ANSD, Dakar
ISE 1

Optimisation et convexité

Chapitre 3 : Méthodes de relaxation

Dr. Oumar Diop
oumardiop32@yahoo.fr

25 avril 2025

Généralités

Considérons dans ce séquence le cas particulier où l'ensemble des contraintes U est l'espace $\mathbb{R}^n$ et on considère le le problème de minimisation sans contraintes

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

où $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ est une fonction donnée. On suppose que la solution x^* du problème (1) existe et est unique. L'objectif de ce séquence de trouver une approximation numérique de x^* en construisant une suite $\{x^{(k)}\}_{k \in \mathbb{N}}$ de $\mathbb{R}^n$ qui converge vers x^*.
Nous avons vu dans le séquence précédent que :

- Si f est $\mathcal{C}^1$ alors on a nécessairement

$$\nabla f(x^*) = 0. \qquad (Eq.\ \ d'Euler)$$

- Si f est convexe alors l'équation d'Euler est une condition suffisante de minimum. Si de plus, f est strictement convexe, alors x^* est unique.

Alors une éventuelle méthode numérique aurait pour but de résoudre l'équation d'Euler qui est un système de n équations à n inconnues du type

$$F(x) = 0, \quad x \in \mathbb{R}^n.$$

où $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ est une fonction donnée ($F(x) = \nabla f(x)$).
Pour résoudre le système d'équations non linéaires $F(x) = 0$, on peut utiliser la méthode de Broyden ou celle de Newton-Raphson.

Généralités

Dans le cas particulier où

$$f(x) = \frac{1}{2} < Ax, x > - < b, x > + c,$$

avec A une matrice symétrique, b un vecteur et c un scalaire, alors l'équation d'Euler devient un système d'équations linéaires donné par :

$$Ax = b.$$

Dans ce cas, on peut utiliser les méthodes de Gauss, Lu, Cholesky, Jacobi, ... Nous allons présenter, dans ce séquence, des méthodes spécifiques à l'optimisation. Elles consistent à construire une suite récurrente $\{x^{(k)}\}$ dans le but qu'elles convergent vers la solution de x^* de l'équation d'Euler. La forme générale de la suite est :

$$x^{(k+1)} = x^{(k)} + \rho_k d^{(k)}.$$

- Le vecteur $d^{(k)}$ est appelé direction de descente ;
- Le scalaire ρ_k est appelé facteur de descente.

Il y a beaucoup de méthodes numériques de minimisation. Elles diffèrent selon le choix de $d^{(k)}$ et ρ_k.

Généralités : algorithme de descente

On construit alors une suite $(x^{(k)})$ vérifiant $f(x^{(k+1)}) \leq f(x^{(k)})$ pour tout $k \in \mathbb{N}$. L'algorithme général est donné par les trois étapes suivants.

- Etape 1 : Initialisation
  - Choisir $k = 0$;
  - choisir $x^{(0)} \in \mathbb{R}^n$, par exemple $x^{(0)} = 0$;
  - choisir le nombre maximum d'itération (par exemple $k_{max} = 10^7$) ;
  - fixer la tolérance $\epsilon = 10^{-7}$ par exemple.
- Etape 2 : Boucle Tant que ("test d'arrêt" est faux) et ($k < k_{max}$) faire :
  - $x^{(k+1)} = x^{(k)} + \rho_k d^{(k)}$;
  - Incrémenter le compteur : $k = k + 1$.
- Etape 3 : Convergence
  - Si ("test darrêt" est vrai) alors la suite est convergente c'est à dire $x^{(k)}$ est une approximation du minimum recherché ;
  - Sinon la méthode n'a pas convergée.

NB : le test d'arrêt sera $\|\nabla f(x^{(k)}\| \leq \epsilon$;

Plan

Généralités

Méthode de relaxation
- Description de la méthode
- Algorithme de la méthode de relaxation
- Application aux fonctions quadratiques

Description de la méthode de relaxation

- La méthode de relaxation considère les directions de descente suivantes :

$$d^{(0)} = e_1, d^{(1)} = e_2, d^{(2)} = e_3, \ldots, d^{(n-1)} = e_n,$$

ensuite

$$d^{(n)} = e_1, d^{(n+1)} = e_2, d^{(n+2)} = e_3, \ldots, d^{(2n-1)} = e_n.$$

et ainsi de suite, ...

En d'autre termes $d^{(k)} = e_l$ si et seulement si l est le reste de la division euclidienne de $k+1$ par n.

On rappelle que $\{e_1, e_2, e_3, \ldots, e_n\}$ sont les vecteurs de la base canonique de $\mathbb{R}^n$.

- Choix des facteurs de descente : Ils doivent vérifier

$$f(x^{(k+1)}) = \min_{\rho \in \mathbb{R}} f(x^{(k)} + \rho d^{(k)}) \text{ pour tout k } \in \mathbb{N}.$$

Ici, on pose :

$$x^{(k+1)} = x^{(k)} + \rho_k d^{(k)}.$$

Plan

Généralités

Méthode de relaxation
- Description de la méthode
- Algorithme de la méthode de relaxation
- Application aux fonctions quadratiques

Algorithme de la méthode de relaxation

- Supposons que le vecteur $x^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots, x_n^{(k)}\right)$ est connu et que

$$x^{(k+1)} = x^{(k)} + \rho_k d^{(k)}.$$

- On calcule le vecteur $x^{(k+1)} = \left(x_1^{(k+1)}, x_2^{(k+1)}, x_3^{(k+1)}, \dots, x_n^{(k+1)}\right)$ en n pas successifs par les formules suivantes.
  - Etape 1

$$f\left(x_1^{(k+1)}, x_2^{(k)}, x_3^{(k)} \dots, x_n^{(k)}\right) = \min_{\alpha\in\mathbb{R}} f\left(\alpha, x_2^{(k)}, x_3^{(k)} \dots, x_n^{(k)}\right)$$

  - Etape 2

$$f\left(x_1^{(k+1)}, x_2^{(k+1)}, x_3^{(k)} \dots, x_n^{(k)}\right) = \min_{\alpha\in\mathbb{R}} f\left(x_1^{(k+1)}, \alpha, x_3^{(k)} \dots, x_n^{(k)}\right)$$

$$\dots \qquad\qquad \dots$$

  - Etape n

$$f\left(x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_n^{(k+1)}\right) = \min_{\alpha\in\mathbb{R}} f\left(x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{n-1}^{(k+1)}, \alpha\right).$$

Montrons maintenant que l'algorithme de ralaxation est bien défini, au moins pour le cas des fonctions elliptiques.

Algorithme de la méthode de relaxation

Proposition 1

Soit une fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *et* a, b *deux vecteurs de* $\mathbb{R}^n$. *On définit la fonction* $g : \mathbb{R} \longrightarrow \mathbb{R}$ *par :*

$$g(t) = f(a + bt) \quad \text{pour tout } t \in \mathbb{R}.$$

Alors on a :

1. *Si* f *est convexe, alors* g *est convexe,*
2. *Si* f *est strictement convexe et* $b \neq 0$ *alors* g *est strictement convexe,*
3. *Si* f *est elliptique et* $b \neq 0$, *alors* g *est elliptique.*

Pour la preuve de cette proposition,

1. On utilise la définition de la convexité de f en posant $t_1, t_2 \in \mathbb{R}$ et $\alpha \in [0, 1], \ldots$.
2. Même chose avec $t_1 \neq t_2$;
3. Soient $t_1, t_2 \in \mathbb{R}$, on calcule

$$\left(g'(t_1) - g'(t_2)\right)(t_1 - t_2) = \ldots.$$

et on utilise la propriété selon laquelle f est elliptique.

Exercice : Reprendre la preuve.

Corollaire 2

Soit une fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ et g sa fonction partielle par rapport à la ième variable. Alors on a :

1. *Si f est convexe alors g est convexe,*
2. *Si f est strictement convexe alors g est strictement convexe,*
3. *Si f est elliptique alors g est elliptique.*

On donne le résultat suiavnt.

Théorème 3

Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une fonction elliptique. Alors la méthode de relaxation est bien définie et elle converge
i.e : la suites récurrente $(x^{(k)})_{k\in\mathbb{N}}$ converge vers l'unique minimum x^* de la fonction f.

Plan

Généralités

Méthode de relaxation
- Description de la méthode
- Algorithme de la méthode de relaxation
- Application aux fonctions quadratiques

Application aux fonctions quadratiques

On considère la fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ définie par :

$$f(x) = \frac{1}{2} < Ax, x > - < b, x > + c, \qquad (2)$$

où A est une matrice symétrique définie positive, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$.
Problème : Déterminer $y^* \in \mathbb{R}^n$ vérifiant :

$$g(y^*) = \min_{y \in \mathbb{R}} g(y)$$

où g est une fonction réelle d'une variable réelle définie par :

$$g(y) = f\left(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, y, x_{i+1}^{(k)}, \ldots, x_n^{(k)}\right)$$

avec

$$x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \ldots, x_n^{(k)} \in \mathbb{R} \quad \text{données.}$$

Posons

$$v_1 = x_1^{(k+1)}, \ldots, v_{i-1} = x_{i-1}^{(k+1)}, v_{i+1} = x_{i+1}^{(k)}, \ldots, v_n = x_n^{(k)}.$$

1. La fonction g admet-elle un unique minimum ?
2. Ecrire l'équation d'Euler associée à la fonction g.
3. Vérifier que la solution y^* est donnée par :

$$y^* = \frac{1}{A_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} A_{ij} v_j\right).$$

Application aux fonctions quadratiques

Proposition 4

La méthode de relaxation appliquée à la fonction quadratique f donnée par (2) (où A est symétrique définie positive) s'écrit :

$$x_1^{k+1} = \frac{1}{A_{11}} \left(b_1 - \sum_{j>1}^{n} A_{1j} x_j^{(k)} \right) ;$$

$$x_2^{k+1} = \frac{1}{A_{22}} \left(b_2 - A_{21} x_1^{(k+1)} - \sum_{j>2}^{n} A_{2j} x_j^{(k)} \right) ;$$

$$\dots$$

$$x_i^{k+1} = \frac{1}{A_{ii}} \left(b_i - \sum_{j<i}^{n} A_{2j} x_j^{(k+1)} - \sum_{j>i}^{n} A_{2j} x_j^{(k)} \right) ;$$

$$\dots$$

$$x_n^{k+1} = \frac{1}{A_{nn}} \left(b_n - \sum_{j<n}^{n} A_{nj} x_j^{(k+1)} \right) .$$

La méthode de relaxation converge.

TP 5 (A présenter par groupe de deux étudiants)

On considère la fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *définie par :*

$$f(x) = \frac{1}{2} < Ax, x > - < b, x > +c, \quad (3)$$

où A *est une matrice symétrique définie positive,* $b \in \mathbb{R}^n$ *et* $c \in \mathbb{R}$*.*
Ecrire un code scilab permettant d'obtenir le minimum de la fonction f *par la méthode de relaxation.*
Indication : *le programme prendra en argument une matrice* A*, un vecteur* b *et un réel* c*. Il permettra de vérifier si la matrice* A *est symétrique définie positive et de chercher le minimum.*