Minimisation convexe

TP 1 : Méthodes de gradient

Exercice 1

Le but de cet exercice est de vous familiariser avec les méthodes numériques de base pour la résolution de problèmes de minimisation convexe. Pour cela, nous considérerons dans un premier temps la fonctionnelle $f : \mathbb{R}^2 \to \mathbb{R}$ suivante

$$f(x_1, x_2) = 2x_1^2 + x_2^2 + x_1 x_2 + x_1 + x_2 \tag{1}$$

montrée sur la figure 1.

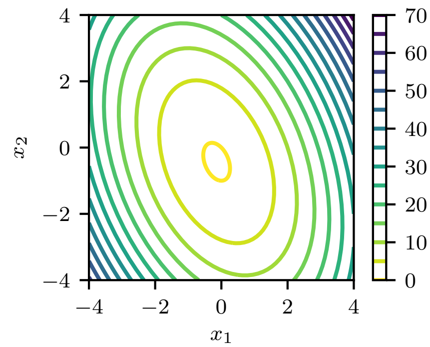

Figure 1: Isocontours de la fonction $f(x_1, x_2)$ donnée par (1).

L'objectif est donc de trouver $\mathbf{x}^* = \begin{bmatrix} x_1^* & x_2^* \end{bmatrix}^T$ tel que

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\ f(\mathbf{x}) \tag{2}$$

Bien que, dans le cas présent, ce problème puisse être résolu analytiquement, nous utiliserons ce petit exemple comme un cas test afin de valider les divers algorithmes de minimisation que vous aurez à implémenter avant de s'attaquer à des problèmes plus réalistes.

Question n°1

Mettez la fonction $f(x_1, x_2)$ sous la forme suivante

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}, \tag{3}$$

puis exprimez le gradient $\nabla f(\mathbf{x})$ et la matrice Hessienne $\mathbf{H}_{ij} = \dfrac{\partial^2 f}{\partial x_i \partial x_j}$ en fonction de $\mathbf{A}$, $\mathbf{x}$ et $\mathbf{b}$.

Question n^{o}2

Montrez que minimiser la fonctionnelle quadratique (3) revient à résoudre le problème linéaire suivant

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{4}$$

puis résoudre ce problème analytiquement pour $f(\mathbf{x})$ donnée par (1).

Question n^{o}3

Bien que, pour le cas présent, le système linéaire (4) puisse être résolu analytiquement, il est préférable en règle générale (e.g. $\mathbf{A} \in \mathbb{R}^{n \times n}$ avec $n > 100$) d'utiliser des méthodes itératives. Deux des méthodes les plus utilisées sont :

1. la méthode de descente de gradient
2. la méthode du gradient conjugué.

Le but ce premier exercice est d'implémenter ces deux méthodes et de comparer leurs performances. En vous basant sur le code python fourni, votre travail consiste à :

1. Coder la fonction python évaluant $f(\mathbf{x})$ et son gradient $\nabla f(\mathbf{x})$.
2. Coder la méthode de la descente de gradient.
3. Coder la méthode du gradient conjugué.

En utilisant les différentes courbes sorties par le code, discutez des avantages et inconvénients des deux méthodes en terme de vitesse de convergence, temps de calcul, nombre d'itérations, du choix du pas pour la descente de gradient, etc.

Exercice 2

Intéressons nous maintenant à un exemple plus concret de minimisation convexe. Une entreprise souhaite ouvrir un nouvel entrepôt à partir duquel elle sera amenée à livrer son produit dans les 8 villes listées dans le tableau 1 et représentées sur la figure 2. L'objectif est donc de trouver où placer l'entrepôt de sorte à minimiser la distance parcourue par les camions de livraison. Un tel problème pratique peut être formulé comme un problème de minimisation convexe et résolu à l'aide des méthodes développées dans l'exercice précédent.

	Lille	Orléans	Lyon	Paris	Marseille	Strasbourg	Rennes	Metz
Nord-Sud	197	-105	-343	0	-617	-31	-88	30
Est-Ouest	52	-33	185	0	225	403	-300	285

Table 1: Liste des villes clientes de l'entreprise. Les distances nord-sud et est-ouest (en km) sont données en utilisant Paris comme point de référence.

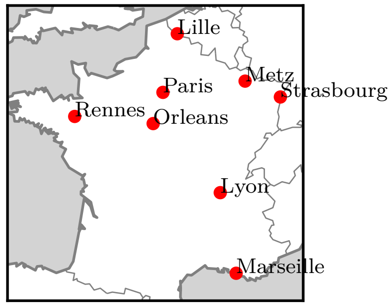

Figure 2: Carte des villes de France devant être livrées et conditionnant l'emplacement du nouvel entrepôt.

Question n°1

En définissant $\mathbf{x}_e = \begin{bmatrix} x_e & y_e \end{bmatrix}^T$ comme étant la position optimale de l'entrepôt par rapport au centre de Paris, montrez que trouver cette position $\mathbf{x}_e$ revient à minimiser une fonctionelle quadratique du type

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b} + c$$

pour laquelle vous déterminerez l'expression des termes $\mathbf{A}$, $\mathbf{b}$ et c.

Question n°2

En utilisant le code développé lors de l'exercice 1, déterminez la position optimale de l'entrepôt. Cette position vous semble-t-elle raisonnable ?

Exercice 3

Les deux exercices précédents avaient pour but de vous familiariser avec les méthodes de minimisation convexe en utilisant des problèmes pour lesquels il était possible de visualiser directement la fonction à minimiser. Intéressons nous maintenant à un problème en apparence plus complexe, pour lequel on ne peut visualiser $f(\mathbf{x})$, mais pouvant être résolu tout aussi efficacement que précédement.

Une large classe de problèmes pratiques peuvent être mis sous la forme

$$\mathbf{y} = \mathbf{M}\mathbf{x}. \tag{5}$$

où $\mathbf{M} \in \mathbb{R}^{m \times n}$, $\mathbf{y} \in \mathbb{R}^m$ et $\mathbf{x} \in \mathbb{R}^n$ avec $m \geq n$. Malheureusement, $\mathbf{M}$ étant une matrice $m \times n$ (avec $m \geq n$), ce problème n'admet pas (en général) de solution exacte. Le mieux que l'on puisse faire est donc de trouver une solution approchée $\mathbf{x}_s$ de sorte à ce que l'erreur

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{M}\mathbf{x} - \mathbf{y}\|_2^2 \tag{6}$$

soit minimale pour $\mathbf{x} = \mathbf{x}_s$.

Exemple Soit une plaque carrée de côté 1 ($-0.5 \leq x \leq 0.5$ et $-0.5 \leq y \leq 0.5$) pour laquelle vous êtes capable de mesurer la température T_i en dix points (x_i, y_i). Votre objectif est alors, à partir de ces dix points de mesure, d'estimer quelle est la distribution de température $T(x,y)$ sur toute la plaque. Pour le cas présent, il est raisonnable de supposer que le vrai champ de température peut être approximé par

$$T(x,y) \simeq \theta_0 + \theta_1 x + \theta_2 y + \theta_3 x^2 + \theta_4 xy + \theta_5 y^2, \tag{7}$$

où les coefficients θ sont à estimer à partir des mesures effectuées. Le modèle ci-dessus peut être ré-écrit sous la forme

$$T_i \simeq \begin{bmatrix} 1 & x_i & y_i & x_i^2 & x_i y_i & y_i^2 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} \tag{8}$$

En se basant sur les dix mesures effectuées, le vecteur $\boldsymbol{\theta}$ peut alors être estimé en minimisant la fonctionnelle

$$f(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{T}\|_2^2 \tag{9}$$

où $\mathbf{X} \in \mathbb{R}^{10\times 6}$, $\mathbf{T} \in \mathbb{R}^{10}$ et $\boldsymbol{\theta} \in \mathbb{R}^6$.

Question n°1

Montrez que $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{M}\mathbf{x} - \mathbf{y}\|_2^2$ peut être ré-écrite sous la forme d'une fonctionnelle quadratique

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{b} + c \tag{10}$$

pour laquelle vous donnerez les expressions de $\mathbf{A}$, $\mathbf{b}$ et c en fonction de $\mathbf{M}$ et $\mathbf{y}$.

Question n°2

Montrez que la solution $\mathbf{x}$ qui minimise $f(\mathbf{x})$ est donnée par

$$\mathbf{x} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{y} \tag{11}$$

où $\mathbf{M}^\dagger = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$ est appelée *pseudoinverse de Moore-Penrose*.

Question n°3

Pour les différentes paires $(\mathbf{M}, \mathbf{y})$ qui vous sont fournies, calculer la solution optimale $\mathbf{x}$ des trois manières suivantes :

1. en calculant explicitement la pseudoinverse de Moore-Penrose via la fonction numpy.linalg.inv,
2. en utilisant la méthode de descente de gradient pour laquelle vous déterminerez à chaque fois le pas fixe optimal par différents essais,
3. en utilisant la méthode du gradient conjugué.

Tout comme précédemment, comparez les trois méthodes en terme de temps de calcul et/ou de nombre d'itérations en fonction des tailles m et n de la matrice $\mathbf{M}$ considérée.