Université de Strasbourg **Techniques d'Analyse Numérique (TAN1)**
L3 Mathématiques (S5) **Année 2019-2020**

TD-TP4. Méthodes de gradient

Rappels : Principe des méthodes de gradient.

Soit $A \in \mathcal{M}_n(\mathbb{R})$ une matrice symétrique définie positive. Résoudre $Ax = b$ est équivalent à minimiser

$$f : x \in \mathbb{R}^n \mapsto \frac{1}{2} < Ax, x > - < b, x > . \tag{1}$$

En effet, le minimiseur $\bar{x}$ de f (qui existe et qui est unique puisque f est strictement convexe et coercice) vérifie $\nabla f(\bar{x}) = 0$. Or $\nabla f(x) = Ax - b$.
Une méthode de descente est une méthode itérative qui approche le minimiseur de f (et donc la solution de $Ax = b$) en construisant la suite $(x_k)_{k\in\mathbb{N}}$:

$$\begin{cases} x_0 \in \mathbb{R}^n \\ x_{k+1} = x_k + \alpha_k d_k, \end{cases}$$

Dans ce qui précède, pour $k \in \mathbb{N}$,
• $\alpha_k > 0$ est le pas de la méthode,
• $d_k \in \mathbb{R}^n$ est une direction de descente, c'est-à-dire un vecteur de $\mathbb{R}^n$ tel que $< \nabla f(x_k), d_k > \leq 0$. Lorsque l'inégalité précédente est stricte, on parle de direction stricte de descente.
• On définira aussi le résidu r_k par $r_k = \nabla f(x_k) = Ax_k - b$.

Exemples de méthodes de descente

Méthode de gradient à pas fixe

Pour tout $k \geq 0$,
• la direction de descente vaut
$d_k = -\nabla f(x_k) = b - Ax_k$
• le pas de descente est fixé : $\alpha_k = \alpha > 0$

Méthode de gradient à pas optimal

Pour tout $k \geq 0$,
• la direction de descente vaut
$d_k = -\nabla f(x_k) = b - Ax_k$
• le pas de descente est choisi de manière optimale :
$\alpha_k = \underset{\alpha>0}{\operatorname{argmin}} f(x_k - \alpha \nabla f(x_k))$.
Pour la fonction quadratique f définie par (1), on a $\alpha_k = \frac{<r_k, r_k>}{<Ar_k, r_k>}$.

Méthode de gradient conjugué

Pour tout $k \geq 0$,
• la direction de descente vaut

$d_k = -r_k + \frac{<Ar_k, d_{k-1}>}{<Ad_{k-1}, d_{k-1}>} d_{k-1}$, avec l'initialisation $d_0 = -r_0 = b - Ax_0$,

• le pas de descente est choisi de manière optimale : $\alpha_k = \underset{\alpha>0}{\operatorname{argmin}} f(x_k + \alpha d_k)$.
Pour la fonction quadratique f définie par (1), on a $\alpha_k = -\frac{<r_k, d_k>}{<Ad_k, d_k>}$.

1 Compréhension des méthodes

Exercice 1 : Questions de cours
Méthode du gradient à pas fixe :

1. Montrer que $d_k = -\nabla f(x_k)$ est une direction de descente stricte.

Méthode du gradient à pas optimal :

2. Montrer que, pour f définie par (1), le pas de descente vaut $\alpha_k = \frac{<r_k, r_k>}{<Ar_k, r_k>}$.

Méthode du gradient conjugué :

3. Montrer que les résidus vérifient la relation de récurrence suivante : $r_{k+1} = r_k + \alpha_k A d_k, \forall k \geq 0$.
4. Montrer que, pour tout $k \geq 0$, $< r_{k+1}, d_k >= 0$.
5. En déduire que d_k est une direction de descente stricte.

Exercice 2 : Convergence de la méthode du gradient conjugué

On suppose que l'on n'a pas encore convergé à l'itération k, ce qui signifie que $r_p \neq 0$ pour tout $0 \leq p \leq k$.

1. Montrer, par récurrence, que $< r_k, d_p >= 0$, pour tout $0 \leq p < k$ (on pourra se servir de la question (4) de l'exercice 1).
2. Montrer, par récurrence, que $< r_k, r_p >= 0$, pour tout $0 \leq p < k$ puis que $< Ad_k, d_p >= 0$, pour tout $0 \leq p < k$.
3. En déduire que les $(d_k)_{0 \leq k \leq n}$ forment une famille libre, tant que $r_k \neq 0$.
4. En déduire qu'il existe $k_0 \leq n$ tel que $r_{k_0} = 0$, puis que la suite $(x_k)_{k \in \mathbb{N}}$ converge. En combien d'itérations (au plus) la suite converge-t-elle ?
5. Remarque : Montrer que $\frac{<Ar_k, d_{k-1}>}{<Ad_{k-1}, d_{k-1}>} = \frac{||r_k||^2}{||r_{k-1}||^2}$. La direction de descente de la méthode du gradient conjugué peut donc être calculée par $d_k = -r_k + \frac{||r_k||^2}{||r_{k-1}||^2} d_{k-1}$, avec l'initialisation $d_0 = -r_0$.

Exercice 3 : Méthode de Jacobi vue comme une méthode de descente

On considère la fonction

$$f : x \mapsto \frac{1}{2} < Ax, x > - < b, x >$$

avec A une matrice de $\mathcal{M}_n(\mathbb{R})$ symétrique définie positive et b un vecteur de $\mathbb{R}^n$.

1. Montrer que la méthode itérative de Jacobi (utilisée pour résoudre le système $Ax = b$) peut s'écrire comme une méthode de descente à pas fixe pour la minimisation de f. On l'appellera la *méthode de descente de Jacobi*. Donner l'expression du pas de descente α et de la direction de descente d_k.
2. Vérifier que ce d_k est bien une direction de descente (stricte) pour tout $k \geq 0$.
3. On souhaite améliorer la *méthode de descente de Jacobi* en prenant non plus un pas fixe α dans l'algorithme mais un pas optimal, défini à l'itération k par

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k).$$

   On conserve la même direction de descente d_k que pour la *méthode de descente de Jacobi*. On parlera dans la suite de *méthode de descente de Jacobi à pas optimal*.
   (a) Justifier l'existence et l'unicité de ce pas optimal α_k.
   (b) Donner son expression à chaque itération.
4. On suppose maintenant que la diagonale extraite de la matrice A s'écrit $D = \beta \mathrm{I}_n$, avec I_n la matrice identité.
   (a) Écrire la *méthode de descente de Jacobi* dans ce cas particulier et la *méthode de descente de Jacobi à pas optimal*.
   (b) Quel(s) algorithme(s) de descente vu(s) en classe reconnaissez-vous ?

2 Implémentation python

Exercice 4 : Méthode du gradient à pas fixe : illustration graphique
On va illustrer numériquement le principe de la méthode de gradient à pas fixe en s'assurant que la suite des itérées $(x_k)_{k\in\mathbb{N}}$ converge bien vers le minimum de $f : x \mapsto \frac{1}{2} < Ax, x > - < b, x >$.

1. Écrire une fonction python `GradientPasFixe`, qui à partir de A une matrice symétrique définie positive, b un vecteur de $\mathbb{R}^n$, α un pas de descente, x_0 un vecteur initialisation, $\varepsilon > 0$ la précision voulue et k_{max} le nombre maximal d'itérations renvoie la valeur approchée de la solution de $Ax = b$. On utilisera pour critère d'arrêt le critère suivant

   `arrêt lorsque` $\|Ax_k - b\| \leqslant \varepsilon$ `ou` $k > k_{max}$.

2. Appliquer cette fonction à $f(x, y) = 2x^2 + 3x + y^2$ et $g(x, y) = x^2 + y^2 - 2y$. On prendra soin d'expliciter les matrices A_f, A_g et les vecteurs b_f et b_g correspondants. On pourra par exemple prendre une précision de $\varepsilon = 10^{-10}$, un pas $\alpha = 0.01$ et on pourra partir de $x_0 = (7.5, 5)^t$. Vérifier que le nombre total d'itérations n'atteint pas votre k_{max}.
3. Vérifier que le résultat fourni par votre fonction `GradientPasFixe` est bien le même que celui obtenu par une méthode itérative (Jacobi ou Gauss-Seidel par exemple).
4. On va visualiser graphiquement la suite $(x_k)_{k\in\mathbb{N}}$ obtenue par la méthode de gradient à pas fixe.
   (a) Créer une grille de taille $[-10, 10] \times [-10, 10]$. Les abscisses des nœuds de cette grille seront stockés dans un vecteur `X` et les ordonnées dans `Y`. Pour cela, on pourra utiliser le code
   ```
   X_bord = np.arange(-10,10,0.01)
   Y_bord = np.arange(-10,10,0.01)
   X, Y = np.meshgrid(X_bord, Y_bord)
   ```
   (b) Tracer les lignes de niveau de la fonction f au moyen de la commande `plt.contour(X, Y, f(X, Y))`.
   (c) Afficher, sur la figure, les points $(x_k)_{k\in\mathbb{N}}$ au moyen d'un marqueur `o`. On pourra pour cela modifier notre fonction python `GradientPasFixe` pour qu'elle ressorte la liste des abscisses et des ordonnées des $(x_k)_{k\in\mathbb{N}}$. La figure est-elle conforme à nos attentes ?
   (d) Dans une seconde figure, afficher la liste des itérées $(x_k)_{k\in\mathbb{N}}$ et les lignes de niveau pour la fonction g.

Exercice 5 : Matrice de Hilbert
On veut résoudre $Ax = b$ pour A la matrice de Hilbert de terme général $a_{i,j} = \frac{1}{i+j-1}$, pour $1 \leq i, j \leq N$. On notera $\bar{x}$ la solution.

1. Écrire une fonction `MatHilbert(N)` afin d'implémenter la matrice A en python (attention, les indices en python commencent à 0, il faut donc adapter la formule de $a_{i,j}$).
2. Coder l'algorithme de gradient à pas optimal dans une fonction `GradientOptimal`, qui prend en argument une matrice A symétrique définie positive, un vecteur b, une initialisation x_0, une précision ε et un nombre maximal d'itérations k_{max} et qui redonne en sortie l'approximation de $\bar{x}$ par la méthode du gradient à pas optimal ainsi que le nombre effectif d'itérations.
3. De même, coder une fonction `GradientConjugue` avec les mêmes arguments et les mêmes sorties que précédemment mais pour la méthode du gradient conjugué.
4. Comparer ces deux méthodes avec A la matrice de Hilbert de taille $N = 10$, $b = A^{-1}\bar{x}$ où $\bar{x} = (1, 1, ..., 1)^t$. On pourra prendre une précision de $\varepsilon = 10^{-10}$. Quel est le résultat (théorique) attendu ? Quels sont les résultats numériques ? Comparer le nombre effectif d'itérations pour ces deux méthodes : pouvait-on prévoir un nombre maximal d'itérations pour la méthode du gradient conjugué (si oui, lequel) ?

Exercice 6 : Comparaison des vitesses de convergence

On prendra dans cet exercice $A = \begin{pmatrix} 1 & & & & 0 \\ & 2 & & & \\ & & 3 & & \\ & & & \ddots & \\ 0 & & & & N \end{pmatrix}$ et b tel que la solution $\bar{x}$ recherchée soit $\bar{x} = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}^t$.

1. Modifier vos fonctions `GradientPasFixe`, `GradientOptimal` et `GradientConjugue` pour qu'elles ressortent (en plus des sorties précédentes) la liste des erreurs $||\bar{x} - x_k||$ pour $k \geq 0$. On pourra calculer $\bar{x}$ avec la commande `np.linalg.solve(A,b)`.
2. Sur un même graphique, comparez les vitesses de convergence de ces trois méthodes pour $N = 100$. Pour cela, afficher les erreurs en fonction du nombre d'itérations (pour la méthode de gradient à pas fixe, on pourra prendre le pas $\alpha = 0.01$). On affichera les courbes avec une échelle logarithmique en ordonnée au moyen de la commande `plt.yscale('log')`. Penser à mettre un titre, des légendes et des couleurs différentes à vos courbes.

3 Pour aller plus loin : minimisation sur un sous-espace vectoriel

Exercice 7 : Minimisation d'une fonctionnelle quadratique
Soient $A \in \mathcal{M}_n(\mathbb{R})$ une matrice symétrique définie positive et b, c, d trois vecteurs de $\mathbb{R}^n$ avec $d \neq 0$. On cherche à minimiser $f : x \mapsto \frac{1}{2} < Ax, x > - < b, x >$ sur $K = \{x \in \mathbb{R}^n, < d, x >= c\}$.

1. Montrer que l'ensemble K est non vide, fermé et convexe. En déduire que f admet un unique minimiseur sur K.
2. Montrer que si $\bar{x}$ est le minimiseur de f sur K alors il existe $\lambda \in \mathbb{R}$ tel que $y = (\bar{x}, \lambda)^t$ soit l'unique solution du système $\begin{pmatrix} A & d \\ d^t & 0 \end{pmatrix} y = \begin{pmatrix} b \\ c \end{pmatrix}$.

Exercice 8 : Applications concrètes de la minimisation
On cherche à maximiser l'aire d'un rectangle de périmètre ℓ donné.

1. Montrer que ce problème peut se réécrire sous la forme d'un problème de minimisation sur un sous-espace vectoriel K.
2. À l'aide des multiplicateurs de Lagrange, résoudre ce problème d'optimisation.

On étudie maintenant le solide S constitué d'un cylindre de hauteur h et de rayon r et de deux demi-sphères de rayon r (voir figure 1). On cherche à minimiser sa surface pour un volume fixe de $\frac{32}{3}\pi$.

3. Réécrire ce problème sous la forme d'un problème d'optimisation sous contrainte et le résoudre au moyen d'un multiplicateur de Lagrange.

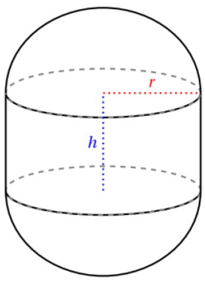

Figure 1 – Solide S