Optimisation continue

Farida Meghatria

University of Khemis Miliana

Cours et exercices, Master 1, Analyse Mathématique et Applications

Contents

1 Introduction

2 Algorithme de descente général

Introduction

Introduction

On considère un problème de minimisation sans contrainte :

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

où $f : \mathbb{R}^n \to \mathbb{R}$ est supposée **continûment différentiable**.

Définition

Un vecteur $d \in \mathbb{R}^n$ est une **direction de descente** pour f en un point x si :

$$\nabla f(x)^T d < 0$$

Cela garantit que pour un pas $\alpha > 0$ suffisamment petit, on a :

$$f(x + \alpha d) < f(x)$$

Algorithme de descente général

Principe de l'algorithme

L'algorithme de descente général construit une suite $(x_k)_{k\in\mathbb{N}}$ définie par :

$$x_{k+1} = x_k + \alpha_k d_k$$

où :

- $\surd$ $x_k \in \mathbb{R}^n$ est le point courant à l'itération k,
- $\surd$ $d_k \in \mathbb{R}^n$ est une direction de descente telle que $\nabla f(x_k)^T d_k < 0$,
- $\surd$ $\alpha_k > 0$ est un pas de descente.

Choix de la direction d_k

Différents choix sont possibles :

- $\surd$ Descente du gradient : $d_k = -\nabla f(x_k)$,
- $\surd$ Méthode de Newton : $d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$,
- $\surd$ Méthodes quasi-Newton : $d_k = -B_k^{-1} \nabla f(x_k)$,
- $\surd$ Gradient conjugué, etc.

Choix du pas α_k

Trois approches courantes:

Pas fixe : $\alpha_k = \alpha$,

Recherche linéaire exacte :

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

Recherche linéaire inexacte : conditions d'Armijo ou de Wolfe.

Pseudo-code de l'algorithme de descente

Data : Point initial $x_0 \in \mathbb{R}^n$, tolérance $\varepsilon > 0$, max itérations $K_{\max}$
Result : Approximation d'un minimum de f

Initialiser $k \leftarrow 0$
repeat
  Calculer le gradient $\nabla f(x_k)$
  Choisir une direction de descente d_k telle que $\nabla f(x_k)^T d_k < 0$
  Choisir un pas $\alpha_k > 0$ (par exemple via une recherche linéaire)
  Mettre à jour : $x_{k+1} \leftarrow x_k + \alpha_k d_k$
  $k \leftarrow k + 1$
until $\|\nabla f(x_k)\| \leq \varepsilon$ ***ou*** $k \geq K_{\max}$
return x_k

Algorithme 1 : Algorithme de descente général

Critères d'arrêt

L'algorithme s'arrête si l'un des critères suivants est satisfait :

- $\|\nabla f(x_k)\| \leq \varepsilon$
- $\|x_{k+1} - x_k\| \leq \varepsilon$
- $|f(x_{k+1}) - f(x_k)| \leq \varepsilon$
- Nombre maximal d'itérations atteint : $k \geq K_{\max}$

Exercice

Application de l'algorithme de descente du gradient sur une fonction simple.

1. Appliquer la méthode de descente du gradient.
2. Choisir un pas fixe.
3. Implémenter tout cela dans un algorithme clair.

Problème

On cherche à minimiser la fonction :

$$f(x) = (x-2)^2$$

Son gradient est donné par :

$$\nabla f(x) = 2(x-2)$$

Data : Point initial $x_0 = 0$, pas $\alpha = 0.1$, tolérance $\varepsilon = 10^{-4}$, max itérations $K_{\max} = 100$
Result : Approximation d'un minimum de f
Initialiser $k \leftarrow 0$
repeat
  Calculer le gradient : $g_k \leftarrow 2(x_k - 2)$
  Calculer la direction de descente : $d_k \leftarrow -g_k$
  Mettre à jour : $x_{k+1} \leftarrow x_k + \alpha \cdot d_k$
  $k \leftarrow k + 1$
until $|f'(x_k)| \leq \varepsilon$ *ou* $k \geq K_{\max}$
return x_k

Algorithme 2 : Descente du gradient pour $f(x) = (x-2)^2$

Méthode du gradient à pas fixe

On prend $d^{(k)} = -\nabla f(x^{(k)})$, et un pas constant $\alpha_k = \alpha$.
Avantages : simplicité. Inconvénient : lenteur, sensibilité au choix de α.

Principe

On cherche à minimiser une fonction différentiable $f : \mathbb{R}^n \to \mathbb{R}$.
L'algorithme du gradient à pas fixe repose sur la mise à jour suivante :

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

où :

- x_k est l'approximation de la solution à l'itération k,
- $\alpha > 0$ est un **pas fixe** (ou *learning rate*),
- $\nabla f(x_k)$ est le gradient de f en x_k.

Algorithme

Require: Fonction $f : \mathbb{R}^n \to \mathbb{R}$, gradient ∇f,
point initial x_0, pas $\alpha > 0$,
tolérance $\varepsilon > 0$, nombre maximal d'itérations $N_{\max}$

$x \leftarrow x_0$
$k \leftarrow 0$
while $\|\nabla f(x)\| > \varepsilon$ **and** $k < N_{\max}$ **do**
    $x \leftarrow x - \alpha \nabla f(x)$
    $k \leftarrow k + 1$
end while
return x

Algorithme 3 : Méthode du gradient à pas fixe

Conditions de convergence

Si f est convexe et de classe $\mathcal{C}^1$ avec un gradient L-Lipschitzien, alors la méthode converge pour :

$$0 < \alpha < \frac{2}{L}$$

Exemple (fonction quadratique)

Considérons :

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

avec $A \in \mathbb{R}^{n\times n}$ symétrique définie positive.
Alors :

$$\nabla f(x) = Ax - b$$

Et l'algorithme devient :

$$x_{k+1} = x_k - \alpha(Ax_k - b)$$

Implémentation MATLAB de la Méthode du Gradient à Pas Fixe

Fonction `gradient_fixe.m`

[caption=Méthode du gradient à pas fixe] function [x, k, history] = gradient$_fixe(f, grad_f, x0, alpha, epsilon, N_max)$

x = x0; k = 0; history = x;

while norm(grad$_f(x)) > epsilonk < N_max x = x - alpha * grad_f(x); k = k + 1; history = [history, x]; endend$

Exemple d'utilisation

[caption=Minimisation d'une fonction quadratique] A = [3 0; 0 1]; b = [1; 2];
f = @(x) 0.5 * x' * A * x - b' * x; grad$_f$ = $@(x)A * x - b$;
x0 = [0; 0]; alpha = 0.1; epsilon = 1e-6; N$_max$ = 1000;
[x$_sol$, $k, hist$] = $gradient_fixe(f, grad_f, x0, alpha, epsilon, N_max)$;
fprintf('Solution trouvée en disp(x$_sol$);
plot(hist(1,:), hist(2,:), '-o');
xlabel('x$'_1$); $ylabel('x'_2)$; $title('Trajectoire du gradient pas fixe')$; $gridon$;

Fonction choisie : Fonction de Rosenbrock

$$f(x,y) = (1-x)^2 + 100(y-x^2)^2$$

Gradient :

$$\nabla f(x,y) = \begin{bmatrix} -2(1-x) - 400x(y-x^2) \\ 200(y-x^2) \end{bmatrix}$$

[caption=Minimisation de la fonction de Rosenbrock] f = @(x) (1 - x(1))$^2 + 100 * (x(2) - x(1)^2)^2$;
grad$_f$ = @(x)[$-2 * (1 - x(1)) - 400 * x(1) * (x(2) - x(1)^2); 200 * (x(2) - x(1)^2)$];
x0 = [-1.2; 1]; alpha = 1e-3; epsilon = 1e-6; N$_max$ = 10000;
[x$_sol, k, hist$] = $gradient_fixe(f, grad_f, x0, alpha, epsilon, N_max)$;
fprintf('Solution trouvée après disp(x$_sol$);
plot(hist(1,:), hist(2,:), '-o'); xlabel('x'); ylabel('y'); title('Trajectoire sur la fonction de Rosenbrock'); grid on;
Le point converge vers $(1, 1)$.