Algorithmes de Newton et de quasi-Newton

I Préliminaires

1 Règles et orientation

Deux techniques permettent de générer des suites convergeant vers une solution d'un problème d'optimisation **sans** contrainte.

(1) Recherche linéaire (RL)

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

α_k : pas > 0 déterminé par RL

d_k : direction de descente de f

(2) Régions de confiance (RC)

$$\begin{cases} \min & g_k^T s + \frac{1}{2} s^T M_k s \\ \|s\|_2 \leq \Delta_k \end{cases} \longrightarrow \text{solution } s_k$$

+ réglage de $\Delta_k > 0$ (le rayon de confiance)

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

Aujourd'hui on se concentre sur deux directions de descente

- la direction de Newton
- une direction de quasi-Newton (BFGS)

Toutes les deux donnent une **convergence rapide**.

Mais qu'est-ce qu'une convergence rapide ?

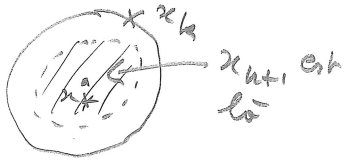
B Vitesses de convergence des suites

Comment quantifier la vitesse de convergence d'une suite $(x_k) \subset$ e.v. $\mathbb{E}$ vers un point $x_* \in \mathbb{E}$?

On suppose que $x_k \neq x_*$, $\forall k \geq 1$.

Linéaire : $\exists$ norme $\|\cdot\|$ $\exists r \in [0,1[$, $\exists k_0 \in \mathbb{N}$, $\forall k \geq k_0$

$$\|x_{k+1} - x_*\| \leq r \, \|x_k - x_*\|$$



- dépend de la norme choisie
- pour beaucoup d'algorithmes (proximal, lagrangien augmenté, ...)
- pas très bon en pratique (trop lent si r proche de 1)

Superlinéaire $\dfrac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \longrightarrow 0$

- typique des algorithmes de quasi-Newton
- souvent TB en pratique (la vitesse dépend souvent de $\dim \mathbb{E}$)

Quadratique $\exists C > 0$, $\forall k \geq 1$:

$$\|x_{k+1} - x_*\| \leq C \, \|x_k - x_*\|^2$$

- typique de l'algorithme de Newton
- très rapide (dès que x_k est "proche" de x_* il devient très proche en 4-5 itérations)

Evidemment

quadratique $\Rightarrow$ superlinéaire $\Rightarrow$ linéaire

prop (vague, à préciser)

si • $(x_k) \longrightarrow x_*$ quadratiquement

• $x_* \neq 0$

alors "asymptotiquement" le nombre de chiffres significatifs corrects de x_k "double" à chaque itération.

- D'abord, on suppose que $x_* \neq 0$, sinon on ne peut pas parler de nombre de chiffres significatifs corrects de x_k
- C'est quoi le nombre de chiffres significatifs corrects de x_k ?

$$x_* = 0.\underbrace{1234\ldots}_{\hookrightarrow = \text{les chiffres significatifs de } x_*} 10^{27}$$

le nombre de chiffres significatifs corrects de x_k =

$$\nu_k := -\log_{10} \frac{\|x_k - x_*\|}{\|x_*\|}$$

- Montrons que, asymptotiquement, ν_k double à chaque itération.

$$\|x_{k+1} - x_*\| \leq C \|x_k - x_*\|^2$$

$$\frac{\|x_{k+1} - x_*\|}{\|x_*\|} \leq C \|x_*\| \left(\frac{\|x_k - x_*\|}{\|x_*\|}\right)^2$$

$\nu_k \to \infty$

$$\nu_{k+1} \geq C_1 + 2\nu_k$$

$$\liminf_{k \to \infty} \frac{\nu_{k+1}}{\nu_k} \geq 2$$

Comparison of Newton's and Broyden's algorithms

Find a zero of $F : \mathbb{R}^2 \to \mathbb{R}^2$ defined at $x \in \mathbb{R}^2$ by

$$F(x) = \begin{pmatrix} x_1 + x_2^2 \\ x_2 + x_1^3 \end{pmatrix},$$

which has two zeros, $(0, 0)$ and $(-1, 1)$.

Starting at $x_0 = (1, 1)$, the Newton and Broyden algorithms generate sequences of iterates $\{x_k\}$ with the following values of $F(x_k)$.

```
Newton iterations (quadratic convergence)
^^^^^^^^^^^^^^^^^
it      |F(x)|     |F(x)|/|F(x_)|^2
 0   2.000000e+00
 1   6.400000e-01    1.600000e-01
 2   6.717265e-01    1.639957e+00
 3   6.667554e-01    1.477684e+00
 4   3.956913e-01    8.900686e-01
 5   1.578483e-01    1.008154e+00
 6   7.859331e-03    3.154321e-01
 7   6.176909e-05    1.000000e+00
 8   4.713501e-13    1.235382e-04
 9   2.221709e-25    1.000000e+00
Solution found

Quasi-Newton iterations (superlinear convervence)
^^^^^^^^^^^^^^^^^^^^^^^
it      |F(x)|     |F(x)|/|F(x_)|  |F(x)|/|F(x_)|^2
 0   2.000000e+00
 1   2.000000e+00   1.000000e+00     5.000000e-01
 2   8.888889e-01   4.444444e-01     2.222222e-01
 3   5.000000e-01   5.625000e-01     6.328125e-01
 4   2.825471e-01   5.650943e-01     1.130189e+00
 5   2.428675e-01   8.595644e-01     3.042198e+00
 6   1.347208e-01   5.547093e-01     2.284000e+00
 7   2.128868e-02   1.580207e-01     1.172949e+00
 8   6.544300e-03   3.074074e-01     1.443995e+01
 9   6.469757e-04   9.886095e-02     1.510642e+01
10   2.175462e-05   3.362509e-02     5.197273e+01
11   1.790352e-07   8.229758e-03     3.782994e+02
12   1.019739e-10   5.695746e-04     3.181355e+03
13   7.146168e-14   7.007838e-04     6.872187e+06
14   1.382865e-17   1.935114e-04     2.707904e+09
15   1.751731e-23   1.266741e-06     9.160264e+10
Solution found
```

One observes the quadratic convergence of the Newton algorithm and the superlinear (but not quadratic) convergence of the Broyden algorithm.

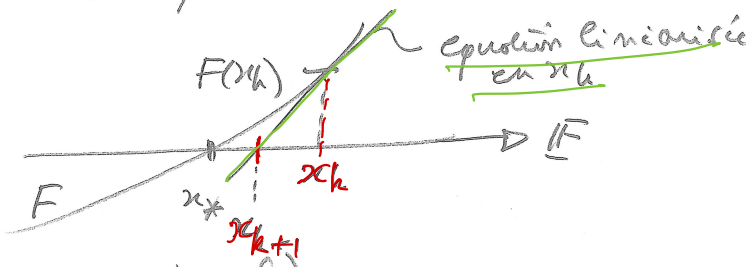
2) Algorithme de Newton (1669, 1671)

(aussi et surtout SIMPSON, 1740)

A Algorithme de Newton pour système d'équations

- L'algorithme de Newton est d'abord une méthode pour rechercher un système d'équations non linéaires

$$F(x) = 0$$



où $F : \mathbb{E} \to \mathbb{F}$ (un espace vectoriel) ; avec autant d'équations que d'inconnues (i.e., dim $\mathbb{E}$ = dim $\mathbb{F}$)

- Le principe est le suivant : $x_k \to x_{k+1}$ par

⊗ on linéarise F en x_k :

$$x \in \mathbb{E} \longmapsto F(x_k) + F'(x_k)(x - x_k)$$

⊗ on calcule un zéro de l'équation linéarisée

$$F(x_k) + F'(x_k)\underbrace{(x_{k+1} - x_k)}_{=: d_k} = 0$$

Donc, si $F'(x_k)$ est inversible on a

$$\boxed{x_{k+1} = x_k + d_k} \quad \text{avec} \quad \boxed{d_k = -F'(x_k)^{-1} F(x_k)}$$

direction de Newton

- Conditions de bon fonctionnement
  - F différentiable et F' lipschitzienne (i.e., $F \in C^{1,1}$)
  - $F'(x_*)$ inversible
  - x_1 proche de x_*

Voyons cela.

Théo (convergence quadratique locale)

Si . $x_* \in E$ est tel que $F(x_*) = 0$

. F est $C^{1,1}$ dans un voisinage de x_*

. $F'(x_*)$ est inversible

alors $\exists$ voisinage V de x_*, tel que si $x_1 \in V$ alors l'algorithme de Newton

- est bien défini et génère une suite (x_k)
- $(x_k) \to x_*$ quadratiquement

Dem Si $x_k \in V_1 = B(x_*, r_1)$

$$x_{k+1} - x_* = x_k - x_* - F'(x_k)^{-1} F(x_k)$$

$$= -F'(x_k)^{-1} \left[F(x_k) - \underbrace{F(x_*)}_{=0} - F'(x_k)(x_k - x_*) \right]$$

$$= -\underbrace{F'(x_k)^{-1}}_{\text{bornée}} \int_0^1 \underbrace{\left[F'(x_* + t(x_k - x_*)) - F'(x_k) \right]}_{\|\cdot\| \le L \|(t-1)(x_k - x_*)\| = L(1-t)\|x_k - x_*\|} (x_k - x_*)\, dt$$

$$\|x_{k+1} - x_*\| \le C \frac{L}{2} \|x_k - x_*\|^2$$

$$= \underbrace{\left(C \frac{L}{2} \|x_k - x_*\| \right)}_{\le \frac{1}{2} \text{ si } x_k \in \overline{B}\left(x_*, \frac{1}{CL}\right) \subset V_1} \|x_k - x_*\|$$

$\Rightarrow x_k \to x_*$, quadratiquement

□

(B) Algorithme de Newton en optimisation

- On se contente de rechercher qu'un point stationnaire du problème "$\min f(x)$", c-à-d de résoudre

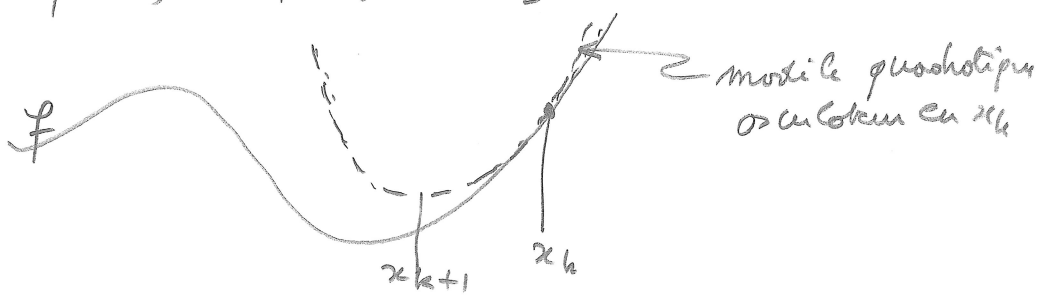
$$\nabla f(x) = 0$$

Donc $F = \nabla f$ dans le cadre précédent. On obtient

$$\boxed{\begin{aligned} x_{k+1} &= x_k + d_k \\ d_k &= -\nabla^2 f(x_k)^{-1} \nabla f(x_k) \end{aligned}}$$

- On a le même résultat de convergence que précédemment avec $F = \nabla f$, qui demande que
  - f est $C^{2,1}$ dans un voisinage de x_*
  - $\nabla^2 f(x_*)$ inversible
  - x_1 proche de x_*
- Attention : l'algorithme ne fait pas de distinction entre un minimum ou un maximum. Tous les points stationnaires "répulsifs" lui conviennent.
- L'algorithme fonctionne donc bien dans le voisinage d'un minimum local faisant les CS2 ($\nabla f(x_*) = 0$ et $\nabla^2 f(x_*) > 0$).
- Si $\nabla^2 f(x_k) > 0$, on obtient la même direction d_k que ci-dessus en résolvant le problème quadratique ci-dessous

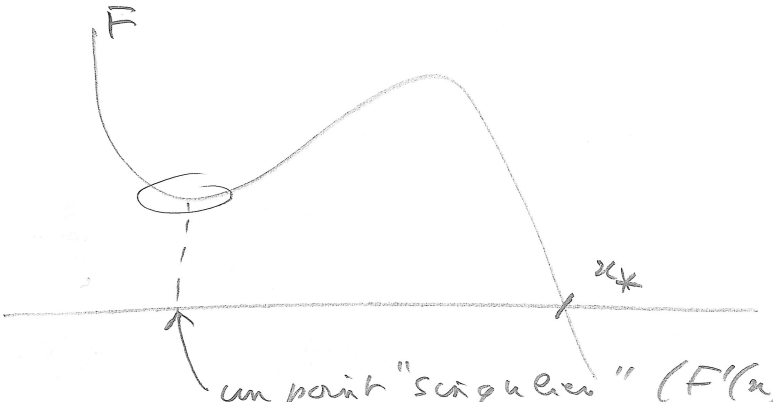
$$\min_d \; f(x_k) + \langle \nabla f(x_k), d \rangle + \frac{1}{2} \langle \nabla^2 f(x_k) d, d \rangle$$



[c] Globalisation pour système d'équations

(1) Remarques préliminaires

- Globaliser veut dire forcer la convergence même si x_1 est éloigné d'une "solution". (pas de rapport avec la recherche d'un minimum global)
- Il n'existe pas d'algorithme permettant de trouver un zéro de F quel que soit F et x_1. Le problème n'est de



un point "singulier" ($F'(x)$ n'est pas inversible) qui est attractif pour beaucoup d'algorithmes; en ce point on ne peut pas dire s'il faut partir à droite ou à gauche pour trouver un éventuel zéro, avec une information locale sur F en quantité finie

(exception : si F est très structurée, par ex un polynôme)

- On essaye seulement d'améliorer l'algorithme local précédent.
- On sait que l'algorithme du gradient "converge" ; donc on se ramène à un problème d'optimisation

(2) Fonction de moindres-carrés

- On remplace le problème

$$F(x) = 0 \qquad (1)$$

par celui mieux maîtrisé :

$$\min_x \left(\varphi(x) := \frac{1}{2} \| F(x) \|_2^2 \right)$$

- Observation cruciale : la direction de Newton d sur (1) est toujours une direction de descente de φ en x si $F(x) \neq 0$:

$$\varphi'(x) \cdot d = \underbrace{F(x)^T F'(x) d}^{= -F(x)}$$

$$= - \| F(x) \|_2^2$$

$$< 0 \quad \text{si } F(x) \neq 0.$$

(3) Globalisation par RL

- Donc, on peut chercher à faire décroître φ le long de la direction de Newton + recherche linéaire

- $x_{k+1} = x_k + \alpha_k d_k$ où $d_k = - F(x_k)^{-1} F(x_k)$ et le pas $\alpha_k > 0$ tel que

$$\varphi(x_k + \alpha_k d_k) \leq \varphi(x_k) + \omega \alpha_k \varphi'(x_k) \cdot d_k$$

ou

$$\boxed{\| F(x_k + \alpha_k d_k) \|_2^2 \leq (1 - 2\omega\alpha_k) \| F(x_k) \|_2^2} \qquad (4)$$

et α_k qui n'est pas trop petit (par exemple $\alpha_k = 2^{-i_k}$ où i_k est le premier entier ≥ 0 qui permet de vérifier (4))

Théo (convergence globale avec RL)

Si • $\mathrm{Cond}_2(F'(x_k)) := \|F'(x_k)^{-1}\|_2 \, \|F'(x_k)\|_2$ est borné

• la condition de Zoutendijk est vérifiée

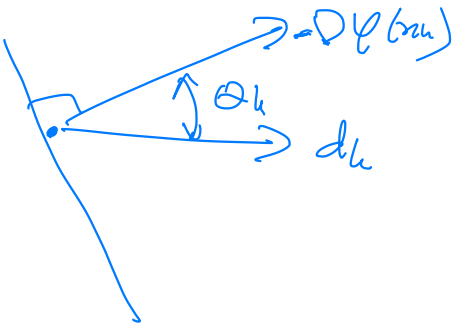
$$\sum_{k \geq 1} \|\nabla \varphi(x_k)\|^2 \cos^2 \theta_k < +\infty$$

alors 1) $\nabla \varphi(x_k) \to 0$

2) **si**, de plus, $(F'(x_k)^{-1})$ est bornée

alors $F(x_k) \to 0$

Demi 1) $\cos \theta_k \geq c > 0$?



$$\cos\theta_k = \frac{-\nabla\varphi(x_k)^T d_k}{\|\nabla\varphi(x_k)\| \, \|d_k\|}$$

$$= \frac{\|F(x_k)\|_2^2}{\|F'(x_k)^T F(x_k)\|_2 \, \|F'(x_k)^{-1} F(x_k)\|_2}$$

$$\leq \|F'(x_k)\|_2 \, \|F(x_k)\|_2 \qquad \leq \|F'(x_k)^{-1}\|_2 \, \|F(x_k)\|_2$$

$$\geqslant \frac{1}{\|F'(x_k)\|_2 \, \|F'(x_k)^{-1}\|_2} = \frac{1}{\mathrm{Cond}_2(F'(x_k))} \geqslant c$$

par (1)

2) $\nabla \varphi(x_k) = F'(x_k)^T F(x_k) \to 0$

$$\|F(x_k)\| = \|F'(x_k)^{-T} F'(x_k)^T F(x_k)\|$$

$$\leq \|F'(x_k)^{-1}\| \, \|\nabla \varphi(x_k)\|$$

$F'(x_k)^{-1}$ borné, $\nabla\varphi(x_k) \to 0$

$\to 0$

□

Remarques

(1) La méthodes des zoomides - concés ne permet que de trouver un point x_* tel que $\nabla \varphi(x_*) = 0$. Rien de mieux en toute généralité.

(2) Le résultat de convergence a des hypothèses qui excluent des situations fâcheuses qui se produisent de temps en temps :

- $x_k \longrightarrow \bar{x}$ (ou une sous-suite)
- $F'(\bar{x})$ n'est pas inversible
- $\bar{x}$ n'a aucune propriété particulière (en particulier $\nabla \varphi(\bar{x}) \neq 0$)

Un remède à cette dernière situation sont les régions de confiance ; c'est pour cela qu'elles ont été inventées.

D Globalisation en optimisation

- L'algorithme le plus classique est "Newton tronqué". Voici comment il fonctionne pour résoudre

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

- la direction de Newton $-\nabla^2 f(x)^{-1} \nabla f(x)$ n'est pas nécessairement une direction de descente de f en x.
- On construit une direction de descente en minimisant par gradient conjugué le problème quadratique associé

$$\min_d \quad \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d$$

La minimisation est partielle, tronquée, dès que l'on rencontre une direction à courbure négative, c-à-d

$$v^T \nabla^2 f(x) v < 0$$
$$(\text{ou} \leq \varepsilon \|v\|^2)$$

Alors le d trouvé est une direction de descente de f en x

- On fait de même RL le long de ce d.

3) Algorithmes de quasi-Newton en optimisation

A) Motivation

quasi- : préfixe attribué à une famille d'algorithmes, bien particulière, qui ne contient pas tous les algorithmes qui "ressemblent" à Newton

- défauts de l'algorithme de Newton
  - besoin de calculer les dérivées secondes
  - ne génère pas toujours des directions de descente
- qualités des algorithmes de quasi-Newton
  - remédient aux 2 problèmes précédents
  - garantissent la convergence superlinéaire (au lieu de quadratique)

B L'algorithme de BFGS

↓

(Broyden, Fletcher, Goldfarb, Shanno)

On prend

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

$d_k = ?$

Newton : $d_k = -\nabla^2 f(x_k)^{-1} g_k$ $\quad (g_k := \nabla f(x_k))$

Quasi-Newton : $d_k = -M_k^{-1} g_k$

avec $M_k \simeq \nabla^2 f(x_k)$ généré par l'algorithme

Schéma de l'algorithme $\quad (x_k, M_k) \longrightarrow (x_{k+1}, M_{k+1})$

1) arrêt si $g_k \simeq 0$
2) $d_k = -M_k^{-1} g_k$
3) RL de Wolfe (pour BFGS) $\longrightarrow \alpha_k > 0$
4) $x_{k+1} = x_k + \alpha_k d_k$
5) mise à jour de la matrice $M_k \rightsquigarrow M_{k+1}$

Comment calculer M_{k+1} ?

$$M_{k+1} \simeq \nabla^2 f(\cdot)$$

On utilise la variation du gradient (qui donne de l'information sur $\nabla^2 f$)

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

$$y_k := g_{k+1} - g_k$$

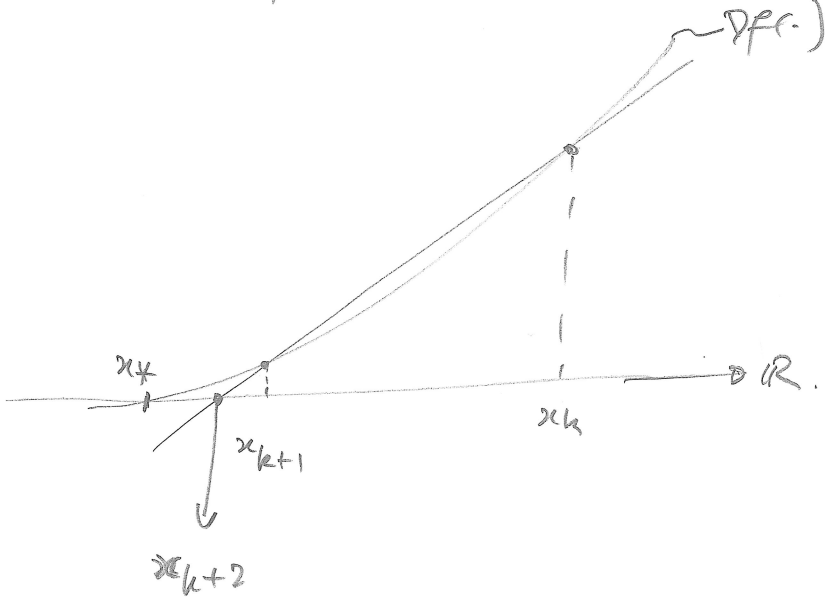
$$= \left[\int_0^1 \nabla^2 f(x_k + t\, s_k)\, dt \right] s_k \qquad (\text{si } f \in C^{2,1})$$

↑ hessien moyen entre x_k et x_{k+1}

⟹ on demande que M_{k+1} vérifie la même équation que le hessien moyen :

$$\boxed{y_k = M_{k+1} s_k}$$ équation de quasi-Newton

- ça ne permet pas de déterminer M_{k+1} si $n \geq 2$
- si $n = 1$, c'est la méthode de la sécante



On demande aussi que $M_{k+1} \in S^n$ (symétrie)
Comme le hessien

Par d'autres conditions sur M_{k+1} $\Rightarrow$ on prend M_{k+1} le plus proche de M_k et vérifiant les 2 contraintes précédentes :

$$\begin{cases} \min\limits_{M} \quad \psi(M, M_k) & \text{(écart entre } M \text{ et } M_k\text{)} \\ y_k = M s_k \\ M \in S^n \\ M \text{ définie positive} \leftarrow \text{cette contrainte est prise en compte par } \psi \end{cases}$$

$$\psi(M, M_k) = \underbrace{\operatorname{tr}\left(M_k^{-1/2} M M_k^{-1/2}\right) - \log\det\left(M_k^{-1/2} M M_k^{-1/2}\right)}_{\begin{cases} \geq 1 \\ = 1 \text{ ssi } M = M_k \end{cases}}$$

On peut calculer la solution analytiquement (sur papier) :

$$\boxed{M_{k+1} = M_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{M_k s_k s_k^T M_k}{s_k^T M_k s_k}}$$

formule de BFGS