

Partie

II

Optimisation différentiable sans contrainte

Cette première partie comporte deux chapitres. Le chapitre deux, consacré à l'optimisation de fonctions d'une variable réelle sert de présentation d'un plan d'étude pour les problèmes d'optimisation en général. Nous adopterons ce plan d'étude dans la suite de ce manuscrit.

En bref, nous débutons par formaliser le problème, la caractérisation et le calcul de minima locaux de la fonction objectif. Puis, nous introduisons des conditions d'optimalité nécessaires et suffisantes. Ces conditions d'optimalité constituent la base de toute l'étude de minima locaux. Parfois, on peut formuler plusieurs telles conditions.

Par la suite, nous déduisons des algorithmes itératifs du fait qu'un point donné x_0 ne satisfait pas à une condition nécessaire d'optimalité. Chaque variante de condition nécessaire peut mener à un nouvel algorithme. Finalement, nous étudions plus en détail les propriétés de convergence de ces algorithmes itératifs.

Le chapitre trois traite de problèmes multidimensionnels, les variables étant dans \mathbb{R}^n . La même démarche conduit à l'étude de nombreux algorithmes regroupés dans des familles dites "de recherche linéaire" et "de région de confiance".

Chapitre 2

Préliminaires



Sujets du chapitre

- Vocabulaire des problèmes d'optimisation.
- Notion de conditions d'optimalité.
- Algorithmes de descente.
- Convergence globale des algorithmes.
- Vitesse de convergence des algorithmes.

Introduction

Ce texte est consacré à l'étude de *problèmes d'optimisation*. Ce chapitre de préliminaires vise à fixer les éléments de vocabulaire qui permettent de décrire, de parler de ce type de problèmes, de leurs solutions, des algorithmes de résolution. On profite aussi de ces préliminaires pour effectuer certains rappels d'analyse mathématique, d'algèbre linéaire, et d'analyse numérique.

2.1 Optimisation

Avant d'entreprendre l'étude de problèmes d'optimisation, il est bon de bien définir ce qu'est un problème d'optimisation. Dans toute sa généralité, le problème d'optimisation consiste à déterminer la plus petite (grande) valeur possible qu'une fonction réelle $f : E \rightarrow \mathbb{R}$ nommée *fonction objectif* puisse prendre dans l'ensemble E nommé *ensemble réalisable*. Sous forme mathématique, ceci s'exprime (pour le cas de la minimisation)

$$f^* = \inf_{x \in E} f(x),$$

et signifie :

$$f(x) \geq f^*, \forall x \in E \text{ et } \forall \epsilon > 0 \exists x_\epsilon \in E : f(x_\epsilon) < f^* + \epsilon;$$

Une autre façon d'écrire la signification de *inf* est d'introduire l'ensemble \mathcal{F} des valeurs qui bornent inférieurement $f(x)$ sur l'ensemble $E : \mathcal{F} = \{v : f(x) \geq v, \forall x \in E\}$. Cet ensemble est une demi-droite réelle, possiblement ouverte ; sa fermeture, $\overline{\mathcal{F}}$ est une demi-droite fermée. Alors, $f^* = \max\{v \in \overline{\mathcal{F}}\}$.

Pour le cas de maximisation, l'expression est $f^* = \sup_{x \in E} f(x)$, et signifie : $f(x) \leq f^*, \forall x \in E$ et $\forall \epsilon > 0 \exists x_\epsilon \in E : f(x_\epsilon) > f^* - \epsilon$. En fait, maximiser f revient à minimiser $-f$. Dans la suite de ce texte, arbitrairement, on considère les problèmes de minimisation.

Sans ajouter d'hypothèse sur la fonction f et l'ensemble E , il n'est pas certain que l'on puisse trouver un élément x^* de l'ensemble E pour lequel $f(x^*) = f^*$. Lorsque c'est le cas, la formulation mathématique devient (toujours pour le cas de la minimisation)

$$f(x^*) = f^* = \min_{x \in E} f(x),$$

et signifie : $f(x) \geq f(x^*), \forall x \in E$. La notation *min* est réservée aux problèmes pour lesquels il existe un $x^* \in E$ qui minimise f .

Exemple 2.1.1 Soit $E = \mathbb{R}$ et $f(x) = e^x$. On a $0 = \inf e^x$, mais $e^x > 0$ pour toute valeur réelle x . Par contre, si $f(x) = x^2$, on a $0 = f(0) = \min x^2$. \square

Exercice 2.1.1 [*Min et inf*] Vérifiez que $0 = \inf e^x$ et $0 = f(0) = \min x^2$ satisfont à la définition de inf et min.

Nous nous intéressons plutôt aux problèmes pour lesquels il existe un élément de l'ensemble tel que $f(x^*) = f^*$. Un tel point de l'ensemble E est nommé un *optimum*, qui peut être un *minimum* ou encore un *maximum*.

Les chapitres qui suivent considèrent différentes hypothèses sur les données du problème (la fonction f et l'ensemble E). La suite de ce chapitre de préliminaires propose une classification des *solutions*, ou optima ainsi que quelques notions élémentaires sur les conditions d'optimalité et les algorithmes.

2.2 Types d'optima

Sans hypothèse additionnelle, les problèmes tels que décrits plus haut sont en général impossible à résoudre. En effet, considérons la figure 2.1. Où se situe le minimum de la fonction ? De quel outil peut-on disposer pour identifier celui (ou ceux) des nombreux (infinis) minima apparents qui est le plus petit ?

Pour arriver à étudier les problèmes d'optimisation, nous allons classer les optima selon différents critères. Par exemple, sur la figure 2.1, l'origine est un minimum local, mais ce minimum local n'est pas isolé alors que les autres minima locaux sont des minima locaux isolés. Voyons ces notions en détail.

2.2.1 Optima locaux et globaux

Jusqu'à maintenant, nous avons défini les optima en comparant la valeur de la fonction f à l'optimum avec sa valeur en *tout* autre point de E . Ce type de problème est connu sous le nom d'*optimisation globale*.

Si l'on affaiblit ses attentes, et que l'on compare les valeurs de la fonction f dans un voisinage d'un candidat optimum x^* , alors x^* est un *minimum local* de f s'il existe un $\epsilon > 0$ tel que $f(x) \geq f(x^*)$, $\forall x \in E \cap V_\epsilon(x^*)$, où $V_\epsilon(x^*)$ dénote un voisinage de diamètre ϵ centré en x^* .¹ La raison principale de se restreindre aux optima locaux est de pouvoir utiliser les outils de l'analyse mathématique. Un optimum global correspond au plus petit des optima locaux.

1. Nous effectuons notre étude dans des espaces euclidiens, et donc munis d'une norme. Alors, on peut définir $V_\epsilon(x^*)$ explicitement par $V_\epsilon(x^*) \stackrel{\text{def}}{=} \{x : |x - x^*| \leq \epsilon\}$.

Cette fonction comporte une infinité de minima locaux près de l'origine, mais il est très difficile d'identifier son(ses) minima globaux

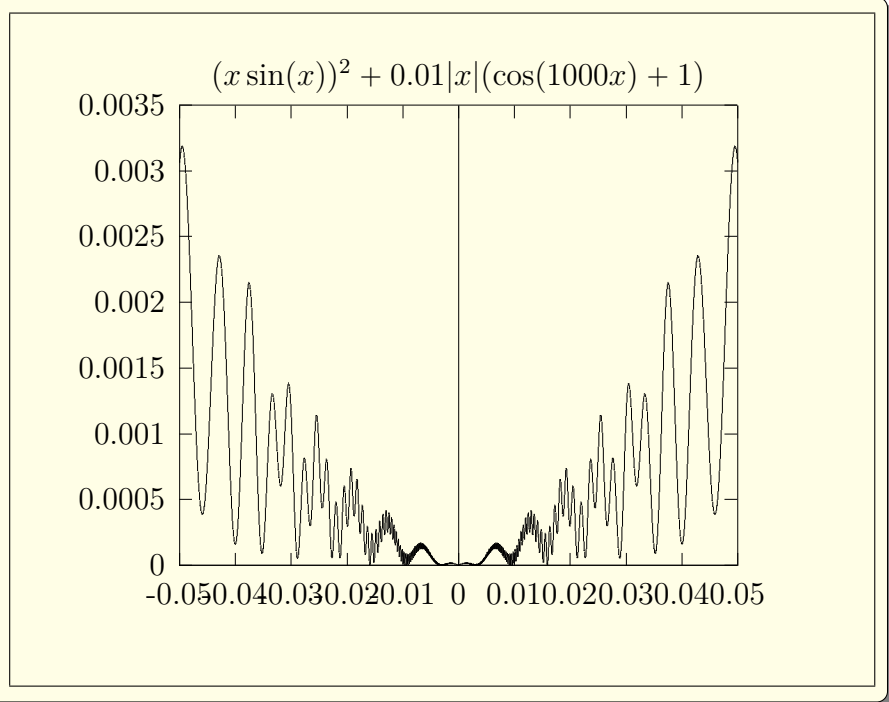


FIGURE 2.1 – Minimisation globale difficile

Sur cette fonction, les deux maxima locaux sont entourés d'hexagones alors que les minima sont encerclés. Un des minima est exposé, et on peut constater que dans le voisinage que décrit le cercle, la fonction atteint son point le plus bas

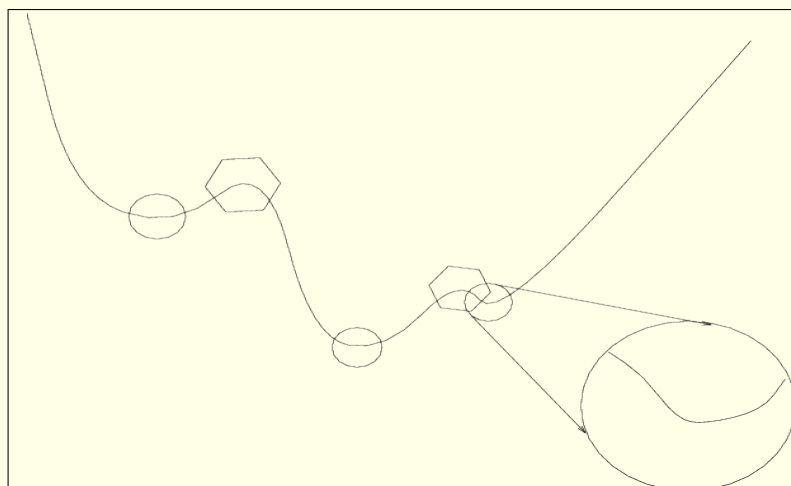


FIGURE 2.2 – Optima locaux

2.2.2 Optima stricts

La notion d'optimum strict concerne le fait que d'autres points du voisinage ne puissent pas avoir la même valeur de fonction que x^* . Par exemple, sur la figure 2.3, les zones encadrées contiennent des minima locaux qui ne sont pas stricts alors que les autres optima locaux sont stricts.

Définition 2.2.1 *Un minimum local x^* est dit strict s'il existe une valeur $\epsilon > 0$ telle que quel que soit $x \in V_\epsilon(x^*)$, $x \neq x^*$, $f(x) > f(x^*)$; l'inégalité est stricte.*

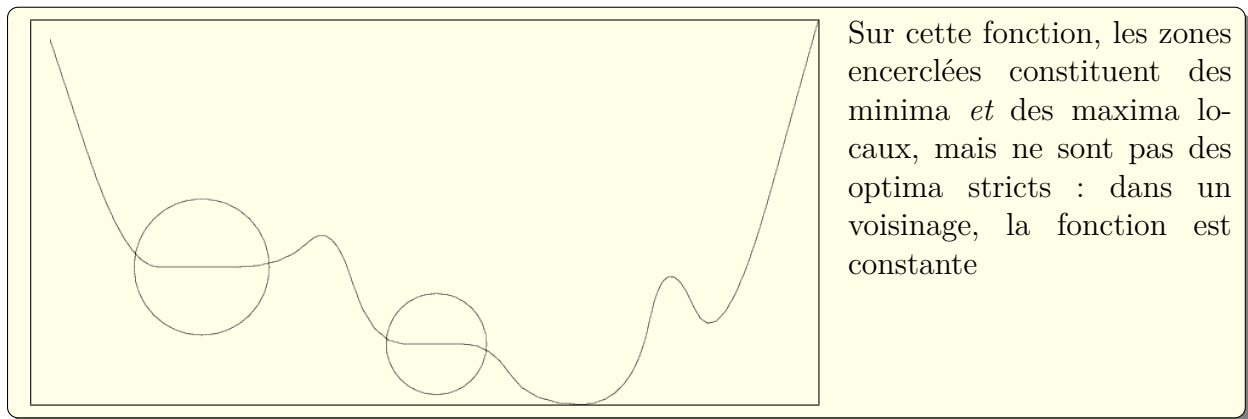


FIGURE 2.3 – Optima stricts

2.2.3 Optima isolés

Sur la figure 2.3, les optima encadrés ne sont pas stricts, mais chaque ensemble d'optima est séparé des autres. La notion d'*optimum isolé* formalise cette situation. Par exemple, la fonction $(x \sin(\frac{1}{x}))^2$ comporte une accumulation de minima locaux à l'origine. L'origine n'est donc pas un optimum isolé ni strict car une infinité de points valent zéro près de l'origine.

Définition 2.2.2 *Un ensemble connexe d'optima O est dit isolé si quelque soit un optimum $x \notin O$, la distance de x à O est bornée inférieurement par une constante positive.*

Exercice 2.2.1 [*Optima isolés*] Trouvez un exemple de fonction qui admet une accumulation de minima locaux à l'origine, pour laquelle l'origine est un minimum global.

2.2.4 Remarques

Chacun des minima locaux de la fonction $(x \sin(1/x))^2$ lorsque x s'approche de zéro est un minimum satisfait à la définition 2.2.1. Cependant, plus le minimum s'approche de zéro, plus la valeur de ϵ diminue de sorte que l'origine ne satisfait plus à la définition. La notion de minimum isolé est semblable. Bien que l'origine ne satisfasse pas à la définition d'un minimum local strict pour la fonction $(x \sin(1/x))^2$, on peut trouver des valeurs de x *aussi proches de zéro* que l'on veut pour lesquelles $(x \sin(1/x))^2 > 0$. Proche de zéro, on a donc une infinité de minima locaux, ainsi qu'une infinité de points qui ne sont pas des minima locaux. L'origine n'est pas un minimum local isolé.

La notion de minimum isolé s'applique à des ensembles connexes de minima locaux. Par exemple, la fonction $f(x) = \max(\sin(x), 0)$ possède plusieurs intervalles $([\pi, 2\pi], [3\pi, 4\pi], \dots)$ de minima locaux. Ces intervalles sont tous à une distance de π , et donc sont isolés. Par opposition, la fonction $f(x) = x^2 \max(\sin(1/x), 0)$ possède également de nombreux intervalles de minima locaux. Cependant, près de l'origine, on retrouve le phénomène que les intervalles sont de plus en plus rapprochés les uns des autres, de sorte que l'on retrouve une infinité de petits intervalles dans tout voisinage de l'origine.

2.3 Conditions d'optimalité

Malgré les classifications que nous venons d'effectuer, il serait difficile, à ce stade, d'identifier ou encore de vérifier qu'un point x^* constitue bel et bien un optimum d'un problème. En effet, les définitions que nous avons exigeraient de comparer la valeur d'une fonction en un point x^* avec sa valeur en une infinité de points voisins de x^* . L'analyse mathématique vient alors à notre secours, et lorsque la fonction possède certaines propriétés de différentiabilité, on peut établir des équations que tout optimum doit satisfaire. Il est alors facile de vérifier qu'un candidat est un optimum : il suffit de vérifier qu'il satisfait aux équations, qui sont nommées *conditions d'optimalité*.

Le but de cette section est d'illustrer le principe des conditions d'optimalité sur des exemples simples de fonctions réelles.

2.3.1 Conditions pour un point stationnaire

On apprend dans les premiers cours de calcul différentiel qu'une fonction réelle possède des minima locaux en des points qui annulent sa dérivée. De tels points sont nommés *points stationnaires*. Si l'on dépose une bille infiniment petite en un point stationnaire du graphe d'une fonction, celle-ci demeure immobile.

Définition 2.3.1 Soit le problème de minimisation $\min_{x \in \mathbb{R}} f(x)$. Un point \bar{x} qui annule la dérivée de f , $f'(\bar{x}) = 0$ est nommé point stationnaire pour le problème de minimisation.

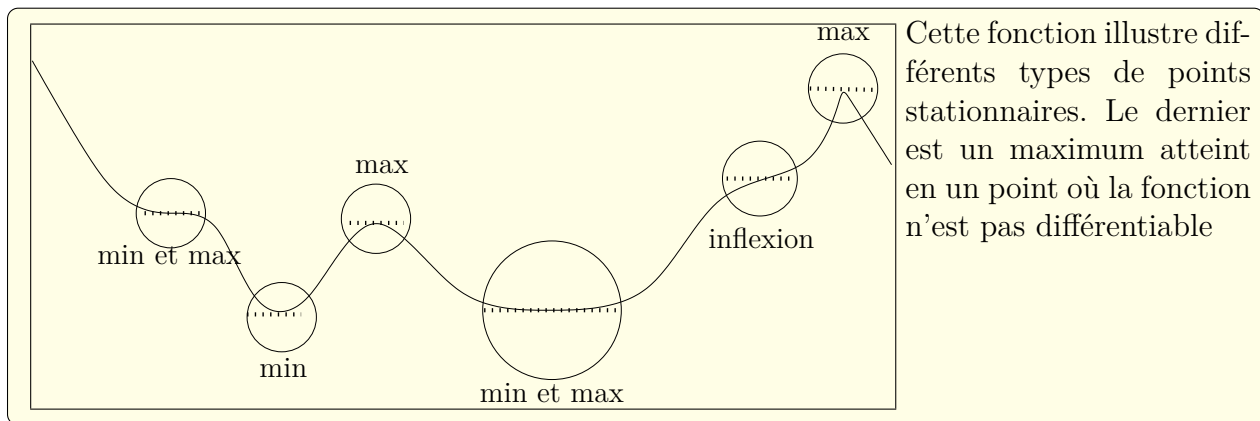


FIGURE 2.4 – Points stationnaires

On observe donc que les points stationnaires, pour une fonction réelle, peuvent être de trois types : minima locaux, maxima locaux ou points d'inflexion. Par conséquent, tout minimum local est un point stationnaire, mais la contraposée n'est pas vraie.

Les points stationnaires satisfont donc à la *condition nécessaire d'optimalité de premier ordre*. Il est remarquable que le résultat ne suppose pas que la fonction est dérivable ailleurs qu'au point x^* . Nous aurons fréquemment besoin d'hypothèses plus fortes, que la fonction est dérivable et même que sa dérivée satisfait à une condition de Lipschitz, mais pour l'instant, il suffit que la dérivée existe en x^* .

Théorème 2.3.1 *Soit x^* un minimum local de la fonction f ; si f est différentiable en x^* . Alors, $f'(x^*) = 0$.*

Corollaire 2.3.1 *Soit x^* un maximum local de la fonction f ; si f est différentiable en x^* . Alors, $f'(x^*) = 0$.*

On peut maintenant utiliser ce résultat pour démontrer le théorème de Rolle :

Théorème 2.3.2 (Rolle) *Soit une fonction f différentiable sur un intervalle $[a, b]$ et telle que $f(a) = f(b)$; alors, il existe un point $c \in [a, b]$ tel que $f'(c) = 0$.*

Ce résultat se généralise dans le théorème dit de la moyenne, ou des accroissements finis :

Théorème 2.3.3 (de la moyenne ou des accroissements finis) *Soit f une fonction différentiable sur un intervalle $[a, b]$. Alors, il existe un point $c \in [a, b]$ tel que $f'(c) = \frac{f(b)-f(a)}{b-a}$.*

Une généralisation importante du théorème de la moyenne concerne le développement de Taylor d'une fonction f , et sera utilisé abondamment tout au long de ce texte.

Théorème 2.3.4 (Taylor) Soit une fonction f de classe C^p au voisinage d'un point x_0 ; on peut approcher la valeur de la fonction en un point voisin $x_0 + d$ de x_0 (d représente un déplacement) par un polynôme

$$f(x_0 + d) \approx \mathcal{P}(d) = f(x_0) + f'(x_0)d + \frac{f''(x_0)}{2}d^2 \dots \frac{f^{(p-1)}(x_0)}{(p-1)!}d^{(p-1)}$$

De plus, on peut exprimer l'erreur commise par un reste :

$$f(x_0 + d) - \mathcal{P}(d) = \frac{f^{(p)}(\xi)}{p!}d^p$$

où le point ξ est situé entre x_0 et $x_0 + d$, et peut être exprimé par $\xi = x_0 + \theta d$ pour une valeur de θ comprise entre 0 et 1, $\theta \in [0, 1]$.

Remarquons que si $p = 1$, en identifiant $x_0 = a$, $x_0 + d = b$ et $\xi = c$, on retrouve le théorème de la moyenne.

2.3.2 Conditions pour un optimum

On apprend aussi dans les cours de calcul que si, en un point stationnaire, la dérivée seconde d'une fonction est positive, il s'agit d'un minimum local ; si elle est négative, il s'agit d'un maximum local ; si elle s'annule, on ne peut rien conclure.

Les points stationnaires qui satisfont à la *condition suffisante d'optimalité de second ordre* ($f''(x^*) > 0$) sont donc des minima.

Par ailleurs, si x^* est un minimum local de la fonction f , alors *nécessairement* on a $f'(x^*) = 0$ et $f''(x^*) \geq 0$. En effet, si la condition $f''(x^*) \geq 0$ n'est pas satisfaite, x^* satisfait à la condition suffisante pour un maximum de la fonction f .

Le cas $f''(x^*) = 0$ comporte une ambiguïté, illustrée par les fonctions $f_1(x) = x^4$ et $f_2(x) = -x^4$. En $x^* = 0$, pour les 2 fonctions, on a $f'(0) = f''(0) = 0$ bien que l'origine soit un minimum pour f_1 et un maximum pour f_2 .

Exercice 2.3.1 [Condition d'optimalité]

- a) L'origine est un point stationnaire, c'est-à-dire un point pour lequel $f'(x) = 0$, de la fonction $f(x) = x^2 \cos x$. Déterminez s'il s'agit d'un minimum (local), maximum (local) ou point d'inflexion.
- b) Même question pour les fonctions
 - i) $f(x) = x^2 \sin x$;
 - ii) $f(x) = x^2(1 - \cos x)$;
 - iii) $f(x) = x^2(\cos x - 1)$.

Exercice 2.3.2 [Condition d'optimalité]

a) Montrez que la fonction

$$f(x) = (2 - x^2)^2 + x^5$$

possède deux points stationnaires, un maximum et un minimum local.

b) Trouvez un exemple de fonction qui possède trois points stationnaires, un minimum local, un maximum local et un point d'inflexion.

Théorème 2.3.5 Soit f de classe \mathcal{C}^2 . Un point x^* qui satisfait à la condition suffisante d'optimalité ($f'(x^*) = 0$ et $f''(x^*) > 0$) est un minimum local strict et isolé.

Preuve Montrons que x^* est un minimum local strict. Exprimons f à l'aide d'un développement de Taylor d'ordre 2 autour de x^* :

$$f(x) = f(x^*) + f'(x^*)(x - x^*) + \frac{1}{2}f''(\xi)(x - x^*)^2,$$

avec ξ compris entre x et x^* . Les conditions suffisantes nous instruisent que $f'(x^*) = 0$ alors que $f''(x^*) > 0$. Posons $0 < A = f''(x^*)$. Puisque f'' est une fonction continue, choisissons un voisinage V de x^* $V = \{x : |x - x^*| \leq \epsilon\}$ tel que quel que soit $\xi \in V$, $\frac{A}{2} < f''(\xi) < 2A$. Alors, dans ce voisinage,

$$\begin{aligned} f(x) - f(x^*) &= f'(x^*)(x - x^*) + \frac{1}{2}f''(\xi)(x - x^*)^2 \\ &= \frac{(x - x^*)^2}{2}f''(\xi) \\ &> \frac{(x - x^*)^2}{2} \frac{A}{2}. \end{aligned}$$

Montrons maintenant que x^* est un minimum isolé. Pour ce faire, nous allons vérifier que $f'(x) \neq 0$ dans un voisinage de x^* . Utilisons le théorème de la moyenne dans V

$$f'(x) - f'(x^*) = f''(\xi)(x - x^*).$$

Les conditions suffisantes nous informent que $f'(x^*) = 0$ alors que $f''(x^*) > 0$. Donc, si $x \in V$ et $x \neq x^*$, $|f'(x)| > \frac{A}{2}|x - x^*| > 0$. \square

Exercice 2.3.3 [Analyse d'une fonction] Considérez la fonction $h(\theta) = \frac{4\theta-7}{\theta^2+\theta-2}$.

- Identifiez *tous* les points stationnaires de h ;
- parmi ceux-ci, identifiez *tous* les minima locaux de h ;
- identifiez deux points où la fonction n'est pas bornée.

Exercice 2.3.4 [Analyse d'une fonction] Considérez la fonction $h(\theta) = 1 - 12\theta + 7.5\theta^2 - \theta^3$.

- Identifiez *tous* les points stationnaires de h ;
- parmi ceux-ci, identifiez *tous* les minima locaux de h .

Exercices géométriques

Exercice 2.3.5 [Hyperboles] Suite de l'exercice 1.4.2 page 25 Considérez une hyperbole de \mathbb{R}^2 , $y = 1/x$. Soit un point $P = \begin{pmatrix} P_x \\ P_y \end{pmatrix}$, on veut trouver le point de l'hyperbole le plus proche de P . Comme les points de l'hyperbole sont de la forme $\begin{pmatrix} x \\ 1/x \end{pmatrix}$, on peut ramener l'étude à la minimisation selon la seule variable x de la fonction $d(x) = (x - P_x)^2 + (\frac{1}{x} - P_y)^2$.

- Écrivez les conditions d'optimalité (d'ordre un et deux) de ce problème $\min_{x \in \mathbb{R}} d(x)$.
- Si le point P est sur la droite $y = x$, on devine qu'il y a 2 candidats solution naturels : $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ et $\begin{pmatrix} -1 \\ -1 \end{pmatrix}$. En vous limitant aux $x > 0$, vérifiez cette intuition en utilisant les conditions d'optimalité ; vérifiez d'abord que $x = 1$ est un point stationnaire pour $d(x)$. Vérifiez ensuite que l'intuition n'est pas vraie si P est assez loin de l'origine ; quantifiez ce "loin" de l'origine en utilisant les conditions d'optimalité d'ordre 2.)

Exercice 2.3.6 [Cosinus] Suite de l'exercice 1.4.3 page 25 On dispose d'un point $P_0 = (x_0, y_0)^t$ et on veut obtenir le point le plus proche de P_0 sur la courbe $P(x) = (x, \cos x)^t$. Formulez ce problème comme problème de minimisation $\min_{x \in \mathbb{R}} f(x)$ selon la variable x , donc fournissez une expression pour $f(x)$.

- Écrivez les conditions d'optimalité (d'ordre un et deux) de ce problème $\min_{x \in \mathbb{R}} d(x)$.
- Impossible d'aller plus loin dans le cas général sans outil de calcul. Considérons donc un point $P_0 = (0, y_0)^t$. Vérifiez que $x^* = 0$ satisfait aux conditions nécessaire d'ordre un

pour le problème. Donc, le candidat pour le point le plus proche (minimum local) est $P^* = (x^*, \cos x^*)^t = (0, 1)^t$.

- c) Le candidat P^* n'est pas un minimum local de la distance pour n'importe quelle valeur de y_0 . Utilisez les conditions d'ordre deux pour obtenir les valeurs de y_0 garantissant que $P^* = (0, 1)^t$ est bel et bien minimum local de la distance.

Exercice 2.3.7 [*Norme et somme de normes*] Soient deux fonctions $f_1(x) = x$ et $f_2(x) = 2(x - 1)$.

- a) Considérez la somme des normes (valeurs absolues dans ce cas simpliste) $f(x) = |f_1(x)| + |f_2(x)|$. Vérifiez que $f(1) = 1$ et $f(x) \geq 1$ pour tout x . Indice : justifiez que f est de la forme

$$f(x) = \begin{cases} \text{expression 1,} & \text{si } x \leq 0 \\ \text{expression 2,} & \text{si } 0 \leq x \leq 1 \\ \text{expression 3,} & \text{si } x \geq 1 \end{cases}$$

- b) Considérez maintenant la somme des normes au carré, $\hat{f}(x) = f_1(x)^2 + f_2(x)^2$. Calculez x^* un minimum de $\hat{f}(x)$ en utilisant les conditions d'optimalité.
- c) Utilisez ce qui précède pour justifier que minimiser une somme de normes n'est pas équivalent à minimiser une somme de carrés de normes.

2.4 Algorithmes de descente

Maintenant que nous savons caractériser des optima locaux par les conditions d'optimalité, penchons-nous sur le problème de calculer, d'identifier un optimum local d'une fonction. Le principe du calcul est d'utiliser un *algorithme itératif* : à partir d'un point qui ne satisfait pas aux conditions d'optimalité, on va produire un autre point, meilleur que le précédent. Puis, on va recommencer, et engendrer une suite de points qui s'approche d'un optimum.

Nous avons vu que la recherche de minima locaux d'une fonction peut passer par la recherche de points stationnaires, c'est-à-dire, de racines de la dérivée de la fonction $f'(x) = 0$. Cependant, ce qui distingue les problèmes d'optimisation des problèmes d'équations, c'est justement la fonction f , qui permet de comparer la qualité de deux points : puisque l'on veut minimiser f , il est intuitif de préférer un point x_1 à un point x_2 lorsque $f(x_1) < f(x_2)$. Nous allons tenter de formaliser cette intuition en engendrant une suite de points $\{x_n\}$ telle que pour tous les membres de la suite, $f(x_{i+1}) < f(x_i)$. La méthode qui produit une telle suite de points est nommée *algorithme de descente*.

Si un point n'est pas stationnaire, alors le signe de la dérivée de la fonction objectif permet de savoir de quel côté la fonction diminue. On peut donc envisager de rechercher un minimum local en effectuant de petits déplacements dans la direction de diminution de la fonction. Deux indices nous permettent de savoir que nous avons cerné un minimum local :

- la fonction se met à augmenter, c'est-à-dire $f(x_k) > f(x_{k-1})$ pour la première fois, donc $f(x_i) < f(x_{i-1}) \forall i < k$;
- la pente de la fonction change de signe pour la première fois : $f'(x_k)f'(x_{k-1}) < 0$ et $f'(x_i)f'(x_{i-1}) > 0 \forall i < k$.

Si jamais on n'a pas cerné de minimum local après un très grand nombre d'essais, c'est un indice que la fonction f est *peut-être* non-bornée inférieurement, ou encore qu'il n'existe pas de x^* tel que $f(x^*) = \inf f(x)$, puisque nous avons identifié de nombreux termes d'une suite présumée monotone décroissante.

```

Trouve_intervalle
{ Donnés :  $x_0, \delta > 0$  et  $0 < M < \infty$  }
cerné ← faux
borné ← vrai
 $x^+ \leftarrow x_0$ 
tantque (  $\neg$  cerné  $\wedge$  borné )
   $x \leftarrow x^+$ 
   $x^+ \leftarrow x - \text{signe}(f'(x)) * \delta$ 
  cerné ←  $(f(x^+) > f(x)) \vee (f'(x) * f'(x^+) < 0)$ 
  borné ←  $|x| < M$ 

```

Algorithme 2.1: Trouve intervalle.

Cette idée simple est à la base des algorithmes de descente. On voit que les algorithmes de descente sont construits à partir de points qui *ne satisfont pas* aux conditions d'optimalité. On démontre habituellement un théorème de convergence globale dont l'énoncé ressemble au théorème 2.4.1. Observons que l'algorithme `Trouve_intervalle` est codé à l'aide de variables informatiques : les variables x et x^+ représentent deux points successifs dans la suite $\{x_k\}$, c'est-à-dire x_k et x_{k+1} pour une certaine valeur de k .

Théorème 2.4.1 *Soit $M = +\infty$ dans l'algorithme `Trouve_intervalle`. Alors, si l'algorithme termine, l'intervalle d'extrémités x_{k-2} et x_k contient un minimum local de la fonction f , supposée de classe \mathcal{C}^1 .*

Preuve Si l'algorithme termine, alors au moins une des deux conditions de terminaison est satisfaite. Vérifions que chacune de ces conditions assure que l'intervalle contient un minimum local.

- $f(x_k) > f(x_{k-1})$ et $f(x_{k-1}) < f(x_{k-2})$. Puisque f est continue, elle atteint son minimum sur l'intervalle d'extrémités x_{k-2} et x_k ; de plus, puisque son minimum est de valeur au moins aussi petite que $f(x_{k-1})$, il existe un point à l'intérieur de l'intervalle qui minimise f sur l'intervalle. Ce minimum, global sur l'intervalle, est en particulier un minimum local.
- $f'(x_k)f'(x_{k-1}) < 0$. Supposons sans perte de généralité que $x_k < x_{k-1}$ et que $f'(x_k) < 0$. Alors, $f'(x_{k-1}) > 0$. Donc, puisque f' est une fonction continue, elle doit traverser l'axe des X en au moins un point x appartenant à l'intervalle, et ce de manière croissante.

□

Remarquons que cet algorithme possède la propriété de *convergence globale*, c'est-à-dire qu'aucune hypothèse n'est requise sur les propriétés du point initial. Cependant, il ne cerne que des *minima locaux*. Il est donc important de différencier la notion de minimum local et global qui précise la nature du minimum, et la notion de convergence globale d'un algorithme qui garanti la convergence peu importe l'estimé initial.

2.4.1 Réduction d'intervalle avec f'

Une fois qu'un intervalle contenant un minimum local est cerné, on utilise des techniques de *réduction d'intervalle* pour améliorer notre estimé de la solution. Par exemple, l'*algorithme de bisection* évalue la dérivée de f au point milieu de l'intervalle, et conserve la moitié de l'intervalle pour laquelle les dérivées des extrémités sont de signe opposé. Il serait dangereux d'interrompre l'algorithme lorsque la dérivée au point milieu s'annule : rien ne garantirait alors que ce point stationnaire est un minimum local. Tel qu'énoncé, l'algorithme est assuré de cerner soit un minimum local ou un point d'inflexion (mais pas un maximum local) car les conditions assurent que $f'(x_1)$ demeure toujours positif alors que $f'(x_0)$ demeure toujours négatif ou nul.

Bisection

{ Données : $x_0 < x_1$ tels que $f'(x_0) \leq 0$ et $f'(x_1) \geq 0$; }

{ $f'(x_1) - f'(x_0) > 0$ }

{ un critère d'arrêt, ϵ . }

répéter

| $x_m \leftarrow (x_0 + x_1)/2$

| **si** ($f'(x_m) \leq 0$) **alors** $x_0 \leftarrow x_m$

| **sinon** $x_1 \leftarrow x_m$

jusqu'à ($|x_1 - x_0| < \epsilon$)

Algorithme 2.2: Réduit intervalle—bisection.

Théorème 2.4.2 *L'algorithme Bisection cerne un point stationnaire \bar{x} tel que $f''(\bar{x}) \geq 0$.*

Preuve Les conditions de l'algorithme assurent que $f'(x_0) \leq 0$ alors que $f'(x_1) > 0$. Puisque $x_1 - x_0 > 0$, et que $(x_1 - x_0) \rightarrow 0$, $(f(x_1) - f(x_0)) \rightarrow 0$,

$$0 \leq \lim_{(x_1 - x_0) \rightarrow 0} \frac{f'(x_1) - f'(x_0)}{x_1 - x_0} = f''(\bar{x}).$$

□

L'algorithme **Bissection** assure la convergence vers un point stationnaire pour lequel la dérivée n'est pas décroissante. C'est un point qui satisfait aux conditions nécessaires d'optimalité. Il est en général impossible de concevoir un algorithme convergeant vers un point satisfaisant aux conditions suffisantes d'optimalité.

Exercice 2.4.1 [*Point milieu stationnaire*]

- Trouvez un exemple (une fonction f et un intervalle $[a, b]$) pour lequel les dérivées de f sont de signe opposé aux extrémités de l'intervalle ($f'(a) < 0$ et $f'(b) > 0$) pour lequel le point milieu de l'intervalle est un maximum local.
- Trouvez un pareil exemple pour lequel le point milieu de l'intervalle est un point d'inflexion.
- Vérifiez que l'algorithme 2.2 ne converge pas vers un maximum local pour vos deux exemples.

Le tableau 1.1 illustre l'application de l'algorithme avec la fonction $e^{x(x-1)}$ et l'intervalle initial $[-2, 1.8349]$.

L'algorithme de bisection requiert un intervalle de départ qui satisfait à la seconde condition de terminaison de l'algorithme qui identifie un intervalle cernant un minimum local. Si jamais l'algorithme 2.1 (`trouve_intervalle`) avait terminé avec un intervalle satisfaisant à la première condition, c'est-à-dire $f(x_k) > f(x_{k-1})$ et $(f(x_{k-1}) < f(x_i) \forall i < k - 2)$, alors il faudrait d'abord identifier un point de l'intervalle de bornes x_k et x_{k-1} dont la dérivée est de signe opposé à la dérivée en x_0 .

Exercice 2.4.2 [*Initialisation de bisection*]

- Dans le contexte de la première condition de terminaison de l'algorithme de recherche d'intervalle, justifiez qu'il existe un point de l'intervalle de bornes x_k et x_{k-1} dont la dérivée est de signe opposé à la dérivée en x_0 ;
- Proposez une technique de réduction d'intervalle pour identifier un tel point.

iter	x_0	x_1	$f'(x_0)$	$f'(x_1)$
0	-2.000000000	1.834900000	-2017.1439674	12.353773626
1	-0.082550000	1.834900000	-1.2740125688	12.353773626
2	-0.082550000	0.8761749999	-1.2740125688	0.6749976921
3	0.3968125000	0.8761749999	-0.1624455002	0.6749976921
4	0.3968125000	0.6364937500	-0.1624455002	0.2166009131
5	0.3968125000	0.5166531249	-0.1624455002	0.0259461281
6	0.4567328125	0.5166531249	-0.0675193202	0.0259461281
7	0.4866929687	0.5166531249	-0.0207307233	0.0259461281
8	0.4866929687	0.5016730468	-0.0207307233	0.0026059477
9	0.4941830078	0.5016730468	-0.0090608627	0.0026059477
10	0.4979280273	0.5016730468	-0.0032273217	0.0026059477
11	0.4998005371	0.5016730468	-0.0003106837	0.0026059477
12	0.4998005371	0.5007367919	-0.0003106837	0.0011476289
13	0.4998005371	0.5002686645	-0.0003106837	0.0004184723
14	0.4998005371	0.5000346008	-0.0003106837	0.0000538943
15	0.4999175689	0.5000346008	-0.0001283947	0.0000538943
16	0.4999760848	0.5000346008	-0.0000372501	0.0000538943
17	0.4999760848	0.5000053428	-0.0000372501	0.0000083220
18	0.4999907138	0.5000053428	-0.0000144640	0.0000083220
19	0.4999980283	0.5000053428	-0.0000030710	0.0000083220
20	0.4999980283	0.5000016856	-0.0000030710	0.0000026255

TABLE 2.1 – Itérations de bisection sur la fonction $e^{x(x-1)}$.

Exercices géométriques

Exercice 2.4.3 [Parabole] Suite de la discussion de la section 1.4.2 pour la fonction $d(x) = (x - P_x)^2 + (x^2 - P_y)^2$.

- À partir de $x = 1$, appliquez l'algorithme `trouve_intervalle` pour le point $P = \left(\frac{1}{2}\right)$.
- À partir de l'intervalle trouvé en a), effectuez une étape de l'algorithme `bissection` (toujours pour le point $P = \left(\frac{1}{2}\right)$).
- Recherchez d'autres points stationnaires pour ce problème, et identifiez leur nature.

Exercice 2.4.4 [Ellipse] Suite de l'exercice 1.4.1 page 24

- Vérifiez qu'en prenant parmi les points sur les axes les deux plus proche de l'origine, on ne peut pas appliquer directement l'algorithme de bisection car une des dérivées est $-\infty$.
- Comme la dérivée à l'autre point est positive, faites comme si la dérivée infinie était finie (et négative) et effectuez des itérations de bisection jusqu'à ce que le point à dérivée infinie soit remplacé.

Exercice 2.4.5 [Hyperboles] Suite de l'exercice 2.3.5 page 58

- À partir de $x = 1$, appliquez l'algorithme `trouve_intervalle` pour le point $P = \left(\frac{1}{2}\right)$; utilisez un paramètre $\delta = 0.5$.
- À partir de l'intervalle trouvé en a), effectuez une étape de l'algorithme `bissection` (toujours pour le point $P = \left(\frac{1}{2}\right)$).

2.4.2 Réduction d'intervalles sans utiliser la dérivée

Dans certaines applications, la dérivée de f peut-être difficile à calculer. Il est possible de concevoir une méthode de réduction d'intervalle qui n'utilise que la fonction f . Supposons tout de même que la fonction f est continue. Soient trois points $a < x_1 < b$. Si $f(x_1) < f(a)$ et $f(x_1) < f(b)$, alors l'intervalle $[a, b]$ contient certainement un minimum local de f . Maintenant, choisissons un point $x_2 \neq x_1$ dans l'intervalle $[a, b]$. Sans perte de généralité, supposons que $x_2 > x_1$ et que $f(x_2) \leq f(x_1)$. Nous discuterons du cas limite où $f(x_2) = f(x_1)$ plus loin. Donc, pour l'instant, supposons que $f(x_2) < f(x_1)$. Alors, le triplet $\bar{a} = x_1 < x_2 < b$

satisfait aux mêmes conditions que le triplet $a < x_1 < b$, c'est-à-dire $f(x_2) < f(\bar{a})$ et $f(x_2) < f(b)$. On a donc réduit l'intervalle $[a, b]$ à l'intervalle $[\bar{a}, b]$ en maintenant un point intérieur à l'intervalle de valeur de fonction plus petite que ses extrémités.

Si le cas $f(x_1) = f(x_2)$ ne se présente jamais, nous avons en main une stratégie de réduction d'intervalle qui cerne un minimum local de f à condition que f possède la propriété que tous ses minimas locaux stricts dans $[a, b]$ sont *uniformément isolés*, c'est-à-dire qu'il existe un ϵ tel que pour tout minimum local strict x^* de f dans $[a, b]$, $f(x) > f(x^*)$, $\forall x : |x - x^*| \leq \epsilon$.

Maintenant, si le cas $f(x_1) = f(x_2)$ se présente infiniment souvent, il est possible que le point cerné appartienne à une zone plateau de la fonction f , dont l'intérieur est constitué de minima locaux dans un sens faible.

Il faut également s'assurer que le choix de x_2 conduit bel et bien à une suite d'intervalles imbriqués dont la longueur converge à zéro. Nous avons maintenant tout ce qu'il faut pour montrer la convergence globale d'une famille de techniques de réduction d'intervalles n'utilisant que la fonction f .

Lemme 2.4.1 *Soit un intervalle $[a^0, b^0]$ et une fonction f , continue sur $[a^0, b^0]$ et un point x_1^0 tels que*

1. $f(a^0) \geq f(x_1^0)$,
2. $f(b^0) \geq f(x_1^0)$
3. *il existe un ϵ tel que tout ensemble connexe de minima locaux de f est isolé de tout autre minimum local par une distance d'au moins ϵ ;*

soit également un choix x_2^k tel que $\frac{|x_2^k - x_1^k|}{b^k - a^k} \rightarrow 0$, $\frac{|x_2^k - a^k|}{b^k - a^k} \rightarrow 0$ et $\frac{|x_2^k - b^k|}{b^k - a^k} \rightarrow 0$. Alors, $\lim_{k \rightarrow \infty} a^k = \lim_{k \rightarrow \infty} x_1^k = \lim_{k \rightarrow \infty} b^k = \bar{x}$ et \bar{x} est un minimum local de f dans $]a^0, b^0[$.

Preuve Pour une infinité de points $a^k < \bar{x} : |a - \bar{x}| < \epsilon$ et une infinité de points $b^k > \bar{x} : |b^k - \bar{x}| < \epsilon$, on a $f(a^k) \geq f(\bar{x})$ et $f(b^k) \geq f(\bar{x})$.

Raisonnons par l'absurde et supposons que \bar{x} n'est pas un minimum local de f . Il existe alors une suite de points \tilde{x}^j qui s'approchent de \bar{x} tels que $f(\tilde{x}^j) < f(\bar{x})$. La suite \tilde{x}^j est entrelacée avec au moins une des suites a^k ou b^k . Par conséquent, chacun des \tilde{x}^j est encadré d'un couple c^{k_j}, c^{k_j+1} où $c = a$ ou b . Donc, pour chaque j , on a soit un triplet $a^{k_j} < \tilde{x}^j < a^{k_j+1}$, soit $b^{k_j} < \tilde{x}^j < b^{k_j+1}$ dont les extrémités définissent un intervalle qui doit contenir un minimum local de f , ce qui contredit que les minima locaux sont isolés uniformément d'une distance d'au moins ϵ . \square

L'algorithme de Fibonacci utilise ce principe, et de plus, choisit x_1 et x_2 de telle sorte qu'après avoir conservé le triplet a, x_1, x_2 (ou encore le triplet x_1, x_2, b si $f(x_2) < f(x_1)$), le point x_1 joue le rôle d'un des deux points intermédiaires dans le nouvel intervalle $[a, b]$. Ceci évite de recalculer $f(x_1)$. Le principe est le suivant. On dénote $T^k = b^k - a^k$, de sorte que $T^{k+1} = b^k - x_1^k = x_2^k - a^k$ et $T^{k+2} = x_1^k - a^k = b^k - x_2^k$ les longueurs des intervalles de trois itérations consécutives. Alors, $T^k = T^{k+1} + T^{k+2}$ et on reconnaît la récurrence inverse de

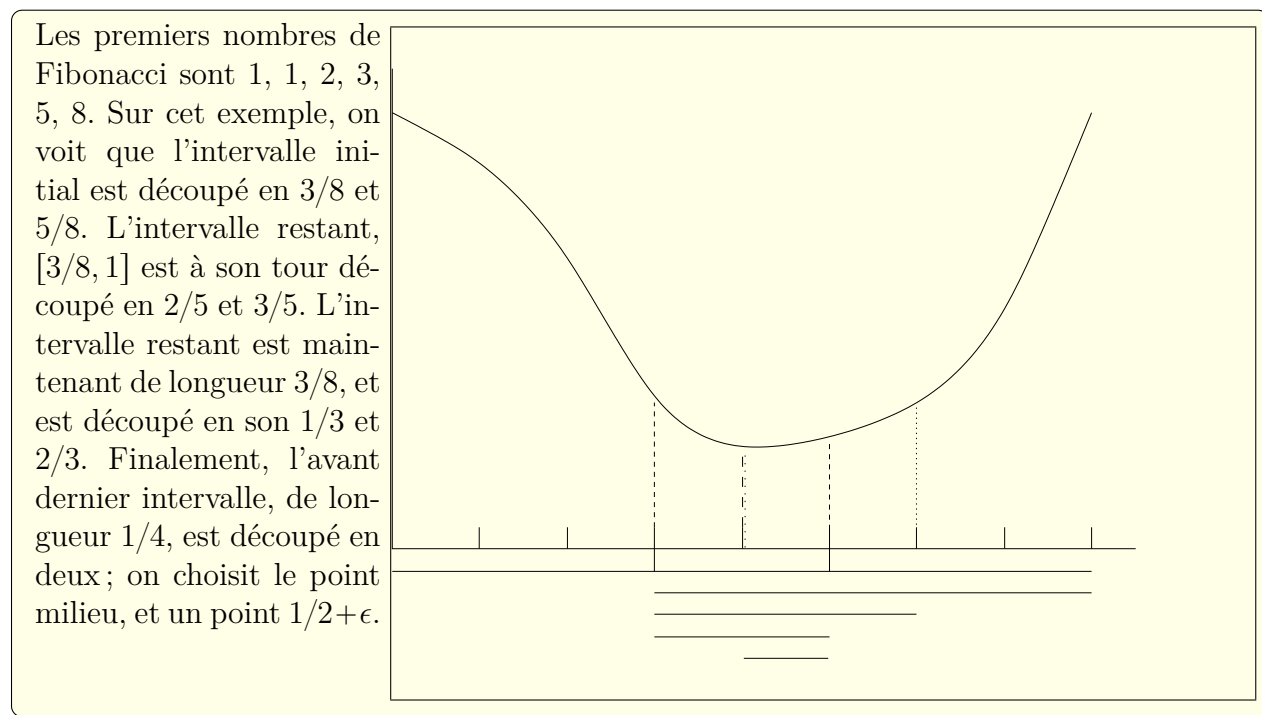


FIGURE 2.5 – Algorithme de Fibonacci

celle des nombres de Fibonacci. On fixera donc *a priori* le nombre de réduction d'intervalles pour que la dernière proportion soit $\frac{1}{2} - \frac{1}{2}$, soit $\frac{F_1}{F_3} - \frac{F_2}{F_3}$, où F_i est le $i^{\text{ème}}$ nombre de la suite de Fibonacci, 1, 1, 2, 3, 5, ... définie par $F_1 = 1$, $F_2 = 1$, $F_{k+2} = F_{k+1} + F_k$.

Un détail nous reste à régler pour avoir une méthode d'application générale : si le point x_1^0 est produit par la méthode, rien ne garanti que $f(x_1^0) < f(a^0)$. On va donc supposer que les dérivées directionnelles de f aux extrémités du segment $[a^0, b^0]$ sont de signes respectivement négatif et positif, assurant ainsi l'existence d'un minimum local de f dans l'intervalle.

Théorème 2.4.3 *Soit un intervalle $[a^0, b^0]$ et une fonction f , continue sur $[a^0, b^0]$ croissante en b^0 et décroissante en a^0 et telle qu'il existe un ϵ tel que tout ensemble connexe de minima locaux de f est isolé de tout autre minimum local par une distance d'au moins ϵ . Soit un algorithme qui choisit x_1^0 arbitrairement, et un choix x_2^k tel que $\frac{|x_2^k - x_1^k|}{b^k - a^k} \rightarrow 0$, $\frac{|x_2^k - a^k|}{b^k - a^k} \rightarrow 0$ et $\frac{|x_2^k - b^k|}{b^k - a^k} \rightarrow 0$. Alors, $\lim_{k \rightarrow \infty} a^k = \lim_{k \rightarrow \infty} x_1^k = \lim_{k \rightarrow \infty} b^k = \bar{x}$ et \bar{x} est un minimum local de f dans (a^0, b^0) .*

Preuve La difficulté nouvelle est la nature arbitraire de x_1^0 . Si jamais une situation se présente telle que $f(x_1^0) > f(x_2^0) > f(b^0)$ (ou sa symétrique), et que l'algorithme conserve $a^1 = x_1^0$, alors pour un certain nombre d'étapes, on peut se retrouver avec la situation $f(a^k) > f(x_1^k) > f(b^k)$. Cependant, dans ce cas, comme les valeurs de a^k sont monotones

croissantes, et puisque f est croissante en $b^k = b^{k-1} = \dots = b^0$, tôt ou tard, $f(x_1^{\bar{k}})$ deviendra plus petit que $f(b^0)$, et à partir de ce moment, le lemme 2.4.1 s'applique, assurant que l'algorithme s'approche d'un minimum local. \square

Fibonacci

```

{ Donnés :  $a$  et  $b$  tels que  $f'(a) < 0$  et  $f'(b) > 0$ ; }
{ un nombre de réductions de l'intervalle,  $n$ ; }
{ la suite des nombres de Fibonacci }
{  $F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2}, n = 2 \dots$  }
{ Hypothèse : les minima locaux de  $f$  sont uniformément isolés. }
 $L \leftarrow b - a$ 
 $x_1 \leftarrow a + \frac{F_n}{F_{n+2}} * L$ 
 $x_2 \leftarrow a + \frac{F_{n+1}}{F_{n+2}} * L$ 
 $f_1 \leftarrow f(x_1); f_2 \leftarrow f(x_2)$ 
 $k \leftarrow 1$ 
répéter
  | si ( $f_1 < f_2$ ) alors
  |   |  $b \leftarrow x_2; x_2 \leftarrow x_1$ 
  |   |  $x_1 \leftarrow (a * F_{n-k+1} + b * F_{n-k}) / F_{n-k+2}$ 
  |   |  $f_2 \leftarrow f_1; f_1 \leftarrow f(x_1)$ 
  |   | sinon
  |   |   |  $a \leftarrow x_1; x_1 \leftarrow x_2$ 
  |   |   |  $x_2 \leftarrow (b * F_{n-k+1} + a * F_{n-k}) / F_{n-k+2}$ 
  |   |   |  $f_1 \leftarrow f_2; f_2 \leftarrow f(x_2)$ 
  |   |  $k \leftarrow k + 1$ 
  | jusqu'à ( $k = n$ )
  |  $x_2 \leftarrow x_2 + \epsilon$ 
  |  $f_2 \leftarrow f(x_2)$ 
  | si ( $f_1 < f_2$ ) alors
  |   | résultat  $\leftarrow [a, x_2]$ 
  | sinon
  |   | résultat  $\leftarrow [x_1, b]$ 

```

Algorithme 2.3: Réduit intervalle—Fibonacci.

Exercice 2.4.6 [Applications d'algorithmes]

- a) Pour la fonction $f(x) = e^{x(x-1)}$, effectuez trois itérations des deux algorithmes (bisection, Fibonacci) à partir de l'intervalle initial $[-1, 1]$.

- b) Même question pour la fonction $f(x) = x^2(1 - \cos x)$ sur l'intervalle $[-1, 0.5]$.
- c) Un inconvénient de la méthode de Fibonacci est de devoir fixer *a priori* le nombre d'itérations; si on se fixait une infinité d'itérations, en sachant que

$$\lim_{i \rightarrow \infty} \frac{F_{i+1}}{F_i} = \tau,$$

le nombre d'or $\tau = 1 + \frac{1}{\tau} = \frac{(1+\sqrt{5})}{2} = 1.618\dots$, on utiliserait ce rapport à chaque itération. Cette méthode, nommée *section dorée*, consiste donc à utiliser

$$a, \quad a + (2 - \tau)(b - a), \quad a + (\tau - 1)(b - a), \quad b$$

comme quatre points initiaux, et à conserver ce rapport à chaque itération. Effectuez trois itérations de cet algorithme sur les fonctions de a) et b).

Réduction d'intervalle utilisant f et f'

Nous venons de voir qu'il est possible, à partir d'un intervalle contenant un minimum local, de le réduire systématiquement en utilisant une bisection sur la dérivée $f'(x)$, ou encore directement en utilisant $f(x)$. Si on dispose de f et f' , on devrait pouvoir s'en servir. En fait, l'algorithme `trouve_intervalle` fournit un intervalle qui garanti inclure un minimum local, bien que ne satisfaisant pas aux conditions de l'algorithme de bisection sur $f'(x)$. Voici une variante de bisection qui maintient pour le sous-intervalle conservé les conditions de sortie de `trouve_intervalle`. Supposons que l'intervalle courant soit x_g, x_d et que l'extrémité gauche soit celle pour laquelle $f'(x_g) < 0$, et donc soit $f(x_d) > f(x_g)$, soit $f'(x_d) \geq 0$; les intervalles imbriqués que nous allons produire satisfont tous à l'invariant

$$(f'(x_g) < 0) \wedge (f(x_d) > f(x_g) \vee f'(x_d) \geq 0).$$

Nommons x_s le point suivant calculé. Alors, si $f(x_s) > f(x_g)$, on remplace x_d par x_s et l'invariant est satisfait. Autrement, $f(x_s) \leq f(x_g)$ et donc si $f'(x_s) < 0$, on remplace x_g par x_s , sinon, on remplace x_d par x_s . Le cas où $f'(x_d) > 0$ et soit $f(x_g) > f(x_d)$, soit $f'(x_g) \leq 0$ se traite similairement.

Exercice 2.4.7 [*Bisection utilisant f et f'*] Écrivez l'algorithme complet de bisection applicable dès la sortie de l'algorithme `trouve_intervalle`.

2.5 Algorithmes d'approximation polynomiale

Les algorithmes de la section précédente sont intéressants car ils assurent de cerner un minimum local. Cependant, comme nous allons l'étudier plus tard, ces algorithmes ne convergent pas vite. Il est donc d'usage de changer d'algorithme lorsque l'intervalle contenant un minimum local est assez petit. Nous verrons plus loin ce que nous entendons par assez petit. Penchons nous maintenant sur une famille d'algorithmes consistant à approcher la fonction f ou la fonction f' par un polynôme.

Avant d'exposer des algorithmes, replaçons nous dans le contexte que les algorithmes sont déduits de la non-satisfaction de conditions d'optimalité.

Proposition 2.5.1 *Soit f une fonction de classe \mathcal{C}^2 et x^* un minimum local de f . Alors, $d^* = 0$ est un minimum local de la fonction $q_{x^*}(d) = f(x^*) + f'(x^*)d + \frac{1}{2}f''(x^*)d^2$.*

Preuve Nous savons que $f'(x^*) = 0$ et $f''(x^*) \geq 0$. Par conséquent, $f'(x^*)d + \frac{1}{2}f''(x^*)d^2 \geq 0$ pour tout $d \neq 0$. \square

Si x_0 n'est pas un minimum local, nous envisageons utiliser $d_0 = \arg \min q_{x_0}(d)$ pour produire un prochain point $x_1 = x_0 + d_0$. Nous présentons dans cette section diverses variantes d'approximation polynomiale de la fonction f , dont $q_x(d)$ est l'exemple le plus commun.

2.5.1 Approximation quadratique de la fonction f

Plusieurs stratégies permettent d'approcher (localement) la fonction f par un polynôme p de degré deux. La plus naturelle, lorsque la fonction f est de classe \mathcal{C}^2 est d'utiliser un développement de Taylor d'ordre 2 autour d'un point x_0 :

$$p(x) \stackrel{\text{def}}{=} f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2.$$

Nous verrons plus loin que cette approximation correspond à la méthode de Newton pour calculer une racine de $f'(x)$. Une fois que nous avons déterminé une approximation quadratique de f , il suffit de calculer son minimum pour obtenir un estimé (que l'on espère meilleur) du minimum de f . Utilisons la condition d'optimalité sur la fonction $p(x)$: $p'(x) = 0$, c'est-à-dire $f''(x_0)(x - x_0) + f'(x_0) = 0$. Pour définir le prochain point x_N , la méthode nommée méthode de Newton utilise donc la formule suivante : $x_N = x_0 - \frac{f'(x_0)}{f''(x_0)}$.

En fait, un polynôme quadratique p est défini par trois coefficients $p(x) = a + b(x - x_0) + \frac{c}{2}(x - x_0)^2$, et il suffit de fournir trois équations linéaires permettant de définir ces trois coefficients inconnus. Pour l'approximation de Taylor, les trois équations sont les suivantes : $p(x_0) = f(x_0)$, $p'(x_0) = f'(x_0)$, et $p''(x_0) = f''(x_0)$. Une autre possibilité consiste à choisir trois points, x_0, x_1, x_2 et utiliser les équations $p(x_0) = f(x_0)$, $p(x_1) = f(x_1)$, et $p(x_2) = f(x_2)$.

On observe deux des possibilités d'approximation quadratique exactes en $x_0 = 0$. Remarquons que l'approximation sécante possède la même pente que f en $x_1 = -0.2$. En général, le point auxiliaire est plus éloigné de la solution, et l'approximation de Taylor est meilleure que l'approximation sécante.

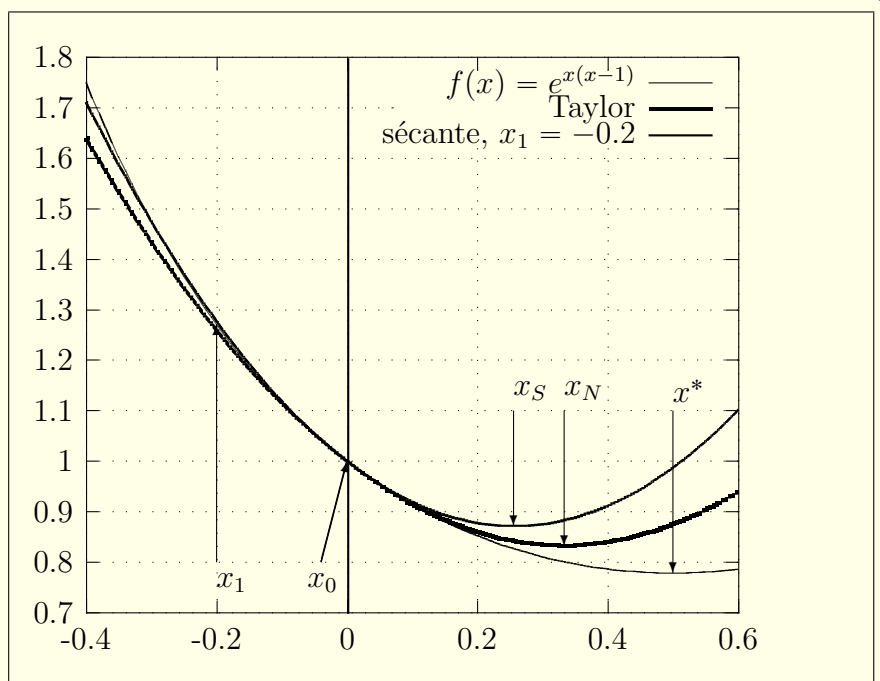


FIGURE 2.6 – Approximation quadratique

Exercice 2.5.1 [Approximation quadratique] Fournissez les coefficients du polynôme quadratique issu des conditions $p(x_0) = f(x_0)$, $p(x_1) = f(x_1)$, et $p(x_2) = f(x_2)$. Fournissez aussi une expression pour le minimum de la fonction quadratique.

D'autres possibilités sont envisageables, comme d'utiliser deux points x_0 et x_1 et les trois équations suivantes : $p(x_0) = f(x_0)$, $p'(x_0) = f'(x_0)$, $p'(x_1) = f'(x_1)$. Dans ce cas, les coefficients du polynôme sont : $a = f(x_0)$, $b = f'(x_0)$ et $c = \frac{f'(x_1) - f'(x_0)}{x_1 - x_0}$. Nous verrons plus loin que cette approximation correspond à la méthode de sécante pour calculer une racine de $f'(x)$. Comme toutes les approximations quadratiques de f , un estimé (que l'on espère meilleur) du minimum de f est donné par $x^+ = x_0 - \frac{b}{c}$, à condition bien sûr que $p''(x) = c > 0$.

2.5.2 Approximation linéaire de la fonction f'

Puisque nous recherchons une racine de $f'(x)$, nous pouvons envisager utiliser une approximation linéaire de $g \stackrel{\text{def}}{=} f'$ pour identifier un meilleur estimé. Si l'on approche g par un développement de Taylor d'ordre un, $g(x) \sim g(x_0) + g'(x_0)(x - x_0)$, on obtient la méthode de Newton. Le prochain estimé est une racine de l'approximation linéaire, c'est-à-dire

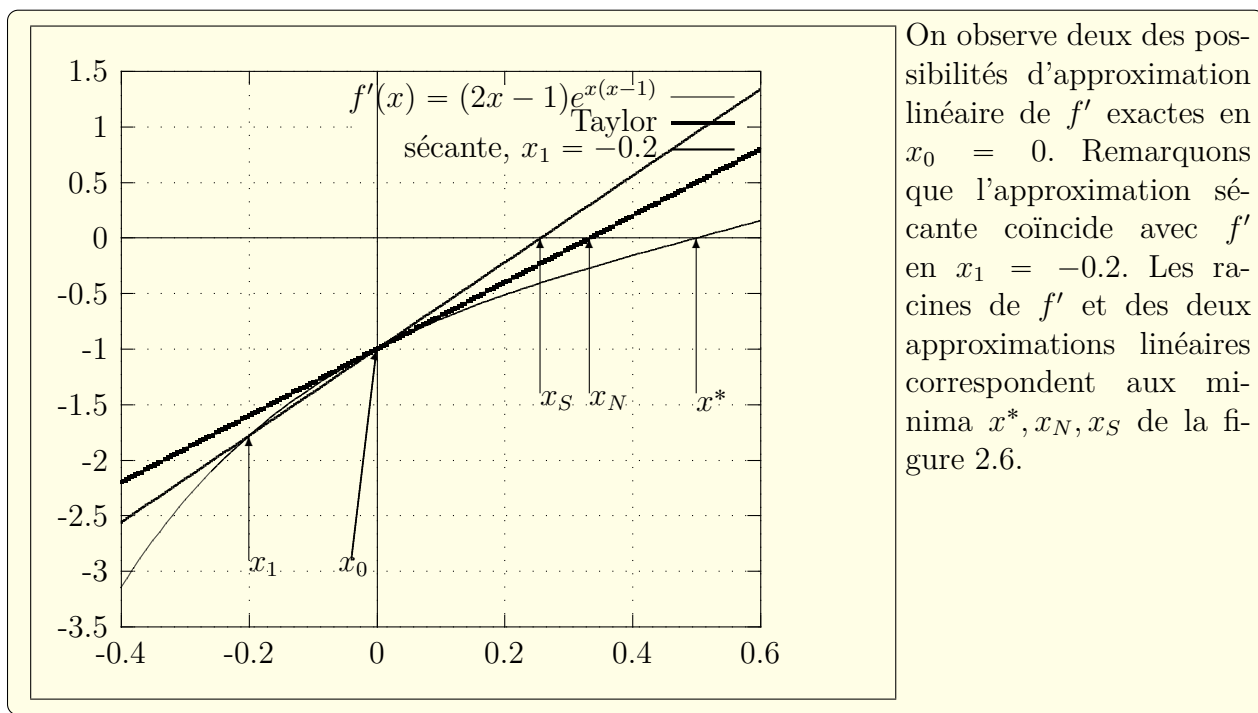


FIGURE 2.7 – Approximation linéaire

$x^+ = x_0 - \frac{g(x_0)}{g'(x_0)}$ qui correspond au même point que d'utiliser une approximation quadratique de f .

Si l'on utilise plutôt deux points pour calculer une approximation sécante de g , on obtient $g(x) \sim g(x_0) + \frac{g(x_1) - g(x_0)}{x_1 - x_0}(x - x_0)$ et le prochain point, toujours donné par une racine de l'approximation linéaire, est $x_S = x_0 - \frac{g(x_0)(x_1 - x_0)}{g(x_1) - g(x_0)}$. C'est la méthode de la sécante.

Exercice 2.5.2 [Applications d'algorithmes]

- Pour la fonction $f(x) = e^{x(x-1)}$, effectuez trois itérations des trois algorithmes (Newton, sécante, Approximation quadratique passant par trois points) à partir de l'intervalle initial $[-1, 1]$. Pour l'approximation utilisant trois points, choisir le troisième point en 0.
- Même question pour la fonction $f(x) = x^2(1 - \cos x)$ sur l'intervalle $[-1, 0.5]$.

Iter	x	$f'(x)$	$x - x^*$
0	-2.0000000000000000	-2017.1439674636756	-2.5000000000000000
1	-1.8148148148148149	-765.74138725328316	-2.3148148148148149
2	-1.6172499842815362	-291.78369169742325	-2.1172499842815364
3	-1.4047918990578092	-111.69278625297289	-1.9047918990578092
4	-1.1740888018727405	-42.990945144742859	-1.6740888018727405
5	-0.9206366236151708	-16.650840481988023	-1.4206366236151708
6	-0.6385637391018000	-6.4833811628303968	-1.1385637391018000
7	-0.3216494208351671	-2.5138450692553556	-0.8216494208351671
8	0.0279565647948271	-0.9187767726223022	-0.4720434352051729
9	0.3544833396867886	-0.2315076371696912	-0.1455166603132114
10	0.4940877267267001	-0.0092092880147647	-0.0059122732732999
11	0.4999995867021617	-0.0000006437533603	-0.0000004132978383
12	0.5000000000000000	0.0000000000000000	0.0000000000000000

TABLE 2.2 – Itérations de Newton sur la fonction $e^{x(x-1)}$, depuis un point de départ éloigné

Exercice 2.5.3 [*Sécante*] Suite de l'exercice 2.3.3 page 58 sur la fonction $h(\theta) = \frac{4\theta-7}{\theta^2+\theta-2}$.

- Peut-on utiliser la méthode de la sécante pour identifier un minimum local de $h(\theta)$ dans l'intervalle $[-1.0, 0.9]$? Est-ce que le prochain point est meilleur que le point milieu de l'intervalle?
- Est-ce que le prochain point de la méthode de la sécante à partir des points 0 et 0.3 est meilleur que le point milieu de l'intervalle $[0, 0.3]$?
- Dans la méthode de la sécante, est-il possible que $x_{k+1} = x_k$? Si oui, dans quelles circonstances (donnez un exemple)? Si non, pourquoi?

Exercice 2.5.4 [*Parabole*] Suite de l'exercice 2.4.3 page 64

- À partir du dernier point obtenu par bisection, effectuez une itération de Newton-Raphson et vérifiez que le point obtenu satisfait approximativement aux conditions d'optimalité.
- À partir du dernier intervalle obtenu par bisection, effectuez une itération de la méthode de sécante et vérifiez que le point obtenu satisfait approximativement aux conditions d'optimalité.

Exercice 2.5.5 [Ellipses...] Suite de l'exercice 2.4.4 page 64...

- a) À partir du dernier point obtenu par bisection, effectuez une itération de Newton-Raphson et vérifiez que le point obtenu satisfait approximativement aux conditions d'optimalité.
- b) À partir du dernier intervalle obtenu par bisection, effectuez une itération de la méthode de sécante et vérifiez que le point obtenu satisfait approximativement aux conditions d'optimalité.

Exercice 2.5.6 [Hyperboles...] Suite de l'exercice 2.4.5 page 64... À partir de chacune des extrémités de l'intervalle final trouvé à l'exercice 2.4.5, page 64, effectuez une itération de l'algorithme Newton-Raphson (toujours pour le point $P = \binom{1}{2}$).

2.5.3 Approximation cubique de la fonction f

La méthode de la sécante utilise $f'(x_{k+1})$ et $f'(x_k)$ dans ses calculs. Cependant, on dispose d'autre information, soit $f(x_{k-1})$ et $f(x_k)$, ce qui permet de développer une méthode plus performante compte tenu de l'information plus riche qu'elle utilise.

Il s'agit d'utiliser une approximation cubique de f en se basant sur les valeurs de f et de sa dérivée f' en deux points x_k et x_{k-1} ; notons $f(x_k) = f_k$, $f(x_{k-1}) = f_{k-1}$, $f'(x_k) = f'_k$, $f'(x_{k-1}) = f'_{k-1}$.

Cherchons donc un polynôme cubique $p(t)$ tel que $p(0) = f_k$, $p'(0) = f'_k$, $p(\alpha) = f_{k-1}$ et enfin $p'(\alpha) = f'_{k-1}$, où $\alpha = (x_{k-1} - x_k)$.

Considérons $p(t) = f_k + f'_k t + At^2 + Bt^3$, qui satisfait d'emblée aux deux premières conditions. $p'(t) = f'_k + 2At + 3Bt^2$, d'où on tire les deux équations nous permettant de déterminer A et B :

$$\alpha^2 A + \alpha^3 B = f_{k-1} - f_k - f'_k \alpha \quad (2.1)$$

$$2\alpha A + 3\alpha^2 B = f'_{k-1} - f'_k. \quad (2.2)$$

Il est plus facile de résoudre ces équations pour obtenir αA et $\alpha^2 B$:

$$\alpha A = -(f'_k + z) \quad (2.3)$$

$$\alpha^2 B = \frac{1}{3}(f'_k + f'_{k-1} + 2z) \quad (2.4)$$

où $z = \frac{3}{\alpha}(f_k - f_{k-1}) + f'_k + f'_{k-1}$. On exprime maintenant

$$p'(t) = f'_k - 2(f'_k + z)\frac{t}{\alpha} + (f'_k + f'_{k-1} + 2z)\left(\frac{t}{\alpha}\right)^2.$$

Les racines $\frac{t}{\alpha}$ sont

$$\frac{t}{\alpha} = \frac{f'_k + z \pm \sqrt{(f'_k + z)^2 - f'_k(f'_k + f'_{k-1} + 2z)}}{f'_k + f'_{k-1} + 2z} \quad (2.5)$$

qui s'écrit, dénotant $w = \sqrt{z^2 - f'_k f'_{k-1}}$,

$$\frac{t}{\alpha} = \frac{f'_k + z \pm w}{f'_k + f'_{k-1} + 2z}.$$

Pour déterminer laquelle des deux racines on doit choisir, quelle branche du \pm dans l'équation 2.5, puisque nous voulons un minimum local du polynôme cubique, nous choisissons celui pour lequel $p''(t) > 0$, et on constate que p'' des deux racines vaut $\pm 2w/\alpha$, et donc on prend la branche de signe opposé à α pour obtenir t^*/α et finalement, $x_{k+1} = x_k + t^*$.

Les méthodes de Newton et de la sécante peuvent produire des suites ne s'approchant pas d'un minimum local, et même échouer si leur dénominateur est négatif. Cette méthode basée sur le polynôme d'interpolation cubique de f peut échouer dans des circonstances additionnelles : lorsque le discriminant $z^2 - f'_k f'_{k-1}$ est négatif, ou encore lorsque le terme en t^2 dans l'expression de p' , $(f'_k + f'_{k-1} + 2z) = 0$.

Exercice 2.5.7 [*Échec de l'approximation cubique*]

- Vérifiez que le polynôme d'interpolation cubique de la fonction de la figure 2.6 en utilisant les points de la figure ($x_0 = 0$ et $x_1 = -0.2$) ne possède pas de minimum local. Illustrez graphiquement, et interprétez cet échec de la méthode.
- Vérifiez que la méthode d'interpolation cubique échoue pour une fonction f quadratique. Proposez une technique plus robuste.

*2.5.4 Utilisation de l'approximation cubique de f

Les conditions de sortie de l'algorithme 2.1 permettent de garantir qu'une fonction cubique interpolant x_k, f_k, g_k et $x_{k-1}, f_{k-1}, g_{k-1}$ possède son minimum local dans l'intervalle d'extrémités x_k et x_{k-1} . Par conséquent, on peut utiliser l'interpolation cubique $p(t)$ de la fonction f pour obtenir le point \bar{x} qui minimise p . Alors, sans perte de généralité, supposons que $p'(0) < 0$ et que $x_k < x_{k-1}$. Dans ce cas, on a $p'(\alpha) > 0$ ou $p(\alpha) > p(0)$.

- $f'(\bar{x}) > 0$: on conserve l'intervalle $[x_k, \bar{x}]$;

- $f'(\bar{x}) < 0$: il faut considérer deux cas :
 - $f(\bar{x}) \geq f(x_k)$: on choisit $[x_k, \bar{x}]$;
 - $f(\bar{x}) < f(x_k)$: on choisit $[\bar{x}, x_{k-1}]$;
- $f'(\bar{x}) = 0$: deux possibilités :
 - $f(\bar{x}) \geq f(x_k)$: on choisit $[x_k, \bar{x}]$;
 - $f(\bar{x}) < f(x_k)$: l'algorithme termine.

Ainsi, les conditions pour que le nouvel intervalle contienne un minimum local sont préservées. Le cas où $f'(\bar{x}) = 0$ et $f(\bar{x}) < f(x_k)$ est problématique car l'algorithme d'interpolation cubique peut s'y terminer, même s'il s'agit d'un maximum local pour lequel $f''(\bar{x}) > 0$.

***Exercice 2.5.8** [*Panne de la réduction d'intervalle par polynome cubique*] Fournissez un exemple pour lequel l'algorithme termine en un point qui satisfait $f'(\bar{x}) = 0$ et $f(\bar{x}) < f(x_k)$, mais pour lequel \bar{x} est un maximum local de f .

Il ne faut pas s'illusionner et espérer que l'ordre de convergence de cette stratégie de réduction d'intervalle soit quadratique : en général, l'algorithme remplace toujours la même extrémité de l'intervalle, et la convergence asymptotique n'est que linéaire. Cependant, en combinant cette stratégie de réduction d'intervalle à la véritable interpolation cubique (conserver une paire de points consécutifs en plus des extrémités de l'intervalle), on atteint l'ordre quadratique sans utiliser la dérivée de second ordre.

2.6 Analyse asymptotique

Nous nous intéresserons tout au long de ce texte au comportement asymptotique des algorithmes itératifs. En examinant la table 2.2, on peut observer que les trois dernières itérations réduisent l'erreur (distance à la solution, ou encore $|f'(x)|$) très rapidement. Le comportement de l'algorithme est donc bien meilleur proche de la solution que dans les premières itérations, plus loin de la solution. Ce comportement proche de la solution est nommé "convergence asymptotique". Cette section présente des outils pour faciliter l'analyse asymptotique.

Nous présentons d'abord les symboles de Landau qui permettent d'exprimer le comportement d'une fonction pour des valeurs soit très petites proches de 0, soit très grandes. Les algorithmes itératifs que nous étudions réduisent progressivement l'erreur, de sorte que l'erreur à une itération donnée peut être vue comme une fonction de l'erreur à l'itération précédente. À quel point cette erreur sera plus petite d'une itération à l'autre ?

2.6.1 Notation “grand O” et “petit o”

Soient deux fonctions, f et g , de \mathbb{R} dans \mathbb{R} , d'un même paramètre x telles que $f(\bar{x}) = g(\bar{x}) = 0$.

Définition 2.6.1 Nous notons $f \sim \mathcal{O}(g)$ au voisinage de \bar{x} si et seulement si il existe $\gamma \geq 0$ tel que

$$\limsup_{x \rightarrow \bar{x}} \left| \frac{f(x)}{g(x)} \right| = \gamma < \infty.$$

Une autre manière d'exprimer cette notation est qu'il existe un x_0 et une valeur $M > 0$ tels que $\forall x : |x - \bar{x}| < |x_0 - \bar{x}|$ on a $|f(x)| \leq M|g(x)|$.

En prose, f est “grand O” de g lorsque f s'approche de zéro au moins aussi vite que g s'approche de zéro quand x tend vers \bar{x} . On laisse à montrer, en exercices, les propriétés suivantes.

Exercice 2.6.1 [Notation \mathcal{O}]

- a) Démontrez les propriétés de la notation “grand O”
- i) $f \sim \mathcal{O}(g)$ entraîne que $\alpha f \sim \mathcal{O}(g)$, où α est une constante finie ;
 - ii) $f + g \sim \mathcal{O}(\max(|f|, |g|))$;
- b) que peut-on dire de $\frac{1}{f} \sim \mathcal{O}(\cdot)$?

Par exemple, deux fonctions linéaires $f(x) \stackrel{\text{def}}{=} a_f x$ et $g(x) \stackrel{\text{def}}{=} a_g x$ sont mutuellement “grand O” l'une de l'autre proche de $x = 0$ si $a_f \neq 0$ et $a_g \neq 0$ car, selon la définition, $f \sim \mathcal{O}(g)$ et inversement, $g \sim \mathcal{O}(f)$ puisque $\frac{f}{g} = \frac{a_f}{a_g}$. Ces deux fonctions s'approchent de zéro *linéairement*.

Dans un second exemple, supposons f comme ci-haut, et $g \stackrel{\text{def}}{=} a_g x^2$ près de $x = 0$. Maintenant, selon la définition, $g \sim \mathcal{O}(f)$ car $\frac{g}{f} = \frac{a_g}{a_f} x \rightarrow 0$ mais pas l'inverse car $\frac{f}{g} = \frac{a_f}{a_g x} \rightarrow \infty$. g s'approche de zéro quadratiquement, ce qui est plus vite que f qui s'approche de zéro linéairement.

Exercice 2.6.2 [Ordre de convergence à zéro] Soit g une fonction réelle de classe $\mathcal{C}^1(\mathbb{R})$ avec $g(\bar{x}) = 0$ et $g'(\bar{x}) \neq 0$. Montrez qu'il existe un voisinage de \bar{x} (défini comme $\{x : |x - \bar{x}| \leq \epsilon\}$) et des constantes appropriées $K_1, K_2, K_3, K_4 > 0$ telles que

$$\left. \begin{array}{l} K_1|x - \bar{x}| \leq |g(x)| \leq K_2|x - \bar{x}| \\ K_3|g(x)| \leq |x - \bar{x}| \leq K_4|g(x)| \end{array} \right\} \forall x : |x - \bar{x}| \leq \epsilon$$

Ceci démontre, selon la notation \mathcal{O} , que $g(x) \sim \mathcal{O}(x - \bar{x})$ et que $(x - \bar{x}) \sim \mathcal{O}(g(x))$ si $g(\bar{x}) = 0$ et $g'(\bar{x}) \neq 0$. g pourrait être la dérivée d'une fonction à minimiser $f' = g$ et \bar{x} un minimum local de f et dans ce cas, $g'(\bar{x}) > 0$.

Similairement, lorsque f est négligeable par rapport à g , on peut utiliser la notation "petit o ".

Définition 2.6.2 Nous notons $f \sim o(g)$ si et seulement si $\limsup_{x \rightarrow \bar{x}} \left| \frac{f(x)}{g(x)} \right| = 0$.

L'utilisation de cette notation grand et petit \mathcal{O} permet de dissimuler les constantes sous-jacentes pour ne retenir que l'essentiel du comportement asymptotique des fonctions que nous comparons. Il faut cependant faire preuve d'une grande prudence en utilisant ces notations raccourcies. Voici quelques exercices en guise de mise en garde.

Exercice 2.6.3 [Propriétés de \mathcal{O}] Soient f_1, f_2 et f_3 trois fonctions qui s'annulent en \bar{x} .

- Montrez que $f_1 \sim \mathcal{O}(f_3)$ et $f_2 \sim \mathcal{O}(f_3)$ implique $(f_1 + f_2) \sim \mathcal{O}(f_3)$.
- Montrez que $f_1 \sim \mathcal{O}(f_2)$ et $f_1 \sim \mathcal{O}(f_3)$ n'implique pas que $f_1 \sim \mathcal{O}(f_2 + f_3)$.

Nous profitons de l'introduction de la notation grand \mathcal{O} pour donner une forme du théorème de Taylor que nous utiliserons fréquemment par la suite.

Théorème 2.6.1 Soit une fonction f de classe \mathcal{C}^p au voisinage d'un point x_0 ; on peut approcher la valeur de la fonction en un point voisin $x_0 + d$ de x_0 par un polynôme

$$f(x_0 + d) \approx \mathcal{P}(d) = f(x_0) + f'(x_0)d + \frac{f''(x_0)}{2}d^2 + \dots + \frac{f^{(p-1)}(x_0)}{(p-1)!}d^{(p-1)}$$

De plus, on peut estimer l'erreur commise par un reste :

$$f(x_0 + d) - \mathcal{P}(d) \sim \mathcal{O}(|d|^p).$$

Si f est seulement de classe \mathcal{C}^{p-1} , mais toutefois $f^{(p-1)}$ est une fonction Lipschitz, on a la borne :

$$f(x_0 + d) - \mathcal{P}(d) \sim o(|d|^{p-1}).$$

2.6.2 Vitesse de convergence d'algorithmes itératifs

Nous introduisons maintenant le vocabulaire permettant de comparer l'efficacité asymptotique de divers algorithmes.

Lorsqu'on parle de vitesse de convergence, il faut supposer que toute la suite $\{x_k\}$ converge, c'est-à-dire que la suite ne possède qu'un seul point d'accumulation, \bar{x} . Dénotons l'erreur $\epsilon_k \stackrel{\text{def}}{=} x_k - \bar{x}$ ainsi que sa valeur absolue $e_k \stackrel{\text{def}}{=} |\epsilon_k|$.

Définition 2.6.3 *La vitesse de convergence de la suite $\{x_k\}$ vers le point \bar{x} , telle que $\forall k, x_k \neq \bar{x}$ s'exprime à l'aide des scalaires p et γ dans l'expression suivante*

$$\limsup_{k \rightarrow \infty} \frac{|\epsilon_{k+1}|}{|\epsilon_k|^p} = \gamma < \infty.$$

L'ordre de convergence de la suite est donné par la plus grande valeur que p puisse prendre pour que la limite ci-haut demeure finie. Lorsque $p = 1$, γ est nommée le taux de convergence.

Les cas importants (fréquents) sont le cas $p = 1$, dite *convergence linéaire*, et le cas $p = 2$, dite *convergence quadratique*. Un cas particulier de convergence linéaire est également important, soit le cas où $p = 1$ et $\gamma = 0$; on nomme ce type de comportement *convergence superlinéaire*.

Évidemment, puisque l'on s'intéresse au comportement limite de la suite $\{x_k\}$ qui converge vers \bar{x} , ce type d'analyse ne qualifie que le comportement local d'un algorithme, et n'est valable que lorsque x_k devient proche de \bar{x} .

Exemple 2.6.1 Dans l'algorithme de bisection, on a la garantie que l'intervalle contient un point stationnaire. La largeur de l'intervalle borne donc l'erreur que l'on commet. Puisque l'intervalle est réduit de moitié à chaque itération, nommant δ_k cette largeur à l'itération k , on a (puisque δ_k converge vers zéro)

$$\limsup_{k \rightarrow \infty} \frac{|\delta_{k+1}|}{|\delta_k|^1} = \frac{1}{2} < \infty,$$

et donc l'ordre de convergence de l'algorithme de bisection est un, et on parle de convergence linéaire de taux $1/2$. Remarquons que pour cet algorithme, nous avons utilisé la mesure de la largeur de l'intervalle δ_k plutôt que la distance à la solution $\epsilon_k = x_k - x^*$. La distance à la solution ne décroît pas de manière monotone, et il est possible que la mesure ϵ_k décroisse à un taux moins bon que $\frac{1}{2}$, taux de la décroissance de δ_k . \square

En observant le tableau 1.1, on constate que l'algorithme identifie une décimale additionnelle de la solution environ à toutes les trois itérations. C'est typique d'un algorithme qui converge linéairement. Par opposition, sur le tableau 1.2, bien que les premières itérations ne soient pas très efficaces, on constate que chacune des trois dernières itérations *double* le nombre de décimales exactes de la solution, ce qui est typique des algorithmes qui convergent quadratiquement.

En utilisant la notation grand O, on peut résumer

- convergence quadratique : $|\epsilon_{k+1}| \sim \mathcal{O}(|\epsilon_k|^2)$;

- convergence superlinéaire : $|\epsilon_{k+1}| \sim o(|\epsilon_k|)$;
 - convergence linéaire : $|\epsilon_{k+1}| \sim \mathcal{O}(|\epsilon_k|)$ avec une constante cachée plus petite que un.
- Habituellement, par abus de langage, on écrira simplement par exemple $\epsilon_{k+1} \sim \mathcal{O}(\epsilon_k^2)$ en omettant les valeurs absolues.

Exercice 2.6.4 [Une propriété de la convergence superlinéaire] Montrez :

a) toute suite $\epsilon_k = x_k - \bar{x}$ convergeant à 0 satisfait :

$$\left| 1 - \frac{|x_{k+1} - x_k|}{|\epsilon_k|} \right| \leq \frac{|\epsilon_{k+1}|}{|\epsilon_k|}$$

b) si la vitesse de convergence est superlinéaire, alors

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - x_k|}{|\epsilon_k|} = 1.$$

2.7 Convergence locale quadratique de l'itération de Newton

L'analyse de convergence locale peut très bien être réalisée dans le contexte de la recherche d'une racine de l'équation $g(x) = 0$. Pour les problèmes d'optimisation, $g = f'$, mais cette propriété n'est pas utile pour l'analyse locale de la convergence comme elle l'a été jusqu'ici pour l'analyse de la convergence globale des algorithmes.

Il est facile de se convaincre que l'ordre de convergence de l'itération de Newton, *localement*, est quadratique. Il est souvent plus concis d'utiliser la notation "grand O", permettant d'escamoter des détails de preuves fastidieux. Cependant, l'utilisation de cette notation nécessite une grande prudence. Nous analysons donc ici la vitesse de convergence locale de la méthode de Newton en utilisant les techniques d'analyse classiques, ainsi que, pour fins de comparaison, les formulations de ces analyses avec la notation "grand O".

2.7.1 Analyse de l'itération de Newton

Plaçons-nous dans le contexte d'identifier une racine de la dérivée de f , $g(x) \stackrel{\text{def}}{=} f'(x)$. On veut donc résoudre l'équation $g(x) = 0$. Dans ce contexte de solution d'une équation non-linéaire, rappelons que l'itération de Newton s'écrit :

$$x_{k+1} \leftarrow x_k - \frac{g(x_k)}{g'(x_k)}.$$

Nous nous intéressons ici à la convergence locale de l'algorithme, c'est-à-dire à la convergence de l'algorithme lorsque x_k est devenu très proche de x^* , une racine de l'équation $g(x) = 0$ pour laquelle $g'(x^*) \neq 0$.

Cette itération de Newton peut se reformuler ainsi :

$$\begin{aligned} x_{k+1} &\leftarrow x_k + d_k; \\ d_k &\leftarrow -\frac{g(x_k)}{g'(x_k)}. \end{aligned}$$

On voit donc que d_k satisfait à la relation

$$g'(x_k)d_k + g(x_k) = 0.$$

Introduisons la notion d'erreur à l'itération k ; $\epsilon_k \stackrel{\text{def}}{=} x^* - x_k$. Selon la notation “grand O”, nous allons montrer que $|\epsilon_{k+1}| \sim \mathcal{O}(|\epsilon_k|^2)$, c'est à dire que lorsque $\epsilon_k \rightarrow 0$, la quantité $|\epsilon_{k+1}|$ (vue comme fonction de ϵ_k) s'approche de zéro au moins aussi vite que la quantité $|\epsilon_k|^2$. Selon la définition 2.6.3,

$$\limsup_{\epsilon_k \rightarrow 0} \frac{|\epsilon_{k+1}|}{|\epsilon_k|^2} = \gamma < \infty,$$

ce qui peut se récrire, pour un certain $\bar{\gamma} > \gamma$ et k assez grand,

$$|\epsilon_{k+1}| \leq \bar{\gamma}|\epsilon_k|^2.$$

Théorème 2.7.1 *Soit x^* une racine isolée de la fonction g telle que $g'(x^*) \neq 0$, avec la fonction g' Lipschitz, c'est-à-dire qu'il existe une constante L , dite constante de Lipschitz telle que $\forall x, y, |g'(x) - g'(y)| \leq L|x - y|$. Alors, il existe un voisinage de x^* tel que si la méthode de Newton est initialisée dans ce voisinage, elle produit une suite convergente vers x^* et la vitesse de convergence asymptotique est quadratique.*

2.7.2 Preuve concise de la convergence quadratique

Nous allons donc donner une première preuve de l'ordre quadratique de convergence pour l'itération de Newton en utilisant la notation “grand O”, et ensuite, fournirons les détails d'une preuve qui manipule explicitement les constantes mises en jeu par le processus, mais cachées dans la première preuve.

D'abord, exprimons $g(x^*)$ en utilisant une série de Taylor autour du point x_k :

$$\begin{aligned} 0 = g(x^*) &= g(x_k) + g'(x_k)\epsilon_k + \mathcal{O}(|\epsilon_k|^2) \\ &= (g(x_k) + g'(x_k)d_k) + g'(x_k)(\epsilon_k - d_k) + \mathcal{O}(|\epsilon_k|^2) \\ &= 0 + g'(x_k)(\epsilon_{k+1}) + \mathcal{O}(|\epsilon_k|^2). \end{aligned}$$

De cette dernière équation, on tire que ϵ_{k+1} satisfait à la relation

$$g'(x_k)(\epsilon_{k+1}) \sim \mathcal{O}(|\epsilon_k|^2).$$

Utilisons maintenant les faits que

- $g'(x^*) > 0$, et
 - g' est une fonction continue, donc $g'(x_k) \neq 0$,
- pour conclure que $\epsilon_{k+1} \sim \mathcal{O}(|\epsilon_k|^2)$. On voit donc que la convergence quadratique de l'itération de Newton est assurée.

2.7.3 Preuve détaillée de la convergence quadratique

Maintenant, examinons une preuve détaillée du même résultat, illustrant la puissance de la notation “grand O”.

D'abord, exprimons $g(x^*)$ en utilisant une série de Taylor autour du point x_k :

$$\begin{aligned} 0 = g(x^*) &= g(x_k) + g'(x_k)\epsilon_k + R_2(\epsilon_k) \\ &= (g(x_k) + g'(x_k)d_k) + g'(x_k)(\epsilon_k - d_k) + R_2(\epsilon_k) \\ &= 0 + g'(x_k)(\epsilon_{k+1}) + R_2(\epsilon_k), \end{aligned}$$

où R_2 est le reste du développement de Taylor d'ordre un ; la notation R_2 souligne le fait que ce reste est, si g est \mathcal{C}^2 , quadratiquement petit en ϵ , propriété explicitement utilisée dans la preuve abrégée.

Comme dans la preuve abrégée, de cette dernière équation, on tire que ϵ_{k+1} est solution du système linéaire

$$g'(x_k)(\epsilon_{k+1}) = R_2(\epsilon_k).$$

Gardons en tête la propriété de Lipschitz de g' et écrivons

$$g'(x_k)(\epsilon_{k+1}) = g'(x^*)(\epsilon_{k+1}) + (g'(x_k) - g'(x^*))(\epsilon_{k+1}),$$

ce qui se transforme en prenant les normes de chaque côté en

$$\begin{aligned} |g'(x_k)(\epsilon_{k+1})| &= |g'(x^*)(\epsilon_{k+1}) + (g'(x_k) - g'(x^*))(\epsilon_{k+1})| \\ &\geq |g'(x^*)(\epsilon_{k+1})| - |(g'(x_k) - g'(x^*))(\epsilon_{k+1})| \\ &\geq |g'(x^*)(\epsilon_{k+1})| - L \cdot |\epsilon_k| |\epsilon_{k+1}|. \end{aligned}$$

Maintenant, $|g'(x^*)| = K_1 > 0$, mais, plus important, $g'(x^*)$ est bornée, d'où l'on tire que $K_1 < \infty$. Les précédents développements nous permettent maintenant d'écrire :

$$(K_1 - L|\epsilon_k|)|\epsilon_{k+1}| \leq |g'(x_k)(\epsilon_{k+1})| = |R_2(\epsilon_k)|.$$

Cependant, le théorème de Taylor nous assure qu'il existe un $K_3 > 0$ tel que $|R_2(\epsilon_k)| \leq K_3|\epsilon_k|^2$, et donc, on a

$$(K_1 - L|\epsilon_k|)|\epsilon_{k+1}| \leq K_3|\epsilon_k|^2,$$

et donc

$$|\epsilon_{k+1}| \leq \frac{K_3}{(K_1 - L|\epsilon_k|)}|\epsilon_k|^2,$$

ce qui complète la preuve puisque $K_1 > 0$ est une constante indépendante de ϵ_k alors que $L|\epsilon_k|$ converge vers zéro lorsque $|\epsilon_k|$ s'approche de zéro. Notons que nous pouvons maintenant spécifier rigoureusement ce que nous entendons par *localement*. En effet, nous avons l'assurance que $|\epsilon_{k+1}| < |\epsilon_k|$ dès que

$$\frac{K_3}{(K_1 - L|\epsilon_k|)}|\epsilon_k| < 1,$$

qui est équivalent à

$$|\epsilon_k| < \left(\frac{K_1}{K_3 + L} \right).$$

Commentaires

La preuve concise n'est pas beaucoup plus brève pour ce résultat, puisque on ne nécessite que trois constantes pour exprimer le raisonnement.

Exercice 2.7.1 [Méthode de Newton]

- Fournissez un exemple (une fonction et un point de départ) pour lequel l'algorithme de Newton diverge.
- Fournissez un exemple (une fonction et un point de départ) pour lequel l'algorithme de Newton oscille entre deux points.
- Fournissez un exemple pour lequel l'algorithme de Newton converge vers une racine, mais pour laquelle $g'(x^*) = 0$; vérifiez que dans ce cas, la vitesse de convergence asymptotique est seulement linéaire.
- Pour résoudre une équation $g(x) = 0$, analysez la vitesse de convergence de l'algorithme de Newton pour la famille de fonctions $g(x) = x^p + x^{(p+1)}$ proche de $x^* = 0$, une racine de $g(x)$; distinguez les cas $p = 1$ et $p \geq 2$.

*2.8 Analyse des approximations polynômiales

Nous présentons dans cette section les analyses concises de la méthode de la sécante ainsi que l'approximation cubique de la fonction f .

2.8.1 Ordre de convergence de la méthode de la sécante

Rappelons que la méthode de la sécante est décrite par la formule

$$x_{k+1} = x_k - g(x_k) \frac{x_k - x_{k-1}}{g(x_k) - g(x_{k-1})}.$$

Par le théorème de la moyenne, on sait que $\frac{x_k - x_{k-1}}{g(x_k) - g(x_{k-1})} = \frac{1}{g'(\xi)}$ pour un certain ξ entre x_k et x_{k-1} . Nous allons comparer x_{k+1} au point qui aurait été obtenu par la formule de Newton :

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)} + \frac{g(x_k)}{g'(x_k)} - \frac{g(x_k)}{g'(\xi)} \quad (2.6)$$

$$= x_N + g(x_k) \frac{g'(\xi) - g'(x_k)}{g'(x_k)g'(\xi)} \quad (2.7)$$

Maintenant, dénotons $\epsilon_k = x_k - x^*$, et utilisons les faits que $x_N - x^* \sim \mathcal{O}(\epsilon_k^2)$, que $g(x_k) \sim \mathcal{O}(\epsilon_k)$, que $\xi - x_k \sim \mathcal{O}(\epsilon_{k-1})$, et donc, puisque g' est supposée continue, $g'(\xi) - g'(x_k) \sim \mathcal{O}(\epsilon_{k-1})$ pour écrire

$$\epsilon_{k+1} \sim \mathcal{O}(\epsilon_k^2 + \epsilon_k \epsilon_{k-1}).$$

Puisque ϵ_k tend vers 0, $\epsilon_k \epsilon_{k-1}$ doit dominer ϵ_k^2 , et si on suppose que asymptotiquement $\epsilon_k = \epsilon_{k-1}^a$, $\epsilon_{k+1} = \epsilon_k^a$, on obtient la relation

$$\epsilon_{k-1}^{a^2} \sim \mathcal{O}(\epsilon_{k-1}^{a+1})$$

et donc a doit satisfaire à la relation $a^2 - 1 - a = 0$, dont la racine plus grande que l'unité est le nombre d'or $a = \tau \approx 1.618$.

2.8.2 Ordre de convergence de l'approximation cubique de f

Un raisonnement intuitif pour obtenir l'ordre de convergence de la méthode de sécante consiste à réaliser que $\epsilon_{k+1} = \Phi(\epsilon_k, \epsilon_{k-1})$, c'est-à-dire que l'erreur à l'étape $k+1$ est fonction des erreurs aux étapes k et $k-1$. Maintenant, si x_k tend vers x_{k-1} , on retrouve la méthode de Newton, donc Φ est une fonction d'ordre 2. Puisque les rôles de x_k et x_{k-1} sont symétriques, on arrive à la conclusion que $\epsilon_{k+1} \sim \mathcal{O}(\epsilon_k \epsilon_{k-1})$.

Similairement, pour l'approximation cubique de f , on voit que si x_{k-1} tend vers x_k , on a un polynôme de Taylor d'ordre trois, et la méthode résultante aurait un ordre de convergence

cubique. La symétrie des rôles de x_k et x_{k-1} encore une fois permet de conclure que l'erreur obéit à une relation de la forme

$$\epsilon_{k+1} \sim \mathcal{O}(\epsilon_k^2 \epsilon_{k-1} + \epsilon_k \epsilon_{k-1}^2).$$

Encore une fois, on ne conserve que le terme dominant, on suppose que, asymptotiquement, $\epsilon_k = \epsilon_{k-1}^a$, et on constate que a doit satisfaire à la relation $a^2 - a - 2 = 0$, et donc que l'ordre de convergence de la méthode basée sur l'approximation cubique de la fonction f est quadratique.

2.9 Algorithme combiné

Nous avons étudié deux familles d'algorithmes : des algorithmes qui possèdent la propriété de convergence globale, et des algorithmes qui possèdent la propriété de convergence locale rapide. Nous avons évoqué que l'on devrait utiliser les algorithmes de type Newton dès que l'intervalle contenant un minimum local est assez petit. Nous allons préciser ce point.

Dans un premier temps, nous allons combiner l'itération de Newton avec la méthode de bisection pour produire un algorithme robuste de réduction d'intervalle qui possède un ordre de convergence quadratique. Dans les chapitres subséquents, nous utiliserons cette stratégie de combiner une méthode de type Newton à une méthode robuste.

Ensuite, nous verrons que la méthode basée sur l'approximation cubique de f peut être utilisée pour réduire l'intervalle, et possède également un ordre de convergence quadratique.

2.9.1 Algorithme de Newton modifié

L'idée est la suivante : nous allons retenir l'itération de Newton seulement lorsqu'elle appartient à l'intervalle spécifié par l'algorithme de réduction d'intervalle. Il est possible que l'intervalle ne soit pas réduit d'autant que par la méthode de réduction d'intervalle, mais la vitesse de convergence asymptotique de la méthode de Newton nous assure que ses estimés seront bien meilleurs que ceux de la méthode de réduction d'intervalle. Bien que l'idée soit simple, l'algorithme final se complique car les extrémités x_0 et x_1 des intervalles générés ne sont pas nécessairement $x_0 < x_1$ et il faut traiter les deux possibilités.

Pour être vraiment intéressant, l'algorithme combiné devrait posséder les propriétés de convergence globale de la méthode de bisection, et la vitesse de convergence asymptotique de la méthode de Newton.

Théorème 2.9.1 *L'algorithme combiné est assuré d'approcher soit un minimum local de f ou encore un point d'inflexion. Si f de classe \mathcal{C}^3 et si le minimum local cerné par l'algorithme combiné est tel que $f''(x^*) > 0$, alors, en posant $\epsilon = 0$, à partir d'un certain point, x^+ sera toujours choisi égal à x_N .*

Bissection-Newton
$\{ \text{Donnés : } x_0 \text{ et } x_1 \text{ tels que } f'(x_0) * f'(x_1) < 0 \text{ et } (f'(x_1) - f'(x_0))(x_1 - x_0) > 0; \}$ $\{ \text{un critère d'arrêt, } \epsilon > 0; \}$ $\{ \text{un seuil de réduction d'intervalle, } \frac{1}{2} < \gamma < 1. \}$
répéter
$d_N \leftarrow -f'(x_1)/f''(x_1)$ si $((x_0 - x_1)d_N > 0) \wedge (d_N/(x_0 - x_1) < \gamma)$ alors $x^+ \leftarrow x_1 + d_N$ sinon $x^+ \leftarrow (x_0 + x_1)/2$ si $(x_1 > x_0)$ alors si $(f'(x^+) \leq 0)$ alors $x_0 \leftarrow x_1; \quad x_1 \leftarrow x^+$ sinon $x_1 \leftarrow x^+$ sinon si $(f'(x^+) \leq 0)$ alors $x_1 \leftarrow x^+$ sinon $x_0 \leftarrow x_1; \quad x_1 \leftarrow x^+$
jusqu'à $(x_1 - x_0 < \epsilon \vee (f'(x_1) < \epsilon \wedge f''(x_1) \geq 0))$

Algorithme 2.4: Réduit intervalle accéléré—bissection—Newton.

Preuve

- Montrons d'abord que tout point limite de l'algorithme satisfait aux conditions nécessaires d'ordre deux. Si chacune des deux extrémités x_0 et x_1 sont mises à jour infiniment souvent, le raisonnement du théorème 2.4.2 permet de conclure.

Si x_0 cesse d'être mise-à-jour, l'intervalle limite n'est plus réduit à zéro. Deux cas peuvent se produire :

- $f'(x_0) = 0$ et donc $f'(x_1)$ sont toutes du même signe et $f''(\bar{x}) \geq 0$;
- $f'(x_0) \neq 0$; sans perte de généralité, supposons $f'(x_0) < 0$; dans ce cas, les itérations $f'(x_1) \searrow 0$ et encore $f''(\bar{x}) \geq 0$.
- Supposons maintenant que le point limite satisfasse aux conditions suffisantes d'ordre deux, $f''(\bar{x}) > 0$. On aura que x_1 est plus proche de \bar{x} que x_0 infiniment souvent. Lorsque $|x_1 - x_0|$ devient suffisamment petit, la propriété de convergence quadratique de l'itération de Newton nous assure que $|x_N - \bar{x}| \sim \mathcal{O}(x_1 - \bar{x})^2$ et donc $(x_1 - x_N)/(x_1 - x_0) < \frac{1}{2} < \gamma$ et le point x_N est retenu, x_1 devient x_N qui demeure plus proche que x_0 de \bar{x} , et la suite des itérations consiste en des pas de Newton.

□

Sur le tableau 1.3, on peut observer la réduction de l'intervalle initial par des pas de bissection, ainsi que la convergence quadratique des dernières itérations. En fait, même en $x = 0$, $f''(x) < 0$ et donc l'itération de Newton n'a pas de sens. Cependant, dès que x atteint 0.5, la combinaison Newton/bissection change automatiquement au profit de l'itération de Newton, et seulement trois autres itérations sont requises pour identifier le minimum local

On observe que la fonction possède un minimum isolé en $x = 0.6$; cependant, l'algorithme de Newton ne peut pas être utilisé loin de $x = 0.6$ car la fonction est *courbée vers le bas*, condition due à une dérivée seconde négative.

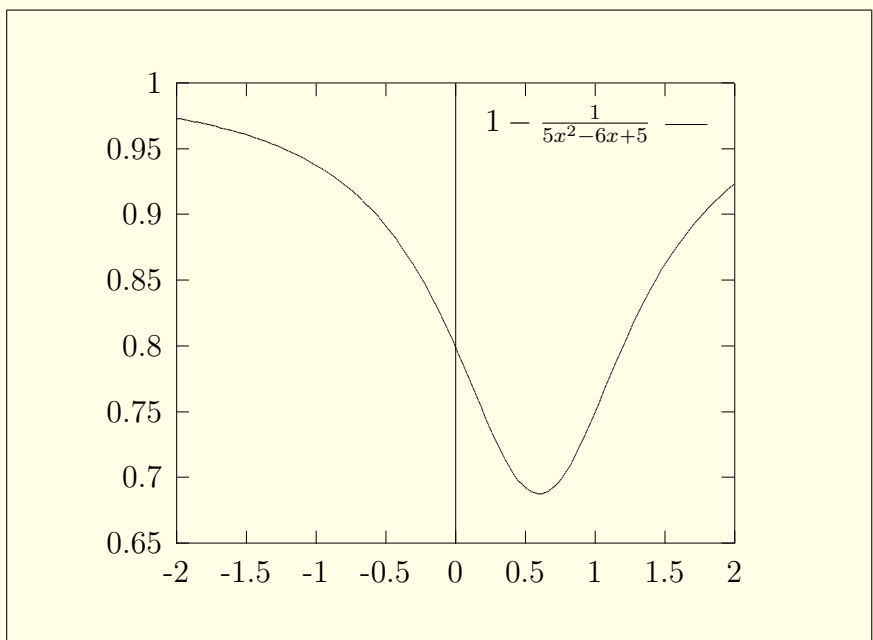


FIGURE 2.8 – Fonction $1 - \frac{1}{5x^2 - 6x + 5}$

Iter	x_0	x_1	$f'(x_1)$
0	-2.0000000000000000	2.0000000000000000	0.0828402366863905
1	2.0000000000000000	0.0000000000000000	-0.2400000000000000
2	0.0000000000000000	1.0000000000000000	0.2500000000000000
3	1.0000000000000000	0.5000000000000000	-0.0946745562130177
4	0.5000000000000000	0.6065573770491803	0.0064028281317895
5	0.6065573770491803	0.5999982373826895	-0.0000017213059672
6	0.6065573770491803	0.6000000000000000	0.0000000000000000

TABLE 2.3 – Itérations de l'algorithme combiné sur la fonction $1 - \frac{1}{5x^2 - 6x + 5}$.

à la précision de la machine, 10^{-16} .

Exercice 2.9.1 [*Combinaisons d'algorithmes*] Suite de l'exercice 2.3.4 page 58 Considérez la fonction $h(\theta) = 1 - 12\theta + 7.5\theta^2 - \theta^3$.

- Appliquez l'algorithme de Fibonacci pour réduire l'intervalle $[0, 3]$ qui contient un minimum local à un intervalle de longueur $\leq 1 + \epsilon$;
- appliquez l'algorithme de la section dorée pour réduire l'intervalle $[0, 3]$ qui contient un minimum local à un intervalle de longueur $\leq 1 + \epsilon$.
- Effectuez *une* itération de Newton à partir de $\theta = 0$; cette itération se rapproche-t-elle d'un minimum local de h ?
- Effectuez *une* itération de Newton à partir de $\theta = 3$; cette itération se rapproche-t-elle d'un minimum local de h ?
- Vérifiez qu'en appliquant l'algorithme combiné (avec $\gamma = 0.9$) à partir de l'intervalle $[\theta_0 = 0, \theta_1 = 3]$ (on tente une itération de Newton à partir de $\theta_1 = 3$), on passe à la méthode de Newton après seulement une utilisation de la bisection.
- Vérifiez qu'en appliquant l'algorithme combiné (avec $\gamma = 0.9$) à partir de l'intervalle $[\theta_1 = 0, \theta_0 = 3]$ (on tente une itération de Newton à partir de $\theta_1 = 0$), on passe à la méthode de Newton sans utiliser la bisection.

2.10 Région de confiance

Une autre approche pour profiter des méthodes de type Newton consiste à revenir à la condition d'optimalité de la proposition 2.5.1 page 69. Si $f''(x^*) > 0$, alors dans un voisinage de x^* , $f''(x) > 0$ comme nous l'avons vu. Cependant, en s'éloignant de x^* , il est fort possible que $f''(x) < 0$. La stratégie de combinaison avec l'algorithme de bisection permet de contourner cette difficulté. Une autre manière consiste à définir un voisinage de x , $|x| \leq \Delta$ à l'intérieur duquel $q_x(d)$ est une bonne approximation de $f(x+d)$ et de calculer le minimum (global) de $q_x(d)$ dans le voisinage. Il n'est pas difficile de se convaincre que le minimum est soit $d_N = \frac{-f'(x)}{f''(x)}$, soit $d_m = -\Delta$, soit $d_M = \Delta$. Si $f''(x) \leq 0$, le minimum n'est pas d_N .

La taille du voisinage Δ est mise à jour en utilisant la réduction prévue $pred(x) = q_x(d) - q_x(0)$ et la réduction actuelle $ared(x) = f(x+d) - f(x)$ dans le rapport $r(x) = \frac{ared(x)}{pred(x)}$. Si $r(x)$ est trop petit, on réduit Δ sans modifier x (par prudence); autrement on pose $x^+ \leftarrow x + d$ et si $r(x)$ est assez grand, on peut se permettre d'augmenter Δ .

La démonstration de convergence de ce type de méthode est un peu plus complexe, mais gardons en tête que l'algorithme 2.5 joue les rôles combinés de `trouve_intervalle` et

```

Newton—région de confiance( $x, \Delta, \epsilon, f$ )
{ Donnés :  $x$  ; }
{ un critère d'arrêt,  $\epsilon > 0$  ; }
{ une fonction deux fois différentiable  $f$  ; }
{ une taille initiale  $\Delta$  de la région de confiance. }
répéter
  si ( $q(\Delta) < q(-\Delta)$ ) alors  $d_R \leftarrow \Delta$ 
  sinon  $d_R \leftarrow -\Delta$ 
   $d_N \leftarrow -f'(x)/f''(x)$ 
  si ( $|d_N| < \Delta \wedge q(d_R) > q(d_N)$ ) alors  $d_R \leftarrow d_N$ 
   $ared \leftarrow f(x) - f(x + d_R)$ 
   $pred \leftarrow q(0) - q(d_R)$ 
   $r \leftarrow \frac{ared}{pred}$ 
  si ( $r < 0.25$ ) alors  $\Delta \leftarrow \Delta/2$ 
  sinon
     $x \leftarrow x + d_R$ 
  si ( $r > 0.75$ ) alors
     $\Delta \leftarrow 2 * \Delta$ 
jusqu'à ( $|f'(x)| < \epsilon$ )
  Résultat  $\leftarrow x$ 

```

Algorithme 2.5: Newton avec région de confiance.

bissection-Newton. Nous dénotons x_k et Δ_k , $ared_k$, $pred_k$ et r_k les quantités à une itération k .

Théorème 2.10.1 *Si les points x_k engendrés par l'algorithme demeurent dans un compact K , avec f de classe \mathcal{C}^2 dans K , et f'' uniformément bornée dans K , alors il existe un point d'accumulation \bar{x} de la suite $\{x_k\}$, et tout point d'accumulation satisfait que $f'(\bar{x}) = 0$ et $f''(\bar{x}) \geq 0$. De plus, si un point d'accumulation \bar{x} satisfait aux conditions suffisantes d'ordre deux, alors pour k assez grand, $d_R \equiv d_N$.*

Preuve Montrons d'abord que tout point d'accumulation satisfait aux conditions nécessaires d'ordre deux. Nous devons considérer les deux possibilités mutuellement exclusives que

1. $\Delta_k > \bar{\Delta} > 0$;
2. il y ait une sous-suite telle que $\lim_{i \rightarrow \infty} \Delta_{k_i} = 0$.

Dans tous les cas, puisque les x_k sont supposés demeurer dans un compact K et que la suite $f(x_k)$ est monotone décroissante, on a donc $\sum ared_k < \infty$, ce qui entraîne que $\lim_{k \rightarrow \infty} ared_k = 0$.

cas 1 Puisque $\Delta_k \geq \bar{\Delta} > 0$, il doit y avoir une sous-suite telle que $r_{k_i} > 1/4$. Puisque $r_{k_i} = \frac{ared_{k_i}}{pred_{k_i}}$ ne converge pas vers 0 on a $pred_{k_i} \rightarrow 0$. Ceci nous permet de justifier que $\bar{d} = 0$ est solution optimale de $\min_{|d| \leq \bar{\Delta}} \bar{q}(d) = f(\bar{x}) + f'(\bar{x})d + \frac{1}{2}f''(\bar{x})d^2$, qui constitue une condition nécessaire d'ordre deux pour que \bar{x} soit un minimum local selon la proposition 2.5.1. En effet, soient $\bar{d} \in \arg \min_{|d| \leq \bar{\Delta}} \bar{q}(d)$, $\hat{x} = \bar{x} + \bar{d}$ et $\hat{d}_k = \hat{x} - x_k$. Alors, comme d_k minimise q_{x_k} ,

$$q_{x_k}(\hat{d}_k) \geq q_{x_k}(d_k) = f(x_k) - pred_k$$

et en passant à la limite, $\bar{q}(\bar{d}) \geq f(\bar{x}) = \bar{q}(0)$ qui justifie que $\bar{d} = 0$ est solution optimale de $\min_{|d| \leq \bar{\Delta}} \bar{q}(d)$ et donc que \bar{x} satisfait à la condition nécessaire de la proposition 2.5.1.

cas 2 Considérons que la sous-suite est choisie telle que $\lim_{i \rightarrow \infty} \Delta_{k_i} = 0$ et en plus $r_{k_i} < \frac{1}{4}$ pour tout i . La direction d_R ne peut prendre que deux valeurs, $d_N = -f'(x_k)/f''(x_k)$ ou encore $d_M = \pm \Delta_k$ de signe opposé à $f'(x_k)$. Dans le premier cas, $pred_k = \frac{1}{2}|d_N||f'(x_k)|$ alors que dans l'autre cas, $pred_k = d_M(-f'(x_k) - \frac{1}{2}f''(x_k)d_M)$. Si $f'(\bar{x}) \neq 0$, dans les deux cas, on peut borner $pred_{k_i} \geq \frac{1}{2}\Delta_k|f'(x_k)|$. Si $f'(\bar{x}) = 0$ mais $f''(\bar{x}) < 0$, la direction n'est jamais d_N et on obtient $pred_{k_i} \geq \frac{1}{2}\Delta_k^2|f''(x_k)|$. Maintenant, $pred_k = f'(x_k)d_k + \frac{1}{2}f''(x_k)d_k^2$ et un développement de Taylor nous donne que $ared_k = f'(x_k)d_k + \frac{1}{2}f''(x_k)d_k^2 + o(|d_k|^2)$, où par définition $|d_k| \leq \Delta_k$.

En exprimant

$$r_k = \frac{f'(x_k)d_k + \frac{1}{2}f''(x_k)d_k^2 + \mathcal{O}(|d_k|^3)}{f'(x_k)d_k + \frac{1}{2}f''(x_k)d_k^2} = 1 + \frac{\mathcal{O}(|d_k|^3)}{pred_{k_i}},$$

on contredit que $r_{k_i} < \frac{1}{4}$ car $\frac{o(|d_{k_i}|^2)}{\text{pred}_{k_i}}$ converge vers zéro puisque $|d_k| \ll \Delta_k$ et $\text{pred}_{k_i} \geq \frac{1}{2}\Delta_k^2|f''(x_k)|$.

Montrons maintenant que si un point d'accumulation satisfait aux conditions suffisantes, l'algorithme se réduit à la méthode de Newton pure. Écrivons $\text{pred}_k = -f'(x_k)d_R - \frac{1}{2}f''(x_k)d_R^2 = f''(x_k)(d_N d_R - \frac{1}{2}d_R^2) \geq \frac{1}{2}f''(x_k)\Delta_k^2$. Dans ce cas, $r_k = 1 + \frac{\mathcal{O}(|d_k|^3)}{\text{pred}_{k_i}}$ s'approche de un, Δ_k demeure éloigné de 0 et d_N s'approche de zéro. \square

Sur la même fonction, voici le comportement de l'approche région de confiance.

Iter	x	$ f'(x) $	Δ	pred	ared
0	2.0	0.0828402	1		
1	1.0	0.25	2.	-0.142467	-0.1730769
2	1.0	0.25	1.	-0.25	+0.1875
3	1.0	0.25	0.5	-0.1875	+0.05
4	0.5	0.0946746	0.5	-0.109375	-0.0576923
5	0.6065574	0.0064028	1.0	-0.0050441	-0.0047867
6	0.5999982	0.0000017	2.0	-0.0000210	-0.0000210
7	0.6	0.0	4.	-1.517×10^{-12}	-1.517×10^{-12}

TABLE 2.4 – Itérations de l'algorithme de région de confiance sur la fonction $1 - \frac{1}{5x^2 - 6x + 5}$.

Exercice 2.10.1 [Parabole] Suite de la discussion de la section 1.4.2 pour la fonction $d(x) = (x - P_x)^2 + (x^2 - P_y)^2$. À partir de $x = 1$, appliquez l'algorithme 2.5 pour le point $P = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Exercice 2.10.2 [Ellipse] Suite de l'exercice 2.4.4 page 24 On a vérifié (exercice 2.4.4) qu'en prenant parmi les points sur les axes les deux plus proche de l'origine, l'un d'eux a une dérivée $-\infty$. Comme la dérivée à l'autre point est positive, appliquez l'algorithme 2.5 à partir de cet autre point.

Exercice 2.10.3 [Hyperboles] Suite de l'exercice 2.4.5 page 64 À partir de $x = 1$, appliquez l'algorithme 2.5.

2.11 Organisation des chapitres

La structure générale des chapitres qui suivent prend donc la forme suivante :

1. Formulation du problème.
2. Énoncé de conditions d'optimalité.
3. Dédution d'algorithmes à partir de points qui ne satisfont pas aux conditions d'optimalité.
4. Étude détaillée de quelques algorithmes spécifiques.
5. Solution des calculs non-triviaux des algorithmes spécifiques.

2.12 Tous les exercices du chapitre

Exercice (2.1.1, page 51) [*Min et inf*] Vérifiez que $0 = \inf e^x$ et $0 = f(0) = \min x^2$ satisfont à la définition de inf et min.

Exercice (2.2.1, page 53) [*Optima isolés*] Trouvez un exemple de fonction qui admet une accumulation de minima locaux à l'origine, pour laquelle l'origine est un minimum global.

Exercice (2.3.1, page 56) [*Condition d'optimalité*]

- a) L'origine est un point stationnaire, c'est-à-dire un point pour lequel $f'(x) = 0$, de la fonction $f(x) = x^2 \cos x$. Déterminez s'il s'agit d'un minimum (local), maximum (local) ou point d'inflexion.
- b) Même question pour les fonctions
 - i) $f(x) = x^2 \sin x$;
 - ii) $f(x) = x^2(1 - \cos x)$;
 - iii) $f(x) = x^2(\cos x - 1)$.

Exercice (2.3.2, page 57) [*Condition d'optimalité*]

- a) Montrez que la fonction

$$f(x) = (2 - x^2)^2 + x^5$$

possède deux points stationnaires, un maximum et un minimum local.

- b) Trouvez un exemple de fonction qui possède trois points stationnaires, un minimum local, un maximum local et un point d'inflexion.

Exercice (2.3.3, page 58) [*Analyse d'une fonction*] Considérez la fonction $h(\theta) = \frac{4\theta - 7}{\theta^2 + \theta - 2}$.

- a) Identifiez *tous* les points stationnaires de h ;

- b) parmi ceux-ci, identifiez *tous* les minima locaux de h ;
 c) identifiez deux points où la fonction n'est pas bornée.

Exercice (2.3.4, page 58) [*Analyse d'une fonction*] Considérez la fonction $h(\theta) = 1 - 12\theta + 7.5\theta^2 - \theta^3$.

- a) Identifiez *tous* les points stationnaires de h ;
 b) parmi ceux-ci, identifiez *tous* les minima locaux de h .

Exercice (2.3.5, page 58) [*Hyperboles*] Suite de l'exercice 1.4.2 page 25 Considérez une hyperbole de \mathbb{R}^2 , $y = 1/x$. Soit un point $P = \begin{pmatrix} P_x \\ P_y \end{pmatrix}$, on veut trouver le point de l'hyperbole le plus proche de P . Comme les points de l'hyperbole sont de la forme $\begin{pmatrix} x \\ 1/x \end{pmatrix}$, on peut ramener l'étude à la minimisation selon la seule variable x de la fonction $d(x) = (x - P_x)^2 + (\frac{1}{x} - P_y)^2$.

- a) Écrivez les conditions d'optimalité (d'ordre un et deux) de ce problème $\min_{x \in \mathbb{R}} d(x)$.
 b) Si le point P est sur la droite $y = x$, on devine qu'il y a 2 candidats solution naturels : $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ et $\begin{pmatrix} -1 \\ -1 \end{pmatrix}$. En vous limitant aux $x > 0$, vérifiez cette intuition en utilisant les conditions d'optimalité; vérifiez d'abord que $x = 1$ est un point stationnaire pour $d(x)$. Vérifiez ensuite que l'intuition n'est pas vraie si P est assez loin de l'origine; quantifiez ce "loin" de l'origine en utilisant les conditions d'optimalité d'ordre 2.)

Exercice (2.3.6, page 58) [*Cosinus*] Suite de l'exercice 1.4.3 page 25 On dispose d'un point $P_0 = (x_0, y_0)^t$ et on veut obtenir le point le plus proche de P_0 sur la courbe $P(x) = (x, \cos x)^t$. Formulez ce problème comme problème de minimisation $\min_{x \in \mathbb{R}} f(x)$ selon la variable x , donc fournissez une expression pour $f(x)$.

- a) Écrivez les conditions d'optimalité (d'ordre un et deux) de ce problème $\min_{x \in \mathbb{R}} d(x)$.
 b) Impossible d'aller plus loin dans le cas général sans outil de calcul. Considérons donc un point $P_0 = (0, y_0)^t$. Vérifiez que $x^* = 0$ satisfait aux conditions nécessaire d'ordre un pour le problème. Donc, le candidat pour le point le plus proche (minimum local) est $P^* = (x^*, \cos x^*)^t = (0, 1)^t$.
 c) Le candidat P^* n'est pas un minimum local de la distance pour n'importe quelle valeur de y_0 . Utilisez les conditions d'ordre deux pour obtenir les valeurs de y_0 garantissant que $P^* = (0, 1)^t$ est bel et bien minimum local de la distance.

Exercice (2.3.7, page 59) [*Norme et somme de normes*] Soient deux fonctions $f_1(x) = x$ et $f_2(x) = 2(x - 1)$.

- a) Considérez la somme des normes (valeurs absolues dans ce cas simpliste) $f(x) = |f_1(x)| + |f_2(x)|$. Vérifiez que $f(1) = 1$ et $f(x) \geq 1$ pour tout x . Indice : justifiez que f est de la forme

$$f(x) = \begin{cases} \text{expression 1,} & \text{si } x \leq 0 \\ \text{expression 2,} & \text{si } 0 \leq x \leq 1 \\ \text{expression 3,} & \text{si } x \geq 1 \end{cases}$$

- b) Considérez maintenant la somme des normes au carré, $\hat{f}(x) = f_1(x)^2 + f_2(x)^2$. Calculez x^* un minimum de $\hat{f}(x)$ en utilisant les conditions d'optimalité.
- c) Utilisez ce qui précède pour justifier que minimiser une somme de normes n'est pas équivalent à minimiser une somme de carrés de normes.

Exercice (2.4.1, page 62) [*Point milieu stationnaire*]

- a) Trouvez un exemple (une fonction f et un intervalle $[a, b]$) pour lequel les dérivées de f sont de signe opposé aux extrémités de l'intervalle ($f'(a) < 0$ et $f'(b) > 0$) pour lequel le point milieu de l'intervalle est un maximum local.
- b) Trouvez un pareil exemple pour lequel le point milieu de l'intervalle est un point d'inflexion.
- c) Vérifiez que l'algorithme 2.2 ne converge pas vers un maximum local pour vos deux exemples.

Exercice (2.4.2, page 62) [*Initialisation de bisection*]

- a) Dans le contexte de la première condition de terminaison de l'algorithme de recherche d'intervalle, justifiez qu'il existe un point de l'intervalle de bornes x_k et x_{k-1} dont la dérivée est de signe opposé à la dérivée en x_0 ;
- b) Proposez une technique de réduction d'intervalle pour identifier un tel point.

Exercice (2.4.3, page 64) [*Parabole*] Suite de la discussion de la section 1.4.2 pour la fonction $d(x) = (x - P_x)^2 + (x^2 - P_y)^2$.

- a) À partir de $x = 1$, appliquez l'algorithme `trouve_intervalle` pour le point $P = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.
- b) À partir de l'intervalle trouvé en a), effectuez une étape de l'algorithme `bisection` (toujours pour le point $P = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$).
- c) Recherchez d'autres points stationnaires pour ce problème, et identifiez leur nature.

Exercice (2.4.4, page 64) [*Ellipse*] Suite de l'exercice 1.4.1 page 24

- a) Vérifiez qu'en prenant parmi les points sur les axes les deux plus proche de l'origine, on ne peut pas appliquer directement l'algorithme de bisection car une des dérivées est $-\infty$.
- b) Comme la dérivée à l'autre point est positive, faites comme si la dérivée infinie était finie (et négative) et effectuez des itérations de bisection jusqu'à ce que le point à dérivée infinie soit remplacé.

Exercice (2.4.5, page 64) [*Hyperboles*] Suite de l'exercice 2.3.5 page 58

- a) À partir de $x = 1$, appliquez l'algorithme `trouve_intervalle` pour le point $P = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$; utilisez un paramètre $\delta = 0.5$.
- b) À partir de l'intervalle trouvé en a), effectuez une étape de l'algorithme `bisection` (toujours pour le point $P = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$).

Exercice (2.4.6, page 67) [*Applications d'algorithmes*]

- Pour la fonction $f(x) = e^{x(x-1)}$, effectuez trois itérations des deux algorithmes (bissection, Fibonacci) à partir de l'intervalle initial $[-1, 1]$.
- Même question pour la fonction $f(x) = x^2(1 - \cos x)$ sur l'intervalle $[-1, 0.5]$.
- Un inconvénient de la méthode de Fibonacci est de devoir fixer *a priori* le nombre d'itérations; si on se fixait une infinité d'itérations, en sachant que

$$\lim_{i \rightarrow \infty} \frac{F_{i+1}}{F_i} = \tau,$$

le nombre d'or $\tau = 1 + \frac{1}{\tau} = \frac{(1+\sqrt{5})}{2} = 1.618\dots$, on utiliserait ce rapport à chaque itération. Cette méthode, nommée *section dorée*, consiste donc à utiliser

$$a, \quad a + (2 - \tau)(b - a), \quad a + (\tau - 1)(b - a), \quad b$$

comme quatre points initiaux, et à conserver ce rapport à chaque itération. Effectuez trois itérations de cet algorithme sur les fonctions de a) et b).

Exercice (2.4.7, page 68) [*Bissection utilisant f et f'*] Écrivez l'algorithme complet de bissection applicable dès la sortie de l'algorithme `trouve_intervalle`.

Exercice (2.5.1, page 70) [*Approximation quadratique*] Fournissez les coefficients du polynôme quadratique issu des conditions $p(x_0) = f(x_0)$, $p(x_1) = f(x_1)$, et $p(x_2) = f(x_2)$. Fournissez aussi une expression pour le minimum de la fonction quadratique.

Exercice (2.5.2, page 71) [*Applications d'algorithmes*]

- Pour la fonction $f(x) = e^{x(x-1)}$, effectuez trois itérations des trois algorithmes (Newton, sécante, Approximation quadratique passant par trois points) à partir de l'intervalle initial $[-1, 1]$. Pour l'approximation utilisant trois points, choisir le troisième point en 0.
- Même question pour la fonction $f(x) = x^2(1 - \cos x)$ sur l'intervalle $[-1, 0.5]$.

Exercice (2.5.3, page 72) [*Sécante*] Suite de l'exercice 2.3.3 page 58 sur la fonction $h(\theta) = \frac{4\theta - 7}{\theta^2 + \theta - 2}$.

- Peut-on utiliser la méthode de la sécante pour identifier un minimum local de $h(\theta)$ dans l'intervalle $[-1.0, 0.9]$? Est-ce que le prochain point est meilleur que le point milieu de l'intervalle?
- Est-ce que le prochain point de la méthode de la sécante à partir des points 0 et 0.3 est meilleur que le point milieu de l'intervalle $[0, 0.3]$?
- Dans la méthode de la sécante, est-il possible que $x_{k+1} = x_k$? Si oui, dans quelles circonstances (donnez un exemple)? Si non, pourquoi?

Exercice (2.5.4, page 72) [*Parabole*] Suite de l'exercice 2.4.3 page 64

- a) À partir du dernier point obtenu par bisection, effectuez une itération de Newton-Raphson et vérifiez que le point obtenu satisfait approximativement aux conditions d'optimalité.
- b) À partir du dernier intervalle obtenu par bisection, effectuez une itération de la méthode de sécante et vérifiez que le point obtenu satisfait approximativement aux conditions d'optimalité.

Exercice (2.5.5, page 73) [*Ellipses...*] Suite de l'exercice 2.4.4 page 64...

- a) À partir du dernier point obtenu par bisection, effectuez une itération de Newton-Raphson et vérifiez que le point obtenu satisfait approximativement aux conditions d'optimalité.
- b) À partir du dernier intervalle obtenu par bisection, effectuez une itération de la méthode de sécante et vérifiez que le point obtenu satisfait approximativement aux conditions d'optimalité.

Exercice (2.5.6, page 73) [*Hyperboles...*] Suite de l'exercice 2.4.5 page 64... À partir de chacune des extrémités de l'intervalle final trouvé à l'exercice 2.4.5, page 64, effectuez une itération de l'algorithme **Newton-Raphson** (toujours pour le point $P = \binom{1}{2}$).

Exercice (2.5.7, page 74) [*Échec de l'approximation cubique*]

- a) Vérifiez que le polynôme d'interpolation cubique de la fonction de la figure 2.6 en utilisant les points de la figure ($x_0 = 0$ et $x_1 = -0.2$) ne possède pas de minimum local. Illustrez graphiquement, et interprétez cet échec de la méthode.
- b) Vérifiez que la méthode d'interpolation cubique échoue pour une fonction f quadratique. Proposez une technique plus robuste.

***Exercice (2.5.8, page 75)** [*Panne de la réduction d'intervalle par polynome cubique*] Fournissez un exemple pour lequel l'algorithme termine en un point qui satisfait $f'(\bar{x}) = 0$ et $f(\bar{x}) < f(x_k)$, mais pour lequel \bar{x} est un maximum local de f .

Exercice (2.6.1, page 76) [*Notation \mathcal{O}*]

- a) Démontrez les propriétés de la notation "grand O"
 - i) $f \sim \mathcal{O}(g)$ entraîne que $\alpha f \sim \mathcal{O}(g)$, où α est une constante finie;
 - ii) $f + g \sim \mathcal{O}(\max(|f|, |g|))$;
- b) que peut-on dire de $\frac{1}{f} \sim \mathcal{O}(\cdot)$?

Exercice (2.6.2, page 76) [*Ordre de convergence à zéro*] Soit g une fonction réelle de classe $\mathcal{C}^1(\mathbb{R})$ avec $g(\bar{x}) = 0$ et $g'(\bar{x}) \neq 0$. Montrez qu'il existe un voisinage de \bar{x} (défini comme $\{x : |x - \bar{x}| \leq \epsilon\}$) et des constantes appropriées $K_1, K_2, K_3, K_4 > 0$ telles que

$$\left. \begin{array}{l} K_1|x - \bar{x}| \leq |g(x)| \leq K_2|x - \bar{x}| \\ K_3|g(x)| \leq |x - \bar{x}| \leq K_4|g(x)| \end{array} \right\} \forall x : |x - \bar{x}| \leq \epsilon$$

Exercice (2.6.3, page 77) [*Propriétés de \mathcal{O}*] Soient f_1, f_2 et f_3 trois fonctions qui s'annulent en \bar{x} .

- Montrez que $f_1 \sim \mathcal{O}(f_3)$ et $f_2 \sim \mathcal{O}(f_3)$ implique $(f_1 + f_2) \sim \mathcal{O}(f_3)$.
- Montrez que $f_1 \sim \mathcal{O}(f_2)$ et $f_1 \sim \mathcal{O}(f_3)$ n'implique pas que $f_1 \sim \mathcal{O}(f_2 + f_3)$.

Exercice (2.6.4, page 79) [*Une propriété de la convergence superlinéaire*] Montrez :

- toute suite $\epsilon_k = x_k - \bar{x}$ convergeant à 0 satisfait :

$$\left| 1 - \frac{|x_{k+1} - x_k|}{|\epsilon_k|} \right| \leq \frac{|\epsilon_{k+1}|}{|\epsilon_k|}$$

- si la vitesse de convergence est superlinéaire, alors

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - x_k|}{|\epsilon_k|} = 1.$$

Exercice (2.7.1, page 82) [*Méthode de Newton*]

- Fournissez un exemple (une fonction et un point de départ) pour lequel l'algorithme de Newton diverge.
- Fournissez un exemple (une fonction et un point de départ) pour lequel l'algorithme de Newton oscille entre deux points.
- Fournissez un exemple pour lequel l'algorithme de Newton converge vers une racine, mais pour laquelle $g'(x^*) = 0$; vérifiez que dans ce cas, la vitesse de convergence asymptotique est seulement linéaire.
- Pour résoudre une équation $g(x) = 0$, analysez la vitesse de convergence de l'algorithme de Newton pour la famille de fonctions $g(x) = x^p + x^{(p+1)}$ proche de $x^* = 0$, une racine de $g(x)$; distinguez les cas $p = 1$ et $p \geq 2$.

Exercice (2.9.1, page 87) [*Combinaisons d'algorithmes*] Suite de l'exercice 2.3.4 page 58
Considérez la fonction $h(\theta) = 1 - 12\theta + 7.5\theta^2 - \theta^3$.

- Appliquez l'algorithme de Fibonacci pour réduire l'intervalle $[0, 3]$ qui contient un minimum local à un intervalle de longueur $\leq 1 + \epsilon$;
- appliquez l'algorithme de la section dorée pour réduire l'intervalle $[0, 3]$ qui contient un minimum local à un intervalle de longueur $\leq 1 + \epsilon$.
- Effectuez *une* itération de Newton à partir de $\theta = 0$; cette itération se rapproche-t-elle d'un minimum local de h ?
- Effectuez *une* itération de Newton à partir de $\theta = 3$; cette itération se rapproche-t-elle d'un minimum local de h ?

- e) Vérifiez qu'en appliquant l'algorithme combiné (avec $\gamma = 0.9$) à partir de l'intervalle $[\theta_0 = 0, \theta_1 = 3]$ (on tente une itération de Newton à partir de $\theta_1 = 3$), on passe à la méthode de Newton après seulement une utilisation de la bisection.
- f) Vérifiez qu'en appliquant l'algorithme combiné (avec $\gamma = 0.9$) à partir de l'intervalle $[\theta_1 = 0, \theta_0 = 3]$ (on tente une itération de Newton à partir de $\theta_1 = 0$), on passe à la méthode de Newton sans utiliser la bisection.

Exercice (2.10.1, page 90) [*Parabole*] Suite de la discussion de la section 1.4.2 pour la fonction $d(x) = (x - P_x)^2 + (x^2 - P_y)^2$. À partir de $x = 1$, appliquez l'algorithme 2.5 pour le point $P = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Exercice (2.10.2, page 90) [*Ellipse*] Suite de l'exercice 2.4.4 page 24 On a vérifié (exercice 2.4.4) qu'en prenant parmi les points sur les axes les deux plus proche de l'origine, l'un d'eux a une dérivée $-\infty$. Comme la dérivée à l'autre point est positive, appliquez l'algorithme 2.5 à partir de cet autre point.

Exercice (2.10.3, page 90) [*Hyperboles*] Suite de l'exercice 2.4.5 page 64 À partir de $x = 1$, appliquez l'algorithme 2.5.

