Projet d'Optimisation Numérique :
Méthodes numériques pour les problèmes d'optimisation

Grégoire Martini 2ING

Mardi 9 Février 2016

Table des matières

1 Optimisation sans contraintes

1.1 Algorithme de Newton local

L'algorithme implémenté converge en une seule itération pour la fonction de test f1 car le gradient de celle-ci est linéaire et donc sa hessienne est constante. Ainsi le calcul de la première direction de Newton amène directement au point de gradient nul qui est unique (f1 strictement convexe car sa hessienne est définie positive).

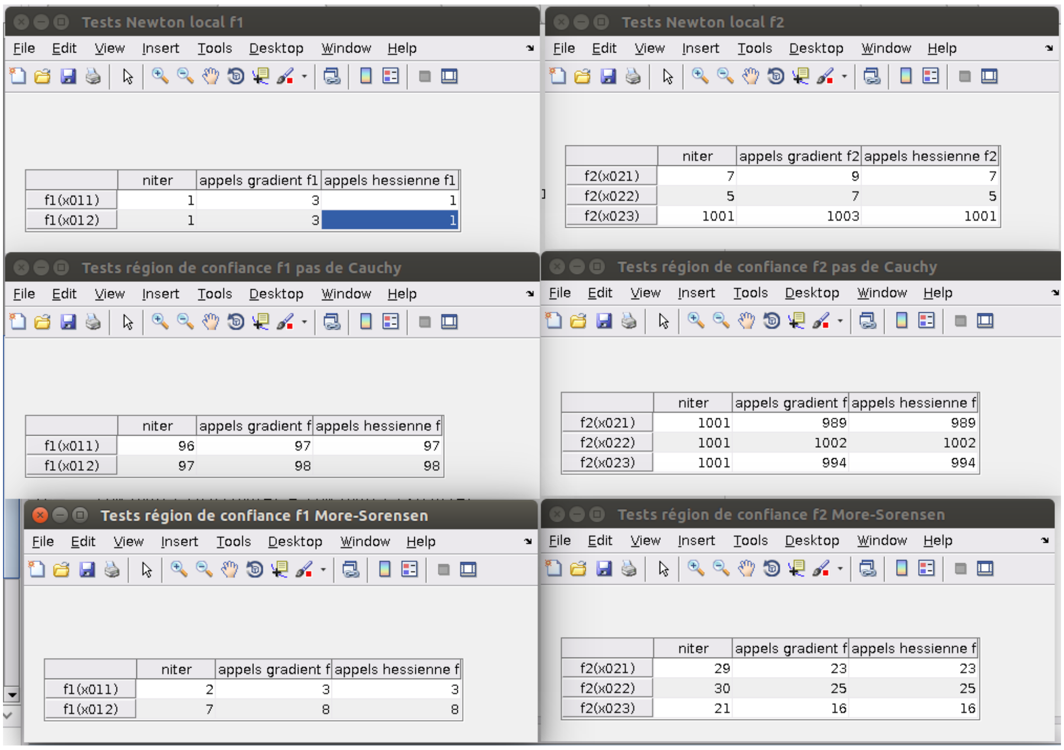
L'algorithme peut en revanche ne pas converger pour la fonction de test f2 car la hessienne de celle-ci peut ,en fonction du point où elle est évaluée, ne pas être définie positive. La méthode utilisée ici n'est alors pas bien définie et l'algorithme peut diverger.

1.2 Régions de confiance

Le modèle de Taylor de la fonction f1 à l'ordre 2 est égale à la fonction car celle-ci est une quadratique. Le pas de Cauchy est une méthode qui approche la fonction par un modèle quadratique qui est donc ici égale à la fonction. On peut donc comparer l'algorithme de Newton local à celui de la région de confiance avec le pas de Cauchy.

On remarque ainsi que la région de confiance avec le pas de Cauchy ne converge pas aussi souvent que Newton local (notamment pour f2) et nécessite dans tout les cas plus d'itérations pour arriver à la solution (notamment pour f1). Néanmoins, la région de confiance avec Moré-Sorensen permet de converger pour f2(x023) alors que les autres algorithmes ne le permettent pas.

L'avantage de More-Sorensen est d'être une méthode exacte alors que le pas de Cauchy est une méthode approchée qui utilise un modèle quadratique. L'inconvénient est la complexité de l'algorithme et donc du calcul alors que le pas de Cauchy est très simple.



Tests Newton local f1

	niter	appels gradient f1	appels hessienne f1
f1(x011)	1	3	1
f1(x012)	1	3	1

Tests Newton local f2

	niter	appels gradient f2	appels hessienne f2
f2(x021)	7	9	7
f2(x022)	5	7	5
f2(x023)	1001	1003	1001

Tests région de confiance f1 pas de Cauchy

	niter	appels gradient f	appels hessienne f
f1(x011)	96	97	97
f1(x012)	97	98	98

Tests région de confiance f2 pas de Cauchy

	niter	appels gradient f	appels hessienne f
f2(x021)	1001	989	989
f2(x022)	1001	1002	1002
f2(x023)	1001	994	994

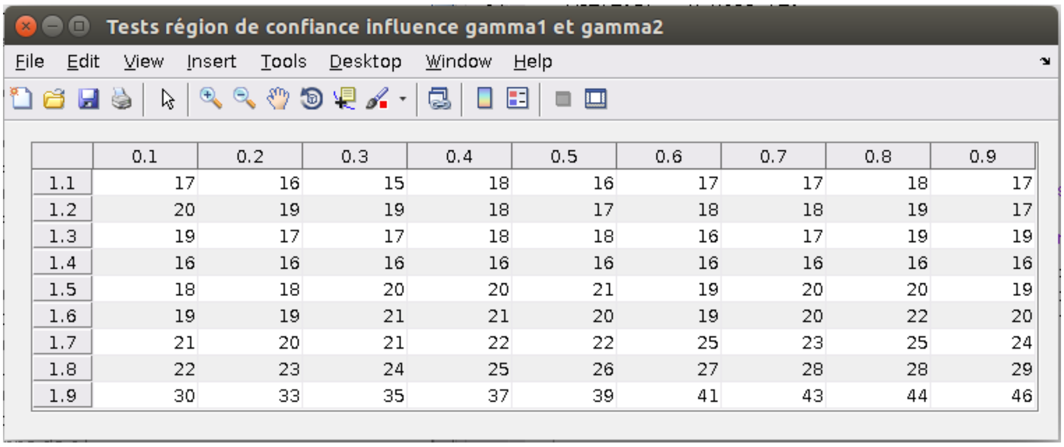
Tests région de confiance f1 More-Sorensen

	niter	appels gradient f	appels hessienne f
f1(x011)	2	3	3
f1(x012)	7	8	8

Tests région de confiance f2 More-Sorensen

	niter	appels gradient f	appels hessienne f
f2(x021)	29	23	23
f2(x022)	30	25	25
f2(x023)	21	16	16

Pour modifier la vitesse de convergence de la région de confiance on peut modifier delta_max la taille maximale de la région de confiance, gamma_1 et gamma_2 les facteurs de modification de la taille de la région de confiance et eta_1 et eta_2 les seuils de modification de la taille de la région de confiance. On obtiens entre autres les résultats suivant :



Tests région de confiance influence gamma1 et gamma2

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
1.1	17	16	15	18	16	17	17	18	17
1.2	20	19	19	18	17	18	18	19	17
1.3	19	17	17	18	18	16	17	19	19
1.4	16	16	16	16	16	16	16	16	16
1.5	18	18	20	20	21	19	20	20	19
1.6	19	19	21	21	20	19	20	22	20
1.7	21	20	21	22	22	25	23	25	24
1.8	22	23	24	25	26	27	28	28	29
1.9	30	33	35	37	39	41	43	44	46

On observe ainsi que la diminution de gamma_1 et gamma_2 réduit globalement le nombre d'itérations de l'algorithme.

Le seul critère d'arrêt pertinent pour la méthode de Newton pour les équations non linéaires est la distance à la solution (f(x)=0). On peut utiliser la distance des bornes de la dichotomie à la solution comme bonne approximation si tenté que lambda_min ou lambda_max n'est pas égale à la solution au départ. Grâce à la dichotomie il ne peut y avoir de stagnation.

Dans le cas où l'utilisateur ne fournit pas de couple lambda_min lambda_max vérifiant la condition, on peut augmenter la zone lambda_min lambda_max en diminuant lambda_min et augmentant lambda_max jusqu'à vérifier la condition.

2 Optimisation avec contraintes

2.1 Lagrangien augmenté

Résultat

Tests Lagrangien augmenté f2

	lambda	mu	niter	appels f2	appels gradient f2	appels hessienne f2	appels c2	appels gradient c2	appels hessienne c2	appels jacobienne c2
f2(xc21)	0.0387	11.3906	14	153	48	33	410	129	66	81
f2(xc22)	0.0387	11.3906	13	151	45	31	399	121	62	76

Tests Lagrangien augmenté f1

	lambda	mu	niter	appels f1	appels gradient f1	appels hessienne f1	appels c1	appels gradient c1	appels hessienne c1	appels jacobienne c1
f1(xc11)	4.5000	86.4976	19	105	57	37	332	57	74	94
f1(xc12)	4.5000	86.4976	19	105	57	37	332	57	74	94

Influence de tau

	niter
1	2
2	16
3	14
4	11
5	11
6	11
7	10
8	9
9	10
10	9
11	11
12	9
13	9
14	9
15	13

Au vu des résultats précédents, il semble difficile de déduire une influence précise de tau sur la performance de l'algorithme.

Supplément. Sans modifier l'algorithme, on peut transformer les contraintes d'inégalité en contraintes d'égalité en ajouter une variable réelle à la fonction contrainte dont le signe dépendra du sens de l'inégalité. On trouve ainsi une solution possible, il faut vérifier que le signe des variables est bien celui attendu, dans le cas contraire la solution est fausse et il faut utiliser une autre méthode pour trouver une solution.

Les performances des algorithme ont été ici étudiées en termes de nombre d'itérations. On aurait pu les étudier en termes de temps de calcul mais il aurait fallu supprimer les effets de bords qui pénalisent l'exécution.