

INP TOULOUSE - ENSEEIHT

OPTIMISATION NUMÉRIQUE

MÉTHODES NUMÉRIQUES POUR LES PROBLÈMES D'OPTIMISATION

Rapport de Projet

Auteur :
Kevin ECCLI

3 février 2015



Table des matières

1	Algorithmes de Descente	2
1.1	Fonctions de tests	2
1.2	Méthode de Newton	2
1.2.1	Critères d'arrêts	2
1.2.2	Résultats	2
1.2.3	Interprétations	3
1.3	Méthode de Quasi-Newton	3
1.3.1	Résultats	3
1.3.2	Interprétations	3
1.4	Comparaisons	3
2	Recherche Linéaire	4
2.1	Backtracking	4
2.1.1	Résultats	4
2.1.2	Interprétations	7
2.2	Bisection	8
2.2.1	Résultats	8
2.2.2	Interprétations	8
2.3	Interpolation	9
2.3.1	Résultats	9
2.3.2	Interprétations	9
2.4	Approche-Finition	10
2.4.1	Résultats	10
2.4.2	Interprétations	10
3	Synthèse	11
3.1	Notre point de référence : FMINUNC	11
3.2	Résultats	11
3.3	Interprétations	11
4	Le Lagrangien Augmenté	12
4.1	Critères de convergence	12
4.2	Problèmes	12
4.3	Résultats	12
5	Bilan	13

Chapitre 1

Algorithmes de Descente

1.1 Fonctions de tests

Les fonctions de tests sont les suivantes :

$$f1 = 2(x1 + x2 - 2)^2 + (x1 - x2)^2$$

$$f2 = 100(x2 - x1^2)^2 + (1 - x1)^2$$

Minimas

1.2 Méthode de Newton

1.2.1 Critères d'arrêts

Les critères d'arrêt utilisés sont les suivants :

- Nombre d'itérations dépassant une valeur fixée
- Variation relative de la position du minimum inférieure à epsilon fixée

1.2.2 Résultats

Départs	Temps	Flag	Nbr Appel Fct	Nbr Appel Grad	Nbr Appel Hessien
0;0	0,000355	1	1	2	2
10;10	0,000198	1	1	2	2
0;1	0,000249	1	1	2	2
1;0	0,000219	1	1	2	2
1;3	0,000201	1	1	2	2
3;1	0,000209	1	1	2	2

FIGURE 1.1 – Fonction f1 :Résultats en fonction du point de départ pour Newton Locale

Départs	Temps	Flag	Nbr Appel Fct	Nbr Appel Grad	Nbr Appel Hessien
0;0	0,0747	1	1	3	3
10;10	0,00111	2	1	5	5
0;1	0,000853	1	1	6	6
1;0	0,000373	1	1	2	2
1;3	0,000449	1	1	2	2
3;1	0,000711	2	1	5	5

FIGURE 1.2 – Fonction f2 :Résultats en fonction du point de départ pour Newton Locale

1.2.3 Interprétations

On remarque que quelque soit le point de départ, l'algorithme est plus lent pour f2. Ce qui est notamment du aux plus nombreux appels aux gradients et hessiens. On remarque dans le cas de f2 que la recherche est d'autant plus lente que l'on démarre d'un point loin du minimum, ceci est moins vrai pour f1. En effet f1 étant une fonction pouvant etre mise sous la forme d'un paraboloïde, la droite engendrée par le gradient en tout point de la courbe intersecte le point minimal.

L'erreur relative est dans tous les cas sous le seuil de précision machine.

1.3 Méthode de Quasi-Newton

1.3.1 Résultats

Départs	Temps	Flag	Nbr Fct	Nbr Grad	Nbr Hessien	Erreur relative
0;0	0,0145	2	1	9	1	$\leq \epsilon$
10;10	0,00619	2	1	9	1	$\leq \epsilon$
0;1	0,00318	2	1	18	1	$\leq \epsilon$
1;0	0,000618	2	1	18	1	$\leq \epsilon$
1;3	0,000612	2	1	21	1	$\leq \epsilon$
3;1	0,000596	2	1	21	1	$\leq \epsilon$

FIGURE 1.3 – Fonction f1 :Résultats en fonction du point de départ pour Quasi-Newton

Départs	Temps	Flag	Nbr Fct	Nbr Grad	Nbr Hessien	Erreur relative
0;0	0,00607	2	1	21	1	27.1353
10;10	0,000897	2	1	24	1	4.4986e+15
0;1	0,00100	2	1	27	1	4.5785e+08
1;0	0,000634	2	1	21	1	7.2934e-04
1;3	0,000949	2	1	27	1	4.3617
3;1	0,000952	2	1	24	1	1.0776e+04

FIGURE 1.4 – Fonction f2 :Résultats en fonction du point de départ pour Quasi-Newton

1.3.2 Interprétations

Dans le cas de f1, l'erreur relative est sous le seuil de la précision machine, cependant pour f2 l'algorithme est beaucoup moins précis, ceci étant surement du au fait que pour l'instant aucune méthode de recherche de pas optimale n'a était mise en place. On remarque en effet un nombre important d'appel du gradient du à l'estimation de la hessienne à chaque itération. Le nombre d'itération augmentant, la hessienne est de moins en moins précise.

1.4 Comparaisons

La méthode de QuasiNewton se révèle plus longue dans le cas d'une fonction simple tel que f1 et inutilisable pour f2. Les gains de performance pour la méthode de Quasi-Newton ne sont pas au rendez-vous. La méthode de QuasiNewton semble ne pas pouvoir se passer d'une recherche de pas optimale au contraire de la méthode de NewtonLocale.

Chapitre 2

Recherche Linéaire

2.1 Backtracking

2.1.1 Résultats

Vitesse en fonction du point de départ et du paramètre de contraction

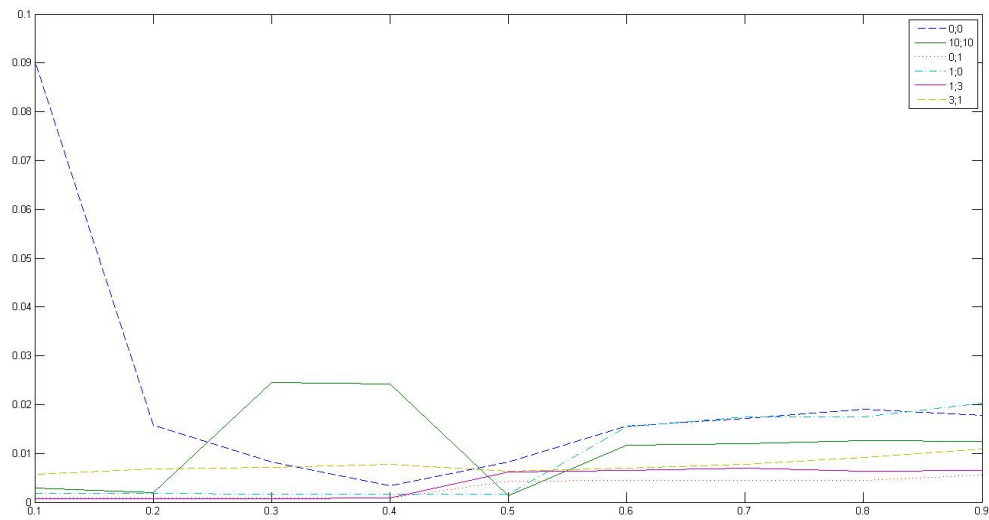


FIGURE 2.1 – Temps d'exécution pour les différents points de départ suivant le paramètre de contraction ($c_1=0.1$ Newton-Local f_2)

Nombre d'appel en fonction du paramètre de contraction

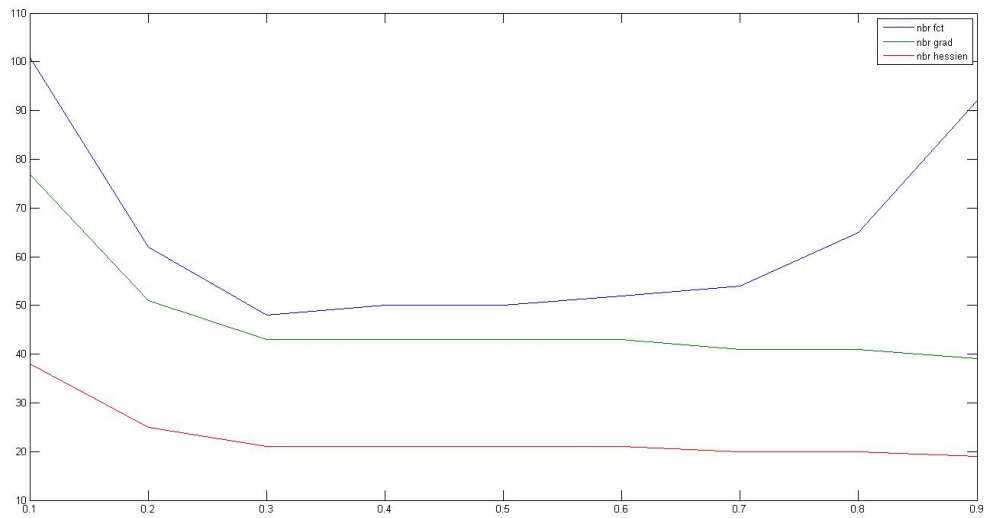


FIGURE 2.2 – Nombre d'appel de la fonction, gradient et hessienne en fonction du parametre de contraction($x_0=3;1$ f2 Newton Locale)

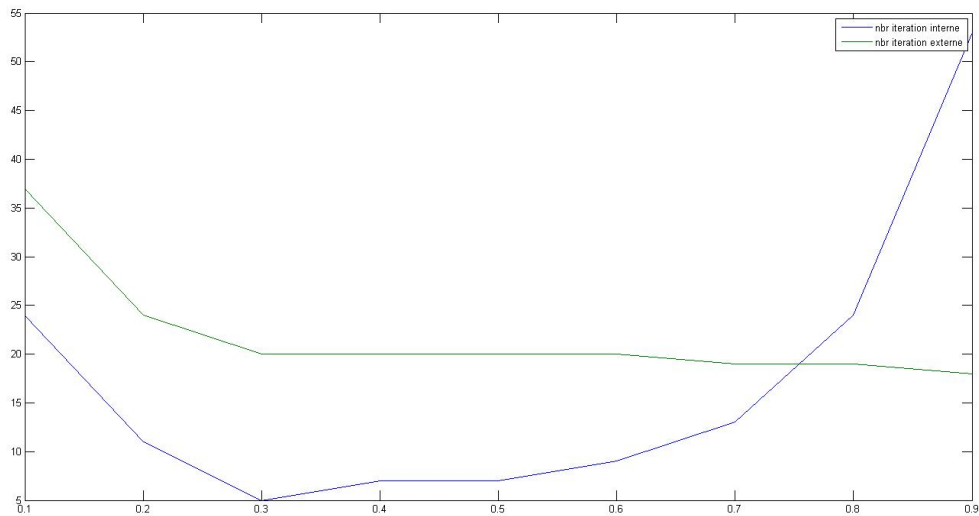


FIGURE 2.3 – Nombre d'itérations interne et externes en fonction du paramètre de contraction($x_0=3;1$ f2 Newton Locale)

Vitesse et erreur relative en fonction du paramètre de contraction

Pour la fonction f2 avec quasiNewton du point de départ $3;1$ $c_1 = 0.1$. Variation de la vitesse et de l'erreur relative en fonction de ρ :

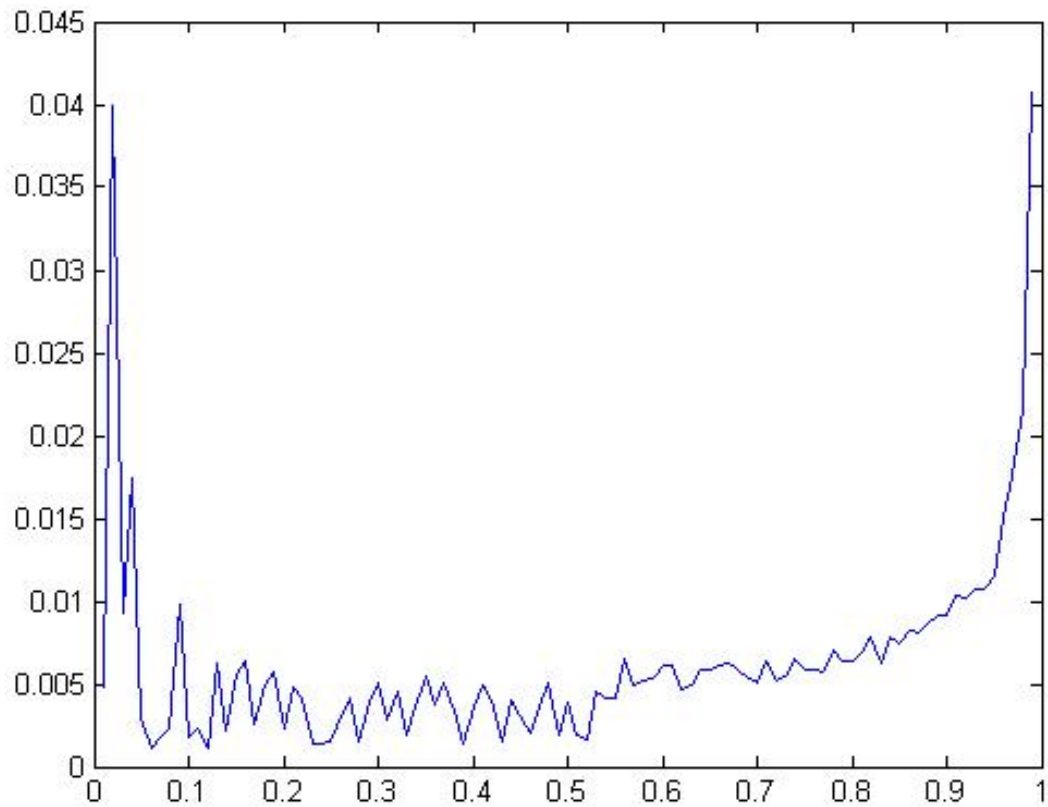


FIGURE 2.4 – Variation de la vitesse en fonction de rho

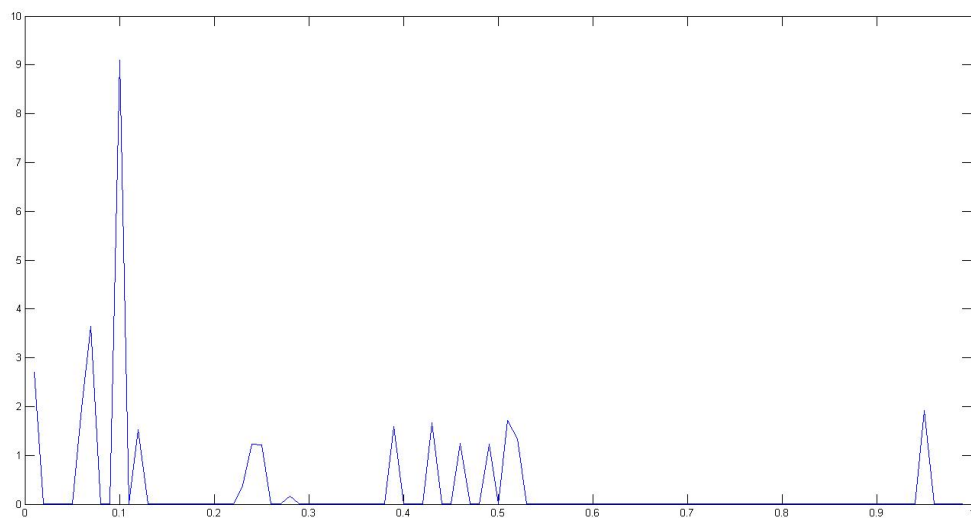


FIGURE 2.5 – Erreur relative en fct de rho

Vitesse et erreur relative en fonction du critère de Wolf1

Pour la fonction f2 avec quasiNewton du point de départ 3;1 rho=0.85 Variation de la vitesse en fonction de c1 :

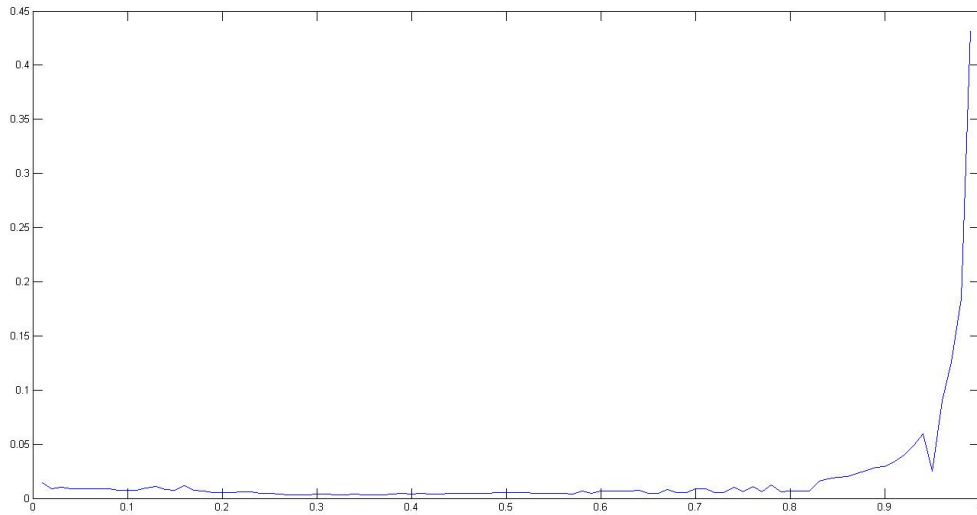


FIGURE 2.6 – Temps d'exécution en fonction de $c1$

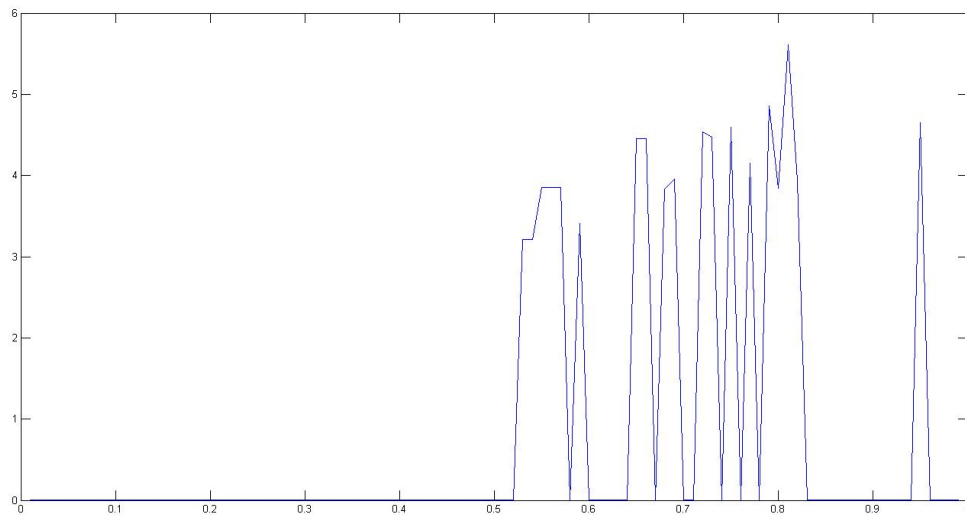


FIGURE 2.7 – Erreur relative en fonction de $C1$

2.1.2 Interprétations

On remarque qu'un paramètre de Wolf élevé équivalent à l'absence de critère de Wolf est responsable d'une augmentation du temps de recherche. A contrario un critère proche de 0 signifiant une exigence importante au niveau de la décroissance est responsable d'une erreur plus élevée. En effet le pas cessera d'augmenter lorsqu'il est proche du minimum(car la décroissance n'est plus assez importante pour satisfaire $Wolf1$) et l'approchera donc plus grossièrement.

On remarque qu'un paramètre de contraction proche de 1 est corrélé à une vitesse faible ainsi qu'à une erreur relative faible. A contrario un paramètre de contraction proche de 0 implique une vitesse plus élevée mais un risque d'erreur plus élevé. En effet un paramètre proche de 1 permet un plus grand nombre d'approche successive du pas optimale en progressant petit à petit ce qui limite le risque de "dépasser" le pas optimale. Un paramètre proche de 0 augmente le risque de tirer trop loin et par conséquent l'erreur générée devra être compensé par un plus grand nombre d'itération externe.

2.2 Bisection

2.2.1 Résultats

Influence de $c1$ et $c2$ sur la vitesse et l'erreur relative

$c1$ $c2$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1	0	0,303	0,00272	0,00310	0,00399	0,00307	0,00257	0,00454	0,003253	0,00308
0.2	0	0	0,00267	0,00322	0,00349	0,003583	0,00297	0,00279	0,00325	0,00263
0.3	0	0	0	0,00497	0,00267	0,00324	0,002840	0,00315	0,00388	0,00427
0.4	0	0	0	0	0,00258	0,00332	0,00287	0,00527	0,003253	0,00402
0.5	0	0	0	0	0	0,00357	0,00465	0,00326	0,00357	0,00303
0.6	0	0	0	0	0	0	0,00614	0	0,00823	0,0064190
0.7	0	0	0	0	0	0	0	0,00801	0,00635	0,008306
0.8	0	0	0	0	0	0	0	0	0,0150	0,01473
0.9	0	0	0	0	0	0	0	0	0	0,0356

FIGURE 2.8 – Temps de recherche en fonction de couples de critères de Wolf

	1	2	3	4	5	6	7	8	9	10
1	0	2.7136	0.8296	1.3643e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13
2	0	0	0.8296	1.3643e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13
3	0	0	0	1.3643e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13
4	0	0	0	0	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13	9.1539e-13
5	0	0	0	0	0	1.6312e-10	1.6312e-10	1.6312e-10	1.6312e-10	1.6312e-10
6	0	0	0	0	0	0	7.8886e-07	0	7.8886e-07	7.8886e-07
7	0	0	0	0	0	0	0	6.0741e-07	6.0741e-07	6.0741e-07
8	0	0	0	0	0	0	0	0	2.7688e-06	2.7688e-06
9	0	0	0	0	0	0	0	0	0	6.7804e-06

FIGURE 2.9 – Erreur relative en fonction de $C1$ ([0.1 :0.9] verticale) et $C2$ ([0.1 :1] horizontale)

2.2.2 Interprétations

On remarque que le couple $c1 = 0.1$, $c2 = 0.7$ est responsable de la vitesse d'exécution la plus rapide.

L'erreur relative est dans la majorité des cas de l'ordre de 10^{-13} sauf si le paramètre $c2$ est trop faible (≤ 0.4). En effet le paramètre $c2$ traduit le fait que le gradient se soit suffisamment rapproché de 0 et 0.4 est visiblement trop exigeant pour s'approcher suffisamment du minimum.

2.3 Interpolation

2.3.1 Résultats

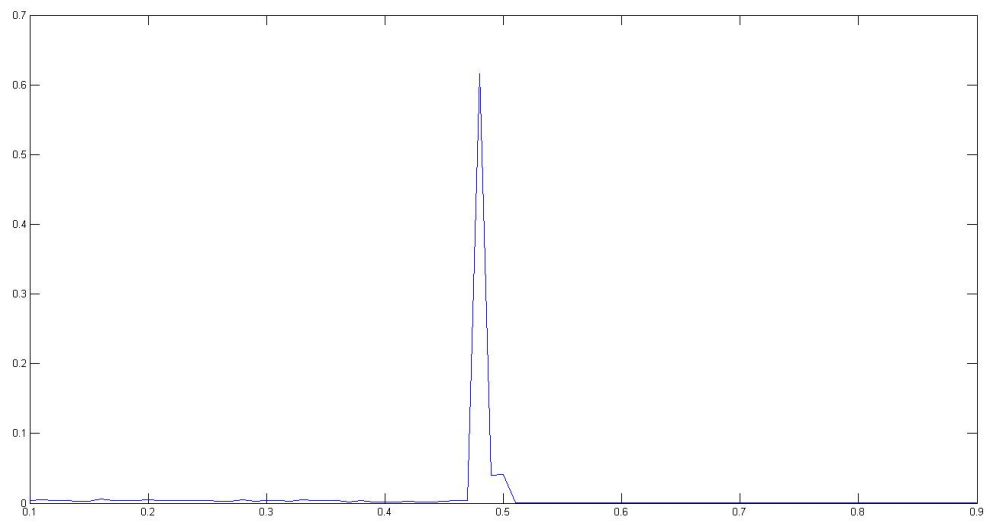


FIGURE 2.10 – Temps en fonction de C1

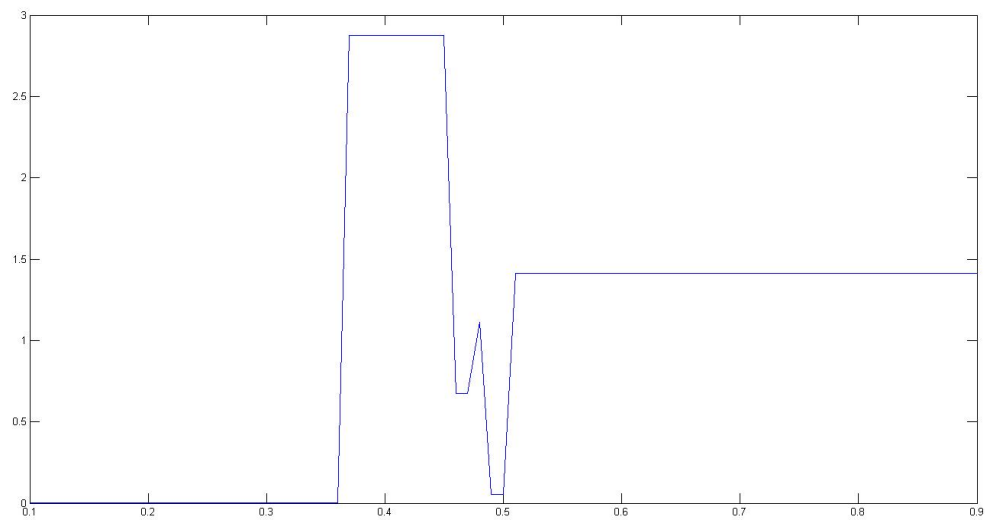


FIGURE 2.11 – Erreur relative en fonction de C1

2.3.2 Interprétations

On remarque un temps d'exécution dans l'ensemble faible et uniforme, cependant aux alentours de $c1 = 0.5$, le temps d'exécution atteint un pic.

L'erreur relative est quand à elle faible pour $c1=0.3$, au delà l'erreur relative décolle.

2.4 Approche-Finition

2.4.1 Résultats

	1	2	3	4	5	6	7	8	9	10
1	0	6.0314	1.4142	5.8295	1.4142	5.7140	5.8356	5.9058	8.4741e-06	8.5718e-06
2	0	0	1.4142	5.9431	5.9802	1.4142	5.9084	5.9724	6.5359	8.5718e-06
3	0	0	0	1.4142	5.9756	5.2904	5.8601	5.9140	6.5663	8.5718e-06
4	0	0	0	0	4.1983e-07	5.4965	6.0216	6.0716	6.5203	8.5718e-06
5	0	0	0	0	0	1.8452e-06	1.8777e-08	4.7484e-08	7.7858e-08	8.5718e-06
6	0	0	0	0	0	0	1.2110e-06	0	1.3521e-06	8.5718e-06
7	0	0	0	0	0	0	0	5.6690e-07	6.3809e-06	8.5718e-06
8	0	0	0	0	0	0	0	0	1.4142	8.5718e-06
9	0	0	0	0	0	0	0	0	0	8.5718e-06

FIGURE 2.12 – Temps d'exécution en fonction de $C1([0.1 : 0.9]$ verticale) et $C2([0.1 : 1]$ horizontale)

	1	2	3	4	5	6	7	8	9	10
1	0	0.6338	0.5662	0.5577	0.5626	0.5611	0.5538	0.5622	0.0126	0.0290
2	0	0	0.5599	0.5647	0.5579	0.5535	0.5556	0.5580	0.5653	0.0293
3	0	0	0	0.5624	0.5666	0.5647	0.5585	0.5626	0.5646	0.0296
4	0	0	0	0	0.0140	0.5694	0.5662	0.5655	0.5623	0.0348
5	0	0	0	0	0	0.0165	0.0185	0.0111	0.0149	0.0301
6	0	0	0	0	0	0	0.0201	0	0.0189	0.0301
7	0	0	0	0	0	0	0	0.0250	0.0233	0.0289
8	0	0	0	0	0	0	0	0	0.5601	0.0285
9	0	0	0	0	0	0	0	0	0	0.0294

FIGURE 2.13 – Erreur relative en fonction de $C1([0.1 : 0.9]$ verticale) et $C2([0.1 : 1]$ horizontale)

2.4.2 Interprétations

On remarque que globalement l'erreur relative est assez élevée. Cependant le choix de $c1=0.5$ et $c2=0.8$ permet d'obtenir une vitesse d'exécution rapide avec une erreur relative faible.

D'autres valeurs pour les critères de Wolf semblent aussi intéressantes : $(0.5 ; 0.7)$, $(0.1 ; 0.9)$ ou $(0.6 ; 0.7)$.

Chapitre 3

Synthèse

3.1 Notre point de référence : FMINUNC

On utilise la fonction inclut dans la toolbox optimisation de Matlab avec des paramètres de précision epsilon similaire ainsi qu'un nombre d'itération maximale similaire. De plus on utilise le même algorithme de Newton via les options pour avoir des résultats comparables.

3.2 Résultats

Le test est fait avec la fonction la plus complexe f2 et le point de départ (3;1).

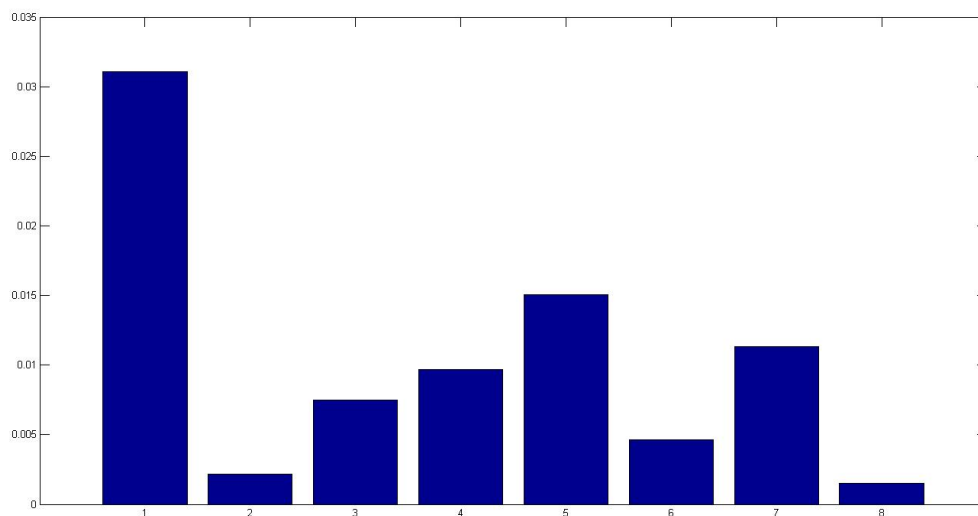


FIGURE 3.1 – Les temmps d'exécution des différentes méthodes (1 :Fminunc 2 :Newtonlocale 3 :NewtonLocale Bactracking 4 :Newton Locale Bissection 5 :NewtonLocale Interpolation 6 :QuasiNewton 7 :QuasiNewton Bactracking 8 :QuasiNewton Bissection 9 :QuasiNEWTON Interpolation)

3.3 Interprétations

On remarque que l'algorithme le plus rapide est le QuasiNewton avec recherche linéaire Bissection. Bizarrement introduire une méthode de recherche de pas optimale augmente le temps de recherche. Globalement tous les algorithmes codés sont plus rapide que la fonction fminunc.

Tous les algorithmes ont vu leurs paramètres optimisés pour générer une erreur relative sous le seuil de précision machine.

Chapitre 4

Le Lagrangien Augmenté

4.1 Critères de convergence

J'ai récupéré les conditions d'optimalités KKT les plus simples qui sont :

- $\mu(k)_i = 0$
- $\mu(k)^* \text{norm}(c(x), 2) = 0$

4.2 Problèmes

J'ai rencontré des problèmes pour créer dynamiquement la fonction du Lagrangien augmenté et ensuite la passer à `fminunc`. En conséquence je n'ai pas pu réaliser de test de performance sur l'algorithme de Lagrangien augmenté. Cependant un script de test a tout de même été mis en place ainsi que la totalité du code du Lagrangien augmenté bien qu'il ne soit pas fonctionnel.

4.3 Résultats

Pas de résultats à présenter du fait de la non fonctionnalité de l'algorithme (Voir ci dessus).

Chapitre 5

Bilan

Ce projet m'a permis d'améliorer mes connaissances en Matlab, notamment dans l'utilisation de la toolbox d'optimisation. D'autre part j'ai pu acquérir une conscience plus grande des problèmes de complexité de ces algorithmes d'optimisation et de leurs contraintes numériques.