École Nationale Supérieure
d'Électrotechnique, d'Électronique,
d'Informatique,
d'Hydraulique et des Télécommunications

Institut National Polytechnique

Optimisation Numérique : Rapport Projet

XAMBILI Robin

SOMMAIRE

1 Introduction

La première partie de ce TP-projet concerne les problèmes d'optimisation sans contraintes. On étudie la méthode de Newton et sa globalisation par l'algorithme des régions de confiance. La résolution du sous-problème des régions de confiance sera réalisée de deux façons, soit à l'aide du point de Cauchy, soit par l'algorithme de Moré-Sorensen. La seconde partie du projet exploite la partie précédente pour résoudre des problèmes d'optimisation avec contraintes par l'algorithme du Lagrangien augmenté.

2 Algorithme de Newton Local

2.1 Présentation

La fonction f étant C^2, on peut remplacer f au voisinage de l'iteré courant x_k par son développement de Taylor au second ordre, soit :
$q(y) = f(x_k) + \nabla f(x_k)^T(y - x_k) + 1/2(y - x_k)^T\nabla^2 f(x_k)(y - x_k)$.

Lorsque la matrice hessienne de la fonction est définie positive en x_k, une itération de la méthode de Newton locale revient à minimiser le modèle quadratique de la fonction en x_k.

Cela revient donc à trouver x qui vérifie : $\nabla q(x) = 0$, soit :
$x_{k+1} = x_k + d_k$
avec d_k unique solution de :
$\nabla^2 f(x_k)d_k = -\nabla f(x_k)$

2.2 Algorithme

L'algorithme prend en entrée $f, \nabla f, \nabla^2 f, x_0$, tol1 (tolérance de convergence sur le gradient), tol2 (tolérance de stagnation) et itermax et renvoie le minimum x.

Critère d'arrêt :
— convergence du gradient : $||\nabla f(x_k)|| \leq tol1 \times ||\nabla f(x_0)||$,
— stagnation : $||x_(k+1) - x_k|| \leq tol2$,
— nombre d'itération : $iter \geq itermax$

L'algorithme se déroule comme suit :
Tant que test de convergence non satisfait :

1. calculer d_k
2. mettre à jour x_k et k
3. mettre à jour le test de convergence

2.3 Tests de l'algorithme

Les test sont réalisés sur les fonctions f_1 et f_2 données en annexes (9.1) :

```
point de départ x011 pour f1

ans =

    1.0000
    1.0000
    1.0000

Nb d'appel à f : 0
Nb d'appel au gradient : 2
Nb d'appel à la Hessienne : 1
```

```
point de départ x022 pour f2

ans =

     1
     1

Nb d'appel à f : 0
Nb d'appel au gradient : 6
Nb d'appel à la Hessienne : 5
```

```
point de départ x023 pour f2
ans =

   1.0e+19 *

   -0.0000
    2.5000

Nb d'appel à f : 0
Nb d'appel au gradient : 5
Nb d'appel à la Hessienne : 4
```

Non convergent

2.4 Interprétations

- f_1 converge en une seule itération car c'est une fonction quadratique ;
- f_2 converge avec x_022 pour point initial cependant elle ne converge pas avec x_023 comme point initial ;
- divergence possible si le point de départ est trop éloigné de la solution ;
- vitesse de convergence quadratique dans des conditions favorables.

En bref, cette méthode converge rapidement mais la convergence n'est pas garantie et dépend du point de départ choisi .

3 Algorithme des Région de Confiance

3.1 Présentation

L'introduction d'une région de confiance dans la méthode de Newton permet de garantir la convergence globale de celle-ci, i.e. la convergence vers un optimum local quel que soit le point de départ. Cela suppose certaines conditions sur la résolution locale des sous-problèmes issus de la méthode, qui sont aisément imposables.

On veut approcher notre fonction f à minimiser par une fonction m_k dans l'espace $R_k = x_k + s; ||s|| < \Delta_k$ pour un Δ_k fixé, cette région doit être petite pour que l'approximation $m_k(x_k + s)$ $f(x_k + s)$ soit vraie.

Un exemple de modèle que l'on peut considérer est l'approximation de Taylor à l'ordre 2 (modèle quadratique) :
$m_k(x_k + s) = q_k(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s$
avec $g_k = \nabla f(x_k)$ et $H_k = \nabla^2 f(x_k)$.

L'astuce est de trouver un point $s_k (<= \Delta_k)$ qui minimise $m_k(x_k + s)$ puis de mettre à jour x_k et Δ_k.

3.2 Algorithme

L'algorithme prend en entrée $f, \nabla f, \nabla^2 f, x_0, \Delta_{Max}, \Delta_0, \gamma_1, \gamma_2, \eta_1, \eta_2$, tol1 (tolérance de convergence sur le gradient), tol2 (tolérance de stagnation) et itermax et renvoie le minimum x.

Critère d'arrêt :
— convergence du gradient : $||\nabla f(x_k)|| \leq tol1 \times ||\nabla f(x_0)||$,
— stagnation : $||x_(k+1) - x_k|| \leq tol2$,
— nombre d'itération : $iter \geq itermax$

L'algorithme se déroule comme suit :
Tant que test de convergence non satisfait :

1. calculer le point s_k soit par la méthode de Cauchy soit par Moré-Sorensen.
2. calculer $\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$
3. mettre à jour $x_{k+1} = \begin{cases} x_k + s_k & \text{si } \rho_k \geq \eta_1 \\ x_k & \text{sinon} \end{cases}$

4. Mettre à jour la région de confiance : $\Delta_{k+1} = \begin{cases} \min \gamma_2 \Delta_k, \Delta_{max} & \text{si } \rho_k \geq \eta_2 \\ \Delta_k & \text{si } \rho_k \in [\eta_1, \eta_2) \\ \gamma_1 \Delta_k & \text{sinon} \end{cases}$
5. on retourne x_k

3.3 Tests de l'algorithme

Les test sont réalisés sur les fonctions f_1 et f_2 données en annexes (9.1) avec la méthode du pas de Cauchy :

```
point de départ x011 pour f1

ans =

    1.0000
    1.0000
    1.0000

Nb d'appel à f : 103
Nb d'appel au gradient : 104
Nb d'appel à la Hessienne : 52
```

3.4 Interprétations

On remarque que les appels à f, ∇f et $\nabla^2 f$ sont plus nombreux pour l'algorithme des régions de confiance avec la méthode du pas de Cauchy qu'avec l'algorithme de Newton local. Donc, cette méthode est moins efficace que celle de Newton.

4 Algorithme du Pas de Cauchy

4.1 Présentation

On considère ici le modèle quadratique $q_k(s)$. Le sous-problème de régions de confiance correspondant peut se révéler difficile à résoudre (parfois autant que le problème de départ). Il est donc intéressant de se restreindre à une résolution approchée de ce problème. Le pas de Cauchy appartient à la catégorie des solutions approchées. Il s'agit de se restreindre au sous-espace engendré par le vecteur g_k ; le sous-problème s'écrit alors :

$$\begin{cases} \min q_k(s) \\ s = -t * g_k \\ t > 0 \\ ||s|| < \Delta_k \end{cases}$$

4.2 Algorithme

L'algorithme prend en entrée $\nabla_k f, \nabla_k^2 f, \Delta$ et renvoie le minimum s du problème ci-dessus.

4.3 Tests de l'algorithme

Les test sont réalisés sur les quadratiques données en annexes (9.2) :

```
---------------------------------------
Quadratique 1

ans =

     0

---------------------------------------
Quadratique 2, delta=0.1

ans =

   -0.0949
   -0.0316
```

```
---------------------------------------
Quadratique 2, delta=1

ans =

   -0.9231
   -0.3077

---------------------------------------
Quadratique 3, delta=3

ans =

    2.6833
   -1.3416
```

4.4 Interprétations

Ces tests permettent de valider l'implémentation de l'algorithme du pas de Cauchy.

5 Algorithme de Newton non Linéaire

5.1 Présentation

On voudra par la suite, résoudre des équations de la forme $\phi(\lambda) = 0$, où la fonction ϕ est une fonction non linéaire de la variable réelle. Cela sera réalisé (de façon approchée) par l'utilisation de la méthode de Newton, combinée avec une technique de dichotomie pour assurer sa convergence.

5.2 Algorithme

L'algorithme prend en entrée ϕ (la fonction dont on veut calculer le minimum), $\nabla\phi$ (fonction qui calcule le gradient de la fonction f), λ_{min} et λ_{max} (valeurs telle que point qui annule $\phi \in [\lambda_{min}; \lambda_{max}]$), tol (le critère d'arrêt de l'algorithme) et itermax ; et renvoie la racine λ du problème.

Critère d'arrêt :
— condition d'arrêt : $min(|\phi(\lambda_{min})|, |\phi(\lambda_{max})|) < tol$,
— nombre d'itération : $iter \geq itermax$

L'algorithme se déroule comme suit :

1. Si $min|\phi(\lambda_min)|, |\phi(\lambda_max)| < epsilon$, on termine avec la valeur appropriée pour λ^* ; sinon on prend $\lambda = \lambda_max$.
2. calculer $\lambda^N = \lambda - \frac{\phi(\lambda)}{\phi'(\lambda)}$.
3. si $\lambda^N \in [\lambda_{min}, \lambda_{max}]$ et $|\phi(\lambda^N)| < \frac{1}{2}|\phi(\lambda)|$, alors $\lambda = \lambda^N$.
4. sinon calculer $\lambda^D = \frac{\lambda_{min}+\lambda_{max}}{2}$, et on affecte à λ_{min} ou λ_{max} λ^D tel que le produit $\phi(\lambda_{min}) * \phi(\lambda_{max}) \leq 0$ reste valide, enfin on affecte $\lambda = \lambda^D$
5. si le critère d'arrêt est satisfait on sort avec $\lambda^* = \lambda$ sinon on reprend à λ^N.

5.3 Tests de l'algorithme

Les test sont réalisés sur les fonctions données en annexes (9.3) :

```
--------------------------------------
Test phi1 : delta = 0.5

ans =

    3.4965

Nb d'appel à phi : 229
Nb d'appel au gradphi : 100


--------------------------------------
Test phi21 : delta = 0.2

ans =

   82.6112

Nb d'appel à phi : 232
Nb d'appel au gradphi : 100


--------------------------------------
Test phi22 : delta = 0.7

ans =

   41.2304

Nb d'appel à phi : 231
Nb d'appel au gradphi : 100
```

5.4 Interprétations

Si le couple $(\lambda_{min}, \lambda_{max})$ fourni par l'utilisateur ne vérifie pas la condition $\phi(\lambda_{min}) \times \phi(\lambda max) < 0$ alors il faut envisager un pré-traitement. Ce pré-traitement pourrait consister en incrémentant $\lambda_m in$ jusqu'à que la condition

soit vérifiée ou $\lambda_m in = \lambda_m ax$, dans ce cas on décrémente $\lambda_m in$ jusqu'à que la condition soit vérifiée.

6 Algorithme de Moré-Sorensen

6.1 Présentation

Si s* est solution du problème $m_k(x_{k+1}) = \begin{cases} \min m_k(x_k + s) \\ ||s|| < \Delta_k \end{cases}$, avec le modèle quadratique pour approximation de la fonction à minimiser, que l'on représentera de manière générique par $\begin{cases} \min g^T s + \frac{1}{2} s^T H s \\ ||s|| < \Delta_k \end{cases}$, alors il existe $\lambda* \geq 0$ tel que $\begin{cases} (H + \lambda * I)s* = -g \\ \lambda * (||s * || - \Delta) = 0 \\ H + \lambda * I \succ 0 \\ \lambda* \geq 0 \\ ||s * || \leq \Delta \end{cases}$.

La résolution de ce système d'équations passe par la détermination de $\lambda*$.

7 Algorithme du Lagrangien Augmenté

7.1 Présentation

La méthode du lagrangien augmenté appartient à une classe d'algorithmes qui permettent la résolution des problèmes avec contraintes. Elle s'apparente aux méthodes de pénalisation, dans lesquelles on résout le problème avec contraintes à travers une suite de problèmes sans contraintes.

Ici on ne s'intéresse qu'aux problèmes avec contraintes d'égalité c'est à dire :

$$\begin{cases} \min f(x) \\ x \in R^n \\ c(x) = 0, \text{ où } c : R^n \to R^m \end{cases}$$

7.2 Algorithme

L'algorithme prend en entrée f (la fonction dont on veut calculer le minimum), ∇f (fonction qui calcule le gradient de la fonction f), $\nabla^2 f$ (fonction qui calcule la hessienne de la fonction f), c (la fonction qui calcule les contraintes), J_c (la fonction qui calcule la jacobienne des contraintes), H_c (la hessienne de c), x_0 (le point de départ de notre algorithme), μ_0 (le paramètre de pénalité), τ (valeur qui contrôle l'augmentation du paramètre de pénalité), λ_0 (la valeur initiale du multiplicateur de Lagrange), tol1, tol2, tol3 (les tolérances), itermax et solveur (qui correspond à l'algorithme utilisé pour le sous-problème sans contraintes) et renvoie le minimum x du problème ci-dessus, λ le multiplicateur de Lagrange associé aux contraintes d'égalité et μ le paramètre de pénalité.

Critère d'arrêt :

— CN1 : $||\nabla La(x_k)|| \leq tol1 \&\& ||c(x_k)|| \leq tol2$,
— stagnation : $||x_(k+1) - x_k|| \leq tol3 \&\& ||c(x_k)|| \leq tol2$,
— nombre d'itération : $iter \geq itermax$

L'algorithme se déroule comme suit :

Tant que test de convergence non satisfait :

1. on initialise les paramètres initiaux.
2. on résout avec les algorithmes vus précédemment le problèmes sans contraintes suivant : (avec x_k comme point de départ)
   $\min L_A(x, \lambda_k, \mu_k) = f(x) + \lambda_k^T c(x) + \frac{\mu_k}{2}||c(x)||^2$
3. on met à jour le test de convergence et on s'arrête si il y a convergence

4. sinon si $||c(x_{k+1})|| \leq \eta_k$ on met à jours les multiplicateurs :
$$\begin{cases} \lambda_{k+1} = \lambda_k + \mu_k c(x_{k+1}) \\ \mu_{k+1} = \mu_k \\ \epsilon_{k+1} = \frac{\epsilon_k}{\mu_k} \\ \eta_{k+1} = \frac{\eta_k}{\mu_k^{\beta}} \\ k = k + 1 \end{cases}$$
5. sinon on met à jour le paramètre de pénalité :
$$\begin{cases} \lambda_{k+1} = \lambda_k \\ \mu_{k+1} = \tau * \mu_k \\ \epsilon_{k+1} = \frac{\epsilon_0}{\mu_{k+1}} \\ \eta_{k+1} = \frac{\hat{\eta}_0}{\mu_{k+1}^{\alpha}} \\ k = k + 1 \end{cases}$$

7.3 Tests de l'algorithme

Les tests sont réalisés sur les fonctions f_1 et f_2 données en annexe (9.5) avec l'algorithme de Newton local pour le sous-problème sans contraintes

```
---------------------------------------
point de départ xc11 pour f1

ans =

    0.5000
    1.2500
    0.5000

Nb d'appel à f : 0
Nb d'appel au gradient : 3
Nb d'appel à la Hessienne : 1
```

```
-------------------------------------
point de départ xc12 pour f1

ans =

    0.5000
    1.2500
    0.5000

Nb d'appel à f : 0
Nb d'appel au gradient : 3
Nb d'appel à la Hessienne : 1
```

7.4 Interprétations

On retrouve une convergence en une seule itération puisqu'on utilise Newton local

8 Annexes : Problèmes et cas tests

8.1 Problèmes sans contraintes

Les problèmes de minimisation sans contraintes à résoudre sont les suivants :

Problème 1

$$\begin{aligned} f_1 : \quad \mathbb{R}^3 &\to \mathbb{R} \\ (x_1, x_2, x_3) &\mapsto 2\,(x_1 + x_2 + x_3 - 3)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2. \end{aligned}$$

On cherchera à minimiser f_1 sur $\mathbb{R}^3$, en partant des points suivants

$$x_{011} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad x_{012} = \begin{bmatrix} 10 \\ 3 \\ -2.2 \end{bmatrix}.$$

Problème 2

$$\begin{aligned} f_2 : \quad \mathbb{R}^2 &\to \mathbb{R} \\ (x_1, x_2) &\mapsto 100\,(x_2 - x_1^2)^2 + (1 - x_1)^2. \end{aligned}$$

On cherchera à minimiser f_2 sur $\mathbb{R}^2$, en partant des points suivants

$$x_{021} = \begin{bmatrix} -1.2 \\ 1 \end{bmatrix}, \quad x_{022} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}, \quad x_{023} = \begin{bmatrix} 0 \\ \frac{1}{200} + \frac{1}{10^{12}} \end{bmatrix}.$$

8.2 Cas tests pour le calcul du pas de Cauchy

On considère des fonctions quadratiques de la forme $q(s) = s^\top g + \frac{1}{2} s^\top H\, s$.

Quadratique 1

$$g = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad H = \begin{bmatrix} 7 & 0 \\ 0 & 2 \end{bmatrix}.$$

Quadratique 2

$$g = \begin{bmatrix} 6 \\ 2 \end{bmatrix}, \quad H = \begin{bmatrix} 7 & 0 \\ 0 & 2 \end{bmatrix}.$$

Quadratique 3

$$g = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \quad H = \begin{bmatrix} -2 & 0 \\ 0 & 10 \end{bmatrix}.$$

8.3 Fonctions tests pour l'algorithme de résolution d'équations non linéaires

On pourra tester sur des fonctions du type

$$\varphi(\lambda) = \|s(\lambda)\|^2 - \delta^2 \quad \text{ou} \quad \varphi(\lambda) = \frac{1}{\|s(\lambda)\|^2} - \frac{1}{\delta^2},$$

avec :

1) $\|s(\lambda)\|^2 = \frac{4}{(\lambda+2)^2} + \frac{36}{(\lambda+14)^2}$ et $\delta = 0.5$;

2) $\|s(\lambda)\|^2 = \frac{4}{(\lambda-38)^2} + \frac{400}{(\lambda+20)^2}$ et $\delta \in \{0.2, 0.7\}$.

8.4 Cas tests pour l'algorithme de Moré-Sorensen

On reprendra les 3 quadratiques testées avec le pas de Cauchy, auxquelles on ajoutera :

Quadratique 4

$$g = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad H = \begin{bmatrix} -2 & 0 \\ 0 & 10 \end{bmatrix}.$$

Quadratique 5

$$g = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad H = \begin{bmatrix} 4 & 6 \\ 6 & 5 \end{bmatrix}.$$

Quadratique 6

$$g = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad H = \begin{bmatrix} 4 & 0 \\ 0 & -15 \end{bmatrix}.$$

8.5 Problèmes avec contraintes

Retour sur f_1 On s'intéresse à la valeur minimale de f_1 sur un ensemble défini par une contrainte linéaire. La formulation du problème sera alors

$$\min_{x\in\mathbb{R}^3} f_1(x) \quad \text{s.t.} \quad x_1 + x_3 = 1.$$

On choisira comme point initial

$$x_{c11} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \text{(réalisable)} \quad \text{ou} \quad x_{c12} = \begin{bmatrix} 0.5 \\ 1.25 \\ 1 \end{bmatrix} \text{(non réalisable).}$$

Retour sur f_2 On cherche à minimiser la fonction f_2 décrite dans la partie précédente, en se restreignant maintenant à une sphère. Le problème s'écrit :

$$\min_{x\in\mathbb{R}^2} f_2(x) \quad \text{s.t.} \quad x_1^2 + x_2^2 = 1.5.$$

On choisira comme point initial

$$x_{c21} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{(non réalisable)} \quad \text{ou} \quad x_{c22} = \begin{bmatrix} \sqrt{3}/2 \\ \sqrt{3}/2 \end{bmatrix} \text{(réalisable).}$$

Un problème avec contraintes d'inégalité (supplément)

$$\begin{cases} \min_{(x,y)\in\mathbb{R}^2} f_3(x,y) & = & (x-1)^2 + (y-2.5)^2 \\ x - 2y + 2 & \geq & 0 \\ -x - 2y + 6 & \geq & 0 \\ -x + 2y + 2 & \geq & 0 \\ x & \geq & 0 \\ y & \geq & 0 \end{cases}$$

L'origine pourra être prise comme point initial.