{(\lambda^j, \xi^j)\}$ satisfait les conditions $\sigma e_m - \lambda \geq \underline{\sigma} e_m > 0$ et $\omega e_p - D\xi \geq \underline{\omega} e_p > 0$. Le problème (3.81) est donc réalisable et le (σ, ω) -centre analytique est bien défini.

Soit $(\bar{x}, \bar{\lambda}, \bar{\xi})$ une solution primale-duale du problème (\mathcal{P}) telle que $\sigma e_m - \bar{\lambda} > 0$ et $\omega e_p - D\bar{\xi} > 0$. Alors le lagrangien $\ell(\cdot, \bar{\lambda}, \bar{\xi})$ est minimisé en \bar{x} . De plus, $\bar{\lambda}^\top c(\bar{x}) = 0$ et $A\bar{x} = b$. On a donc

$$f(\bar{x}) = \ell(\bar{x}, \bar{\lambda}, \bar{\xi}) \leq \ell(x^j, \bar{\lambda}, \bar{\xi}) = f(x^j) + \bar{\lambda}^\top c(x^j) + \bar{\xi}^\top (b - Ax^j). \quad (3.82)$$

Reprenons le résultat du lemme 3.12.1 :

$$\begin{aligned} f(x^j) + \sigma^j e_m^\top s^j + \omega^j e_p^\top y^j &\leq \ell(x, \lambda^j, \xi^j) + m^{\frac{1}{2}} (\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}} \epsilon_y^j \\ &\quad + (2m + p)\mu^j + \|x^j - x\| \epsilon_l^j \end{aligned}$$

On reprend cette formule en $x = \bar{x}$ et rappelant que pour j assez grand, $\sigma^j = \sigma$ et $\omega^j = \omega$, on obtient

$$\begin{aligned} f(x^j) + \sigma e_m^\top s^j + \omega e_p^\top y^j &\leq f(\bar{x}) + (\lambda^j)^\top c(\bar{x}) + m^{\frac{1}{2}} (\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}} \epsilon_y^j \\ &\quad + (2m + p)\mu^j + \|x^j - \bar{x}\| \epsilon_l^j. \end{aligned}$$

Avec l'inégalité (3.82), on obtient

$$\begin{aligned} f(x^j) + \sigma e_m^\top s^j + \omega e_p^\top y^j &\leq f(x^j) + c(x^j)^\top \bar{\lambda} + (\lambda^j)^\top c(\bar{x}) + \bar{\xi}^\top (b - Ax^j) \\ &\quad + m^{\frac{1}{2}} (\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}} \epsilon_y^j \\ &\quad + (2m + p)\mu^j + \|x^j - \bar{x}\| \epsilon_l^j. \end{aligned}$$

Alors,

$$\begin{aligned} 0 &\leq c(x^j)^\top \bar{\lambda} + (\lambda^j)^\top c(\bar{x}) + \bar{\xi}^\top (b - Ax^j) - \sigma e_m^\top s^j - \omega e_p^\top y^j \\ &\quad + m^{\frac{1}{2}} (\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}} \epsilon_y^j + (2m + p)\mu^j + \|x^j - \bar{x}\| \epsilon_l^j \\ &\leq (c(x^j) - s^j)^\top \bar{\lambda} + (\bar{\lambda} - \sigma e_m)^\top s^j - (y^j)^\top (\omega e_p - D\xi^j) + (\lambda^j)^\top c(\bar{x}) \\ &\quad + m^{\frac{1}{2}} (\epsilon_c^j + \epsilon_s^j) + p^{\frac{1}{2}} \epsilon_y^j + (2m + p)\mu^j + \|x^j - \bar{x}\| \epsilon_l^j. \end{aligned}$$

Comme la suite $\{x^j\}$ est bornée et que $\epsilon^j = o(\mu^j)$, on en déduit que

$$(s^j - c(x^j))^\top \bar{\lambda} + (\sigma e_m - \bar{\lambda})^\top s^j + (y^j)^\top (\omega e_p - D\bar{\xi}) - (\lambda^j)^\top c(\bar{x}) \leq (2m + p)\mu^j + o(\mu^j).$$

Par définition des ensembles B et N, on peut éliminer certains termes de l'inégalité précédente

$$\begin{aligned} (s_N^j - c_N(x^j))^\top \bar{\lambda}_N + (\sigma e_m - \bar{\lambda})^\top s^j &\quad + (y^j)^\top (\omega e_p - D\bar{\xi}) - c_B(\bar{x})^\top \lambda_B^j \\ &\leq (2m + p)\mu^j + o(\mu^j). \end{aligned} \quad (3.83)$$

La suite de la démonstration s'inspire d'une idée de MacLinden ([37]). On définit les quantités suivantes

$$\begin{aligned} \Upsilon^j &= \Lambda^j (s^j - c(x^j)) - \mu^j e_m \\ \Gamma^j &= S^j (\sigma^j e_m - \lambda^j) - \mu^j e_m \\ \Delta^j &= Y^j (\omega^j e_p - D\xi^j) - \mu^j e_p. \end{aligned}$$

On a alors pour $i \in \{1, \dots, m\}$,

$$\lambda_i^j = \frac{\Upsilon_i^j + \mu^j}{s_i^j - c_i(x^j)}, s_i^j - c(x^j) = \frac{\Upsilon_i^j + \mu^j}{\lambda_i^j}, s_i^j = \frac{\Gamma_i^j + \mu^j}{\sigma^j - \lambda_i^j}, y_i^j = \frac{\Delta_i^j + \mu^j}{\omega^j - d_i \xi_i^j}.$$

En injectant ces valeurs dans l'inégalité (3.83), on a

$$\begin{aligned} & \sum_{i \in N} \frac{\bar{\lambda}_i}{\lambda_i^j} (\Upsilon_i^j + \mu^j) + \sum_{i=1}^m \left(\frac{\sigma - \bar{\lambda}_i}{\sigma^j - \lambda_i^j} \right) (\Gamma_i^j + \mu^j) \\ & + \sum_{i=1}^p \left(\frac{\omega - d_i \bar{\xi}_i}{\omega^j - d_i \xi_i^j} \right) (\Delta_i^j + \mu^j) + \sum_{i \in B} \frac{-c_i(\bar{x})}{s_i^j - c_i(x^j)} (\Upsilon_i^j + \mu^j) \\ & \leq (2m + p) \mu^j + o(\mu^j), \end{aligned}$$

et finalement,

$$\begin{aligned} & \sum_{i \in N} \frac{\bar{\lambda}_i}{\lambda_i^j} \frac{\Upsilon_i^j + \mu^j}{\mu^j} + \sum_{i=1}^m \left(\frac{\sigma - \bar{\lambda}_i}{\sigma^j - \lambda_i^j} \right) \frac{\Gamma_i^j + \mu^j}{\mu^j} \\ & + \sum_{i=1}^p \left(\frac{\omega - d_i \bar{\xi}_i}{\omega^j - d_i \xi_i^j} \right) \frac{\Delta_i^j + \mu^j}{\mu^j} + \sum_{i \in B} \frac{-c_i(\bar{x})}{s_i^j - c_i(x^j)} \frac{\Upsilon_i^j + \mu^j}{\mu^j} \\ & \leq (2m + p) + o(1). \end{aligned}$$

On a par hypothèse que $\Upsilon_i^j = o(\mu^j)$, $\Gamma_i^j = o(\mu^j)$ et $\Delta_i^j = o(\mu^j)$. Soit (x^*, λ^*, ξ^*) un point limite de la suite $\{(x^j, \lambda^j, \xi^j)\}$. Par passage à la limite dans l'inégalité précédente, en rappelant que $s^j \rightarrow 0$ et que $s^j - c(x^j) \rightarrow -c(x^*)$, on a

$$\frac{1}{2m + p} \left(\sum_{i \in N} \frac{\bar{\lambda}_i}{\lambda_i^*} + \sum_{i=1}^m \frac{\sigma - \bar{\lambda}_i}{\sigma - \lambda_i^*} + \sum_{i=1}^p \frac{\omega - d_i \bar{\xi}_i}{\omega - d_i \xi_i^*} + \sum_{i \in B} \frac{c_i(\bar{x})}{c_i(x^*)} \right) \leq 1.$$

On a nécessairement $\lambda_N^* > 0$, $\sigma e_m - \lambda^* > 0$, $\omega e_p - D\xi^* > 0$ et $c_B(x^*) > 0$. Par stricte complémentarité, on a exactement $(2m + p)$ fractions positives dans le membre gauche de l'inégalité précédente. Par monotonie et concavité de la fonction \log , on obtient

$$\sum_{i \in N} \log\left(\frac{\bar{\lambda}_i}{\lambda_i^*}\right) + \sum_{i=1}^m \log\left(\frac{\sigma - \bar{\lambda}_i}{\sigma - \lambda_i^*}\right) + \sum_{i=1}^p \log\left(\frac{\omega - d_i \bar{\xi}_i}{\omega - d_i \xi_i^*}\right) + \sum_{i \in B} \log\left(\frac{c_i(\bar{x})}{c_i(x^*)}\right) \leq 0.$$

En prenant $\lambda^* = \bar{\lambda}$ et $\xi^* = \bar{\xi}$ on a

$$\sum_{i \in B} \log\left(\frac{-c_i(\bar{x})}{-c_i(x^*)}\right) \leq 0,$$

et donc

$$\sum_{i \in B} \log(-c_i(\bar{x})) \leq \sum_{i \in B} \log(-c_i(x^*)),$$

puis, avec $x^* = \bar{x}$, on a

$$\sum_{i \in N} \log\left(\frac{\bar{\lambda}_i}{\lambda_i^*}\right) + \sum_{i=1}^m \log\left(\frac{\sigma - \bar{\lambda}_i}{\sigma - \lambda_i^*}\right) + \sum_{i=1}^p \log\left(\frac{\omega - d_i \bar{\xi}_i}{\omega - d_i \xi_i^*}\right) \leq 0,$$

et donc

$$\sum_{i \in N} \log(\bar{\lambda}_i) + \sum_{i=1}^m \log(\sigma - \bar{\lambda}_i) + \sum_{i=1}^p \log(\omega - d_i \bar{\xi}_i) \leq \sum_{i \in N} \log(\lambda_i^*) + \sum_{i=1}^m \log(\sigma - \lambda_i^*) + \sum_{i=1}^p \log(\omega - d_i \xi_i^*).$$

On en déduit donc que x^* est une solution du problème (3.79) et que (λ^*, ξ^*) en est une du problème (3.81). Par unicité de ces solutions, on a le résultat cherché. \square

3.14 Conclusion

On a étudié dans ce chapitre une méthode de points intérieurs et de quasi-Newton pour résoudre des problèmes avec contraintes d'inégalité et contraintes d'égalité linéaire à partir d'un point initial non réalisable. On peut étendre cette méthode à la résolution de problèmes de grande taille en utilisant les formules de BFGS à mémoire limitée présentée au chapitre précédent. La convergence r-linéaire des itérations internes et la convergence des itérés externes sont alors conservées.

Conclusion

Dans le chapitre 1 de ce travail, on a étudié la mise en œuvre d'une méthode de points intérieurs utilisant les formules de quasi-Newton présentée dans [4]. Le code NOPTIQ ainsi obtenu est robuste et numériquement comparable au code LANCELOT qui est un code de référence. Cela nous a encouragé pour étendre cette méthode à la résolution de problèmes de grande taille.

Dans le chapitre 2, on a montré que les formules de BFGS à mémoire limitée sont efficaces pour résoudre des problèmes de grande taille avec des contraintes d'inégalité non linéaire. On a aussi montré que les méthodes de points intérieurs utilisant les formules de BFGS à mémoire limitée se prêtent bien à la résolution de tels problèmes. Dans le cas d'un problème convexe, les itérés internes générés par cette méthode convergent r -linéairement. L'extension du code NOPTIQ utilisant les formules de BFGS à mémoire limitée pour résoudre des problèmes de grande taille a été mise en œuvre. Les résultats numériques obtenus montrent que dans le cas où ne sont que des contraintes de bornes, NOPTIQ est comparable au code 1-BFGS-B qui est un code de référence. Dans le cas où des contraintes générales sont présentes dans le problème, on a vu que le code NOPTIQ est comparable au code LANCELOT.

Dans le chapitre 3, on a présenté un algorithme employant une méthode de points intérieurs et de quasi-Newton pour résoudre un problème de minimisation non linéaire avec contraintes d'inégalité et d'égalité linéaire à partir d'un point initial non réalisable. Cette méthode étend la méthode proposée dans [2] pour prendre en compte des contraintes d'égalité linéaire. Un résultat de convergence q -superlinéaire des itérés vers la solution du problème barrière a été montré, ainsi que la convergence globale des itérés vers la solution du problème initial.

La mise en œuvre du code NOPTIQ présentée aux chapitres 1 et 2 de ce mémoire fait l'objet d'un dépôt de logiciel à l'Agence pour la Protection des Programmes ([5]). Le code NOPTIQ est disponible sur simple demande à ses auteurs.

Une piste de recherche possible est la mise en œuvre de l'algorithme décrit dans le chapitre 3. L'utilisateur ne devrait alors plus fournir à NOPTIQ un point initial strictement réalisable. Cela faciliterait l'utilisation du code. Une autre piste possible est l'étude de la décroissance du paramètre de perturbation μ . En optimisation linéaire, la décroissance de ce paramètre est obtenue par des formules performantes. L'extension de ces formules en optimisation non linéaire est plus difficile. Les méthodes de prédiction-corrrection sont une voie possible ([59, 54, 43]). Enfin, une autre piste de recherche serait de trouver

Conclusion

une méthode d'initialisation des multiplicateurs. Une possibilité serait de résoudre le problème de moindres carrés suivant en λ , où x^0 est un point initial fourni par l'utilisateur :

$$\begin{cases} \min \|\nabla f(x^0) + \nabla c(x^0)\lambda\|^2 \\ \lambda \geq 0, \end{cases}$$

et d'en déduire le paramètre de perturbation initial. Ainsi les conditions d'optimalité du premier problème barrière seraient satisfaites et les itérés se trouveraient alors proche du chemin central.

Annexes

Résultats numériques de NOPTIQ (chapitre 1)

On présente dans cette partie les résultats numériques de NOPTIQ utilisant les formules de quasi-Newton à mémoire pleine. Les problèmes testés sont ceux des tableaux (1.1), (1.2) et (1.3). Les tableaux suivants énumèrent ces résultats où

- nom représente le nom du problème testé,
- n représente le nombre total de variables,
- m représente le nombre total de contraintes,
- nf représente le nombre total d'appels au simulateur pour évaluer f et c,
- f représente la valeur finale de f au dernier point calculé par NOPTIQ,
- test1 représente la valeur de la norme euclidienne du gradient du lagrangien au dernier point calculé par NOPTIQ,
- test2 représente la valeur de la norme euclidienne des conditions de complémentarité au dernier point calculé par NOPTIQ,
- info représente le mode d'arrêt de NOPTIQ
 - 0 NOPTIQ stoppe normalement,
 - 2 NOPTIQ stoppe car la matrice M devient singulière,
 - 5 NOPTIQ stoppe car le nombre maximal d'appels au simulateur est atteint,
 - 6 NOPTIQ stoppe car le pas initial dans la recherche linéaire est trop petit,
 - 7 NOPTIQ stoppe car la direction dans l'espace primal est trop petite. Si les conditions d'optimalité sont pratiquement satisfaites, cette sortie n'est pas un échec pour le code.

nom	n	m	nf	f	test1	test2	info
3PK	30	30	328	0.1720E + 01	0.1060E - 06	0.1000E - 09	7
ALLINIT	4	3	1	0.3789E + 40	0.8437E + 40	0.1000E + 01	6
BQPIVAR	1	2	13	0.1000E - 07	0.1266E - 15	0.1000E - 07	7
BQPGABIM	50	92	41	-0.3790E - 04	0.1033E - 04	0.1000E - 09	7
BQPGASIM	50	100	47	-0.5520E - 04	0.6533E - 05	0.1000E - 09	7
CAMEL6	2	4	27	-0.1032E + 01	0.2317E - 11	0.1000E - 06	7
CHEBYQAD	100	200	2054	0.4878E - 02	0.1602E - 05	0.1000E - 06	7
DECONVB	61	72	62	0.8936E + 02	0.5308E + 03	0.8398E - 04	2
DPIGRI	7	4	24	0.6806E + 03	0.2622E - 07	0.1000E - 07	7
EG1	3	4	54	-0.1429E + 01	0.2597E - 07	0.1000E - 05	7
HART6	6	12	38	-0.3323E + 01	0.8504E - 07	0.1000E - 06	7
HATFLDA	4	4	41	0.1152E - 13	0.5066E - 08	0.1000E - 07	0
HATFLDB	4	5	35	0.5573E - 02	0.1852E - 08	0.1000E - 08	0
HATFLDC	25	48	44	0.4711E - 15	0.3533E - 07	0.1000E - 06	7
HIMMELPI	2	4	19	-0.6205E + 02	0.6260E - 14	0.1000E - 08	0
HS100MOD	7	4	28	0.6787E + 03	0.8059E - 05	0.1000E - 05	7
HS100	7	4	24	0.6806E + 03	0.2622E - 07	0.1000E - 07	7
HS110	200	400	1	-0.1478E + 39	0.4645E + 38	0.1000E + 01	6
HS113	10	8	21	0.2431E + 02	0.6133E - 09	0.1000E - 08	0
HS117	15	20	31	0.3451E + 03	0.3715E + 04	0.1000E + 01	2
HS12	2	1	19	-0.3000E + 02	0.2627E - 08	0.1000E - 06	7
HS1	2	1	69	0.9551E - 17	0.4579E - 08	0.1000E - 07	0
HS24	2	5	18	-0.1000E + 01	0.1815E - 06	0.1000E - 04	7
HS25	3	6	135	0.1358E - 11	0.2244E - 11	0.1000E - 05	7
HS29	3	1	36	-0.2263E + 02	0.9380E - 07	0.1000E - 04	7
HS2	2	1	29	0.4941E + 01	0.1344E - 07	0.1000E - 07	7
HS351	3	7	21	0.1111E + 00	0.4660E - 12	0.1000E - 08	0
HS35MOD	3	3	24	0.2500E + 00	0.6316E - 10	0.1789E - 08	0
HS35	3	4	20	0.1111E + 00	0.3480E - 08	0.1000E - 05	7
HS36	3	7	32	-0.3300E + 04	0.1037E - 11	0.1000E - 06	7
HS37	3	8	24	-0.3456E + 04	0.1887E - 07	0.1000E - 07	7
HS38	4	8	80	0.1520E - 15	0.2210E - 06	0.1000E - 08	7
HS3MOD	2	1	20	0.1000E - 07	0.6573E - 10	0.1000E - 07	0
HS3	2	1	23	0.1000E - 04	0.1802E - 21	0.1000E - 03	7
HS43	4	3	39	-0.4400E + 02	0.9304E - 10	0.1000E - 07	7
HS44NEW	4	10	23	-0.1500E + 02	0.1254E - 06	0.1000E - 03	7
HS45	5	10	28	0.1664E + 01	0.6865E - 02	0.1000E + 01	2
HS4	2	2	14	0.2667E + 01	0.3496E - 13	0.1000E - 07	7
HS57	2	3	71	0.3065E - 01	0.1980E - 01	0.1565E - 01	2
HS5	2	4	21	-0.1913E + 01	0.2744E - 09	0.1000E - 07	7
HS67	3	20	31	-0.1162E + 04	0.3278E - 14	0.1000E - 08	0
HS70	4	9	75	0.7498E - 02	0.1838E - 09	0.1000E - 08	0
HS761	4	11	21	-0.4682E + 01	0.1344E - 07	0.1000E - 06	7
HS76	4	7	20	-0.4682E + 01	0.2253E - 08	0.1000E - 06	7
HS84	5	16	35	-0.5280E + 07	0.9971E - 08	0.1000E - 04	7
HS85	5	48	31	-0.2216E + 01	0.5922E - 09	0.1000E - 06	7
HS93	6	8	197	0.1351E + 03	0.4657E - 07	0.1000E - 08	7
KSIP	20	1001	53	0.5758E + 00	0.1911E - 08	0.1000E - 09	0
LOGROS	2	2	194	0.0000E + 00	0.4369E - 07	0.1000E - 11	7
MAXLIKA	8	16	79	0.1136E + 04	0.3631E - 04	0.1000E - 05	7
MDHOLE	2	1	164	0.1000E - 07	0.1604E - 08	0.1000E - 07	0
MIFFLIN1	3	2	18	-0.1000E + 01	0.3475E - 08	0.1000E - 06	7
MINMAXBD	5	20	7	0.6530E + 03	0.7408E + 02	0.1000E + 01	6
OSLBQP	8	11	23	0.6250E + 01	0.1960E - 15	0.1788E - 08	0

TAB. 3.1 – Résultats du code NOPTIQ avec les formules de BFGS approchant le hessien du lagrangien et utilisant la correction de Powell.

nom	n	m	nf	f	test1	test2	info
PALMER1A	6	2	230	0.8988E-01	0.5839E-08	0.1000E-09	0
PALMER1B	4	2	64	0.3447E+01	0.9845E-04	0.1000E-06	7
PALMER1E	8	1	250	0.8352E-03	0.1085E-06	0.1000E-11	7
PALMER1	4	3	53	0.1175E+05	0.2347E-05	0.1000E-06	7
PALMER2A	6	2	182	0.1711E-01	0.4459E-08	0.1000E-10	0
PALMER2B	4	2	33	0.1336E+04	0.2409E+03	0.9809E+00	2
PALMER2E	8	1	271	0.2065E-03	0.3382E-08	0.1000E-11	0
PALMER2	4	3	36	0.4838E+04	0.1432E+04	0.1000E+01	2
PALMER3A	6	2	448	0.2043E-01	0.9062E-08	0.1000E-09	0
PALMER3B	4	2	72	0.4228E+01	0.1169E-05	0.1000E-07	7
PALMER3E	8	1	261	0.5074E-04	0.1455E-07	0.1000E-10	7
PALMER3	4	3	51	0.2417E+04	0.3820E-03	0.1000E-04	7
PALMER4A	6	2	52	0.1592E+03	0.4388E+02	0.1080E+00	2
PALMER4B	4	2	103	0.6835E+01	0.1713E-06	0.1000E-06	7
PALMER4E	8	1	171	0.1480E-03	0.1868E-08	0.1000E-10	0
PALMER4	4	3	64	0.2424E+04	0.8217E-04	0.1153E-05	2
PALMER5A	8	2	26	0.1264E+05	0.1297E+04	0.1000E+01	2
PALMER5B	9	2	3368	0.9752E-02	0.6622E-06	0.1000E-09	7
PALMER5D	8	4	43	0.8734E+02	0.2577E-06	0.1000E-07	7
PALMER5E	8	1	10000	0.2094E-01	0.3600E-02	0.1000E-03	5
PALMER6A	6	2	50	0.1198E+03	0.3682E+02	0.1000E+01	2
PALMER6E	8	1	223	0.2240E-03	0.9665E-08	0.1000E-10	0
PALMER7A	6	2	10000	0.1034E+02	0.2666E-01	0.1000E-03	5
PALMER7E	8	1	3055	0.1015E+02	0.2989E-04	0.1000E-07	7
PALMER8A	6	2	129	0.7401E-01	0.2223E-07	0.1000E-09	7
PALMER8E	8	1	117	0.6339E-02	0.5475E-07	0.1000E-09	7
PROBPENL	500	1000	5	0.3992E-06	0.2823E-06	0.1000E+00	7
PSPDOC	4	1	27	0.2414E+01	0.2870E-06	0.1000E-04	7
S368	8	16	66	-0.4375E+00	0.5199E-07	0.1000E-06	7
SIM2BQP	2	2	13	0.1000E-07	0.9483E-16	0.1000E-07	7
SIMBQP	2	2	14	0.1000E-07	0.1424E-07	0.1000E-07	7
SIPOW1M	2	2000	22	-0.1000E+01	0.2132E-13	0.1018E-08	0
SIPOW1	2	2000	23	-0.1000E+01	0.1199E-13	0.1014E-08	0
SIPOW2M	2	2000	52	-0.1000E+01	0.1288E-13	0.1021E-08	0
SIPOW2	2	2000	51	-0.1000E+01	0.4441E-14	0.1128E-08	0
SIPOW3	4	2000	30	0.5347E+00	0.1278E-13	0.1009E-08	0
SIPOW4	4	2000	29	0.2724E+00	0.5381E-14	0.1113E-08	0
SPECAN	9	18	50	0.1671E-12	0.6765E-05	0.1000E-08	7
SPIRAL	3	2	112	0.9459E-01	0.1146E-01	0.9998E-01	6
WEEDS	3	4	152	0.2587E+01	0.1728E-08	0.1000E-08	0
WOMFLET	3	3	213	0.4153E+01	0.1067E+01	0.1002E+01	2
YFIT	3	1	160	0.6670E-12	0.9936E-08	0.1000E-08	0
ZY2	3	6	19	0.2000E+01	0.1705E-05	0.1000E-04	7

TAB. 3.2 – Résultats du code NOPTIQ avec les formules de BFGS approchant le hessien du lagrangien et utilisant la correction de Powell.

nom	n	m	nf	f	test1	test2	info
3PK	30	30	519	0.1720E + 01	0.2223E - 08	0.1000E - 10	0
ALLINIT	4	3	1	0.3789E + 40	0.8437E + 40	0.1000E + 01	6
BQPIVAR	1	2	13	0.1000E - 07	0.9540E - 16	0.1000E - 07	7
BQPGABIM	50	92	48	-0.3790E - 04	0.8973E - 05	0.1000E - 09	7
BQPGASIM	50	100	51	-0.5520E - 04	0.1453E - 04	0.1000E - 09	7
CAMEL6	2	4	18	-0.1032E + 01	0.1280E - 10	0.1000E - 06	7
CHEBYQAD	100	200	456	0.5658E + 00	0.1600E + 02	0.9999E + 00	7
DECONVB	61	72	227	0.3524E + 02	0.4382E + 04	0.1874E - 02	7
DPIGRI	7	4	27	0.6806E + 03	0.4696E - 07	0.1000E - 07	7
EG1	3	4	75	-0.1429E + 01	0.7322E - 07	0.1000E - 05	7
HART6	6	12	27	-0.3323E + 01	0.9445E - 09	0.1000E - 07	7
HATFLDA	4	4	51	0.1160E - 13	0.1496E - 06	0.1000E - 06	7
HATFLDB	4	5	38	0.5573E - 02	0.1648E - 07	0.1000E - 07	7
HATFLDC	25	48	40	0.4816E - 13	0.3577E - 06	0.1000E - 05	7
HIMMELPI	2	4	21	-0.6205E + 02	0.2073E - 14	0.1000E - 08	0
HS100MOD	7	4	29	0.6787E + 03	0.2988E - 05	0.1000E - 05	7
HS100	7	4	27	0.6806E + 03	0.4696E - 07	0.1000E - 07	7
HS110	200	400	1	-0.1478E + 39	0.4645E + 38	0.1000E + 01	6
HS113	10	8	24	0.2431E + 02	0.1028E - 07	0.1000E - 08	7
HS117	15	20	39	0.1119E + 04	0.3715E + 04	0.1000E + 01	4
HS12	2	1	19	-0.3000E + 02	0.1284E - 10	0.1000E - 06	7
HS1	2	1	83	0.1604E - 16	0.1807E - 10	0.1000E - 07	7
HS24	2	5	47	-0.1000E + 01	0.1856E - 05	0.1000E - 04	7
HS25	3	6	9	0.3283E + 02	0.8659E - 03	0.1000E + 01	7
HS29	3	1	30	-0.2263E + 02	0.2388E - 06	0.1000E - 03	7
HS2	2	1	40	0.4941E + 01	0.1047E - 05	0.1000E - 07	7
HS351	3	7	21	0.1111E + 00	0.4747E - 07	0.1000E - 05	7
HS35MOD	3	3	24	0.2500E + 00	0.2364E - 11	0.1789E - 08	0
HS35	3	4	23	0.1111E + 00	0.1288E - 07	0.1000E - 06	7
HS36	3	7	38	-0.3300E + 04	0.2746E - 11	0.1000E - 06	7
HS37	3	8	27	-0.3456E + 04	0.3013E - 07	0.1000E - 07	7
HS38	4	8	102	0.1522E - 16	0.1447E - 06	0.1000E - 07	7
HS3MOD	2	1	17	0.9997E - 08	0.2801E - 10	0.9997E - 08	0
HS3	2	1	6	0.1000E + 00	0.9259E - 08	0.1000E + 01	4
HS43	4	3	39	-0.4400E + 02	0.1367E - 10	0.1000E - 07	7
HS44NEW	4	10	92	-0.6502E + 01	0.1859E + 01	0.1000E + 01	7
HS45	5	10	333	0.1669E + 01	0.6992E - 02	0.1000E + 01	4
HS4	2	2	16	0.2667E + 01	0.1826E - 12	0.1000E - 07	7
HS57	2	3	568	0.0000E + 00	0.0000E + 00	0.0000E + 00	0
HS5	2	4	19	-0.1913E + 01	0.4261E - 13	0.1000E - 07	7
HS67	3	20	38	-0.1162E + 04	0.3712E - 12	0.1001E - 08	0
HS70	4	9	82	0.7498E - 02	0.1788E - 08	0.1000E - 08	0
HS761	4	11	20	-0.4682E + 01	0.9033E - 07	0.1000E - 05	7
HS76	4	7	22	-0.4682E + 01	0.1178E - 08	0.1000E - 06	7
HS84	5	16	41	-0.5280E + 07	0.2608E - 04	0.1000E - 04	7
HS85	5	48	89	-0.1967E + 01	0.4268E - 02	0.1000E + 00	4
HS93	6	8	2310	0.1351E + 03	0.9039E - 07	0.1000E - 08	7
KSIP	20	1001	61	0.5758E + 00	0.1069E - 08	0.1000E - 09	0
LOGROS	2	2	173	0.0000E + 00	0.2336E - 06	0.1000E - 11	7
MAXLIKA	8	16	106	0.1136E + 04	0.2731E - 04	0.1000E - 05	7
MDHOLE	2	1	150	0.1000E - 07	0.6364E - 07	0.1000E - 07	7
MIFFLIN1	3	2	17	-0.1000E + 01	0.5847E - 12	0.9908E - 09	0
MINMAXBD	5	20	394	0.1157E + 03	0.5351E - 09	0.1000E - 08	0
OSLBQP	8	11	23	0.6250E + 01	0.1960E - 15	0.1788E - 08	0

TAB. 3.3 – Résultats du code NOPTIQ avec les formules de BFGS approchant l'inverse du hessien du lagrangien et utilisant la correction de Powell.

nom	n	m	nf	f	test1	test2	info
PALMER1A	6	2	478	0.8988E-01	0.1497E-06	0.1000E-09	7
PALMER1B	4	2	74	0.3447E+01	0.1166E-05	0.1000E-07	7
PALMER1E	8	1	314	0.1136E+00	0.2681E-02	0.1000E-07	7
PALMER1	4	3	75	0.1175E+05	0.8112E-03	0.1000E-04	7
PALMER2A	6	2	264	0.1711E-01	0.1867E-08	0.1000E-10	0
PALMER2B	4	2	87	0.6233E+00	0.1058E-06	0.1000E-08	7
PALMER2E	8	1	607	0.2065E-03	0.9860E-07	0.1000E-10	7
PALMER2	4	3	50	0.3651E+04	0.1029E-05	0.1000E-05	7
PALMER3A	6	2	202	0.2872E-01	0.2435E-02	0.1000E-04	7
PALMER3B	4	2	81	0.4228E+01	0.2631E-06	0.1000E-07	7
PALMER3E	8	1	540	0.5074E-04	0.1005E-06	0.1000E-10	7
PALMER3	4	3	73	0.2417E+04	0.2776E-03	0.1000E-04	7
PALMER4A	6	2	273	0.4061E-01	0.1016E-07	0.1000E-09	7
PALMER4B	4	2	85	0.6835E+01	0.5050E-05	0.1000E-06	7
PALMER4E	8	1	442	0.1480E-03	0.4396E-06	0.1000E-09	7
PALMER4	4	3	215	0.2285E+04	0.3177E-05	0.1000E-05	7
PALMER5A	8	2	29	0.1238E+05	0.1297E+04	0.1000E+01	7
PALMER5B	9	2	477	0.3605E-01	0.1124E-02	0.1000E-06	4
PALMER5D	8	4	42	0.8734E+02	0.1524E-06	0.1000E-06	7
PALMER5E	8	1	1245	0.4119E-01	0.4368E-01	0.1000E-02	4
PALMER6A	6	2	552	0.5595E-01	0.1703E-06	0.1000E-08	7
PALMER6E	8	1	485	0.2240E-03	0.3683E-08	0.1000E-10	0
PALMER7A	6	2	1531	0.1068E+02	0.4126E+00	0.1000E-02	7
PALMER7E	8	1	1123	0.1015E+02	0.3105E+00	0.1000E-03	7
PALMER8A	6	2	239	0.1082E+00	0.3949E-01	0.1000E-03	7
PALMER8E	8	1	247	0.6339E-02	0.4207E-08	0.1000E-09	0
PROBPENL	500	1000	85	0.3992E-06	0.2823E-06	0.1001E-01	7
PSPDOC	4	1	28	0.2414E+01	0.4602E-07	0.1000E-05	7
S368	8	16	30	-0.1983E+00	0.4599E+00	0.9935E+00	7
SIM2BQP	2	2	13	0.1000E-07	0.9483E-16	0.1000E-07	7
SIMBQP	2	2	14	0.1000E-07	0.1356E-11	0.1000E-07	7
SIPOW1M	2	2000	3	0.0000E+00	0.0000E+00	0.0000E+00	0
SIPOW1	2	2000	3	0.0000E+00	0.0000E+00	0.0000E+00	0
SIPOW2M	2	2000	3	0.0000E+00	0.0000E+00	0.0000E+00	0
SIPOW2	2	2000	3	0.0000E+00	0.0000E+00	0.0000E+00	0
SIPOW3	4	2000	3	0.0000E+00	0.0000E+00	0.0000E+00	0
SIPOW4	4	2000	3	0.0000E+00	0.0000E+00	0.0000E+00	0
SPECAN	9	18	62	0.1684E-12	0.3731E-04	0.1000E-07	7
SPIRAL	3	2	388	0.1987E-08	0.2417E-10	0.9937E-09	0
WEEDS	3	4	86	0.2587E+01	0.1470E-06	0.1000E-07	7
WOMFLET	3	3	2410	0.3000E-04	0.1212E-03	0.1000E-03	7
YFIT	3	1	207	0.6670E-12	0.5337E-08	0.1000E-09	0
ZY2	3	6	126	0.2260E+01	0.3928E+00	0.9956E+00	4

TAB. 3.4 – Résultats du code NOPTIQ avec les formules de BFGS approchant l'inverse du hessien du lagrangien et utilisant la correction de Powell.

nom	n	m	nf	f	test1	test2	info
3PK	30	30	246	0.1720E + 01	0.6915E - 06	0.1000E - 08	7
ALLINIT	4	3	1	0.3789E + 40	0.8437E + 40	0.1000E + 01	6
BQPIVAR	1	2	18	0.1000E - 05	0.7109E - 07	0.1000E - 04	7
BQPGABIM	50	92	65	-0.3790E - 04	0.1472E - 04	0.1000E - 09	7
BQPGASIM	50	100	72	-0.5520E - 04	0.1821E - 04	0.1000E - 09	7
CAMEL6	2	4	32	-0.1032E + 01	0.9942E - 07	0.1000E - 06	7
CHEBYQAD	100	200	1856	0.4878E - 02	0.6038E - 07	0.1000E - 07	7
DECONVB	61	72	1898	0.5348E - 06	0.3276E - 02	0.1927E - 09	2
DIPIGRI	7	4	30	0.6806E + 03	0.1424E - 06	0.1000E - 06	7
EG1	3	4	46	-0.1429E + 01	0.1348E - 05	0.1000E - 04	7
HART6	6	12	75	-0.3323E + 01	0.6860E - 06	0.1000E - 05	7
HATFLDA	4	4	49	0.1152E - 13	0.3722E - 08	0.1000E - 07	0
HATFLDB	4	5	28	0.5573E - 02	0.1373E - 08	0.1000E - 08	0
HATFLDC	25	48	77	0.6948E - 13	0.3267E - 06	0.1000E - 05	7
HIMMELPI	2	4	19	-0.6205E + 02	0.6260E - 14	0.1000E - 08	0
HST00MOD	7	4	33	0.6787E + 03	0.8437E - 06	0.1000E - 06	7
HS100	7	4	29	0.6806E + 03	0.7418E - 05	0.1000E - 05	7
HS110	200	400	1	-0.1478E + 39	0.4645E + 38	0.1000E + 01	6
HS113	10	8	25	0.2431E + 02	0.1140E - 07	0.1000E - 08	7
HS117	15	20	30	0.4632E + 03	0.3715E + 04	0.1000E + 01	2
HS12	2	1	27	-0.3000E + 02	0.4707E - 06	0.1000E - 05	7
HS1	2	1	78	0.2246E - 16	0.1605E - 07	0.1000E - 07	7
HS24	2	5	18	-0.1000E + 01	0.3066E - 06	0.1000E - 04	7
HS25	3	6	143	0.1358E - 11	0.1380E - 11	0.1000E - 05	7
HS29	3	1	40	-0.2263E + 02	0.5026E - 06	0.1000E - 04	7
HS2	2	1	26	0.4941E + 01	0.8555E - 07	0.1000E - 07	7
HS351	3	7	27	0.1111E + 00	0.2649E - 07	0.1000E - 06	7
HS35MOD	3	3	60	0.2500E + 00	0.1124E - 07	0.1000E - 06	7
HS35	3	4	28	0.1111E + 00	0.3173E - 07	0.1000E - 06	7
HS36	3	7	22	-0.3300E + 04	0.4152E - 12	0.1000E - 06	7
HS37	3	8	30	-0.3456E + 04	0.6159E - 07	0.1000E - 07	7
HS38	4	8	93	0.3757E - 15	0.5634E - 06	0.1000E - 08	7
HS3MOD	2	1	19	0.1000E - 07	0.1139E - 08	0.1000E - 07	0
HS3	2	1	23	0.1000E - 04	0.6698E - 21	0.1000E - 03	7
HS43	4	3	35	-0.4400E + 02	0.4621E - 07	0.1000E - 06	7
HS44NEW	4	10	23	-0.1500E + 02	0.9843E - 07	0.1000E - 03	7
HS45	5	10	23	0.1665E + 01	0.6865E - 02	0.1000E + 01	2
HS4	2	2	15	0.2667E + 01	0.7804E - 13	0.1000E - 07	7
HS57	2	3	160	0.3065E - 01	0.2835E - 08	0.1000E - 08	0
HS5	2	4	51	-0.1913E + 01	0.1196E - 07	0.1000E - 06	7
HS67	3	20	31	-0.1162E + 04	0.3278E - 14	0.1000E - 08	0
HS70	4	9	50	0.7498E - 02	0.1873E - 09	0.1000E - 08	0
HS761	4	11	22	-0.4682E + 01	0.1135E - 09	0.1000E - 08	0
HS76	4	7	20	-0.4682E + 01	0.1080E - 07	0.1000E - 05	7
HS84	5	16	36	-0.5280E + 07	0.5078E - 08	0.1000E - 04	7
HS85	5	48	31	-0.2216E + 01	0.5922E - 09	0.1000E - 06	7
HS93	6	8	208	0.1351E + 03	0.1862E - 06	0.1000E - 08	7
KSIP	20	1001	53	0.5758E + 00	0.1911E - 08	0.1000E - 09	0
LOGROS	2	2	180	0.0000E + 00	0.1794E - 05	0.1000E - 11	7
MAXLIKA	8	16	67	0.1136E + 04	0.2843E - 04	0.1000E - 05	7
MDHOLE	2	1	147	0.1000E - 07	0.3688E - 07	0.1000E - 07	7
MIFFLIN1	3	2	20	-0.1000E + 01	0.1363E - 06	0.1000E - 04	7
MINMAXBD	5	20	7	0.6530E + 03	0.7408E + 02	0.1000E + 01	6
OSLBQP	8	11	29	0.6250E + 01	0.6741E - 08	0.1803E - 11	0

TAB. 3.5 – Résultats du code NOPTIQ utilisant les mises à jour des facteurs de Cholesky des formules de BFGS approchant le hessien du lagrangien et utilisant la correction de Powell.

nom	n	m	nf	f	test1	test2	info
PALMER1A	6	2	141	0.8988E-01	0.1169E-06	0.1000E-09	7
PALMER1B	4	2	89	0.3447E+01	0.9214E-04	0.1000E-06	7
PALMER1E	8	1	201	0.8352E-03	0.3495E-06	0.1000E-11	7
PALMER1	4	3	894	0.1175E+05	0.1790E-03	0.1000E-05	7
PALMER2A	6	2	233	0.1711E-01	0.2636E-08	0.1000E-10	0
PALMER2B	4	2	65	0.6233E+00	0.4041E-06	0.1000E-07	7
PALMER2E	8	1	275	0.2065E-03	0.3730E-06	0.1000E-10	7
PALMER2	4	3	188	0.3651E+04	0.3128E-09	0.1000E-08	0
PALMER3A	6	2	362	0.2043E-01	0.4772E-08	0.1000E-10	0
PALMER3B	4	2	126	0.4228E+01	0.1842E-05	0.1000E-07	7
PALMER3E	8	1	172	0.5074E-04	0.2632E-08	0.1000E-10	0
PALMER3	4	3	593	0.2266E+04	0.5380E-04	0.1000E-05	7
PALMER4A	6	2	252	0.4061E-01	0.8897E-09	0.1000E-10	0
PALMER4B	4	2	149	0.6835E+01	0.3438E-05	0.1000E-07	7
PALMER4E	8	1	139	0.1480E-03	0.1469E-06	0.1000E-10	7
PALMER4	4	3	37	0.2419E+04	0.1425E-01	0.1000E-03	6
PALMER5A	8	2	10000	0.4072E-01	0.1278E-01	0.1000E-03	5
PALMER5B	9	2	1235	0.9752E-02	0.2318E-08	0.1000E-11	0
PALMER5D	8	4	38	0.8734E+02	0.2650E-07	0.1000E-07	7
PALMER5E	8	1	10000	0.2095E-01	0.1982E-01	0.1000E-02	5
PALMER6A	6	2	445	0.5595E-01	0.6779E-06	0.1000E-07	7
PALMER6E	8	1	187	0.2240E-03	0.9993E-09	0.1000E-11	0
PALMER7A	6	2	10000	0.1034E+02	0.4400E-01	0.1000E-03	5
PALMER7E	8	1	1121	0.1015E+02	0.4510E-03	0.1000E-06	7
PALMER8A	6	2	221	0.7401E-01	0.5886E-07	0.1000E-08	7
PALMER8E	8	1	96	0.6339E-02	0.9264E-08	0.1000E-10	0
PROBPENL	500	1000	10	0.4065E-06	0.3832E-01	0.1000E-05	7
PSPDOC	4	1	27	0.2414E+01	0.5549E-08	0.1000E-05	7
S368	8	16	74	-0.4375E+00	0.4514E-07	0.1000E-06	7
SIM2BQP	2	2	20	0.1001E-05	0.2831E-06	0.1000E-04	7
SIMBQP	2	2	23	0.1000E-07	0.3950E-06	0.1000E-06	7
SIPOW1M	2	2000	22	-0.1000E+01	0.2132E-13	0.1018E-08	0
SIPOW1	2	2000	23	-0.1000E+01	0.1199E-13	0.1014E-08	0
SIPOW2M	2	2000	52	-0.1000E+01	0.1288E-13	0.1021E-08	0
SIPOW2	2	2000	51	-0.1000E+01	0.4441E-14	0.1128E-08	0
SIPOW3	4	2000	30	0.5347E+00	0.1278E-13	0.1009E-08	0
SIPOW4	4	2000	29	0.2724E+00	0.5381E-14	0.1113E-08	0
SPECAN	9	18	69	0.4890E-12	0.5611E-03	0.1000E-06	7
SPIRAL	3	2	236	0.2000E-07	0.1128E-08	0.1000E-06	7
WEEDS	3	4	77	0.2587E+01	0.5287E-06	0.1000E-07	7
WOMFLET	3	3	116	0.3914E+01	0.1034E+01	0.1002E+01	2
YFIT	3	1	158	0.6670E-12	0.8006E-09	0.1000E-09	0
ZY2	3	6	19	0.2000E+01	0.1702E-05	0.1000E-04	7

TAB. 3.6 – Résultats du code NOPTIQ utilisant les mises à jour des facteurs de Cholesky des formules de BFGS approchant le hessien du lagrangien et utilisant la correction de Powell.

nom	n	m	nf	f	test1	test2	info
3PK	30	30	10000	0.3656E + 04	0.1700E + 03	0.1000E + 00	5
ALLINIT	4	3	1	0.3789E + 40	0.8437E + 40	0.1000E + 01	6
BQP1VAR	1	2	13	0.1000E - 07	0.1266E - 15	0.1000E - 07	7
BQPGABIM	50	92	41	-0.3790E - 04	0.1033E - 04	0.1000E - 09	7
BQPGASIM	50	100	47	-0.5520E - 04	0.6533E - 05	0.1000E - 09	7
CAMEL6	2	4	27	-0.1032E + 01	0.2317E - 11	0.1000E - 06	7
CHEBYQAD	100	200	6341	0.4878E - 02	0.1510E - 05	0.1000E - 06	7
DECONVB	61	72	10000	0.9089E - 08	0.9503E - 04	0.1000E - 10	5
DIPIGRI	7	4	24	0.6806E + 03	0.2622E - 07	0.1000E - 07	7
EG1	3	4	41	-0.1429E + 01	0.9584E - 10	0.1000E - 05	7
HART6	6	12	48	-0.3323E + 01	0.1133E - 06	0.1000E - 06	7
HATFLDA	4	4	46	0.1152E - 13	0.5133E - 08	0.1000E - 07	0
HATFLDB	4	5	36	0.5573E - 02	0.2270E - 08	0.1000E - 07	7
HATFLDC	25	48	44	0.4711E - 15	0.3533E - 07	0.1000E - 06	7
HIMMELP1	2	4	175	-0.6205E + 02	0.1831E - 05	0.1000E - 04	7
HS100MOD	7	4	28	0.6787E + 03	0.8059E - 05	0.1000E - 05	7
HS100	7	4	24	0.6806E + 03	0.2622E - 07	0.1000E - 07	7
HS110	200	400	1	-0.1478E + 39	0.4645E + 38	0.1000E + 01	6
HS113	10	8	21	0.2431E + 02	0.6133E - 09	0.1000E - 08	0
HS117	15	20	51	0.3235E + 02	0.7263E - 08	0.1000E - 09	0
HS12	2	1	19	-0.3000E + 02	0.2627E - 08	0.1000E - 06	7
HS1	2	1	10000	0.2126E - 04	0.2479E - 01	0.1000E - 03	5
HS24	2	5	20	-0.1000E + 01	0.3899E - 07	0.1000E - 04	7
HS25	3	6	10000	0.2248E + 00	0.8250E - 04	0.1000E + 01	5
HS29	3	1	26	-0.2263E + 02	0.9758E - 09	0.1000E - 04	7
HS2	2	1	915	0.4941E + 01	0.9019E - 05	0.9999E - 07	7
HS35I	3	7	21	0.1111E + 00	0.4660E - 12	0.1000E - 08	0
HS35MOD	3	3	24	0.2500E + 00	0.6316E - 10	0.1789E - 08	0
HS35	3	4	20	0.1111E + 00	0.3480E - 08	0.1000E - 05	7
HS36	3	7	19	-0.3300E + 04	0.8274E - 12	0.1000E - 06	7
HS37	3	8	15	-0.3456E + 04	0.8213E - 06	0.1000E - 05	7
HS38	4	8	10000	0.4391E - 01	0.1636E + 01	0.1000E - 02	5
HS3MOD	2	1	43	0.1000E - 07	0.2673E - 08	0.1000E - 07	7
HS3	2	1	10000	0.1001E + 01	0.2546E - 03	0.1000E + 01	5
HS43	4	3	39	-0.4400E + 02	0.9304E - 10	0.1000E - 07	7
HS44NEW	4	10	19	-0.1500E + 02	0.3091E - 07	0.1000E - 03	7
HS45	5	10	149	0.1001E + 01	0.2725E - 05	0.1000E - 02	7
HS4	2	2	14	0.2667E + 01	0.3496E - 13	0.1000E - 07	7
HS57	2	3	10000	0.3065E - 01	0.3093E - 02	0.1000E - 02	5
HS5	2	4	21	-0.1913E + 01	0.2744E - 09	0.1000E - 07	7
HS67	3	20	50	-0.1162E + 04	0.6122E - 11	0.1000E - 08	0
HS70	4	9	635	0.7498E - 02	0.1537E - 09	0.1000E - 08	0
HS76I	4	11	21	-0.4682E + 01	0.1344E - 07	0.1000E - 06	7
HS76	4	7	20	-0.4682E + 01	0.2253E - 08	0.1000E - 06	7
HS84	5	16	3855	-0.5280E + 07	0.1554E - 01	0.1000E - 04	7
HS85	5	48	10000	-0.1812E + 01	0.6762E - 02	0.1000E + 00	5
HS93	6	8	10000	0.1351E + 03	0.2552E - 01	0.1000E - 03	5
KSIP	20	1001	53	0.5758E + 00	0.1911E - 08	0.1000E - 09	0
LOGROS	2	2	690	0.0000E + 00	0.9333E - 07	0.1000E - 11	7
MAXLIKA	8	16	10000	0.1137E + 04	0.3868E + 00	0.1000E - 01	5
MDHOLE	2	1	5416	0.1001E - 05	0.5412E - 04	0.1000E - 04	7
MIFFLINI	3	2	16	-0.1000E + 01	0.1140E - 07	0.1000E - 04	7
MINMAXBD	5	20	130	0.1157E + 03	0.1278E - 08	0.1000E - 08	0
OSLBQP	8	11	23	0.6250E + 01	0.1960E - 15	0.1788E - 08	0

TAB. 3.7 – Résultats du code NOPTIQ avec les formules de BFGS approchant le hessien du lagrangien et utilisant la méthode du skipping.

nom	n	m	nf	f	test1	test2	info
PALMER1A	6	2	10000	0.1412E+05	0.1694E+04	0.9980E+00	5
PALMER1B	4	2	10000	0.1984E+05	0.1512E+05	0.1000E+01	5
PALMER1E	8	1	10000	0.3177E+05	0.2997E+04	0.1000E-01	5
PALMER1	4	3	10000	0.1227E+05	0.3193E+04	0.1000E+01	5
PALMER2A	6	2	10000	0.1113E+03	0.4770E+02	0.1000E+00	5
PALMER2B	4	2	10000	0.7361E+03	0.2409E+03	0.9809E+00	5
PALMER2E	8	1	10000	0.6707E+03	0.2164E+03	0.9997E-02	5
PALMER2	4	3	4117	0.3651E+04	0.1430E-02	0.1000E-04	7
PALMER3A	6	2	10000	0.5571E+01	0.5117E+01	0.1000E-01	5
PALMER3B	4	2	284	0.4228E+01	0.3202E-06	0.1000E-07	7
PALMER3E	8	1	10000	0.1386E+03	0.1673E+03	0.1000E-01	5
PALMER3	4	3	9932	0.2266E+04	0.5700E-05	0.1000E-06	7
PALMER4A	6	2	10000	0.2888E+02	0.4388E+02	0.1080E+00	5
PALMER4B	4	2	10000	0.6855E+01	0.3523E+01	0.1000E-01	5
PALMER4E	8	1	10000	0.1386E+03	0.1643E+03	0.1000E-01	5
PALMER4	4	3	10000	0.2424E+04	0.4696E-03	0.1000E-04	5
PALMER5A	8	2	10000	0.5033E+04	0.1297E+04	0.1000E+01	5
PALMER5B	9	2	10000	0.6454E+04	0.1050E+04	0.1032E+00	5
PALMER5D	8	4	10000	0.1135E+03	0.1962E+02	0.1000E+00	5
PALMER5E	8	1	10000	0.2243E-01	0.1596E-02	0.1000E-03	5
PALMER6A	6	2	10000	0.1031E+02	0.2220E+02	0.1000E+00	5
PALMER6E	8	1	10000	0.2144E+02	0.1658E+02	0.1000E-01	5
PALMER7A	6	2	10000	0.1580E+02	0.5556E+01	0.1000E-01	5
PALMER7E	8	1	10000	0.7478E+02	0.4546E+02	0.1000E-01	5
PALMER8A	6	2	10000	0.8424E-01	0.4293E+00	0.1000E-02	5
PALMER8E	8	1	10000	0.3506E+02	0.1259E+02	0.1000E-01	5
PROBPENL	500	1000	5	0.3992E-06	0.2823E-06	0.1000E+00	7
PSPDOC	4	1	27	0.2414E+01	0.2870E-06	0.1000E-04	7
S368	8	16	25	-0.7500E+00	0.5988E-08	0.1000E-05	7
SIM2BQP	2	2	13	0.1000E-07	0.9483E-16	0.1000E-07	7
SIMBQP	2	2	14	0.1000E-07	0.1424E-07	0.1000E-07	7
SIPOW1M	2	2000	27	-0.1000E+01	0.1799E-07	0.1000E-08	7
SIPOW1	2	2000	32	-0.1000E+01	0.4237E-07	0.1002E-09	7
SIPOW2M	2	2000	47	-0.1000E+01	0.1341E-07	0.1002E-09	7
SIPOW2	2	2000	63	-0.1000E+01	0.1691E-07	0.1000E-09	7
SIPOW3	4	2000	44	0.5347E+00	0.4700E-07	0.1009E-08	7
SIPOW4	4	2000	70	0.2724E+00	0.4495E-07	0.1113E-08	7
SPECAN	9	18	6453	0.6618E-08	0.6953E-02	0.1000E-05	7
SPIRAL	3	2	10000	0.4424E-01	0.1067E-01	0.1000E+00	5
WEEDS	3	4	336	0.2587E+01	0.8584E-08	0.1000E-08	0
WOMFLET	3	3	10000	0.3501E+01	0.1067E+01	0.1002E+01	5
YFIT	3	1	5866	0.6670E-12	0.3725E-07	0.1000E-09	7
ZY2	3	6	19	0.2000E+01	0.2958E-07	0.1000E-04	7

TAB. 3.8 – Résultats du code NOPTIQ avec les formules de BFGS approchant le hessien du lagrangien et utilisant la méthode du skipping.

nom	n	m	nf	f	test1	test2	info
3PK	30	30	10000	0.9389E + 04	0.1716E + 05	0.1000E + 01	5
ALLINIT	4	3	1	0.3789E + 40	0.8437E + 40	0.1000E + 01	6
BQP1VAR	1	2	13	0.1000E - 07	0.9540E - 16	0.1000E - 07	7
BQPGABIM	50	92	10000	0.8793E - 02	0.1824E + 02	0.1275E - 01	5
BQPGASIM	50	100	10000	0.1176E - 01	0.8117E + 01	0.1996E - 01	5
CAMEL6	2	4	217	-0.1032E + 01	0.1088E - 05	0.1000E - 05	7
CHEBYQAD	100	200	10000	0.1571E - 01	0.1527E - 01	0.1000E - 02	5
DECONVB	61	72	10000	0.1015E - 04	0.3467E - 02	0.1000E - 09	5
DIPIGRI	7	4	1731	0.6806E + 03	0.8140E - 05	0.1000E - 05	7
EG1	3	4	39	-0.1429E + 01	0.2103E - 11	0.1000E - 06	7
HART6	6	12	1498	-0.3323E + 01	0.1062E - 05	0.1001E - 05	7
HATFLDA	4	4	169	0.1151E - 13	0.8231E - 08	0.1000E - 07	0
HATFLDB	4	5	31	0.5573E - 02	0.1101E - 08	0.1000E - 08	0
HATFLDC	25	48	40	0.4816E - 13	0.3577E - 06	0.1000E - 05	7
HIMMELP1	2	4	669	-0.6205E + 02	0.1923E - 05	0.1000E - 04	7
HS100MOD	7	4	269	0.6787E + 03	0.5958E - 06	0.1000E - 06	7
HS100	7	4	1709	0.6806E + 03	0.8954E - 05	0.1000E - 05	7
HS110	200	400	1	-0.1478E + 39	0.4645E + 38	0.1000E + 01	6
HS113	10	8	24	0.2431E + 02	0.6187E - 07	0.1000E - 07	7
HS117	15	20	454	0.3235E + 02	0.2592E - 06	0.1001E - 08	7
HS12	2	1	19	-0.3000E + 02	0.1284E - 10	0.1000E - 06	7
HS1	2	1	10000	0.1896E - 04	0.2189E - 01	0.1012E - 03	5
HS24	2	5	20	-0.1000E + 01	0.3899E - 07	0.1000E - 04	7
HS25	3	6	10000	0.1766E + 01	0.8250E - 04	0.1000E + 01	5
HS29	3	1	28	-0.2263E + 02	0.1183E - 07	0.1000E - 04	7
HS2	2	1	10000	0.5055E - 01	0.3434E - 02	0.1501E - 03	5
HS35I	3	7	24	0.1111E + 00	0.9571E - 08	0.1000E - 05	7
HS35MOD	3	3	24	0.2500E + 00	0.6672E - 13	0.1788E - 08	0
HS35	3	4	24	0.1111E + 00	0.8821E - 08	0.1000E - 06	7
HS36	3	7	20	-0.3300E + 04	0.3932E - 06	0.1000E - 06	7
HS37	3	8	135	-0.3456E + 04	0.3258E - 07	0.1000E - 07	4
HS38	4	8	10000	0.1284E + 01	0.2208E + 03	0.9979E + 00	5
HS3MOD	2	1	17	0.9997E - 08	0.2801E - 10	0.9997E - 08	0
HS3	2	1	6	0.1000E + 00	0.9259E - 08	0.1000E + 01	4
HS43	4	3	39	-0.4400E + 02	0.1367E - 10	0.1000E - 07	7
HS44NEW	4	10	25	-0.1500E + 02	0.7623E - 06	0.1000E - 03	7
HS45	5	10	149	0.1000E + 01	0.1148E - 06	0.1000E - 03	7
HS4	2	2	16	0.2667E + 01	0.1826E - 12	0.1000E - 07	7
HS57	2	3	10000	0.3065E - 01	0.1965E - 05	0.1000E - 05	5
HS5	2	4	19	-0.1913E + 01	0.4261E - 13	0.1000E - 07	7
HS67	3	20	38	-0.1162E + 04	0.3712E - 12	0.1001E - 08	0
HS70	4	9	10000	0.8117E - 02	0.1744E - 02	0.1268E - 02	5
HS76I	4	11	20	-0.4682E + 01	0.9033E - 07	0.1000E - 05	7
HS76	4	7	22	-0.4682E + 01	0.1178E - 08	0.1000E - 06	7
HS84	5	16	30	-0.5280E + 07	0.3239E + 00	0.1000E - 01	7
HS85	5	48	56	-0.2216E + 01	0.2217E - 10	0.1000E - 06	7
HS93	6	8	10000	0.1351E + 03	0.1147E + 01	0.1087E + 00	5
KSIP	20	1001	61	0.5758E + 00	0.1069E - 08	0.1000E - 09	0
LOGROS	2	2	10000	0.3353E + 00	0.1878E + 01	0.6939E - 06	5
MAXLIKA	8	16	10000	0.1138E + 04	0.2902E + 01	0.1983E + 00	5
MDHOLE	2	1	10000	0.9485E - 03	0.3502E - 01	0.1352E - 02	5
MIFFLINI	3	2	17	-0.1000E + 01	0.5847E - 12	0.9908E - 09	0
MINMAXBD	5	20	394	0.1157E + 03	0.5351E - 09	0.1000E - 08	0
OSLBQP	8	11	23	0.6250E + 01	0.1960E - 15	0.1788E - 08	0

TAB. 3.9 – Résultats du code NOPTIQ avec les formules de BFGS approchant l'inverse du hessien du lagrangien et utilisant la méthode du skipping.

nom	n	m	nf	f	test1	test2	info
PALMER1A	6	2	10000	0.4572E + 04	0.1637E + 04	0.9997E + 00	5
PALMER1B	4	2	10000	0.1103E + 05	0.1483E + 04	0.5186E + 00	5
PALMER1E	8	1	10000	0.2917E + 05	0.1692E + 06	0.1000E + 01	5
PALMER1	4	3	1844	0.2188E + 05	0.3193E + 04	0.1000E + 01	6
PALMER2A	6	2	10000	0.1546E + 03	0.2249E + 03	0.9990E + 00	5
PALMER2B	4	2	10000	0.9335E + 03	0.2409E + 03	0.9809E + 00	5
PALMER2E	8	1	10000	0.9459E + 03	0.8990E + 04	0.1000E + 01	5
PALMER2	4	3	10000	0.3651E + 04	0.1368E + 03	0.4037E + 00	5
PALMER3A	6	2	10000	0.2943E + 02	0.5424E + 02	0.9997E + 00	5
PALMER3B	4	2	10000	0.4471E + 02	0.7619E + 02	0.9911E + 00	5
PALMER3E	8	1	10000	0.1643E + 03	0.2514E + 04	0.1000E + 01	5
PALMER3	4	3	10000	0.2417E + 04	0.1337E - 01	0.1000E - 03	5
PALMER4A	6	2	10000	0.2673E + 02	0.4392E + 02	0.9997E + 00	5
PALMER4B	4	2	10000	0.1334E + 03	0.7104E + 02	0.9910E + 00	5
PALMER4E	8	1	10000	0.1831E + 03	0.1313E + 05	0.1000E + 01	5
PALMER4	4	3	10000	0.2424E + 04	0.8579E - 02	0.1925E - 03	5
PALMER5A	8	2	10000	0.4074E + 04	0.1297E + 04	0.1000E + 01	5
PALMER5B	9	2	10000	0.1258E + 03	0.1502E + 03	0.1980E - 01	5
PALMER5D	8	4	10000	0.1446E + 03	0.1617E + 02	0.8794E - 01	5
PALMER5E	8	1	10000	0.2989E + 00	0.4570E - 01	0.1000E - 02	5
PALMER6A	6	2	10000	0.1034E + 02	0.3652E + 02	0.9973E + 00	5
PALMER6E	8	1	10000	0.1556E + 03	0.9394E + 03	0.1000E + 01	5
PALMER7A	6	2	10000	0.1693E + 02	0.3952E + 02	0.1917E + 00	5
PALMER7E	8	1	10000	0.7978E + 02	0.2371E + 04	0.1000E + 01	5
PALMER8A	6	2	10000	0.1005E + 02	0.2167E + 02	0.1935E + 00	5
PALMER8E	8	1	10000	0.3890E + 02	0.1025E + 04	0.1000E + 01	5
PROBPENL	500	1000	10000	0.5727E - 01	0.2439E + 01	0.9918E + 00	5
PSPDOC	4	1	28	0.2414E + 01	0.4602E - 07	0.1000E - 05	7
S368	8	16	32	-0.7500E + 00	0.2651E - 07	0.1000E - 06	7
SIM2BQP	2	2	13	0.1000E - 07	0.9483E - 16	0.1000E - 07	7
SIMBQP	2	2	40	0.1000E - 07	0.1929E - 07	0.1000E - 07	7
SIPOW1M	2	2000	3	0.0000E + 00	0.0000E + 00	0.0000E + 00	0
SIPOW1	2	2000	3	0.0000E + 00	0.0000E + 00	0.0000E + 00	0
SIPOW2M	2	2000	3	0.0000E + 00	0.0000E + 00	0.0000E + 00	0
SIPOW2	2	2000	3	0.0000E + 00	0.0000E + 00	0.0000E + 00	0
SIPOW3	4	2000	3	0.0000E + 00	0.0000E + 00	0.0000E + 00	0
SIPOW4	4	2000	3	0.0000E + 00	0.0000E + 00	0.0000E + 00	0
SPECAN	9	18	10000	0.1057E + 00	0.3128E + 04	0.9996E + 00	5
SPIRAL	3	2	10000	0.3017E + 00	0.1196E + 00	0.9956E + 00	5
WEEDS	3	4	10000	0.1486E + 04	0.8251E + 03	0.1000E + 01	5
WOMFLET	3	3	1460	0.3000E - 05	0.6150E - 07	0.1013E - 04	7
YFIT	3	1	10000	0.2046E + 03	0.2452E + 03	0.1000E + 01	5
ZY2	3	6	19	0.2000E + 01	0.2958E - 07	0.1000E - 04	7

TAB. 3.10 – Résultats du code NOPTIQ avec les formules de BFGS approchant l'inverse du hessien du lagrangien et utilisant la méthode du skipping.

nom	n	m	nf	f	test1	test2	info
3PK	30	30	10000	0.2151E + 01	0.1716E + 01	0.1000E - 02	5
ALLINIT	4	3	1	0.3789E + 40	0.8437E + 40	0.1000E + 01	6
BQPIVAR	1	2	18	0.1000E - 05	0.7109E - 07	0.1000E - 04	7
BQPGABIM	50	92	66	-0.3790E - 04	0.1260E - 04	0.1000E - 09	7
BQPGASIM	50	100	72	-0.5520E - 04	0.1821E - 04	0.1000E - 09	7
CAMEL6	2	4	35	-0.1032E + 01	0.1114E - 06	0.1000E - 06	7
CHEBYQAD	100	200	3568	0.4878E - 02	0.1759E - 05	0.1000E - 06	7
DECONVB	61	72	893	0.6433E - 13	0.3201E - 08	0.1000E - 15	0
DIPIGRI	7	4	30	0.6806E + 03	0.1424E - 06	0.1000E - 06	7
EG1	3	4	26	-0.1429E + 01	0.3587E - 06	0.1000E - 04	7
HART6	6	12	75	-0.3323E + 01	0.3590E - 06	0.1000E - 05	7
HATFLDA	4	4	38	0.1152E - 13	0.5717E - 08	0.1000E - 07	0
HATFLDB	4	5	36	0.5573E - 02	0.1293E - 08	0.1000E - 08	0
HATFLDC	25	48	77	0.6948E - 13	0.3267E - 06	0.1000E - 05	7
HIMMELP1	2	4	175	-0.6205E + 02	0.1831E - 05	0.1000E - 04	7
HS100MOD	7	4	33	0.6787E + 03	0.8437E - 06	0.1000E - 06	7
HS100	7	4	29	0.6806E + 03	0.7418E - 05	0.1000E - 05	7
HS110	200	400	1	-0.1478E + 39	0.4645E + 38	0.1000E + 01	6
HS113	10	8	25	0.2431E + 02	0.1140E - 07	0.1000E - 08	7
HS117	15	20	55	0.3235E + 02	0.1424E - 07	0.1000E - 09	7
HS12	2	1	27	-0.3000E + 02	0.4707E - 06	0.1000E - 05	7
HS1	2	1	926	0.7778E - 14	0.2363E - 05	0.1000E - 07	7
HS24	2	5	20	-0.1000E + 01	0.3899E - 07	0.1000E - 04	7
HS25	3	6	10000	0.2343E - 01	0.8250E - 04	0.1000E + 01	5
HS29	3	1	31	-0.2263E + 02	0.4463E - 07	0.1000E - 05	7
HS2	2	1	30	0.4941E + 01	0.1445E - 07	0.1000E - 07	7
HS35I	3	7	27	0.1111E + 00	0.2649E - 07	0.1000E - 06	7
HS35MOD	3	3	60	0.2500E + 00	0.1124E - 07	0.1000E - 06	7
HS35	3	4	28	0.1111E + 00	0.3173E - 07	0.1000E - 06	7
HS36	3	7	18	-0.3300E + 04	0.7062E - 07	0.1000E - 06	7
HS37	3	8	26	-0.3456E + 04	0.3210E - 07	0.1000E - 07	7
HS38	4	8	10000	0.3089E + 00	0.1636E + 01	0.1000E - 02	5
HS3MOD	2	1	18	0.1000E - 07	0.2654E - 09	0.1000E - 07	0
HS3	2	1	10000	0.1001E + 01	0.2546E - 03	0.1000E + 01	5
HS43	4	3	35	-0.4400E + 02	0.4621E - 07	0.1000E - 06	7
HS44NEW	4	10	20	-0.1500E + 02	0.6750E - 07	0.1000E - 03	7
HS45	5	10	149	0.1001E + 01	0.2725E - 05	0.1000E - 02	7
HS4	2	2	16	0.2667E + 01	0.8997E - 13	0.1000E - 07	7
HS57	2	3	10000	0.3065E - 01	0.3093E - 02	0.1000E - 02	5
HS5	2	4	51	-0.1913E + 01	0.1196E - 07	0.1000E - 06	7
HS67	3	20	50	-0.1162E + 04	0.6125E - 11	0.1000E - 08	0
HS70	4	9	209	0.7498E - 02	0.1654E - 08	0.1000E - 08	0
HS76I	4	11	22	-0.4682E + 01	0.1135E - 09	0.1000E - 08	0
HS76	4	7	20	-0.4682E + 01	0.1080E - 07	0.1000E - 05	7
HS84	5	16	4171	-0.5280E + 07	0.1660E - 01	0.1000E - 04	7
HS85	5	48	10000	-0.1812E + 01	0.6762E - 02	0.1000E + 00	5
HS93	6	8	10000	0.1351E + 03	0.2552E - 01	0.1000E - 03	5
KSIP	20	1001	53	0.5758E + 00	0.1911E - 08	0.1000E - 09	0
LOGROS	2	2	3416	0.1150E - 11	0.7069E - 04	0.1000E - 10	7
MAXLIKA	8	16	1519	0.1136E + 04	0.5301E - 05	0.1000E - 05	7
MDHOLE	2	1	1822	0.1000E - 07	0.1900E - 06	0.1000E - 07	7
MIFFLIN1	3	2	19	-0.1000E + 01	0.3447E - 10	0.1000E - 05	7
MINMAXBD	5	20	130	0.1157E + 03	0.1278E - 08	0.1000E - 08	0
OSLBQP	8	11	29	0.6250E + 01	0.6741E - 08	0.1803E - 11	0

TAB. 3.11 – Résultats du code NOPTIQ utilisant les mises à jour des facteurs de Cholesky des formules de BFGS approchant le hessien du lagrangien et utilisant la méthode du skipping.

nom	n	m	nf	f	test1	test2	info
PALMER1A	6	2	10000	0.1015E+05	0.2244E+04	0.1000E+01	5
PALMER1B	4	2	10000	0.1330E+05	0.1512E+04	0.1000E+01	5
PALMER1E	8	1	10000	0.8355E+02	0.3533E+02	0.1000E-03	5
PALMER1	4	3	10000	0.1239E+05	0.3193E+04	0.1000E+01	5
PALMER2A	6	2	10000	0.8332E+02	0.4770E+02	0.1000E+00	5
PALMER2B	4	2	10000	0.2475E+03	0.2409E+03	0.9809E+00	5
PALMER2E	8	1	10000	0.4485E+00	0.4030E+00	0.1000E-04	5
PALMER2	4	3	4630	0.3651E+04	0.1429E-02	0.1000E-04	7
PALMER3A	6	2	1300	0.2043E-01	0.8176E-08	0.1000E-09	0
PALMER3B	4	2	10000	0.4228E+01	0.3324E-01	0.1000E-03	5
PALMER3E	8	1	10000	0.8675E+00	0.1674E+01	0.1000E-03	5
PALMER3	4	3	1429	0.2266E+04	0.1463E-03	0.1000E-05	7
PALMER4A	6	2	10000	0.1824E+00	0.5590E+01	0.1000E-01	5
PALMER4B	4	2	4827	0.6835E+01	0.3520E-04	0.1000E-06	7
PALMER4E	8	1	10000	0.7145E+00	0.1645E+01	0.1000E-03	5
PALMER4	4	3	10000	0.2326E+04	0.1177E+00	0.1000E-02	5
PALMER5A	8	2	10000	0.3877E+04	0.1297E+04	0.1000E+01	5
PALMER5B	9	2	10000	0.3514E+04	0.1631E+04	0.1000E+00	5
PALMER5D	8	4	10000	0.8786E+02	0.1964E+01	0.1000E-01	5
PALMER5E	8	1	10000	0.2391E-01	0.4347E-02	0.1000E-03	5
PALMER6A	6	2	10000	0.1053E+00	0.2224E+00	0.1000E-02	5
PALMER6E	8	1	10000	0.5341E-01	0.1656E-01	0.1000E-04	5
PALMER7A	6	2	10000	0.1310E+02	0.5553E+00	0.1000E-02	5
PALMER7E	8	1	10000	0.1139E+02	0.4550E+00	0.1000E-03	5
PALMER8A	6	2	10000	0.1094E+00	0.4289E+01	0.1000E-01	5
PALMER8E	8	1	10000	0.1227E+01	0.1260E+01	0.1000E-02	5
PROBPENL	500	1000	10	0.4065E-06	0.3832E-01	0.1000E-05	7
PSPDOC	4	1	27	0.2414E+01	0.5549E-08	0.1000E-05	7
S368	8	16	28	-0.7500E+00	0.2760E-07	0.1000E-06	7
SIM2BQP	2	2	20	0.1001E-05	0.2831E-06	0.1000E-04	7
SIMBQP	2	2	23	0.1000E-07	0.3950E-06	0.1000E-06	7
SIPOW1M	2	2000	27	-0.1000E+01	0.1799E-07	0.1000E-08	7
SIPOW1	2	2000	32	-0.1000E+01	0.4237E-07	0.1002E-09	7
SIPOW2M	2	2000	47	-0.1000E+01	0.1341E-07	0.1002E-09	7
SIPOW2	2	2000	63	-0.1000E+01	0.1691E-07	0.1000E-09	7
SIPOW3	4	2000	44	0.5347E+00	0.4700E-07	0.1009E-08	7
SIPOW4	4	2000	70	0.2724E+00	0.4495E-07	0.1113E-08	7
SPECAN	9	18	2724	0.3307E-09	0.6941E-03	0.1000E-06	7
SPIRAL	3	2	10000	0.3193E+00	0.7340E-01	0.1000E+01	5
WEEDS	3	4	81	0.2587E+01	0.1013E-07	0.1000E-08	7
WOMFLET	3	3	8947	0.3586E+01	0.1034E+01	0.1002E+01	6
YFIT	3	1	6752	0.6670E-12	0.1507E-07	0.1000E-09	7
ZY2	3	6	19	0.2000E+01	0.2958E-07	0.1000E-04	7

TAB. 3.12 – Résultats du code NOPTIQ utilisant les mises à jour des facteurs de Cholesky des formules de BFGS approchant le hessien du lagrangien et utilisant la méthode du skipping.

Résultats numériques de NOPTIQ (chapitre 2)

On présente dans cette partie les résultats numériques de NOPTIQ utilisant les formules de BFGS à mémoire limitée. Les problèmes testés sont ceux des tableaux (2.1), (2.2) et (2.3). Les tableaux suivants énumèrent ces résultats où

- nom représente le nom du problème testé,
 - n représente le nombre total de variables,
 - m représente le nombre total de contraintes,
 - nf représente le nombre total d'appels au simulateur pour évaluer f et c,
 - f représente la valeur finale de f au dernier point calculé par NOPTIQ,
 - test représente le maximum de la valeur de la norme euclidienne du gradient du lagrangien et la valeur de la norme euclidienne des conditions de complémentarité au dernier point calculé par NOPTIQ,
 - tmain temps total de résolution du problème,
 - tcode temps total passé dans le code NOPTIQ ($tcode = tmain - tfunc$),
 - tfunc temps total passé dans le simulateur,
 - info représente le mode d'arrêt de NOPTIQ
 - 0 NOPTIQ stoppe normalement,
 - 2 NOPTIQ stoppe car la matrice M devient singulière,
 - 5 NOPTIQ stoppe car le nombre maximal d'appels au simulateur est atteint,
 - 6 NOPTIQ stoppe car le pas initial dans la recherche linéaire est trop petit,
 - 7 NOPTIQ stoppe car la direction dans l'espace primal est trop petite.
- Si les conditions d'optimalité sont pratiquement satisfaites, cette sortie n'est pas un échec pour le code.

nom	n	m	nf	f	test	tmain	tcode	tfunc	info
BDEXP	5000	5000	11	0.1740E-09	0.1406E-08	0.2800E+00	0.2000E+00	0.8000E-01	0
BDEXP	1000	1000	11	0.1581E-09	0.1276E-08	0.5000E-01	0.2000E-01	0.3000E-01	0
BQPGABIM	50	92	53	-0.3790E-04	0.1227E-04	0.1000E-01	0.1000E-01	0.0000E+00	7
BQPGASIM	50	100	50	-0.5520E-04	0.3284E-04	0.1000E-01	0.0000E+00	0.1000E-01	7
CVXBQP1	1000	2000	36	0.2252E+05	0.1118E-06	0.1700E+00	0.1300E+00	0.4000E-01	7
EXPLIN2	1200	2400	115	-0.7200E+08	0.3435E-03	0.5800E+00	0.5000E+00	0.8000E-01	7
EXPLIN	1200	2400	255	-0.7193E+08	0.8773E-05	0.1300E+01	0.1180E+01	0.1200E+00	7
JNLBRNG1	10000	9604	1093	-0.1806E+00	0.1937E-03	0.8952E+02	0.4985E+02	0.3967E+02	7
JNLBRNG2	10000	9604	2458	-0.4149E+01	0.1163E-05	0.1643E+03	0.9034E+02	0.7392E+02	7
JNLBRNGA	10000	9604	3	0.7797E-13	0.1282E-02	0.1700E+00	0.9000E-01	0.8000E-01	7
JNLBRNGB	10000	9604	3	0.4857E-12	0.6410E-02	0.1400E+00	0.4000E-01	0.1000E+00	7
LINVERSE	1999	1000	286	0.6820E+03	0.2810E+00	0.3930E+01	0.1710E+01	0.2220E+01	7
LINVERSE	999	500	235	0.3408E+03	0.3104E+01	0.1430E+01	0.6900E+00	0.7400E+00	7
MCCORMCK	10000	20000	24	-0.9133E+04	0.3489E-07	0.2020E+01	0.1410E+01	0.6100E+00	7
MCCORMCK	5000	10000	25	-0.4567E+04	0.1206E-06	0.9300E+00	0.7500E+00	0.1800E+00	7
MCCORMCK	1000	2000	22	-0.9137E+03	0.4213E-07	0.1300E+00	0.1000E+00	0.3000E-01	7
NCVXBQP1	1000	2000	95	-0.1987E+09	0.5623E-09	0.3400E+00	0.2600E+00	0.8000E-01	7
NCVXBQP2	1000	2000	152	-0.1332E+09	0.2155E-04	0.5700E+00	0.4600E+00	0.1100E+00	7
NCVXBQP3	1000	2000	586	-0.6494E+08	0.1984E-03	0.2370E+01	0.2140E+01	0.2300E+00	7
NOBNDTOR	5476	10368	1071	-0.4488E+00	0.6400E-06	0.2821E+02	0.1671E+02	0.1150E+02	7
NONSCOMP	1000	2000	52	0.2455E-06	0.1038E-05	0.2500E+00	0.1900E+00	0.6000E-01	7
OBSTCLAE	5625	10658	535	0.1863E+01	0.8237E-06	0.1642E+02	0.1007E+02	0.6350E+01	7
OBSTCLAL	10000	19208	1133	0.1886E+01	0.3990E-06	0.7909E+02	0.4975E+02	0.2934E+02	7
OBSTCLBL	10000	19208	471	0.7275E+01	0.3618E-06	0.2595E+02	0.1603E+02	0.9920E+01	7
OBSTCLBM	10000	19208	484	0.7275E+01	0.1024E-05	0.2667E+02	0.1656E+02	0.1011E+02	7
OBSTCLBU	10000	19208	471	0.7275E+01	0.3618E-06	0.2596E+02	0.1620E+02	0.9760E+01	7
PENTDI	5000	5000	3	0.4996E-04	0.3000E+01	0.7000E-01	0.5000E-01	0.2000E-01	7
PROBPENL	500	1000	6	0.3992E-06	0.2823E-06	0.1000E-01	0.0000E+00	0.1000E-01	7
S368	8	16	25	-0.1000E+01	0.1781E-05	0.1000E-01	0.1000E-01	0.0000E+00	7
TORSION1	5476	10368	751	-0.4281E+00	0.1565E-05	0.2072E+02	0.1242E+02	0.8300E+01	7
TORSION2	5476	10368	751	-0.4281E+00	0.1565E-05	0.2066E+02	0.1244E+02	0.8220E+01	7
TORSION3	5476	10368	419	-0.1213E+01	0.1067E-05	0.1267E+02	0.7380E+01	0.5290E+01	7
TORSION4	5476	10368	419	-0.1213E+01	0.1067E-05	0.1261E+02	0.7640E+01	0.4970E+01	7
TORSION5	5476	10368	226	-0.2859E+01	0.2034E-06	0.7510E+01	0.4430E+01	0.3080E+01	7
TORSION6	5476	10368	226	-0.2859E+01	0.2034E-06	0.7410E+01	0.4510E+01	0.2900E+01	7
TORSIONA	5476	10368	779	-0.4162E+00	0.1878E-05	0.2276E+02	0.1266E+02	0.1010E+02	7
TORSIONB	5476	10368	779	-0.4162E+00	0.1878E-05	0.2264E+02	0.1307E+02	0.9570E+01	7
TORSIONC	5476	10368	393	-0.1201E+01	0.3706E-06	0.1285E+02	0.7420E+01	0.5430E+01	7
TORSIOND	5476	10368	393	-0.1201E+01	0.3706E-06	0.1271E+02	0.7730E+01	0.4980E+01	7
TORSIONE	5476	10368	257	-0.2846E+01	0.2609E-06	0.8670E+01	0.4950E+01	0.3720E+01	7
TORSIONF	5476	10368	257	-0.2846E+01	0.2609E-06	0.8610E+01	0.4870E+01	0.3740E+01	7

TAB. 3.13 – Résultats du code NOPTIQ utilisant les formules de BFGS à mémoire limitée avec $m_\ell = 3$.

nom	n	m	nf	f	test	tmain	tcode	tfunc	info
BDEXP	5000	5000	11	0.1739E-09	0.1405E-08	0.3600E+00	0.2600E+00	0.1000E+00	0
BDEXP	1000	1000	11	0.1574E-09	0.1270E-08	0.6000E-01	0.6000E-01	0.0000E+00	0
BQPGABIM	50	92	48	-0.3790E-04	0.8226E-05	0.2000E-01	0.2000E-01	0.0000E+00	7
BQPGASIM	50	100	41	-0.5520E-04	0.1550E-04	0.2000E-01	0.2000E-01	0.0000E+00	7
CVXBQP1	1000	2000	36	0.2252E+05	0.1882E-06	0.2400E+00	0.2200E+00	0.2000E-01	7
EXPLIN2	1200	2400	107	-0.7200E+08	0.2938E-03	0.8300E+00	0.8000E+00	0.3000E-01	7
EXPLIN	1200	2400	293	-0.7193E+08	0.7685E-02	0.1800E+01	0.1680E+01	0.1200E+00	7
JNLBRNG1	10000	9604	684	-0.1806E+00	0.1156E-02	0.7291E+02	0.4848E+02	0.2443E+02	7
JNLBRNG2	10000	9604	2037	-0.4149E+01	0.7393E-06	0.1627E+03	0.1046E+03	0.5807E+02	7
JNLBRNGA	10000	9604	3	0.7797E-13	0.1282E-02	0.1500E+00	0.4000E-01	0.1100E+00	7
JNLBRNGB	10000	9604	3	0.4857E-12	0.6410E-02	0.1600E+00	0.8000E-01	0.8000E-01	7
LINVERSE	1999	1000	808	0.6811E+03	0.1871E+01	0.1479E+02	0.8150E+01	0.6640E+01	7
LINVERSE	999	500	353	0.3403E+03	0.3065E+01	0.2860E+01	0.1650E+01	0.1210E+01	7
MCCORMCK	10000	20000	22	-0.9133E+04	0.2781E-07	0.2300E+01	0.1730E+01	0.5700E+00	7
MCCORMCK	5000	10000	22	-0.4567E+04	0.9401E-07	0.1020E+01	0.7800E+00	0.2400E+00	7
MCCORMCK	1000	2000	22	-0.9137E+03	0.5561E-07	0.1700E+00	0.1200E+00	0.5000E-01	7
NCVXBQP1	1000	2000	94	-0.1987E+09	0.3888E-09	0.3700E+00	0.3700E+00	0.0000E+00	7
NCVXBQP2	1000	2000	231	-0.1332E+09	0.1746E-01	0.7200E+00	0.6300E+00	0.9000E-01	7
NCVXBQP3	1000	2000	481	-0.6491E+08	0.1600E-03	0.2840E+01	0.2350E+01	0.4900E+00	7
NOBNDTOR	5476	10368	1654	-0.4499E+00	0.3365E-06	0.4937E+02	0.3275E+02	0.1662E+02	7
NONSCOMP	1000	2000	50	0.2455E-06	0.9619E-06	0.3400E+00	0.2600E+00	0.8000E-01	7
OBSTCLAE	5625	10658	479	0.1863E+01	0.5338E-06	0.1886E+02	0.1311E+02	0.5750E+01	7
OBSTCLAL	10000	19208	1354	0.1886E+01	0.4279E-06	0.1268E+03	0.8993E+02	0.3685E+02	7
OBSTCLBL	10000	19208	437	0.7275E+01	0.3154E-06	0.2863E+02	0.1959E+02	0.9040E+01	7
OBSTCLBM	10000	19208	393	0.7275E+01	0.3342E-06	0.2716E+02	0.1877E+02	0.8390E+01	7
OBSTCLBU	10000	19208	437	0.7275E+01	0.3154E-06	0.2867E+02	0.1955E+02	0.9120E+01	7
PENTDI	5000	5000	3	0.4996E-04	0.3000E+01	0.7000E-01	0.5000E-01	0.2000E-01	7
PROBPENL	500	1000	6	0.3992E-06	0.2823E-06	0.2000E-01	0.1000E-01	0.1000E-01	7
S368	8	16	25	-0.9375E+00	0.1728E-07	0.1000E-01	0.1000E-01	0.0000E+00	7
TORSION1	5476	10368	647	-0.4281E+00	0.9783E-06	0.2162E+02	0.1480E+02	0.6820E+01	7
TORSION2	5476	10368	647	-0.4281E+00	0.9783E-06	0.2155E+02	0.1483E+02	0.6720E+01	7
TORSION3	5476	10368	331	-0.1213E+01	0.1027E-05	0.1215E+02	0.7930E+01	0.4220E+01	7
TORSION4	5476	10368	331	-0.1213E+01	0.1027E-05	0.1222E+02	0.8310E+01	0.3910E+01	7
TORSION5	5476	10368	249	-0.2859E+01	0.5880E-06	0.1038E+02	0.6730E+01	0.3650E+01	7
TORSION6	5476	10368	249	-0.2859E+01	0.5880E-06	0.1032E+02	0.7040E+01	0.3280E+01	7
TORSIONA	5476	10368	575	-0.4162E+00	0.2917E-06	0.2133E+02	0.1397E+02	0.7360E+01	7
TORSIONB	5476	10368	575	-0.4162E+00	0.2917E-06	0.2131E+02	0.1396E+02	0.7350E+01	7
TORSIONC	5476	10368	352	-0.1201E+01	0.5929E-06	0.1339E+02	0.8650E+01	0.4740E+01	7
TORSIOND	5476	10368	352	-0.1201E+01	0.5929E-06	0.1332E+02	0.8700E+01	0.4620E+01	7
TORSIONE	5476	10368	245	-0.2846E+01	0.2898E-06	0.9980E+01	0.6440E+01	0.3540E+01	7
TORSIONF	5476	10368	245	-0.2846E+01	0.2898E-06	0.9960E+01	0.6520E+01	0.3440E+01	7

TAB. 3.14 – Résultats du code NOPTIQ utilisant les formules de BFGS à mémoire limitée avec $m_\ell = 5$.

nom	n	m	nf	f	test	tmain	tcode	tfunc	info
BDEXP	5000	5000	11	0.1734E-09	0.1401E-08	0.4300E+00	0.3300E+00	0.1000E+00	0
BDEXP	1000	1000	11	0.1553E-09	0.1253E-08	0.7000E-01	0.6000E-01	0.1000E-01	0
BQPGABIM	50	92	50	-0.3790E-04	0.3372E-05	0.5000E-01	0.5000E-01	0.0000E+00	7
BQPGASIM	50	100	37	-0.5520E-04	0.1166E-04	0.5000E-01	0.5000E-01	0.0000E+00	7
CVXBQPI	1000	2000	36	0.2252E+05	0.1407E-06	0.9100E+00	0.9000E+00	0.1000E-01	7
EXPLIN2	1200	2400	132	-0.7200E+08	0.1747E-02	0.3900E+01	0.3820E+01	0.8000E-01	7
EXPLIN	1200	2400	201	-0.7193E+08	0.8938E-02	0.6010E+01	0.5910E+01	0.1000E+00	7
JNLBRNG1	10000	9604	1110	-0.1806E+00	0.6963E-06	0.4499E+03	0.4131E+03	0.3672E+02	7
JNLBRNG2	10000	9604	1610	-0.4149E+01	0.7808E-06	0.4861E+03	0.4431E+03	0.4299E+02	7
JNLBRNGA	10000	9604	3	0.7797E-13	0.1282E-02	0.1500E+00	0.7000E-01	0.8000E-01	7
JNLBRNGB	10000	9604	3	0.4857E-12	0.6410E-02	0.1700E+00	0.9000E-01	0.8000E-01	7
LINVERSE	1999	1000	218	0.6811E+03	0.1830E+01	0.1634E+02	0.1462E+02	0.1720E+01	7
LINVERSE	999	500	106	0.3400E+03	0.1175E+01	0.2820E+01	0.2420E+01	0.4000E+00	7
MCCORMCK	10000	20000	21	-0.9133E+04	0.7425E-07	0.5320E+01	0.4770E+01	0.5500E+00	7
MCCORMCK	5000	10000	22	-0.4567E+04	0.4970E-07	0.2460E+01	0.2180E+01	0.2800E+00	7
MCCORMCK	1000	2000	21	-0.9137E+03	0.1203E-06	0.3600E+00	0.3400E+00	0.2000E-01	7
NCVXBQPI	1000	2000	94	-0.1987E+09	0.3763E-09	0.5900E+00	0.5000E+00	0.9000E-01	7
NCVXBQP2	1000	2000	155	-0.1332E+09	0.1401E-04	0.1620E+01	0.1520E+01	0.1000E+00	7
NCVXBQP3	1000	2000	548	-0.6494E+08	0.9299E-04	0.1080E+02	0.1050E+02	0.3000E+00	7
NOBNDTOR	5476	10368	866	-0.4488E+00	0.4022E-06	0.8973E+02	0.8059E+02	0.9140E+01	7
NONSCOMP	1000	2000	62	0.2442E-09	0.4610E-06	0.1840E+01	0.1790E+01	0.5000E-01	7
OBSTCLAE	5625	10658	407	0.1863E+01	0.5936E-06	0.5309E+02	0.4788E+02	0.5210E+01	7
OBSTCLAL	10000	19208	899	0.1886E+01	0.2372E-06	0.3047E+03	0.2812E+03	0.2346E+02	7
OBSTCLBL	10000	19208	382	0.7275E+01	0.3087E-06	0.8962E+02	0.8150E+02	0.8120E+01	7
OBSTCLBM	10000	19208	419	0.7275E+01	0.4927E-06	0.9535E+02	0.8669E+02	0.8660E+01	7
OBSTCLBU	10000	19208	382	0.7275E+01	0.3087E-06	0.8946E+02	0.8121E+02	0.8250E+01	7
PENTDI	5000	5000	3	0.4996E-04	0.3000E+01	0.8000E-01	0.6000E-01	0.2000E-01	7
PROBPENL	500	1000	6	0.3992E-06	0.2823E-06	0.2000E-01	0.1000E-01	0.1000E-01	7
S368	8	16	27	-0.9375E+00	0.4072E-07	0.1000E-01	0.1000E-01	0.0000E+00	7
TORSION1	5476	10368	425	-0.4281E+00	0.3044E-06	0.5001E+02	0.4532E+02	0.4690E+01	7
TORSION2	5476	10368	425	-0.4281E+00	0.3044E-06	0.5010E+02	0.4525E+02	0.4850E+01	7
TORSION3	5476	10368	262	-0.1213E+01	0.2616E-06	0.3841E+02	0.3492E+02	0.3490E+01	7
TORSION4	5476	10368	262	-0.1213E+01	0.2616E-06	0.3832E+02	0.3444E+02	0.3880E+01	7
TORSION5	5476	10368	161	-0.2859E+01	0.5938E-06	0.2334E+02	0.2131E+02	0.2030E+01	7
TORSION6	5476	10368	161	-0.2859E+01	0.5938E-06	0.2336E+02	0.2135E+02	0.2010E+01	7
TORSIONA	5476	10368	405	-0.4162E+00	0.3910E-06	0.5316E+02	0.4765E+02	0.5510E+01	7
TORSIONB	5476	10368	405	-0.4162E+00	0.3910E-06	0.5325E+02	0.4775E+02	0.5500E+01	7
TORSIONC	5476	10368	224	-0.1201E+01	0.2821E-06	0.3551E+02	0.3202E+02	0.3490E+01	7
TORSIOND	5476	10368	224	-0.1201E+01	0.2821E-06	0.3534E+02	0.3190E+02	0.3440E+01	7
TORSIONE	5476	10368	159	-0.2846E+01	0.1787E-05	0.2347E+02	0.2101E+02	0.2460E+01	7
TORSIONF	5476	10368	159	-0.2846E+01	0.1787E-05	0.2347E+02	0.2113E+02	0.2340E+01	7

TAB. 3.15 – Résultats du code NOPTIQ utilisant les formules de BFGS à mémoire limitée avec $m_\ell = 17$.

nom	n	m	nf	f	test	tmain	tcode	tfunc	info
BDEXP	5000	5000	11	0.1734E-09	0.1401E-08	0.4400E+00	0.3200E+00	0.1200E+00	0
BDEXP	1000	1000	11	0.1553E-09	0.1253E-08	0.8000E-01	0.7000E-01	0.1000E-01	0
BQPGABIM	50	92	38	-0.3790E-04	0.8330E-05	0.7000E-01	0.7000E-01	0.0000E+00	7
BQPGASIM	50	100	41	-0.5520E-04	0.3599E-05	0.1100E+00	0.9000E-01	0.2000E-01	7
CVXBQP1	1000	2000	36	0.2252E+05	0.1170E-06	0.1820E+01	0.1800E+01	0.2000E-01	7
EXPLIN2	1200	2400	105	-0.7200E+08	0.1065E-02	0.9750E+01	0.9720E+01	0.3000E-01	7
EXPLIN	1200	2400	236	-0.7193E+08	0.2818E-02	0.1647E+02	0.1638E+02	0.9000E-01	7
JNLBRNG1	10000	9604	827	-0.1806E+00	0.7929E-06	0.8022E+03	0.7755E+03	0.2662E+02	7
JNLBRNG2	10000	9604	1979	-0.4149E+01	0.1268E-05	0.1434E+04	0.1382E+04	0.5142E+02	7
JNLBRNGA	10000	9604	3	0.7797E-13	0.1282E-02	0.1800E+00	0.1000E+00	0.8000E-01	7
JNLBRNGB	10000	9604	3	0.4857E-12	0.6410E-02	0.1800E+00	0.8000E-01	0.1000E+00	7
LINVERSE	1999	1000	255	0.6820E+03	0.1753E+01	0.4844E+02	0.4643E+02	0.2010E+01	7
LINVERSE	999	500	219	0.3404E+03	0.1881E+01	0.1856E+02	0.1783E+02	0.7300E+00	7
MCCORMCK	10000	20000	21	-0.9133E+04	0.7425E-07	0.5540E+01	0.5030E+01	0.5100E+00	7
MCCORMCK	5000	10000	22	-0.4567E+04	0.4969E-07	0.2560E+01	0.2340E+01	0.8000E+00	7
MCCORMCK	1000	2000	21	-0.9137E+03	0.1203E-06	0.3800E+00	0.3700E+00	0.1000E-01	7
NCVXBQP1	1000	2000	94	-0.1987E+09	0.3763E-09	0.6100E+00	0.5700E+00	0.4000E-01	7
NCVXBQP2	1000	2000	148	-0.1332E+09	0.9296E-05	0.2800E+01	0.2670E+01	0.1300E+00	7
NCVXBQP3	1000	2000	535	-0.6494E+08	0.7323E-02	0.2900E+02	0.2864E+02	0.3600E+00	7
NOBNDTOR	5476	10368	1025	-0.4499E+00	0.4012E-06	0.2741E+03	0.2630E+03	0.1113E+02	7
NONSCOMP	1000	2000	63	0.2443E-09	0.5396E-06	0.4250E+01	0.4200E+01	0.5000E-01	7
OBSTCLAE	5625	10658	410	0.1863E+01	0.2929E-06	0.1342E+03	0.1289E+03	0.5300E+01	7
OBSTCLAL	10000	19208	774	0.1886E+01	0.4559E-06	0.6141E+03	0.5944E+03	0.1975E+02	7
OBSTCLBL	10000	19208	364	0.7275E+01	0.3633E-06	0.1917E+03	0.1840E+03	0.7670E+01	7
OBSTCLBM	10000	19208	368	0.7275E+01	0.4232E-06	0.1905E+03	0.1829E+03	0.7580E+01	7
OBSTCLBU	10000	19208	364	0.7275E+01	0.3633E-06	0.1921E+03	0.1843E+03	0.7820E+01	7
PENTDI	5000	5000	3	0.4996E-04	0.3000E+01	0.7000E-01	0.2000E-01	0.5000E-01	7
PROBPENL	500	1000	6	0.3992E-06	0.2823E-06	0.1000E-01	0.1000E-01	0.0000E+00	7
S368	8	16	27	-0.9375E+00	0.3961E-07	0.1000E-01	0.1000E-01	0.0000E+00	7
TORSION1	5476	10368	349	-0.4281E+00	0.2064E-06	0.1010E+03	0.9678E+02	0.4220E+01	7
TORSION2	5476	10368	349	-0.4281E+00	0.2064E-06	0.1010E+03	0.9686E+02	0.4130E+01	7
TORSION3	5476	10368	233	-0.1213E+01	0.6036E-06	0.6940E+02	0.6652E+02	0.2880E+01	7
TORSION4	5476	10368	233	-0.1213E+01	0.6036E-06	0.6831E+02	0.6541E+02	0.2900E+01	7
TORSION5	5476	10368	158	-0.2859E+01	0.1308E-05	0.5130E+02	0.4902E+02	0.2280E+01	7
TORSION6	5476	10368	158	-0.2859E+01	0.1308E-05	0.5122E+02	0.4912E+02	0.2100E+01	7
TORSIONA	5476	10368	363	-0.4162E+00	0.2516E-06	0.1076E+03	0.1027E+03	0.4970E+01	7
TORSIONB	5476	10368	363	-0.4162E+00	0.2516E-06	0.1103E+03	0.1052E+03	0.5070E+01	7
TORSIONC	5476	10368	237	-0.1201E+01	0.4725E-06	0.7552E+02	0.7257E+02	0.2950E+01	7
TORSIOND	5476	10368	237	-0.1201E+01	0.4725E-06	0.7549E+02	0.7216E+02	0.3330E+01	7
TORSIONE	5476	10368	154	-0.2846E+01	0.6095E-06	0.5186E+02	0.4966E+02	0.2200E+01	7
TORSIONF	5476	10368	154	-0.2846E+01	0.6095E-06	0.5164E+02	0.4949E+02	0.2150E+01	7

TAB. 3.16 – Résultats du code NOPTIQ utilisant les formules de BFGS à mémoire limitée avec $m_\ell = 29$.

nom	n	m	nf	f	test	tmain	tcode	tfunc	info
3PK	30	30	10000	0.3769E + 01	0.2422E + 01	0.6400E + 00	0.5500E + 00	0.9000E - 01	5
ALLINIT	4	3	19	0.1671E + 02	0.1301E - 07	0.1000E - 01	0.1000E - 01	0.0000E + 00	7
BQPIVAR	1	2	11	0.2500E - 08	0.2243E - 15	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
BQPGABIM	50	92	48	-0.3790E - 04	0.8226E - 05	0.1000E - 01	0.1000E - 01	0.0000E + 00	7
BQPGASIM	50	100	41	-0.5520E - 04	0.1550E - 04	0.2000E - 01	0.2000E - 01	0.0000E + 00	7
CAMEL6	2	4	16	-0.1032E + 01	0.6323E - 08	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
CHEBYQAD	100	200	10000	0.1088E - 01	0.1807E - 02	0.1210E + 03	0.2910E + 01	0.1181E + 03	5
DECONVB	61	72	180	0.3240E - 10	0.4946E - 05	0.8000E - 01	0.4000E - 01	0.4000E - 01	7
DIPIGRI	7	4	37	0.6880E + 03	0.1111E + 03	0.1000E - 01	0.1000E - 01	0.0000E + 00	7
EG1	3	4	24	-0.1429E + 01	0.1534E - 06	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HART6	6	12	28	-0.3323E + 01	0.1449E - 05	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HATFLDA	4	4	55	0.5131E - 16	0.1583E - 07	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HATFLDB	4	5	36	0.5573E - 02	0.1107E - 09	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HATFLDC	25	48	56	0.9358E - 15	0.1200E - 06	0.1000E - 01	0.1000E - 01	0.0000E + 00	7
HIMMELP1	2	4	26	-0.5174E + 02	0.7271E - 08	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS100MOD	7	4	40	0.6787E + 03	0.1794E - 03	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS100	7	4	37	0.6806E + 03	0.4362E - 03	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS110	200	400	1	-0.1478E + 39	0.9990E + 00	0.1000E - 01	0.1000E - 01	0.0000E + 00	6
HS113	10	8	28	0.2431E + 02	0.1153E - 03	0.1000E - 01	0.1000E - 01	0.0000E + 00	7
HS117	15	20	77	0.3235E + 02	0.1165E - 05	0.2000E - 01	0.2000E - 01	0.0000E + 00	7
HS12	2	1	21	-0.3000E + 02	0.2870E - 06	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS1	2	1	49	0.1440E - 15	0.5357E - 06	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS24	2	5	10	-0.9979E + 00	0.1013E + 01	0.0000E + 00	0.0000E + 00	0.0000E + 00	4
HS25	3	6	1	0.3284E + 02	0.0000E + 00	0.0000E + 00	0.0000E + 00	0.0000E + 00	0
HS29	3	1	30	-0.2263E + 02	0.2622E - 06	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS2	2	1	22	0.4941E + 01	0.1148E - 07	0.1000E - 01	0.1000E - 01	0.0000E + 00	7
HS351	3	7	33	0.1111E + 00	0.3077E - 07	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS35MOD	3	3	44	0.2500E + 00	0.1907E - 07	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS35	3	4	31	0.1111E + 00	0.1998E - 07	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS36	3	7	17	-0.3300E + 04	0.9890E - 04	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS37	3	8	14	-0.3456E + 04	0.3075E - 03	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS38	4	8	198	0.9458E - 14	0.2967E - 05	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS3MOD	2	1	14	0.2500E - 08	0.5721E - 10	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS3	2	1	12	0.2500E - 08	0.1680E - 13	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS43	4	3	40	-0.4400E + 02	0.2865E - 06	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS44NEW	4	10	17	-0.1500E + 02	0.2587E - 05	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS45	5	10	16	0.1000E + 01	0.7834E - 05	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS4	2	2	12	0.2667E + 01	0.1351E - 09	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS57	2	3	10000	0.3065E - 01	0.1039E - 02	0.6400E + 00	0.2400E + 00	0.4000E + 00	5
HS5	2	4	17	-0.1913E + 01	0.3401E - 09	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS67	3	20	34	-0.1034E + 04	0.5010E + 01	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS70	4	9	599	0.7498E - 02	0.3089E - 08	0.2100E + 00	0.2000E - 01	0.1900E + 00	7
HS761	4	11	27	-0.1997E + 01	0.4500E + 01	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS76	4	7	27	-0.2004E + 01	0.4967E + 01	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
HS84	5	16	24	-0.5155E + 07	0.4424E + 07	0.1000E - 01	0.1000E - 01	0.0000E + 00	7
HS85	5	48	375	-0.1226E + 01	0.3278E + 00	0.6000E - 01	0.4000E - 01	0.2000E - 01	7
HS93	6	8	41	0.1365E + 03	0.1182E + 04	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
KSIP	20	1001	55	0.5758E + 00	0.4940E - 08	0.5139E + 03	0.5137E + 03	0.2500E + 00	0
LOGROS	2	2	156	0.0000E + 00	0.2221E - 07	0.1000E - 01	0.1000E - 01	0.0000E + 00	7
MAXLIKA	8	16	3022	0.1136E + 04	0.1930E - 03	0.1520E + 01	0.8000E - 01	0.1440E + 01	7
MDHOLE	2	1	71	0.2500E - 08	0.4786E - 08	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
MIFFLIN1	3	2	20	0.0000E + 00	0.3500E + 01	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
MINMAXBD	5	20	22	0.8256E + 03	0.6131E + 02	0.0000E + 00	0.0000E + 00	0.0000E + 00	7
MOSARQP1	900	930	23	-0.3801E + 03	0.3691E - 06	0.1620E + 01	0.8000E - 01	0.1540E + 01	7
MOSARQP2	900	930	27	-0.5098E + 03	0.1119E - 05	0.1730E + 01	0.9000E - 01	0.1640E + 01	7
OSLBQP	8	11	53	0.6250E + 01	0.3101E - 15	0.1000E - 01	0.1000E - 01	0.0000E + 00	7

TAB. 3.17 – Résultats du code NOPTIQ utilisant les formules de BFGS à mémoire limitée avec $m_\ell = 5$.

nom	n	m	nf	f	test	tmain	tcode	tfunc	info
PALMER1A	6	2	10000	0.8988E-01	0.3574E-02	0.1500E+00	0.6000E-01	0.9000E-01	5
PALMER1B	4	2	324	0.3447E+01	0.7455E-05	0.1000E-01	0.0000E+00	0.1000E-01	7
PALMER1E	8	1	10000	0.1227E+00	0.1049E+00	0.2300E+00	0.7000E-01	0.1600E+00	5
PALMER1	4	3	2710	0.1175E+05	0.9857E-03	0.2000E-01	0.1000E-01	0.1000E-01	7
PALMER2A	6	2	10000	0.1711E-01	0.5589E-02	0.1400E+00	0.9000E-01	0.5000E-01	5
PALMER2B	4	2	140	0.6233E+00	0.2608E-08	0.1000E-01	0.1000E-01	0.0000E+00	0
PALMER2E	8	1	10000	0.2129E-01	0.1820E+00	0.2000E+00	0.9000E-01	0.1100E+00	5
PALMER2	4	3	6593	0.3651E+04	0.3310E-03	0.1300E+00	0.1000E+00	0.3000E-01	7
PALMER3A	6	2	10000	0.2045E-01	0.3997E+00	0.1300E+00	0.2000E-01	0.1100E+00	5
PALMER3B	4	2	129	0.4228E+01	0.6675E-05	0.1000E-01	0.0000E+00	0.1000E-01	7
PALMER3E	8	1	10000	0.2954E-01	0.2556E-01	0.1400E+00	0.4000E-01	0.1000E+00	5
PALMER3	4	3	29	0.2398E+04	0.2526E+07	0.0000E+00	0.0000E+00	0.0000E+00	7
PALMER4A	6	2	10000	0.4061E-01	0.1029E+00	0.1400E+00	0.6000E-01	0.8000E-01	5
PALMER4B	4	2	109	0.6835E+01	0.1101E-05	0.1000E-01	0.0000E+00	0.1000E-01	7
PALMER4E	8	1	10000	0.6444E-01	0.1528E-02	0.1900E+00	0.7000E-01	0.1200E+00	5
PALMER4	4	3	47	0.2291E+04	0.2319E+06	0.0000E+00	0.0000E+00	0.0000E+00	4
PALMER5A	8	2	10000	0.1396E+00	0.1329E+02	0.1400E+00	0.6000E-01	0.8000E-01	5
PALMER5B	9	2	10000	0.5665E-01	0.6168E-01	0.1200E+00	0.1000E+00	0.2000E-01	5
PALMER5D	8	4	24	0.8734E+02	0.1271E-05	0.0000E+00	0.0000E+00	0.0000E+00	7
PALMER5E	8	1	10000	0.4633E-01	0.1370E+00	0.1600E+00	0.9000E-01	0.7000E-01	5
PALMER6A	6	2	10000	0.5596E-01	0.3297E+00	0.1100E+00	0.7000E-01	0.4000E-01	5
PALMER6E	8	1	10000	0.1826E-01	0.3758E-01	0.1500E+00	0.1100E+00	0.4000E-01	5
PALMER7A	6	2	10000	0.1067E+02	0.4354E+02	0.1100E+00	0.8000E-01	0.3000E-01	5
PALMER7E	8	1	10000	0.1018E+02	0.4737E+01	0.1300E+00	0.5000E-01	0.8000E-01	5
PALMER8A	6	2	10000	0.7401E-01	0.4220E-02	0.1100E+00	0.8000E-01	0.3000E-01	5
PALMER8E	8	1	10000	0.6551E-02	0.5393E+00	0.1400E+00	0.6000E-01	0.8000E-01	5
PROBPENL	500	1000	6	0.3992E-06	0.2823E-06	0.1000E-01	0.1000E-01	0.0000E+00	7
PSPDOC	4	1	27	0.2414E+01	0.2133E-08	0.0000E+00	0.0000E+00	0.0000E+00	7
S368	8	16	25	-0.9375E+00	0.1728E-07	0.0000E+00	0.0000E+00	0.0000E+00	7
SIM2BQP	2	2	11	0.2501E-08	0.4098E-15	0.1000E-01	0.1000E-01	0.0000E+00	7
SIMBQP	2	2	13	0.1000E-08	0.8122E-07	0.0000E+00	0.0000E+00	0.0000E+00	7
SIPOW1M	2	2000	5	0.2439E+00	0.2904E+03	0.2592E+03	0.2592E+03	0.9995E-02	4
SIPOW1	2	2000	5	0.2439E+00	0.2904E+03	0.2606E+03	0.2606E+03	0.2000E-01	4
SIPOW2M	2	2000	4	0.1485E+00	0.1049E+03	0.1956E+03	0.1956E+03	0.1000E-01	4
SIPOW2	2	2000	4	0.1485E+00	0.1049E+03	0.1953E+03	0.1953E+03	0.1000E-01	4
SIPOW3	4	2000	52	0.1200E+01	0.8099E+04	0.6672E+02	0.6668E+02	0.4001E-01	7
SIPOW4	4	2000	53	0.1200E+01	0.7990E+04	0.6666E+02	0.6664E+02	0.2000E-01	7
SPECAN	9	18	270	0.3439E-10	0.2249E-03	0.4980E+01	0.5000E-01	0.4930E+01	7
SPIRAL	3	2	10000	0.2232E+00	0.6362E-01	0.4100E+00	0.3500E+00	0.6000E-01	5
WEEDS	3	4	239	0.2587E+01	0.8887E-05	0.1000E-01	0.1000E-01	0.0000E+00	7
WOMFLET	3	3	10000	0.2839E+01	0.1013E+02	0.4500E+00	0.4000E+00	0.5000E-01	5
YFIT	3	1	645	0.6670E-12	0.2682E-05	0.2000E-01	0.1000E-01	0.1000E-01	7
ZY2	3	6	20	0.2000E+01	0.3846E-05	0.0000E+00	0.0000E+00	0.0000E+00	7

TAB. 3.18 – Résultats du code NOPTIQ utilisant les formules de BFGS à mémoire limitée avec $m_\ell = 5$.

Bibliographie

- [1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GRENNEBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV ET D. SORENSEN, *LAPACK's User Guide*, SIAM, Philadelphia, 1992.
- [2] P. ARMAND, *A quasi-Newton penalty barrier method for convex minimization problems*, Accepté pour publication dans Computational Optimization and Applications (2001).
- [3] P. ARMAND, J.-C. GILBERT ET S. JAN-JÉGOU, *A BFGS-IP algorithm for solving strongly convex optimization problems with feasibility enforced by an exact penalty approach*, Mathematical Programming 92 (2002), no. 3, Ser. B, 393–424.
- [4] P. ARMAND, J. C. GILBERT ET S. JAN-JÉGOU, *A feasible BFGS interior point algorithm for solving strongly convex minimization problems*, SIAM J. Optim. 11 (2000), no. 1, 199–222.
- [5] P. ARMAND ET P. SEGALAT *Dépôt du logiciel NOPTIQ à l'Agence pour la Protection des Programmes*, (2002).
- [6] I. BONGARTZ, A. R. CONN, N. GOULD ET PH. L. TOINT, *CUTE : Constrained and Unconstrained Testing Environment*, ACM Transactions on Mathematical Software 21 (1) (1995), pp. 123-160.
- [7] F. BONNANS, J.-C. GILBERT, C. LEMARÉCHAL ET C. SAGASTIZÁBAL, *Optimisation Numérique : Aspects théoriques et pratiques*, Mathématiques et Applications 27, Springer-Verlag, 1997.
- [8] C. G. BROYDEN, *The convergence of a class of double-rank minimization algorithms*, J. Inst. Math. Applic. 6 (1970), pp. 76-90.
- [9] R. H. BYRD, J.-C. GILBERT ET J. NOCEDAL, *A trust region method based on interior point techniques for nonlinear programming*, Mathematical Programming 89 (2000), pp. 149-185.
- [10] R. H. BYRD, M. E. HRIBAR ET J. NOCEDAL, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim. 9 (1999), pp. 877-900.
- [11] R. H. BYRD, P. LU, J. NOCEDAL ET C. ZHU, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16 (1995), pp. 1190-1208.
- [12] R. H. BYRD ET J. NOCEDAL, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 727–739.

- [13] R. H. BYRD, J. NOCEDAL ET R. B. SCHNABEL, *Representations of quasi-Newton matrices and their use in limited memory methods*, Mathematical Programming 63 (2, Ser. A) (1994), pp. 129-156.
- [14] R.H. BYRD, R.A. TAPIA ET Y. ZHANG, *An SQP augmented Lagrangian BFGS algorithm for constrained optimization*, SIAM J. Optim. vol 2. (1992), pp. 210-241.
- [15] A. R. CONN, N. I. M. GOULD ET P. L. TOINT, *LANCELOT : a FORTRAN package for large-scale nonlinear optimization (Release A)*, no 17 in Springer Series in Computational Mathematics, Springer-Verlag, New-York, 1992.
- [16] A. R. CONN, D. ORBAN ET PH. L. TOINT, *CUTEr (and SifDec), a Constrained and Unconstrained Testing Environment, revisited*, Cerfacs Technical report TR/PA/01/04 (2002).
- [17] G. B. DANTZIG, *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, 1963.
- [18] W. C. DAVIDON, *Variable metric methods for minimization*, A. E. C. Res. and Develop. Report ANL-5990 (1959), Argonne National Laboratory, Argonne, Illinois.
- [19] A. V. FIACCO ET G. P. MCCORMICK, *Nonlinear programming : Sequential unconstrained minimization techniques*, John Wiley and Sons, Inc., New York-London-Sydney, 1968.
- [20] R. FLETCHER, *A new approach to variable metric algorithms*, Computer Journal 13 (1970), pp. 317-322.
- [21] R. FLETCHER, *Practicals Methods of Optimization* , John Wiley & Sons, New York (1981), pp. 55-56.
- [22] R. FLETCHER ET M. J. D. POWELL, *A rapidly convergent descent method for minimization*, Computer Journal 7 (1963), pp. 163-168.
- [23] K. R. FRISCH, *The logarithmic potential method of convex programming*, Memorandum of May 13 (1955), University Institute of Economics, Oslo, Norway.
- [24] D.M. GAY, M.L. OVERTON ET M.H. WRIGHT, *A primal-dual interior method for nonconvex nonlinear programming*, Advances in nonlinear programming (Beijing, 1996), 31-56, Appl. Optim. 14, Kluwer Acad. Publ., Dordrecht, 1998.
- [25] J.-C. GILBERT ET C. LEMARÉCHAL, *Some Numerical Experiments with variable-storage quasi-Newton Algorithms*, Mathematical Programming 45 (1989), pp. 407-435.
- [26] P. E. GILL, W. MURRAY, M. A. SAUNDERS ET M. H. WRIGHT, *SNOPT : an SQP algorithmn for large-scale constrained optimization*, Numerical Analysis Report 97-2, Department of Mathematics, University of California, San Diegp, La Jolla, CA, 1997.
- [27] P.E. GILL, W. MURRAY ET M.H. WRIGHT, *Practical Optimization*, Academic Press, London and New-York (1981), pp. 41-43.

- [28] D. GOLDFARB, *A family of variable metric methods derived by variational means*, Mathematics of Computation 24 (1970), pp. 31-40.
- [29] J. GONDZIO, *Multiple centrality corrections in a primal-dual method for linear programming*, Computational Optimization and Applications 6 (1996), pp. 137-156.
- [30] J. GONDZIO ET R. KOUWENBERG, *High Performance Computing for Asset Liability Management*, Operations Research 49 no 6 (2001), pp. 879-891.
- [31] A. GRIEWANK ET P. L. TOINT, *Local convergence analysis of partitioned quasi-Newton updates*, Numerische Mathematik 39 (1982), pp. 429-448.
- [32] A. GRIEWANK ET P. L. TOINT, *Partitioned variable metric updates for large structured optimization problems*, Numerische Mathematik 39 (1982), pp. 119-137.
- [33] J.-B. HIRIART-URRUTY ET C. LEMARÉCHAL, *Convex analysis and minimization algorithms. I*, Springer-Verlag, Berlin, 1993. Fundamentals.
- [34] N. KARMAKAR, *A new polynomial-time algorithm for linear programming*, Combinatorics 4 (1984), pp. 373-395.
- [35] L. G. KHACHIAN, *A polynomial algorithm in linear programming*, Soviet Mathematics Doklady, 20 (1979), pp. 191-194.
- [36] V. KLEE ET G. J. MINTY, *How good is the simplex algorithm ?*, Inequalities, O. Shisha, ed, Academic Press, New York (1972), pp. 159-175.
- [37] L. MCLINDEN *An analogue of Moreau's proximation theorem, with application to the nonlinear complementary problem*, Pacific J. Math., 88 (1980), pp. 101-161.
- [38] D. C. LIU ET J. NOCEDAL, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming 45 (1989), pp. 503-528.
- [39] F. A. LOOTSMA, *Hessian matrices of penalty functions for solving constrained optimization problems*, Philips Res. Repts. 24 (1969), pp. 322-331.
- [40] F. A. LOOTSMA, *Boundary properties of penalty functions for constrained optimization problems*, Philips Res. Repts. Suppl. 3 (1970).
- [41] F. A. LOOTSMA, *A survey of methods for solving constrained optimization problems via unconstrained optimization*, in Numerical Methods for Non-Linear Optimization (F. A. Lootsma, ed.) Academic Press, London and New York (1972).
- [42] I. J. LUSTIG, R. E. MARSLÉN ET D. F. SHANNO, *Interior Point for Linear Programming : Computational State of Art*, ORSA Journal on Computing, vol 6 (1994), pp. 1-14.
- [43] M. MARAZZI, *A note on the use of predictor-corrector techniques in Nonlinear Optimization*, Presented at Nonlinear Optimization and Applications, Erice, Italy, June-July 1998.
- [44] J. L. MORALES, *A numerical study of limited memory BFGS methods*, in Appl. Math. Lett., vol 15 no 4 (2002), pp. 481-487.
- [45] W. MURRAY *An algorithm for constrained minimization*, in Optimization (R. Fletcher, ed), Academic Press London and New York (1969), pp. 247-258.

- [46] W. MURRAY *Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions*, J. Optim. Theory Appl. 7 (1971), pp. 189-196.
- [47] W. MURRAY ET M. H. WRIGHT, *Line Search Procedures for the Logarithmic Barrier Function*, SIAM J. Optim. 4 (1994), pp. 229-246.
- [48] S.G. NASH ET J. NOCEDAL, *A numerical study of the limited memory BFGS method and the truncated Newton method for large scale optimization*, SIAM J. Optim. 1 (1991), pp. 358-372.
- [49] J. NOCEDAL, *Updating quasi-Newton matrices with limited storage*, Mathematical Programming 35 (1980), pp. 773-782.
- [50] J. NOCEDAL ET S. J. WRIGHT, *Numerical optimization*, Springer-Verlag, New York, 1999.
- [51] S. OREN ET E. SPEDICATO, *Optimal conditioning of self-scaling variable metric algorithms*, Mathematical Programming 10 (1976), pp. 70-90.
- [52] J. M. ORTEGA ET W. C. RHEINBOLDT, *Iterative solution of nonlinear equations in several variables*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1970 original.
- [53] J. D. PEARSON, *Variable metric methods of minimisation*, Computer Journal 12 (1969/1970), pp. 171-178.
- [54] F. POTRA, *An infeasible interior point predictor corrector algorithm for linear programming*, SIAM J. Optim. 6 (1996), pp. 19-32.
- [55] M. J. D. POWELL, *On the convergence of the variable metric algorithm*, J. Inst. Math. Appl., 7 (1971), pp. 21-36.
- [56] M. J. D. POWELL, *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis Dundee 1977, G.A. Watson, ed., Springer-Verlag, Berlin (1977), pp. 144-157.
- [57] D. F. SHANNO, *Conditioning of quasi-Newton methods for function minimization*, Mathematics of Computation 24 (1970), pp. 647-657.
- [58] D. F. SHANNO ET R. J. VANDERBEL, *An Interior-Point Algorithm for Nonconvex Nonlinear Programming*, Computational Optimization and Applications 13 (1999), pp. 231-252.
- [59] R. TAPIA, Y. ZHANG, M. SALTZMAN ET A. WEISER, *The Mehrotra predictor-corrector interior point method as a perturbed composite Newton method*, SIAM J. Optim. 6 (1996), pp. 47-56.
- [60] M. H. WRIGHT, *Interior methods for constrained optimization*, Acta Numerica 1991, Cambridge University Press, pp. 341-407.
- [61] S. J. WRIGHT, *Primal-Dual Interior-Point Methods*, SIAM Publications, Philadelphia, Pa (1997).

Résumé : Cette thèse s'intéresse à des méthodes de points intérieurs et de quasi-Newton en optimisation non linéaire et à leurs mises en œuvre. On présente le code NOPTIQ utilisant les formules de BFGS à mémoire limitée pour résoudre des problèmes de grande taille. L'originalité de cette approche est l'emploi de ces formules dans le cadre des méthodes de points intérieurs. L'espace mémoire et le coût en opérations du calcul d'une itération sont alors faibles. Le code NOPTIQ est robuste et a des performances comparables avec les codes de références 1-BFGS-B et LANCELOT. On présente aussi un algorithme non réalisable utilisant les méthodes précédentes pour résoudre un problème non linéaire avec contraintes d'inégalité et contraintes d'égalité linéaire. L'idée est de pénaliser le problème à l'aide de variables de décalage et d'une variante de la méthode *big-M*. La convergence q-superlinéaire des itérés internes et la convergence globale des itérés externes sont démontrées.

Mots-clés : Optimisation numérique, optimisation avec contraintes, algorithmes de points intérieurs, quasi-Newton, recherche linéaire, problèmes de grande taille, convergence superlinéaire, fortran.

Interior point and quasi-Newton methods.

Abstract : This thesis is interested in interior point and quasi-Newton methods in nonlinear optimization and with their implementation. One presents the code NOPTIQ using the limited memory BFGS formulas to solve large scale problems. The originality of this approach is the use of these formulas within the framework of interior point methods. The storage requirement and the computing cost of one iteration are then low. One shows that NOPTIQ is robust and that its performance are comparable with the reference codes 1-BFGS-B and LANCELOT. One presents also an infeasible algorithm using the preceding methods to solve a nonlinear problem with inequality constraints and linear equality constraints. The idea is to penalize the problem using shift variables and a variant of the *big-M* method of linear programming. The q-superlinear convergence of the inner iterates and the global convergence of outer iterates are shown.

Keywords : Numerical optimization, constrained optimization, interior point algorithms, quasi-Newton, linesearch, large scale problems, superlinear convergence, fortran.

Discipline : Mathématiques et ses applications

Laboratoire : LACO, Faculté des Sciences, 123 avenue Albert Thomas, 87060 Limoges Cedex, France.
