Ma 321 - Optimisation quadratique

Projet 1 - Minimisation des fonctionnelles quadratiques par des méthodes à directions de descente

Aéro 3 - Tronc commun

Igor Ciril
igor.ciril@ipsa.fr

Institut Polytechnique des Sciences Avancées - I.P.S.A.
63 boulevard de Brandebourg 94200 Ivry-sur-Seine

Février 2021

Contents

1 Introduction

Ce projet (accompagné d'une partie numérique traitée en Travaux Pratiques) poursuit deux objectifs. Le premier consiste à étudier (au sens des mathématiques) un problème d'analyse de données (plus précisément un problème de calibrage ou identification de paramètres d'un modèle) posé au sens des moindres carrés[1]. Cette étude fait l'objet de la section 2. Afin de résoudre numériquement le problème, nous allons considérer trois algorithmes à direction de descente : deux du type à direction de plus forte pente (l'un à pas fixe et l'autre à pas optimal), et le troisième basé sur les directions dites des gradients conjugués (algorithme des gradients conjugués). Le second objectif consiste à présenter, étudier puis comparer théoriquement et numériquement ces trois algorithmes à directions de descentes. La présentation et l'étude numérique de ces trois méthodes feront l'objet de la section 3, tandis que la sections 4 sera consacré à l'étude théorique (mathématiques) des méthodes. On pourra ainsi mettre en relation les résultats numériques avec les développements théoriques.

2 Formulation et analyse mathématique (75 pts)

2.1 Identification au sens des moindres carrés linéaires (15 pts)

La finalité de cette sous-section est d'expliciter le problème d'optimisation quadratique sans contrainte découlant de l'application de l'approche aux moindres carrés dans l'identification de paramètres d'un modèle (Linéaire vis-à-vis de ces paramètres *à déterminer*).

2.1.1 Principe général

Il s'agit d'un problème d'analyse de données. Supposons qu'au cours d'une expérience physique on mesure une grandeur ou quantité q qui dépend d'un paramètre p. On fait m expériences et mesures différentes en faisant varier le paramètre m. Le problème consiste alors à savoir comment déterminer à partir de ses m mesures une loi exprimentale (plutôt simple) qui permettrait d'approcher *au mieux* (le *mieux* est à définir) la quantité étudier comme une fonction du paramètre p. La forme de cette loi expérimentale est imposée : dans le cas le plus courant, on considère une fonction $f(p;\mathbf{c})$ d'une variable p, et dépendant de n paramètres inconnus $c_1, c_2, \dots c_n$ (constituant le vecteur $\mathbf{c}$). En général, le nombre de mesures m est très grand par rapport au nombre de n de paramètres. En d'autres termes, étant donné m valeurs $p_1, p_2, \dots, p_m$ du paramètre p (variable de la loi f) et m valeurs correspondantes de la grandeur mesuré $q_1, q_2, \dots, q_m$, on cherche à déterminer une fonction $f(p;\mathbf{c}^*)$ de la famille de fonction $f(p;\mathbf{c})$ indéxée par les paramètres $c_1, c_2, \dots c_n$ qui minimise l'erreur commise entre la valeur expérimentale q_i et la valeur théorique $f(p_i;\mathbf{c})$. On décide de mesurer l'erreur **au sens des moindres carrés**, *c.-à-d.* qu'on minimise la somme des carrés des erreurs individuelles, autrement dit la quantité

$$\forall (c_1, \dots c_n) \in \mathbb{R}^n, \quad E(c_1, \dots c_n) = \sum_{i=1}^{m} \left(q_i - f(p_i; c_1, \dots, c_n)\right)^2 . \tag{1}$$

On parlera de **régression linéaire** dans le cas où la loi $f(p; c_1, \dots, c_n)$ dépend linéairement des n paramètres $c_1, c_2, \dots, c_n$, en d'autres termes on écrit $f(p; c_1, \dots, c_n)$ s'écrit dans une base $\left(\Phi_j\right)_{j=0}^{n}$ de

[1] La méthode des moindres carrés, indépendamment élaborée par Legendre et Gauss au début du XIX^e siècle, permet de comparer des données expérimentales, généralement entachées d'erreurs de mesure, à un modèle mathématique censé décrire ces données.

fonctions (par exemple polynômes, fractions rationnelles, polynômes trigonométriques, exponentielle, etc.) par

$$f(p; c_1, \dots, c_n) = \sum_{j=1}^{n} c_j \Phi_j(p). \tag{2}$$

Son extrême simplicité fait que cette méthode est très couramment utilisée de nos jours en sciences expérimentales. Une application courante est le lissage des données expérimentales par une fonction empirique (fonction linéaire, polynômes ou splines). Cependant, son usage le plus important est probablement la mesure de quantités physiques à partir de données expérimentales.

2.1.2 Ajustement linéaire

Nous allons traîter dans ce projet l'approche simple (la plus simple) de l'ajustement d'une loi expérimentale par régression linéaire : **l'ajustement affine**.

Pour illustrer nos propos, on considère l'archive `ex1tp1.zip` (voir Claroline: Ma33sp ↦ Documents ↦ TP) des données des hauteurs d'un échantillon de $m = 50$ enfants entre 2 et 8 ans. L'archive contient:

- `dataP.dat`: fichier des âges des enfants;
- `dataQ.dat`: fichier des hauteurs des enfants.

1. (1 pt) Lire les fichiers des données et sauvegarder les âges dans le vecteur p et les hauteurs dans le vecteur q. **Fonction de la librairie *numpy* :** *numpy.loadtxt load*. Puis, tracer dans un graphique les couples $(p_i, q_i) \quad i = 1, \dots, m$.

Dans ce cas, on choisit comme modèle affine (relativement aux paramètres) pour **approximer** la tendance des données:

$$\forall p \in \mathbb{R}, \quad f(p; c_1, c_2) := c_1 + c_2 p, \tag{3}$$

où p est la variable indépendante qui répresente l'âge des enfants, q_i est la variable dépendante (c'est à dire, la hauteur) et c_1 et c_2 sont deux paramètres à ajuster (ou identifier) en utilisant les données. De plus, entre la valeur expérimentale q_i et la valeur théorique $f(p_i; c_1, c_2)$, pour un couple de paramètres (inconnus) (c_1, c_2) on peut écrire les m équations :

$$q_i = c_1 + c_2 p_i + r_i \quad , \quad i = 1, \dots, m \tag{4}$$

où r_i sont les résidus du modèle (3) relativement aux paramètres (inconnus) c_1 et c_2. Ces résidus représentent les erreurs induites par les instruments de mesure ainsi que par le modèle (3). Dans la suite, pour alléger les écriture on notera juste r_i le résidu $r_i(c_1, c_2)$.

À partir des données $(p_1, q_1), (p_2, q_2), \dots, (p_m, q_m)$, on cherche à estimer (identifier) au sens des moindres carrés les paramètres inconnus c_1 et c_2 du modèle (3)

2. (2 pt) Donner une formulation du problème dans laquelle on fait référence à la famille de fonctions affines (3) indexée par les paramètres c_1 et c_2. C'est de cette formulation du problème que l'on tire l'appellation : ajustement linéaire.

3. (1 pt) Vérifier que le système d'équations (4) se réécrit sous la forme matricielle

$$\mathbf{q} = X\mathbf{c} + \mathbf{r} \tag{5}$$

$$\mathbf{q} = \begin{pmatrix} q_1 \\ \vdots \\ q_m \end{pmatrix} \in \mathbb{R}^m \quad , \quad X = \begin{pmatrix} 1 & p_1 \\ \vdots & \vdots \\ 1 & p_m \end{pmatrix} \in \mathbb{R}^{m \times 2} \quad , \quad \mathbf{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_m \end{pmatrix} \quad , \quad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \in \mathbb{R}^2$$

4. (5 pts) Montrer que poser au sens des moindres carrés le problème d'identification des paramètres c_1 et c_2 du modèle (3) consiste à résoudre le problème de minimisation sans contrainte :

$$\begin{cases} \text{Minimiser} \quad F(\mathbf{c}) \\ \mathbf{c} \in \mathbb{R}^2 \end{cases}, \tag{Q}$$

où F est donnée par l'expression matricielle (relativement au repère canonique de $\mathbb{R}^2$, noté par la suite $\mathcal{E}_0 := (O, \mathcal{E})$ où $\mathcal{E} = (\mathbf{e}_1, \mathbf{e}_2)$) :

$$\forall \mathbf{c} \in \mathbb{R}^2, \quad F(\mathbf{c}) = \frac{1}{2}\mathbf{c}^T \left(X^T X\right)\mathbf{c} - \left(X^T \mathbf{q}\right)^T \mathbf{c} + \frac{1}{2}\|\mathbf{q}\|^2. \tag{6}$$

5. (3 pt) Vérifier que :

$$X^T X := \begin{pmatrix} m & \mathbf{p}^T\mathbf{1} \\ \mathbf{p}^T\mathbf{1} & \|\mathbf{p}\|^2 \end{pmatrix} \quad , \quad X^T\mathbf{q} := \begin{pmatrix} \mathbf{q}^T\mathbf{1} \\ \mathbf{p}^T\mathbf{q} \end{pmatrix} \quad \text{où} \quad \mathbf{1} := \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^m \quad , \quad \mathbf{p} = \begin{pmatrix} p_1 \\ \vdots \\ p_m \end{pmatrix} \in \mathbb{R}^m \tag{7}$$

puis calculer $X^T X$ et $X^T\mathbf{q}$ à l'aide des données numériques `dataP.dat` et `dataQ.dat`.

6. (3 pts) La fonction F est-elle quadratique ? (justifier votre réponse)

2.2 Le problème (Q) est-il bien poser ? (60 pts)

La finalité de cette sous-section est de répondre à une question mathématique fondamentale que l'on se pose concernant tout problème : le problème de minimisation (Q) est-il bien posé au sens d'Hadamard[2]? En d'autres termes, nous allons tout d'abord chercher à savoir si le problème admet une solution et si elle est unique. Puis nous allons étudier la sensibilité de la solution (si il y a bien unicité) du problème (Q) vis-à-vis de la perturbation des données (ici les matrices $X^T X$ et $X^T\mathbf{q}$).

2.2.1 Autour de la fonction quadratique F (44 pts)

Avant d'établir si le problème (Q) est bien posé ou non, nous allons étudier la fonction F. Pour cela nous allons traiter des points ci-dessous :

- Établir quelques propriétés élémentaires de la fonction (définie positivité, coercivité et stricte convexité sur $\mathbb{R}^2$, caractérisation de son unique point critique);
- Étudier ses fonctions partielles (expression générale, stricte convexité sur $\mathbb{R}$, explicité son unique minimiseur global stricte, tracé la courbe représentative);
- Expliciter et tracer la carte de niveaux (déterminer la nature géométrique des lignes de niveau);
- Expliciter et tracer la surface représentative (déterminer sa nature géométrique).

[2]Jacques Salomon Hadamard, né le 8 décembre 1865 à Versailles et mort le 17 octobre 1963 à Paris, est un mathématicien français, connu pour ses travaux en théorie des nombres, en analyse complexe, en analyse fonctionnelle, en géométrie différentielle et en théorie des équations aux dérivées partielles.

Cette étude nécessite en amont le calcul des vecteurs et valeurs propres de la matrice X^TX (Matrice définissant la partie quadratique *pure* de F). C'est l'objet de la première question.

1. Couples propres et conditionnement de X^TX (4 pts)
   L'objectif de cette question est de calculer numériquement (sous Python) les couples propres, et ainsi en déduire le conditionnement (associé à la norme euclidienne) de la matrice X^TX.

   Pour cela, on désigne par :

   - λ_1 et λ_2 les deux valeurs propres, rangées dans l'ordre croissant ($\lambda_1 < \lambda_2$), de la matrice X^TX, et par $\mathbf{v}_1$ et $\mathbf{v}_2$ les deux vecteurs propres (unitaire) associés respectivement à λ_1 et λ_2;
   - $\mathcal{V}$ la base orthonormée formée des vecteurs propres $\mathbf{v}_1$ et $\mathbf{v}_2$, et par P la matrice de passage de la base canonique $\mathcal{E}$ vers la base orthonormée $\mathcal{V}$.
   - $\text{cond}_2(X^TX)$ le conditionnement de norme 2 (ou enclidienne) de la matrice X^TX.

   (a) À l'aide de la librairie *numpy*, calculer numériquement les couples propres $(\lambda_1, \mathbf{v}_1)$ et $(\lambda_2, \mathbf{v}_2)$ (**Utiliser le lien :**WikiMath-Éléments propres).

   (b) Déduire de la question 1.(a) le conditionnement de la matrice X^TX (**Utiliser le lien :**Norme matricielle et conditionnement). Comparer ce résultat à celui obtenu en calculant à l'aide de la librairie *numpy* le conditionnement de X^TX (**Fonction :** *numpy.linalg.cond*).

2. Quelques propriétés élémentaires de F (4 pts)
   L'objectif de cette question est d'établir quelques propriétés élémentaires de la fonction F telles que : 1) Sa définie positivité; 2) Étudier sa coercivité et sa convexité; 3) Établir l'unicité du point critique.

   (a) Montrer que la fonction quadratique F est définie positive.

   (b) Que peux-t-on dire de la coercivité et de la convexité de F sur $\mathbb{R}^2$? (Justifier votre réponse)

   (c) Montrer que la fonction quadratique F admet un unique point critique, notée $\mathbf{c}^*$, qui est l'unique solution des équations normales.

$$(X^TX)\mathbf{c} = X^T\mathbf{q}. \tag{8}$$

3. Fonctions partielles de F (10 pts)
   L'objectif de cette question est d'étudier une fonction partielle quelconque de F puis de tracer sa courbe représentative. Plus précisément, nous allons étudier : 1) sa définie positivité; 2) sa coercivité et convexité; 3) l'existence d'un minimiseur global stricte. Tout cela nous permettra de donner le tracé le plus précis de sa courbe représentative.

   On considère un point quelconque de représentant matricielle $\mathbf{a}$ dans le repère canonique $\mathcal{E}_0$, et un vecteur unitaire de représentant matricielle $\mathbf{d}$ dans la base canonique $\mathcal{E}$.

   (a) Donner l'expression analytique de la fonction partielle, notée $F_{a,d}$, de F en $\mathbf{a}$ suivant $\mathbf{d}$.

   (b) Montrer que $F_{a,d}$ est une fonction quadratique définie postive d'une variable.

   (c) Que peux-t-on dire de la coercivité et de la convexité de $F_{a,d}$ sur $\mathbb{R}$? (Justifier votre réponse)

   (d) Expliciter l'unique minimiseur globale strict de F. On le notera $t^*_{a,d}$.

(e) Préciser la nature géométrique de la courbe représentative de $F_{a,d}$.

(f) Tracer, sur un intervalle $[-10, 10]$ les fonctions partielles $F_{c^*,d}$ pour $\mathbf{d} := \mathbf{e}_1, \mathbf{e}_2, \mathbf{v}_1$ et $\mathbf{v}_2$. (**Utiliser le lien :**Tracé de courbes)

4. Carte de niveau (18 pts)
L'objectif de cette question est d'expliciter et de déterminer la nature géométrique des lignes de niveau constituant la carte de niveaux de F. Ceci nous permettra aussi de donner une interprétation géométrique du conditionnement (associé à la norme 2) de la matrice X^TX.

(a) Construire la matrice $Z = X^TX = \left(Z_{ij}\right)_{1\leq i,j\leq 2} \in \mathbb{R}^{2\times 2}$, le vecteur $\mathbf{w} = X^T\mathbf{q} = (w_1, w_2)^T \in \mathbb{R}^2$ et le scalaire $s = \mathbf{q}^T\mathbf{q} \in \mathbb{R}$. En déduire que la fonctionnelle (6) peut s'écrire sous la forme analytique suivante :

$$\forall \mathbf{c} := \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \quad F(\mathbf{c}) := F(c_1, c_2) = \frac{1}{2}\Big(Z_{1,1}c_1^2 + 2Z_{1,2}c_1c_2 + Z_{2,2}c_2^2 - 2(w_1c_1 + w_2c_2) + s\Big) \qquad (9)$$

(b) On considère un pas de discrétisation $\delta = 0.5$. Définir le pavé $[-10, 10] \times [-10, 10]$ avec un pas δ dans chaque direction (**Fonction :** *numpy.meshgrid*). Puis, afficher les courbes de niveau de F dans le pavé $[-10, 10] \times [-10, 10]$ (**Avec Matplotlib utiliser :** *contour*).

(c) Donner la forme réduite (ou canonique), notée $\mathcal{F}$, de la fonction quadratique F en précisant le repère de réduction, noté $\mathcal{V}^*$.

(d) Expliciter[3] la carte de niveaux de la fonction F.

(e) Les résultats obtenus par le calcul à la question 4.(d) coincident-ils avec le tracé des courbes de niveau de la question 4. (b).

(f) Donner une caractérisation géométrique (à partir des lignes de niveau de F) du conditionnement de la matrice X^TX.

5. Surface Représentative (8 pts)
L'objectif de cette question est de reconnaître la nature géométrique de la surface représentative de F d'en donner ses caractéristiques.

(a) Tracer la fonction F dans le pavé $[-10, 10] \times [-10, 10]$. (**Voir l'usage de** *plot_surface* dans la documentation de **Matplotlib**).

(b) Precisez les caractéritiques géométrique de la surface représentative, notée S_F, de F.

(c) Pourquoi dit-on que S_F est un paraboloïde elliptique ? (justifier le plus rigoureusement possible votre réponde)

2.2.2 Unicité de la solution de (Q) (7 pts)

Comme son titre l'indique, la finalité de ce paragrahe est d'établir l'unicité de la solution du problème (Q). Le problème étant de dimension 2, nous donneront une forme explicite de la solution.

6. (2 pts) Montrer que l'unique solution du problème d'optimisation est l'unique point critique $\mathbf{c}^*$ (son représentant relativement à le repère canonique $\mathcal{E}_0$) de F.

[3] On cherche à expliciter l'expression cartesienne des lignes de niveaux de F et ainsi en déduire leur nature géométrique

7. (1 pt) En utilisant la question précédente, préciser le problème équivalent à notre problème de minimisation quadratique définie positive (Q).

8. (4 pts) Dans cette question nous allons expliciter $\mathbf{c}^*$, le représentant dans le repère canonique $\mathcal{E}_0$ de l'unique solution de (Q).

   (a) Expliciter $\mathbf{c}^*$ en fonction des vecteurs $\mathbf{q}$, $\mathbf{p}$ et $\mathbf{1}$.

   (b) Calculer numériquement $\mathbf{c}^*$ à partir de son expression établie à la question précédente.

2.2.3 Conditionnement du problème (8 pts)

Dans ce paragraphe, nous allons cherché estimer les erreurs commises sur la solution $\mathbf{x}^*$ du problème d'optimsiation quadratique sans contrainte (Q) en fonction des erreurs commises sur les données du problème, *c.-à-d.* la matrice X^TX et le vecteur $X^T\mathbf{q}$ définissant la fonction F. Comme nous avons établi à la sous-section 2.2.2 que le problème de minimisation (Q) et le problème de résolution du système linéaire inversible (8) sont équivalent (au sens fort d'avoir la même solution, qui est de plus unique), notre étude correspond à l'étude de la sensibilité aux données d'un système linéaire carré inversible Il faut noter que cette problématique a déjà été traité au premier semestre dans le cadre de l'enseignement Ma 313 *Algèbre linéaire numérique*.

Avant de poursuivre dans le vif du sujet, on peut se poser la question de l'utilité et de l'impact de cette étude. En effet, plus que la résolution théorique des problèmes équivalents (Q) et (8) c'est leur résolution pratique sur ordinateur (résolution numérique) qui nous interresse ici en finalité (c'est notamment l'objectif principal d'un ingénieur!). Vous avez déjà vu précédement (autres enseignements d'analyse numérique comme Ma 123, Ma 223, Ma 313) que les algorithmes de résolution doivent être efficaces, *c.-à-d.* être rapide (en minimisant le nombre d'opération élémentaires, d'itération ou de temps CPU) et nécessiter peu de place mémoire. Par ailleur, il existe une autre exigence pratique pour les méthodes numériques : leur précision. En effet, limité à cause du nombre de bits utilisés pour représenter les nombres réels : d'habitude 32 ou 64 bits (ce qui fait à peu près 9 ou 16 chiffres significatifs). Il faut donc faire très attention aux inévitables erreurs d'arrondi et à leur propagation au cours d'un calcul. Dans le contexte de notre problème, la matrice X^TX et le vecteur $X^T\mathbf{q}$ sont connue à une erreur près et les trois méthodes considérées sont suceptibles de propager ses erreurs. On aimerait que l'erreur commise sur les données du problème (ici la matrice X^TX et le vecteur $X^T\mathbf{q}$) n'ait pas une conséquence trop grave sur le calcul de la solution du problème (ici $\mathbf{c}^*$ les coefficients du modèle linéaire (3)). Si par exemple 1% d'erreur sur les coefficients de X^TX et$X^T\mathbf{q}$ entraîne 100% d'erreur sur $\mathbf{c}^*$, le modèle linéaire (3) obtenu ne sera pas d'une utilité redoutable. Dans ce contexte, Il faut s'assurer qu'une méthode de résolution (notamment les trois méthodes étudiées dans ce projet) ne favorise pas une telle amplification : c'est ce qu'on appelle la stabilité d'une méthode. Cette question est d'autant plus importante pour les méthodes itérative qui ne calcule pas comme les méthodes directes une solution exacte (au sens de l'arithmétique parfaite sans arrondi) mais une suite de solutions approchées qui converge vers la solution exacte. Les méthodes qui seront étudier par la suite sont itératives. Dans la suite de ce projet, au niveau du paragraphe 3.1.3 (quastions 6 et 7) étudirons l'impacte des erreurs d'arrondi sur la précision des résultats numériques obtenus.

Pour quantifier le phénomène des erreurs d'arrondi, on considère la notion de conditionnement de la matrice symétrique définie positive X^TX (**Utiliser le lien :**Norme matricielle et conditionnement).

Au niveau de ce paragraphe, nous allons nous contenter d'estimer la sensibilité de la solution lorsque l'on perturbe (juste) le second membre $X^T\mathbf{q}$ du problème (8). Plus précisément, nous allons

établir une majoration qui estime la sensibilité relative de la solution du problème en fonction de la perturbation relative du vecteur $X^T\mathbf{q}$. Nous en déduirons que la sensibilité de l'erreur relative de la solution de (Q) est estimée par le conditionnement (en norme euclidienne) de la matrice X^TX.

9. (8 pts) La finalité de cette questions est d'évaluer l'impact du conditionnement de la matrice X^TX sur la sensibilité de la solution de (Q). On se contente de perturbé le second membre du système (8). Pour cela, on désigne par :

   - $\delta\left(X^T\mathbf{q}\right)$ une perturbation du second membre (donnée) $X^T\mathbf{q}$ du problème (8);
   - $\mathbf{c}^* + \delta\mathbf{c}^*$ l'unique solution du système linéaire *perturbé* inversible $X^TX\mathbf{c} = X^T\mathbf{q} + \delta\left(X^T\mathbf{q}\right)$.

   Dans cette question, nous allons tout d'abord établir la majoration suivante :

$$\frac{\|\delta\mathbf{c}^*\|}{\|\mathbf{c}^*\|} \leq \text{cond}_2(X^TX)\,\frac{\left\|\delta\left(X^T\mathbf{q}\right)\right\|}{\left\|X^T\mathbf{q}\right\|}, \tag{10}$$

   (a) Montrer que l'on a :

$$\|\delta\mathbf{c}^*\| \leq \left\|(X^TX)^{-1}\right\|\,\left\|\delta\left(X^T\mathbf{q}\right)\right\|. \tag{11}$$

   où la quantité

$$\left\|(X^TX)^{-1}\right\| = \sup_{\mathbf{x}\neq 0}\frac{\left\|(X^TX)^{-1}\mathbf{x}\right\|}{\|\mathbf{x}\|} \tag{12}$$

   désigne la norme matricielle subordonnée à la norme euclidienne ou norme 2.

   (b) En utilisant la majoration (11), établir la majoration (10).

   (c) À partir de (10), Que pouvez vous conclure sur la sensibilté (conditionnement) du problème (Q).

3 Méthodes à directions de descente : une étude numérique (125 pts)

3.1 Principes généraux (33 pts)

Dans ce sous-section, nous allons construire le schéma général de l'itération de la classe importante d'algorithmes de résolution d'un problème d'optimisation sans contrainte : les méthodes à direction de descente. Ces méthode d'applique sur des fonctions suffisamment régulières. C'est l'objet de l'enseignement de Filière système Ma 324 *Une introduction à l'optimisation différentiable*.

Dans le contexte de ce problème, on considère le problème *général* d'optimisation quadratique définie positif sans contrainte de dimension 2 :

$$\begin{cases}\text{Minimiser}\ \ f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^TA\mathbf{x} - \mathbf{b}^T\mathbf{x} + c\\ \mathbf{x} := \begin{pmatrix}x_1\\x_2\end{pmatrix} \in \mathbb{R}^2\end{cases}, \tag{G}$$

où

- $A \in \mathcal{M}_2(\mathbb{R})$ symétrique définie positive dont les deux valeurs propres sont : $0 < \lambda_1(A) \leq \lambda_2(A)$;

- $\mathbf{b} := \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \in \mathbb{R}^2$;
- $c \in \mathbb{R}$.

Nous allons maintenant établir le schéma général de la $k-$ième itération de la classe des méthodes à directions de descente associée à la résolution du problème *général* (G). Nous supposons au début de l'itération k que l'on dipose d'un itéré noté $\mathbf{x}_k$. Nous allons maintenant montrer comment on calcul l'itéré suivant, noté bien évidement $\mathbf{x}_{k+1}$.

3.1.1 Direction de descente (4 pts)

Comme son nom l'indique, on s'intéresse à une classe d'algorithmes fondée sur la notion de **direction de descente**.

1. (1 pts) Rappeler la définition d'une direction de descente de la fonction f au point $\mathbf{x}_k$.
2. (3 pts) Faire une figure (même à main levée) dans le plan de représentation des variables x_1 et x_2 : 1) représentant les courbes de niveaux passant par le point $\mathbf{x}_k$ respective de la fonction f et de son approximation tangentielle linéaire T_{f,x_k} au point $\mathbf{x}_k$; 2) représenter le vecteur $\nabla f(\mathbf{x}_k)$; 3) colorier ou hachurer en rouge le demi-espace contenant les directions de descente de f au point $\mathbf{x}_k$.

Dans la suite de projet nous allons considérer deux choix de direction de descente : 1) descente du gradient ou de la plus profonde descente qui consiste à poser $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$; 2) descente du gradient conjugué qui sera une correction de la direction de plus profonde descente qui tiendra compte au travers de la matrice A de la géométrie du problème (courbure directionnelle au travers de la différentielle seconde).

3.1.2 Pas de descente (20 pts)

On déduit du shcéma établi à la question précédente, qu'il existe une infinité de choix de direction de descente. Dès lors, une fois choisie la façon de calculer la direction de descente $\mathbf{d}_k$ à l'itéré $\mathbf{x}_k$, il apparaît que l'on peut *descendre* ou faire décroître la fonction f à partir de $\mathbf{x}_k$ suivant cette direction $\mathbf{d}_k$. En d'autres termes il existe α_k tel que :

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) < f(\mathbf{x}_k). \tag{13}$$

On dira alors que α_k est un **pas de descente** de f en $\mathbf{x}_k$ suivant $\mathbf{d}_k$. Nous allons montrer qu'il existe une infinité de choix de pas de descente de la fonction f à partir du point $\mathbf{x}_k$ suivant $\mathbf{d}_k$.

3. (10 pts) Dans cette question nous allons montrer le lemme ci-dessous qui justifie l'existence d'une infinité de pas de descentes pour une direction de descente calculée.

   Proposition 3.1 (CNS d'existence d'un pas de descente) *Soit $\mathbf{d}_k \neq \mathbf{0}$ une direction de descente de f au point $\mathbf{x}_k$. Alors tout réel $\alpha \in]0, \tilde{\alpha}_k[$, où*

   $$\tilde{\alpha}_k := -\frac{2\left(A\mathbf{x}_k - \mathbf{b}\right)^T \mathbf{d}_k}{\mathbf{d}_k^T A \mathbf{d}_k}, \tag{14}$$

   est un pas de descente de F en $\mathbf{x}_k$ suivant $\mathbf{d}_k$.

(a) L'objectif est de réaliser deux figures qui permettent d'interpréter le résultat du lemme 3.1. Dans ce contexte faite les deux figures suivantes :

- Figure 1 - Dans le plan de représentation des variables x_1 et x_2: 1) tracer les courbes de niveaux passant par le point $\mathbf{x}_k$ respective de la fonction f et de son approximation tangentielle linéaire T_{f,x_k} au point $\mathbf{x}_k$; 2) représenter le vecteur $\nabla f(\mathbf{x}_k)$ et la direction de descente $\mathbf{d}_k$ de f en $\mathbf{x}_k$; 3) placer le point $\mathbf{x}_k + \tilde{\alpha}_k \mathbf{d}_k$; 4) Le point $\mathbf{x}_k + \alpha_k \mathbf{d}_k$ où α_k est un pas de descentes de f en $\mathbf{x}_k$ suivant $\mathbf{d}_k$.
- Figure 2 - Dans le plan de représentation d'une fonction d'une variable : 1) tracer les courbes représentatives de la fonction partielle f_{x_k,d_k} de f et la fonction partielle $\left(T_{f,x_k}\right)_{x_k,d_k}$; 2) représenter suivant l'axe des abscisses la dérivée directionnelle $D_{d_k} f(x_k)$; 3) placer le point de f d'abscisse $\tilde{\alpha}_k$; 4) placer le point de f d'abscisse α_k correspondant à un pas de descentes de f en $\mathbf{x}_k$ suivant $\mathbf{d}_k$.

(b) Rappeler (sans démonstration) l'expression de la fonction partielle f_{x_k,d_k} de f au point $\mathbf{x}_k$ dans la direction $\mathbf{d}_k$.

(c) Determiner le réel $\tilde{\alpha}_k > 0$ tel que $f_{x_k,d_k}(\tilde{\alpha}_k) = f_{x_k,d_k}(0)$.

(d) Déduire de la question précédente la proposition 3.1.

Ainsi il suffit de faire ce déplacement quantifié par le pas α_k le long $\mathbf{d}_k$ et obtenir le nouvelle itéré $\mathbf{x}_{k+1}$. Les méthodes à directions de descente utilisent cette stratégie pour *descendre* vers le minimiseur de la fonction f. Elle construise la suite des itérés $(\mathbf{x}_k)_{k\in\mathbb{N}}$ approchant l'unique solution du problème d'optimisation (G), notée $\mathbf{x}^*$, par la récurrence

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k. \tag{15}$$

Par conséquent pour définir une méthodes à directions de descnete il faut spécifier les deux choses :

- dire comment la direction de descente $\mathbf{d}_k$ est calculée; la manière de procéder donne le nom à l'algorithme;
- dire comment on détermine le pas de descente α_k; c'est ce qu'on appelle la **recherche linéaire**.

Au vu du lemme 3.1, il existe de très nombreuses stratégies de calculer à l'itéré $\mathbf{x}_k$ un pas de descente α_k le long d'une direction de descente $\mathbf{d}_k$. Dans ce projet nous allons considérer deux stratégies. La première va consister de garder le même pas à toutes les itérations, on parlera alors de méthode à direction de descentes à pas constant. La seconde, est bien plus naturelle. En effet, comme on cherche à minimiser f, il semble naturel à l'itéré $\mathbf{x}_k$ de chercher à minimiser la fonction objectif f le long de la direction de descente $\mathbf{d}_k$ et donc de déterminer le pas α_k comme solution du problème de minisation d'une variable réelle :

$$\begin{cases} \text{Minimiser} \quad f_{\mathbf{x}_k,\mathbf{d}_k}(\alpha),\ \alpha \in \mathbb{R} \\ \text{Sous la contrainte :} \\ \alpha > 0. \end{cases} \tag{16}$$

Dans ce cas α_k sera appelé pas optimal. Si cette stratégie de recherche linéaire est choisie, on parlera alors de méthode à direction de descente à pas optimal.

4. (10 pts) Dans cette question nous allons montrer que le pas optimal, noté α_k^*, de f en $\mathbf{x}_k$ suivant la direction de descente (non nulle) $\mathbf{d}_k$ est donnée par :

$$\alpha_k^* = -\frac{(A\mathbf{x}_k - \mathbf{b})^T \mathbf{d}_k}{\mathbf{d}_k^T A \mathbf{d}_k}. \tag{17}$$

   (a) Montrer que α_k^* défini par l'expression (17) est le minimiseur global stricte de f_{x_k,d_k}.

   (b) Montrer que $\alpha_k^* > 0$, et conclure.

   (c) Quel est la position du $(k+1)-$ième itéré $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k^* \mathbf{d}_k$ relativement au segment $[\mathbf{x}_k, \mathbf{x}_k + \tilde{\alpha}_k \mathbf{d}_k]$ ou $\tilde{\alpha}_k$ est défni par (14) (justifier rigoreusement vos propos en considérant les propriétés de la courbe représentative de la fonction partielle f_{x_k,d_k}).

   (d) Réaliser les figures de la question 3 item (a) dans le cas où α_k désigne le pas optimal de f en $\mathbf{x}_k$ suivant $\mathbf{d}_k$.

3.1.3 Critère d'arrêt (9 pts)

Au vu de la construction précédente (*c.f.* la formule (15) de calcul de $\mathbf{x}_{k+1}$ à partir de $\mathbf{x}_k$), les algorithmes à direction de descente sont des algorithmes de type itératif susceptible de *résoudre* le problème (G). Dans un monde idéal (*c.-à-d.* en supposant tous les calculs exacts et la capacité de calcul de la machine illimitée), soit l'algorithme produit une suite fini $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k$ qui identifie l'unique solution $\mathbf{x}^*$ à la $k-$ième iteration (*c.-à-d.* $\mathbf{x}_k = \mathbf{x}^*$), soit il construit (théoriquement) une suite infinie $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k, \ldots$ de points de $\mathbb{R}^2$ qui converge vers $\mathbf{x}^*$.

En pratique, il faut choisir un moyen d'arrêter l'algorithme. On parle alors de **test** ou **critère d'arrêt** de l'algorithme. Ce test d'arrêt devra être choisi pour garantir que l'algorithme s'arrête toujours après un nombre fini d'itérations et que le dernier point calculé soit *suffisamment* proche de $\mathbf{x}^*$. En d'autre terme, pour une précision fixée notée $\varepsilon > 0$, l'algorithme doit s'arrêter à une certaine étape (itération) k^* de manière que $\mathbf{x}_{k^*}$ soit une $\varepsilon-$ approximation de la solution $\mathbf{x}^*$, *c.-à-d.* $\|\mathbf{x}_{k^*} - \mathbf{x}^*\| \leq \varepsilon$. Cependant, il est claire qu'en pratique, étant donné que la solution $\mathbf{x}^*$ n'est pas connue, on ne peut pas évaluer la quantité $\|\mathbf{x}_{k^*} - \mathbf{x}^*\|$ et par conséquent considérer ce test d'arrêt. Il faut donc, choisir un critère d'arrêt que l'on peut calculer (évaluer) numériquement à chaque itération.

Pas conséquent, on modélise (mathématiquement) un critère d'arrêt par une fonction $\Delta : \mathbb{R}^N \to \mathbb{R}^+$

- pour laquelle on doit pouvoir l'évaluer à la $k-$ème itération, *c.-à-d.* calculer $\Delta(\mathbf{x}_k)$;

- qui garantit la précision de la solution, *c.-à-d.*

$$\Delta(\mathbf{x}_k) \leq \varepsilon \;\Rightarrow\; \|\mathbf{x}_k - \mathbf{x}^*\| \leq \varepsilon. \tag{18}$$

5. (3 pts) Dans cette question nous allons donner deux interprétations du critère dit du résidu.

   (a) Rappeler la définition du critère du résidu.

   (b) Donner les deux interprétations de ce critère : l'une liée à la résolution du problème de minimisation quadratique (G), l'autre associée au problème équivalent de résolution du système linéaire inversible $A\mathbf{x} = \mathbf{b}$.

On finit ce paragraphe, par une étude de la **qualité** du critère d'arrêt du résidu. Un critère d'arrêt Δ sera dit de *bonne* qualité si et seulement si il permet de garantir la précision ε fixé de la solution approchée, *c.-à-d.* si l'implication (18) soit vérifiée. Cette étude s'applique à l'ensemble des méthodes de types itératives succeptibles de résoudre le problème (G).

6. (3 pts) Soit $\mathbf{x}_k$ le $k-$ième itéré générée par l'algorithme itératif dont la $k-$ième itération est défini par (15). En réinterprétant la majoration (10) établit dans le cas particulier du problème d'optimisation quadratique sans contrainte (Q), montrer que :

$$\forall k \in \mathbb{N}, \quad \frac{\|\mathbf{x}_k - \mathbf{x}^*\|}{\|\mathbf{x}^*\|} \leq \text{cond}_2(A) \ \frac{\|\mathbf{r}_k\|}{\|\mathbf{b}\|}, \tag{19}$$

   où $\mathbf{r}_k$ désigne le résidu associé au système linéaire $A\mathbf{x} = \mathbf{b}$ de l'itéré $\mathbf{x}_k$.

7. (3 pts) À partir de l'inégalité (19), que peut-on dire sur la qualité du critère dans le contexte de la résolution du problème (G), et par conséquent sur celle de l'approximation calculée.

3.1.4 Schéma général

Nous pouvons maintenant décrire cette classe d'algortihme de manière présise.

Algorithme 3.1 (Méthode à directions de descente)

- ***Objectif*** *Calculer une approximation $\tilde{\mathbf{x}}$ de l'unique solution optimale $\mathbf{x}^*$ du problème de minimisation* (G).
- ***Entrée***
  – *Données sur le problème* (G) *: la matrice symétrique définie positive $A \in \mathcal{M}_2(\mathbb{R})$, le vecteur $\mathbf{b} \in \mathbb{R}^2$.*
  – *La précision $\varepsilon > 0$ associé au critère d'arrêt.*
- ***Sortie*** *Une $\epsilon-$approximation*[4] *$\tilde{\mathbf{x}}$ de $\mathbf{x}^*$.*
- ***Initialisation*** *$k = 0$.*
- ***Itération*** *On suppose qu'au début de l'itération k, on dispose d'un itéré $\mathbf{x}_k \in \mathbb{R}^2$.*
  1. *Test d'arrêt : Si $\|\mathbf{r}_k\| = \|A\mathbf{x}_k - \mathbf{b}\| \leq \varepsilon$, alors $\tilde{\mathbf{x}} := \mathbf{x}_k$.*
  2. *Calcul de la direction $\mathbf{d}_k$ de descente de f en $\mathbf{x}_k$:*
  $$\textit{déterminer } \mathbf{d}_k \textit{ tel que } \ D_{x_k} f(\mathbf{x}_k) = \nabla f(\mathbf{x}_k)^T \mathbf{d}_k = (A\mathbf{x}_k - \mathbf{b})^T \mathbf{d}_k < 0. \tag{20}$$
  3. *Calcul du pas α_k de descente de f au point $\mathbf{x}_k$ suivant $\mathbf{d}_k$:*
  $$\textit{déterminer } \alpha_k \textit{ tel que } \ f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) < f(\mathbf{x}_k). \tag{21}$$
  4. *Calcul du nouveau itéré $\mathbf{x}_{k+1}$:*
  $$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k. \tag{22}$$

Dans la suite de ce projet nous allons étudier 3 algorithmes : L'algorithme du gradient à pas fixe et pas optimal, et l'algorithme du gradient conjugué.

[4] $\tilde{\mathbf{x}}$ est une $\epsilon-$approximation de $\mathbf{x}^*$ si et seulement si $\|\tilde{\mathbf{x}} - \mathbf{x}^*\| \leq \epsilon$.

3.2 Méthode de descente du gradient ou de plus forte descente (37 pts)

Dans cette sous-section, dans le contexte de résolution du problème d'ajustement linéaire (Q), nous allons explicité, implémenter et tester numériquement les deux méthodes de descentes dite de plus forte descente suivante : 1) algorithme de descente du gradient à pas fixe; 2) algorithme de descente du gradient à pas optimal.

3.2.1 Direction de plus forte descente (5 pts)

Comme son nom l'indique, c'est la direction de descente $\mathbf{d}_k$ (vérifie (20)) qui indique au point courant $\mathbf{x}_k$ de l'algorithme 3.1 la direction de plus forte descente (pente négative) :

$$\forall \mathbf{d} \in \mathbb{R}^2 \text{ tel que } \|\mathbf{d}\| = \|\mathbf{d}_k\|, \quad \mathbf{d}_k^T \nabla f(\mathbf{x}_k) \leq \mathbf{d}^T \nabla f(\mathbf{x}_k). \tag{23}$$

1. (5 pts) Dans cette question nous allons montrer que la direction $\mathbf{d}_k := -\nabla f(\mathbf{x}_k)$ est une direction de plus forte descente de f au point $\mathbf{x}_k$.

   (a) Montrer que $\mathbf{d}_k := -\nabla f(\mathbf{x}_k)$ est bien une direction de descente de f au point $\mathbf{x}_k$.

   (b) Montrer que :

   $$\forall \mathbf{d} \in \mathbb{R}^2 \text{ tel que } \|\mathbf{d}\| = \| - \nabla f(\mathbf{x}_k)\|, \quad \mathbf{d}^T \nabla f(\mathbf{x}_k) \leq \|\nabla f(\mathbf{x}_k)\|^2 \tag{24}$$

   (c) Déduire de (24) l'inégalité (23).

3.2.2 L'algorithme de descente du gradient à pas fixe (21 pts)

Comme son nom l'indique, ses itérations sont définies par les choix suivants :

- Choix du calcul de la direction de descent : *méthode de gradient* signifie de choisir la direction de plus profonde descente à chaque itération ;
- Choix du calcul du pas de descente : *à pas fixe* signifie de choisir le même pas (constant) à chaque itération.

Nous allons expliciter la méthode dite de gradient à pas fixe dans le contexte de résolution du problème d'optimisation quadratique (Q).

2. (2 pt) L'algorithme l'algorithme du gradient à pas fixe s'écrit comme ci-dessous.

   Algorithme 3.2 (Méthode du gradient à pas fixe - une itération)
   On suppose qu'au début de l'itération k, on dispose d'un itéré $\mathbf{x}_k \in \mathbb{R}^2$.

   (a) Test d'arrêt : Si $\|\mathbf{r}_k\| = \|X^T X \mathbf{x}_k - X^T \mathbf{q}\| \leq \epsilon$, *alors* $\tilde{\mathbf{x}} := \mathbf{x}_k$.

   (b) Direction de descente : $\mathbf{d}_k := -\nabla F(\mathbf{x}_k) = -\left(X^T X \mathbf{x}_k - X^T \mathbf{q}\right) = -\mathbf{r}_k$.

   (c) Pas de descente : $\alpha_k := \alpha > 0$.

   (d) Nouveau itéré : $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{r}_k$

   Justifier les différentes étapes de algorithme 3.2.

Il apparaît clairement dans l'algorithme 3.2, qu'il est nécessaire de spécifier le choix du pas de descente fixe $\alpha_k := \alpha$ afin qu'à chaque itération l'algorithme fasse décroître la valeur de la fonction, *c.-à-d.* le pas fixe α vérifie pour toutes les itérations la condition (21). C'est l'objet de la question ci-dessous.

3. (5 pts) Dans cette question nous allons donner la preuve de la proposition ci-dessous qui permet de garantir que l'algorithme du gradient à pas fixe est bien une méthode de descente.

   Proposition 3.2 (CS d'xistence d'un pas fixe de descente)
   Soit α un réel vérifiant la condition :
   $$0 < \alpha < \frac{2}{\lambda_2}. \tag{25}$$
   Alors l'algorithme 3.2 à pas fixe α est bien une méthode à direction de descente.

   (a) Montrer que si le réel α vérifie la condition
   $$\forall k \in \mathbb{N}, \quad 0 < \alpha < \tilde{\alpha}_k := \frac{2\|\mathbf{r}_k\|^2}{\mathbf{r}_k^T X^T X \mathbf{r}_k}, \tag{26}$$
   alors l'algorithme 3.2 à pas fixe α est bien une méthode à direction de descente.

   (b) Montrer que
   $$\forall k \in \mathbb{N}, \quad \tilde{\alpha}_k := \frac{2\|\mathbf{r}_k\|^2}{\mathbf{r}_k^T X^T X \mathbf{r}_k} \geq \frac{2}{\lambda_2}. \tag{27}$$

   (c) Déduire de la question précédent la conclusion de la proposition 3.2.

Maintenant nous allons implémenter et tester l'algorithme 3.2.

4. (14 pts) Programmer une fonction *Python* `sol,xit,nit = gradientPasFixe(A,b,x0,rho,tol)` pour résoudre le problème $A\mathbf{x} = \mathbf{b}$ par la méthode du gradient à pas fixe (Algorithme 1). La fonction reçoit comme entrée la matrice A, le vecteur b, un point initial $\mathbf{x}_0$, un pas ρ et la tolerance `tol`. En sortie, on a la solution finale `sol`, la donnée des $x^{(i)}$ calculés à chaque itération `xit` et le nombre d'itérations `nit`.

   (a) Donner le code en langage python qui vous à permis de faire les tests numériques. Expliquer votre code le plus rigoureusement possible.

   (b) On fixe $\mathbf{x}_0 := \mathbf{c}_0 = (-9, -7)^T$ et un pas $\rho = 10^{-3}$. Résoudre le système linéaire $(X^T X)\mathbf{c} = X^T \mathbf{q}$ par la méthode du gradient à pas fixe considérée. Combien d'itérations sont nécessaires avec une tolérance fixéee à 10^{-6} ?

   (c) Qu'est-ce qu'on observe si on choisit $\rho = 10^{-1}$? Et si on prend $\rho = 10^{-5}$?

3.2.3 L'algorithme de descente du gradient à pas optimal (11 pts)

Comme son nom l'indique, ses itérations sont définies par les choix suivants :

- Choix du calcul de la direction de descent : *méthode de gradient* signifie de choisir la direction de plus profonde descente à chaque itération ;

- Choix du calcul du pas de descente : *à pas optimal* signifie de choisir à chaque au point courant $\mathbf{x}_k$ le pas α_k qui minimise la fonction f dans la direction de plus profonde descente (associée à $\mathbf{x}_k$).

Nous allons expliciter la méthode dite de gradient à pas optimal dans le contexte de résolution du problème d'optimisation quadratique (Q).

5. (2 pts) L'algorithme du gradient à pas optimal s'écrit comme ci-dessous.

Algorithme 3.3 (Méthode du gradient à pas optimal - une itération)
On suppose qu'au début de l'itération k, on dispose d'un itéré $\mathbf{x}_k \in \mathbb{R}^2$.

(a) Test d'arrêt : Si $\|\mathbf{r}_k\| = \|X^T X\mathbf{x}_k - X^T\mathbf{q}\| \leq \epsilon$, *alors* $\tilde{\mathbf{x}} := \mathbf{x}_k$.

(b) Direction de descente : $\mathbf{d}_k := -\nabla F(\mathbf{x}_k) = -\left(X^T X\mathbf{x}_k - X^T\mathbf{q}\right) = -\mathbf{r}_k$.

(c) Pas de descente : $\alpha_k := \frac{\|\mathbf{r}_k\|^2}{\mathbf{r}_k^T\left(X^T X\right)\mathbf{r}_k}$.

(d) Nouveau itéré : $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\mathbf{r}_k$

En utilisant la question 4 de la sous-section 3.1.2, justifier les différentes étapes de algorithme 3.3.

Maintenant nous allons implémenter et tester l'algorithme 3.3 .

6. (9 pts) Programmer une fonction *Python* `sol,xit,nit = gradientPasOptimal(A,b,x0,tol)` pour résoudre le problème $A\mathbf{x} = \mathbf{b}$ par la méthode du gradient à pas optimal (Algorithme 2). La fonction reçoit comme entrée la matrice A, le vecteur $\mathbf{b}$, le point initial $\mathbf{x}_0$ et la tolerance `tol`. En sortie, on a la solution finale `sol`, la solution à chaque itération `xit` et le nombre d'itérations `nit`.

   (a) Donner le code en langage python qui vous à permis de faire les tests numériques. Expliquer votre code le plus rigoureusement possible.

   (b) On fixe $\mathbf{x}_0 := \mathbf{c}_0 = (-9, -7)^T$. Résoudre le système linéaire $(X^T X)\mathbf{c} = X^T\mathbf{q}$ par la méthode du gradient à pas optimal, avec la tolérance `tol` $:= 10^{-6}$.

   (c) Qu'est-ce qu'on observe par rapport au nombre d'itérations?

3.3 L'algorithme des gradients conjugués (30 pts)

Dans cette sous-section, dans le contexte de résolution du problème d'ajustement linéaire (Q), nous allons explicité, implémenter et tester numériquement l'algorithme des gradients conjugués.

3.3.1 Direction du gradient conjugué (14 pts)

La méthode du gradient conjugué peut être vu comme méthode à directions de descente dont les directions de descente successives correspondent à des *corrections* des directions de plus forte descente.

- la direction de descente sera définie par ($k = 0$ est l'indice du premier itéré) :

$$\mathbf{d}_k := \begin{cases} -\nabla f(\mathbf{x}_0) & \text{si } k = 0 \\ -\nabla f(\mathbf{x}_k) + \beta_{k-1}\mathbf{d}_{k-1} & \text{si } k \geq 1, \end{cases} \tag{28}$$

  où le réel β_{k-1}, communément appelé coefficient de conjugaison, est choisi de manière que les deux directions successives $\mathbf{d}_k$ et $\mathbf{d}_{k-1}$ soient $A-$**conjuguées**, *c.-à-d.* $\mathbf{d}_k^T A\mathbf{d}_{k-1} = 0$. C'est pour cette raison que cette direction est appelée **direction du gradient conjugué**.

- Choix du calcul du pas de descente : *à pas optimal* signifie de choisir à chaque au point courant $\mathbf{x}_k$ le pas α_k qui minimise la fonction f dans la direction de gradient conjuguée (associée à $\mathbf{x}_k$).

Nous allons montrer que la direction $\mathbf{d}_k$ définie par (28) est une direction de descente de f au point $\mathbf{x}_k$. Pour cela on procède par récurrence (sur l'indice k).

1. (1 pt) Montrer que $\mathbf{d}_0$ défini par (28) est une direction de descente de f au point $\mathbf{x}_0$.
2. (11 pts) On suppose que $\mathbf{d}_{k-1}$ est une direction de descente de f au point courant $\mathbf{x}_{k-1}$.
   (a) Soit $k \in \mathbb{N}^*$. Montrer que le pas de descente optimal α_{k-1} de f au point courant $\mathbf{x}_{k-1}$ suivant direction $\mathbf{d}_{k-1}$ est donné par :
   $$\alpha_{k-1} = -\frac{\mathbf{r}_{k-1}^T \mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T A\mathbf{d}_{k-1}}. \tag{29}$$
   (b) Montrer que pour tout $k \in \mathbb{N}^*$ on a :
   $$\mathbf{r}_k = \mathbf{r}_{k-1} + \alpha_{k-1} A\mathbf{d}_{k-1}. \tag{30}$$
   (c) Montrer que pour tout $k \in \mathbb{N}^*$ on a :
   $$\mathbf{r}_k^T \mathbf{d}_{k-1} = 0. \tag{31}$$
   (d) Donner la définition d'une direction de descente de q_h au point $\mathbf{x}_k$.
   (e) En utilisant la propriété (31), montrer que $\mathbf{d}_k$ est bien une direction de descente de q_h au point $\mathbf{x}_k$.
3. (2 pts) En utilisant les questions précédentes, montrer que la méthodes du gradient conjugué est bien une méthode à directions de descente.

3.3.2 *Pseudo-code* et test numérique (16 pts)

Nous allons expliciter la méthode dite des gradients conjugués dans le contexte de résolution du problème d'optimisation quadratique (Q).

4. (6 pts) Soit $k \in \mathbb{N}^*$. On suppose que l'on dispose d'un itéré $\mathbf{x}_k \in \mathbb{R}^2$ et de la direction $\mathbf{d}_{k-1} \in \mathbb{R}^2$ de descente de F en $\mathbf{x}_{k-1}$ définie par (28).
   (a) Montrer que le coefficient de conjugaison β_{k-1} entre les directions $\mathbf{d}_{k-1}$ et $\mathbf{d}_k$ est défini par :
   $$\beta_{k-1} = \frac{\mathbf{r}_k^T \left(X^T X\right) \mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T \left(X^T X\right) \mathbf{d}_{k-1}}. \tag{32}$$

(b) L'algorithme des gradients conjugués s'écrit comme ci-dessous.

Algorithme 3.4 (Méthode des gradients Conjugué - une itération)
On suppose qu'au début de l'itération $k-1$ *(pour* $k \geq 1$*) on dispose d'un itéré* $\mathbf{x}_k \in \mathbb{R}^2$ *et de la direction* $\mathbf{d}_{k-1} \in \mathbb{R}^2$.

(a) Test d'arrêt : Si $\|\mathbf{r}_{k-1}\| = \|X^TX\mathbf{x}_{k-1} - X^T\mathbf{q}\| \leq \epsilon$, *alors* $\tilde{\mathbf{x}} := \mathbf{x}_{k-1}$.

(b) Pas de descente : $\alpha_{k-1} := \frac{\mathbf{r}_{k-1}^T \mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T (X^TX)\mathbf{d}_{k-1}}$.

(c) Nouveau itéré : $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_{k-1}\mathbf{d}_{k-1}$.

(d) Coefficient de conjugaison : $\beta_{k-1} = \frac{\mathbf{r}_k^T (X^TX)\mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T (X^TX)\mathbf{d}_{k-1}}$.

(e) Nouvelle direction (conjuguée à $\mathbf{d}_{k-1}$*) :* $\mathbf{d}_k = \mathbf{r}_k + \beta_{k-1}\mathbf{d}_k$.

En combinant la question 4 de la sous-section 3.1.2 et la question précédente, justifier les différentes étapes de algorithme 3.4.

Maintenant nous allons implémenter et tester l'algorithme.

5. (10 pts) Programmer une fonction *Python* `sol,xit,nit = gradientConjugue(A,b,x0,tol)` pour résoudre le problème $A\mathbf{x} = \mathbf{b}$ par la méthode du gradient conjugué (Algorithme 3). La fonction reçoit comme entrée la matrice A, le vecteur b, le point initial $\mathbf{x}_0$ et la tolerance `tol`. En sortie, on a la solution finale `sol`, la solution à chaque itération `xit` et le nombre d'itérations `nit`.

(a) Donner le code en langage python qui vous à permis de faire les tests numériques. Expliquer votre code le plus rigoureusement possible.

(b) On fixe $\mathbf{x}_0 := \mathbf{c}_0 = (-9, -7)^T$. Résoudre le système linéaire $(X^TX)\mathbf{c} = X^T\mathbf{q}$ par la méthode du gradient conjugué. (On fixera la tolérance `tol` $:= 10^{-6}$).

(c) Qu'est-ce qu'on observe par rapport au nombre d'itérations?

3.4 Analyse des résultats *numériques* (20 pts)

L'objectif de cette sous-section est de commenter *numériquement* les comportements des trois méthodes : descente du gradient à pas fixe, à pas optimal et des gradients conjugués. Pour cela, pour chacune des méthodes : 1) on trace la (les) trajectoire(s) générée(s) (il y a autant de trajectoires que de points initiaux choisis); 2) on compte le nombre d'itérations (on peut aussi mesurer le temps CPU); 3) on donne une représentation du modèle *calculé* vis-à-vis des données. Bien évidement, si vous le désirez vous pouvez ajouter des critères de comparaison. Vous y êtes même encouragé !

3.4.1 Les résultats (10 pts)

On considère l'expression algébrique (9) de la fonction F.

1. (4 pts) Trajectoires générées par les méthodes.

(a) On considère un pas de discrétisation $\delta = 0.5$. Définir le pavé $[-10, 10] \times [-10, 10]$ avec un pas δ dans chaque direction. **Fonction :** *numpy.meshgrid*.

(b) Afficher les courbes de niveau de $F(c_1, c_2)$, et son gradient dans le pavé $[-10, 10] \times [-10, 10]$. **Avec Matplotlib utiliser :** *contour*, *quiver*.

(c) Sur les figures obtenues à la question précédente, à partir du vecteur initial c_0 tracer pour chaque méthode qui converge les trajectoires reliant la solution à chaque itération (variable `xit` en sortie des algorithmes). **Avec Matplotlib utiliser :** *plot*.

2. (3 pts) Visualisation des modèles linéaires (4) calculés.
Tracer dans un graphique les couples $(p_i, q_i) \quad i = 1, \ldots, m$ (voir point (b)). Sur la même figure, afficher les trois modèles linéaires $q = c_1 + c_2 p$ contruits à partir des solutions avec les trois méthodes de gradient.

3. (3 pts) Convergence & *vitesse*.
Récapituler, pour chaque méthode, si l'algorithme converge et le nombre d'itérations.

3.4.2 Comportements des méthodes (10 pts)

4. (10 pts) Commenter le comportement des trois méthodes. Bien évidement, si vous le désirez vous pouvez ajouter des critères de comparaison. Vous y êtes même encouragé !

4 Méthodes à directions de descente : étude théorique (150 pts)

4.1 Méthode de descente du gradient à pas fixe

Au paragraphe 3.2.2, nous avons établi (sous la condition (25) sur le choix du pas fixe α) que l'algorithme 3.2 est bien une méthode à direction de descente. Dans cette sous-section, nous allons démontrer un résultat de convergence et de vitesse de convergence (linéaire). Nous déterminerons aussi le choix du pas fixe qui permet d'atteindre le taux de convergence optimal de la méthode.

4.1.1 Résultats de convergence et vitesse de convergence

Le théorème ci-dessous regroupe le résultat de convergence global (convergence quel que soit le choix de l'itéré initial) et de vitesse de convergence linéaire de l'algorithme 3.2.

Théoreme 4.1 (Convergence et vitesse de convergence)
Quel que soit l'itéré initial $\mathbf{x}_0$, *la suite* $(\mathbf{x}_k)_{n\in\mathbb{N}}$ *générée par l'algorithme 3.2 converge vers l'unique solution* $\mathbf{c}^*$ *du problème de minimisation quadratique* (Q), *si et seulement si le pas de descente* α *vérifie la condition* (25). *De plus, on a :*

$$\forall k \in \mathbb{N}, \quad \|\mathbf{x}_{k+1} - \mathbf{c}^*\| \leq \max\left(|1 - \alpha\lambda_2|, |1 - \alpha\lambda_1|\right) \|\mathbf{x}_k - \mathbf{c}^*\|. \tag{33}$$

L'inégalité (33) caractérise la vitesse de convergence de l'algorithme 3.2, et le réel

$$\tau(\alpha) := \max\left(|1 - \alpha\lambda_2|, |1 - \alpha\lambda_1|\right), \tag{34}$$

appelé le **taux de convergence**, mesure de la vitesse de convergence. On dira que la vitesse de convergence de l'algorithme 3.2 est **linéaire** de taux de convergence $\tau(\alpha)$.

L'objectif du paragraphe est de démontrer le théorème 4.1.

1. Montrer que quel que soit l'itéré initial, noté $\mathbf{x}_0$, la suite générée $(\mathbf{x}_k)_{n\in\mathbb{N}}$ par l'algorihme 3.2 vérifie l'égalité suivante :

$$\forall k \in \mathbb{N}^*, \quad \mathbf{x}_k - \mathbf{c}^* = \left(I - \alpha\, X^T X\right)\left(\mathbf{x}_{k-1} - \mathbf{c}^*\right) = \left(I - \alpha\, X^T X\right)^k \left(\mathbf{x}_0 - \mathbf{c}^*\right), \tag{35}$$

   où I désigne la matrice unité d'ordre 2.

2. Dans cette question nous allons calculer le rayon spectral de la matrice $I - \alpha X^T X$.

   (a) Pour une matrice A symètrique de dimension 2, donner (sans preuve) son rayon spectral noté $\varrho(A)$.

   (b) Montrer que si λ est une valeur propre de la matrice $X^T X$ associée au vecteur propre $\mathbf{v}$ alors $1 - \alpha\lambda$ est valeur propre de la matrice $I - \alpha X^T X$ associée au vecteur propre $\mathbf{v}$.

   (c) Déduire des deux questions précédentes que l'on a :

$$\varrho\left(I - \alpha X^T X\right) = \max\left(|1 - \alpha\lambda_2|, |1 - \alpha\lambda_1|\right). \tag{36}$$

   (d) Montrer que $\varrho\left(I - \alpha X^T X\right) < 1$ si et seulement si si et seulement si le pas de descente constant α vérifie la condition (25).

3. Déduire des questions précédentes que l'algorithme 3.2 converge globalement vers $\mathbf{c}^*$ si et seulement si le pas de descente constant α vérifie la condition (25).

4. Dans cette question nous allons établir pour toutes suites $(\mathbf{x}_k)_{n\in\mathbb{N}}$ générée par l'algorithme 3.2 l'inégalité de *vitesse de convergence* (33).

   (a) Pour une matrice A symètrique de dimension 2, donner (sans preuve) sa norme matricielle subordonnée à la norme vectorielle euclidienne. Dans la suite on note $\|A\|$ cette norme matricielle de A.

   (b) En utilisant l'égalité (35), déterminé l'inégalité (33).

Nous allons finir ce paragraphe en montrons que l'algorithme 3.2 peut être interprété comme une méthode itérative basé sur une décomposition régulière de la matrice $X^T X$. Dans ce contexte, elle est aussi connue sous le nom de Méthode de Richardson.

5. Rappeler la définition de la décomposition régulière d'une matrice A inversible.

6. Expliciter le choix particulier de la décomposition régulière de $X^T X$ qui nous permet de retrouver l'algorithme 3.2.

7. À partir de cette reformulation, retrouver le résultat de convergence global énoncé au théorème 4.1.

4.1.2 Pas de descente fixe optimal

Dans le paragraphe précédent, pour un pas fixe α donné vérifiant la condition (25), nous avons établit le taux de convergence $\tau(\alpha)$ de l'algorithme 3.2. Au vu de l'inégalité (33), on peut dire que plus $\tau(\alpha)$ sera proche 0 plus l'algorithme 3.2 sera rapide et inversement plus $\tau(\alpha)$ sera proche de 1 plus l'algorithme 3.2 sera lent (voir d'une vitesse rédibitoire). Maintenant nous allons établir le pas fixe pour lequel le taux de convergence est le plus proche de 0, puis expliciter ce taux et étudier l'impact du conditionnement de la matrice $X^T X$ sur la *vitesse* de la méthode.

8. On considère la fonction *Taux de convergence* définie sur $\mathbb{R}_+^*$ par :

$$\tau(\alpha) := \max\left(|1 - \alpha\lambda_2|, |1 - \alpha\lambda_1|\right). \tag{37}$$

   (a) Tracer la courbe représentative de la fonction τ.

   (b) À l'aide de cette courbe retrouver le résultat de convergence global énoncé au théorème 4.1.

9. On désigne par α^* le pas fixe pour lequel le taux de convergence est le *plus proche* de 0 et par $\tau^* := \tau(\alpha^*)$ le taux de convergence associé.

   (a) Caractériser mathématiquement α^*.

   (b) Montrer que

$$\alpha^* = \frac{2}{\lambda_1 + \lambda_2}, \tag{38}$$

   puis en déduire le taux de convergence optimal τ^*.

   (c) Que peut-on dire de l'impact du conditionnement du problème (*c.-à-d.* du conditionnement de la matrice X^TX) sur la *vitesse* de l'algorithme 3.2.

4.2 Méthode de descente du gradient à pas optimal

On consacre cette sous-section à l'étude mathématique du *comportement* de la methode de descente du gradient à pas optimal. Plus précisément, nous allons : 1) Établir des propriétés géométriques de la trajectoire; 2) Démontrer un résultat de convergence et de vitesse de convergence; 3) Étudier la vitesse de convergence en fonction du conditionnement du problème et du choix du point initial.

Afin de simplifier les calculs, nous allons considérer le problème d'optimisation quadratique (définie positif) sans contrainte suivant :

$$\begin{cases} \text{Minimiser} \quad \mathcal{F}_p(\mathbf{y}) \\ \mathbf{y} \in \mathbb{R}^2 \end{cases}, \tag{Q^*}$$

où $\mathcal{F}_p$ est définie sur $\mathbb{R}^2$ par l'expression analytique :

$$\forall \mathbf{y} := \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad \mathcal{F}_p(y_1, y_2) = \frac{1}{2}\left(y_1^2 + p y_2^2\right), \tag{39}$$

avec $p := \text{cond}_2\left(X^TX\right)$. En effet, les problèmes (Q^*) et (Q) sont équivalents au sens où on passe de la solution $\mathbf{c}^*$ de (Q) à la solution $\mathbf{y}^* := \mathbf{0}$ de (Q^*) et réciproquement par la simple transformation affine inversible $\mathbf{c} - \mathbf{c}^* := P\mathbf{y} \Leftrightarrow \mathbf{y} = P^T(\mathbf{c} - \mathbf{c}^*)$ (caractérise le passage du repère canonique $\mathcal{E}_0$ au repère $\mathcal{V}^*$ et réciproquement).

4.2.1 Propriétés géométrique des *trajectoires*

On note $(\mathbf{y}_k)_{k\in\mathbb{N}}$ une suite générée par l'algorithme 3.3 (du gradient à pas optimal) appliqué au problème d'optimisation (Q^*). Dans ce contexte, on note $(y_k)_1$ et $(y_k)_2$ les composantes (*resp.* les coordonnées) du vecteur (*resp.* du point) $\mathbf{y}_k$ dans la base (*resp.* le repère) de réduction $\mathcal{V}$ (*resp.* $\mathcal{V}^*$) :

$$\mathbf{y}_k = \begin{pmatrix} (y_k)_1 \\ (y_k)_2 \end{pmatrix}.$$

L'objectif de la première question est d'expliciter, en utilisant la question 5 du paragraphe 3.2.3, *les trajectoires* $(\mathbf{y}_k)_{k\in\mathbb{N}}$ générée par l'algorithme 3.3.

1. Vérifier que $(y_{k+1})_1, (y_{k+1})_2$ et le critère d'arrêt s'exprime en fonction de $(y_k)_1, (y_k)_2$ et p comme ci-dessous :

$$(y_{k+1})_1 = \frac{p^2(p-1)(y_k)_1(y_k)_2^2}{(y_k)_1^2 + p^3(y_k)_2^2} \quad , \quad (y_{k+1})_2 = \frac{(1-p)(y_k)_1^2(y_k)_2}{(y_k)_1^2 + p^3(y_k)_2^2} \tag{40}$$

$$\|\mathbf{r}_k\| = \frac{1}{2}\sqrt{(y_k)_1^2 + p^2(y_k)_2^2}. \tag{41}$$

Maintenant, nous allons établir des propriétés géométiques des *trajectoires* de la méthode.

2. Dans cette question nous allons montrer que deux directions successives de l'algorithme 3.3 sont orthogonaux (au sens du produit scalaire canonique).

   (a) Montrer que :

$$\forall k \in \mathbb{N}, \quad \mathbf{d}_{k+1} = \mathbf{d}_k - \alpha_k A\mathbf{d}_k. \tag{42}$$

   (b) Déduire de l'égalilté (42) que l'on a :

$$\forall k \in \mathbb{N}, \quad \mathbf{d}_k^T\, \mathbf{d}_{k+1} = 0. \tag{43}$$

3. On suppose dans cette question que le point $\mathbf{y}_k$ généré par l'algorithme 3.3 à la $k-$ième itération n'appartient pas aux droites D_{c^*,v_1} (Droite passant par le point critique de F et de vecteur directeur $\mathbf{v}_1$) et D_{c^*,v_2} (Droite passant par le point critique de F et de vecteur directeur $\mathbf{v}_2$). Soit $t_k = \frac{(y_)k_2}{(y_k)_1}$ la pente de la droite passant par le point critique de F (*c.-à-d.* le point de représentant matriciel $\mathbf{y}^* := \mathbf{0} = \begin{pmatrix}0\\0\end{pmatrix}$, solution du problème (Q^*)) et $\mathbf{y}_k$.

   (a) Montrer que :

$$t_{k+1} = -\frac{1}{p^2 t_k}. \tag{44}$$

   (b) Déduire de la question précédente que le $(k+1)-$ième itéré $\mathbf{y}_{k+1}$ calculé par l'algorithme 3.3 à partir de $\mathbf{y}_k$ n'appartient pas aux droites D_{c^*,v_1} et D_{c^*,v_2}.

4. Montrer que pour toute suite $(\mathbf{y}_k)_{k\in\mathbb{R}}$ générée par l'algorithme 3.3 de point initial $\mathbf{y}_0$ n'appartenant pas aux droites D_{c^*,v_1} et D_{c^*,v_2}, on a :

$$\begin{gathered}\forall k \in \mathbb{N}, \quad \mathbf{y}_k \notin D_{c^*,v_1} \quad \text{et} \quad \mathbf{y}_k \notin D_{c^*,v_2} \\ \forall k \in \mathbb{N}, \quad t_{2k} = t_0 \quad \text{et} \quad t_{2k+1} = t_1\end{gathered} \tag{45}$$

5. En utilisant les questions 2. et 4. du paragraphe justifier la phrase suivante :
   "Dans le contexte de résolution du problème (Q), les trajectoires générées par l'algorithme 3.3 de point initial $\mathbf{y}_0$ n'appartenant pas aux droites D_{c^*,v_1} et D_{c^*,v_2} zigzag entre les droites $(\mathbf{0}, \mathbf{y}_0)$ et $(\mathbf{0}, \mathbf{y}_1)$ (non confondues avec les droites D_{c^*,v_1} et D_{c^*,v_2} suivant des directions successives orthogonales."

4.2.2 Étude de la convergence et de la vitesse de convergence

Dans ce paragraphe, nous allons étudier la convergence et la vitesse de convergence de l'algorithme 3.3. Plus précisément, nous allons établir un résultat de convergence global, puis un résultats de vitesse de convergence en fonction du conditionnement de la matrice X^TX et du choix du point initial.

6. Dans cette question, on cherche à établir que toute suite $(\mathbf{y}_k)_{k\in\mathbb{R}}$ générée par l'algorithme 3.3 de point initial $\mathbf{y}_0$ n'appartenant pas aux droites D_{c^*,v_1} et D_{c^*,v_2} vérifie l'égalité suivante :

$$\forall k \in \mathbb{N}, \quad \mathbf{y}_{k+2} = \tau^2 \mathbf{y}_k, \tag{46}$$

où τ le facteur moyen de réduction d'erreur vérifie :

$$\tau^2 := \Big(\frac{p-1}{p+1}\Big)^2 \frac{1}{1+\frac{p}{(p+1)^2}\Big(pt-\frac{1}{pt}\Big)^2} \tag{47}$$

où $t \in \{t_0, t_1\}$.

(a) Montrer que l'on a :

$$\forall k \in \mathbb{N}, \quad \frac{(y_{k+1})_2}{(y_k)_2} = \frac{1-p}{1+p^3t_k^2}. \tag{48}$$

(b) Déduire de (48) l'égalité suivante :

$$\forall k \in \mathbb{N}, \quad \frac{(y_{k+2})_2}{(y_k)_2} = \frac{(1-p)^2}{\left(1+\frac{1}{pt_k^2}\right)\left(1+p^3t_k^2\right)}. \tag{49}$$

(c) Déduire de (49) l'égalité suivante :

$$\forall k \in \mathbb{N}, \quad \mathbf{y}_{k+2} = \tau^2 \mathbf{y}_k, \tag{50}$$

où τ le facteur moyen de réduction d'erreur vérifie :

$$\tau^2 := \frac{(1-p)^2}{\left(1+\frac{1}{pt^2}\right)(1+p^3t^2)} \tag{51}$$

où $t \in \{t_0, t_1\}$.

(d) Montrer que :

$$\left(1+\frac{1}{pt^2}\right)\left(1+p^3t^2\right) = (p+1)^2\left(1+\frac{p}{(p+1)^2}\Big(pt-\frac{1}{pt}\Big)^2\right). \tag{52}$$

Puis en déduire (46)-(47).

7. Montrer que pour quelque soit $p \geq 1$, si on choisit le point initial $\mathbf{y}_0$ sur l'une des droites D_{c^*,v_1} et D_{c^*,v_1} alors la suite $(\mathbf{y}_k)_{k\in\mathbb{N}}$ générée par l'algorithme 3.3 identifie la solution $\mathbf{c}^*$ en une itérations (Pensez à donner une preuve reposant sur les propriétés *géométriques* de la méthode).

8. Déduire des questions 6 et 7 un résultat de convergence de l'algorithme 3.3.

9. Dans cette question nous allons évaluer la vitesse de convergence de l'algorithme caractérisé par le taux de de réduction d'erreur τ donné par l'expression (47).

   (a) Déterminer la (les) valeur(s) de t pour lesquelles τ est maximal. En déduire que la valeur du taux de réduction d'erreur correspondant est

   $$\tilde{\tau}^2 = \Big(\frac{p-1}{p+1}\Big)^2 . \tag{53}$$

   (b) Énoncer un résultat de vitesse de convergence (justifier rigoureusement vos propos).

   (c) Étudier la vitesse de convergence de l'algorithme en fonction du conditionnement du problème.

4.3 Méthode des gradients conjugués

4.3.1 Produit scalaire pondéré

Le concept de $A-$conjugaison de deux directions successives sur lequel repose la méthode du gradient conjugué n'est rien d'autre que l'orthogonalité au sens d'un autre produit scalaire que le produit scalaire canonique. Ce nouveau produit scalaire est communément appelé **produit scalaire pondéré** par la matrice A ou $A-$**produit scalaire**. Bien évidement, ce produit scalaire induit sur l'espace $\mathbb{R}^2$ une *nouvelle* géométrie, en comparaison de la géométrie *classique* définie par le produit scalaire canonique. La finalité de se paragraphe est d'exhiber certaines propriétés géométriques qui seront indipensables pour donner quelques interprétations géométriques de la méthodes.

On considère, le temps de ce paragaphe, une matrice $A \in \mathcal{M}_2(\mathbb{R})$ symétrique définie positive :

- les deux valeurs propres sont rangées dans l'ordre croissant : $0 < \alpha_1 \leq \alpha_2$;
- $\forall i \in \{1, 2\}$ tel que $\mathbf{w}_i$ vecteur propre associée à la valeur propre α_i;
- Base des vecteurs propres : $\mathcal{W} := (\mathbf{w}_1, \mathbf{w}_2)$.
- P matrice de passage de la base canonique à la base des vecteurs propres $\mathcal{W}$:

$$D := \operatorname{diag}(\alpha_1, \alpha_2) := P^T A P. \tag{54}$$

1. L'objectif principale de la question est de montrer que l'application notée $\langle \cdot, \cdot \rangle_A$ et définie sur $\mathbb{R}^2 \times \mathbb{R}^2$ à valeur dans $\mathbb{R}$ par

   $$\forall (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^2 \times \mathbb{R}^2, \quad \langle \mathbf{u}, \mathbf{v} \rangle_A := \mathbf{u}^T A \mathbf{v}. \tag{55}$$

   est un produit scalaire sur $\mathbb{R}^2$. Dans l'égalité (55) $\mathbf{u}$ et $\mathbf{v}$ désignent respectivement le vecteur et son représentant (matriciel) dans la base canonique.

   (a) Rappeler la définition (algébrique) d'un produit scalaire.

   (b) Montrer que $\langle \cdot, \cdot \rangle_A$ définit un produit scalaire sur $\mathbb{R}^2$. On appelle communément ce produit scalaire : $A-$**produit scalaire** ou **produit scalaire pondéré** par A.

   (c) Définir la norme induite sur $\mathbb{R}^2$ par le $A-$produit scalaire. On parlera de $A-$**norme** ou **norme pondérée** par A et on la notera $\| \cdot \|_A$.

(d) Vérifier que le produit scalaire canonique est un produit scalaire pondéré pour une matrice symétrique définie positive à préciser.

(e) Soient $\mathbf{u} \in \mathbb{R}^N$, $\mathbf{v} \in \mathbb{R}^N$ de représentants matriciels respectif $\mathbf{y} := \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = P^T\mathbf{u}$ et $\mathbf{z} := \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = P^T\mathbf{v}$ relativement à la base $\mathcal{W}$. Montrer que :

$$\langle \mathbf{u}, \mathbf{v} \rangle_A = \mathbf{u}^T A \mathbf{v} = \mathbf{y}^T D \mathbf{z} = \sum_{\ell=1}^{N} \alpha_\ell y_\ell z_\ell \tag{56}$$

$$\|\mathbf{u}\|_A^2 = \mathbf{u}^T A \mathbf{u} = \mathbf{y}^T D \mathbf{y} = \|\mathbf{y}\|_D^2 = \sum_{\ell=1}^{N} \alpha_\ell y_\ell^2. \tag{57}$$

2. (17 pts) Dans cette question nous allons comparer la norme canonique et la $A-$norme en explicitant l'ensembles des vecteurs de même norme au sens de chacune des normes.

   (a) Pour tout $\beta \geq 0$ préciser la nature géométrique de la $\beta-$courbe de niveau de la fonction $\|\cdot\| : \mathbf{x} \in \mathbb{R}^2 \mapsto \|\mathbf{x}\|$, noté $L_\beta(\|\cdot\|)$.

   (b) Tracer les courbes de niveau $\beta = 0, 1, 2$ de $\|\cdot\|$.

   (c) Pour $\beta \geq 0$, expliciter la nature géometrique de la $\beta-$courbe de niveau de la fonction $\|\cdot\|_A$, noté $L_\beta(\|\cdot\|_A)$.

   (d) Tracer les courbes de niveau $\beta = 0, 1, 2$ de $\|\cdot\|_A$.

   (e) À partir des résultats et schémas des questions (a)-(d), proposer une comparaison (point de vu géométrique) des normes canonique et $A-$pondérée.

Maintenant, dans le contexte du plan affine $\mathbb{R}^2$ nous allons montrer la propriété géométrique :

(**P**) *Soient $\beta > 0$ et $D(M)$ une droite tangente en un point M de la $\beta-$courbe de niveau de la $A-$norme. Alors la droite passant par le point M et $A-$conjugué à la droite $D(M)$ contient l'origine du repère canonique.*

Ce résultat sera essentiel par la suite pour expliquer la stratégie de la méthode du gradient conjugué dans le contexte des méthodes à directions de descente. Dans le cas du plan affine $\mathbb{R}^2$, il permet d'illustrer visuellement la différence entre l'orthogonalité *classique* (induite par le produit scalaire canonique) et la $A-$conjugaison (avec $A \neq I$) de deux vecteurs. Cela permet de comprendre que deux produits scalaires différents induisent sur le même espace (affine) des *géométries* différentes (en terme de calcul d'angle et par conséquent d'orthogonalité de direction, etc.).

3. (20 pts) Dans cette question, on établit la propriété (**P**) dans le cas du plan affine $\mathbb{R}^2$. Pour cela on considère le repère orthonormé, noté $\mathcal{W}_O$: 1) d'origine, notée O, correspond à l'origine du repère canonique; 2) de base orthonormée correspondant à la base des vecteurs propres de A, *c.-à-d.* $\mathcal{W} := (\mathbf{w}_1, \mathbf{w}_2)$.

   (a) Soit $\beta > 0$. Montrer que la courbe de niveau $L_\beta(\|\cdot\|_A)$ admet le paramètrage suivant dans le repère $\mathcal{W}_O$:

$$\forall \theta \in [0, 2\pi], \quad \begin{cases} y_1(\theta) = \frac{\beta}{\sqrt{\alpha_1}} \cos\theta \\ y_2(\theta) = \frac{\beta}{\sqrt{\alpha_2}} \sin\theta \end{cases} \tag{58}$$

(b) Soient $\beta > 0$ et un point quelconque $M(\theta)$ de la courbe $L_\beta(\|\cdot\|_A)$. Vérifier que la droite, noté $\mathcal{T}(M(\theta))$, passant par le point $M(\theta)$ et de vecteur directeur $\mathbf{T}(\theta)$, de représentant

$$\mathbf{T}(\theta) := \begin{pmatrix} -\frac{\beta}{\sqrt{\alpha_1}} \sin\theta \\ \frac{\beta}{\sqrt{\alpha_2}} \cos\theta \end{pmatrix} \tag{59}$$

dans $\mathcal{W}_O$, est tangent à la courbe $L_\beta(\|\cdot\|_A)$ au point $M(\theta)$.

(c) Donner l'équation cartésienne (dans $\mathcal{W}_O$) de la droite, notée $\mathcal{D}(M, \mathbf{T}(\theta))$, passant par le point $M(\theta)$ de la courbe $L_\beta(\|\cdot\|_A)$ et orthogonale au sens du produit scalaire canonique à la droite $\mathcal{T}(M(\theta))$.

(d) Donner l'équation cartésienne (dans $\mathcal{W}_O$) de la droite, notée $\mathcal{D}_A(M, \mathbf{T}(\theta))$, passant par le point $M(\theta)$ de la courbe $L_\beta(\|\cdot\|_A)$ et $A-$conjuguée à la droite $\mathcal{T}(M(\theta))$.

(e) Montrer que si $A \neq I$ alors $O \notin \mathcal{D}(M, \mathbf{T}(\theta))$, et $O \in \mathcal{D}_A(M, \mathbf{T}(\theta))$.

(f) Proposer une (des) figure(s) qui nous permet(tent) de visualiser les *différences* des géométries induites respectivement par le produit scalaire canonique et le $A-$produit scalaire (avec $A \neq I$). Accompagner ces illustrations par des commentaires explicatifs et aidant à comprendre ces *différences*.

4.3.2 Convergence finie

Dans ce paragraphe nous allons montrer et visualiser comment, dans le contexte *simple* du problème Q de dimension 2, l'algorithme du gradient conjugué prend pleinement en compte la *géométrique* du problème en considérant le produit scalaire $\left(X^T X\right)-$pondéré. Plus précisément, Nous allons montrer, comment la méthode corrige la trajectoire en *zig-zag* de la méthode du gradient à pas optimal en produisant une trajectoire qui en deux itérations identifie la solution $\mathbf{c}^*$ du problème.

4. Dans cette question, nous allons démontrer, en utilisant une approche géométrique et sans calculs analytiques, que la méthode identifie la solution en deux itérations.

   (a) Soient $\mathbf{d}_0$ la direction de descente calculé par la méthode des gradient conjugués au point d'initialisation $\mathbf{x}_0$ et $\mathbf{x}_1$ l'itéré suivant. Montrer (sans calcul) que la droite passant par le point $\mathbf{x}_0$ et de vecteur directeur $\mathbf{d}_0$ tangente au point $\mathbf{x}_1$ la courbe de niveau de F (passant par $\mathbf{x}_1$).

   (b) Soient $\mathbf{g}_1 := -\nabla F(\mathbf{x}_1)$ la direction de descente calculé par la méthode de gradient à pas optimale au premier itéré $\mathbf{x}_1$. Montrer que la solution du problème (Q^*) n'appartient pas à la droite passant par le point $\mathbf{x}_1$ et de vecteur directeur $\mathbf{g}_1$.

   (c) Soient $\mathbf{d}_1$ la direction de descente calculé par la méthode des gradients conjugués au premier itéré $\mathbf{x}_1$. Montrer que la solution du problème (Q^*) appartient à la droite passant par le point $\mathbf{x}_1$ et de vecteur directeur $\mathbf{d}_1$.

   (d) Conclure que le second itéré $\mathbf{x}_2$ calculé par la méthode des gradients conjugués **identitfie** la solution du problème (Q).

   (e) Pourquoi le résultat numérique obtenu à la question 5(c) du paragraphe 3.3.2 ne vérifie pas *exactement* le résultat (mathématique) de convergence établi à la question précédente ? (Justifier votre réponse.)

5. En combinant : 1) les représentations faites à la question 1 du paragraphe 3.4.1 des trajectoires générées respectivement par l'algorithme de descente du gradient à pas optimal et des gradients conjugués, 2) les propriétés géométriques des trajectoires de l'algorithme de descente du gradient à pas optimal (*c.f.* paragraphe 4.2.1), 3) la démonstration *géométrique* de la convergence *finie* de la méthodes des gradients conjugués, expliquer le plus rigoureusement possible comment la méthode des gradients conjugués modifie l'algorithme de descente du gradient à pas optimal afin d'*indentifier* en deux itérations la solution c^* du problème (Q^*).

4.4 Comparaison numérique et théorique des méthodes

En combinant l'étude numérique du comportement des 3 méthodes réalisée à la sous-section 3.4 et leur étude théorique faite au niveau de cette section, commenter et comparer le comportement des trois algorithmes.

Il vous est fortement recommandé, au vu des résultats théoriques établis dans cette section, d'ajouter des tests numériques dans le but d'affiner votre comparaison.

Les algorithmes de type gradient

Point initial: $x^{(1)} = x_0 \in \mathbb{R}^n$.
Tolerance pour arrêter l'algorithme: `tol` $= 10^{-6}$.
Nombre maximal d'itérations: `iMax` $= 5 \cdot 10^4$.
Pas pour la méthode 1: voir chaque exercice.

Algorithme 1: Méthode du gradient à pas fixe

Soit $i = 1$.
do
1. $r^{(i)} = Ax^{(i)} - b$;
2. $d^{(i)} = -r^{(i)}$;
3. $x^{(i+1)} = x^{(i)} + \rho d^{(i)}$;
4. $i = i + 1$;

while $\|r^{(i)}\| >$ `tol` & $i <$ `iMax`

Algorithme 2: Méthode du gradient à pas optimal

Soit $i = 1$.
do
1. $r^{(i)} = Ax^{(i)} - b$;
2. $d^{(i)} = -r^{(i)}$;
3. $\rho^{(i)} = \frac{{r^{(i)}}^T r^{(i)}}{{r^{(i)}}^T A r^{(i)}}$;
4. $x^{(i+1)} = x^{(i)} + \rho^{(i)} d^{(i)}$;
5. $i = i + 1$;

while $\|r^{(i)}\| >$ `tol` & $i <$ `iMax`

Algorithme 3: Méthode du gradient conjugué

Soit $i = 1$.
do
1. $r^{(i)} = Ax^{(i)} - b$;
2. **if** $i = 1$
   $d^{(i)} = -r^{(i)}$;
   else
   $\beta^{(i)} = \frac{\|r^{(i)}\|^2}{\|r^{(i-1)}\|^2}$;
   $d^{(i)} = -r^{(i)} + \beta^{(i)} d^{(i-1)}$;
3. $\rho^{(i)} = \frac{{r^{(i)}}^T r^{(i)}}{{d^{(i)}}^T A d^{(i)}}$;
4. $x^{(i+1)} = x^{(i)} + \rho^{(i)} d^{(i)}$;
5. $i = i + 1$;

while $\|r^{(i)}\| >$ `tol` & $i <$ `iMax`

References