Algorithmes de Newton

Soit l'équation $x^3 - 2x - 5 = 0$ dont on cherche une racine. Prenez un nombre comme 2, qui diffère de moins de 10 % de la vraie valeur d'une racine. Écrivez $x = 2 + d_1$ et remplacez x par $2 + d_1$ dans l'équation. Vous aurez $d_1^3 + 6d_1^2 + 10d_1 - 1 = 0$, dont il faut trouver la racine pour l'ajouter à 2. Négligez $d_1^3 + 6d_1^2$ à cause de sa petitesse ; il restera $10d_1 - 1 = 0$ ou $d_1 = 0.1$, ce qui est très près de la vraie valeur de d_1. C'est pourquoi, j'écris $d_1 = 0.1 + d_2$ et substituant comme auparavant j'ai $d_2^3 + 6.3d_2^2 + 11.23d_2 + 0.061 = 0$. Négligeant les deux premiers termes, il reste $11.23d_2 + 0.061 = 0$ ou $d_2 = -0.0054$ à peu près. [...] *Et je continue ainsi les opérations aussi longtemps qu'il convient.*

I. Newton (1736). Methodus fluxionum et serierum infinitorum. (Voir les notes en fin de chapitre.)

The central idea in this essay is that narcissism is an advantageous trait for succeeding in science. Scientists with a high ego are better able to convince others of the importance of their research. [...] Narcissists emerge as charismatic leaders but the cost of their attitude is invisible, paid by others.

B. Lemaitre [386].

On donne aujourd'hui le nom de *méthode de Newton* à toute approche algorithmique procédant par *linéarisation* des fonctions définissant le *système* dont on cherche une solution. C'est faire un grand honneur, peut-être excessif, à Isaac Newton, cet important contributeur de la science. Le terme *système* est pris ici dans un sens très large puisqu'il peut s'agir d'équations, d'inéquations, d'inclusions, d'équations différentielles ou aux dérivées partielles, d'inéquations variationnelles, *etc.* De même, le terme *linéarisation* doit être pris dans un sens étendu, car on utilise aussi la méthode de Newton pour résoudre des systèmes définis par des fonctions non différentiables dans le sens classique. On peut donc mesurer le chemin parcouru depuis l'algorithme proposé au XVII$^{\text{e}}$ siècle par Newton pour déterminer une racine d'un polynôme réel d'une variable réelle, décrit dans les quelques lignes de l'épigraphe de ce chapitre, alors que la notion de dérivée n'existait pas encore. Il aurait d'ailleurs été préférable d'utiliser le nom de Simpson pour décrire ces méthodes (voir les notes de fin de cha-

pitre), mais l'usage actuel en a décidé autrement. Nous aurions aussi pu utiliser la locution *méthodes de linéarisation*, mais nous n'avons pas franchi le pas et avons suivi la tradition.

Il y a de nombreuses monographies consacrées à l'algorithme de Newton ou à un aspect de cette approche par linéarisation (voir les notes en fin de chapitre), si bien que notre présentation ne pourra être que partielle, se concentrant sur des sujets qui nous paraissent essentiels ou en rapport direct avec l'esprit de cet ouvrage. Notre description commencera par le cas simple et instructif dans lequel on cherche à résoudre un système d'équations non linéaires, à en trouver un zéro (section 10.1.1). Ce cas est important en optimisation pour au moins deux raisons. D'abord il se présente lorsqu'on cherche à minimiser la fonction nulle sous des contraintes d'égalité. Par ailleurs, l'algorithme de Newton en optimisation sans contrainte est un cas particulier du précédent, si bien que certaines de ses propriétés, non attractives pour l'optimisation, trouveront leur origine dans le fait que cette approche est d'abord destinée à la résolution d'équations non linéaires. Nous verrons ensuite comment adapter l'algorithme à la minimisation de fonctions sans contrainte (section 10.1.2) ; le cas des problèmes avec contraintes sera examiné en détail au chapitre 15.

La propriété la plus attrayante de l'algorithme de Newton, qui en fait une référence, est sa convergence quadratique locale (théorèmes 10.2 et 10.3). Il a malheureusement aussi beaucoup de défauts ; nous les détaillerons. Comme remède à ces imperfections nous examinerons en détail les méthodes inexactes (section 10.2) et la globalisation de la convergence (section 10.3).

Connaissances supposées. Conditions d'optimalité pour les problèmes sans contrainte (section 4.2) ; algorithme du gradient conjugué (chapitre 8, utile pour l'algorithme de Newton tronqué à la section 10.3.1).

10.1 Méthodes locales

10.1.1 Systèmes d'équations

On s'intéresse ici à la recherche d'un *zéro* d'un système d'équations non linéaires, c'est-à-dire d'un point $x \in \mathbb{R}^n$ qui vérifie

$$F(x) = 0, \tag{10.1}$$

où $F : \mathbb{R}^n \to \mathbb{R}^n$ est une fonction différentiable. Il faut donc que $F_i(x) = 0$ pour tout $i = 1, \ldots, n$. Le système (10.1) étant formé de n équations aux n inconnues $x = (x_1, \ldots, x_n)$, il a quelques chances d'être bien posé.

L'algorithme de Newton génère une suite $\{x_k\}$ par une idée très simple, qui est illustrée à la figure 10.1 dans le cas où $n = 1$. On commence par linéariser l'équation en l'itéré courant x_k, ce qui donne la fonction $x \mapsto F(x_k) + F'(x_k) \cdot (x - x_k)$ dont le graphe est représenté par la ligne en tirets à la figure 10.1. Puis on cherche un zéro de cette fonction linéaire, s'il existe. C'est un opération simple puisqu'il suffit de résoudre un système linéaire. Ce zéro est l'itéré suivant x_{k+1}, qui est donc défini par

$$F(x_k) + F'(x_k) \cdot (x_{k+1} - x_k) = 0.$$

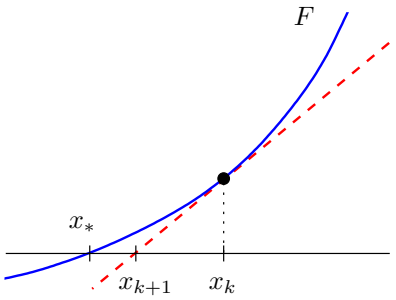


Fig. 10.1. Une itération de Newton

Cette équation peut certainement être résolue si $F'(x_k)$ est inversible. On renforcera l'analogie avec les méthodes à directions de descente du chapitre 6 en écrivant

$$x_{k+1} = x_k + d_k, \tag{10.2}$$

où d_k est la solution de l'*équation de Newton*, qui est le système linéaire suivant

$$F'(x_k)d_k = -F(x_k). \tag{10.3}$$

On peut maintenant décrire l'algorithme de Newton, que l'on qualifie de *local* car, comme on le verra, sa convergence n'est garantie que si le premier itéré est proche d'un zéro régulier de F.

Algorithme 10.1 (Newton local pour système non linéaire) On suppose qu'au début de l'itération k, on dispose d'un itéré $x_k \in \mathbb{R}^n$.

1. *Test d'arrêt.* Si $F(x_k) \simeq 0$, arrêt de l'algorithme.
2. *Direction.* Calculer d_k comme solution de (10.3).
3. *Nouvel itéré.* $x_{k+1} := x_k + d_k$.

Le coût de l'itération repose essentiellement sur l'évaluation de la jacobienne $F'(x_k)$ et sur la résolution du système linéaire (10.3) à l'étape 2. Cet algorithme ramène donc la résolution du système non linéaire (10.1) à une *suite* de systèmes linéaires, plus simples à résoudre.

L'intérêt principal de l'algorithme Newton est de générer des suites q-quadratiquement convergentes, c'est ce que nous allons montrer dans le théorème 10.2 ci-dessous. Les conditions assurant un tel comportement sont à peine plus fortes que celles requises pour que la méthode soit bien définie : il faut que F ait une dérivée lipschitzienne (alors que seule la dérivabilité de F est nécessaire à la définition de l'algorithme) et que F' soit inversible en la solution x_* recherchée (alors $F'(x_k)$ sera inversible pour un itéré x_k proche de x_*).

Le théorème suivant analyse la convergence d'une méthode un peu plus générale que l'algorithme 10.1, dans laquelle, à l'étape 2, la direction d_k est solution du système linéaire

$$M_k d_k = -F(x_k), \tag{10.4}$$

où M_k est une matrice inversible, pouvant être différente de $F'(x_k)$. Les méthodes de quasi-Newton entrent dans ce cadre (voir le chapitre 11).

Théorème 10.2 (convergence locale de l'algorithme de Newton) *On suppose que F a un zéro x_*, que F est de classe C^1 dans un voisinage Ω de x_* et que $F'(x_*)$ est inversible.*

1) *Alors, il existe $\varepsilon_x > 0$ et $\varepsilon_M > 0$ tels que si*

$$\|x_1 - x_*\| \leqslant \varepsilon_x \qquad \text{et} \qquad \|M_k - F'(x_k)\| \leqslant \varepsilon_M, \quad \forall k \geqslant 1,$$

l'algorithme de Newton avec d_k solution de (10.4), plutôt que de (10.3), est bien défini et génère une suite $\{x_k\}$ convergeant q-linéairement vers x_*.

2) *Si de plus*

$$\big(M_k - F'(x_*)\big)(x_k - x_*) = o(\|x_k - x_*\|),$$

alors la convergence est q-superlinéaire.

3) *Si de plus F' est lipschitzienne sur Ω et*

$$\big(M_k - F'(x_*)\big)(x_k - x_*) = O(\|x_k - x_*\|^2),$$

alors la convergence est q-quadratique.

Démonstration. On note $\beta := \|F'(x_*)^{-1}\|$ et on choisit $\varepsilon_M > 0$ tel que $\beta\varepsilon_M < 1$ et

$$r := \frac{3\beta\varepsilon_M}{1 - \beta\varepsilon_M} < 1.$$

On détermine ensuite $\varepsilon_x > 0$ tel que $\bar{B}(x_*, \varepsilon_x) \subseteq \Omega$ et tel que $\|x - x_*\| \leqslant \varepsilon_x$ implique que $\|F'(x) - F'(x_*)\| \leqslant \varepsilon_M$ (possible par la continuité de F').

Si une matrice M vérifie $\|M - F'(x_*)\| \leqslant \varepsilon_M$, alors $\|F'(x_*)^{-1}(M - F'(x_*))\| \leqslant \beta\varepsilon_M < 1$ et, par le lemme A.2 de perturbation de Banach, la matrice M est inversible et vérifie $\|M^{-1}\| \leqslant \beta/(1 - \beta\varepsilon_M)$. En appliquant cela à $M = M_k$ ou $M = F'(x)$, on trouve que, pour tout $k \geqslant 1$ et tout $x \in \bar{B}(x_*, \varepsilon_x)$, M_k et $F'(x)$ sont inversibles et

$$\|M_k^{-1}\| \quad \text{et} \quad \|F'(x)^{-1}\| \leqslant \frac{\beta}{1 - \beta\varepsilon_M}.$$

Dans ce cas, la formule (10.4) définit bien la direction d_k.

En utilisant $F(x_*) = 0$ et le fait que F est de classe C^1 sur $\bar{B}(x_*, \varepsilon_x)$ (ce qui autorise un développement de Taylor avec reste intégral), on a si $x_k \in \bar{B}(x_*, \varepsilon_x)$

$$\begin{aligned} x_{k+1} - x_* &= x_k - x_* + d_k \\ &= M_k^{-1}\big(M_k(x_k - x_*) - F(x_k)\big) \\ &= M_k^{-1}\big(M_k - F'(x_k)\big)(x_k - x_*) \\ &\quad + M_k^{-1}\int_0^1 \Big(F'(x_k) - F'(x_* + t(x_k - x_*))\Big)(x_k - x_*)\,\mathrm{d}t. \end{aligned}$$

En utilisant le fait que la norme d'une intégrale est plus petite que l'intégrale de la norme de l'intégrant, on en déduit que $\|x_{k+1} - x_*\| \leqslant r\|x_k - x_*\|$. Dès lors, par récurrence, toute la suite $\{x_k\} \subseteq \bar{B}(x_*, \varepsilon_x)$ si $x_1 \in \bar{B}(x_*, \varepsilon_x)$ (car $r \leqslant 1$). De plus $x_k \to x_*$ (car $r < 1$). Ceci démontre le point 1 du théorème.

Sous la condition additionnelle du point 2, l'estimation de l'erreur $x_{k+1} - x_*$ ci-dessus montre que $x_{k+1} - x_* = o(\|x_k - x_*\|)$, c'est-à-dire la convergence superlinéaire de $\{x_k\}$. Sous les conditions additionnelles du point 3, on trouve à partir de l'estimation de l'erreur $x_{k+1} - x_*$ ci-dessus que, pour une constante C, $\|x_{k+1} - x_*\| \leqslant C\|x_k - x_*\|^2$; on obtient la convergence quadratique de $\{x_k\}$. □

Le résultat de convergence ci-dessus s'applique directement à l'algorithme de Newton, c'est-à-dire lorsque $M_k = F'(x_k)$ pour tout $k \geqslant 1$. En particulier, on voit que sous les conditions de régularité de F spécifiées dans les trois parties du théorème, l'algorithme est bien défini et génère une suite convergeant quadratiquement, dès que le premier itéré x_1 est pris assez proche de x_*.

Le théorème de Kantorovitch offre une autre manière de montrer la convergence de l'algorithme de Newton. Il a la particularité intéressante de ne pas supposer l'existence d'un zéro de F, mais de l'affirmer. Ce résultat offre donc aussi un moyen de démontrer l'existence d'un zéro d'une équation non linéaire. Il est d'ailleurs apparenté à des théorèmes d'existence de points fixes (voir les notes en fin de chapitre). Le résultat s'exprime simplement : si x_1 est presqu'un zéro ($F(x_1) \simeq 0$) et si F' est inversible en x_1 et ne change pas trop vite, alors il doit y avoir un zéro dans un voisinage de x_1 ; de plus l'algorithme de Newton démarrant en x_1 converge vers ce zéro.

Théorème 10.3 (Kantorovitch) *Supposons que F soit différentiable sur un ouvert convexe $\Omega \subseteq \mathbb{R}^n$. On suppose également qu'en $x_1 \in \Omega$, $F'(x_1)$ est inversible, que $F'(x_1)^{-1}F'(\cdot)$ est lipschitzienne de module $L > 0$ sur Ω et que, pour $\delta := \|F'(x_1)^{-1}F(x_1)\|$ et $r := (1 - \sqrt{1-2\delta L})/L$, on a*

$$2\delta L \leqslant 1 \qquad \text{et} \qquad \bar{B}(x_1, r) \subseteq \Omega.$$

Alors,

1) *F a un zéro $x_* \in \bar{B}(x_1, r)$,*
2) *F n'a pas d'autre zéro que x_* dans $(\bar{B}(x_1, r) \cup B(x_1, r_+)) \cap \Omega$, où $r_+ := (1 + \sqrt{1-2\delta L})/L$,*
3) *l'algorithme de Newton démarrant en x_1 est bien défini et génère une suite $\{x_k\} \subseteq \bar{B}(x_1, r)$ convergeant vers x_*.*

Démonstration. □

Concluons cette section par une propriété de l'algorithme de Newton importante pour les applications : l'algorithme est invariant par changement de variables. De manière plus précise, supposons que l'on fasse le changement de variables

$$\tilde{x} = Ax,$$

où A est une matrice d'ordre n inversible. Soit $\tilde{F} = F \circ A^{-1}$ l'expression de F dans l'espace des $\tilde{x}$; donc $\tilde{F}(\tilde{x}) = F(x)$ si $\tilde{x}$ et x sont reliés par la relation ci-dessus. On a le résultat suivant.

Proposition 10.4 (invariance par changement de variables) *Dans les conditions décrites ci-dessus, si $\{x_k\}$ [resp. $\{\tilde{x}_k\}$] est la suite des itérés générés par l'algorithme de Newton pour résoudre le système non linéaire $F(x) = 0$ [resp. $\tilde{F}(\tilde{x}) = 0$] à partir d'un premier itéré x_1 [resp. $\tilde{x}_1 = Ax_1$], alors $\tilde{x}_k = Ax_k$ pour tout $k \geqslant 1$.*

Démonstration. Soit d_k la direction de Newton sur F en x_k et $\tilde{d}_k$ la direction de Newton sur $\tilde{F}$ en $\tilde{x}_k$. Si $\tilde{x}_k = Ax_k$, on a

$$\tilde{d}_k = -\tilde{F}'(\tilde{x}_k)^{-1}\tilde{F}(\tilde{x}_k) = -AF'(x)^{-1}F(x) = Ad_k,$$

car $\tilde{F}'(\tilde{x}_k) = F'(x_k)A^{-1}$ et $\tilde{F}(\tilde{x}_k) = F(x_k)$. On en déduit que

$$\tilde{x}_{k+1} = \tilde{x}_k + \tilde{d}_k = A(x_k + d_k) = Ax_{k+1}.$$

Le résultat s'en ensuit alors par récurrence. □

On obtient un résultat d'invariance analogue si, au lieu de pré-composer la fonction F par une application linéaire inversible, on la post-compose : $\tilde{F} = A \circ F$. Ces résultats nous montrent qu'il ne sert à rien de préconditionner l'algorithme de Newton par pré- ou post-composition avec une application linéaire inversible, puisque les itérés générés n'en seraient pas affectés. Un préconditionnement peut toutefois avoir une incidence en arithmétique flottante et dans les algorithmes de Newton tronqués, dans lesquels le système linéaire n'est résolu que partiellement (section **??**).

10.1.2 Optimisation

On considère le problème d'optimisation non linéaire sans contrainte suivant :

$$\begin{cases} \min f(x) \\ x \in \mathbb{R}^n, \end{cases} \tag{10.5}$$

dans lequel f est supposée régulière. Son équation d'optimalité s'écrit :

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

où $\nabla f(x)$ est le gradient de f en x pour un produit scalaire arbitraire donné (on le note $\langle \cdot, \cdot \rangle$). Il s'agit d'un système de n équations non linéaires à n inconnues, que l'on peut résoudre par l'algorithme de Newton de la section 10.1.1, avec $F \equiv \nabla f$. Dans ce cas, l'équation de Newton (10.3) s'obtient en linéarisant en x_k l'équation d'optimalité ci-dessus, qui s'écrit aussi $f'(x) \cdot h = 0$, pour tout $h \in \mathbb{R}^n$. Cela donne $f'(x_k) \cdot h + f''(x_k) \cdot (d_k, h) = 0$, pour tout $h \in \mathbb{R}^n$; ou encore

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k). \tag{10.6}$$

Dans (10.6), $\nabla^2 f(x_k)$ est donc la hessienne de f en x_k pour le produit scalaire ayant servi à calculer le gradient $\nabla f(x_k)$. On adapte ainsi aisément l'algorithme de la section 10.1.1 au cas de l'optimisation.

Algorithme 10.5 (Newton local en optimisation) On suppose qu'au début de l'itération k, on dispose d'un itéré $x_k \in \mathbb{R}^n$.

1. *Test d'arrêt.* Si $\nabla f(x_k) \simeq 0$, arrêt de l'algorithme.
2. *Direction.* Calculer d_k comme solution de (10.6).
3. *Nouvel itéré.* $x_{k+1} := x_k + d_k$.

Une autre approche conduisant au même résultat est la suivante. Étant donné l'itéré x_k, on cherche à trouver x_{k+1} en minimisant l'approximation quadratique de f. Ceci conduit au *problème quadratique osculateur* en x_k, qui est le problème en d suivant

$$\min_{d\in\mathbb{R}^n} \; f(x_k) + \nabla f(x_k)^\mathsf{T} d + \frac{1}{2} d^\mathsf{T} \nabla^2 f(x_k) d. \tag{10.7}$$

S'il a un *point stationnaire*, disons d_k, on prend alors $x_{k+1} = x_k + d_k$. Il est aisé de montrer qu'il s'agit du même algorithme de Newton : l'équation d'optimalité de (10.7) n'est autre que (10.6).

Il est important d'observer que l'algorithme de Newton construit des suites convergeant vers des points stationnaires, sans faire de distinction entre les minima ou les maxima, par exemple. Ceci est dû au fait qu'il est conçu pour trouver des zéros de $\nabla f(x) = 0$. Par conséquent, sans modification adéquate, si le premier itéré est proche d'un point stationnaire « régulier », la suite générée convergera vers ce point stationnaire. On comprend que, si l'on cherche à minimiser f, converger vers un maximum local n'est pas une propriété satisfaisante. Il sera donc nécessaire de modifier l'algorithme de Newton de manière à le contraindre à éviter les points stationnaires qui ne sont pas des minima. Ce n'est pas une tâche facile : si $\nabla f(x_1) = 0$, la direction de Newton est nulle et donc l'itéré suivant x_2 est identique au premier ! Cette question est toujours un objet d'études. Nous en reparlerons aux sections 10.3.1 et 10.3.2.

10.1.3 Défauts et remèdes

Les inconvénients de la méthode de Newton pour résoudre des systèmes d'équations non linéaires [resp. des problèmes d'optimisation] sont bien connus :

1. il faut calculer les dérivées premières de F [resp. les dérivées secondes de f], ce qui peut être coûteux en temps de calcul (n^2 éléments à évaluer), en effort humain (l'expression analytique de ces dérivées n'est pas toujours simple à obtenir) et en espace mémoire ;
2. l'algorithme n'est pas globalement convergent (si le premier itéré est éloigné d'une solution, le comportement des itérés suivants est souvent erratique) ;
3. l'algorithme n'est pas nécessairement défini aux points x où $F'(x)$ [resp. $\nabla^2 f(x)$] est singulière ;
4. pour les problèmes d'optimisation, si f n'est pas fortement convexe, l'algorithme ne génère pas nécessairement des directions de descente de f ;

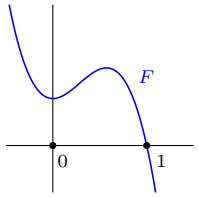
5. un système linéaire d'ordre n doit être résolu à chaque itération.

Voilà bien des défauts pour un algorithme aux propriétés locales tant souhaitées (il converge localement quadratiquement). Dans les sections suivantes, nous étudions des modifications de la méthode de Newton qui remédient partiellement à ces points faibles, tout en essayant de conserver son intérêt majeur qui est de générer des suites convergeant rapidement.

Les remèdes foisonnent et il n'est pas aisé de les exposer de manière concise. On peut en effet s'intéresser à la résolution de systèmes non linéaires ou à l'optimisation ; la globalisation de la convergence peut se faire par recherche linéaire, par région de confiance ou des méthodes de suivi de chemin ; le système de Newton peut être résolu exactement ou de manière approchée, par des méthodes directes ou itératives ; les algorithmes peuvent faire l'effort de ne pas utiliser la transposée de la jacobienne $F'(x)$ ou pas. Voilà donc beaucoup de possibilités à décrire et elles peuvent toutes être intéressantes en fonction du problème à résoudre. Nous serons brefs sur certaines approches si elles peuvent se déduire de méthodes déjà décrites ailleurs. Par ailleurs, nous ne considérerons pas les cas singuliers, où la jacobienne n'est pas inversible en la solution, où la fonction est non lisse, où la solution recherchée n'est pas isolée, *etc*, qui sont tous très importants pour pouvoir aborder des problèmes plus généraux que les deux considérés ici (annuler une fonction non linéaire et l'optimisation), comme les problèmes d'inéquations variationnelles ou de complémentarité, l'optimisation sous contrainte, *etc*.

Mais soyons clair : nonobstant cette abondance, il n'y a pas d'algorithme newtonien qui garantisse la convergence vers un zéro d'une fonction non linéaire F arbitraire quel que soit l'itéré initial. Cependant, les techniques numériques que nous allons présenter dans les sections suivantes améliorent grandement les qualités (efficacité et robustesse) des algorithmes en pratique, si bien qu'on ne peut les négliger. La difficulté se rencontre déjà en dimension 1, pour la fonction suivante

$$F(x) = \tfrac{1}{2} + 3x^2 - \tfrac{7}{2}x^3, \tag{10.8}$$



laquelle a un unique zéro en $x = 1$. Si l'on prend comme itéré initial $x_0 = 0$, les algorithmes échouent lamentablement. Il faut noter que la jacobienne de F y est nulle et donc que la direction de Newton n'y est pas définie. Par ailleurs, la fonction $x \mapsto |F(x)|$ a un minimum local en zéro, ce qui rend ce point attrayant aux yeux de beaucoup d'algorithmes. En réalité, il n'y a pas aujourd'hui de remède universel à cette difficulté fondamentale, qui trouve son origine dans le fait que l'algorithme de Newton est une méthode locale (en chaque itéré, elle n'utilise que les valeurs de F et de sa dérivée) alors que la détermination d'un zéro de F est de nature globale (dans l'exemple ci-dessus, en n'examinant F que dans le voisinage de 0, il est très difficile de savoir s'il faut s'éloigner de 0 en partant vers la gauche ou vers la droite — ce n'est pas impossible de faire le bon choix lorsque F est analytique et que l'on dispose des dérivées de tous ordres de F en zéro, mais en pratique il n'est possible d'utiliser qu'une quantité finie d'information).

On notera enfin que la situation est beaucoup plus favorable si les composantes de F sont des *polynômes*. La nature globale des zéros de F ne pose alors pas de difficulté aux techniques algébriques (via l'utilisation de *base de Gröbner* par exemple [147]) ou numériques (en utilisant des méthodes d'*optimisation globale* [375]) pourvu que le nombre de variables ou le degré des polynômes reste faible.

10.2 Méthodes inexactes ▲

10.2.1 Systèmes d'équations

Dans les problèmes de grande taille, il peut être coûteux de résoudre les systèmes linéaires de Newton (10.3) avec précision. Souvent même, une résolution exacte n'est pas possible, si bien qu'il faut définir un seuil de tolérance. Par ailleurs, on conviendra également qu'un calcul précis, qui fait entièrement confiance à la linéarisation de F, n'est probablement pas utile lorsque l'itéré courant x_k est éloigné d'un zéro de F, parce qu'en de tels points la direction de Newton d_k est généralement grande et qu'alors $F(x_k + d_k)$ est souvent éloigné de la valeur nulle prédite par le modèle linéarisé de F. Si le nombre de variables est important, les systèmes linéaires sont en général résolus par des méthodes itératives, dont l'arrêt est contrôlé par un test ; il est alors naturel d'avoir un test permissif, autorisant un important résidu $F(x_k) + F'(x_k)d_k$, lorsque $F(x_k)$ est grand et un test plus contraignant lorsque $F(x_k)$ est petit. Ces différentes considérations conduisent à la notion suivante.

On parle de *méthode de Newton inexacte* lorsque l'algorithme cherche à calculer des directions d_k, dites *de Newton inexactes*, vérifiant la condition suivante :

$$\|F(x_k) + F'(x_k)d_k\| \leqslant \eta_k \|F(x_k)\|, \tag{10.9}$$

où $\|\cdot\|$ est une norme arbitraire et $\eta_k \in [0, 1[$ est appelé le *facteur d'inexactitude*. Il est naturel de prendre $\eta_k < 1$ de manière à ne pas accepter une direction nulle. Par ailleurs, la direction de Newton, quand elle existe, annule le membre de gauche, si bien que (10.9) peut être vu comme une condition acceptant davantage de directions que celle de Newton. Comme annoncé, la condition (10.9) contrôle la précision avec laquelle il faut résoudre le système linéaire de Newton (10.3) au moyen de la grandeur $\|F(x_k)\|$, qui mesure la précision avec laquelle l'itéré courant résout le système non linéaire (10.1).

La condition (10.9) n'est pas nécessairement réalisable. La proposition suivante montre que l'on peut trouver une direction de Newton inexacte pour une norme arbitraire, essentiellement lorsque la direction de Newton elle-même existe, c'est-à-dire lorsque $F(x_k) \in \mathcal{R}(F'(x_k))$. Cependant, si la direction de Newton n'existe pas, on pourra parfois trouver une direction de Newton inexacte pour une norme particulière et un facteur d'inexactitude assez grand. Par exemple, si $F : \mathbb{R}^2 \to \mathbb{R}^2$ est la fonction linéaire définie par

$$F(x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} x,$$

on a $F(0) \notin \mathcal{R}(F'(0))$, alors que $\|F(0) + F'(0)d\|_2 \leqslant \eta \|F(0)\|_2$ est réalisable pourvu que $\eta \in [\sqrt{2}/2, 1[$.

Proposition 10.6 (existence d'une direction de Newton inexacte) *Supposons que F soit différentiable en un itéré x_k tel que $F(x_k) \neq 0$. Alors, les propriétés suivantes sont équivalentes :*

(i) $F(x_k) \in \mathcal{R}(F'(x_k))$,
(ii) *pour toute norme $\|\cdot\|$ et tout $\eta_k \in [0,1[$, il existe un d_k vérifiant (10.9),*
(iii) *pour toute norme $\|\cdot\|$ associée à un produit scalaire, il existe un $\eta_k \in [0,1[$ et un d_k vérifiant (10.9).*

Démonstration. $[(i) \Rightarrow (ii)]$ Si $F(x_k) \in \mathcal{R}(F'(x_k))$, on peut trouver un d_k tel que $F(x_k) + F'(x_k)d_k = 0$. Cette direction de Newton d_k vérifie évidemment (10.9) quels que soient la norme et le $\eta_k \in [0,1[$.

$[(ii) \Rightarrow (iii)]$ Évident.

$[(iii) \Rightarrow (i)]$ Si $F(x_k) \notin \mathcal{R}(F'(x_k))$, on peut construire une base de $\mathbb{R}^n$ en complétant une base de $\mathcal{R}(F'(x_k))$ à laquelle on joint le vecteur $F(x_k)$. On prend sur $\mathbb{R}^n$ le produit scalaire $\langle\cdot,\cdot\rangle$ associé à cette base, qui est le produit scalaire euclidien des coordonnées dans cette base, et la norme associée, que l'on note $\|\cdot\|$. Alors $F(x_k)$ est orthogonal à $\mathcal{R}(F'(x_k))$, ce qui s'écrit

$$F'(x_k)^* F(x_k) = 0.$$

On en déduit que $d = 0$ minimise la fonction convexe différentiable $d \mapsto \|F(x_k) + F'(x_k)d\|$, c'est-à-dire que $\|F(x_k) + F'(x_k)d\| \geqslant \|F(x_k)\|$ pour tout $d \in \mathbb{R}^n$. Dès lors, quel que soit $\eta_k \in [0,1[$, (10.9) n'est pas réalisable pour la norme $\|\cdot\|$. □

10.3 Globalisation de la convergence

Grâce au théorème 10.2, on sait que la convergence de l'algorithme de Newton local 10.1 est garantie si l'itéré initial est « suffisamment » proche d'une solution (un zéro de F ou un minimum de f). Si le premier itéré est « éloigné » d'une solution, l'algorithme pourra générer une suite au comportement erratique, qui pourra accidentellement se retrouver dans le voisinage d'une solution et converger vers celle-ci, mais qui le plus souvent divergera (voir [43 ; 2012] pour un cas de cyclage, avec une fonction non différentiable, que l'on pourrait facilement lisser). En général, il est difficile de dire si un itéré initial est dans le voisinage d'une solution qui garantit la convergence de l'algorithme de Newton. Il est donc important de disposer de techniques permettant d'éviter le comportement désordonné indésirable probable de ses suites générées.

On entend par *globalisation de la convergence* de l'algorithme de Newton toute technique permettant d'améliorer la convergence des itérés vers une solution du problème, même si l'itéré initial est éloigné d'une solution. Cette notion n'a donc pas de lien avec la recherche d'un minimum global d'une fonction.

10.3.1 Recherche linéaire

Newton inexact ▲

On appelle *fonction de mérite*, toute fonction réelle qui atteint un minimum (si possible global) en une solution du problème que l'on cherche à résoudre. Si le problème considéré est celui de la minimisation sans contrainte (10.5), la fonction de mérite idéale est la fonction coût elle-même. Dans le cas où l'on recherche un zéro de l'équation (10.1), une fonction de mérite naturelle est la *fonction de moindres-carrés* $\varphi : \mathbb{R}^n \to \mathbb{R}$, définie par

$$\varphi(x) = \frac{1}{2}\|F(x)\|_2^2. \tag{10.10}$$

Le facteur $\frac{1}{2}$ n'est utile que pour simplifier l'expression de la dérivée ; la norme ℓ_2 et le carré assurent quant à eux la différentiabilité de φ.

La fonction φ atteint une valeur optimale nulle en toute solution de (10.1). Elle peut toutefois avoir des minima locaux qui ne sont pas solutions de (10.1). Ceux-ci vérifient la condition d'optimalité du premier ordre

$$F'(x)^T F(x) = 0.$$

Ces points stationnaires seront donc des solutions de (10.1) si $F'(x)$ y est inversible, ce qui est loin d'être toujours le cas. Ce raisonnement simple met en évidence le rôle critique que jouera, dans cette approche, le lieu des points où la jacobienne $F'(x)$ est singulière :

$$\mathcal{S} := \{x \in \mathbb{R}^n : F'(x) \text{ est singulière}\}.$$

La situation est cependant plus compliquée : même si $\mathcal{S}$ est vide, certains algorithmes utilisant φ comme fonction de mérite pourront rencontrer des difficultés lorsque la fonction $x \mapsto F'(x)^{-1}$ n'est pas bornée. Autrement dit, une matrice $F'(x)$ « singulière à l'infini » peut aussi être une source de difficultés.

Les techniques de globalisation de la convergence utilisent souvent de telles fonctions de mérite, car on sait comment forcer la convergence d'itérés vers des minima locaux de fonctions, par recherche linéaire (chapitre 6) ou par régions de confiance (chapitre 9), alors que l'on ne connaît pas de méthode systématique permettant de trouver un zéro d'une fonction. On peut dire qu'en adoptant une telle approche, ces techniques renoncent à trouver un zéro de F et se contentent d'un point stationnaire ou d'un minimum local de φ. Dans ce cadre, cette recherche de zéro revient à trouver un minimum global de la fonction φ ci-dessus, tâche considérée aujourd'hui comme très difficile, parfois impossible, et de toutes façons très coûteuse en toute généralité.

Nous nous intéressons donc, dans cette section, à la globalisation de la convergence de l'algorithme de Newton pour résoudre le système (10.1), au moyen de la fonction de mérite φ définie ci-dessus. Cette approche permet d'ailleurs de résoudre de manière approchée l'équation de Newton (10.3). On se satisfait en effet d'une direction d_k qui vérifie

$$\|F(x_k) + F'(x_k)d_k\|_2 \leqslant \eta_k \|F(x_k)\|_2, \tag{10.11}$$

où $0 \leqslant \eta_k \leqslant \eta < 1$ (η est une constante). En général, on prend η_k proche de 1 lorsque x_k est loin d'une solution, de manière à ne pas passer trop de temps dans la résolution d'un système linéaire qui n'est sans doute pas un bon modèle de F dans ce cas, et l'on

prend η_k proche de zéro lorsque x_k se rapproche d'une solution, de manière à bénéficier de la convergence rapide de l'algorithme de Newton dans le voisinage d'une solution. Rappelons que le fait que l'on puisse trouver une direction d_k vérifiant (10.11) cache une hypothèse sur $F'(x_k)$; voir la proposition 10.6.

A priori, on ne voit pas pourquoi la direction de Newton (inexacte) serait une direction de descente de φ, laquelle est définie de manière naturelle, mais sans lien évident avec l'algorithme de Newton. Le fait qu'il en soit ainsi est le premier miracle du couple Newton-φ (voir la proposition 10.15 pour le second).

Proposition 10.7 (descente) *Si $F(x_k) \neq 0$, toute direction d_k vérifiant (10.11) est une direction de descente (non nulle) de φ en x_k car on a*

$$\nabla\varphi(x_k)^{\mathsf{T}} d_k = F'(x_k)^{\mathsf{T}} F(x_k) d_k \leqslant -2(1-\eta_k)\varphi(x_k) < 0. \tag{10.12}$$

Démonstration. On a en effet $\nabla\varphi(x_k) = F'(x_k)^{\mathsf{T}} F(x_k)$. Puis en utilisant l'inégalité de Cauchy-Schwarz :

$$\nabla\varphi(x_k)^{\mathsf{T}} d_k = F(x_k)^{\mathsf{T}} \left(F(x_k) + F'(x_k)d_k\right) - \|F(x_k)\|_2^2 \leqslant -(1-\eta_k)\|F(x_k)\|_2^2 < 0.$$

Forcément, comme toute direction de descente, d_k ne peut être nulle. □

Comme on cherche à annuler F et qu'un zéro de F est un minimum global de φ, la propriété remarquable précédente légitime le fait de trouver l'itéré suivant x_k en faisant de la recherche linéaire le long de d_k. C'est ce que fait l'algorithme ci-dessous, qui porte le nom d'*algorithme de Newton inexact*, malgré la connotation péjorative de cette appellation. Cette approche, que l'on retrouvera pour l'algorithme de Newton en optimisation, semble providentielle. Nous verrons cependant qu'elle a ses propres limites.

Algorithme 10.8 (Newton inexact) On suppose qu'au début de l'itération k, on dispose d'un itéré $x_k \in \mathbb{R}^n$.

1. *Test d'arrêt.* Si $F(x_k) \simeq 0$, arrêt de l'algorithme.
2. *Direction.* Calculer d_k vérifiant (10.11). Si ce n'est pas possible l'algorithme échoue.
3. *Recherche linéaire.* Déterminer un pas $\alpha_k > 0$ « suffisamment grand » le long de d_k de manière à faire décroître φ « suffisamment ».
4. *Nouvel itéré.* $x_{k+1} := x_k + \alpha_k d_k$.

La description de la recherche linéaire utilisée à l'étape 3 est vague et sera précisée dans les résultats de convergence ci-dessous. Il est courant cependant d'utiliser la règle d'Armijo (section 6.3.3) : pour ω et $\beta \in]0,1[$, le pas α_k est pris égal à $\beta^{i_k}\alpha_k^1$ où α_k^1 plus grand qu'une constante strictement positive et i_k est le plus petit entier positif tel que

$$\varphi(x_k + \alpha_k d_k) \leqslant \varphi(x_k) - 2\omega\alpha_k(1-\eta_k)\varphi(x_k). \tag{10.13}$$

En utilisant la proposition 10.7, on voit facilement qu'un tel pas existe.

Le premier énoncé de convergence globale que nous donnons ci-après montre que les points d'adhérence *réguliers* (dans un sens précisé dans l'énoncé) de la suite $\{x_k\}$ générée par l'algorithme 10.8 sont des zéros de F. Il faut se garder de penser que la question de la convergence globale de l'algorithme de Newton est réglée avec ce résultat, car il se peut très bien que de tels points stationnaires réguliers n'existent pas et que la suite générée converge vers un point non régulier qui n'est pas un zéro de F. Néanmoins, un tel résultat est une première indication sur la bonne conception de l'algorithme et nous l'énonçons et le démontrons pour cette raison.

Proposition 10.9 (Newton inexact et points d'adhérence) *On considère l'algorithme de Newton inexact 10.8 pour résoudre le système $F(x) = 0$ où $F : \mathbb{R}^n \to \mathbb{R}^n$ est différentiable avec $F'(x_k)$ inversible en tout itéré x_k généré par l'algorithme. On suppose que l'algorithme utilise la règle de recherche linéaire d'Armijo décrite autour de (10.13). Alors, s'il existe un point d'adhérence $\bar{x}$ de $\{x_k\}$ tel que $F'(\bar{x})$ est inversible et si F' y est continue, il s'ensuit que $\varphi(x_k) \to 0$ et $F(\bar{x}) = 0$.*

Démonstration. Si $\bar{x}$ est un point d'adhérence de $\{x_k\}$, il existe une sous-suite d'indices $\mathcal{K}$ tel que $x_k \to \bar{x}$ lorsque $k \to \infty$ dans $\mathcal{K}$.

Montrons que $\{d_k\}_{k\in\mathcal{K}}$ *est bornée* (c'est une conséquence de la régularité de $\bar{x}$). On raisonne par l'absurde, en supposant que $\{d_k\}$ n'est pas bornée. Alors, en extrayant une sous-suite au besoin, on peut supposer que $\|d_k\| \to \infty$ et $d_k/\|d_k\| \to d \neq 0$ lorsque $k \to \infty$ dans $\mathcal{K}$. En divisant les deux membres de l'inégalité (10.11) par $\|d_k\|$ et en passant à la limite lorsque $k \to \infty$, on trouve que $F'(\bar{x})d = 0$, ce qui contredit l'inversibilité supposée de $F'(\bar{x})$ puisque $d \neq 0$.

Par la règle d'Armijo, la suite $\{\varphi(x_k)\}$ est décroissante. Comme elle est aussi borné inférieurement (par zéro), elle converge. Alors, la règle d'Armijo et $\eta_k \leqslant \eta < 1$ impliquent que

$$\alpha_k\varphi(x_k) \to 0.$$

On poursuit en examinant deux cas complémentaires.

1) Supposons d'abord que $\alpha_k \not\to 0$. Alors, $\varphi(x_k) \to 0$ pour une sous-suite d'indices tendant vers l'infini. Par la décroissance de la suite $\{\varphi(x_k)\}$, on voit que toute la suite $\{\varphi(x_k)\} \to 0$. Dès lors, tout point d'adhérence $\bar{x}$ de $\{x_k\}$ vérifie $\varphi(\bar{x}) = 0$ ou $F(\bar{x}) = 0$.

2) Considérons à présent le cas plus difficile où $\alpha_k \to 0$. On peut supposer que $\alpha_k < 1$, ce qui veut dire que le pas $\hat{\alpha}_k := \alpha_k/\beta$ n'est pas accepté par la règle d'Armijo ou encore, qu'au point $\hat{x}_k := x_k + \hat{\alpha}_k d_k$, on a

$$\varphi(\hat{x}_k) > \varphi(x_k) - 2\omega\hat{\alpha}_k(1-\eta_k)\varphi(x_k). \tag{10.14}$$

Notons que $\hat{\alpha}_k \to 0$ et que, par la bornitude de $\{d_k\}_{k\in\mathcal{K}}$, $\hat{x}_k \to \bar{x}$ pour $k \to \infty$ dans $\mathcal{K}$. On peut estimer l'écart $\varphi(\hat{x}_k) - \varphi(x_k)$ comme suit. Par le théorème des accroissements finis (corollaire C.13), on a

$$\|F(\hat{x}_k) - F(x_k) - F'(x_k)(\hat{x}_k - x_k)\| \leqslant \left(\sup_{z \in]x_k, \hat{x}_k[} \|F'(z) - F'(x_k)\| \right) \|\hat{x}_k - x_k\|.$$

Par la continuité supposée de F' en $\bar{x}$, le facteur entre parenthèses du membre de droite tend vers zéro quand $k \to \infty$ dans $\mathcal{K}$. En utilisant $\hat{x}_k - x_k = \hat{\alpha}_k d_k$, on obtient $F(\hat{x}_k) = F(x_k) + \hat{\alpha}_k F'(x_k) d_k + o(\hat{\alpha}_k)$ et donc

$$\begin{aligned} \varphi(\hat{x}_k) &= \varphi(x_k) + \hat{\alpha}_k F(x_k)^{\mathsf{T}} F'(x_k) d_k + o(\hat{\alpha}_k) \\ &\leqslant \varphi(x_k) - 2\hat{\alpha}_k (1 - \eta_k) \varphi(x_k) + o(\hat{\alpha}_k) \qquad [(10.12)]. \end{aligned}$$

Alors (10.14) conduit à

$$0 \leqslant 2(1 - \omega)\hat{\alpha}_k (1 - \eta_k) \varphi(x_k) \leqslant o(\hat{\alpha}_k).$$

En divisant chaque membre de ces inégalités par $\hat{\alpha}_k > 0$, en utilisant $\omega < 1$, en extrayant une sous-suite convergente de $\{\eta_k\}_{k \in \mathcal{K}} \subseteq]0, \eta]$ et en passant à la limite lorsque $k \to \infty$ dans $\mathcal{K}$, on obtient que $\varphi(\bar{x}) = 0$ ou $F(\bar{x}) = 0$. □

La recherche linéaire est considérée comme raisonnable dans le résultat de convergence qui suit, si elle permet d'obtenir la condition de Zoutendijk (6.19). La règle d'Armijo (algorithme 6.3) avec le pas initial $\alpha_k^1 = 1$ est souvent utilisée. Selon le chapitre 6, il faut souvent que φ soit $\mathcal{C}^{1,1}$ pour que la condition de Zoutendijk soit vérifiée par les recherches linéaires qui y sont étudiées. Dès lors, exiger la condition de Zoutendijk cache une hypothèse de régularité sur F.

Proposition 10.10 (convergence globale de Newton inexact) *Considérons l'algorithme de Newton inexact 10.8, avec une recherche linéaire vérifiant la condition de Zoutendijk (6.19), et supposons qu'il génère une suite $\{x_k\}$ telle que le conditionnement $\kappa_2(F'(x_k))$ soit borné. Alors*
1) $\nabla\varphi(x_k) \to 0$,
2) *si, de plus, la suite $\{F'(x_k)^{-1}\}$ est bornée, alors $F(x_k) \to 0$.*

Démonstration. Comme $\varphi(x_k)$ est bornée inférieurement, la proposition 6.8 montre que (6.21) a lieu. Le point 1 sera démontré si l'on prouve que le cosinus de l'angle θ_k entre $\nabla\varphi(x_k) = F'(x_k)^{\mathsf{T}} F(x_k)$ et $-d_k$ est uniformément positif. On a par (10.11)

$$\|d_k\|_2 \leqslant \|F'(x_k)^{-1}\|_2 \, \|F'(x_k) d_k\|_2 \leqslant (1 + \eta_k) \|F'(x_k)^{-1}\|_2 \, \|F(x_k)\|_2.$$

Dès lors, si C est une borne sur $\kappa_2(F'(x_k))$, on a

$$\cos\theta_k = \frac{-\nabla\varphi(x_k)^{\mathsf{T}} d_k}{\|\varphi(x_k)\|_2 \, \|d_k\|_2} \geqslant \frac{1 - \eta_k}{1 + \eta_k} \frac{1}{\|F'(x_k)\|_2 \, \|F'(x_k)^{-1}\|_2} \geqslant \frac{1 - \eta}{2C}.$$

Pour le point 2, on déduit de $\nabla\varphi(x_k) = F'(x_k)^{\mathsf{T}} F(x_k) \to 0$ et du caractère borné de $\{F'(x_k)^{-1}\}$ que

$$\|F(x_k)\|_2 \leqslant \|F'(x_k)^{-\mathsf{T}}\|_2 \, \|F'(x_k)^{\mathsf{T}} F(x_k)\|_2 \to 0.$$ □

Proposition 10.11 (convergence locale de Newton inexact) *On suppose que F a un zéro x_*, que F est de classe C^1 dans un voisinage Ω de x_* et que $F'(x_*)$ est inversible. On considère l'algorithme de Newton inexact 10.8, avec pas unité. Alors ...*

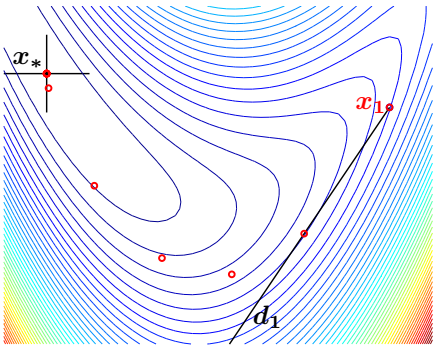
DÉMONSTRATION. □

D'après la proposition 10.10, tout se passe de manière très satisfaisante si $F'(x_k)$ forme une suite bornée d'inverses bornés (il ne suffit pas que $F'(x_k)$ soit inversible, comme le montrera l'exemple 10.16). Cette hypothèse est vérifiée dans beaucoup de problèmes, mais il n'est pas difficile de la violer et de piéger l'algorithme 10.8. C'est ce que l'on s'attache à mettre en évidence dans les exemples ci-dessous, qui sont des variations autour de l'exemple très favorable suivant.

Exemple 10.12 On considère la fonction $F : \mathbb{R}^2 \to \mathbb{R}^2$ définie par

$$F(x) = \begin{pmatrix} x_1 \\ -(x_1-2)^2 + x_2 + 4 \end{pmatrix}$$

$$F'(x) = \begin{pmatrix} 1 & 0 \\ -2(x_1-2) & 1 \end{pmatrix}.$$



La fonction F a un unique zéro en $x_* = 0$. □

La proposition 10.10 a mis en évidence le rôle joué par $F'(x)$. Si l'algorithme 10.8 est utilisé dans l'exemple ci-dessus, on a $\|F(x_k)\|_2 \leqslant \|F(x_1)\|_2$, ce qui implique que $\{x_k\}$ est bornée et donc aussi $\{F'(x_k)\}$ et $\{F'(x_k)^{-1}\}$. D'après la proposition 10.10, $F(x_k) \to 0$ et même $x_k \to x_*$ si l'on reprend le raisonnement de la démonstration du théorème **??**. C'est bien ce que l'on observe dans le tracé à droite ci-dessus, dans lequel on a indiqué les itérés générés par l'algorithme 10.8 avec recherche linéaire inexacte, les courbes de niveaux de φ et la direction de Newton d_1 en x_1.

Exemple 10.13 Dans cet exemple, on modifie le terme x_2 de $F_2(x)$ de l'exemple précédent de manière à introduire une singularité dans F' :

$$F(x) = \begin{pmatrix} x_1 \\ -(x_1-2)^2 + (x_2-1)^2 + 3 \end{pmatrix}, \quad F'(x) = \begin{pmatrix} 1 & 0 \\ -2(x_1-2) & 2(x_2-1) \end{pmatrix}. \tag{10.15}$$

La fonction F a toujours un zéro en $x_* = 0$ (et un autre en $(0, 2)$), mais $F'(x)$ est singulière sur la droite $\mathcal{S} := \{x \in \mathbb{R}^2 : x_2 = 1\}$. □

Évidemment, l'algorithme de Newton n'est pas défini en un $x \in \mathcal{S}$, mais la situation est bien plus délicate que cela. Comme le montre le tracé de gauche à la figure 10.3.1, ce lieu de singularités $\mathcal{S}$ est attrayant pour les itérés, qui peuvent s'y précipiter en

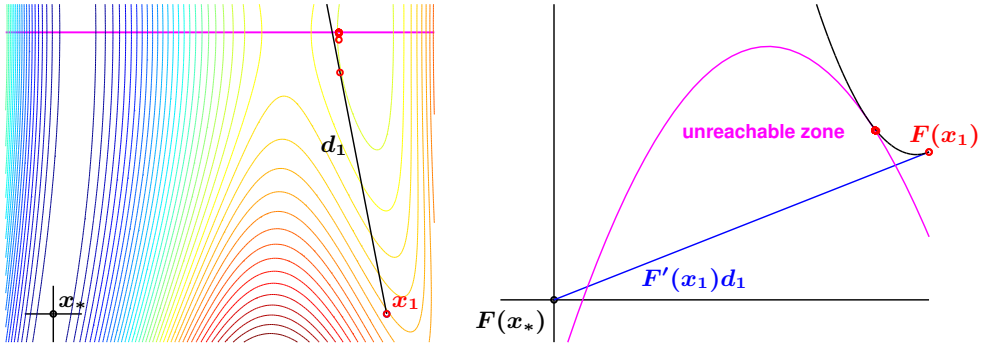


Fig. 10.2. Itérés de Newton avec recherche linéaire pour le système (10.15)

quelques itérations. Les courbes de niveau de φ apportent un premier éclairage sur ce comportement indésirable : les itérés sont piégés dans un bassin de φ, qui ne contient pas x_* et qu'ils ne peuvent quitter car $\varphi(x_k)$ décroît à chaque itération. Plus étrange, $\nabla\varphi(\bar{x}) \neq 0$ au point $\bar{x}$ vers lequel les itérés convergent. Un autre éclairage est apporté par l'observation de l'image $F(x_k)$ des itérés dans l'espace image (tracé de droite). Dans cet exemple, F n'est pas surjective.

Exemple 10.14 Dans ce dernier exemple, on modifie le terme x_2 de $F_2(x)$ de l'exemple 10.12 de manière à rendre $F'(x)$ inversible en tout $x \in \mathbb{R}^2$, mais d'inverse non borné :

$$F(x) = \begin{pmatrix} x_1 \\ -(x_1-2)^2 + e^{x_2} + 3 \end{pmatrix}, \quad F'(x) = \begin{pmatrix} 1 & 0 \\ -2(x_1-2) & e^{x_2} \end{pmatrix}. \tag{10.16}$$

La fonction F a un unique zéro en $x_* = 0$ et $F'(x)$ présente une « singularité à l'infini » (c.-à-d., $\|F'(x)^{-1}\|_2$ explose pour $x_2 \to -\infty$). □

Nous examinons ci-après la question de l'*admissibilité asymptotique du pas unité* par la recherche linéaire dans le voisinage d'un zéro de F. La question que l'on se pose est la suivante. Dans quelles conditions peut-on garantir que le pas $\alpha = 1$ est admis par l'inégalité de décroissance suffisante

$$\varphi(x + \alpha d) \leqslant \varphi(x) + \omega\,\alpha\,\varphi'(x)d, \tag{10.17}$$

si l'itéré courant x est proche d'un zéro de F ? Dans cette inégalité, φ est la fonction de moindres-carrés définie par (10.10). Cette propriété est importante, car elle permet localement à l'algorithme avec recherche linéaire d'avoir la convergence quadratique locale de l'algorithme de Newton. Comme φ n'a pas de lien évident avec l'algorithme de Newton, on ne voit pas trop pourquoi il en serait ainsi. C'est le second miracle du couple Newton-φ (voir la proposition 10.7 pour le premier). Le résultat suivant donne des conditions pour que cette propriété d'admissibilité du pas unité ait lieu.

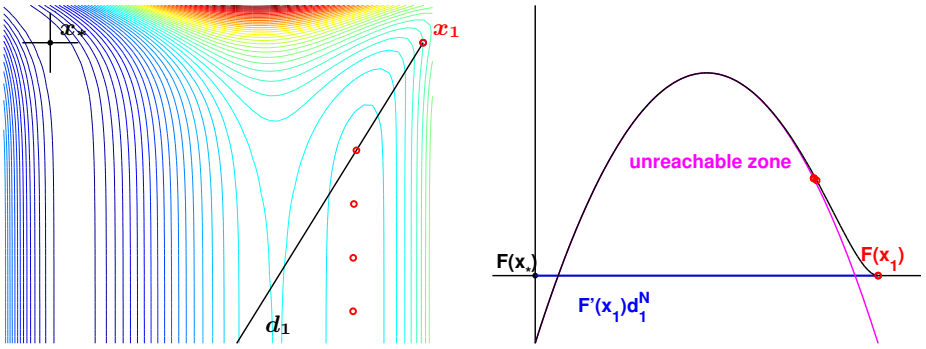


Fig. 10.3. Itérés de Newton avec recherche linéaire pour le système (10.16)

Proposition 10.15 (admissibilité asymptotique du pas unité – Newton) *On suppose que F est de classe C^1 dans le voisinage d'un zéro x_* de F et que $F'(x_*)$ est inversible. Si $\omega \in]0, \frac{1}{2}[$, si x est proche de x_* et si $d := -F'(x)^{-1}F(x)$ est la direction de Newton en x, alors l'inégalité (10.17) est vérifiée avec $\alpha = 1$.*

Démonstration. Si x est suffisamment proche de x_*, $F'(x)$ est inversible (parce que $F'(x_*)$ est inversible par hypothèse, que F' est continue dans un voisinage de x_* par hypothèse et que l'ensemble des opérateurs inversibles est un ouvert, une conséquence du lemme A.2). Il s'ensuit que la direction de Newton d est bien définie en des points proches de x_*. On supposera ci-dessous que les x voisins de x_* considérés sont différents de x_*, car autrement $d = 0$ et (10.17) est trivialement vérifiée.

De l'inversibilité de $F'(x_*)$ et de la proximité de x et x_*, on déduit l'existence d'une constante positive C telle que pour x voisin de x_*, on a

$$\|d\| \leqslant C\|F(x)\|. \tag{10.18}$$

En effet $F'(x_*)d = F'(x)d + o(\|d\|) = -F(x) + o(\|d\|)$ et $2C^{-1}\|d\| \leqslant \|F'(x_*)d\|$. On a noté ici et on notera ci-dessous $o(\|d\|^\sigma)$ une quantité qui dépend de x et telle que pour tout $\varepsilon > 0$, il existe un voisinage V de x_* tel que $o(\|d\|^\sigma)/\|d\|^\sigma < \varepsilon$ lorsque $x \in V \setminus \{x_*\}$ (on observera que $d \neq 0$ lorsque x est voisin et différent de x_*).

On cherche à présent à montrer la négativité de $\varphi(x+d) - \varphi(x) - \omega\,\varphi'(x)d$. En utilisant la dérivabilité de F, on a

$$\|F(x+d) - F(x) - F'(x)d\| \leqslant \left(\sup_{z \in]x, x+d[} \|F'(z) - F'(x)\| \right) \|d\|.$$

Par (10.18), on voit que $d \to 0$ lorsque $x \to \bar{x}$. On déduit déduit alors de l'inégalité précédente que

$$F(x+d) = F(x) + F'(x)d + o(\|d\|) \tag{10.19}$$

où $o(\|d\|)$ désigne un terme tel que $o(\|d\|)/\|d\| \to 0$ lorsque $x \to \bar{x}$ avec $x \neq \bar{x}$ (ce n'est donc pas le « petit o » de la différentiabilité, puisque x est également modifié

dans cette estimation). Pour l'algorithme de Newton, $F(x) + F'(x)d = 0$. On obtient alors les estimatioins suivantes

$$F(x+d) = o(\|d\|), \qquad [(10.19)]$$
$$\varphi(x+d) = \frac{1}{2}\|F(x+d)\|_2^2 = o(\|d\|^2).$$

Comme $\varphi(x) = \frac{1}{2}\|F(x)\|_2^2$ et $\varphi'(x)d = F(x)^\mathsf{T} F'(x)d = -\|F(x)\|_2^2$, on obtient finalement

$$\varphi(x+d) - \varphi(x) - \omega\,\varphi'(x)d = -(\tfrac{1}{2} - \omega)\|F(x)\|_2^2 + o(\|F(x)\|_2^2),$$

où on a aussi utilisé (10.18) poue transformer le $o(\|d\|^2)$ en $o(\|F(x)\|_2^2)$. Comme $\omega < \frac{1}{2}$, le membre de droite est négatif lorsque x est suffisamment proche de x_*. Ceci montre que l'inégalité (10.17) est vérifiée avec $\alpha = 1$ dans ce cas. □

Newton modifié ▲

On modifie la hessienne, pour en construire une approximation auto-adjointe définie positive. Cela peut se faire, soit en en calculant le spectre de $\nabla^2 f(x)$ (opération coûteuse), soit en modifiant sa factorisation de Cholesky.

Les résultats numériques ont montré que ces techniques ne sont pas très robustes (convergence souvent lente, voire inexistante en pratique) et qu'il est préférable d'utiliser l'approche par régions de confiance (section 10.3.2), qui elles aussi modifient la hessienne, mais avec une interprétation géométrique dans l'espace primal claire.

Newton tronqué

Les algorithmes étudiés dans cette section apportent un remède aux inconvénients de l'algorithme de Newton original sur les deux points suivants : (1) les problèmes de consistance et de convergence de la recherche linéaire et (2) le coût de résolution du système linéaire requis à chaque itération de l'algorithme de Newton. L'idée est de résoudre de manière partielle ce système linéaire (d'où le mot *tronqué*), ce qui permettra du même coup d'obtenir une direction de descente de qualité. On suppose toutefois que des dérivées premières de F (ou secondes de f en optimisation) sont évaluées, mais il n'est pas nécessaire de calculer toute la jacobienne $F'(x)$ (toute la hessienne $\nabla^2 f(x)$ en optimisation).

On peut décrire l'*algorithme de Newton tronqué* brièvement, comme suit. C'est une méthode à directions de descente, dans laquelle les directions sont déterminées en résolvant de manière approchée l'*équation de Newton*, qui est l'équation linéaire en $d_k \in \mathbb{R}^n$ suivante

$$H_k d_k = -g_k. \qquad (10.20)$$

On y a noté $H_k := \nabla^2 f(x_k)$ la hessienne de f en x_k et $g_k := \nabla f(x_k)$ son gradient en x_k. L'algorithme fait ensuite de la recherche linéaire le long de d_k pour déterminer un pas $\alpha_k > 0$. Ceci conduit au nouveau point $x_{k+1} := x_k + \alpha_k d_k$.

Ce que l'on vient de décrire est une *itération externe* de l'algorithme. La résolution approchée de l'équation de Newton se fait en général par un processus itératif (le plus souvent il s'agit d'itérations de gradient conjugué) que l'on arrête avant d'avoir trouvé

la solution et dont les itérations sont dites *internes*. Il y a dans ce cas deux processus itératifs imbriqués. On dit que l'algorithme de Newton est tronqué, pour exprimer le fait que le processus interne est interrompu avant convergence. Certains auteurs utilisent le terme « inexact » pour exprimer que (10.20) n'est pas résolue exactement à chaque itération (voir []), mais ce terme peut laisser penser que la méthode n'est pas très précise, ce qui n'est pas le cas.

Cette approche est justifiée par les considérations suivantes. La résolution précise de l'équation de Newton (10.20) peut prendre beaucoup de temps de calcul (pensez au cas où $n = 10^3 \dots 10^6$ et au fait qu'un système linéaire général se résout en $O(n^3)$ opérations), si bien qu'il est tentant d'en calculer une solution approchée à un coût inférieur. D'autre part, si la direction de Newton est bonne près d'une solution, il n'en est pas de même si l'itéré en est éloigné. Il est donc raisonnable de penser que l'on va être plus efficace et réduire le temps de calcul total en résolvant (10.20) grossièrement lorsqu'on est loin de la solution et avec plus de précision lorsqu'on s'en rapproche. En pratique, c'est la stratégie qu'il faut suivre, mais l'on voit que le choix du nombre d'itérations internes à exécuter par itération externe est délicat. C'est le talon d'Achille de la méthode : il faut que l'algorithme « sente » la proximité d'une solution pour bien doser l'effort à faire à chaque itération externe. Ceci demande souvent un réglage qui peut dépendre du problème.

On utilise souvent le gradient conjugué (GC) pour résoudre le système (10.20) de manière approchée et c'est avec ce processus itératif interne que nous présenterons l'algorithme. Par là on cherche à annuler ou à faire décroître le *résidu* (c'est le gradient de la fonction quadratique $\varphi_k(d) = \frac{1}{2} d^\mathsf{T} H_k d + g_k^\mathsf{T} d$)

$$r_k := H_k d_k + g_k.$$

Ceci revient aussi à minimiser partiellement le problème quadratique osculateur (10.7). De plus, l'algorithme du GC est démarré avec $d_k^0 = 0$. Dans ce cas, le résidu initial est $r_k^0 = g_k$ et la première direction de recherche est $-g_k$. Si l'algorithme s'arrête après la première itération, d_k sera approché par une direction parallèle à $-g_k$, si bien que l'algorithme de Newton tronqué se ramène à la méthode de la plus forte pente. D'autre part, plus on fait d'itérations internes, plus l'algorithme de Newton tronqué se rapproche de l'algorithme de Newton. Il s'agit donc d'une méthode intermédiaire entre ces deux extrêmes. Si on suit la stratégie mentionnée ci-dessus, la méthode est proche de l'algorithme du gradient dans les premières itérations et obtient la convergence rapide de l'algorithme de Newton proche de la solution. Cet algorithme converge si f est régulière et si $\{\nabla^2 f(x_k)\}$ reste bornée. Il n'est pas nécessaire que $\{\nabla^2 f(x_k)^{-1}\}$ soit bornée.

Voyons cela de manière plus précise. Soit $\{x_k\}$ la suite générée. L'algorithme a besoin que l'on spécifie deux constantes (indépendantes de l'itération k), $\gamma \in]0, 1[$ et $\omega_1 \in]0, \frac{1}{2}[$, qui sont utilisées dans la recherche linéaire. Ensuite, l'algorithme doit détecter quand est-ce qu'une direction interne générée par le gradient conjugué correspond à une courbure positive de f trop proche de zéro. Ceci se fait au moyen d'une valeur-seuil $\nu = \nu_k > 0$ qui pourra éventuellement être modifiée au cours des itérations externes. On dit alors qu'une direction v est *à courbure quasi-négative* pour f en x si

$$v^\mathsf{T} \nabla^2 f(x) v < \nu \|v\|_2^2. \tag{10.21}$$

Nous pouvons maintenant décrire une itération de l'algorithme, celle qui démarre en $x_k \in \mathbb{R}^n$.

Algorithme 10.16 (Newton tronqué en optimisation) On suppose qu'au début de l'itération k, on dispose d'un itéré $x_k \in \mathbb{R}^n$.

1. *Test d'arrêt.* Si $\nabla f(x_k) \simeq 0$, arrêt de l'algorithme.
2. *Direction.* On calcule d_k par i_k itérations (internes) de gradient conjugué qui démarrent en $d_k^0 := 0$. Pour $j \geqslant 0$:

   2.1. Calcul de la direction conjuguée interne ($r_k^j := H_k d_k^j + g_k$) :

   $$v_k^j := \begin{cases} -r_k^0 & (= -g_k) & \text{si } j = 0 \\ -r_k^j + \beta_k^j v_k^{j-1} & (\beta_k^j := \|r_k^j\|_2^2 / \|r_k^{j-1}\|_2^2) & \text{si } j \geqslant 1. \end{cases}$$

   2.2. Test d'arrêt : on interrompt les itérations internes (et on va au point 3) quand on veut, mais certainement quand $r_k^j = 0$ ou quand la direction interne v_k^i est à courbure « quasi-négative », c'est-à-dire si elle vérifie (10.21) avec $v = v_k^j$ et $\nu = \nu_k$. Dans ce cas, on prend

   $$d_k := \begin{cases} -g_k & \text{si } j = 0 \\ d_k^j & \text{si } j \geqslant 1 \end{cases}$$

   et on passe à l'étape 3.

   2.3. Nouvel itéré interne

   $$d_k^{j+1} := d_k^j + t_k^j v_k^j,$$

   où le pas $t_k^j > 0$ est calculé par la formule habituelle

   $$t_k^j := -\frac{(r_k^j)^\mathsf{T} v_k^j}{(v_k^j)^\mathsf{T} H_k v_k^j}.$$

3. *Calcul du pas.* On calcule un pas $\alpha_k > 0$ par la règle d'Armijo : α_k est le premier nombre (et le plus grand) dans $\{1, \gamma, \gamma^2, \gamma^3, \ldots\}$ tel que l'on ait
   $$f(x_k + \alpha_k d_k) \leqslant f(x_k) + \omega_1 \alpha_k g_k^\mathsf{T} d_k.$$
4. *Nouvel itéré.* $x_{k+1} := x_k + \alpha_k d_k$.

Voici quelques remarques sur l'algorithme.

– Il n'utilise de la hessienne $H_k = \nabla^2 f(x_k)$ que ses produits $H_k v$ par une direction conjuguée v. Il n'est donc pas nécessaire de calculer la hessienne complètement. Une routine qui calcule ces produits suffira. Rappelons que $H_k v$ est la dérivée du gradient en x_k et dans la direction v.

– Le contrôle du nombre d'itérations internes par itération externe est une tâche délicate. Nous avons donné les tests d'arrêt minimal. Comme on peut s'arrêter quand on veut (pour avoir convergence, d'après la proposition 10.17), on peut ajouter d'autres conditions d'arrêt librement. C'est dans ce sens que l'algorithme décrit ci-dessus est dit être dans sa version minimale.

– À la première étape, l'algorithme du GC ne voit pas la non définie positivité éventuelle de H_k, puisque $(v_k^j)^T H_k v_k^j \geqslant \nu_k \|v_k^j\|_2^2$ pour toute direction interne v_k^j acceptée. Il est donc bien défini.

  Si la première direction direction interne du GC, qui n'est autre que $-g_k$, est à courbure quasi-négative, l'algorithme ne la rejette pas (comme c'est le cas dans les itérations internes suivantes), mais la prend : $d_k = -g_k$. Donc, même si $\nabla^2 f(x_k) = 0$, cette étape de l'algorithme est bien définie et fournit $d_k = -g_k$ comme direction de recherche.

– On n'a pas précisé comment choisir le seuil ν_k au cours des itérations externes. C'est clairement un point délicat. Le résultat de convergence ci-dessous autorise plusieurs règles. On peut maintenir ν_k supérieur à un seuil constant $\nu > 0$, mais c'est assez restrictif, car il est difficile de savoir quelle est la bonne valeur de ν. La proposition analyse aussi le cas où ν_k n'est décru que lorsque le pas unité est accepté par la recherche linéaire. On a alors un résultat plus faible ($\liminf \|g_k\| = 0$), mais si l'on maintient ν_k supérieur à un seuil proportionnel à $\|g_k\|^p$ (p étant une constante positive), on retrouve un résultat de convergence satisfaisant ($g_k \to 0$). Cette dernière règle permet à ν_k de décroître dans le voisinage d'une solution, ce qui permet de ne pas empêcher la convergence quadratique de l'algorithme.

  Au lieu de contrôler par ν_k la petitesse des quotients de Rayleigh $v^T H_k v / \|v\|_2^2$ de H_k, on peut aussi contrôler celle de

$$\cos\theta_k := \frac{-g_k^T d_k}{\|g_k\| \, \|d_k\|}$$

  qui, contrairement aux quotients de Rayleigh, a l'élégance de décroître de façon monotone au cours des itérations internes (voir exercice 8.2).

L'algorithme de Newton tronqué permet d'avoir un résultat de convergence relativement fort ($g_k \to 0$), sous la seule condition que les hessiennes $\nabla^2 f(x_k)$ forment une suite bornée. On n'a pas besoin que l'inverse des hessiennes (qui n'existent peut être pas !) forment une suite bornée.

Proposition 10.17 (convergence de Newton tronqué) *Supposons que f soit deux fois dérivable. On considère l'algorithme de Newton tronqué décrit ci-dessus.*

(i) Si $x_k \in \mathbb{R}^n$ n'est pas un point stationnaire de f, la direction d_k est de descente pour f en x_k et l'algorithme est bien défini en x_k.

(*ii*) *Supposons que la suite* $\{f(x_k)\}$ *soit bornée inférieurement, que la suite* $\{\nabla^2 f(x_k)\}$ *soit bornée et qu'aucun itéré* x_k *généré ne soit un point stationnaire de* f.
   (*a*) *Si* ν_k *est maintenu plus grand qu'une constante* $\nu > 0$, *alors* $\nabla f(x_k) \to 0$.
   (*b*) *Si* ν_k *n'est décru que si le pas unité est accepté par la recherche linéaire à l'étape* k, *alors* $\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$.
   (*c*) *Si* ν_k *n'est décru que si le pas unité est accepté par la recherche linéaire à l'étape* k *et si* ν_k *vérifie*

$$\nu_k \geqslant \nu \|\nabla f(x_k)\|^p, \tag{10.22}$$

où $\nu > 0$ *et* $p \geqslant 0$ *sont des constantes, alors* $\nabla f(x_k) \to 0$. *Le même résultat a lieu si l'on prend* $\cos\theta_k$ *au lieu de* ν_k *dans* (10.22).

Démonstration. Commençons par donner une formule de la direction d_k. Pour cela, on constate que si $i_k \geqslant 1$, on a pour $0 \leqslant j \leqslant i_k$

$$(v_k^j)^\mathsf{T} H_k d_k^j = (v_k^j)^\mathsf{T} H_k \left(\sum_{l=0}^{j-1} t_k^l v_k^l \right) = 0,$$

parce que les directions internes v_k^l d'indices l différents sont conjuguées. Dès lors $(v_k^j)^\mathsf{T} r_k^j = (v_k^j)^\mathsf{T}(H_k d_k^j + g_k) = (v_k^j)^\mathsf{T} g_k$. On en déduit que

$$d_k = \sum_{j=0}^{i_k-1} t_k^j v_k^j = -\sum_{j=0}^{i_k-1} \frac{v_k^j (v_k^j)^\mathsf{T} g_k}{(v_k^j)^\mathsf{T} H_k v_k^j} = -J_k g_k,$$

où J_k est la matrice semi-définie positive de rang i_k donnée par la formule

$$J_k := \sum_{j=0}^{i_k-1} \frac{v_k^j (v_k^j)^\mathsf{T}}{(v_k^j)^\mathsf{T} H_k v_k^j}.$$

Si $i_k = 0$, on a aussi $d_k = -J_k g_k$, avec cette fois $J_k = I$.

On voit alors facilement que, si x_k n'est pas stationnaire, d_k est une direction de descente de f en x_k. En effet, si $i_k = 0$, $g_k^\mathsf{T} d_k = -\|g_k\|_2^2$. Si $i_k \geqslant 1$, en utilisant le fait que $(v_k^j)^\mathsf{T} H_k v_k^j > 0$ et que $v_k^0 = -g_k$, on a

$$g_k^\mathsf{T} d_k = -\sum_{j=0}^{i_k-1} \frac{(g_k^\mathsf{T} v_k^j)^2}{(v_k^j)^\mathsf{T} H_k v_k^j} \leqslant -\frac{(g_k^\mathsf{T} v_k^0)^2}{(v_k^0)^\mathsf{T} H_k v_k^0} = -\frac{\|g_k\|_2^4}{g_k^\mathsf{T} H_k g_k} \leqslant -\frac{\|g_k\|_2^2}{\|H_k\|_2}.$$

En rassemblant les deux cas:

$$g_k^\mathsf{T} d_k \leqslant -\min\left(1, \frac{1}{\|H_k\|_2}\right) \|g_k\|_2^2. \tag{10.23}$$

Donc $g_k^\mathsf{T} d_k < 0$ si $g_k \neq 0$.

Supposons à présent que, pour tout $k \geqslant 1$, $\nu_k \geqslant \nu$, où $\nu > 0$ est une constante. Montrons qu'il existe une constante $C > 0$ telle que

$$f(x_{k+1}) \leqslant f(x_k) - C\|g_k\|^2. \tag{10.24}$$

La convergence de $g_k \to 0$ s'en déduit du fait que $\{f(x_k)\}$ est décroissante et bornée inférieurement. D'après la proposition 6.11, il existe une constante $C_1 > 0$ telle que $\forall k \geqslant 1$, on ait soit

$$f(x_{k+1}) \leqslant f(x_k) - C_1|g_k^{\mathsf{T}} d_k|, \tag{10.25}$$

soit

$$f(x_{k+1}) \leqslant f(x_k) - C_1\|g_k\|^2 \cos^2\theta_k, \tag{10.26}$$

où $\cos\theta_k = -(g_k^{\mathsf{T}} d_k)/(\|g_k\|_2\|d_k\|_2)$. Si la première inégalité (10.25) a lieu, on a par l'estimation (10.23) de $g_k^{\mathsf{T}} d_k$:

$$f(x_{k+1}) \leqslant f(x_k) - C_1 \min\left(1, \frac{1}{\|H_k\|_2}\right)\|g_k\|_2^2. \tag{10.27}$$

On en déduit (10.24) du fait que $\{H_k\}$ est supposée bornée. Supposons à présent que la seconde inégalité (10.26) ait lieu. Notons d'abord que $\|uu^{\mathsf{T}}\|_2 = \|u\|_2^2$ et que $(v_k^j)^{\mathsf{T}} H_k v_k^j \geqslant \nu_k \|v_k^j\|_2^2$ pour tout $j = 0, \ldots, i_k$. Dès lors $\|J_k\|_2 \leqslant \max(1, n\nu_k^{-1})$ et par (10.23)

$$\cos\theta_k = \frac{-g_k^{\mathsf{T}} d_k}{\|g_k\|_2\|d_k\|_2} \geqslant \min\left(1, \frac{1}{\|H_k\|_2}\right)\frac{\|g_k\|_2}{\|d_k\|_2} \geqslant \min\left(1, \frac{1}{\|H_k\|_2}\right)\min\left(1, \frac{\nu_k}{n}\right).$$

La suite $\{H_k\}$ étant supposée bornée, le cosinus de θ_k est uniformément positif et on obtient également (10.24).

Si ν_k n'est décru que lorsque le pas unité est accepté par la recherche linéaire, deux cas peuvent se présenter. Soit $\liminf \nu_k > 0$ et on est ramené au point $(ii\text{-}a)$, selon lequel $g_k \to 0$. Soit il existe une sous-suite d'itérés pour lesquels le pas unité est accepté. Pour les indices k correspondants, on a (10.25) avec $C_1 = \omega_1$, donc (10.27), et du fait que $\{H_k\}$ est bornée, cela implique que $g_k \to 0$ pour les indices k considérés.

Considérons pour terminer le cas où ν_k n'est décru que lorsque le pas unité est accepté par la recherche linéaire et où (10.22) a lieu (éventuellement avec $\cos\theta_k$ au lieu de ν_k). On sait déjà que $\liminf \|g_k\| = 0$. Si toute la suite $\{g_k\}$ ne converge pas vers zéro, on peut trouver une constante $\gamma > 0$ et une suite d'indices $\{l_k\}_{k\geqslant 0}$ strictement croissante telle que pour tout $k \geqslant 0$:

$$\|g_{l_{2k}}\| \geqslant \gamma \quad \text{et} \quad \|g_{l_{2k+1}}\| \leqslant \gamma/2.$$

Pour $l_{2k} \leqslant l < l_{2k+1}$, en utilisant la borne inférieure sur $\cos\theta_k$ ci-dessus et (10.22), on obtient

$$f(x_{l+1}) \leqslant f(x_l) - \omega_1\|g_l\|\|s_l\|\cos\theta_l \leqslant f(x_l) - C\|s_l\|,$$

où $s_l = x_{l+1} - x_l$ et $C > 0$ est une constante indépendante de k et de l. On en déduit

$$\|x_{l_{2k+1}} - x_{l_{2k}}\| \leqslant \sum_{l=l_{2k}}^{l_{2k+1}-1}\|s_l\| \leqslant \frac{1}{C}\Big(f(x_{l_{2k}}) - f(x_{l_{2k+1}})\Big).$$

Dès lors $\|x_{l_{2k+1}} - x_{l_{2k}}\| \to 0$ et, par l'uniforme continuité de ∇f, $\|g_{l_{2k+1}} - g_{l_{2k}}\| \to 0$, ce qui contredit le fait que $\|g_{l_{2k+1}} - g_{l_{2k}}\| \geqslant \gamma/2$. □

10.3.2 Régions de confiance ▲

Cette globalisation de la convergence offre plus de robustesse (résultats de convergence meilleurs, moins de problème à l'utilisation), mais elle n'est pas toujours utilisable pour résoudre des systèmes non linéaires de très grande taille.

Systèmes non linéaires

Présenter l'algortithme classique qui minimise $\|F(\cdot)\|$ ou $\frac{1}{2}\|F(\cdot)\|_2^2$ avec le pas de Cauchy $-\alpha^c F'(x)^\top F(x)$. Discuter des méthodes avec résolution directe ou itérative du système de Newton. Pour les résolutions itératives, discuter des méthodes qui permettent d'avoir la croissance du pas de Newton approché au cours des itérations internes.

Cet algorithme classique a l'inconvénient de requérir le calcul du produit de la *transposée* de la jacobienne par un vecteur pour estimer le pas de Cauchy. Ceci peut être un inconvénient majeur pour les grands problèmes dans lesquels les produits jacobienne-vecteur sont estimés par différences finies. Le seul algorithme n'utilisant pas la transposée de la jacobienne semble être celui de Brown et Saad [95 ; 1990], mais sa convergence n'est pas démontrée.

Optimisation

Tout un chapitre est consacré à cette méthode importante (le chapitre 9). Mentionnons seulement ici son principe.

10.3.3 Autres méthodes

Nous évoquons succintement dans cette section d'autres approches de globalisation de la convergence, sans en étudier lleur convergence.

Réduction du pas de temps dans la résolution d'équations différentielles

Supposons que l'on cherche à résoudre l'équation différentielle

$$\frac{\mathrm{d}x}{\mathrm{d}t} + \phi(x) = 0, \quad x(0) = x_0, \tag{10.28}$$

où l'état initial $x_0 \in \mathbb{R}^n$ est donné et $\phi : \mathbb{R}^n \to \mathbb{R}^n$ est une fonction non linéaire.

Un schéma de discrétisation en temps de (10.28) est *implicite* si l'état (approché) x_{i+1} au temps $t_{i+1} > 0$ est solution d'une équation faisant intervenir $\phi(x_{i+1})$. Ainsi, dans le *schéma d'Euler implcite*, x_{i+1} est solution de l'équation non linéaire en x suivante :

$$\frac{x - x_i}{\delta_i} + \phi(x) = 0, \tag{10.29}$$

où $\delta_i := t_{i+1} - t_i > 0$ est un *pas de temps* que l'on se donne.

On cherche parfois une solution de l'équation non linéaire (10.29) par des itérations de Newton. Si δ_i est petit, l'*état précédent* x_i ou l'*état prédit*

$$x_i - \phi(x_i)\,\delta_i \quad \left[= x_i + \frac{\mathrm{d}x}{\mathrm{d}t}(t_i)\,\delta_i\right]$$

sont en général de bons points de départ pour ces itérations. Si la solution de l'équation différentielle (10.28) dépend continûment du temps, ces points de départ seront d'autant meilleurs que δ_i est petit (on calcule alors une approximation de $x(t_i+\delta_i)$, qui dépend du choix de δ_i). Un moyen d'obtenir la convergence des itérés de Newton est de prendre un pas de temps δ_i suffisamment petit et de le réduire si la convergence ne se produit pas en quelques itérations (10 par exemple).

Méthode du régime pseudo-transitoire

Bien que rappelant la section précédente par certains aspects, l'approche décrite ici est bien différente. L'idée est de chercher à calculer un zéro de F comme un *état stationnaire* (c.-à-d., ne dépendant pas du temps) de l'équation différentielle

$$\frac{\mathrm{d}x}{\mathrm{d}t} + F(x) = 0, \quad x(0) = x_0. \tag{10.30}$$

On constate en effet qu'il y a une bijection entre les états stationnaires de (10.30) et les zéros de F. L'équation (10.30) rappelle l'équation différentielle (10.28), mais elle est introduite ici de manière artificielle. De plus, on n'est pas intéressé ici par l'évolution de l'état $x(t)$, la solution de (10.30), au cours du temps fictif t, mais seulement par l'état asymptotique, lorsque $t \uparrow \infty$.

Observons que l'approche du régime pseudo-stationnaire n'est pas symétrique dans le sens suivant. Si l'équation non linéaire $F(x) = 0$ ne change pas si on remplace F par $-F$ (on garde les mêmes zéros), l'équation différentielle (10.30) est sensible au fait de remplacer F par son opposé. Par exemple, si F est donnée par (10.8) et $x_0 = 0$, on a $x(t) < 0$ pour tout $t > 0$ et la trajectoire s'écarte de l'unique état stationnaire $x_* = 1$ lorsque t augmente ; par contre si on change le signe de F, la trajectoire se dirige vers $x_* = 1$ lorsque $t \uparrow \infty$. Techniquement et pratiquement, la convergence de l'approche du régime transitoire ne pourra être garantie que si l'on peut faire l'hypothèse que la trajectoire issue de x_0 converge vers un zéro de F lorsque $t \uparrow \infty$.

Voici la méthode. Dans un premier temps, on discrétise l'équation différentielle (10.30) par un *schéma d'Euler implcite* : en l'itéré x_k ($k \geqslant 0$), on s'intéresse à la solution de l'équation non linéaire

$$\frac{x - x_k}{\delta_k} + F(x) = 0, \tag{10.31}$$

où $\delta_k > 0$ est un *pas de temps*. Le plus souvent, l'itéré suivant est obtenu en faisant une unique itération de Newton pour résoudre cette équation, ce qui conduit à prendre x_{k+1} qui vérifie

$$F(x_k) + \left[\delta_k^{-1} I + F'(x_k)\right](x_{k+1} - x_k) = 0.$$

Si $\delta_k^{-1} I + F'(x_k)$ est inversible, on obtient

$$x_{k+1} = x_k - \left[\delta_k^{-1} I + F'(x_k)\right]^{-1} F(x_k).$$

On retrouve l'algorithme de Newton lorsque $\delta_k = \infty$. Il est coutumier de choisir les pas de temps par des variantes de la règle suivante [446, 608, 352]

$$\delta_k = \frac{\|F(x_{k-1})\|}{\|F(x_k)\|}\, \delta_{k-1},$$

qui fait croître δ_k autant que $\|F(x_k)\|$ décroît. On peut aussi plafonner le pas de temps δ_k par $\delta_{\max} > 0$ si la valeur donnée par la formule précédente dépasse le seuil fixé $\delta_{\max}$ [337] ou le prendre infini dans les mêmes circonstances [195].

Il faut noter que dans l'approche du régime transitoire la suite $\{\|F(x_k)\|\}$ n'est pas nécessairement décroissante, ce qui permet parfois d'éviter les minima locaux de $\|F(\cdot)\|$, une propriété que n'ont pas la recherche linéaire et les régions de confiance.

Des conditions de convergence de cette technique sont données dans [351].

Méthodes de continuation

Les *méthodes de continuation* peuvent constituer une approche intéressante lorsque l'équation non linéaire à résoudre $F(x) = 0$ contient un paramètre $p \in \mathbb{R}$ qui peut atténuer la difficulté du problème lorsqu'on change sa valeur. De façon plus précise, l'équation originale correspond à la valeur $p = p_1$ du paramètre :

$$\forall x \in \mathbb{R}^n : \quad F(x) = \Phi(x, p_1),$$

où $\Phi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, et le système

$$\Phi(x, p) = 0 \tag{10.32}$$

est « facile » à résoudre lorsque $p = p_0$ (que l'on va supposer $< p_1$). On note x_0 une solution (approchée) de (10.32) avec $p = p_0$. Un exemple typique en mécanique des fluides est celui des équations de Navier-Stokes, dans lesquelles le nombre de Reynolds peut jouer le rôle du paramètre p ci-dessus.

La version la plus simple des méthodes de continuation suppose qu'il existe une fonction implicite $p \in \mathbb{R} \mapsto x(p)$ telle que $\Phi(x(p), p) = 0$ pour tout $p \in [p_0, p_1]$ et elle cherche à suivre approximativement le chemin $x([p_0, p_1])$ en faisant croître progressivement p de p_0 à p_1. Connaissant une solution approchée x_i de (10.32) avec $p = p_i$ (indice i fractionnaire dans $[0, 1]$), un bon point de départ pour calculer une solution de (10.32) avec $p = p_{i+1}$ est obtenu par la *phase de prédiction* suivante :

$$x^0_{i+1} = x_i + x'(p_i)(p_{i+1} - p_i).$$

On obtient la dérivée $x'(p)$ de la fonction implicite $p \mapsto x(p)$ en différentiant l'identité $\Phi(x(p), p) \equiv 0$ par rapport à p, ce qui donne

$$x'(p_i) = -\left(\frac{\partial \Phi}{\partial x}(x_i, p_i)\right)^{-1} \frac{\partial \Phi}{\partial p}(x_i, p_i).$$

On peut alors calculer le point de prédiction x^0_{i+1} à partir duquel quelques itérations de Newton sur le système $\Phi(\cdot, p_{i+1}) = 0$ permettent de trouver x_{i+1}.

Le problème est plus compliqué si, le long du chemin suivi, on rencontre des points de bifurcation, de rebroussement, *etc.* Comme points d'entrée sur ce sujet à peine ébauché, citons [367, 616, 10, 11, 129].

Notes

Isaac Newton (1642-1727) était intéressé par le calcul de zéro de polynôme et sa méthode, exposée dans l'épigraphe de ce chapitre, était sensiblement différente de

l'algorithme de Newton tel que nous le connaissons aujourd'hui, celui présenté à la section 10.1.1. Même si les itérés générés sont identiques dans les deux approches, celle de Newton ne s'étend pas aisément aux fonctions non polynomiales. Le texte donné en épigraphe est tiré de *Methodus fluxionum et serierum infinitorum*, qui fut écrit en latin entre 1664 et 1671, édité en anglais en 1736 ; l'algorithme fut également exposé dans *De analysi per aequationes numero terminorum infinitas*, ouvrage composé en 1669 mais seulement publié en 1711. [112]

On associe souvent le nom de *Joseph Raphson* (peut-être 1648-1712) à celui de Newton pour nommer l'algorithme 10.1. Raphson s'intéressait aussi au calcul de zéro de polynôme. Sa contribution, qui date de 1690 et 1697 [506], a été d'écrire l'algorithme sous la forme $x_{k+1} = \varphi(x_k)$, où la fonction rationnelle φ est construite à partir du polynôme considéré, mais sans faire intervenir sa dérivée. [362, 639]

Les apports de *Thomas Simpson* (1710-1761) à l'algorithme 10.1 ont trop souvent été oubliés. Ils furent pourtant essentiels. On peut en citer trois. Le premier est d'avoir fait intervenir la dérivée de la fonction (qu'il appelle *fluxion*, comme Newton) dans le calcul du nouvel itéré, permettant ainsi d'appliquer l'algorithme à des fonctions non polynomiales, ce qu'il fit [553 ; 1740, p. 83-84]. Sa seconde contribution est d'avoir montré comment on pouvait utiliser l'algorithme pour résoudre un système de 2 équations à 2 inconnues, en résolvant un système linéaire dont la matrice est la jacobienne de la fonction en l'itéré courant [553 ; 1740, p. 82]. Enfin, il donne sans doute le premier exemple de maximisation d'une fonction de plusieurs variables sans contrainte, par recherche d'un zéro de son gradient [552 ; 1737]. [639]

Ypma [639] attribue l'absence de reconnaissance aux autres contributeurs à l'algorithme de Newton au livre influent de Fourier [220 ; 1831], lequel l'appelait la *méthode newtonienne*, sans faire référence à Raphson ou Simpson.

En 1939, Kantorovitch [341] a présenté un résultat préliminaire de convergence de l'algorithme de Newton, qu'il améliora substantiellement en 1948/49 [342, 343]. La version du théorème proposée (théorème 10.3, [169, 345]) est parfois qualifiée d'*invariante par transformation linéaire* (« affine invariant »), parce qu'elle est invariante lorsqu'on pré-compose F avec une application linéaire bijective (F devient $F \circ A$, avec A linéaire inversible), comme l'est l'algorithme de Newton (proposition 10.4). Les hypothèses du théorème de Kantorovitch sont plus fortes que celles d'autres théorèmes d'existence de zéro, tels que ceux de Miranda, de Moore, de Borsuk [5, 4] et d'autres théorèmes de point fixe, mais elles donnent aussi plus d'informations, à savoir la convergence des itérés de Newton et donc un moyen numérique de calculer le point fixe. Pour une revue de l'évolution de l'analyse de la convergence de l'algorithme de Newton, on pourra consulter [631].

Le premier exemple de système non linéaire $F(x) = 0$ pour lequel l'algorithme de Newton *avec recherche linéaire* génère des points convergeant vers un point singulier de F' qui n'est ni un zéro de F ni un point stationnaire de $\|F(\cdot)\|_2^2$ est dû à Powell [487 ; 1970]. Cet exemple a motivé l'introduction des méthodes à régions de confiance pour globaliser l'algorithme de Newton pour ces problèmes, approche qui ne présente pas le même inconvénient. En optimisation aussi, l'algorithme de Newton avec recherche linéaire (celle de Wolfe par exemple) présente le même type de défaut : il peut générer des itérés x_k convergeant vers un point où le gradient n'est pas nul, alors que la hessienne est définie positive en tout itéré [412 ; 2008].

Le comportement de l'algorithme de Newton pour résoudre un système d'équations non linéaires dont la jacobienne est singulière en la solution a souvent été exploré, notamment en dimension un [638]. La revue de Griewank [285 ; 1985] considère le cas multidimensionnel et présente quelques modifications de la méthode de Newton pour faire face aux problèmes de convergence et de stabilité numérique que cette singularité entraîne ; mentionnons une technique de *sur-relaxation*, dans laquelle $x_{k+1} = x_k + \alpha_k d_k$, où d_k est la direction de Newton et le pas α_k est pris dans l'intervalle $[1, 2[$. Une autre possibilité, explorée par Schnabel et ses collaborateurs [81 ; 1998], sont les méthodes dites *tensorielles*, dans lesquelles on ajoute à l'approximation linéaire de F, quelques termes d'ordre 2 (des tenseurs), qui sont approchés par des techniques quasi-newtoniennes.

Les *méthodes de Newton inexactes* ont été beaucoup étudiées, car elles sont très utilisées pour résoudre les grands systèmes non linéaires issus de la discrétisation d'équations aux dérivées partielles. L'article fondateur est [162 ; 1982] et on trouvera de nombreux articles de synthèse et de monographies sur cette question (par exemple [356]). Pour la proposition 10.9, nous avons repris les arguments de [184], eux-mêmes inspirés de [472, 301], qui considèrent la situation plus complexe d'équation non lisse. L'*algorithme de Newton tronqué* décrit à la section 10.3.1, qui s'inscrit dans la veine des méthodes inexactes, est dû à Dembo et Steihaug [163 ; 1983].

L'effet de l'arithmétique flottante sur l'algorithme de Newton a été étudié par divers auteurs ; citons Dennis et Walker [167 ; 1984] et Tisseur [587 ; 2001].

L'extension de l'algorithme de Newton à la résolution du système d'équations non linéaires $F(x) = 0$ dans lequel F est *non différentiable* s'est faite suivant plusieurs directions. Le cas des fonctions C^1 par morceaux est analysé par Kojima et Shindo [360 ; 1986] qui montrent que la convergence quadratique locale est préservée par l'algorithme qui utlise une quelconque des jacobiennes des fonctions actives au point courant, pourvu que soient vérifiées des hypothèses naturelles (au vu du théorème 10.2) incluant l'inversibilité des jacobiennes des fonctions actives en la solution ; nous ne connaissons pas de résultat de convergence globale pour cet algorithme. On a ensuite étudié le cas fréquemment rencontré des *fonctions B-différentiables*, qui sont celles qui vérifient l'estimation (C.9) des fonctions Fréchet-différentiables, mais avec une application $h \mapsto Lh \equiv F'(x)h$ qui n'est plus que positivement homogène de degré 1 (on perd la linéarité). Des résultats de convergence locale et globale par recherche linéaire peuvent être obtnus [471 ; 1990], mais l'équation de Newton à résoudre à chaque itération, $F(x) + F'(x)d = 0$, est cette fois non linéaire, ce qui complique l'algorithme. Le cas où F est *semi-lisse* a commencé a être exploré par Qi et Sun [503 ; 1993], qui ont proposé un algorithme ne requérant que la résolution d'un système linéaire à chaque itération, ce qui est attractif, mais dont la globalisation de la convergence est plus difficile à mettre au point. On pourra lire sur ce thème la synthèse très complète de Facchinei et Pang [198], qui appliquent les algorithmes présentés à la résolution des *problèmes de complémentarité* ou d'*inéquations variationnelles*.

Une extension de l'algorithme de Newton à la recherche de zéro de fonction non lisse, avec zéro non isolé et en présence de contrainte est proposée dans [197 ; 2014].

Autres monographies sur l'algorithme de Newton : Kelley [348, 349, 350 ; 1995-2003], dont la dernière référence contient de nombreux conseils sur la mise en œuvre et le contrôle de l'algorithme ; Higham [313 ; 2002, § 2.5] donne une analyse d'erreur ; Deuflhard [168 ; 2004] décrit l'utilisation des algorithmes de Newton dans la résolution

de problèmes gouvernés par des équations différentielles ; Dedieu [160 ; 2006] présente la théorie en dimension infinie (avec des résultats de Smale) ; Argyros [18 ; 2008] ; Ulbrich [598 ; 2011] fait une synthèse en dimension infinie sur la méthode de Newton semi-lisse, Izmailov and Soldov [330 ; 2014] traitent des problèmes d'optimisation et d'inéquations variationnelles.

Exercices

10.1. Montrez qu'en un point non stationnaire, lorsqu'elle est bien définie, la direction de Newton pour minimiser f est une direction de descente de $x \mapsto \varphi(x) := \|\nabla f(x)\|$, où $\|\cdot\|$ est une norme quelconque.

Remarque. On sait qu'au contraire la direction de Newton n'est pas nécessairement une direction de descente de f en un point éloigné d'une solution forte. On pourrait donc penser qu'il est préférable de globaliser la méthode de Newton en cherchant à minimiser φ. Il n'en est rien. Cela vient du fait que φ est moins bien conditionnée que f (pensez au cas quadratique), si bien que loin d'une solution le pas accepté par φ peut être très petit, au point d'empêcher tout progrès significatif vers la solution.

10.2. On considère l'algorithme de Newton pour résoudre l'équation non linéaire $F(x) = 0$, où $F : \mathbb{R}^n \to \mathbb{R}^n$ est $\mathcal{C}^{1,1}$ dans le voisinage de x_*, une solution telle que $F'(x_*)$ soit inversible. Pour mesurer le progrès vers la solution, on utilise la fonction $x \mapsto \varphi(x) := \|F(x)\|$, où $\|\cdot\|$ est une norme quelconque. Montrez que le pas unité le long de la direction de Newton $d = -F'(x)^{-1}F(x)$ est accepté localement par l'inégalité d'Armijo : si x est voisin de x_* et $\omega \in]0, 1[$, on a $\varphi(x + d) \leqslant \varphi(x) + \omega\varphi'(x; d)$.

Montrez qu'il en est de même si $\varphi(x) = \|F(x)\|^p$, avec $p \geqslant 1$ (norme arbitraire).

10.3. *Lignes de flux de Newton* [306]. On considère le problème de la minimisation d'une fonction $f : \mathbb{R}^n \to \mathbb{R}$. Les *lignes de flux de Newton* sont les courbes $t \mapsto x(t)$, solutions de l'équation différentielle

$$\dot{x} = -H(x)^{-1}g(x), \qquad x(0) = x_0,$$

où $\dot{x}$ désigne la dérivée de $x(\cdot)$ par rapport à t, $H(x) := \nabla^2 f(x)$ est supposé inversible aux points visités $x(t)$, $g(x) := \nabla f(x)$ et x_0 est une condition initiale arbitraire. Ces courbes ont des propriétés remarquables dont certaines sont aisées à vérifier.

1) Le gradient le long d'une ligne de flux vérifie $g(x(t)) = e^{-t}g(x_0)$ et donc, si $g(x_0) \neq 0$, le gradient normalisé $g(x)/\|g(x)\|$ y est constant (la norme $\|\cdot\|$ est arbitraire).

Soient $x_0 \in \mathbb{R}$ et $\mathcal{N}_0 := \{x \in \mathbb{R}^n : f(x) \leqslant f(x_0)\}$. On suppose à présent que $H(x) \succ 0$ pour tout $x \in \mathcal{N}_0$ et qu'il existe un point $x_* \in \mathcal{N}_0$ tel que $\nabla f(x_*) = 0$.

2) Montrez que $\lim_{t\to\infty} x(t) = x_*$.
3) Montrez que $\lim_{t\to\infty} e^t\dot{x}(t) = -H(x^*)^{-1}g(x_0)$.
4) Montrez que l'application $x \in \mathcal{N}_0 \mapsto g(x)$ est injective.

On se donne à présent une autre fonction $\tilde{f} : \mathbb{R}^n \to \mathbb{R}$ telle que $\nabla^2\tilde{f}(x) \succ 0$ pour $x \in \mathcal{N}_0$ et qui ne diffère de f que sur un ouvert $\Omega := \{x \in \mathbb{R}^n : f(x) \neq \tilde{f}(x)\}$ tel que $x_0 \notin \overline{\Omega}$. On note $\tilde{x} : [0, +\infty[\mapsto \mathbb{R}^n$ la ligne de flux de Newton associée à $\tilde{f}$, issue de $x_0 := \tilde{x}(0)$.

5) Montrez que $\tilde{x}(t) = x(t)$ si $x(t) \notin \Omega$ ou si $\tilde{x}(t) \notin \Omega$.

Conclusion. On a donc le résultat étonnant suivant : le flux de Newton associé à une perturbation modérée $\tilde{f}$ de f (telle que $\nabla^2\tilde{f}(x) \succ 0$) n'est différent de celui associé