Université de Caen **U. F. R. Sciences**

Optimisation, licences MASS & Maths TP4 2012-2013

Méthodes du gradient

Fabien Navarro

1 Algorithme du gradient à pas constant pour la minimisation d'une forme quadratique

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

$$f(x) = \frac{1}{2}x^T Ax - x^T b,$$

où A et b sont respectivement une matrice symétrique définie positive d'ordre n et un vecteur de $\mathbb{R}^n$.

On considère l'algorithme du gradient à pas fixe :

$$\begin{cases} x_0 \in \mathbb{R}^n \text{ , donné} \\ x_{k+1} = x_k - \rho \nabla f(x_k), \end{cases} \tag{1}$$

Algorithm 1: Gradient à pas fixe

input : x_0 (première approximation de la solution cherchée), A, b, ρ (pas), ε (précision demandée), $n_{\max}$ (nombre maximal d'itérations)

output: $x^\star$, k

Initialisation;
$x_k \longleftarrow x_0$;
for $k = 1 : n_{\max}$ **do**
    $d_k \longleftarrow b - Ax_k$ (direction de descente);
    $x_{k+1} \longleftarrow x_k + \rho d_k$;
    if $\|d_k\| < \varepsilon$ & $k < n_{\max}$ *(critère d'arrêt)* **then**
        La méthode converge après k itérations;
        (`mprintf('La methode converge apres %d iterations.\n ',k);`);
        return
    else
        $x_k \longleftarrow x_{k+1}$;
    end
end
Attention le nombre maximale $nmax$ d'itération est atteint ;

On testera les méthodes sur l'exemple suivant :

$$A = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

Exercice 1.

1) Écrire une fonction `f` (de la forme `function y = f(x,A,b)`).
2) Écrire une fonction `gradientfixe` sur le modèle de l'**Algorithme 1**.
3) Vérifier que la matrice A est définie positive (à l'aide d'une condition `if` sur le spectre de A).
4) Résoudre le problème (1) pour $n = 5$, $\varepsilon = 1e\text{-}7$ et $\varepsilon = 1e\text{-}8$, $n_{\max} = 1000$ et $\rho = 0.1$ et $x_0 = (0, \ldots, 0)^T$. (vous pouvez utiliser la fonction `mprintf` pour l'affichage à la **7** et la fonction `warning` si vous voulez afficher un avertissement dans la console)
5) Refaire la question 4) en remplaçant la boucle `for` par une boucle `while` dans la l'**Algorithme 1**.
6) Déterminer numériquement la valeur ρ conduisant à un nombre d'itérations minimal de l'**Algorithme 1**. (vous pouvez représenter graphiquement le nombre d'itérations en fonction de différentes valeurs de ρ en appelant votre fonction `gradientfixe` dans une boucle. Penser à la vitesse de convergence optimal vue en cours. Attention, si le pas choisi est trop grand, l'algorithme peut diverger. S'il est trop petit, la convergence peut être extrêmement lente).

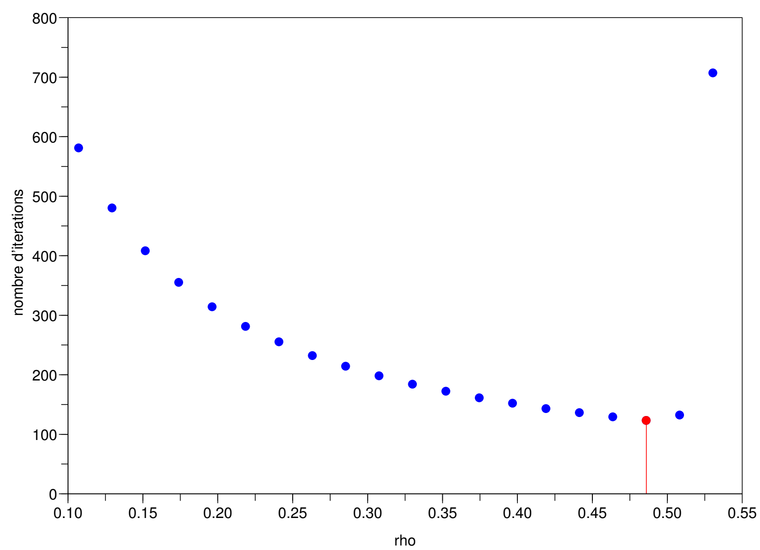


FIGURE 1 – Nombre d'itérations en fonction de différentes valeurs de ρ.

7) Résoudre le problème (1) en utilisant la fonction `derivative` afin d'approximer ∇f et comparer vos résultats. (attention à la syntaxe d'appel de cette fonction. Si vous avez définie une fonction f qui reçoit plusieurs arguments (*i.e.*, de la forme `y = f(x,A,b)`), il faut utiliser la fonction `list` (*e.g.*, `derivative(list(f,A,b),xmin,H_form='blockmat')`).

Exercice 2.

1) Tester votre algorithme sur l'exemple suivant

$$f(x) = \frac{1}{2}x^T Q x + x^T b + 24,$$

où $x = (x_1, x_2)^T$ un vecteur de $\mathbb{R}^2$, $b = (3,6)^T$ et $Q = \begin{pmatrix} 8 & 2\sqrt{2} \\ 2\sqrt{2} & 10 \end{pmatrix}$ une matrice symètrique d'ordre 2.

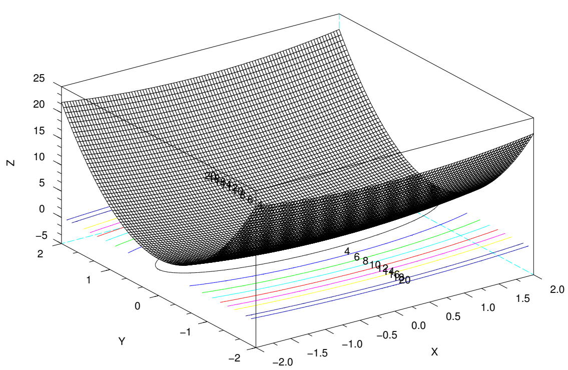
2) Représenter graphiquement le nombre d'itérations en fonction de différentes valeurs de ρ.

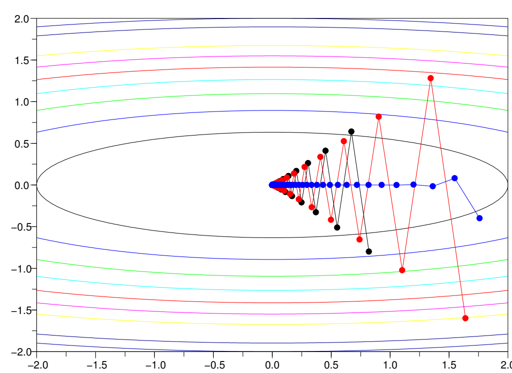
Exercice 3. On souhaite minimiser la forme quadratique

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

où $\eta = 10$

1) Représenter graphiquement la fonction f (en utilisant par exemple `fplot3d` ou `plot3d` et `feval` ou encore `meshgrid` et `mesh`) par exemple sur le pavé $[-2,2] \times [-2,2]$.
2) Calculer le gradient de f.
3) Écrire une fonction `f` qui renvoie f et ∇f.
4) Modifier l'**Algorithme 1** afin qu'il reçoive la fonction définie à la question précédente (indication : on a $d_k = -\nabla f(x_k)$).
5) Résoudre le problème de minimisation en utilisant l'algorithme du gradient à pas fixe en prenant par exemple, $\varepsilon = 1e\text{-}7$, $n_{\max} = 5000$ et $\rho = 1.8/\eta$ et $x_0 = (1,1)^T$.
6) (facultatif) Modifier votre algorithme afin de retourner les itérées x_k et représenter les graphiquement dans le plan pour différentes valeurs de x_0 et ρ. Ajouter quelques lignes de niveaux.





2 Méthode du gradient à pas optimal

La méthode du gradient à pas optimal s'écrit :

$$\left\{ \begin{array}{l} x_0 \in \mathbb{R}^n \text{ , donné} \\ \psi_k(\rho) = f\left(x_k - \rho_k \nabla f(x_k)\right), \\ x_{k+1} = x_k - \rho_k \nabla f(x_k), \end{array} \right. \tag{2}$$

où ρ_k réalise le minimum sur $\mathbb{R}_+^*$ de la fonction ψ_k. Contrairement à la méthode du gradient à pas fixe qui oblige à toujours faire le même pas dans la direction de descente, la méthode du gradient à pas variable consiste à optimiser globalement la fonctionnelle dans la direction du gradient.

Algorithm 2: Gradient à pas optimal

input : x_0, ε, $n_{\max}$, A, b
output: $x^\star$,...
Initialisation;
$x_k \longleftarrow x_0$;
$d_k \longleftarrow b - Ax_k$;
$\rho_k \longleftarrow \frac{\|d_k\|^2}{d_k^T A d_k}$;
$k \longleftarrow 0$;
while $k < n_{\max}$ & $\|d_k\| > \varepsilon$ **do**
    $k \longleftarrow k + 1$;
    $x_{k+1} \longleftarrow x_k + \rho_k dk$;
    $d_{k+1} \longleftarrow b - Ax_{k+1}$;
    $\rho_{k+1} \longleftarrow \frac{\|d_{k+1}\|^2}{d_{k+1}^T A d_{k+1}}$;
end
La méthode converge après k itérations;

Exercice 4.

1) Définir une fonction `gradientopt` sur le modèle de l'**Algorithme 2**.
2) Résoudre le problème (2) pour $n = 5$, $\varepsilon = 1e\text{-}7$, $n_{\max} = 1000$ et $x_0 = (0, \ldots, 0)^T$.
3) Comparer vos résultats (nombre d'itérations, solutions,...) avec ceux obtenus avec la méthode du gradient à pas fixe.
   (vous pouvez également tracer le nombre d'itérations en fonction de différentes valeurs de n pour les deux méthodes).