Introduction à l'optimisation sans contrainte
Aspects numériques

Djaffar Boussaa

CNRS/LMA

Contact : boussaa@lma.cnrs-mrs.fr

Plan

Plan

Direction de descente

Définition

Un vecteur p est dit direction de descente pour la fonction f au point x si $p'\nabla f(x) < 0$.

Exemples

- $p = -\nabla f(x)$ en un point x où $\nabla f(x) \neq 0$.
- Toute direction de la forme $-D(x)\nabla f(x)$ où $D(x)$ est une matrice définie positive en un point x où $\nabla f(x) \neq 0$.

Proposition

Si p est une direction de descente pour la fonction f au point x, alors il existe $\overline{t} > 0$ tel que $f(x + tp) < f(x)$ pour tout $t \in]0, \overline{t}[$.

Direction de courbure négative

Définition

Une direction $p \in \mathbb{R}^n$ est dite direction de courbure négative pour la fonction f au point x si $p\prime\nabla^2 f(x)p < 0$.

En résumé

- **Condition du premier ordre** Si un point $x^\star$ est un minimum local alors il n'y a pas de direction de descente en $x^\star$
- **Condition du second ordre** Si un point $x^\star$ est un minimum local alors il n'y a pas de direction de courbure négative en $x^\star$

Algorithme 1 : Algorithme générique

$x \leftarrow x_0$
pour $k = 0, 1, 2, \ldots$ **faire**
  si x_k *est optimal* **alors**
    retourner x_k
  fin
  calculer une direction de descente p_k
  calculer un pas α_k par recherche linéaire
  mettre à jour x : $x_{k+1} = x_k + \alpha_k p_k$
fin

Remarque

A chaque itération, p_k est pris souvent de la forme $p_k = -B_k^{-1}\nabla f_k$, et est typiquement (et dans toute la suite du cours) une direction de descente, c-à-d $\nabla f_k^T p_k < 0$. Si bien que pour α suffisamment petit, est garantie la relation $f(x_k + \alpha p_k) < f(x_k)$.

Plan

Recherche linéaire exacte

- **Une première idée** Intuitivement, il semble souhaitable de choisir α_k tel que

$$\alpha_k = \arg\min_{\alpha>0} \phi(\alpha)$$

avec

$$\phi(\alpha) = f(x_k + \alpha p_k)$$

On parle de recherche **linéaire exacte**

- **Exemple** Cas quadratique :

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

$$\alpha_k = -\frac{\nabla f_k^T p_k}{p_k^T Q p_k}$$

Recherche linéaire exacte (suite)

- **Constat** En général, pas de solution analytique, d'où recours aux méthodes numériques.
  Vous pouvez mettre en œuvre votre méthode de minimisation unidimensionnelle favorite à cet effet :
  - Méthode de Fibonacci
  - Méthode du nombre d'or
  - interpolation polynomiale
  - etc.
- **Conséquence** La recherche linéaire exacte est "chère" et il existe des méthodes plus économiques : "il ne faut pas perdre de vue : ce qu'on souhaite, c'est minimiser $f(x)$ et non $[\phi(\alpha)]$. **Il est donc totalement inutile de chercher à minimiser f avec précision dans la direction courante, et ceci à chaque itération.**" (C. Lemaréchal)

On parle alors de recherche **linéaire inexacte**

Recherche linéaire inexacte

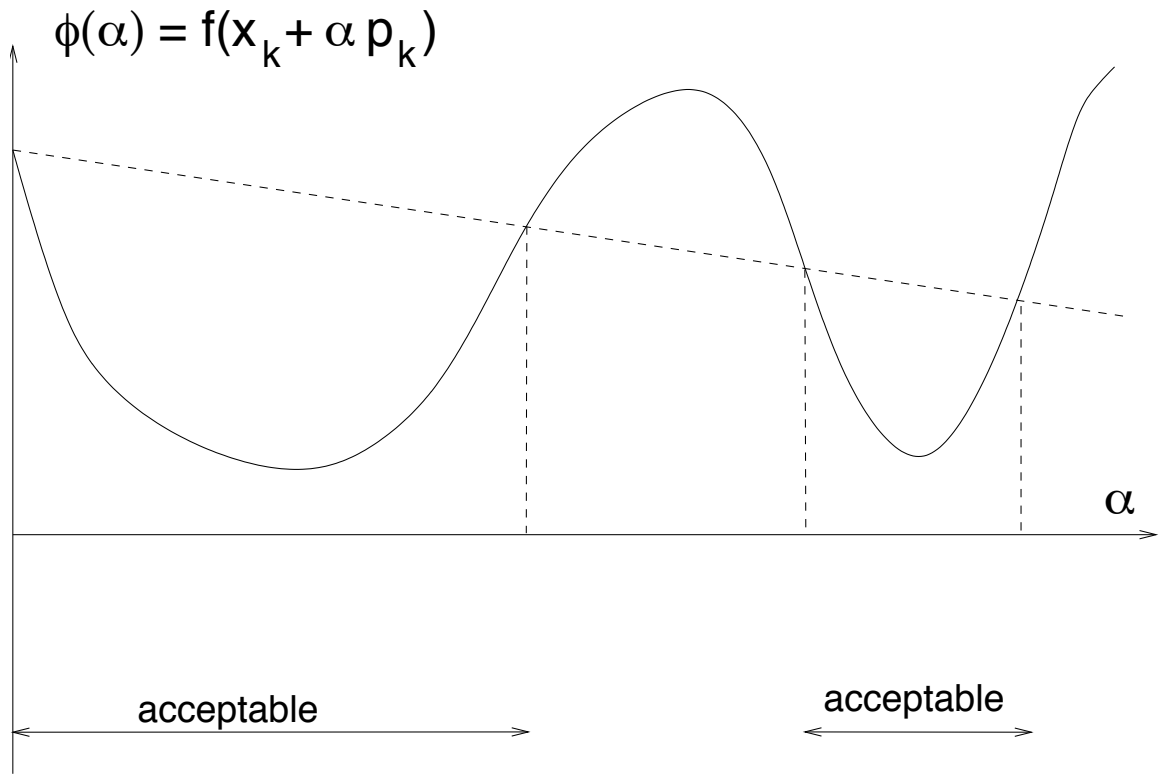
- **Idée** Se contenter d'un α_k non optimal.
- **Question** De quel α_k peut-on se contenter ?
- **Certitude** Toute suite (α_k) tel que $f(x_k + \alpha_k p_k) < f(x_k)$ pour tout k ne suffit pas pour assurer la convergence.

Recherche linéaire inexacte (suite)

- **Exemples à méditer** (Dennis et Schnabel)
  - **Exemple 1**
    - Fonction-objectif : $f(x) = x^2$
    - Itéré initial : $x_0 = 2$
    - Directions de recherche : $\{p_k\} = \{(-1)^{k+1}\}$
    - Pas : $\{\alpha_k\} = \{2 + 3(2^{-(k+1)})\}$
    - Itérés : $\{x_k\} = \{(-1)^k(1 + 2^{-k})\}$
    - La suite $\{x_k\}$ne converge pas (deux points d'accumulation -1 et $+1$)
  - **Exemple 2**
    - Fonction objectif : $f(x) = x^2$
    - Itéré initial : $x_0 = 2$
    - Directions de recherche : $\{p_k\} = \{-1\}$
    - Pas : $\{\alpha_k\} = \{2^{-k+1}\}$
    - Itérés : $\{x_k\} = \{1 + 2^{-k}\}$
    - La suite $\{x_k\}$converge vers 1 qui n'est pas un minimum de f
- **Conclusion** Il faut des **conditions** plus strictes que la simple décroissance pour assurer la convergence

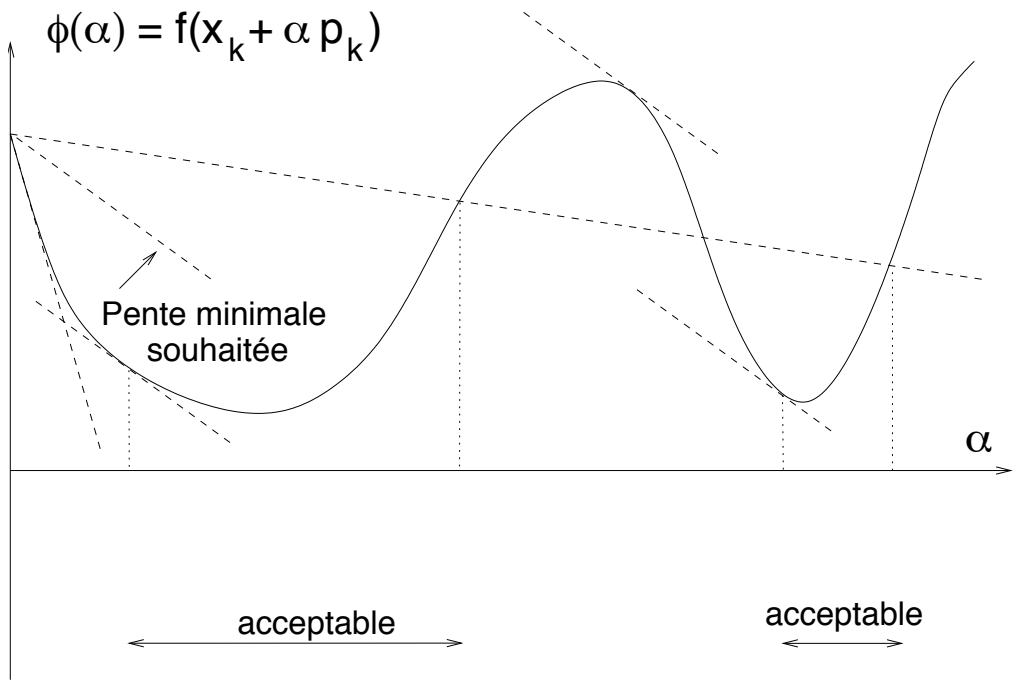
Quelques conditions courantes : conditions d'Armijo

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k$$



Conditions de Wolfe $(0 < c_1 < c_2 < 1)$

$$\begin{aligned} f(x_k + \alpha_k p_k) &\leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \\ \nabla f(x_k + \alpha_k p_k)^T p_k &\geq c_2 \nabla f_k^T p_k \end{aligned}$$



Condition de Goldstein $(0 < c < 1/2)$

$$\begin{aligned} f(x_k) + (1-c)\alpha_k \nabla f_k^T p_k &\leq f(x_k + \alpha_k p_k) \\ f(x_k + \alpha_k p_k) &\leq f_k + c\alpha_k \nabla f_k^T p_k \end{aligned}$$

Conditions fortes de Wolfe $(0 < c_1 < c_2 < 1)$

$$\begin{aligned} f(x_k + \alpha_k p_k) &\leq f(x_k) + c_1 \alpha_k \nabla f_k^T p_k \\ \left| \nabla f(x_k + \alpha_k p_k)^T p \right| &\leq c_2 \left| \nabla f_k^T p_k \right| \end{aligned}$$

Une méthode simple

Algorithme 2 : méthode de rebroussement

Entrées : $\bar{\alpha},\ \rho,\ c \in]0, \frac{1}{2}[$

Sorties : α

$\alpha \leftarrow \bar{\alpha}$

tant que $f(x_k + \alpha p_k) > f(x_k) + c\alpha_k \nabla f_k^T p_k$ **faire**

| $\alpha \leftarrow \rho\alpha$

fin

- **Remarques**
  - En général $\bar{\alpha} = 1$ dans le cas de la méthode de Newton et des méthodes quasi-newtoniennes
  - Un pas acceptable sera trouvé après un nombre fini d'itérations
  - Recherche linéaire acceptable pour la méthode de Newton.
  - Elle est en général insuffisante.

Algorithme 3 : (Wolfe "faible")

Initialisation : Choisir $0 < c_1 < c_2 < 1$
Poser $\alpha = 0, \beta = \infty$ et $t = 1$
répéter
  si $f(x + td) > f(x) + c_1 t f'(x; d)$ **alors**
    $\beta = t$
    $t = \frac{1}{2}(\alpha + \beta)$
  sinon si $f'(x + td; x) < c_2 f'(x; d)$ **alors**
    $\alpha = t$
    si $\beta = +\infty$ **alors**
      $t = 2\alpha$
    sinon
      $t = \frac{1}{2}(\alpha + \beta)$
    fin
  sinon
    retourner t
  fin
fin

Algorithme 4 : Wolfe “fort”

Entrées : α_1 et $\alpha_{\max}$

$\alpha_0 \leftarrow 0$

répéter

  évaluer $\phi(\alpha_i)$

  si $\phi(\alpha_i) > \phi(0) + c_1\alpha_i\phi'(0)$ *ou* $[\phi(\alpha_i) \geq \phi(\alpha_{i-1})$ *et* $i > 1]$ **alors**

    $\alpha_* \leftarrow$ **zoom**(α_{i-1}, α_i) et **stop**

  fin

  évaluer $\phi'(\alpha_i)$

  si $|\phi'(\alpha_i)| \leq -c_2\phi'(0)$ **alors**

    $\alpha_* \leftarrow \alpha_i$ et **stop**

  fin

  si $\phi'(\alpha_i) \geq 0$ **alors**

    $\alpha \leftarrow$ **zoom**(α_i, α_{i-1}) et **stop**

  fin

  choisir $\alpha_{i+1} \in (\alpha_i, \alpha_{\max})$

  $i \leftarrow i + 1$

fin

Algorithme 5 : Algorithme zoom

Entrées : α_{lo}, α_{hi} bornes d'un intervalle contenant des points satisfaisant les conditions de Wolfe

répéter

  trouver (e.g. par interpolation) un pas α entre α_{lo} et α_{hi}

  évaluer $\phi(\alpha)$

  si $\phi(\alpha) > \phi(0) + c_1\alpha\phi'(0)$ *ou* $\phi(\alpha) \geq \phi(\alpha_{lo})$ **alors**

    $\alpha_{\mathrm{hi}} \leftarrow \alpha$

  sinon

    évaluer $\phi'(\alpha)$

    si $|\phi'(\alpha)| \leq -c_2\phi(0)$ **alors**

      $\alpha_* \leftarrow \alpha$ et **stop**

    fin

    si $\phi'(\alpha)(\alpha_{hi} - \alpha_{lo}) \geq 0$ **alors**

      $\alpha_{\mathrm{hi}} \leftarrow \alpha_{\mathrm{lo}}$

    fin

    $\alpha_{\mathrm{lo}} \leftarrow \alpha$

  fin

fin

Plan

Algorithme 6 : Méthode de la plus forte pente

Entrées : f, ∇f, x_0

Sorties : x, une approximation d'un minimum de f

$x \leftarrow x_0$

pour $k = 1, 2, \ldots$ **faire**

  si x *est optimal* **alors**

    retourner x

  fin

  $p \leftarrow -\nabla f(x)$

  calculer un pas α_k par recherche linéaire

  $x \leftarrow x + \alpha p$

fin

Méthode d'intérêt plutôt théorique

- très lente (la suite des x_k est oscillante)
- La méthode devrait être "**interdite**" (C. Lemaréchal)
- La recherche sur la méthode continue ! (e.g. Méthode de Barzilai–Borwein et ramifications)

Application au cas quadratique

- Fonction-objectif

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

où la matrice Q est supposée symétrique définie positive

- Le gradient de f est $\nabla f = Qx + b$
- Le pas donné par une recherche linéaire exact

$$\alpha_k = \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k}$$

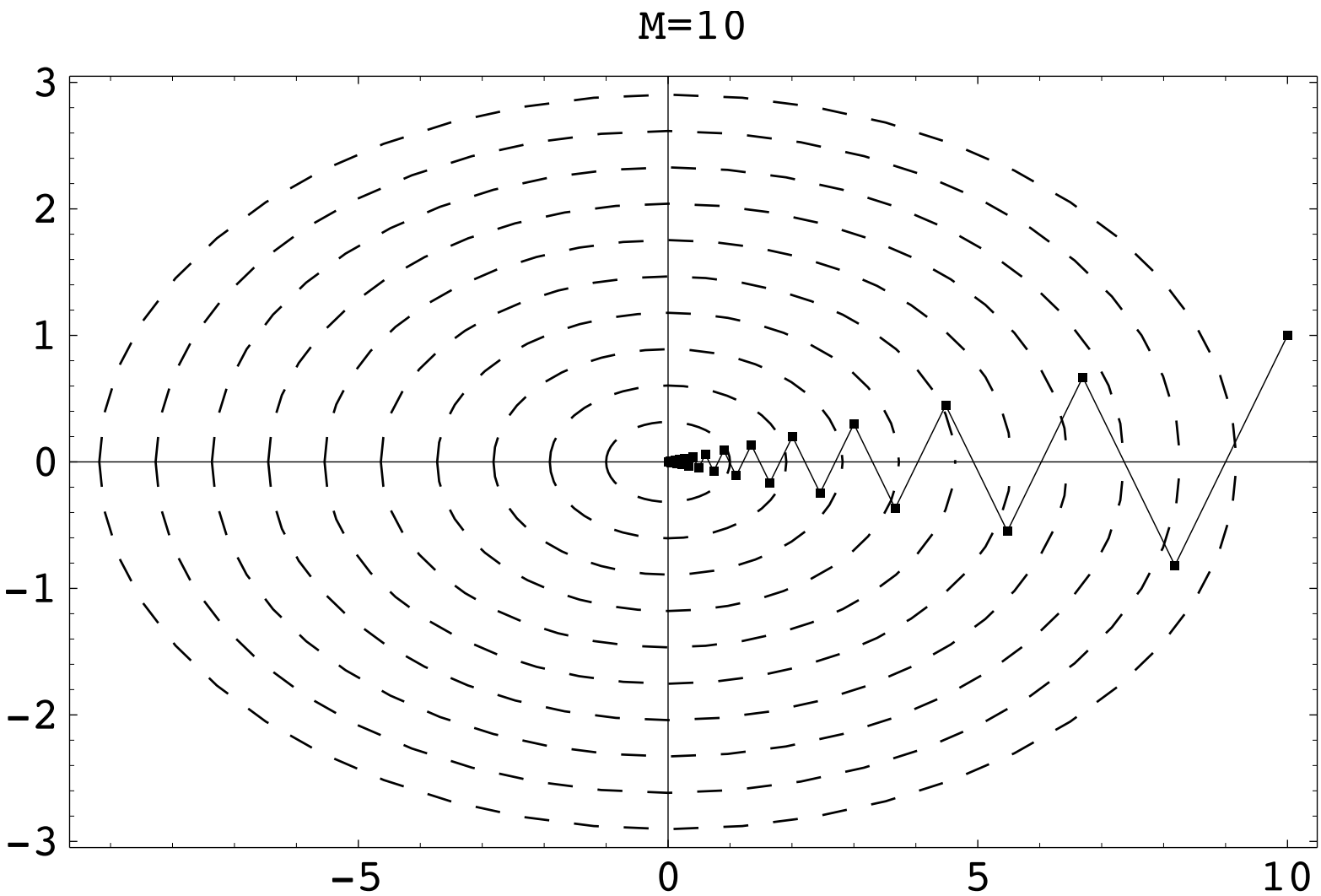
- La mise à jour de x est

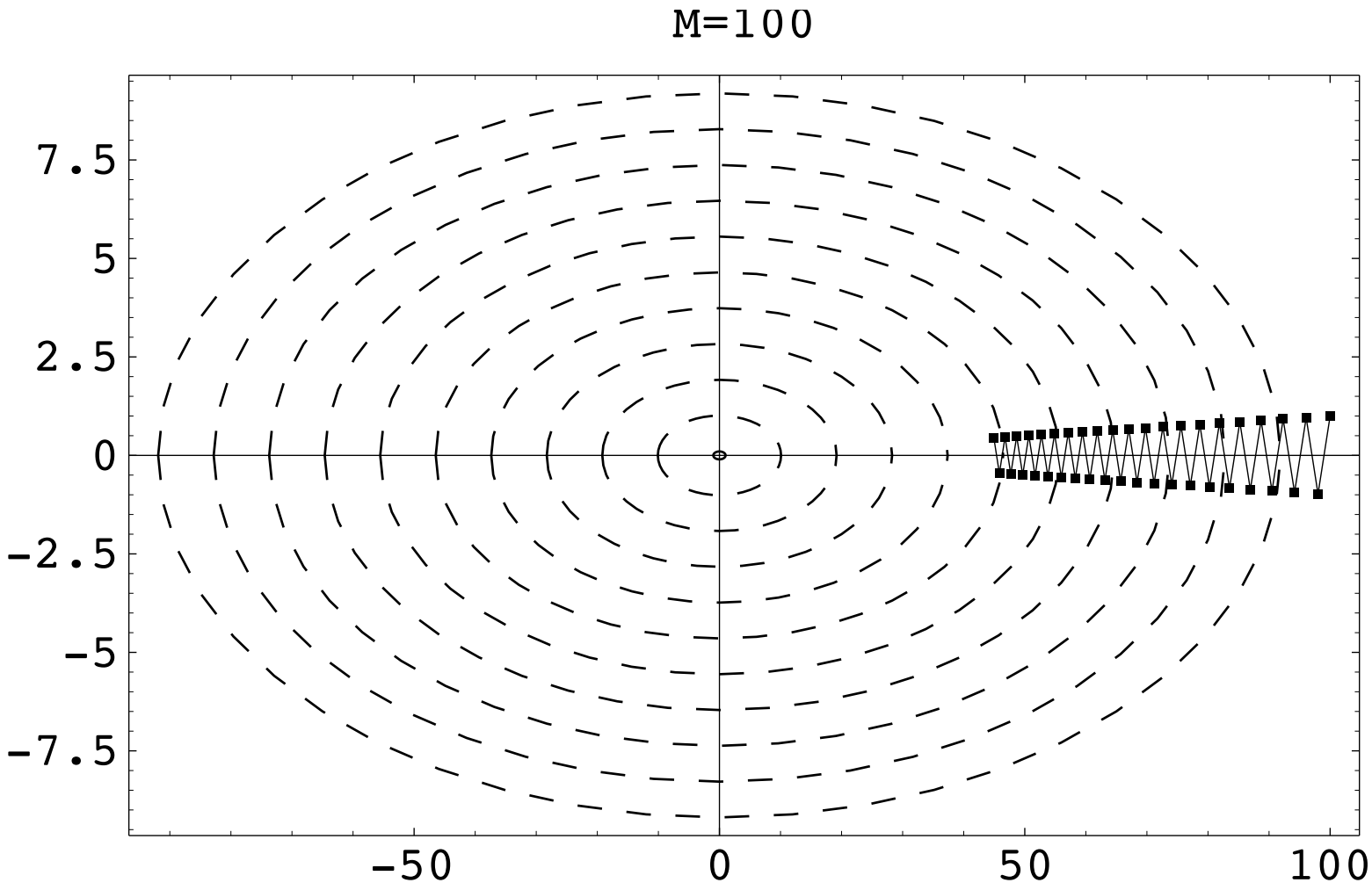
$$x_{k+1} = x_k - \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k} \nabla f_k$$

Mise en oeuvre dans le cas bidimensionnel

$$
\begin{aligned}
Q &= \begin{pmatrix} 1 & 0 \\ 0 & M \end{pmatrix} \qquad (M > 0) \\
f(x_1, x_2) &= \frac{1}{2}(x_1^2 + M x_2^2) \\
x_0 &= \begin{pmatrix} M \\ 1 \end{pmatrix} \\
x_k &= \left(\frac{M-1}{M+1}\right)^k \begin{pmatrix} M \\ (-1)^k \end{pmatrix}
\end{aligned}
$$

- si M est proche de 1, la convergence est très rapide
- si $M \gg 1$ ou $M \ll 1$, convergence très lente en zigzag (Cf. courbes ci-dessous)


M=10
3
2
1
0
−1
−2
−3
−5
0
5
10



- Comportement similaire en dimension supérieure

Plan

Algorithme 7 : Méthode de Newton (forme de base)

Entrées : f, ∇f, $\nabla^2 f$, x_0

Sorties : x, une approximation d'un minimum de f

$x \leftarrow x_0$

pour $k = 1, 2, \ldots$ **faire**

  si x *est optimal* **alors**

    retourner x

  fin

  $p^N \leftarrow -\nabla^2 f_k^{-1} \nabla f(x)$

  $x \leftarrow x + p^N$

fin

Méthode de Newton (suite)

Interprétations

- L'itéré x_{k+1} minimise le développement au 2nd ordre de f au voisinage de x_k

$$q(x) = f(x_k) + \nabla f(x)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k)$$

- L'itéré x_{k+1} est une solution de la linéarisation autour de la condition d'optimalité

$$\nabla f(x_k) + \nabla^2 f(x)(x_{k+1} - x_k) = 0$$

Méthode de Newton (suite)

Avantages

- convergence quadratique si l'itéré initial est bon (combien d'itérartions sont-elles nécessaire dans le cas quadratique défini positif ?)
- méthode non affectée par un changement d'échelle

Inconvénients

- Peut diverger ou échouer à converger
- Converge vers un point stationnaire dans sa forme de base

Des remèdes existent

- Modification de la hessienne
- Contrôle du pas

Plan

Algorithme 8 : Méthodes quasi-newtoniennes

Entrées : x_0, une approximation initiale de la hessienne B_0
($B_0 = I$ est un choix possible)

pour $k = 0, 1, 2, \ldots$ **faire**

si x_k *est optimal* **alors** stop

Calculer une direction quasi-newtonienne p_k en résolvant :

$$B_k p = -\nabla f(x_k)$$

Déterminer un pas α_k par recherche linéaire

Mettre à jour x : $x_{k+1} = x_k + \alpha_k p_k$

Calculer

$$s_k = x_{k+1} - x_k$$

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

Mettre à jour B : $B_{k+1} = B_k + \cdots$

fin

Méthodes quasi-newtoniennes (suite)

Idée remplacer $\nabla^2 f(x)$ par une approximation B moins "chère" à calculer et/ou à stocker
Plusieurs règles de mise à jour de B. Les différents méthodes différent par le choix de B.

Avantages

- B peut-être construite en utilisant les **dérivées premières seulement**. Plus besoin de calculer la hessienne. Travail et risque d'erreur réduits
- La direction de recherche peut être calculée en $O(n^2)$ opérations (plutôt qu'en $O(n^3)$ opérations pour la méthode de Newton)

Inconvénients

- Convergence rapide mais non quadratique
- nécessité de stockage d'une matrice. Toutefois, des versions sans stockage ou avec stockage limité existent (LBFGS)

Conditions (de bon sens) que doit satisfaire une approximation de la hessienne ?

Il semble raisonnable qu'une approximation de la hessienne

- soit symétrique ($n(n+1)/2$ termes à déterminer)
- soit définie positive (n relations)
- satisfasse l'équation de la sécante (n relations)
- soit minimale en un certain sens (pour fermer le problème)

Equation de la sécante

- Dans le cas monodimensionnel

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

- Dans le cas multidimensionnel

$$\nabla^2 f(x_k)(x_k - x_{k-1}) \approx \nabla f(x_k) - \nabla f(x_{k-1})$$

- B_k choisie telle que

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

  On dit que B_k satisfait **l'équation de la sécante**.

- **Notations**

$$\begin{aligned} s_k &= x_{k+1} - x_k \\ y_k &= \nabla f_{k+1} - \nabla f_k \end{aligned}$$

Formule de Sherman-Morrison-Woodbury

- Soit $\overline{A}$ une modification de rang 1 d'une matrice carrée inversible A

$$\overline{A} = A + ab^T$$

Si $\overline{A}$ est inversible alors

$$\overline{A}^{-1} = A^{-1} - \frac{A^{-1}ab^TA^{-1}}{1 + b^TA^{-1}a}$$

Méthode DFP

$$\min_H \|B - B_k\|$$
$$\text{sous les contraintes} \quad B = B,^T \quad Bs_k = y_k$$

$$B_{k+1}^{\text{DFP}} = (1 - \gamma_k y_k s_k^T) B_k (I - \gamma_k s_k y_k^T) + \gamma_k y_k y_k^T$$

$$\gamma = \frac{1}{y_k^T s_k}$$

$$\boxed{H_{k+1}^{\text{DFP}} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}}$$

Méthode BFGS (1970)

$$\min_H \|H - H_k\|$$
$$\text{sous les contraintes} \quad H = H,^T \quad Hy_k = s_k$$

La solution est

$$\boxed{H_{k+1}^{\text{BFGS}} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T}$$

où

$$\rho_k = \frac{1}{y_k^T s_k}$$

- appellation consacrée en l'honneur de ses inventeurs (Broyden, Fletcher, Goldfarb, Shanno)
- Régle de mise à jour

$$B_{k+1}^{\text{BFGS}} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

- Très utilisée en pratique

Algorithme 9 : Algorithme BFGS

Entrées : x_0, ϵ et H_0

tant que $\|\nabla f_k\| \leq \varepsilon$ **faire**

calculer une direction de recherche

$$p_k = -H_k^{\mathrm{BFGS}} \nabla f_k$$

calculer un pas α_k satisfaisant les conditions de Wolfe ($\alpha_0 = 1$)

passer à l'itéré suivant $x_{k+1} = x_k + \alpha_k p_k$

calculer $s_k = x_{k+1} - x_k$ et $y_k = \nabla f_{k+1} - \nabla f_k$

calculer H_{k+1}^{BFGS}

$k \leftarrow k + 1$

fin

Plan

Algorithme 10 : Méthode GC non linéaire (Fletcher–Reeves)

Entrées : x_0, ϵ

évaluer $f_0 = f(x_0)$

évaluer $\nabla f_0 = \nabla f(x_0)$

$p_0 = -\nabla f_0$

$k \leftarrow 0$

tant que $\|\nabla f_k\| \leq \varepsilon$ **faire**

- calculer un pas α_k
- poser $x_{k+1} = x_k + \alpha_k p_k$
- évaluer $\nabla f(x_{k+1})$
- $\beta_{k+1}^{\mathrm{FR}} \leftarrow \dfrac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$
- $p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\mathrm{FR}} p_k$
- $k \leftarrow k + 1$

fin

Méthode du gradient conjugué non linéaire (suite)

Méthode de Polak–Ribière $\beta_{k+1}^{\mathrm{PR}} \leftarrow \dfrac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k}$

Méthode de Polak–Ribière modifiée $\beta_{k+1}^{+} \leftarrow \max\left\{\beta_{k+1}^{\mathrm{PR}}, 0\right\}$