TP-Projet ENSEEIHT
Méthodes numériques pour les problèmes d'optimisation

O.Cots, J. Gergaud, L. Le Gorrec, S. Gratton, C. Royer, D. Ruiz et E. Simon

Année universitaire 2018–2019

Résumé

Ce document constitue le sujet du projet d'Optimisation Numérique pour l'année 2018-2019. Le projet est réalisé lors des séances de TP (5 séances encadrées, et 5 séances non encadrées – un enseignant étant a priori disponible sur demande pendant ces créneaux là), à l'issue desquelles les étudiants doivent produire une étude numérique des algorithmes implémentés. Cette étude se basera sur les tests proposés en annexe, ainsi que les pistes d'interprétation proposées pour chaque partie.

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 du Gradient Conjugué Tronqué.

La seconde partie du projet exploite la partie précédente pour résoudre des problèmes d'optimisation avec contraintes par l'algorithme du Lagrangien augmenté.

Optimisation sans contraintes

Dans cette partie, on s'intéresse à la résolution du problème

$$\min_{x \in \mathbb{R}^n} f(x),$$

où la fonction f est de classe $\mathcal{C}^2$ sur $\mathbb{R}^n$. On cherche donc à exploiter l'information fournie par ses dérivées première et seconde, que l'on représente en tout point x par le *vecteur gradient* $\nabla f(x) \in \mathbb{R}^n$ et la *matrice Hessienne* $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$.[1]

1 Algorithme de Newton local

Principe

La fonction f étant $\mathcal{C}^2$, on peut remplacer f au voisinage de l'itéré courant x_k par son développement de Taylor au second ordre, soit :

$$f(y) \sim q(y) = f(x_k) + \nabla f(x_k)^T (y - x_k) + \frac{1}{2}(y - x_k)^T \, \nabla^2 f(x_k) \, (y - x_k),$$

On choisit alors comme point x_{k+1} le minimum de la quadratique q lorsqu'il existe et est unique, ce qui n'est le cas que si $\nabla^2 f(x_k)$ est définie positive. Or le minimum de q est réalisé par x_{k+1} solution de : $\nabla q(x_{k+1}) = 0$, soit :

$$\nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0,$$

ou encore, en supposant que $\nabla^2 f(x_k)$ est définie positive :

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k).$$

La méthode ne doit cependant jamais être appliquée en utilisant une inversion de la matrice Hessienne (qui peut être de très grande taille et mal conditionnée), mais plutôt en utilisant :

$$x_{k+1} = x_k + d_k,$$

où d_k est l'unique solution du système linéaire

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k),$$

d_k étant appelée direction de Newton.

Cette méthode est bien définie si à chaque itération, la matrice hessienne $\nabla^2 f(x_k)$ est définie positive : ceci est vrai en particulier au voisinage de la solution x^* cherchée si on suppose que $\nabla^2 f(x^*)$ est définie positive (par continuité de $\nabla^2 f$).

1.1 Algorithme

Algorithme 1 ALGORITHME DE NEWTON (LOCAL)

Données : f , x_0 première approximation de la solution cherchée, $\epsilon > 0$ précision demandée.

Sortie : une approximation de la solution du problème $\min_{x \in \mathbb{R}^n} f(x)$.

1.Tant que le test de convergence est non satisfait :

a. Calculer d_k solution du système : $\nabla^2 f(x_k) \, dk = -\nabla f(x_k)$,

b. Mise à jour : $x_{k+1} = x_k + d_k$, $k = k + 1$,

2. Retourner x_k.

1. cf le cours de Calcul Différentiel en 1A.

1.2 Travail à réaliser

Implémentation

1. Coder l'algorithme de Newton local tel que décrit ci-dessus.
2. Tester l'algorithme sur les fonctions f_1, f_2 avec les points initiaux x_{011}, x_{012} (pour f_1) et $x_{021}, x_{022}, x_{023}$ (pour f_2) donnés en Annexe A.

Interprétation

Justifier que

1. l'algorithme implémenté converge en une itération pour f_1,
2. l'algorithme puisse ne pas converger pour f_2 avec certains points initiaux.

2 Régions de confiance - Partie 1

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.

Principe

L'idée de la méthode des régions de confiance est d'approcher f par une fonction modèle plus simple m_k dans une région $R_k = \{x_k + s; \|s\| \leq \Delta_k\}$ pour un Δ_k fixé.

Cette région dite "de confiance" doit être suffisament petite pour que

$$m_k(x_k + s) \sim f(x_k + s).$$

Le principe est que, au lieu de résoudre l'équation : $f(x_{k+1}) = \min_{\|s\| \leq \Delta_k} f(x_k + s)$, on résout :

$$m_k(x_{k+1}) = \min_{\|s\| \leq \Delta_k} m_k(x_k + s) \tag{2.1}$$

Si la différence entre $f(x_{k+1})$ et $m_k(x_{k+1})$ est trop grande, on diminue le Δ_k (et donc la région de confiance) et on résout le modèle (2.1) à nouveau. Un avantage de cette méthode est que toutes les directions sont prises en compte. Par contre, il faut faire attention à ne pas trop s'éloigner de x_k ; en général, la fonction m_k n'approche proprement f que sur une région proche de x_k.

Exemple de modèle : l'approximation de Taylor à l'ordre 2 (modèle quadratique) :

$$m_k(x_k + s) = q_k(s) = f(x_k) + g_k^\top s + \frac{1}{2} s^\top H_k s \tag{2.2}$$

avec $g_k = \nabla f(x_k)$ et $H_k = \nabla^2 f(x_k)$.

2.1 Algorithme

Algorithme 2 MÉTHODE DES RÉGIONS DE CONFIANCE (ALGO GÉNÉRAL)

Données : $\Delta_{max} > 0$, $\Delta_0 \in (0, \Delta_{max})$, $0 < \gamma_1 < 1 < \gamma_2$ et $0 < \eta_1 < \eta_2 < 1$.

Sortie : une approximation de la solution du problème : $\min_{x \in \mathbb{R}^n} f(x)$.

1.Tant que le test de convergence n'est pas satisfait :

a. Calculer approximativement s_k solution du sous-problème (2.1) ;

b. Evaluer $f(x_k + s_k)$ et $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$

c. Mettre à jour l'itéré courant :

$$x_{k+1} = \begin{cases} x_k + s_k & \text{si } \rho_k \geq \eta_1 \\ x_k & \text{sinon.} \end{cases}$$

d. Mettre à jour la région de confiance :

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_2\, \Delta_k, \Delta_{\max}\} & \text{si } \rho_k \geq \eta_2 \\ \Delta_k & \text{si } \rho_k \in [\eta_1, \eta_2) \\ \gamma_1\, \Delta_k & \text{sinon.} \end{cases}$$

2. Retourner x_k.

L'algorithme 2 est un cadre générique. On va s'intéresser à deux raffinages possibles de l'étape a.

2.2 Le pas de Cauchy

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

$$\begin{cases} \min & q_k(s) \\ s.t. & s = -t\, g_k \\ & t > 0 \\ & \|s\| \leq \Delta_k. \end{cases} \tag{2.3}$$

2.3 Travail à réaliser

Implémentation

1. Implémenter une fonction calculant **à part** le pas de Cauchy d'un sous-problème de régions de confiance. La tester sur les quadratiques proposées en Annexe B.
2. Inclure en suivant ce calcul dans un algorithme de régions de confiance ; le tester sur les problèmes de l'Annexe A.

Interprétation

1. Quelle relation lie la fonction test f_1 et son modèle de Taylor à l'ordre 2 ? Comparer alors les performances de Newton et RC-Pas de Cauchy sur cette fonction.
2. Le rayon initial de la région de confiance est un paramètre important dans l'analyse de la performance de l'algorithme. Sur quel(s) autre(s) paramètre(s) peut-on jouer pour essayer d'améliorer cette performance ? Étudier l'influence d'au moins deux de ces paramètres.

3 Régions de confiance - Partie 2

Dans la section précédente, on a pu voir que la technique du pas de Cauchy ne garantit pas une convergence rapide en général ; on retrouve ici le problème d'une méthode de descente de gradient. On souhaite donc étudier une méthode pour la résolution approchée du sous-problème avec région de confiance (2.1), qui puisse récupérer asymptotiquement la convergence quadratique inhérente à la méthode de Newton Local. L'algorithme du Gradient Conjugué Tronqué appartient à cette catégorie.

3.1 Algorithme du Gradient Conjugué Tronqué

On s'intéresse maintenant à la résolution approchée du problème (2.1) à l'itération k de l'algorithme 2 des Régions de Confiance. On considère pour cela l'algorithme du Gradient Conjugué Tronqué (vu en cours), rappelé ci-après :

Algorithme 3 ALGORITHME DU GRADIENT CONJUGUÉ TRONQUÉ

Données : $\Delta_k > 0, x_k, g = \nabla f(x_k), H = \nabla^2 f(x_k)$.

Sortie : le pas s qui approche la solution du problème : $\min_{\|s\| \leq \Delta_k} q(s)$
où $q(s) = g^\top s + \frac{1}{2} s^\top H_k s$.

Initialisations : $s_0 = 0, g_0 = g, p_0 = -g$;

1. Pour $j = 0, 1, 2, \dots$, **faire :**

a. $\kappa_j = p_j^T H p_j$

b. Si $\kappa_j \leq 0$, alors
déterminer σ_j la racine de l'équation $\|s_j + \sigma p_j\|_2 = \Delta_k$
pour laquelle la valeur de $q(s_j + \sigma p_j)$ est la plus petite.
Poser $s = s_j + \sigma_j p_j$ et sortir de la boucle.
Fin Si

c. $\alpha_j = g_j^T g_j / \kappa_j$

d. Si $\|s_k + \alpha_k p_k\|_2 \geq \Delta_k$, alors
déterminer σ_j la racine positive de l'équation $\|s_j + \sigma p_j\|_2 = \Delta_k$.
Poser $s = s_j + \sigma_j p_j$ et sortir de la boucle.
Fin Si

e. $s_{j+1} = s_j + \alpha_j p_j$

f. $g_{j+1} = g_j + \alpha_j H p_j$

g. $\beta_j = g_{j+1}^T g_{j+1} / g_j^T g_j$

h. $p_{j+1} = g_{j+1} + \beta_j p_j$

i. Si la convergence est suffisante, poser $s = s_{j+1}$ et sortir de la boucle.

2. Retourner s**.**

3.2 Travail à réaliser - Étape 2

Implémentation

1. Implémenter l'algorithme du Gradient Conjugué Tronqué, en se basant sur le cours. On validera les résultats sur les fonctions de l'Annexe C.
2. Intégrer finalement l'algorithme du Gradient Conjugué Tronqué dans un code de régions de confiance, et appliquer ce code pour résoudre les exemples proposés en Annexe A.

Interprétation

1. Comparer la décroissance obtenue avec celle du pas de Cauchy, en retournant, dans un premier temps le dernier itéré admissible à courbure positive (c'est à dire, que si l'une ou l'autre des deux conditions (b) ou (d) sont rencontrées dans l'algorithme 3, alors on ne calcule pas σ_j et on retourne le dernier itéré s_j directement).
2. Comparer la décroissance obtenue avec celle du pas de Cauchy, en imposant la sortie dans l'algorithme 3 au bout d'une itération seulement. Que remarquez vous ?
3. Comparer la décroissance obtenue avec celle du pas de Cauchy dans le cas général.
4. Quels sont les avantages et inconvénients des deux approches ?

Optimisation avec contraintes

Dans cette partie, nous nous intéressons à la résolution des problèmes sous contraintes. Le problème se présente donc sous la forme suivante :

$$\min_{x \in \mathbb{R}^n} f(x) \quad \textit{sous la contrainte} \ : x \in C,$$

où C est un sous-ensemble non vide de $\mathbb{R}^n$.

4 Lagrangien Augmenté

4.1 Principe

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 probleme avec contraintes à travers une suite de problèmes sans contraintes.

4.2 Algorithme du Lagrangien augmenté pour contraintes d'égalité

On s'intéresse ici au cas où l'ensemble C est défini par un ensemble d'égalités. Le problème se met ainsi sous la forme

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad c(x) = 0,$$

où $c : \mathbb{R}^n \to \mathbb{R}^m$. L'algorithme suivant est obtenu de Bierlaire, *Introduction à l'optimisation différentiable*.

Algorithme 4 MÉTHODE DU LAGRANGIEN AUGMENTÉ (CONTRAINTES D'ÉGALITÉ)

Données : $\mu_0 > 0$, $\tau > 0$, $\hat{\eta}_0 = 0.1258925$[2], $\alpha = 0.1$, $\beta = 0.9$, $\epsilon_0 = 1/\mu_0$, $\eta_0 = \hat{\eta}_0/\mu_0^\alpha$, et un point de départ du Lagrangien (x_0, λ_0). On pose $k = 0$.

Sortie : une approximation de la solution du problème avec contraintes.

1. Tant qu'il n'y a pas convergence, répéter

a. Calculer approximativement un minimiseur x_{k+1} du problème sans contraintes suivant :

$$\min_{x \in \mathbb{R}^n} L_A(x, \lambda_k, \mu_k) = f(x) + \lambda_k^T c(x) + \frac{\mu_k}{2} \|c(x)\|^2,$$

avec x_k comme point de départ, en terminant lorsque $\|\nabla_x L_A(., \lambda^k, \mu_k)\| \leq \epsilon_k$. Si convergence de l'algorithme global, s'arrêter, sinon aller en b.

b. Si $\|c(x_{k+1})\| \leq \eta_k$, mettre à jour (entre autres) les multiplicateurs :

$$\left\{ \begin{array}{lcl} \lambda_{k+1} & = & \lambda_k + \mu_k \, c(x_{k+1}) \\ \mu_{k+1} & = & \mu_k \\ \epsilon_{k+1} & = & \epsilon_k / \mu_k \\ \eta_{k+1} & = & \eta_k / \mu_k^\beta \\ k & = & k+1 \end{array} \right. ,$$

2. Pour que $\eta_0 = 0.1$.

c. Autrement, mettre à jour (entre autres) le paramètre de pénalité :

$$\begin{cases} \lambda_{k+1} & = & \lambda_k, \\ \mu_{k+1} & = & \tau\, \mu_k \\ \epsilon_{k+1} & = & \epsilon_0/\mu_{k+1} \\ \eta_{k+1} & = & \hat{\eta}_0/\mu_{k+1}^{\alpha} \\ k & = & k+1 \end{cases}.$$

2. Retourner x_k, λ_k, μ_k**.**

4.3 Travail à réaliser

Implémentation

1. Choisir des critères d'arrêt pour la convergence de l'algorithme.
2. Implémenter l'algorithme du lagrangien augmenté, en utilisant les différentes méthodes qui ont été vues en première partie pour la résolution de la suite de problémes sans contraintes.
3. Tester les différentes variantes sur les problèmes en Annexe D.

Interprétation

1. Commenter les résultats obtenus, en étudiant notamment les valeurs de λ_k et μ_k.
2. Étudier l'influence du paramètre τ dans la performance de l'algorithme.
3. **Supplément :** Que proposez-vous comme méthode pour la résolution des problèmes avec des contraintes à la fois d'égalité et d'inégalité ? Implémenter (si le temps le permet) ce nouvel algorithme.

A Problèmes sans contraintes

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

Problème 1

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

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

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

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

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

B 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

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

C Cas tests pour la résolution du sous-problème par l'algorithme du Gradient Conjugué Tronqué

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

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

D 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)

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

L'origine pourra être prise comme point initial.