MÉTHODES NUMÉRIQUES ET OPTIMISATION

I. MAZARI
ÉQUIPE PÉDAGOGIQUE : JULIEN CLAISSE, JOAO PINTO MACHADO, PAUL PEGON

Table des matières

Date: Version du 19 avril 2022.

Ce polycopié peut avoir des différences notables avec le cours dispensé en amphithéâtre (qui seul fixe le programme de l'examen).

Introduction

Objet du cours. Dans ce cours, nous nous proposons de présenter différents algorithmes qui permettent de résoudre des équations numériques et de minimiser des fonctions.

Par "équations numériques", nous entendons des problèmes du type suivant : $f : \mathbb{R}^d \to \mathbb{R}^m$ étant donnée,

$$\text{Trouver } x \in \mathbb{R}^d \text{ tel que } f(x) = 0,$$

tandis que par "minimiser des fonctions" nous entendons des problèmes tels que : $f : \mathbb{R}^d \to \mathbb{R}$ étant donnée,

$$\text{Déterminer } x^* \in \mathbb{R}^d \text{ tel que, pour tout } x \in \mathbb{R}^d\,, f(x^*) \leq f(x).$$

Même si nous évoquerons des questions d'existence (et, possiblement, d'unicité) de solutions à ces problèmes, nous mettrons l'emphase sur des procédés itératifs permettant de répondre à ces deux questions.

Soulignons d'emblée que ces deux problèmes sont fortement reliés par l'observation suivante : en général, il est impossible de faire autrement, quand on part en quête d'un minimiseur, que de vérifier des conditions d'optimalité de premier et second ordre, qui seront revues en temps voulues. La règle de Fermat, en particulier, fournit une information cruciale : si l'on cherche le minimiseur d'une certaine fonction f, et que l'on suppose que cette fonction f est différentiable, de gradient ∇f alors, en général, il est bon de chercher un optimiseur dans l'ensemble

$$\mathcal{X} := \{x^* \in \mathbb{R}^d\,, \nabla f(x^*) = 0\}.$$

En particulier, il faut savoir résoudre l'équation

$$\nabla f = 0$$

ce qui nous ramène au premier type de problème évoqué.

Ce lien lui-même, en ce qu'il fait intervenir la différentielle, nous suggère également d'insister sur le fait que toutes les méthodes que nous proposons d'étudier ici sont des méthodes essentiellement locales, qui feront très rapidement intervenir les notions élémentaires de calcul différentiel.

Les notions étudiées ici seront en particulier illustrées en travaux dirigés, ainsi qu'en travaux pratiques.

Objectifs et plan du cours. Au vu du peu de démonstrations présentes dans leur entiéreté dans le cours, chacune est exigible au partiel et à l'examen. Le plan du cours se décline comme suit et variera perpétuellement :

(1) Premières notions de convergence, et considérations générales.

(2) Optimisation en dimension 1 : Méthode de dichotomie, de la section dorée, de Newton, de la sécante.

(3) Rappels de calcul différentiel, méthode de descente (heuristique).

(4) *Intermezzo* : existence d'optimiseurs dans certaines classes de fonctions, conditions d'optimalité d'ordre un et deux, optima dégénérés.

(5) Convergence de la méthode de descente de gradient à pas fixe. Début du choix d'une direction de descente optimale.

(6) Convergence des méthodes de descente à pas variables.

(7) Algorithmes d'ordre deux, méthodes de descente et convergence.

(8) Méthode du gradient conjugué.

(9) Optimisation sous contraintes : généralités et multiplicateurs de Lagrange.

(10) Optimisation sous contraintes linéaires : l'algorithme du simplexe.

Quelques références. Ce cours ne prétend pas à une grande originalité. Pour le construire, je me suis appuyé sur les cours donnés, les années précédentes, par D. Gontier et A. Frouvelle.

Le contenu de ce cours a été discuté avec l'ensemble de l'équipe pédagogique, MM. Claisse, Machado et Pegon.

Jupyter, Python. Les travaux pratiques du cours auront lieu sur l'interface Jupyter, qui vous permet d'écrire du Python. Assurez-vous d'avoir Python 3 installé avant de vous lancer dans l'installation de Jupyter, et assurez-vous également de disposer des libraires `numpy` et `matplotlib`.

Notations, conventions.

(1) Pour tout $a < b$, l'ensemble $(a; b)$ désigne l'intervalle ouvert de borne inférieure a et de borne supérieure b.

(2) $\mathbb{R}^d$ est l'espace euclidien de dimension d. Sauf mention explicite du contraire, il est muni de la métrique euclidienne.

(3) L'ensemble $M_{mn}(\mathbb{R})$ désigne l'ensemble des matrices à m lignes et n colonnes. On note $M_d(\mathbb{R})$ l'ensemble des matrices carrées de taille d.

(4) L'ensemble $S_d^+(\mathbb{R})$ est l'ensemble des matrices symétriques. L'ensemble $S_d^{++}(\mathbb{R})$ est l'ensemble des matrics symétriques définies positives.

1. Premières notions de convergence et méthodes unidimensionnelles

Pré-requis.

Convergence de suites. Nous l'avons évoqué, il sera nécessaire dans tout ce cours d'étudier des méthodes itératives. En particulier, nous serons amenés à étudier la convergence de suite générées par des algorithmes, disons $\{x_k\}_{k\in\mathbb{N}}$. Ainsi, il est nécessaire d'avoir une bonne maîtrise des techniques classiques permettant de démontrer la convergence des suites et des séries, qu'elles soient numériques ou à valeur dans les espaces de Banach.

Calcul différentiel. En outre, la plupart des méthodes que nous proposons d'étudier reposent sur les notions de dérivées et de différentielles. Assurément, nous nous contenterons détudier l'optimisation de fonctions très régulières, ce qui nous permettra d'éviter ces "plaies que sont les fonctions différentiables nulle part"-non que le sujet n'aie pas d'intérêt, mais il demanderait des notions et un temps qui excèdent largement le présent enseignement. Ainsi, une **bonne** maîtrise technique des calculs de gradients, de Hessiennes, est nécessaires. Nous reverrons les bases de ces notions dans le cours n° 4.

Les fiches de Travaux Dirigés devraient vous aider à reprendre ces bases (si elles vous faisaient défaut), et nous supposerons ici un certain nombre de résultats ou de techniques comme acquis.

Concernant la précision des ordinateurs. Un des autres enjeux cruciaux de tout ce cours est évidemment celui de la précision numérique que l'on peut obtenir par des méthodes numériques implémentées sur ordinateur. En particulier, nous nous intéresserons, dans ce cours, au nombre de décimales que l'on peut gagner à chque itération d'un algorithme.

1.1. Type d'algorithme.

Comme répété plus haut, toutes les méthodes de ce cours sont des méthodes *itératives*. Afin de les expliquer de manière générale, imaginons que l'on cherche à résoudre ou bien

$$\text{Trouver } x \in \mathbb{R}^d \text{ tel que } f(x) = 0 \tag{1.1}$$

ou bien

$$\text{Trouver } x \in \mathbb{R}^d \text{ tel que } f(x) = \min_{\mathbb{R}^d} f. \tag{1.2}$$

Désignons par $x^* \in \mathbb{R}^d$ une solution de l'un ou l'autre de ces problèmes. Un *algorithme itératif* génère, à partir d'un point initial x_0 et de la fonction $f : \mathbb{R}^d \to \mathbb{R}^m$ ($m = 1$ pour (1.2)), une *suite de points* $\{x_k\}_{k\in\mathbb{N}}$ qui devrait satisfaire, si tout se passe bien,

$$x_k \underset{k\to\infty}{\to} x^*. \tag{1.3}$$

Remarque 1.1. On devrait en général prendre garde à la topologie de la convergence (1.3) ; néanmoins, puisque nous travaillons dans tout ce cours en dimension finie, il n'y a pas lieu de distinguer entre convergence en norme (par ailleurs toutes équivalentes) et convergence faible. Cette remarque qui semble faite pour chipoter prend toute son importance dès que l'on commence à traiter de problèmes d'optimisation en dimension infinie, ce qui relève alors du *calcul des variations*.

Bien sûr, la vie étant ce qu'elle est, il est illusoire d'espérer qu'au bout d'un nombre d'itération, même très grand, l'on puisse tomber exactement sur une solution x^* du problème considéré. Il est donc nécessaire d'imposer un *critère de tolérance*, qui joue le rôle d'une règle d'arrêt. Pour le problème (1.1), fixons un seuil $\varepsilon > 0$. On peut considérer que si notre itération produit, pour N assez grand, un point $x_N \in \mathbb{R}^d$ tel que

$$\|f(x_N)\| \leq \varepsilon$$

alors x_N peut être considéré comme une bonne approximation d'une solution x^*. Le cas des problèmes de type (1.2) est un peu plus délicat. En somme, la structure d'un tel *algorithme itératif* est ainsi la suivante :

Algorithm 1 Structure type d'un algorithme itératif

```
def algo(f,x0,...,tol=1e-3,IterMax=10)
Initialisation
xn=x0   On définit le premier élément de la liste
L=[]   On définit une liste que l'on augmentera récursivement, et qui contiendra
tous les éléments générés
for n in range(IterMax):
if ...< tol : then
  return xn,L
else
  L.append(xn)
  xn=...
end if
print("Erreur, l'algorithme n'a pas convergé
après",IterMax,"itérations")
```

Mettons que l'algorithme converge. Évidemment, on veut savoir quelle efficacité cet algorithme peut avoir, ce qui revient à demander à quelle vitesse cet algorithme converge. Dans la prochaine section, nous regardons plusieurs de ces types de convergence.

1.2. **Vitesses de convergence des algorithmes itératifs.** Une première remarque : si la vitesse de convergence d'un algorithme est la vitesse (définie plus loin) à laquelle la suite d'itérations $\{x_k\}_{k\in\mathbb{N}}$ converge vers x^*, cette vitesse de convergence théorique ne coïncide pas forcément avec la vitesse réelle observée sur l'ordinateur, qui doit également prendre en compte le temps dévaluation de certaines quantités par l'ordinateur. Si vous trouvez par exemple un algorithme tel que la suite $\{x_k\}_{k\in\mathbb{N}}$ satisfasse

$$\|x_k - x^*\| \leq e^{-e^k}$$

donc en sur-exponentiel, mais que passer de x_k à x_{k+1} demande à l'ordinateur d'effectuer $e^{e^{e^{e^k}}}$ opérations, l'algorithme risque d'être peu utilisable en pratique.

1.2.1. *Convergence linéaire.* Le premier exemple, le plus fréquemment rencontré, est celui de la **convergence linéaire**.

Définition 1.1 (Convergence linéaire)**.** On dit qu'une suite $\{x_k\}_{k\in\mathbb{N}}$ **converge linéairement vers** x^* **à taux** $\alpha \in (0,1)$ s'il existe une constante $C = C_\alpha > 0$ telle que

$$\|x_k - x^*\| \leq C_\alpha \alpha^k.$$

L'infimum des $\alpha \in (0,1)$ vérifiant cette propriété est appelé **taux de convergence de la suite**. Si cet infimum est égal à 0, on parle de **convergence sur-linéaire**.

Moralement, que nous donne cette définition ? Si l'on suppose que $\alpha = \frac{1}{10}$ et que $C_{\frac{1}{10}} = 1$, alors, à chaque étape de l'algorithme, on améliore la précision d'une décimale par itération. En d'autres termes, *Un algorithme linéaire gagne un nombre fixe de décimales par itérations*. Notez que cela peut être utilisé, *a priori*, comme un critère d'arrêt : on s'arrête une fois déterminée la solution à dix décimales.

Notons également la chose suivante : soit $\{x_k\}_{k\in\mathbb{N}}$ une suite qui converge vers un certain x^* à un taux optimal $\alpha \in (0,1)$, avec une constante optimale C_α. Alors on a

$$\|x_k - x^*\| \sim C_\alpha \alpha^k.$$

Maintenant, comment illustrer cette convergence ? Assez facilement : si on a une suite $\{x_k\}_{k\in\mathbb{N}}$ qui converge vers x^* à un taux $\alpha \in (0;1)$ optimal, alors on a

$$\ln\left(\|x_k - x^*\|\right) \sim \ln(C_\alpha) + k\ln(\alpha).$$

En particulier, si on représente la suite $\{\|x_k - x^*\|\}_{k\in\mathbb{N}}$ et que l'on observe, asymptotiquement, une droite, on peut lire le taux de convergence sur le coefficient directeur de cette droite.

Certes, mais l'on objectera que pour représenter ces taux de convergence, il importe de connaître la limite x^*. En pratique, on calculera les N premiers termes de la suite, pour N grand, et l'on prendre $x_N = x^*$.

Un critère particulièrement pratique pour la convergence linéaire est le suivant :

Proposition 1.1. *Soit $\{x_k\}_{k\in\mathbb{N}}$ une suite de $\mathbb{R}^d$. Supposons qu'il existe $\alpha \in (0,1)$ tel que*

$$\forall k \in \mathbb{N}, \|x_{k+1} - x_k\| \leq \alpha \|x_k - x_{k-1}\|.$$

Alors la suite admet une unique valeur d'adhérence $x^* \in \mathbb{R}^d$, et converge linéairement vers x^* avec un taux inférieur à α.

En d'autres termes, une contraction stricte des distances implique une convergence linéaire.

Preuve de la Proposition 1.1. Par une récurrence immédiate on obtient

$$\forall k \in \mathbb{N}, \|x_{k+1} - x_k\| \leq \alpha^k \|x_1 - x_0\|.$$

Commençons par montrer que ceci implique que la suite $\{x_k\}_{k\in\mathbb{N}}$ est une suite de Cauchy. On se fixe $(k,p) \in \mathbb{N}^2$. Alors

$$\begin{aligned}
\|x_k - x_{k+p}\| &\leq \sum_{i=1}^{p} \|x_{k+i} - x_{k+(i-1)}\| \\
&\leq \alpha^k \|x_1 - x_0\| \sum_{i=1}^{p} \alpha^{i-1} \\
&\leq C\alpha^k
\end{aligned}$$

puisque la série $\sum \alpha^i$ converge. En particulier, la suite est de Cauchy et donc converge vers un certain $x^* \in \mathbb{R}^d$. Par ailleurs, on a l'estimation suivante : il existe $C > 0$ tel que pour tout $k \in \mathbb{N}$, pour tout $p \in \mathbb{N}$,

$$\|x_k - x_{k+p}\| \leq C\alpha^k;$$

en passant à la limite $p \to \infty$ on retrouve la convergence linéaire annoncée. □

1.2.2. *Convergence quadratique.* Il se peut que la suite converge bien plus rapidement que linéairement ; un exemple typique est celui de la **convergence quadratique** :

Définition 1.2 (Convergence quadratique)**.** On dit qu'une suite $\{x_k\}_{k\in\mathbb{N}}$ **converge quadratiquement vers** x^* **à ordre au moins** $\beta > 1$ s'il existe une constante $C > 0$ et $\alpha \in (0, 1)$ tels que

$$\|x_k - x^*\| \leq C\alpha^{\beta^k}.$$

Le supremum des β vérifiant cette condition est appelé **ordre de convergence** de la suite. Si $\beta = 2$, on parle de **convergence quadratique**.

Nous verrons plus tard, quand nous aborderons la méthode de Newton, l'intérêt de spécifier le taux $\beta = 2$. Si on reprend l'exemple d'un $\alpha = \frac{1}{10}$, avec cette fois $\beta = 2$, alors on se rend compte qu'on double, à chaque itération, le nombre de décimales gagnées. Le problème sur ce type de convergence, c'est qu'il est beaucoup plus difficile de l'observer numériquement (même si l'on peut effectivement essayer de tracer des ln(ln()), les résultats sont rarement probants). Néanmoins si, en échelle logarithmique, on décroît toujours plus vite que des droites, on peut *a minima* dire que la convergence est sur-linéaire.

Nous avons pour la convergence quadratique l'analogue de la Proposition 1.1.

Proposition 1.2. *Soit $\{x_k\}_{k\in\mathbb{N}}$ une suite de $\mathbb{R}^d$. Supposons qu'il existe $C > 0\,, \beta > 1$ tels que, d'une part*

$$\alpha := C^{\frac{1}{\beta-1}}\|x_1 - x_0\| < 1$$

et que d'autre part

$$\forall k \in \mathbb{N}\,, \|x_{k+1} - x_k\| \leq C\|x_k - x_{k-1}\|^\beta.$$

Alors la suite admet une unique valeur d'adhérence $x^* \in \mathbb{R}^d$, et converge vers x^* à l'ordre β et à taux α.

Preuve de la Proposition 1.2. Par une récurrence immédiate on obtient

$$\forall k \in \mathbb{N}\,, \|x_{k+1} - x_k\| \leq C^{\sum_{i=0}^{k-1}\beta^i}\|x_1 - x_0\|^{\beta^k} = C^{-\frac{1}{\beta-1}}\alpha^{\beta^k}.$$

Une fois de plus, ceci implique que la suite $\{x_k\}_{k\in\mathbb{N}}$ est une suite de Cauchy : on se fixe $(k, p) \in \mathbb{N}^2$. Alors

$$\begin{aligned}\|x_k - x_{k+p}\| &\leq \sum_{i=0}^{p}\|x_{k+i} - x_{k+(i-1)}\| \\ &\leq C^{-\frac{1}{\beta-1}}\sum_{i=0}^{p}\alpha^{\beta^{k+i-1}}.\end{aligned}$$

Néanmoins, on observe que pour tout indice i on a l'estimation

$$\alpha^{\beta^{k+i}} \leq \alpha^{\beta^k}\alpha^{i(\beta-1)}.$$

Ceci vient simplement de la majoration

$$(1+x)^\gamma \geq 1+\gamma x \text{ si } x \geq 0.$$

Cette estimation implique que, pour tout indice i, on a

$$\beta^i \geq 1 + i\,(\beta - 1)$$

qui, puisque $\alpha < 1$, implique à son tour que

$$\alpha^{\beta^{k+i}} \leq \alpha^{\beta^k(1+i(\beta-1))} \leq \alpha^{\beta^k + i(\beta-1)}$$

On conclut exactement de la même manière qu'avant. □

1.3. **Discrétisation des fonctions.** Un autre aspect essentiel quand nous traiterons d'algorithmes de minimisation ou de résolution est la discrétisation des fonctions considérées. Rappelons donc que, si l'on sait calculer de manière efficace les zéros d'une fonction f, il sera possible de chercher des optimiseurs de certaines fonctions en cherchant leurs points critiques, c'est-à-dire les points où la dérivée de la fonction s'annule. En pratique, on peut évidemment essayer de calculer ∇F, F étant la fonction que l'on cherche à optimiser, de manière explicite, mais, fréquemment, cela n'est ni possible ni souhaitable. On choisit alors d'approcher les dérivées de la fonction F numériquement, par *différences finies*. Il s'agit de "geler" le quotient définissant la dérivée.

Définition 1.3 (Dérivée discrète)**.** Soit $F \in \mathscr{C}^1(\mathbb{R}^d, \mathbb{R})$. Soit $\delta > 0$ un pas de discrétisation. L'approximation par différences finies (resp. différences finies centrées) de ∇F d'ordre δ est la fonction $\tilde{\nabla}_\delta F : \mathbb{R}^d \to \mathbb{R}^d$ définie par

$$\forall i \in \{1,\dots,d\} \left(\tilde{\nabla}_\delta F\right)(x)_i := \frac{F(x+\delta\vec{e_i}) - F(x)}{\delta}$$
$$\left(\text{resp. } \left(\tilde{\nabla}_{\delta,\text{centrée}} F\right)(x)_i := \frac{F(x+\delta\vec{e_i}) - F(x-\delta\vec{e_i})}{2\delta}\right).$$

Dans l'expression ci-dessus, $\{\vec{e_i}\}_{i=1,\dots,d}$ est la base canonique de $\mathbb{R}^d$.

Plus $\delta > 0$ est petit, meilleure est l'approximation et on peut la quantifier. Néanmoins, il est inutile de chercher à travailler avec des pas de discrétisation trop petits. En effet, les ordinateurs ont une précision de calcul limitée η, qui quantifie le nombre de chiffres significatifs qu'un ordinateur peut retenir. En pratique, $\eta \sim 10^{-16}$.

Ceci a pour première conséquence qu'une fonction F numérique n'est jamais définie qu'à un ordre η près ; en d'autres termes, toute fonction F est approchée par une fonction numérique F_η telle que

$$\|F_\eta - F\|_\infty \leq \eta.$$

Et en particulier c'est sur F_η que l'on va calculer le gradient discrétisé.

Quand on travaille sur des gradients discrétisés, il faut donc trouver le pas de discrétisation en-deça duquel il n'y a plus d'intérêt à raffiner. Rendons-cela plus quantitatif par un raisonnement assez caractéristique du domaine, en essayant de quantifier par nos différents paramètres la différence entre la véritable dérivée d'une fonction réelle, et sa dérivée approchée. Dans les deux cas, on doit obtenir une borne de la forme

$$\|\tilde{\nabla}\delta/\tilde{\nabla}_{\delta,\text{centrée}} F_\eta - F'\|_\infty \leq \phi(\delta,\eta).$$

Une telle estimation ne peut être que locale. On se place dans le cadre suivant : soit $F \in \mathscr{C}^2([0,1],\mathbb{R})$. On distingue les deux approximations :

(1) Approximation par différences finies (fonctions $\mathscr{C}^2$) : dans l'approximation par différences finies, considérons donc $x \in [0,1]$. Alors on a

$$\begin{aligned}\left|\frac{F_\eta(x+\delta)-F_\eta(x)}{\delta} - F'(x)\right| &\leq \left|\frac{F_\eta(x+\delta)-F(x+\delta)}{\delta}\right| + \left|\frac{F_\eta(x)-F(x)}{\delta}\right| \\ &\quad + \left|\frac{F(x+\delta)-F(x)}{\delta} - F'(x)\right| \\ &\leq 2\frac{\eta}{\delta} + \frac{\delta}{2}\|F''\|_{L^\infty([0,1])}.\end{aligned}$$

Si l'on veut minimiser l'erreur en $\delta > 0$, on aboutit ainsi à choisir

$$\delta \sim \sqrt{\eta},$$

ce qui donne une erreur d'ordre $\sqrt{\eta}$.

(2) Approximation par différences finies centrées (fonctions $\mathscr{C}^3$) : le calcul est similaire, mais on suppose que F est $\mathscr{C}^3$. On a, pour tout $x \in (0,1)$,

$$\begin{aligned}\left|\frac{F_\eta(x+\delta)-F_\eta(x-\delta)}{2\delta} - F'(x)\right| &\leq \left|\frac{F_\eta(x+\delta)-F(x+\delta)}{2\delta}\right| \\ &\quad + \left|\frac{F_\eta(x-\delta)-F(x-\delta)}{2\delta}\right| \\ &\quad + \left|\frac{F(x+\delta)-F(x-\delta)}{2\delta} - F'(x)\right| \\ &\leq \frac{\eta}{\delta} + \left|\frac{F(x+\delta)-F(x-\delta)}{2\delta} - F'(x)\right|.\end{aligned}$$

Néanmoins, on peut écrire

$$F(x \pm \delta) = F(x) \pm \delta F'(x) + \frac{\delta^2}{2}F''(x) + \frac{\delta^3}{2}\int_0^1 (1-t)^2 F^{(3)}(x+\delta t)dt.$$

En sommant les deux termes, on aboutit ainsi à l'existence d'une constant $M > 0$ telle que

$$\left|\frac{F_\eta(x+\delta)-F_\eta(x-\delta)}{2\delta} - F'(x)\right| \leq \frac{\eta}{\delta} + M\delta^3.$$

Si on minimiser cette fois l'erreur obtenue en δ, on tombe sur $\delta \sim \eta^{\frac{1}{3}}$.

Une dernière remarque : pour le moment, tous nos calculs se généralisent sans aucune difficulté au cas des espaces de Banach.

FEUILLE DE TD N°1 : CONVERGENCE DE SUITES, CONVERGENCE DE SÉRIES

1.4. **Exercices de base sur les suites et leur convergence.**

Exercice 1.1. On définit, pour tout entier naturel $n \geq 1$, u_n par

$$u_n := \sum_{k=1}^{n} \frac{1}{n+k}.$$

Quelle est la nature de $\{u_n\}_{n\geq 1}$? Et si cette suite est convergente, quelle est sa limite ?

Exercice 1.2. On définit pour $n \geq 0$ u_n par

$$u_n := \sum_{k=0}^{n} \frac{1}{n^2+n+k}.$$

Déterminer la nature et la limite éventuelle de $\{u_n\}_{n\geq 0}$ ainsi que de la suite $\{v_n\}_{n\geq 0}$ définie par

$$\forall n \in \mathbb{N},\ v_n = nu_n.$$

Exercice 1.3.
(1) Soit K un compact de $\mathbb{R}^d$. Soit $\{x_k\}_{k\in\mathbb{N}} \in K^{\mathbb{N}}$ une suite n'admettant qu'une seule valeur d'adhérence. Montrer que toute la suite converge.
(2) Soit $C \subset \mathbb{R}^d$. Démontrer que C est compact si, et seulement si, pour toute suite à valeurs dans C, cette suite converge si et seulement si elle admet une unique valeur d'adhérence.

Exercice 1.4 (Théorème de Cesàro et variantes). Il est fréquent qu'une suite ne converge pas, mais qu'une moyenne de ses termes ait de meilleures propriétés. C'est l'essence du théorème de Cesàro, dont nous proposons d'étudier une variante, le théorème de Stolze : on considère deux suites $\{x_n\}_{n\in\mathbb{N}}$ et $\{y_n\}_{n\in\mathbb{N}}$. On suppose que $\{y_n\}_{n\in\mathbb{N}}$ est une suite croissante, non bornée et à termes strictement positifs.

(1) Montrer que si
$$\lim_{n\to+\infty} \frac{x_{n+1}-x_n}{y_{n+1}-y_n} = l \in \overline{\mathbb{R}}$$
alors la suite $\left\{\frac{x_n}{y_n}\right\}_{n\in\mathbb{N}}$ converge vers l. En déduire le théorème de Cesàro.
(2) Soit $\{x_n\}_{n\in\mathbb{N}}$ une suite réelle telle que
$$\lim_{n\to+\infty} \left(x_{n+1} - \frac{1}{2}x_n\right) = 0.$$
Montrer que la suite tend vers 0.

1.5. **Taux de convergence.**

Exercice 1.5 (Pour se mettre en jambe). Démontrer que, si une suite $\{x_k\}_{k\in\mathbb{N}}$ converge linéairement vers x^* avec un taux α alors, pour tout $\alpha' > \alpha$ $\{x_k\}_{k\in\mathbb{N}}$ converge linéairement vers x^* à taux α'.

Exercice 1.6 (Calcul de vitesse de convergence). Calculer les limites et les vitesses de convergence des suites suivantes :

(1) $u_n := \sum_{k=1}^{n} \frac{1}{k(k+1)}$,
(2) $u_n := \sum_{k=1}^{n} e^{-k}$,

(3) $u_n := \sin(n^{-1}) + \cos(n^{-1}) - \ln(1 + n^{-1})$.

Exercice 1.7 (Une convergence sous-linéaire, et une convergence sur-linéaire et sous–quadratique). Estimer la vitesse de convergence des

$$x_k = \sum_{k=1}^{n} \frac{1}{k^s} (s > 1)\,, x_k = \sum_{k=1}^{n} k e^{-k^2}.$$

On n'oubliera pas de justifier la convergence des suites en question.

Exercice 1.8. Déterminer la vitesse de convergence de la suite

$$u_n := \sin\left(\pi(1 + \sqrt{2})^n\right).$$

On pourra montrer que $u_n = -\sin(\pi(\sqrt{2} - 1)^n)$.

1.6. **Suites récurrentes : un exercice fondamental.**

Exercice 1.9 (Points fixes). Soit $f \in \mathscr{C}^2(\mathbb{R})$ et soit $x^* \in \mathbb{R}$ tel que $f(x^*) = x^*$. On suppose qu'il existe $\alpha \in (0, 1)$ tel que $|f'(x^*)| < \alpha$.

(1) Montrer que si $\delta > 0$ est suffisamment petit, $f'\left((x^* - \delta, x^* + \delta)\right) \subset (-\alpha, \alpha)$.

(2) On choisit $x_0 \in (x^* - \delta, x^* + \delta)$ et l'on définit, par récurrence

$$\forall k \in \mathbb{N}\,, x_{k+1} = f(x_k).$$

Illustrer géométriquement cette construction. Montrer que $\{x_k\}_{k\in\mathbb{N}}$ converge linéairement à taux α vers x^*.

(3) Montrer que si $f'(x^*) = 0$, $\{x_k\}_{k\in\mathbb{N}}$ converge à l'ordre 2 vers x^*.

Correction de la feuille de TD n°1

Correction 1.1. *On se sert du résultat suivant : toute suite majorée croissante converge. Il est clair que $\{u_n\}_{n\geq 1}$ est bornée par la constante 1. Par ailleurs, pour tout $n \geq 1$ on a*

$$\begin{aligned} u_{n+1} - u_n &= -\sum_{k=1}^{n} \frac{1}{k+n} + \sum_{k=1}^{n} \frac{1}{k+n+1} \\ &= \frac{-1}{n+1} + \frac{1}{2n+2} + \frac{1}{2n+1} \\ &= \frac{1}{2(n+1)(2n+1)} \geq 0. \end{aligned}$$

La suite converge donc. Pour établir la valeur de sa limite, deux solutions : soit on sait ce qu'est une somme de Riemann, soit on le fait à la main en utilisant l'inégalité

$$\forall x > -1, \ \log(1+x) \leq x.$$

En particulier, pour $y < 1$, on obtient

$$\log\left(\frac{y+1}{y}\right) \leq \frac{1}{y}.$$

L'idée est d'ensuite appliquer un théorème des gendarmes ; il nous faut donc une deuxième estimation. Mais si $x > 1$ alors

$$\log\left(\frac{x}{x-1}\right) = -\log 1 - \frac{1}{x}.$$

La même inégalité appliquée avec $y = -\frac{1}{x}$ donne alors l'encadrement

$$\sum_{k=1}^{n} \log\left(\frac{n+k+1}{n+k}\right) \leq \sum_{k=1}^{n} \frac{1}{n+k} \leq \sum_{k=1}^{n} \log\left(\frac{n+k}{n+k-1}\right).$$

Or ces deux sommes qui encadrent sont téléscopiques et tendent toutes deux vers $\log(2)$. Le théorème des gendarmes permet de conclure.

Correction 1.2. *C'est un exercice sans subtilité : une majoration brutale donne $u_n \to 0$. Pour la convergence de la deuxième suite, on minore : on remarque que*

$$\forall n \geq 0, \ \frac{1}{n^2+2n} \times (n+1) \leq u_n \leq \frac{1}{n^2+n} \times (n+1).$$

Ainsi, la suite v_n tend vers 1.

Correction 1.3. *Classique*

Correction 1.4. *(1) On suppose d'abord que $l \in \mathbb{R}$. Soit $\varepsilon > 0$. Il existe n_ε tel que pour tout $n \geq n_\varepsilon$*

$$l - \frac{\varepsilon}{2} < \frac{x_{n+1} - x_n}{y_{n+1} - y_n} < l + \frac{\varepsilon}{2}.$$

Comme $\{y_n\}_{n\in\mathbb{N}}$ est une suite croissante, pour tout $n \geq n_\varepsilon$ on a

$$\left(l - \frac{\varepsilon}{2}\right)(y_{n+1} - y_n) < x_{n+1} - x_n < \left(l + \frac{\varepsilon}{2}\right)(y_{n+1} - y_n).$$

Soit $m > n_\varepsilon$. On somme les inégalités précédentes entre n_ε et m. On obtient alors

$$\left(l - \frac{\varepsilon}{2}\right)(y_m - y_{n_\varepsilon}) < x_m - x_{n_\varepsilon} < \left(l + \frac{\varepsilon}{2}\right)(y_m - y_{n_\varepsilon}).$$

Comme $\{y_n\}_{n\in\mathbb{N}}$ est une suite à termes strcitement positifs, on obtient

$$l - \frac{\varepsilon}{2} - l\frac{y_{n_\varepsilon}}{y_m} + \frac{\varepsilon y_{n_\varepsilon}}{2y_m} + \frac{x_{n_\varepsilon}}{y_m} < \frac{x_m}{y_m} < l + \frac{\varepsilon}{2} l\frac{y_{n_\varepsilon}}{y_m} + \frac{\varepsilon y_{n_\varepsilon}}{2y_m} + \frac{x_{n_\varepsilon}}{y_m}.$$

Comme $\lim y_n = +\infty$, les restes que nous avons ajoutés sont plus petits que $\frac{\varepsilon}{2}$ pour m assez grand. Ceci conclut la démonstration.
Attaquons à présent le cas $l = +\infty$. En fait, il se traite quasiment de la même manière, ainsi que le cas $l = -\infty$.

(2) On définit $z_n := x_{n+1} - \frac{1}{2}x_n$. Alors

$$2^n x_n = x_0 + \sum_{k=1}^{n} 2^k b_{k-1}.$$

On applique le théorème de Stolze aux suites

$$e_n = \left(x_0 + \sum_{k=1}^{n} 2^k b_{k-1}\right),\ f_n = 2^n.$$

Correction 1.5. *Évident car $\alpha^n < (\alpha')^n$.*

Correction 1.6. *(1) On a $\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1}$, donc x_n est une série télescopique, et on trouve $x_n = 1 - \frac{1}{n+1}$. La suite x_n converge vers 1, et $|1 - x_n| = \frac{1}{n+1}$. C'est une convergence très lente.*

(2) On reconnait une série géométrique, donc $x_n = \frac{1-e^{-n-1}}{1-e^{-1}}$. La suite x_n converge vers $x^* := 1/(1-e^{-1})$, et $x^* - x_n$ est le reste de la série géométrique : $|x^* - x_n| = e^{-n}\frac{1}{1-e^{-n}}$. C'est une convergence linéaire à taux e^{-1}.

(3) On fait le développement limité à l'ordre 3 de la fonction $f(x) = \cos(x) + \sin(x) - \log(1+x)$. On trouve

$$f(x) = 1 - \frac{x^2}{2} + x - \frac{x^3}{6} - x + \frac{x^2}{2} - \frac{x^3}{3} + o(x^3) = 1 - \frac{x^3}{2} + o(x^3).$$

On en déduit que x_n converge vers $x^* = 1$, et que $|x^* - x_n| \approx n^{-3}$. C'est une convergence lente.

Correction 1.7. *Pour la première suite, on passe directement par une comparaison avec une intégrale. En effet, en posant*

$$f : x \mapsto x^{-s} \in L^1([1; +\infty)).$$

on a, en notant x^* la limite de x_k,

$$\int_{n+1}^{\infty} f \leq |x^* - x_n| \leq \int_n^{\infty} f \sim \frac{1}{1-s} n^{1-s}.$$

En particulier, on a une convergence sous-linéaire.

De la même manière, on obtient pour la seconde suite une estimation du reste en

$$\int_n^{\infty} x e^{-x^2} dx \sim \frac{1}{2} e^{-n^2}$$

qui fournit une convergence sur-linéaire, mais sous-quadratique.

Correction 1.8. *Avec la formule du binôme, on a*

$$(1+\sqrt{2})^n + (1-\sqrt{2})^n = \sum_{k=0}^{n} \binom{N}{k} \left(\sqrt{2}^k + (-\sqrt{2})^k\right).$$

Dans chaque parenthèse, on obtient 0 *lorsque* k *est impair, et* $2 \cdot 2^{k/2} \in 2\mathbb{N}$ *si* k *est pair, de sorte que*

$$(1+\sqrt{2})^n + (1-\sqrt{2})^n \in 2\mathbb{N}.$$

On a donc $\pi(1+\sqrt{2})^n = -\pi(1-\sqrt{2})^n\ [2\pi]$, *puis* $\sin(\pi(1+\sqrt{2})^n) = \sin(-\pi(1-\sqrt{2})^n)$.

Puisque $0 < \sqrt{2}-1 < 1$, *on a* $(\sqrt{2}-1)^n \to 0$. *De plus, comme* $\sin(x) \sim x$, *on obtient* $|u_n| \sim \pi(\sqrt{2}-1)^n$. *La suite* u_n *converge vers* 0 *linéairement à taux* $\sqrt{2}-1 \approx 0.41$.

Correction 1.9. *(1) C'est la continuité de* f'.

(2) Montrons par récurrence que $x_n \in B(x^*, \varepsilon)$ *pour tout* $n \in \mathbb{N}$. *En effet,* $x_0 \in B(x^*, \varepsilon)$, *et si* $x_n \in B(x^*, \varepsilon)$, *on a*

$$\begin{aligned}
|x_{n+1} - x^*| &= |f(x_n) - f(x^*)| \\
&= \left| \int_{[x^*, x_n]} f'(t)dt \right| \\
&\leq \int_{[x^*, x_n]} |f'(t)|\, dt \\
&\leq \alpha \int_{[x^*, x_n]} dt \\
&\leq \varepsilon\, |x_n - x^*| \\
&\leq \alpha\varepsilon < \varepsilon,
\end{aligned}$$

ce qui prouve que $x_n \in B(x^*, \varepsilon)$. *De plus, on a aussi montré que* $|x_{n+1} - x^*| \leq \alpha|x_n - x^*| \leq \cdots \leq \alpha^{n+1}|x_0 - x^*|$, *donc la suite* (x_n) *converge linéairement vers* x^* *à taux au moins* α.

(3) On écrit le développement de Taylor à l'ordre 2 *: pour tout* $x \in B(x^*, \varepsilon)$,

$$f(x) = f(x^*) + f'(x^*) + |x - x^*|^2 \int_0^1 f''(x^* + t(x - x^*))(1-t)dt,$$

puis

$$|f(x) - f(x^*)| \leq |x - x^*|^2 \left(\frac{1}{2} \sup_{B(x^*, \varepsilon)} |f''| \right).$$

On pose $C = \frac{1}{2} \sup_{B(x^*, \varepsilon)} |f''|$, *et on obtient*

$$\begin{aligned}
|x_{n+1} - x^*| = |f(x_n) - f(x^*)| &\leq C|x_n - x^*|^2 \\
&\leq \cdots \leq C^{1+2+4+\cdots 2^n} |x_0 - x^*|^{2^{n+1}} \\
&= \frac{1}{C} \left[C|x_0 - x^*|\right]^{2^{n+1}}.
\end{aligned}$$

On a donc convergence quadratique dès que $C|x_0 - x^*| < 1$.

Feuille de TP n°1 : Manipulations de base

L'objectif de cette première feuille de TP est de se familiariser avec certaines fonctions de Python. Nous n'en rappelons que quelques unes.

1.7. Quelques rappels.

(1) Nous utiliserons fréquemment les bibliothèques `numpy` et `matplotlib`. Il est nécessaire de les charger avant le début de tout TP.

(2) Il est toujours mieux d'essayer de mettre en forme son code, n'hésitez-pas à utiliser la commande # pour le commenter. Par ailleurs, quand vous affichez un résultat, faites-le si possible précéder d'un petit élément explicatif dans la commande `print` ; cela se fait via la commande `print('blahblah est égal à',XXX)` où XXX désigne l'objet à imprimer.

(3) La librairie `numpy` traite essentiellement de tableaux (`arrays`), de matrices et d'algèbre linéaire. Les tableaux se définissent par la commande `numpy.array`. Si `A` est un tableau qui ne contient qu'une ligne, on peut obtenir la dimension de `A` en utilisant la commande `A.ndim`. Si `A` est un tableau contenant n lignes et m colonnes, on peut obtenir les dimensions de `A` en utilisant la commande `A.shape`. La commande `A[i,j]` permet d'accéder au (i, j)-ième élement du tableau. On peut également accéder à une ligne entière d'indice i en utilisant `A[i,:]` (et de même pour la colonne).

Exercice 1.10. Appeler le tableau $[1, 2, 3, \pi]$, l'afficher, ainsi que sa longueur.

Appeler le tableau $M = \begin{pmatrix} 1 & 2 & 3 & 5 \\ 8 & 9 & 10 & 0 \end{pmatrix}$. Afficher sa dimension, puis l'élément d'indice $(2, 2)$ (attention à l'indiçage de Python, la numérotation y commence à 0). Afficher sa deuxième colonne.

(4) Une première commande de base utile est `range`. La commande `range(N)`, pour un entier N, renvoie la liste $[0, \dots, N-1]$.

Exercice 1.11. Essayer d'afficher la liste des entiers de 1 à 10 en utilisant uniquement les commandes `range` et `print` ; qu'observez-vous ?

Pour pallier cette difficulté, il est nécessaire de transformer la liste en `array`. Ceci peut se faire par l'*unpacking operator* : si ℓ est une liste, ℓ transforme cette liste en array, qu'il est désormais possible d'imprimer.

Exercice 1.12. Afficher la liste des entiers de 1 à 10 en utilisant `range`, l'unpacking operator et `print`.

(5) Ajouter des éléments à un array est important. On utilisera à cet effect la fonction `append`, qui dans notre cas s'écrira `append`(array, élément à ajouter).

Exercice 1.13. Générer un array vide via la commande `np.array`. Lui ajouter l'élement 1, et imprimer le resultat. Lui ajouter l'élement π et imprimer le résultat.

(6) Les constructions itératives sont également cruciales.

Exercice 1.14. Écrire un petit code utilisant une boucle `for` prenant en entrée un entier N et affichant en sortie le tableau des N premiers entiers.

(7) En particulier, nous aurons à utiliser ces constructions itératives quand nous travaillerons sur des suites construites par récurrence.

Exercice 1.15. À l'aide d'une boucle `for`, écrire un programme qui prenne en entrée un réel x_0, un entier n et qui renvoie à la fois le n-ième terme et la liste des n premiers éléments de la suite définie par récurrence par

$$x_{n+1} = \frac{x_n}{2} + 1.$$

1.8. **Quelques révisions d'algèbre linéaire.**

Exercice 1.16. On considère la matrice $A \in M_{34}\mathbb{R}$ définie par :

$$\begin{pmatrix} 4 & 6 & -2 & 3 \\ 2 & -1 & 0 & 1 \\ -7 & 0 & 1 & 12 \end{pmatrix}.$$

(1) Définir la matrice A comme un `np.array()`.

(2) Imprimer la première ligne et la deuxième colonne de la matrice A.

(3) Modifier la matrice A pour que ses deux premières lignes soient multipliées par 2 et que sa dernière colonne soit divisée par 3. On réalisera ces opérations dans l'ordre indiqué.

(4) Créer une nouvelle matrice B définie par

$$\begin{pmatrix} 4 & 5 & 6 \\ 5 & 10 & 15 \\ 1 & 1 & 1 \end{pmatrix},$$

(5) On reprend la matrice A de la première question (non modifiée). Créer la matrice $C \in M_{33}\mathbb{R}$ extraite de A telle que pour $1 \leq i, j \leq 3$, $c_{ij} = a_{ij}$.

(6) Différents produits matriciels

— Réaliser le produit matriciel D de B et A (`np.dot()`).

— Réaliser le produit d'Hadamard E de B et de C.

*Pour mémoire, le produit d'*Hadamard $E \in M_{33}\mathbb{R}$ *des matrices* $B \in M_{33}\mathbb{R}$ *et* $C \in M_{33}\mathbb{R}$ *est défini par*

$$\forall 1 \leq i, j \leq 3, \quad e_{ij} = c_{ij} b_{ij}.$$

(7) Écrire un petit code pour calculer la somme des éléments de la matrice E et le vecteur colonne $Y \in \mathbb{R}^3$ tel que pour $1 \leq i \leq 3$, $y_i = \sum_{j=1}^{4} d_{ij}$.

1.9. **Tracer des graphiques.** Pour tracer des graphiques ou des courbes paramétrées, la commande de base est `plot(x,y)`. Dans cette expression, `x` et `y` sont des listes (`array`) de même taile, qui peuvent être générées ou déclarées. À titre d'exemple, on pourra utiliser `linspace(a,b,N)` pour représenter en une liste l'intervalle (a, b) avec N points de discrétisation, ici uniforme.

Pour la mise en page, on utilisera bien sûr les fonctions `title, axis, legend, x/ylabel`. On commence par définir la fonction f dont on veut tracer le graphique, avant de définir les discrétisations de ensembles de départ. Ainsi, le code typique prendra la forme

```
1
2  def f(x) : return .... # La fonction f
3
4  xx=linspace(a,b,N) # N points de discr\'etisation entre a et
       b
5  plot(xx, f(xx),'color') # 'color' est optionnel et sert \'a
       choisir la couleur de la courbe
6
7  axis('equal') # pour avoir la même échelle dans les deux axes
8  title("le graphe de la fonction $f$") # N'h\'esitez pas \'a
       \'ecrire en Latex
9  legend("f")
10 xlabel("axe des abscisses")
11 ylabel("axe des ordonn\'ees")
```

Exercice 1.17. (1) Tracer le graphe de la fonction $f : x \mapsto x^2 \cos(10 * x)$ pour $x \in (-\pi; \pi)$ en vert.

(2) Tracer sur le même graphe les fonctions $f_n : x \mapsto x^2 \cos(n^n x)$ pour $x \in (-\pi, \pi)$ et $n = 0, 1, \ldots, 10$. Pour cela, on notera que, si l'on crée un graphe via `plot(xx,f(xx),...)`, si l'on rappelle ensuite la même fonction via `plt.plot(xx,g(xx),....)`, on peut dessiner les deux graphes sur la même figure.

On peut bien sûr également tracer des courbes paramétrées en utilisant les mêmes outils.

Exercice 1.18 (Courbes de Lissajous). Soient (a, b, c, d) quatre paramètres réels. On considère la courbe paramétrée $z_{a,b,c,d}(t) := (x_{a,b,c,d}(t), y_{a,b,c,d}(t))$ définie par

$$x(t) = a\cos(bt)\,, y(t) = c\sin(dt).$$

Écrire un petit programme qui, prenant a, b, c, d en arguments renvoie le graphe de la courbe paramétrée.

Enfin, la fonction `contour` est très efficace pour tracer des lignes de niveaux d'une fonction de plusieurs variables (la maîtrise géométrique des lignes de niveau est essentielle pour les méthodes de type descente de gradient). Pour cela, on considère une fonction de deux variables f, et l'on génère, à partir des ensembles de définition des deux variables, une liste contenant toutes les valeurs de f. On peut le faire de la manière suivante : une fois f, ainsi que les discrétisations de x et y, définies, on peut poser

```
1  z=[[f(x,y) for x in ... ] for y in ...]
```

La fonction `Contour(domaine de x,domaine de y,Z,N)` affiche N lignes de niveaux de f. On peut rajouter la commande `colorbar()` pour obtenir une échelle de couleurs.

Exercice 1.19. Tracer vingt ensemble de niveaux de la fonction

$$f : (x, y) \mapsto e^{-x^2} \sin(\pi x - y)$$

pour $(x, y) \in (-4, 4)^2$. Amusez-vous à changer les couleurs en essayant les options `cmap='inferno',cmap='plasma'`...

1.10. **Ordre de convergence des suites.** On renvoie au cours pour les définitions des taux de convergence. Rappelons néanmoins que la manière de procéder est en général la suivante :

(1) Puisqu'il faut utiliser la valeur limite, on peut soit la calculer théoriquement, soit itérer la suite un grand nombre N de fois, et utiliser x_N comme "presque"-point limite x^*.

(2) On utilise ensuite la fonction `semilogy` de Python pour tracer le graphe de $\|x_n - x*\|$.

Exercice 1.20. On considère les suites récurrentes suivantes :

(1) $x_0 = 1$ et, pour tout $k \in \mathbb{N}$,
$$x_{k+1} := \frac{x_k}{2} + \frac{1}{x_k}.$$

(2) $x_0 = 1$ et pour tout $n \in \mathbb{N}$
$$x_{n+1} = x_n \left(1 - \frac{x_n}{2}\right).$$

Pour ces deux suites, calculer numériquement les N premiers termes de la suite et déterminer, graphiquement, l'ordre de convergence de ces suites (on pourra sinon simplement écrire un code prenant en argument une fonction et une condition initiale, puis renvoyant une illustration graphique, à la fois du calcul des termes de la suite et des ordres de convergence). Quelles convergences sont linéaires ?

2. Premières méthodes en dimension 1 : dichotomies, section dorée, Newton, sécante

Dans ce cours, nous présentons trois méthodes importantes dans le cas unidimensionnel : la méthode de dichotomie, la méthode de la section dorée, et nous faisons une première incursion vers la méthode de Newton. Dans ces trois méthodes, le cadre est le suivant : on travaille avec une fonction $f \in \mathscr{C}^0(\mathbb{R}, \mathbb{R})$, et l'on veut résoudre

$$f(x) = 0. \tag{2.1}$$

En particulier, toutes les méthodes prsentées ici sont unidimensionnelles.

2.1. Méthode de dichotomie (méthode d'ordre 0).

On commence par la méthode la plus classique, la *méthode de dichotomie*. On considère l'équation suivante : f étant une fonction continue, $[a, b] \subset \mathbb{R}$ étant un intervalle réel, déterminer $x^* \in [a, b]$ tel que

$$f(x^*) = 0. \tag{2.2}$$

Pour avoir une chance que cette équation ait une solution, on suppose également que

$$f(a) < 0\,, f(b) > 0$$

ou, plus généralement, que $f(a)f(b) < 0$. Dans ces conditions, le théorème des valeurs intermédiaires garantit l'existence d'une solution x^*.

Remarque 2.1. On n'aborde pas ici de questions d'unicité, mais celle-ci est évidente si la fonction f est strictement monotone.

La **méthode de dichotomie**, aussi appelée, **méthode de bissection**, consiste, à chaque étape, à réduire l'intervalle de manière assez intuitive. Notons d'abord que, puisque l'on renvoie un intervalle, on a en fait besoin de définir trois suites : d'abord, $\{y_k^+\}_{k\in\mathbb{N}}, \{y_k^-\}_{k\in\mathbb{N}}$ pour les bornes de l'intervalle, puis y_k comme point milieu de l'intervale. La procédure est la suivante :

(1) On initialise à $y_0^- = a\,, y_0^+ = b$. On définit $y_0 := \frac{1}{2}(y_0^- + y_0^+)$.

(2) Si $f(y_0) \geq 0$, on pose $y_1^- = y_0^-$ et $y_1^+ = y_0$. Sinon, on pose $y_1^- = y_0$ et $y_1^+ = y_0^+$.

(3) Ensuite, pour tout $k \in \mathbb{N}$, on définit

$$\begin{cases} y_{k+1}^- := y_k^-\,, y_{k+1}^+ := y_k \text{ si } f(y_k) > 0\,, \\ y_{k+1}^- := y_k,, y_{k+1}^+ := y_k^+ \text{ si } f(y_k) < 0\,, \\ y_{k+1}^\pm = y_k \text{ si } f(y_k) = 0, \\ y_{k+1} := \frac{y_{k+1}^+ + y_{k+1}^-}{2}. \end{cases}$$

On peut démontrer assez aisément que cet algorithme converge linéairement à taux $\alpha = \frac{1}{2}$:

Lemme 2.1 (Convergence de la méthode par dichotomie)**.** *Soit $f \in \mathscr{C}^0([a, b], \mathbb{R})$, $f(a) < 0$ et $f(b) > 0$. La méthode de dichotomie converge linéairement, à taux $\alpha = \frac{1}{2}$, vers une solution de (2.2).*

Preuve du Lemme 2.1. On suppose que la suite générée n'est pas stationnaire. On définit pour tout $k \in \mathbb{N}$ l'intervalle $I_k := [x_k^-; x_k^+]$. On note que, d'une part, cette suite est décroissante pour l'inclusion et que, d'autre part,

$$\mathrm{Vol}(I_{k+1}) = \frac{1}{2}\mathrm{Vol}(I_k).$$

Par le théorème des segments emboîtés (ici, simplement des suites adjacentes) on a

$$\cap_{k\in\mathbb{N}} I_k = \{x^*\}$$

pour un certain point x^*. En passant à la limite dans $f(x_k^-) < 0\,, f(x_k^+) \geq 0$ on obtient

$$f(x^*) = 0.$$

Par ailleurs, on sait que

$$\forall k \in \mathbb{N}\,, y_k\,, x^* \in I_k \Rightarrow |x^* - y_k| \leq \mathrm{Vol}(I_k) \leq \frac{1}{2^k}\,|b-a|\,.$$

Montrons que le taux $\alpha = \frac{1}{2}$ est optimal quand la suite n'est pas stationnaire : supposons par l'absurde qu'il existe un réel $\alpha \in (0, \frac{1}{2})$ et une constante C tels que

$$|y_k - x^*| \leq C\alpha^k.$$

On a alors :

$$|y_k - y_{k+1}| \leq C\alpha^k + C\alpha^{k+1} \leq 2C\alpha^k.$$

Néanmoins, par construction, on a également, si la suite n'est pas stationnaire,

$$|y_{k+1} - y_k| = \frac{1}{2^k}|b-a|.$$

C'est une contradiction. □

En TD et en TP, nous verrons comment coder cette méthode, et illustrer ses ordres de convergence.

2.1.1. *Résumé de l'algorithme.* En TP, nous coderons l'algorithme en suivant simplement les grandes lignes suivantes :

(1) Initialisation de l'algorithme en choisissant un couple (a, b). On vérifie que l'on est dans les conditions du théorème $(f(a)f(b) < 0)$

(2) On calcule le point milieu c, et on choisit le nouvel intervalle grâce à la règle de dichotomie.

(3) On intégrera ou bien un critère de tolérance qui correspond à la taille de l'intervalle, ou bien un nombre maximal d'itérations.

2.2. **Minimisation par Fibonacci (méthode d'ordre 0).**

2.2.1. *Premiers pas dans la méthode.* On continue cette brève présentation des méthodes unidimensionnelles par une explication de la méthode de Fibonacci, également appelée **méthode de la section dorée**.

Cette méthode permet de trouver les minima d'une fonction réellé f, à condition qu'on la suppose *unimodale.* On rappelle qu'une fonction $f : [a, b] \to \mathbb{R}$ est dite unimodale s'il existe un point $c \in (a, b)$ tel que f est strictement décroissante sur $[a, c]$, et strictement croissante sur $[c, b]$.

Bien sûr, si l'on a une fonction que l'on sait être unimodale, on peut, pour approcher un minimum, se contenter d'appliquer une méthode de dichotomie à la fonction f'. Néanmoins, dans la pratique, il se peut que f' soit difficile à déterminer, ou

même que f' (qui existe presque partout dans le cas des fonctions unimodales) soit discontinue. La méthode de la section dorée permet de contourner cette difficulté.

Donc, nous nous intéressons au problème de minimisation

$$\text{Trouver } x^* \in [a,b] \text{ tel que } f(x^*) = \min_{[a,b]} f. \tag{2.3}$$

Un premier élément à souligner est que, comme pour la méthode par dichotomie, l'algorithme repose sur la construction de segments emboîtés, *via* la construction de suites adjacentes. Mais là où la dichotomie suivait le schéma : on prend un couple de points, on construit (bien) un troisième point, et on renvoie un nouveau couple de points, la méthode de la section dorée, elle, travaille sur des *triplets* de points.

On se donne donc une fonction f unimodale sur un intervalle $[a;b]$.

Définition 2.1 (Triplet admissible)**.** Pour tout $c \in [a;b]$, on dit que le triplet (a,c,b) est *admissible* si

$$f(c) < \min\left(f(a), f(b)\right).$$

On initialise l'algorithme en un triplet admissible (a_0, c_0, b_0). On note qu'une première étape peut, en géneral, être de trouver un tel triplet admissible initial.

Pour itérer, on doit construire un quatrième point $d \in (a_0;b_0)$, de manière à obtenir un nouveau triplet admissible (a_1, c_1, d_1). Dès que l'on se fixe un point $d \in (a,b)\backslash\{c\}$ on remplace (a,c,b) par

(1) (a,d,c) si $d < c$ et si $f(d) \leq f(c)$,

(2) (d,c,b) si $d < c$ et si $f(c) \leq f(d)$,

(3) (a,c,d) si $c < d$ et si $f(c) \leq f(d)$,

(4) (c,d,b) si $c < d$ et si $f(d) \leq f(c)$.

Le nouveau triplet obtenu est noté (a_1, c_1, d_1) et on itère la construction. Il est facile de voir que, si les suites $\{a_k\}_{k\in\mathbb{N}}$ et $\{b_k\}_{k\in\mathbb{N}}$ sont adjacentes, alors elles convergent vers un point c^* qui, si la fonction est unimodale, est un minimiseur.

2.2.2. *Description un peu plus précise de l'algorithme.* Expliquons maintenant comment construire ces triplets : pour cela, on part de deux points (a_0, b_0). On va construire un troisième et un quatrième points, c_0 et d_0 dans (a_0, b_0), et appliquer au quadruplet (a_0, c_0, d_0, b_0) la distinction de cas expliquée à la section précédente. Quel est l'objectif ? On aimerait qu'indépendamment du résultat de notre distinction de cas, et à chaque itération, on divise par un facteur constant la taille de l'intervalle (a_{k+1}, b_{k+1}) ainsi obtenu. En d'autre termes, partant de l'intervalle (a_k, b_k), on veut pouvoir satisfaire la relation

$$b_{k+1} - a_{k+1} = \alpha(b_k - a_k) \tag{2.4}$$

pour un facteur $\alpha \in (0;1)$ constant. Comment construire c_k et d_k pour garantir cette condition ? Déjà, notons que, quitte à remplacer c_k par d_k et inversement, on peut supposer que

$$c_k < d_k.$$

Ainsi, si on applique notre distinction de cas, on a

$$\text{soit } (a_{k+1}, b_{k+1}) = (a_k, d_k) \text{ ou } (a_{k+1}, b_{k+1}) = (c_k, b_k).$$

Il est donc naturel de chercher à obtenir

$$d_k - a_k = b_k - c_k = \alpha\left(b_k - a_k\right),$$

ce qui donne la règle suivante pour le choix de c_k et d_k :

$$d_k = (1-\alpha)a_k + \alpha b_k\,, c_k = (1-\alpha)b_k + \alpha a_k.$$

On voit ici que pour que la condition $c_k < d_k$ soit satisfaite on doit avoir $\alpha > \frac{1}{2}$.

Par ailleurs, on veut qu'à l'étape suivante, pour réduire les coûts de calcul, on puisse réutiliser le point intermédiaire. En d'autres termes on veut que, si le nouvel intervalle est, par exemple $(c_k; b_k)$, alors d_k soit égal à c_{k+1}. Ceci implique que le paramètre α est déterminé, qui plus est de manière unique. En effet, les calculs étant symétriques si l'on a $(a_{k+}, b_{k+1}) = (a_k; dk)$, il faut que soit satisfaite la relation

$$d_k = c_{k+1} = \alpha c_k + (1-\alpha)b_k = \alpha^2 a_k + (\alpha(1-\alpha) + \alpha)\, b_k.$$

En d'autres termes il faut, pour que ces deux relations soient compatibles, que l'on ait

$$(1-\alpha) = \alpha^2 \text{ et } 1 - \alpha^2 = \alpha.$$

Ainsi, α est une racine (positive) de $\alpha^2 + \alpha - 1 = 0$ qui s'intègre explicitement en

$$\alpha = \frac{\sqrt{5}-1}{2}.$$

On remarque, et c'est l'origine du nom de la méthode, que α est l'inverse du nombre d'or $\phi = \frac{1+\sqrt{5}}{2}$.

Remarque 2.2 (Illustration de la vitesse de convergence)**.** En TP, nous verrons que cette méthode converge assez lentement, linéairement et à taux $\alpha \approx 0.6$, ce qui est plus lent, par exemple, que la méthode de dichotomie. Néanmoins, c'est une méthode plus robuste, et qui ne suppose pas de calculer la dérivée de la fonction.

Remarque 2.3. La convergence de la méthode de la section dorée est immédiate, par construction même.

2.2.3. *Résumé de l'algorithme de la section dorée.* En TP, nous coderons l'algorithme en suivant simplement les grandes lignes suivantes :

(1) Définition du nombre d'or `alpha`.

(2) Initialisation de l'algorithme en choisissant un couple (a, b). On choisit les deux points c et d par la méthode exposée ci-dessus.

(3) On applique la disjonction de cas pour déterminer un nouveau couple (on n'oubliera pas de tester l'admissibilité du triplet ainsi obtenu).

(4) On intégrera ou bien un critère de tolérance qui correspond à la taille de l'intervalle, ou bien un nombre maximal d'itérations.

2.3. **Méthode de Newton en dimension 1 (méthode d'ordre 1).** On passe au premier algorithme d'ordre 1, c'est-à-dire qui fasse intervenir la dérivée. La convergence de ces algorithmes est à double tranchant : alors que les algorithmes de type dichotomie ou section dorée converge toujours, dans le cas des algorithmes d'ordre 1, *l'initialisation est cruciale* : si on la choisit mal, l'algorithme ne converge pas. En revanche, *si l'initialisation est bien choisie, l'algorithme converge de manière spectaculaire.*

Cet algorithme est fondamental, d'une part par le rôle qu'il a joué dans le développement historique de la théorie de l'approximation, d'autre part parce que son étude constitue une forme de propédeutique à des méthodes plus évoluées, comme celles de descentes de gradient.

Cette méthode est utilisée pour résoudre des équations de la forme

$$f(x) = 0. \tag{2.5}$$

Ici, f est une fonction dérivable. Même s'il est fort probable que vous ayiez déjà vu cette méthode, nous en rappelons les grandes étapes.

2.3.1. *Description heuristique de la méthode.* L'idée, c'est que si l'on se place au voisinage d'un point x_0, on peut approcher la fonction f par sa tangente :

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \underset{x_0 \to x}{o}(|x - x_0|).$$

On définit la partie linéaire de f à x_0 par

$$R_f(x_0, \cdot) : x \mapsto f(x_0) + f'(x_0)(x - x_0).$$

On sait que, dans un petit voisinage de x_0, $R_f(x_0, \cdot) \approx f$, de sorte que résoudre $f(x) = 0$ revient peu ou prou à résoudre $R_f(x_0, x) = 0$. Néanmoins, $R_f(x_0, \cdot)$ étant une fonction linéaire, son unique racine est donnée par

$$x_0^* = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Si on suppose que f' est de signe constant sur l'intervalle où l'on travaille, on s'aperçoit immédiatement qu'alors $f(x_0^*) < f(x_0)$ si $f(x_0) > 0$, et $f'(x_0) < f'(x_0^*) < 0$ si $f(x_0) < 0$. Donc, par cette simple résolution, on réussit à améliorer notre critère.

Remarque 2.4. Simplement à partir de la construction des itérations de la méthode de Newton, on voit que "plus la fonction est linéaire", plus la méthode de Newton risque d'être précise.

Ceci suggère la construction suivante pour l'algorithme de Newton : on initialise en un point x_0 (dont nous verrons comment il doit être choisi) et, pour passer de x_k à x_{k+1} on définit simplement

$$x_{k+1} := x_k - \frac{f(x_k)}{f'(x_k)}.$$

2.3.2. *Quelle vitesse de convergence pour cette méthode ?* Avant de nous lancer dans l'étude de la convergence dans le cas général, faisons deux petites remarques destinées à souligner les différents types de régime auxquels la méthode de Newton peut mener. Dans un premier temps, observons que si l'on travaille avec une fonction linéaire de la forme $f(x) = ax + b$ cette méthode converge en une unique itération. En effet, quel que soit x_0, on a

$$x_1 = x_0 - \frac{ax_0 + b}{a} = -\frac{b}{a}.$$

Dans un second temps, remarquons la convergence peut, à l'inverse, être très lente : par exemple, même avec une bête fonction quadratique de la forme $f(x) = x^2$ on s'aperçoit que

$$x_{k+1} = \frac{1}{2}x_k = 2^{-k-1}x_0$$

et que donc la méthode, si elle converge bien, ne converge que linéairement.

2.3.3. *Convergence de la méthode.* En général, la méthode de Newton converge extrêmement rapidement, à condition que l'initialisation soit bien choisie.

Théorème 2.1 (Convergence quadratique de la méthode de Newton)**.** *Soit* $f \in \mathscr{C}^3(\mathbb{R};\mathbb{R})$ *et* $x^* \in \mathbb{R}$ *tel que :*

$$f(x^*) = 0\,, |f'(x^*)| > 0.$$

Il existe $\varepsilon > 0$ *tel que, pour toute initialisation* $x_0 \in (x^* - \varepsilon; x^* + \varepsilon)$*, la méthode de Newton initialisée en* x_0 *converge quadratiquement vers* x^**.*

Preuve du Théorème 2.1. Quitte à remplacer f par $-f$ on peut supposer que

$$f'(x^*) > 0.$$

On peut distinguer deux preuves : une qui repose sur les théorèmes de points fixes, mais qui ne donnera pas tellement d'information précise sur le type d'initialisation à choisir, l'autre qui est plus directe, repose simplement sur la formule de Taylor-Lagrange et permet de quantifier. Donnons la première preuve : f étant de classe $\mathscr{C}^3$, on fixe $\varepsilon_0 > 0$ tel que

$$\inf_{[x^*-\varepsilon_0;x^*+\varepsilon_0]} f'(x^*).$$

On choisit $x_0 \in [x^* - \varepsilon_0; x^* + \varepsilon_0]$.

La preuve se ramène à l'étude d'un problème de point fixe : en définissant

$$g(x) := x - \frac{f(x)}{f'(x)}$$

on a immédiatement que $f(y) = 0$ si et seulement si $g(y) = y$. En particulier, il nous suffit de démontrer que la suite récurrente

$$x_{k+1} = g(x_k)$$

initialisée à x_0 converge quadratiquement vers x^*. Ces suites ont été étudiées lors du TD (TD n°1, Exercice 1.8). Il faut vérifier que

$$|g'(x^*)| < 1.$$

Or,

$$g'(x^*) = 1 + \frac{f(x^*)f''(x^*)}{\left(f'(x^*)\right)^2} - 1 = 0 < 1$$

et g est de classe $\mathscr{C}^2$. En particulier, la suite converge quadratiquement (au moins) vers x^*.

Maintenant, donnons l'autre preuve de la convergence quadratique, qui donne une distance maximale d'où initialiser l'algorithme. Cette second preuve repose uniquement sur la formule de Taylor-Lagrange : l'on sait que

$$0 = f(x^*) = f(x_0) + (x^* - x_0)f'(x_0) + f''(\xi)\frac{(x^* - x_0)^2}{2}$$

pour un certain $\xi \in (x^*, x_0)$. Noter bien le fait important suivant : pour démontrer la convergence de l'algorithme de Newton, on doit faire un développement de Taylor **autour du point d'initialisation** x_0, ce qui est cohérent avec l'idée sous-jacente de la méthode.

Ainsi on obtient en posant $x_1 := x_0 - (f'(x_0)^{-1})f(x_0)$,

$$\begin{aligned} 0 &= -f'(x_0)\left\{x_0 - \frac{f(x_0)}{f'(x_0)}\right\} + x^* f'(x_0) + \frac{1}{2}f''(\xi)\,(x^* - x_0)^2 \\ &= f'(x_0)\,(x^* - x_1) + \frac{1}{2}f''(\xi)\,(x^* - x_0)^2 \end{aligned}$$

de sorte que

$$|x^* - x_1| \leq \frac{\|f''\|_{L^\infty(I)}}{2\inf_I |f'|}|x^* - x_0|^2,$$

où I est n'importe quel intervalle compact contenant x^* et x_0. Ainsi, par le critère de convergence quadratique (Proposition 1.2) on en déduit que la suite des itérées converge quadratiquement si x_0 est choisi de sorte que

$$|x^* - x_0| < \frac{2\inf_I |f'|}{\|f''\|_{L^\infty(I)}}.$$

□

En particulier, par la méthode de Newton, on double le nombre de décimales à chaque itération ; en moyenne, N itérations permettent d'obtenir 2^N décimales, ce qui est prodigieusement rapide.

Insistons une nouvelle fois sur quelques considérations autour de la vitesse de convergence :

(1) Il va de soit que l'on ne peut en général pas démontrer que le taux de convergence est au plus quadratique. Pour s'en convaincre, il suffit de considérer la méthode de Newton appliquée à la fonction $f : x \mapsto x$. Quelle que soit l'initialisation x_0 on a

$$x_1 = x_0 - x_0 = 0$$

et on converge donc en une étape (ce qui est en particulier plus rapide que n'importe quel ordre de convergence).

(2) L'hypothèse $f'(x^*) \neq 0$ est *nécessaire* pour obtenir la convergence quadratique ; à titre d'exemple, la méthode de Newton appliquée à la fonction $f : x \mapsto x^2$ donne, quelle que soit l'initialisation x_0,

$$x_k = \left(\frac{1}{2}\right)^k x_0$$

et l'on obtient donc cette fois une convergence linéaire.

(3) Mais ce n'est pas le pire que d'avoir une convergence lin´eaire. Il se peut même que la méthode diverge : prenons l'exemple

$$f : x \mapsto |x|^{\frac{1}{3}}$$

et initialisons notre méthode en n'importe quel point $x_0 > 0$. On obtient

$$x_1 = x_0 - 3\frac{x_0^{\frac{1}{3}}}{x_0^{-\frac{2}{3}}} = -2x_0.$$

On voit, d'une part, qu'on passe dans le demi-plan des x négatifs et d'autre part qu'en itérant la construction on crée une suite divergente. On verra en TD une généralisation de ce phénomène.

2.3.4. *Résumé de la méthode.* Toutes les formules étant explicites, il ne reste qu'à choisir un critère d'arrêt ; on pourra soit prendre un nombre d'itérations arbitrairement grand, soit utiliser, comme critère de tolérance, la valeur de $|f(x_k)|$, et inclure un nombre maximal d'itérations à ne pas dépasser. On observera, numériquement, que la convergence est effectivement sur-linéaire.

2.3.5. *Derniers commentaires.*

(1) la condition $f'(x^*) \neq 0$ est essentielle. En fait, plus la fonction f est localement linéaire, plus la méthode converge rapidement.

(2) Si on pense désormais en termes d'optimisation, et que l'on cherche à déterminer le minimum d'une fonction $F : \mathbb{R} \to \mathbb{R}$, une possibilité est d'appliquer la méthode de Newton à la fonction $f = F'$. La condition $f'(x^*) = 0$ devient alors $F''(x^*) \neq 0$. En d'autres termes, c'est une méthode très pratique si l'on cherche des optima non-dégénérés. Ce type de non-dégénerescence sera également crucial quand nous verrons des algorithmes de **descente de gradients**.

2.4. **Méthode de la sécante (méthode d'ordre 0).**

2.4.1. *Heuristique et ordre de convergence.* Quel est l'inconvénient de la méthode de Newton, aussi rapide sa convergence soit-elle ? C'est qu'elle fait appel non seulement à la fonction f, mais également à sa dérivée. La méthode des sécantes, à l'inverse, converge un peu moins bien que quadratiquement, mais n'a besoin de faire qu'un seul appel à la fonction f puisqu'au lieu de la dérivée de f, elle ne fait appel qu'à ses taux d'accroissement. En d'autres termes, le terme $f'(x_n)$ est simplement remplacé par le taux

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

Ainsi, les itérations successives de l'algorithme sont données par l'expression

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

La vraie différence est que, si nous n'utilisons pas la fonction f', nous devons choisir deux points d'initialisation, x_0 et x_1 pour définir x_2. Pour cette méthode, on a une convergence sous-quadratique, mais toujours sur-linéaire :

Proposition 2.1. *Si $f : \mathbb{R} \to \mathbb{R}$ est une fonction de classe C^3 avec $f(x^*) = 0$ et $f'(x^*) \neq 0$, et si $x_0, x_1 \in \mathbb{R}$ sont suffisamment proches de x^*, alors la suite $\{x_k\}_{k\in\mathbb{N}}$ définie par la méthode de la sécante converge vers x^* à l'ordre au moins $\varphi = \frac{1}{2}(1 + \sqrt{5}) \approx 1.618$ (le nombre d'or).*

Preuve de la Proposition 2.1. On ne donne la preuve que dans le cas où la fonction f est quadratique afin de ne pas alourdir la preuve. À la différence de la preuve de la convergence de la méthode de Newton, on fait cette fois un développement limité autour de x^*.

On définit donc le terme qu'il nous faut contrôler, à savoir

$$y_k := x_k - x^*.$$

On observe que, la fonction f étant quadratique, on a un développement de Taylor exact qui nous assure que

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

En particulier on a

$$f(x_k) = f'(x^*)y_k + \frac{1}{2}f''(x^*)y_k^2.$$

On peut injecter cette preuve dans la relation de récurrence qui définit la suite des itérations de la méthode de la sécante. Néanmoins, il va falloir être (un peu) plus astucieux que dans le cas de la méthode de Newton et travailler sur la suite des différences $(y_{k+1} - y_k)_{k\in\mathbb{N}}$. En effet, on se rend compte en développant les expressions données par l'algorithme que l'on obtient :

$$\begin{aligned} y_{k+1} - y_k &= x_{k+1} - x_k \\ &= -\frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} \\ &= -\frac{f(x_k)(y_k - y_{k-1})}{f'(x^*)(y_k - y_{k-1}) + \frac{1}{2}f''(x^*)(y_k^2 - y_{k-1}^2)} \\ &= -\frac{f'(x^*)y_k + \frac{1}{2}f''(x^*)y_k^2}{f'(x^*) + \frac{1}{2}f''(x^*)(y_k + y_{k-1})} \end{aligned}$$

d'où l'on déduit que

$$y_{k+1} = y_k - \frac{f(x_k)}{f'(x^*) + \frac{1}{2}f''(x^*)(y_k + y_{k-1})}.$$

Néanmoins, on peut utiliser le fait que

$$f(x_k) = f'(x^*)y_k + \frac{1}{2}f''(x^*)y_k^2$$

pour en déduire que

$$\begin{aligned} y_{k+1} = y_k - \frac{f'(x^*)y_k + \frac{1}{2}f''(x^*)y_k^2}{f'(x^*) + \frac{1}{2}f''(x^*)(y_k + y_{k-1})} = -y_k\frac{1}{2}\cdot\frac{f''(x)y_{k-1}}{f'(x^*) + \frac{1}{2}f''(x^*)(y_k + y_{k-1})} \\ = -y_k T(y_k, y_{k-1}) \end{aligned}$$

où la fonction T est définie par

$$T(x, y) := \frac{f''(x^*)y}{f'(x^*) + \frac{1}{2}f''(x^*)(x + y)}.$$

Soulignons cette expression : on sait désormais que

$$\boxed{\forall k \in \mathbb{N}\,, y_{k+1} = -y_k T(y_k, y_{k-1}).} \tag{2.6}$$

À partir de cette expression auxiliaire, nous allons déduire deux choses :

(1) D'abord, que la suite $\{y_k\}_{k\in\mathbb{N}}$ converge vers 0.

(2) Ensuite, qu'elle converge surlinéairement.

Commençons par le convergence vers 0. On observe que la fonction T est continue en (0,0) et que, si (x_0, x_1) sont choisis de sorte que $T(y_0, y_1) < \frac{1}{2}$ alors la suite $(y_k)_{k\in\mathbb{N}}$ est décroissante en norme et converge vers 0. Par ailleurs, ceci implique que

$$\frac{|y_{k+1}|}{|y_k y_{k-1}|} \underset{k\to\infty}{\to} C_0 := \frac{|f''(x^*)|}{2|f'(x^*)|} > 0$$

(si $f''(x^*) = 0$ on est linéaire, et l'on converge en une itération). En effet, on sait que

$$\left|\frac{y_{k+1}}{y_k y_{k-1}}\right| = \frac{1}{2} \cdot \left|\frac{f''(x^*)}{f'(x^*) + \frac{1}{2}f''(x^*)(y_k + y_{k-1})}\right| \underset{k\to\infty}{\to} \frac{1}{2} \cdot \left|\frac{f''(x^*)}{f'(x^*)}\right|.$$

Pour obtenir un taux de convergence précisé, passons au logarithme dans cette inégalité en posant $z_k := -\ln(y_k)$. Alors

$$z_{k+1} - z_k - z_{k-1} \underset{k\to\infty}{\to} \ln(C_0).$$

Ainsi, la suite $(z_{k+1} - z_k - z_{k-1})_{k\in\mathbb{N}}$ est bornée, disons par A. Soit A un majorant strictement positif de cette suite. Posons $w_k := z_k - A$. Alors

$$w_{k+1} - w_k - w_{k-1} = (z_{k+1} - z_k - z_{k-1}) - A \geq 0$$

et donc

$$w_{k+1} \geq w_k + w_{k-1}.$$

Soit $\varphi := \frac{1}{2}(1+\sqrt{5})$ le nombre d'or. Le nombre φ vérifie $\varphi > 1$ et $\varphi^2 = 1+\varphi$. La suite $(w_k)_{k\in\mathbb{N}}$ diverge vers $+\infty$, donc pour n_0 assez grand, on a $w_{n_0} \geq 1$ et $w_{n_0+1} \geq \varphi$. Par une récurrence immédiate, on en déduit que $w_{n_0+n} \geq \varphi^n$. Cela montre que $z_{n_0+n} \geq \varphi^n + A$, et enfin que $|y_{n_0+n}| \leq e^{-A}\alpha^{\varphi^{n_0+n}}$ avec $\alpha := e^{-\varphi^{n_0}} < 1$, ce qui conclut la preuve.

□

2.4.2. *Résumé de l'algorithme.* Les mêmes causes menant aux mêmes conséquence, on définit l'algorithme de la manière suivante :

(1) Notre algorithme prendra en argument une fonction f, ainsi que deux points d'initialisation x_0 et x_1, une tolérance et un nombre d'itérations maximales.

(2) À chaque étape, tant que l'on n'a pas dépassé le nombre maximal d'itérations autorisé, et tant que l'on est au dessus du seuil de tolérance, on itère la construction.

Feuille de TD n°2 : Suites, méthode de Newton

Exercice 2.1. Résoudre à l'aide des suites l'équation fonctionnelle

$$\forall x \in [0;1],\ f(2x - f(x)) = x,$$

où f est une bijection de $[0;1]$ dans $[0;1]$.

Exercice 2.2 (Nombre d'or).

(1) Soit $\{a_k\}_{k\in\mathbb{N}}$ la suite définie par $x_0 = 2$ et, pour tout entier naturel k,

$$a_{k+1} = 1 + \frac{1}{a_k}.$$

(a) Montrer que pour tout $k \geq 0$ on a $\frac{3}{2} \leq a_k \leq 2$.

(b) On pose $\phi := \frac{1+\sqrt{5}}{2}$. Montrer que $\{a_k\}_{k\in\mathbb{N}}$ converge linéairement vers ϕ à taux $\frac{4}{9}$.

(2) On définit sur $\left(\frac{1}{2}, +\infty\right)$ la fonction f par

$$f : x \mapsto \frac{x^2+1}{2x-1}.$$

(a) En étudiant les variations de f, montrer que $x > \frac{1}{2} \Rightarrow f(x) > \frac{1}{2}$ et en déduire que la suite

$$c_0 = 2\,, c_{k+1} = f(c_k)\,(k \in \mathbb{N})$$

est bien définie.

(b) Montrer que la suite $\{c_k\}_{k\in\mathbb{N}}$ converge quadratiquement vers le nombre d'or $\phi := \frac{1+\sqrt{5}}{2}$

Exercice 2.3. Soit $\Phi(x) := x + \sin(x)$. On pose $x_0 \in \mathbb{R}$ et $x_{n+1} = \Phi(x_n)$.

(1) Montrer que Φ est une fonction strictement croissante, et calculer ses points fixes.

(2) Soit $k \in \mathbb{N}$, et $x_0 \in (k\pi, (k+1)\pi)$. Montrer que pour tout $n \in \mathbb{N}$, on a $x_n \in (k\pi, (k+1)\pi)$, puis montrer que la suite (x_n) est croissante si k est pair, et décroissante si k est impair.

(3) Montrer que si $x_0 = 3$, alors la suite (x_n) converge vers π.

(4) Montrer que pour tout $x \in \mathbb{R}$, on a

$$|\Phi(x) - \Phi(\pi)| \leq \frac{1}{6} \max |\Phi'''| \cdot |x - \pi|^3.$$

(5) En déduire que si $x_0 = 3$, alors la suite (x_n) converge vers π à l'ordre 3.

Exercice 2.4. On veut résoudre l'équation

$$x = e^{-x}\,, x \in [0; +\infty).$$

On considère la méthode itérative

$$\begin{cases} \text{Initialisation à } x_0 \in [0; +\infty)\,, \\ x_{n+1} = e^{-x_n} \text{ pour tout } n \in \mathbb{N}. \end{cases}$$

Déterminer une condition sur x_0 pour que cette méthode soit convergente. En déterminer l'ordre de convergence.

Exercice 2.5 (Points fixes II : une méthode divergente). Montrer que l'équation $x = -\ln(x)$ ($x \in (0, +\infty)$) admet une unique solution. On considère, pour une initialisation $x_0 \in (0, +\infty)$ donnée, la suite définie par récurrence par

$$\forall k \in \mathbb{N}\,, x_{k+1} = -\ln(x_k).$$

Montrer que cette méthode itérative ne converge pas.

Exercice 2.6. Appliquer la méthode de Newton à la résolution de l'équation

$$x = e^{-x}.$$

Quel est l'ordre de convergence ?

Exercice 2.7. Plusieurs questions pour illustrer certains comportements de la méthode de Newton :

(1) Divergence Soit $\alpha > 0$. On considère la méthode de Newton appliquée à la fonction

$$f_\alpha : x \mapsto |x|^\alpha.$$

Démontrer que la méthode diverge (évidemment, quand on initialise hors de 0) quand $\alpha \in \left(0; \frac{1}{2}\right)$ et qu'elle converge pour $\alpha \in \left(\frac{1}{2}; 1\right)$.

(2) Cycles limites On considère la fonction

$$f : x \mapsto x^3 - 5x.$$

Montrer que la méthode de Newton initialisée en $x = 1$ est bloquée dans un cycle limite.

(3) Une méthode de Newton pour trouver i ? On considère

$$f : x \mapsto x^2 + 1.$$

f n'a aucune racine réelle mais on peut tout de même lui appliquer la méthode de Newton :

(a) On part de $x_0 \neq 0$. Déterminer l'expression de x_1.

(b) Démontrer que

$$\frac{\cos(2t)}{\sin(2t)} = \frac{1}{2}\left(\tan(t) - \frac{1}{\tan(t)}\right)$$

(c) Soit t_0 un réel tel que $x_0 = \cot(t_0)$. Justifier qu'un tel réel existe.

(d) Montrer que, pour tout $k \in \mathbb{N}$,

$$x_k = \cot(2^k t_0).$$

(e) Si on part de $t_0 = \frac{\pi}{4}$, que se passe-t-il au bout de deux itérations ?

(f) Généraliser ce phénomène : montrer que si $t_0 = \pi 2^{-K}$ pour un certain $K \in \mathbb{N}^*$, la suite n'est plus bien définie au bout de K itérations.

(g) Qu'observe-t-on si $t_0 = \frac{4}{3}\pi$?

(h) On suppose que $t_0/\pi \in \mathbb{R}\backslash\mathbb{Q}$. Montrer que la suite issue de $x_0 = \cot(t_0)$ n'est pas périodique.

Exercice 2.8 (Accélération d'Aitken). Soit $\Phi : \mathbb{R} \to \mathbb{R}$ de classe C^2 telle que $\Phi(0) = 0$, et $0 < \Phi'(0) < 1$. Soit α tel que $\Phi'(0) < \alpha < 1$.
a/ Montrer qu'il existe $\varepsilon > 0$ tel que, pour tout $x \in (-\varepsilon, \varepsilon)$, on a $|\Phi'(x)| < \alpha$.
b/ Montrer que la suite définie par $x_{n+1} = \Phi(x_n)$ avec $x_0 \in (-\varepsilon, \varepsilon)$ converge linéairement vers 0, à taux au moins α.
c/ Montrer que 0 est l'unique point fixe de Φ dans $(-\varepsilon, \varepsilon)$.

On pose maintenant

$$\Psi(x) := x - \frac{[\Phi(x) - x]^2}{x - 2\Phi(x) + \Phi(\Phi(x))}.$$

d/ Dans le cas où $\Phi(x) = \alpha x$, que vaut $\Psi(x)$?
e/ Dans le cas général, calculer le développement limité de Ψ à l'ordre 1. Montrer que $\Psi(0) = 0$ et $\Psi'(0) = 0$.
f/ On suppose Ψ de classe C^2, et $0 < \Psi''(0) < 1$. Montrer que la suite définie par $x_{n+1} = \Psi(x_n)$ avec x_0 suffisamment proche de 0 converge quadratiquement vers 0.

Correction de la feuille de TD n°2

Correction 2.1. *On remarque déjà par un argument de valeurs intermédiaires que cette équation a un sens. Pour étudier cette équation fonctionnelle, on choisit $x \in [0,1]$ et on pose $u_0 = x$. On définit ensuite, pour $n \geq 0$ u_{n+1} par*

$$u_{n+1} = f(u_n).$$

Donc cette suite vérifie

$$u_{n+1} = 2u_n - u_{n-1} \text{ si et seulement si } \frac{u_{n-1} + u_{n+1}}{2} = u_n.$$

La suite est donc une suite arithmético-géométrique ; ici, elle est de la forme

$$u_n = u_0 + n\lambda.$$

Mais la suite est à valeurs dans $[0;1]$ donc $\lambda = 0$. Ainsi, $u_n = u_0$ pour tout n, et finalement f est en fait l'identité.

Correction 2.2. *Il s'agit d'approcher ϕ de deux manières différentes.*

(1) On démontre cette inégalité par récurrence ; on peut s'en tirer de la manière suivante pour l'hérédité : on a

$$\frac{1}{2} \leq \frac{1}{a_k} \leq \frac{2}{3}$$

donc

$$\frac{3}{2} \leq 1 + \frac{1}{a_k} \leq \frac{5}{3}.$$

En observant que ϕ est le nombre d'or et qu'il résoud donc l'équation

$$\phi = 1 + \frac{1}{\phi}$$

on obtient

$$\begin{aligned} |a_{k+1} - \phi| &= \left| \frac{1}{a_k} - \frac{1}{\phi} \right| \\ &= \frac{a_k - \phi}{a_k \phi}, \end{aligned}$$

mais maintenant a_k , ϕ étant minorés par $\frac{2}{3}$ on récupère

$$|a_{k+1} - \phi| \leq \frac{4}{9} |a_k - \phi|,$$

d'où la convergence linéaire à taux $\frac{4}{9}$ de la suite.

(2) Pour la deuxième approximation du nombre d'or :

(a) On observe que

$$f'(x) = 2\frac{x^2 - x - 1}{(2x-1)^2}. \tag{2.7}$$

Le signe de f' est donné par celui de son numérateur, à savoir par le signe de $x^2 - x - 1$. Ce polynôme de degré 2 admet deux racines, le nombre d'or $\phi > \frac{1}{2}$ et $\psi := \frac{1-\sqrt{5}}{2} < 0$. En particulier, f' ne s'annule,

sur le domaine de f, qu'en un seul point, à savoir en ϕ, et f y admet un minimum global. Or

$$f(\phi) = \frac{\phi^2+1}{2\phi-1} = \phi > \frac{1}{2}.$$

Ceci conclut la preuve.

(b) On va d'abord montrer que la suite $\{c_k\}_{k\in\mathbb{N}}$ converge effectivement. Il suffit de démontrer qu'il s'agit d'une suite décroissante. Or, pour tout $x > \frac{1}{2}$ on a

$$f(x) - x = \frac{x^2+1-2x^2+x}{2x-1} = \frac{-x^2+x+1}{2x-1}.$$

Puisque $c_0 > \phi$, $f(c_0) - c_0 < 0$. En outre, on a nécessairement $f(c_0) > \phi$. En effet, f est croissante sur $(\phi;+\infty)$ donc $f(\phi) \leq f(c_0)$.
La suite $\{c_k\}_{k\in\mathbb{N}}$, puisque minorée décroissante, converge donc. Enfin,

$$\begin{aligned} c_{k+1} - \phi &= f(c_k) - \phi \\ &= \frac{c_k^2 - 2c_k\phi + \phi + 1}{2c_k - 1} \\ &= \frac{c_k^2 - 2c_k\phi + \phi_k^2}{2c_k - 1} \\ &\leq \frac{1}{2}(c_k - \phi)^2. \end{aligned}$$

On conclut par le critère de convergence quadratique.

Correction 2.3. *(1) La fonction Φ est C^∞, et on a $\Phi'(x) = 1 + \cos(x) \geq 0$, donc Φ est croissant. Supposons par l'absurde qu'il existe $a < b$ tel que $\Phi(a) = \Phi(b)$. Alors Φ serait constante sur $[a,b]$, et on aurait $\Phi'(x) = 0$ pour tout $x \in [a,b]$, ce qui est impossible, car Φ' ne s'annule qu'aux points $x = \frac{\pi}{2} + \mathbb{Z}\pi$, qui est discret. Donc Φ est strictement croissant.*

(2) Pour tout $k \in \mathbb{N}$, on a $\Phi(k\pi) = k\pi$, donc les points $k\pi$ sont des points fixes de Φ. Si $k\pi < x_0 < (k+1)\pi$, alors par croissance de Φ et une récurrence immédiate, on a $\Phi^n(k\pi) < \Phi^n(x_0) < \Phi^n((k+1)\pi)$, où Φ^n est composée n fois de Φ. Cette dernière égalité est aussi $k\pi < x_n < (k+1)\pi$.
Si k est paire, la fonction sin *est (strictement) positive sur $(k\pi,(k+1)\pi)$. Donc, si $k\pi \leq x_n \leq (k+1)\pi$, on a $x_{n+1} - x_n = \sin(x_n) \geq 0$, et par une récurrence immédiate, la suite (x_n) est strictement croissante.*
Le raisonnement est similaire si k est impair.

(3) D'après le point b/, si $x_0 = 3 \in (0,\pi)$, la suite (x_n) est strictement croissante, et majorée par π, donc converge vers une limite $x^* \in [3,\pi]$. Par continuité de Φ, on doit avoir $x^* = \Phi(x^*)$. La seule limite possible est donc $x^* = \pi$.

(4) D'après la formule de Taylor à l'ordre 2, il existe $\xi \in (x,\pi)$ tel que

$$\Phi(x) = \Phi(\pi) + \Phi'(\pi)\cdot(x-\pi) + \frac{1}{2}\Phi''(\pi)\cdot(x-\pi)^2 + \frac{1}{6}\Phi'''(\xi)\cdot(x-\pi)^3.$$

Dans notre cas, on a $\Phi'(\pi) = 1 + \cos(\pi) = 0$, $\Phi''(\pi) = -\sin(\pi) = 0$ (et $\Phi'''(\pi) = -\cos(\pi) = 1 \neq 0$). Ainsi,

$$|\Phi(x) - \Phi(\pi)| = \frac{1}{6}|\Phi'''(\xi)|\cdot|x-\pi|^3 \leq \frac{1}{6}\max|\Phi'''|\cdot|x-\pi|^3.$$

(5) *En se rappelant que* $\Phi(\pi) = \pi$, *et si* $x_0 = 3$, *alors* $|\pi - x_0| < 1$. *De plus, comme* $\frac{1}{6} < 1$, *on obtient par récurrence que*

$$|x_n - \pi| < |x_{n-1} - \pi|^3 < \cdots < |x_0 - \pi|^{3^n},$$

et la méthode converge à l'ordre 3.

Correction 2.4. *On pose* $g(x) = e^{-x}$. *Sur* $(0; +\infty)$, *on a* $|g'| < 1$, *on a une fonction contractante, on est donc dans une bonne situation pour essayer d'appliquer un théorème de point fixe. Évidemment, on n'a aucunement le droit de le faire puisque l'on ne travaille pas sur un compact. Il faut nous ramener au cas d'un intervalle fermé borné. Pour cela, il suffit de trouver* a, b *tels que* $g([a, b]) \subset [a, b]$. *Puisque la fonction* g *est décroissante sur* $(0; +\infty)$, *on doit trouver* a *et* b *tels que*

$$g(a) \leq b\,, g(b) \geq a.$$

Il suffit donc de choisir a *suffisamment petit pour que*

$$e^{-a} < \ln\left(\frac{1}{a}\right).$$

Dans ce cas, on choisit n'importe quel b *dans* $(e^{-a}, -\ln(a))$ *et on peut conclure que la suite converge alors vers un point fixe* x^*. *Par ailleurs, on a* $g'(x^*) \neq 0$, *de sorte que la convergence est linéaire.*

Correction 2.5. *Pour appliquer la méthode de Newton à cette équation, on doit définir une fonction auxiliaire, à savoir*

$$h(x) = x - e^{-x}$$

de dérivée

$$h'(x) = 1 + e^{-x} \neq 0.$$

On a donc de manière explicite les itérations de la méthode de Newton, sous la forme

$$x_{k+1} = x_k - \frac{x_k - e^{-x_k}}{1 + e^{-x_k}}.$$

Par les théorèmes généraux sur la méthode de Newton, la convergence est (au moins) quadratique.

Correction 2.6. (1) *On commence par calculer la dérivée de* f_α :

$$f'_\alpha(x) = \mathrm{sgn}(x)\alpha|x|^{\alpha-1}.$$

On en déduit que, quel que soit le point d'initilisation

$$x_{k+1} = x_k\left(1 - \frac{1}{\alpha}\right).$$

La première condition pour que cette méthode converge est d'avoir

$$\left|1 - \frac{1}{\alpha}\right| < 1.$$

Hors si $\alpha < 1$ *cette condition se réécrit*

$$\alpha^{-1} - 1 < 1 \Leftrightarrow \alpha > \frac{1}{2}.$$

(2) On a

$$f'(x) = 3x - 5$$

donc, en partant d'une initialisation x_0 on trouve

$$x_1 = x_0 - \frac{x_0\left(x_0^2 - 5\right)}{3x_0 - 5}$$

donc $x_1 = \pm 1$ si $x_0 = \mp 1$.

(3) (a) Un calcul immédiat donne

$$x_1 = \frac{1}{2}(x_0 - x_0^{-1}).$$

(b) Il suffit de réduire l'expression de droite au même dénominateur et d'utiliser

$$\sin^2(t) = \frac{1 - \cos(2t)}{2}, \cos^2(t) = \frac{1 + \cos(2t)}{2}.$$

(c) Évident (tout comme la non-unicité).

(d) Au bout de deux itérations, la suite n'est plus bien définie.

(e) Immédiat.

(f) Si $t_0 = \frac{4}{3}\pi$, alors, en définissant $t_k := 2^k t_0$, on a

$$t_k = \frac{2^{k+2}}{3}\pi$$

donc

$$t_k - t_0 = \frac{2\left(2^k - 1\right)}{3}\pi.$$

Ainsi, puisque $2^k = (-1)^k \mathrm{mod}(3)$, on a, ou bien $t_k = t_0 \mathrm{mod}(2\pi)$, ou bien $t_k = t_0 + q_k\pi - \frac{1}{3}\pi$, donc la suite est périodique.

(g) Évident. On peut en fait démontrer que la suite est chaotique, car conjuguée au shift de Bernoulli (voir Devaney, A First Course in Chaotic Dynamical Systems Theory and Experiment Second Edition*).*

Correction 2.7. *a/ On a $0 < \Phi'(0) < \alpha$. Donc par continuité de Φ', il existe un voisinage de 0 tel que ces inégalités strictes restent vraies, c'est à dire*

$$\exists \varepsilon > 0,\ \forall x \in (-\varepsilon, \varepsilon), \quad 0 < \Phi'(x) < \alpha, \quad \text{et donc} \quad |\Phi'(x)| < \alpha.$$

b/ On suppose $x_n \in (-\varepsilon, \varepsilon)$. On a

$$|x_{n+1}| = |\Phi(x_n)| = \left|\int_0^{x_n} \Phi'(t)dt\right| \leq \int_0^{x_n} |\Phi'(t)|\, dt \leq \alpha x_n.$$

En particulier, on en déduit que $|x_{n+1}| < \alpha\varepsilon < \varepsilon$, et donc, par récurrence, on a $x_n \in (-\varepsilon, \varepsilon)$ pour tout n. De plus, on a $|x_{n+1}| < \alpha|x_n| \leq \alpha^{n+1}x_0$ par récurrence. On en déduit que (x_n) converge vers 0 avec une convergence linéaire d'au moins α.
c/ Soit x_0 un point fixe de Φ dans $(-\varepsilon, \varepsilon)$. La suite $x_{n+1} = \Phi(x_n)$ est alors constante égale x_0. D'après la question précédente, on a donc $x_0 \to 0$, donc $x_0 = 0$.
On peut aussi montrer que $f(x) = x - \Phi(x)$ est strictement croissante sur $(-\varepsilon, \varepsilon)$.
d/ Si $\Phi(x) = \alpha x$, on a

$$\Psi(x) = x - \frac{[\alpha x - x]^2}{x - 2\alpha x + \alpha^2 x} = x - \frac{(\alpha - 1)^2 x^2}{x(1 - 2\alpha + \alpha^2)} = x - \frac{(\alpha - 1)^2}{(\alpha - 1)^2}\frac{x^2}{x} = 0.$$

e/ Dans le cas général, on a $\Phi(x) = \alpha x + o(x)$*. On en déduit*

$$[\Phi(x) - x]^2 = [\alpha x + o(x) - x]^2 = (\alpha - 1)^2 x^2 + o(x^2)$$

et

$x-2\Phi(x)+\Phi(\Phi(x)) = x-2(\alpha x+o(x))+\Phi(\alpha x+o(x)) = x-2\alpha x+o(x)+\alpha(\alpha x+o(x))+o(x) = x(1-2\alpha+\alpha^2)+o(x).$

Donc

$$\Psi(x) = x - \frac{(\alpha - 1)^2 x^2 + o(x)}{(\alpha - 1)^2 x + o(x)} = o(x).$$

On en déduit que Ψ *est dérivable en* 0*, avec* $\Psi(0) = 0$ *et* $\Psi'(0) = 0$*.*
f/ Si $\beta := \Psi''(0) \neq 0$*, on a* $\Psi(x) \approx \beta x^2 + o(x^2)$*. Au premier ordre, on a donc* $x_{n+1} = \beta x_n^2$*, et on en déduit que la suite* (x_n) *converge quadratiquement vers* 0 *si* x_0 *est proche de* 0*.*

Feuille de TP n°2 : Approximation par différences finies, dichotomie

Exercice 2.9. Écrire une fonction prenant en argument une fonction f, un point x, un pas $\delta > 0$ et renvoyant l'approximation par différences finies de $f'(x)$. L'appliquer à une fonction dont vous connaissez la dérivée (non nulle), comme $f(x) = \sin(x)$ en 1, $f(x) = x^2$ en 2, et afficher le comportement de l'erreur entre la valeur approchée et la valeur exacte en un point choisi, en fonction du pas δ, pour différentes valeurs de $\delta > 0$. Vous pouvez prendre $\delta = 10^{-k}$ pour $k \in \{1, \dots, 16\}$. Observer que, si δ est trop petit, on perd en fait en qualité d'approximation.

Exercice 2.10 (Dichotomie). Écrire un code prenant en argument une fonction réelle f, deux points a, b, un pas $\delta > 0$, une tolérance, et qui calcule, par une méthode de dichotomie appliquée à f', un minimum de f sur $[a, b]$. On supposera que la fonction f est décroissante puis croissante sur l'intervalle $[a, b]$ choisi.

Exercice 2.11 (Section dorée).

(1) Écrire un algorithme qui prenne en argument une fonction f, deux réels $a < b$, une tolérance et un nombre maximal d'itérations, et qui renvoie le minimum calculé numériquement par la méthode de la section dorée. On fera bien attention à ce que l'algorithme vérifie l'admissibilité des différents triplets considérés, et à renvoyer non seulement le dernier point obtenu, mais encore la suite de tous les points construits.

(2) Vérifier graphiquement que la fonction $x \mapsto -e^{\arctan(x)-\cos(5x)}$ est unimodale sur $[0, 1]$ et calculer son minimum par la méthode de la section dorée. On représentera, à chaque étape, le point obtenu sur le graphe de la fonction.

Exercice 2.12 (Méthode de Newton).

(1) On se donne une fonction f et l'on suppose que l'on peut calculer sa dérivée f'. Écrire un code qui prenne en argument une fonction f, une fonction g (destinée à être la fonction f'), une tolérance et un nombre maximal d'itérations, et qui renvoie le point trouvé par la méthode de Newton.

(2) Utiliser cet algorithme pour calculer π à 10 décimales près.

(3) Appliquer cette méthode pour déterminer numériquement la racine n-ième d'un nombre positif à 10 décimales près.

Exercice 2.13 (Méthode de la sécante). Écrire un code qui prenne en argument une fonction f, deux points d'initialisation, une tolérance et un nombre maximal d'itération, et qui renvoie le résultat de la méthode de la sécante avec ces paramètres.

Méthodes multi-dimensionnelles : présentation succinte

Présentation de la descente de gradient. Dans ce mini-cours nous ne faisons que présenter les éléments fondamentaux d'un des algorithmes qui, malgré sa convergence lente, s'avère extrêmement pratique dans les applications, la **descente de gradient**.

Moralement, l'idée est la suivante : on travaille avec une fonction $f : \mathbb{R}^d \to \mathbb{R}$ aussi régulière que souhaité, et l'on désire résoudre (en admettant qu'il admette une solution) le problème

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

On se donne une initialisation $x_0 \in \mathbb{R}^d$. Pour déterminer une construction du point suivant x_1 qui ne soit pas trop bête, une idée possible est d'utiliser un développement de Taylor à l'ordre 1. Expliquons cela : il nous faut une **direction** dans laquelle aller, ainsi que la **taille du pas** dans cette direction. On se fixe done un vecteur $h \in \mathbb{R}^d$, qui définit la direction, et un réel $\tau > 0$ (destiné à être petit).

On peut écrire, au premier ordre

$$f(x_0 + \tau h) \approx f(x_0) + \tau\langle h, \nabla f(x_0)\rangle.$$

Si l'on s'est fixé la taille τ du pas, alors on peut supposer que

$$\|h\| = 1$$

pour la norme euclidienne. Il nous reste à déterminer la direction dans laquelle descendre. Plusieurs possibilités s'offrent à nous, mais l'une d'entre elles est de choisir un vecteur h solution du problème d'optimisation

$$\min_{\|h\|=1} \langle \nabla f(x_0), h\rangle.$$

L'inégalité de Cauchy-Schwarz nous garantit que si $\nabla f(x_0) = 0$ alors une solution du problème d'optimisation ci-dessus est donnée par

$$h_{x_0} := -\frac{\nabla f(x_0)}{\|\nabla f(x_0)\|}.$$

L'algorithme de la **descente de gradient à pas constant** est une mise en œuvre de cette idée. Ici, on part d'une initialisation x_0, on se fixe un pas $\tau > 0$ et l'on définit, pour tout $k \in \mathbb{N}$, la suite

$$x_{k+1} := x_k - \tau\nabla f(x_k).$$

Une grosse partie du cours qui reste vise à appréhender, à analyser et à maîtriser cet algorithme, ainsi que ses variantes habituelles.

Présentation de la méthode de Newton. Un autre algorithme que nous étudierons amplement dans la suite de ce cours est l'algorithme de Newton. De même qu'en dimension 1 cet algorithme nous permettait de déterminer des points critiques et donc, potentiellement, des minimiseurs, en dimension supérieure, l'algorithme de Newton nous permettra, pour une fonction $f \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R})$,d e trouver des points critiques de f, c'est-à-dire de résoudre l'équation

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

Puisque, comme en dimension 1, on a l'approximation

$$\nabla f(x) \approx \nabla f(x_0) + \nabla^2 f(x_0)\cdot(x - x_0)$$

on en déduit que la suite des itérations de l'algorithme de Newton est donnée par, d'une part, une initialisation x_0 et, pour tout entier naturel k,

$$x_{k+1} = x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k).$$

Cet algorithme sera également l'objet d'une analyse approfondie.

3. Descente de gradient I : analyse en dimension 1

3.1. **Introduction.** On présente dans ce cours quelques aspects de la méthode de descente de gradient en dimension 1, qui servira d'introduction à la méthode en plusieurs dimensions. Cette méthode est utilisée pour résoudre des problèmes d'optimisation

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

où f est une fonction suffisamment régulière. Il est important de noter qu'ici on ne travaille qu'avec un problème d'optimisation sans contraintes, et que les conditions d'optimalité suivent donc, dans un premier temps, la règle de Fermat,

$$f'(x^*) = 0$$

et, dans un second, les règles de Lagrange

$$f''(x^*) \geq 0.$$

Ensuite, la méthode de la descente de gradient est une méthode algorithmique qui prend en argument à la fois la fonction f et sa dérivée. L'avantage considérable par rapport à une méthode de type Newton est que nous n'avons pas à calculer de dérivées secondes, ce qui allège le coût computationnel. L'inconvénient principal est que la vitesse de convergence de l'algorithme est plus faible (linéaire au lieu de quadratique).

3.2. **Approche heuristique de la méthode.** L'objectif de cette méthode locale est, partant d'un point initial x_0, d'obtenir un point x_1 qui "fasse mieux" que x_0 en effectuant un petit pas dans une direction bien choisie. Pour cela, le point central est d'utiliser, une fois de plus, les développements de Taylor : en partant d'un point x_0, on a

$$f(x_0 + h) \approx f(x_0) + hf'(x_0).$$

Le paramètre (petit) h quantifie à la fois le sens et l'intensité du pas où l'on doit chercher un minimiseur. Puisque l'on veut minimiser la fonction f, il faut choisir un h tel que

$$hf'(x_0) < 0.$$

Il est a priori loisible de prendre un h très grand en norme, mais cela est en contradiction avec le caractère local de la méthode. Ainsi, plutôt que de tout paramétrer par un même réel h, on distingue (artificiellement en dimension 1) cette direction de descente et la grandeur du pas, en paramétrant plutôt la méthode par un couple (τ, h), où $\tau \in \mathbb{R}_+^*$ est la taille du pas de descente, et $h \in \pm 1$ est défini par

$$h := -\frac{f'}{\|f'\|}(x_0).$$

Cette direction h est appelée *direction de descente pour f en x_0*. On vérifie facilement, grâce à un développement limité, que l'on a bien la résultat suivant :

Proposition 3.1. *Soit $f \in \mathscr{C}^2(\mathbb{R};\mathbb{R})$. Soit $x_0 \in \mathbb{R}$ tel que $f'(x_0) \neq 0$. Soit $h := -f'(x_0)/|f'(x_0)|$. Alors, pour $\tau > 0$ suffisamment petit, si l'on définit x_1 par*

$$x_1 = x_0 + \tau h$$

on a

$$f(x_1) < f(x_0).$$

En particulier, si on l'itère, cette méthode fournit bien une suite qui "devrait" bien s'approcher d'un optimum.

Ceci nous incite à définir un algorithme de la manière suivante :

(1) On initialise en un certain x_0.

(2) Pour tout $k \in \mathbb{N}$, on définit

$$x_{k+1} := x_k - \tau_k f'(x_k)$$

pour un certain pas τ_k (noter qu'ici on ne normalise pas d'entrée de jeu la direction de descente, pour ne pas s'embêter à distinguer les points critiques).

Néanmoins, il reste à déterminer le pas de descente τ, ce qui peut s'avérer crucial. Afin de bien marquer ce point, évoquons les difficultés principales liées à cette méthode.

3.2.1. *Non-convergence pour des pas trop grands.* Insistons encore une fois sur la nécessité de choisir un bon pas pour la descente de gradient. On considère, par exemple, la fonction

$$f(x) = \frac{x^2}{2}.$$

Alors $f'(x) = x$ et, avec à l'étape k un pas τ_k, on a, pour les itérations successives de la méthode de gradient

$$x_{k+1} = (1 - \tau_k)x_k = \cdots = x_0 \prod_{i=0}^{k} (1 - \tau_i).$$

On voit que si on prend des τ_i qui sont tous plus grands que 1, la méthode diverge et que, si on prend $\tau_i := 2(-1)^i$, on génère une suite périodique.

3.2.2. *Sensibilité aux points critiques et vitesse de convergence.* La méthode de descente de gradient est une méthode qui, par nature, est *locale*. En particulier, si on l'initialise en un point critique x_0 (ou si on s'approche d'un point critique x_0) alors on y reste bloqué. En particulier, on fera bien attention à distinguer entre la convergence de la méthode, qui sera vers un point critique, et la caractérisation du point limité comme un minimum global. À titre d'exemple, considérons la fonction

$$f : x \mapsto \frac{x^3}{3}$$

et définissons $x_0 = 1$. On se fixe un pas $\tau \in (0, 1)$. Alors la méthode de descente à pas τ fournit la suite d'itérations

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

Ainsi, la suite fournie est décroissante, et converge vers 0, qui n'est évidemment pas un minimum (ni global, ni même local), et même plus lentement que linéairement. En fait, nous verrons que le critère naturel pour la descente de gradient est que les points critiques soient non-dégénérés.

Ceci peut avoir une conséquence pratique importante : il se peut que $x_{k+1} - x_k$ soit, en norme, inférieur à la précision numérique de l'ordinateur.

L'autre information que cet exemple fourni est que l'on ne peut espérer qu'un paradigme du type

$$\text{Convergence vers } x^* \Rightarrow \text{convergence vers un minimiseur}$$

ne saurait être valable que si l'on travaille avec une fonction f pour laquelle points critiques et minima coïncident, ce qui est le cas, par exemple, pour les fonction convexes.

3.2.3. *Adaptation en dimensions supérieures.* En dimensions supérieures, l'autre difficulté sera de choisir *une direction de descente* et une *taille de pas de descente*, ce qui requérera plusieurs notions fines de calcul différentiel que nous verrons dans la seconde partie de ce cours.

3.3. **Démonstration de plusieurs résultats afférents en dimension 1.** On donne ici les démonstrations de quelques résultats en dimension 1, qui donnent un échauffement pour les preuves en dimension supérieures. Dans tout ce qui suit, $f \in \mathscr{C}^1(\mathbb{R};\mathbb{R})$.

Théorème 3.1 (Quelques résultats en dimension 1)**.** *On a les résultats suivants :*

(1) *Décroissance de la suite des valeurs : on suppose que f' est M-lipschitzienne. Si $0 < \tau < \frac{1}{M}$, la suite $\{f(x_k)\}_{k\in\mathbb{N}}$ est strictement décroissante, $\{x_k\}$ étant la suite générée par descente de gradient à pas fixe τ initialisée en tout x_0.*

(2) *Décroissance de la suite des valeurs précisées : on suppose que f est convexe, que f atteint son minimum en x^* et que f' est M-lipschitzienne. Si $\tau < \frac{1}{2M}$, alors*

$$f(x_k) - f(x^*) \leq \frac{|x_0 - x^*|^2}{2\tau k}.$$

(3) *Convergence linéaire vers les minima non-dégénérés : on suppose que $f \in \mathscr{C}^2$, que x^* est un point critique de f et que f est strictement convexe sur $[x^* - \varepsilon; x^* + \varepsilon]$ pour $\varepsilon > 0$ suffisamment petit. En d'autres termes, il existe $\ell_0\,, \ell_1 > 0$ tels que*

$$\forall x \in [x^* - \varepsilon; x^* + \varepsilon]\,, f''(x) \in [\ell_0; \ell_1].$$

Si $\alpha < \frac{\ell_1}{2}$, si $x_0 \in [x^* - \varepsilon; x^* + \varepsilon]$, la suite $\{x_k\}_{k\in\mathbb{N}}$ converge linéairement vers x^* à taux $\max(|1 - \tau\ell_0|, |1 - \tau\ell_1|)$. On peut optimiser ce taux en choisissant $\tau^* = \frac{2}{\ell_0 + \ell_1}$ et le taux optimal de convergence est alors $\frac{\ell_1 - \ell_0}{\ell_0 + \ell_1}$.

Preuve du théorème 3.1. (1) On suppose que f' est M-lipschitzienne. Par les théorèmes des accroissements finis on sait que pour tout entier naturel k on a

$$f(x_{k+1}) - f(x_k) = (x_{k+1} - x_k) f'(\xi)$$

pour un certain $\xi \in [x_k; x_{k+1}]$. Ainsi,

$$f(x_{k+1}) - f(x_k) \leq f'(x_k)(x_{k+1} - x_k) + M(x_{k+1} - x_k)^2.$$

Or, puisque $x_{k+1} = x_k - \tau f'(x_k)$ ceci implique que

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\leq -\tau \left(f'(x_k)\right)^2 + \tau^2 M (f'(x_k))^2 \\ &= \left(f'(x_k)\right)^2 (M\tau - 1)\, \tau, \end{aligned}$$

d'où la conclusion annoncée.

(2) On reprend l'estimation précédente :

$$f(x_{k+1}) - f(x_k) \leq \left(f'(x_k)\right)^2 (M\tau - 1)\, \tau.$$

On pose $-\delta := (M\tau - 1)\tau < 0$. Maintenant, par convexité de la fonction f, on sait également, une fonction convexe étant au dessus de ses tangentes, que

$$\forall y\,, f(x^*) \geq f(y) + f'(y)(x^* - y),$$

de sorte que l'on obtient en outre

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \delta(f'(x_k))^2 \\ &\leq f(x^*) + f'(x_k)(x_k - x^*) - \delta(f'(x_k))^2 \\ &\Leftrightarrow \\ f(x_{k+1}) - f(x^*) &\leq \frac{1}{\delta}\left(-\delta^2(f'(x_k))^2 + \delta f'(x_k)(x_k - x^*)\right) \\ &\leq \frac{1}{\delta}\left(\frac{|x_k - x^*|^2}{4} - \left(\delta f'(x_k) - \frac{(x_k - x^*)}{2}\right)^2\right). \end{aligned}$$

On ne voit pas encore tout à fait ce que l'on a gagné avec cette méthode, mais observons la chose suivante : si on choisit τ de sorte que

$$\delta > \frac{\tau}{2}$$

ce qui en gardant en tête que $\delta = (1 - M\tau)\tau$, revient à choisir $\tau < \frac{1}{2M}$ on récupère en fait, par la même méthode, dans le terme de droite,

$$\frac{\tau}{2} f'(x_k) - \frac{x_k - x^*}{2} = \frac{x^* - x_{k+1}}{2}.$$

Ainsi, on a finalement

$$f(x_{k+1}) - f(x^*) \leq \frac{1}{4\delta}\left(|x_k - x^*|^2 - |x_{k+1} - x^*|^2\right) \leq \frac{1}{2\tau}\left(|x_k - x^*|^2 - |x_{k+1} - x^*|^2\right).$$

En sommant ces estimations et en utilisant le premier point du théorème, il vient

$$N\left(f(x_N) - f(x^*)\right) \leq \sum_{k=0}^{N-1} \left\{f(x_{k+1}) - f(x^*)\right\} \leq \frac{1}{2\tau}\left(x_0 - x^*\right)^2.$$

La conclusion s'ensuit.

(3) Passons, enfin, au comportement de l'algorithme au voisinage des points critiques non-dégénérés. Ici, la preuve est beaucoup plus simple. Il suffit d'observer que x^* est l'unique point fixe de la fonction $F : x \mapsto x - \tau f'(x)$ dans $[x_0 - \varepsilon; x_0 + \varepsilon]$ si $|1 - \tau\ell_0|\,, |1 - \tau\ell_1| < 1$. Il suffit ensuite de remarquer que l'on travaille simplement sur une suite de points fixes.

□

En particulier, le dernier point de ce théorème indique, s'il était encore nécessaire, que cette méthode ne peut capturer que des minima locaux.

Feuille de TD n°3 : descente de gradient en dimension 1, calcul différentiel (révisions)

Descente de gradient.

Exercice 3.1 (Newton et descente de gradient). Montrer que l'algorithme de Newton est un algorithme de descente de gradient à pas variable bien choisi.

Exercice 3.2. Soit $F : x \mapsto x^2 - \frac{1}{2}x^4$, soit $\tau > 0$ et soit (x_n) la suite définie par $x_0 \in \mathbb{R}$ et $x_{n+1} = x_n - \tau F'(x_n)$.

(1) Montrer que $x^* := 0$ est un minimum local de F.

(2) Montrer que si $x_0 > 1$, alors pour tout $\tau > 0$, la suite (x_n) est croissante et diverge vers $+\infty$.

(3) Dans la suite, on suppose $|x_0| < 1$. Montrer que pour tout $0 < \tau < 1$, la suite $(|x_n|)$ est décroissante, et converge vers x^*.

(4) Montrer que dans ce cas, la vitesse de convergence est linéaire à taux au moins α pour tout $|1 - 2\tau| < \alpha < 1$.

(5) Dans le cas $\tau = \frac{1}{2}$, montrer que (x_n) converge vers x^* à l'ordre 3.

(6) Dans le cas $\tau = 1$. Montrer que la suite $(|x_n|)$ est décroissante, converge vers 0, et qu'on a

$$\frac{1}{x_{n+1}^2} - \frac{1}{x_n^2} = 4\frac{1 - x_n^2}{(1 - 2x_n^2)^2}.$$

En déduire que la suite $\left(\frac{1}{x_{n+1}^2} - \frac{1}{x_n^2}\right)$ converge vers 4, puis que la suite $\sqrt{n}|x_n|$ converge vers $\frac{1}{2}$ (on pourra utiliser le théorème de Césàro). Combien d'itérations faut-il faire pour avoir une précision de 10^{-6} ?

Exercice 3.3. Soit $a > 0$, $F : x \mapsto ax^2$ et $\tau \in \mathbb{R}$. On pose $x_0 = 1$ puis $x_{n+1} = x_n - \tau F'(x_n)$.

(1) Comment se comporte la suite (x_n) pour les différentes valeurs de τ ?

(2) On pose maintenant $x_{n+1} = x_n - \tau\frac{1}{\varepsilon}[F(x_n + \varepsilon) - F(x_n)]$ (différence finie). Que se passe-t-il maintenant ?

(3) Que se passe-t-il avec des différences finies centrées ?

Gradients, ensembles de niveaux.

Exercice 3.4. (1) Calculer le gradient des fonctions suivantes (on suppose que $A \in M_d(\mathbb{R})$ et $b \in \mathbb{R}^d$)

$$F_1(x) := \frac{1}{2}x^T Ax - b^T x, \quad F_2(x) := \mathrm{Tr}(xx^T), \quad F_3(x) := (1+x_1)(1+x_2)\cdots(1+x_d).$$

(2) Soit $g : \mathbb{R} \to \mathbb{R}$ une fonction de classe C^1, et soit $F : \mathbb{R}^2 \to \mathbb{R}$ définie par $F(x) := g(\|x\|)$. Calculer le gradient de F et représenter les courbes de niveau et le gradient de F.

Exercice 3.5. Soit $V \in \mathscr{C}^1(\mathbb{R}^d, \mathbb{R})$. On considère l'équation différentielle

$$x' = -\nabla V(x),$$

assortie d'une certaine condition initiale $x(0) = x_0 \in \mathbb{R}^n$.

(1) Soit $f : t \mapsto V(x(t))$. Calculer f', et montrer que $f' \leq 0$.

(2) On suppose que $\lim_{\|x\|\to\infty} V = +\infty$. Montrer que les trajectoires de l'équation différentielle sont bornées.

(3) Montrer que tout point d'équilibre est un point critique de V.

(4) On suppose que V est de classe $\mathscr{C}^2$ et que $\nabla^2 V \geq \lambda I_d$ pour un certain $\lambda > 0$ et pour tout $x \in \mathbb{R}^d$. Montrer que, pour toute donnée initiale $x_0 \in \mathbb{R}^d$ on a

$$\|x(t) - x^*\|^2 \leq e^{-2\lambda t}\|x(0) - x^*\|^2.$$

Correction de la feuille de TD n°3

Correction 3.1. *Évident.*

Correction 3.2. *(1) On a $F'(0) = 0$ et $F''(0) = 2$, donc 0 est un minimum local strict.*

(2) On a

$$x_{n+1} = x_n - \tau(2x_n - 2x_n^3) = x_n(1 - 2\tau + 2\tau x_n^2) = x_n\tau_n \quad \text{avec} \quad \tau_n := (1 - 2\tau + 2\tau x_n^2).$$

Si $x_n > 1$, on a $\tau_n > 1$, puis $x_{n+1} = \tau_n x_n > x_n$. Ainsi, si $x_0 > 1$, on obtient par une récurrence immédiate que les suites (x_n) et (τ_n) sont croissantes. En particulier, on a $x_{n+1} \geq \tau_0 x_n \geq \cdots \geq \tau_0^n x_0 \to \infty$.

(3) Si $x_n \in (-1, 1)$, on a $x_n^2 \in (0, 1)$, et $\tau_n \in (1 - 2\tau, 1)$. En particulier, si $0 < \tau < 1$, on a $|\tau_n| < 1$, puis $|x_{n+1}| < |x_n|$. La suite $(|x_n|)$ est donc décroissante et minorée par 0, elle converge vers un certain l. Par continuité, on doit avoir $0 \leq l < 1$ et $l = l(1 - 2\tau + 2\tau l^2)$, et donc $l = 0$.

(4) Soit $\alpha > |1 - 2\tau|$. Comme x_n converge vers 0, on a $|\tau_n| < \alpha$ à partir d'un certain rang $n_0 \in \mathbb{N}$. On obtient, pour $n > n_0$, $|x_n| \leq \alpha x_{n-1} \leq \cdots \leq \alpha^{n-n_0} x_{n_0} \leq (x_{n_0}\alpha^{-n_0})\alpha^n$, et la convergence est linéaire de taux α. Ceci étant vrai pour tout $\alpha > |1 - 2\tau|$, on obtient une convergence linéaire à taux $|1 - 2\tau|$.

(5) Dans le cas $\tau = \frac{1}{2}$, on a $x_{n+1} = x_n^3 = \cdots = x_0^{3^{n+1}}$, CQFD.

(6) Si $\tau = 1$, on a $x_{n+1} = x_n(2x_n^2 - 1)$. Si $0 < |x_n| < 1$, alors $|\tau_n| < 1$, et donc $|x_{n+1}| < x_n$, et si $x_n = 0$, alors $x_{n+1} = 0$. Dans tous les cas, la suite $(|x_n|)$ est décroissante. Sa limite l doit vérifier $0 \leq l < 1$ et $l = l(2l^2 - 1)$ (donc $l = 0$ ou $l = \pm 1$). On obtient $l = 0$.
On a

$$\frac{1}{x_{n+1}^2} - \frac{1}{x_n^2} = \frac{1}{x_n^2}\left(\frac{1}{(2x_n^2 - 1)^2} - 1\right) = \frac{1}{x_n^2}\left(\frac{4x_n^2 - 4x_n^4}{(2x_n^2 - 1)^2}\right) = \frac{4 - 4x_n^2}{(2x_n^2 - 1)^2}.$$

Comme x_n converge vers 0, le terme de droite converge vers 4. D'après le théorème de Césàro, on a aussi

$$4 = \lim_{n\to\infty} \frac{1}{n}\left(\frac{1}{x_n^2} - \frac{1}{x_{n-1}^2} + \frac{1}{x_{n-1}^2} - \frac{1}{x_{n-2}^2} + \cdots + \frac{1}{x_1^2} - \frac{1}{x_0^2}\right)$$
$$= \lim_{n\to\infty} \frac{1}{nx_n^2} - \frac{1}{nx_0^2} = \lim_{n\to\infty} \frac{1}{nx_n^2}.$$

Correction 3.3. *(1) On a $F'(x) = 2ax$, donc la formule d'itération se réécrit $x_{n+1} = x_n(1 - 2\tau a)$. On obtient donc par récurrence $x_n = (1 - 2\tau a)^n$.*

- *Si $(1 - 2\tau a) < -1$, c'est à dire $\tau > 1/a$ la suite diverge en oscillant ;*
- *Si $(1 - 2\tau a) = -1$, c'est à dire $\tau = 1/a$, la suite oscille entre -1 et 1 ;*

— *Si* $|1-2\tau a|<1$, *c'est à dire* $0<\tau<1/a$, *la suite converge vers* 0. *La vitesse de convergence est linéaire à taux* $|1-2\tau a|$. *De plus, dans le cas où* $\tau=1/(2a)$, *on a* $x_1=x_2=\cdots=0$;

— *Si* $(1-2\tau a)=1$, *c'est à dire* $\tau=0$, *la suite est constante égale à* 1 ;

— *Si* $(1-2\tau a)>1$, *c'est à dire* $\tau<0$, *la suite diverge vers* $+\infty$.

(2) On a $\frac{1}{\varepsilon}\left[a(x_n+\varepsilon)^2-ax_n^2\right]=2ax_n+a\varepsilon$, *donc la formule d'itération devient* $x_{n+1}=x_n-\tau a(2x_n+\varepsilon)=(1-2\tau a)x_n-\tau a\varepsilon$. *En posant* $y_n=x_n+\frac{1}{2}\varepsilon$, *on obtient* $y_{n+1}=(1-2\tau a)y_n$. *On a les mêmes phénomènes que précédemment pour la suite* (y_n). *En particulier, dans le cas où* $|1-2\tau a|<1$, *la suite* (x_n) *converge vers* $-\varepsilon/2$.

(3) On a $\frac{a}{2\varepsilon}\left[(x_n+\varepsilon)^2-(x_n-\varepsilon)^2\right]=2x_n=F'(x_n)$, *donc on retrouve les résultats de question a/ dans le cas des différences finies centrées.*

Correction 3.4. *(1) On trouve* $\nabla F_1(x)=\frac{1}{2}(A+A^T)x-b$ *(attention à la transposée). Pour* F_2, *on remarque que, par cyclicité de la trace, on a* $\mathrm{Tr}(xx^T)=\mathrm{Tr}(x^Tx)=x^Tx$. *Donc d'après la question précédente, on a* $\nabla F_2(x)=2x$. *Pour* F_3, *on trouve*

$$\nabla F_3(x)=\begin{pmatrix}(1+x_2)(1+x_3)\cdots(1+x_d)\\(1+x_1)(1+x_3)\cdots(1+x_d)\\\vdots\\(1+x_2)(1+x_3)\cdots(1+x_{d-1})\end{pmatrix}.$$

(2) Pour tout $x\neq 0$, *on a* $\nabla\|x\|=\frac{x}{\|x\|}$. *Avec la règle de la composition de la chaîne, on trouve* $\nabla F(x)=g'(\|x\|)\frac{x}{\|x\|}$. *Les courbes de niveau sont des cercles, et le gradient est orthogonal aux courbes de niveau.*

Correction 3.5. *(1) Par un calcul direct on obtient*

$$f'((t)=-\|\nabla V(x(t))\|^2\leq 0.$$

(2) Par le calcul précédent, on en déduit qu'il existe $M\in\mathbb{R}$ *tel que, pour tout* $t\geq 0$, $x(t)\in\{|V|\leq M\}$. *Par hypothèse, ces ensembles de niveau sont bornés, donc les trajectoires sont bornées.*

(3) Évident.

(4) Il faut d'abord obtenir une propriété de monotonie du gradient de la fonction V. *En effet, il est facile d'obtenir que*

$$\forall x,y\in\mathbb{R}^d\,,\langle y-x,\nabla V(y)-\nabla V(x)\rangle\geq\lambda\|y-x\|^2.$$

Pour obtenir cette estimation, on étudie simplement la fonction

$$g:t\mapsto\langle y-x,\nabla V(tx+(1-t)y)\rangle$$

On a alors

$$g(1)-g(0)=\int_0^1 g'(t)dt\geq\lambda\|y-x\|^2.$$

On observe ensuite que la fonction V ne peut avoir, par l'estimation précédente, qu'un unique point critique. Par ailleurs,

$$\frac{d}{dt}\|x(t) - x^*\|^2 = -2\langle x(t) - x^*\,, \nabla V(x(t)) - \nabla V(x^*)\rangle \leq -2\lambda\|x(t) - x^*\|^2,$$

et on conclut par le lemme de Gronwall.

FICHE DE TP N°3 : DESCENTE DE GRADIENTS, ENSEMBLES DE NIVEAUX

3.4. **Exemples unidimensionnels.**

Exercice 3.6 (CF TD n°3).
(1) Coder la descente de gradient à pas fixe en écrivant un algorithme qui prenne en argument une fonction f, une fonction *df* (appelée à être f' en pratique), un point d'initialisation x_0, un nombre maximal d'itérations, un pas $\alpha > 0$ et qui renvoie la suite des premières itérations de la descente de gradient.
(2) Appliquer cet algorithme à la fonction $f : x \mapsto x^2 - x^4/2$ et illustrer visuellement les différents taux de convergence obtenus en fonction de la valeur du paramètre α.

Exercice 3.7. Montrer visuellement, sur la fonction $x \mapsto x\cos(x)$, que les méthodes de descente peuvent ne pas converger vers des minima locaux ; à cet effet, on représentera le graphe de la fonction et la suite de points générées par une descente de gradient à pas fixe pour $\alpha = 2$, $x_0 = 10$ et 5 itérations.

Exercice 3.8 (L'influence de la taille du pas). Pour illustrer l'importance de la taille du pas α, on peut considérer la fonction $f : x \mapsto x^2$ ainsi qu'une descente de gradient à pas fixe :

(1) Montrer, en traçant le graphe de la fonction et chacune des itérations donnée par la méthode de descente, que si α est trop petit ($\alpha \sim 10^{-3}$), il faut un nombre significatif d'itérations pour obtenir une bonne approximation de 0.
(2) Montrer à l'inverse que si $\alpha > 1$ la méthode de gradient diverge grossièrement- on représentera ici aussi graphiquement la suite de points obtenue par descente de gradient.

3.5. **En dimensions supérieures.**

3.5.1. *Coder le gradient.* Il est important d'importe les librairies `random` et `linalg`, et donc il vous faut taper `from numpy import random` et `from numpy import linalg` avant de commencer cette section. Dans ce TP, une variable $\mathbf{x} \in \mathbb{R}^d$ sera codée en Python par un array de taille d : on écrira `x = array([x1, x2, ..., xd])`. Ainsi, une fonction $F : \mathbb{R}^d \to \mathbb{R}$ sera codée par une fonction Python qui prend en entrée un array, et le gradient $\nabla F : \mathbb{R}^d \to \mathbb{R}^d$ prendra en entrée un array, et retournera un array de même dimension.

Exercice 3.9. Ecrire une fonction `gradientDFC(F,d,eps=1e-5)` qui prend une fonction $F : \mathbb{R}^d \to \mathbb{R}$, et qui renvoie la fonction $\nabla_\varepsilon F$, dont les composantes sont calculées par différences finies centrées, c'est à dire

$$(\nabla_\varepsilon F)_i : \mathbf{x} \mapsto \frac{F(\mathbf{x} + \varepsilon \mathbf{e}_i) - F(\mathbf{x} - \varepsilon \mathbf{e}_i)}{2\varepsilon}$$

On rappelle qu'on peut construire (par exemple) le vecteur $\mathbf{e}_i$ avec

```
ei = zeros(d)
ei[i] = 1
```

Exercice 3.10. Pour vérifier que votre code ne fait pas d'erreur, testez le sur la fonction $F : (x, y) \mapsto x^2 + y^2$ en un point (x_0, y_0) aléatoire, et vérifiez que vous obtenez une bonne approximation à 10^{-8} près.

3.5.2. *Régression linéaire.* Nous (re)visitons le problème de la régression linéaire. On cherche deux nombres (α_0, α_1) à partir de données bruitées $(x_i, y_i)_{1\leq i\leq M}$, avec $y_i = \alpha_0 + \alpha_1 x_i + w_i$ où w_i sont des variables aléatoires iid (bruit), avec $\mathbb{E}(w) = 0$. Dans le code suivant, on génère de telles données aléatoires (les x_i sont choisis aléatoirement dans $(-1, 1)$).

```
M = 100 # nombre de données
alpha0, alpha1 = 3,2 # les nombres à trouver

Xi = random.rand(M)*2-1 # M nombres aléatoires entre -1 et 1
Wi = random.rand(M)*2-1 # Le bruit
Yi = alpha0 + alpha1*Xi + Wi

# On affiche les valeurs
plt.plot(Xi,Yi,'o')
```

On suppose qu'on a accès aux données (X_i, Y_i). Notre but est de retrouver une approximation de $\tilde{A} := (\alpha_0, \alpha_1)$. Autrement dit, on cherche la meilleure approximation linéaire des données. Pour $A = (a_0, a_1) \in \mathbb{R}^2$, on pose

$$E(A) := \frac{1}{M}\sum_{i=1}^{M} |y_i - a_0 - a_1 x_i|^2.$$

Exercice 3.11. A étant vu comme un `array` de taille 2, écrire la fonction $E(A)$.

Exercice 3.12. Tracer 30 courbes de niveau de E sur $[0, 5] \times [0, 5]$. On pourra utiliser la fonction contour. Ce dessin vous parait-il cohérent avec notre problème ?

Minimisation de E par descente de gradient à pas constant

Dans cette section, on cherche à minimiser la fonction E avec le gradient à pas constant.

Exercice 3.13. Ecrire une fonction `gradientPasConstant(dF, x0, tau, tol=1e-6, Niter=1000)` qui prend une fonction $\nabla F : \mathbb{R}^d \to \mathbb{R}^d$, un point initial $x_0 \in \mathbb{R}^d$, et un pas $\tau > 0$, et qui renvoie le premier point $x_n \in \mathbb{R}^d$ tel que $\|\nabla F(x_n)\| < tol$, où x_n est définie par

$$x_{n+1} = x_n - \tau \nabla F(x_n).$$

L'algorithme devra aussi renvoyer la liste des points $[x_0, \cdots, x_{n-1}]$.

On voit désormais la fonction E comme une boîte noire, que l'on va minimiser à l'aide d'une descente de gradient à pas constant.

Exercice 3.14. Ecrire une fonction `RegLinGPC(tau, A0=array([0,0]))` qui fait tourner l'algorithme `gradientPasConstant` avec le gradient de E (on utilisera la fonction `gradientDFC` de la première partie), le pas τ, et l'initialisation $A_0 \in \mathbb{R}^2$. Le code devra renvoyer le point final $A_n \in \mathbb{R}^2$ et la liste $[A_0, A_1, \ldots A_{n-1}]$.

Exercice 3.15. Ecrire une fonction `plotCVGPC(tau)` qui prend un paramètre τ, et qui affiche

(1) les courbes de niveau de E.

(2) les points $A_k \in \mathbb{R}^2$ pour $k \in [\![0, n-1]\!]$. On pourra utiliser l'option `'o-'` de plot

(3) le nombre d'itérations dans le titre de la figure.

Tester votre fonction avec différentes valeurs de τ. Qu'observez-vous ? Que se passe-t-il si $\tau = 1$ ou $\tau = 2$?

Exercice 3.16. Afficher le nombre d'itérations que prend la méthode du gradient à pas constant en fonction de $\tau \in [0.1, 0.9]$. Comment expliquez-vous ce phénomène ? Quelle est la meilleure valeur de τ pour ce problème ?

Exercice 3.17. Est-ce qu'on retrouve une bonne approximation de $\tilde{A}$ avec la minimisation de la régression linéaire ? Afficher les données (X_i, Y_i) et la droite $y(x) = a_0 + a_1 x$ que vous obtenez, pour $x \in [-1, 1]$.

4. *Intermezzo* : calcul différentiel, conditions d'optimalité dans les problèmes de minimisation

Dans ce cours plus théorique nous mettons en place les ingrédients et les outils nécessaires à la mise en place des algorithmes de gradient en dimensions supérieures.

4.1. **Prolégomène.** Il faut faire très attention aux trois étapes essentielles de l'étude d'un problème d'optimisation :

(1) L'**existence** d'un minimiseur. Dans ce cours, l'outil essentiel est la coercivité.

(2) L'**identification des candidats à être de minimiseurs**. Dans ce cours, nous utiliserons des critères différentiels et nous identifierons des points critiques, ainsi que des minima locaux.

(3) La **caractérisation des minimiseurs**. Dans ce cours, nous utiliserons des hypothèses de convexité.

Il faut garder en tête que ces trois étapes sont très liées, mais peuvent être traitées dans plusieurs ordres possibles. Identifier des minima locaux sans être certain qu'ils existent, c'est une quasi garantie de foncer dans le mur.

4.2. **Rappels sur les formules de Taylor.** Dans toute la suite, f est une fonction C^2 de $\mathbb{R}^d$ dans $\mathbb{R}$. On note $\{e_k\}_{k=1,\dots,d}$ la base canonique de $\mathbb{R}^d$.

Soit $i \in \{1, \dots, d\}$. La i-ème dérivée partielle de f dans la direction e_i est définie (la limite étant supposée exister) par

$$\frac{\partial f}{\partial x_i} := \lim_{t\to 0} \frac{f(x+te_i) - f(x)}{t}.$$

Si l'on suppose que ces dérivées partielles sont continues en x, alors la fonction f est différentiable au sens classique : pour tout $x \in \mathbb{R}^d$, il existe une application $d_x f \in \mathcal{L}(\mathbb{R}^d, \mathbb{R})$ telle que

$$\forall y \in \mathbb{R}\,, f(x+y) = f(x) + d_x f(y) + o(\|y\|).$$

En particulier, une direction $v \in \mathbb{R}^d$ étant fixée, on a

$$d_x f(v) = \lim_{t\to 0} \frac{f(x+tv) - f(x)}{t},$$

limite qui existe si f est différentiable. Observons alors que

$$d_x f(e_i) = \frac{\partial f}{\partial x_i}. \tag{4.1}$$

Par le théorème de représentation de Riesz, on sait que l'application d_x s'identifie à un vecteur $N_x \in \mathbb{R}^d$, dont il suffit de déterminer les coordonnées dans la base canonique. Puisque

$$\langle N_x, e_i\rangle = d_x f(e_i) \text{ (par Riesz) } = \frac{\partial f}{\partial x_i} \text{ (par (4.1))}$$

on en déduit que

$$N_x = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

et l'on note habituellement ce vecteur ∇f. Le **gradient de f en x** est le vecteur

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

et l'on retiendra que pour une application différentiable f on a

$$f(x+h) = f(x) + \langle \nabla f(x), h \rangle + o\left(\|h\|\right).$$

Il est très important d'être à l'aise avec le gradient, qui interviendra dans les méthodes de descente mais également dans les méthodes de Newton.

Enfin, notons que l'on a, outre le développement limité ci-dessus, la formule de Taylor avec reste intégral

$$\forall x\,, h \in \mathbb{R}^d\,, f(x+h) = f(x) + \int_0^1 \langle \nabla f(x+th), h \rangle dt$$

On passe ensuite à la différentielle seconde. Si l'application $x \mapsto d_x f = \nabla f(x)$ (modulo une identification) est elle-même différentiable en un point $x \in \mathbb{R}^d$, on dit que f est deux fois différentiable en x, et on note $d_x^2 f$ cette différentielle seconde. De même que le gradient était associé à une forme linéaire, la différentielle seconde est associée à une **forme quadratique** [1]

$$d_x^2 f : (h_1, h_2) \mapsto d_x^2 f(h_1, h_2).$$

Il est à noter que, par le même type d'arguments que précédemment, on a une description matricielle de cette différentielle seconde : la différentielle seconde s'identifie à la matrice $\nabla^2 f(x)$ appelée **matrice Hessienne de f en x** [2] et donnée, pour tous indices $i, j \leq d$, par

$$\nabla^2 f(x)_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial x_j}\right).$$

Par "s'identifie", on entend que

$$\forall h_1\,, h_2 \in \mathbb{R}^d\,, d_x^2 f(h_1, h_2) = \langle \nabla^2 f(x) h_1, h_2 \rangle.$$

Dans le cas des fonctions deux fois différentiable en un point x, on a le développement de Taylor à l'ordre 2

$$f(x+h) = f(x) + \langle \nabla f(x), h \rangle + \frac{1}{2}\langle \nabla^2 f(x) h, h \rangle + o\left(\|h\|^2\right).$$

Un théorème fondamental est le **théorème de Schwarz** qui assure que, si f est deux fois différentiable, alors

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i},$$

de sorte que la Hessienne est symétrique.

On a en outre la formule de Taylor avec reste intégral :

$$\forall x\,, h \in \mathbb{R}^d\,, f(x+h) = f(x) + \langle \nabla f(x), h \rangle + \int_0^1 (1-t)\langle \nabla^2 f(x+th)(h)\,, h \rangle dt. \tag{4.2}$$

1. On rappelle qu'une forme quadratique est un polynôme homogène de degré 2.
2. Que l'on note parfois aussi $\mathrm{Hess}(f)(x)$, ou $H(f)(x)$.

Dans ce qui suit, on se préoccupe pas outre mesure des questions de différentiabilité des fonctions considérées, questions qui peuvent rapidement devenir cauchemardesques, et l'on travaillera toujours avec des fonctions de classe au moins C^2. Ceci nous permet en particulier d'identifier la différentielle d'une fonction f avec le gradient de la fonction f, et d'utiliser la hessienne de f pour faire des développements limités d'ordre 2.

4.3. **Identification des candidats : règles de Fermat.** Dans toute cette section, on fixe une fonction $f : \mathbb{R}^d \to \mathbb{R}$ deux fois différentiable et l'on étudie le problème d'optimisation

$$\min_{x\in\mathbb{R}^d} f(x). \tag{4.3}$$

On suppose que ce problème a une solution et on fixe $x^* \in \mathbb{R}^d$ tel que

$$f(x^*) = \min_{x\in\mathbb{R}^d} f(x).$$

En général, il est illusoire de pouvoir exhiber x^*, et l'on doit se contenter de restreindre notre recherche. L'objectif de ce cours, qui sert de fondement à la méthode de gradient, est d'obtenir des conditions locales qui permettent parfois de caractériser x^*.

Insistons sur un aspect très important : **ces conditions ne sont que des conditions locales** et ne permettent d'identifier que des **minima locaux**. Pour pouvoir dire qu'un minimum local est un **minimum global** il faut une information justement globale. Le cas le plus simple est celui des fonctions convexes.

4.3.1. *Critère de Fermat : condition d'optimalité d'ordre 1.* Commençons par le critère d'optimalité d'ordre 1. On insiste encore une fois sur le caractère **nécessaire** et non suffisant de cette règle :

Proposition 4.1 (Critère de Fermat)**.** *En un point de minimum x^* on a*

$$\nabla f(x^*) = 0.$$

Il s'agit exactement de l'analogue du critère unidimensionnel

$$x \text{ minimal pour } f : \mathbb{R} \to \mathbb{R} \ \Rightarrow f'(x) = 0.$$

Preuve de la Proposition 4.1. Dans cette preuve se trouve déjà l'idée de la descente de gradient.

On raisonne par l'absurde : si $\nabla f(x^*) \neq 0$, considérons le vecteur $h := -\frac{\nabla f(x^*)}{\|\nabla f(x^*)\|}$. Alors, on a

$$f(x + th) - f(x) = -t\|\nabla f(x^*)\| + o(t^2),$$

quantité strictement négative quand $t \to 0$. □

On introduit la définition suivante :

Définition 4.1. Soit $f \in C^1(\mathbb{R}^d; \mathbb{R})$. Un point $x \in \mathbb{R}^d$ est appelé **point critique de** f si

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

4.3.2. *Un exemple fondamental.* Un exemple fondamental de point critique est le suivant : si $A \in M_d(\mathbb{R})$ est une matrice et si $b \in \mathbb{R}^d$ un vecteur fixé, la fonction quadratique définie à partir de A et de b est l'application

$$f_{A,b} : x \mapsto \frac{1}{2}\langle x, Ax\rangle - \langle b, x\rangle.$$

Calculons le gradient de f en un point $x \in \mathbb{R}^d$ fixé : h étant une perturbation donnée, on a

$$\begin{aligned} f_{A,b}(x+h) = &\frac{1}{2}\langle x, Ax\rangle - \langle b, x\rangle \\ &+ \frac{1}{2}\langle h, Ax\rangle + \frac{1}{2}\langle x, Ah\rangle - \langle b, h\rangle \\ &+ \frac{1}{2}\langle h, Ah\rangle. \end{aligned}$$

Si l'on définit l'application linéaire

$$L : h \mapsto \frac{\langle Ax, h\rangle + \langle x, Ah\rangle}{2} - \langle b, h\rangle$$

on a donc

$$f_{A,b}(x+h) = f_{A,b}(x) + L(h) + \frac{1}{2}\langle h, Ah\rangle.$$

L est un bon candidat à être la différentielle de f au point x. Il faut pour garantir cela vérifier que le terme $\langle h, Ah\rangle$ est bien un $o(\|h\|)$. Or, rappelons la définition de la norme matricielle de A :

$$\|A\|_{\mathrm{op}} = \sup_{y \neq 0} \frac{\|Ay\|}{\|y\|}.$$

Ainsi, on a

$$\langle h, Ah\rangle \leq \|h\| \cdot \|Ah\| \leq \|h\|^2 \cdot \|A\|_{\mathrm{op}}$$

et on peut en conclure que la différentielle de $f_{A,b}$ en x est l'application L. Pour déterminer le gradient de f en x il suffit de déterminer l'unique vecteur $\xi \in \mathbb{R}^d$ tel que, pour tout $h \in \mathbb{R}^d$,

$$\langle \xi, h\rangle = L(h).$$

Or, remarquons que puisque, pour tout $y, z \in \mathbb{R}^d$ on a

$$\langle y, Az\rangle = \langle A^T y, z\rangle$$

on peut écrire que, pour tout $h \in \mathbb{R}^d$,

$$L(h) = \frac{\langle Ah, x\rangle + \langle Ax, h\rangle}{2} - \langle b, h\rangle = \left\langle \frac{A + A^T}{2}x - b, h \right\rangle.$$

En particulier,

$$\nabla f_{A,b}(x) = \frac{A + A^T}{2}x - b$$

et donc, si la matrice A est symétrique,

$$\nabla f_{A,b}(x) = Ax - b.$$

De la sorte, on obtient la remarque importante suivante : si $A \in S_d(\mathbb{R})$ et $b \in \mathbb{R}^d$, $x \in \mathbb{R}^d$ est un point critique de $f_{A,b}$ si, et seulement si, x est solution de

$$Ax = b.$$

En d'autres termes, si la fonction $f_{A,b}$ admet un minimum, ce minimum est la solution d'une équation linéaire. Ainsi, *on peut dans certains cas résoudre les systèmes linéaires par des méthodes d'optimisation.*

4.3.3. *Condition d'optimalité d'ordre 2.* Le problème des conditions d'ordre 1, c'est que même si elles permettent de restreindre l'espace où l'on cherche ces points critiques, elles ne permettent pas de distinguer, comme en dimension 1, les minima et les maxima locaux. Il nous faut donc chercher une information d'ordre 2, qui se trouve contenue dans la matrice Hessienne de f. Supposons que f est une fonction de classe C^2 dont la hessienne est continue.

Si l'on repart de la formule de Taylor à l'ordre 2, on voit qu'en un point critique x^* on a

$$\forall h\,, f(x^*+h) - f(x^*) = \frac{1}{2}\langle \nabla^2 f(x^*)h, h\rangle + o\left(\|h\|^2\right).$$

En particulier, pour que x^* soit un minimiseur, il est nécessaire que

$$\forall h \in \mathbb{R}^d\,, \langle \nabla^2 f(x^*)h, h\rangle \geq 0. \tag{4.4}$$

On a ainsi établi la proposition suivante :

Proposition 4.2. *En un point de minimum x^* de f on a*

$$\nabla f(x^*) = 0 \text{ et } \nabla^2 f(x^*) \in S_d^+(\mathbb{R})$$

où $S_d^+(\mathbb{R})$ désigne l'ensemble des matrices symétriques positives.

Insistons ici sur le fait qu'il s'agit d'une condition **nécessaire** d'optimalité locale, mais pas d'une condition **suffisante**. Pour s'en convaincre, il suffit de reprendre l'exemple

$$f : x \mapsto x^3$$

qui vérifie $f'(0) = 0\,, f''(0) = 0 \geq 0$, et pour laquelle 0 n'est pas un minimum local.

En revanche, on peut, en la renforçant, faire de cette condition nécessaire une condition suffisante de minimalité locale :

Proposition 4.3. *Soit x^* un point critique de f (i.e. $\nabla f(x^*) = 0$). On suppose que $\nabla^2 f(x^*) \in S_d^{++}(\mathbb{R})$, où $S_d^{++}(\mathbb{R})$ est l'ensemble des matrices symétriques définies positives. En d'autres termes, on suppose qu'il existe une constante $\lambda > 0$ telle que*

$$\forall h \in \mathbb{R}^d\,, \langle \nabla^2 f(x^*)h, h\rangle \geq \lambda \|h\|^2.$$

Alors x^* est un minimum local : il existe $\varepsilon > 0$ tel que

$$\forall x \in \mathbb{R}^d\,, \|x - x^*\| \leq \varepsilon \Rightarrow f(x) > f(x^*).$$

On laisse cette proposition à titre d'exercice.

Notons que bien évidemment la matrice $\nabla^2 f$ peut être définie négative, auquel cas x^* est un maximum local, et qu'elle peut n'être ni positive, ni négative : s'il existe $h_1\,, h_2 \in \mathbb{R}^d$ tels que

$$\langle \nabla^2 f(x^*)h_1, h_1\rangle > 0\,, \langle \nabla^2 f(x^*)h_2, h_2\rangle < 0$$

alors x^* n'est ni un maximum local, ni un minimum local, et l'on parle de **point selle**. Ceci signifie qu'il existe des directions de perturbations dans lesquelles f décroît, et d'autres où f croît. Si l'on prend par exemple la fonction

$$f : (x, y) \mapsto \frac{1}{2} \cdot (x^2 - y^2)$$

on voit que le point $(0,0)$ est un point selle. En effet, on a

$$\nabla^2 f(0,0) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Enfin, il se peut que la forme quadratique $\nabla^2 f$ soit nulle le long de certaines directions, en d'autres termes, qu'il existe $h \in \mathbb{R}^d, h \neq 0$ tel que

$$\langle \nabla^2 f(x^*)h, h\rangle = 0.$$

C'est typiquement le cas de la fonction

$$f : (x,y) \mapsto x^2 - y^4$$

dont la hessienne en 0 vaut

$$\nabla^2 f(0,0) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}.$$

Dans ce cas-là, on ne peut rien dire, et l'on doit, ou bien chercher des arguments d'ordre 3, ou bien poursuivre les calculs à la main.

4.3.4. *En pratique.* Néanmoins, en pratique, travailler avec des matrices symétriques définies positives générales peut s'avérer délicat, et il est certainement plus commode d'avoir affaire à des matrices diagonales.

Rappelons le théorème spectral :

Théorème 4.1 (Théorème spectral, admis)**.** *Toute matrice symétrique réelle est diagonalisable en base orthonormée réelle. En d'autres termes, il existe une matrice orthogonale $P \in O_d(\mathbb{R})$ (*i.e. $PP^T = I_d$*) et une matrice $D \in D_d(\mathbb{R})$, l'ensemble des matrices diagonales, telles que*

$$M = P^T D P.$$

On en déduit donc qu'une matrice symétrique est positive si, et seulement si, toutes ses valeurs propres sont positives, et qu'elle est définie positive si, et seulement si, toutes ses valeurs propres sont strictement positives. Néanmoins, le calcul d'une base de diagonalisation et des valeurs propres d'une matrice peut s'avérer très ardu en toute dimension. En dimension 2 cependant, on a un critère extrêmement simple, qui sera largement utilisé en Travaux Dirigés :

Proposition 4.4 (Critère pour l'appartenance à $S_2^+(\mathbb{R})$ et $S_2^{++}(\mathbb{R})$)**.** *Soit $M \in S_2(\mathbb{R})$. Si*

$$\det(M) > 0\,, \mathrm{Tr}(M) > 0$$

alors $M \in S_2^{++}(\mathbb{R})$.

Si

$$\det(M) \geq 0\,, \mathrm{Tr}(M) \geq 0$$

alors $M \in S_2^{+}(\mathbb{R})$.

Cette proposition se démontre facilement en passant par la base d'orthonormalisation donnée par le théorème 4.1. En dimensions supérieures, ce critère est évidemment faux : il suffit pour s'en convaincre de considérer la matrice

$$M := \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}.$$

4.4. **Unicité éventuelle des points critiques : notions de convexité.** Dans ce paragraphe, nous étudions une condition raisonnable pour avoir unicité des points critiques, la notion de convexité.

Définition 4.2. Une fonction $f : \mathbb{R}^d \to \mathbb{R}$ est **convexe** si

$$\forall x, y \in \mathbb{R}^d, \forall t \in [0;1], f((1-t)x + ty) \leq (1-t)f(x) + tf(y).$$

On dit que f est **strictement convexe** si l'inégalité ci-dessus est stricte pour tout $x \neq y$ et pour tout $t \in]0;1[$.

Cette définition, qui revient à demander que le graphe de la fonction f soit au dessus de chacune de ses tangentes, est une notion géométrique. Pour obtenir des caractérisations plus tractables, on doit supposer plus de régularité sur la fonction f. Quoiqu'une fonction convexe f soit lipschitzienne, et qu'elle soit en conséquence différentiable presque partout (Théorème de Rademacher), on va travailler directement avec des fonctions f de classe $\mathscr{C}^2$.

Il est bien connu qu'en dimension 1, une fonction dérivable f est convexe si, et seulement si, sa fonction dérivée f' est une fonction monotone croissante, ce qui est équivalent, dès que f est de classe $\mathscr{C}^2$, à la positivité de f''. En dimensions supérieures, les notions de convexité peuvent vite devenir pénibles à manipuler, mais nous avons la caractérisation suivante :

Proposition 4.5. *Soit $f \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R})$. Les trois propriétés suivantes sont équivalentes :*

i) f est convexe.

ii) ∇f est monotone au sens suivant :

$$\forall x, y \in \mathbb{R}^d, \langle \nabla f(y) - \nabla f(x), y - x \rangle \geq 0.$$

iii) Pour tout $x \in \mathbb{R}^d$, $\nabla^2 f(x) \in S_d^+(\mathbb{R})$.

Ce fait est classique et ne sera pas démontré ici.

(Stricte) positivité de la hessienne, (stricte) convexité de la fonction Se pose la question de savoir si l'on peut caractériser une fonction strictement convexe à l'aide d'une propriété de stricte positivité de sa hessienne. Une implication est claire, au vu du lemme suivant :

Lemme 4.1. *Soit $f \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R})$ telle que, pour tout $x \in \mathbb{R}^d$, $\nabla^2 f(x) \in S_d^{++}(\mathbb{R})$. Alors f est strictement convexe.*

La réciproque est néanmoins fausse : si l'on considère la fonction $x \mapsto x^4$, alors la fonction f est strictement convexe, et pourtant $f''(0) = 0$. On pourra objecter que c'est un exemple particulier, puisque la hessienne de f (ici sa dérivée seconde) ne s'annule qu'en un unique point. Néanmoins il est facile de construire, à partir des exercices du TD n°2, un exemple d'une fonction strictement convexe telle que f'' s'annule une infinité de foix. Si l'on définit

$$f : x \mapsto \frac{x^2}{2} - \cos(x)$$

alors il a été démontré dans le TD n°2 que $f'(x) = x + \sin(x)$ était une fonction strictement croissante. En particulier, f' est strictement croissante, et f est ainsi strictement convexe. Par ailleurs, $f''(x) = 1 + \cos(x)$ s'annule une infinité de fois.

Du point de vue de l'optimisation, la propriété fondamentale des fonctions convexes est la suivante :

Proposition 4.6. *Soit $f \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R})$ une fonction strictement convexe telle que, pour tout $x \in \mathbb{R}^d$, $\nabla^2 f(x) \in S_d^{++}(\mathbb{R})$. Alors f admet au plus un point critique x^*.*

Remarque 4.1. La stricte convexité est évidemment nécessaire, il suffit pour s'en rendre compte d'étudier la fonction

$$f : x \mapsto \begin{cases} e^{x^2} - 1 \text{ si } x \leq 0 \\ 0 \text{ si } 0 \leq x \leq 1 \\ e^{(x-1)^2} - 1 \text{ si } x \geq 1 \end{cases}$$

Alors tout point $x \in [0;1)$ est critique.

Par ailleurs, ce résultat ne donne évidemment pas de résultat d'existence des points critiques. En effet, si l'on considère la fonction $x \mapsto e^x$, c'est évidemment une fonction strictement convexe qui n'admet pas de point critique.

Ainsi, dans le cas d'une fonction strictement convexe, identifier un point critique revient à identifier l'unique candidat possible pour être un extremum local. En outre, pour une fonction suffisamment convexe et de classe $\mathscr{C}^2$, tout point critique x^* est un minimiseur global, comme indiqué par la proposition suivante :

Proposition 4.7. *Soit $f \in \mathscr{C}^2(\mathbb{R}^d;\mathbb{R})$ telle que, pour tout $x \in \mathbb{R}^d$, $\nabla^2 f(x) \in S_d^{++}(\mathbb{R})$. Si x^* est un point critique de f, alors x^* est un minimiseur global de f.*

Preuve de la Proposition 4.7. On applique la formule de Taylor avec reste intégral (4.2), et on utilise simplement le fait que la hessienne de f est toujours symétrique définie positive. □

Néanmoins, observons les choses suivantes :

(1) Il est possible qu'une fonction convexe n'admette pas de point critique, comme montré par l'étude de la fonction $x \mapsto e^{-x}$.

(2) Plus important, même si l'équation $\nabla f(x) = 0$ a une unique solution x^*, il ne s'ensuit pas que x^* est un extrémum local. Par exemple, la fonction $f : x \mapsto x^3$ a un unique point critique, qui n'est pas un extremum local.

4.5. **Problèmes de minimisation dans $\mathbb{R}^n$: existence de minimiseurs.** Dans ce dernier paragraphe, nous présentons finalement les conditions les plus simples à requérir de la fonction f pour que le problème d'optimisation

$$\inf_{x \in \mathbb{R}^d} f(x)$$

admette une solution. Dans tout ce qui suit, f est une fonction continue.

La méthode naturelle pour obtenir l'existence d'un minimiseur est d'employer ce qui dans le jargon est appelé **méthode directe du calcul des variations** ; derrière ce nom se cache simplement l'utilisation de la définition de l'infimum d'une fonction : en notant

$$m := \inf_{\mathbb{R}^d} f$$

on sait qu'il existe une suite $\{x_k\}_{k\in\mathbb{N}} \in \left(\mathbb{R}^d\right)^{\mathbb{N}}$ telle que

$$f(x_k) \underset{k\to\infty}{\to} m.$$

Puisque la fonction f est continue, il suffit pour obtenir l'existence d'un minimiseur d'obtenir l'existence d'une valeur d'adhérence x^* de la suite $\{x_k\}_{k\in\mathbb{N}}$. Mais l'existence d'une telle valeur d'adhérence est évidemment garantie si l'on sait *a priori*

que les éléments de la suite vivent dans un ensemble compact. Par exemple, ce n'est pas le cas si l'on prend la fonction $f : x \mapsto e^{-x}$: aucune suite minimisante n'a de valeur d'adhérence.

L'hypothèse de **coercivité** permet de répondre à cette question de manière synthétique :

Définition 4.3 (Fonction coercive)**.** On dit qu'une fonction continue $f : \mathbb{R}^d \to \mathbb{R}$ est **coercive** si, pour tout $M \in \mathbb{R}$, l'ensemble de niveau

$$\Omega_M := \{x\,, f(x) \leq M\}$$

est compact.

Puisque f est une fonction continue, Ω_M est compact si et seulement si il est borné, son caractère fermé étant évident.

On a la proposition suivante :

Proposition 4.8. *Soit $f \in \mathscr{C}^0(\mathbb{R}^d; \mathbb{R})$ une fonction coercive. Le problème d'optimisation*

$$\inf_{x \in \mathbb{R}^d} f(x)$$

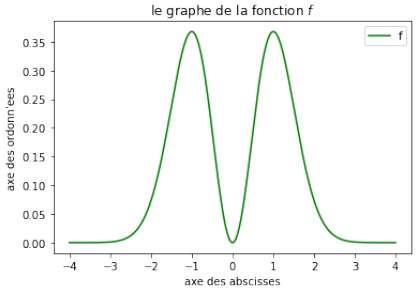
admet une solution x^*.

Preuve de la Proposition 4.8. On considère une suite minimisante $\{x_k\}_{k\in\mathbb{N}}$. Si x_0 n'est pas un minimiseur de f (auquel cas on aurait en fait terminé) alors, pour tout $k \in \mathbb{N}$ suffisamment grand, on a

$$f(x_k) \leq f(x_0).$$

En d'autres termes, pour tout k suffisamment grand, $\{x_k\}_{k\in\mathbb{N}, k\gg 1} \subset \Omega_{f(x_0)} := \left\{x \in \mathbb{R}^d\,, f(x) \leq f(x_0)\right\}$ qui, par hypothèse, est un ensemble compact. On peut donc extraire une sous-suite qui converge vers x^*. Puisque f est continue, x^* est un minimiseur de f. □

La coercivité est une condition suffisante pour avoir l'existence d'un minimiseur, mais elle n'est pas nécessaire : certaines suites minimisantes peuvent avoir des valeurs d'adhérence et d'autres non. C'est le cas par exemple de la fonction $f : x \mapsto x^2 e^{-x^2}$, dont le graphe est donné sur la figure suivante :



Un tableau récapitulatif. Si l'on récapitule tout ce que l'on a vu jusqu'à présent, on a le tableau suivant :

Propriété de la fonction	Propriété obtenue	Propriété manquante
Strictement convexe, $\mathscr{C}^1$	Unicité des points critiques	Existence des points critiques
Coercive	Existence d'un minimiseur	Caractérisation univoque des points critiques

Ainsi, on a le schéma suivant :

f coercive+f strictement convexe $\Rightarrow$ existence et unicité d'un minimiseur x^*.

4.6. **L'exemple paradigmatique.** L'exemple le plus courant et le plus utile de fonction coercive et strictement convexe est celui d'une fonction modelée sur une matrice symétrique définie positive : si $A \in S_d^{++}(\mathbb{R})$, si $b \in \mathbb{R}^d$ est un vecteur fixé, si l'on pose

$$f_{A,b} : x \mapsto \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$$

alors la fonction $f_{A,b}$ est strictement convexe, et elle est coercive.

(1) Preuve de la convexité : il est immédiat devoir que $\nabla^2 f = A$, et donc que f est strictement convexe.

(2) Preuve de la coercivité : la matrice A étant définie positive, sa plus petite valeur propre $\lambda_1(A)$ est strictement positive. En particulier, par l'inégalité de Cauchy-Schwarz,

$$\forall x \in \mathbb{R}^d, f_{A,b}(x) \geq \frac{\lambda_1(A)}{2}\|x\|^2 - \|b\| \cdot \|x\|,$$

de sorte que la fonction est coercive.

Si nous précisons cette fonction, c'est qu'elle est la plus important à partir du chapitre suivant, où nous attaquons les descentes de gradient et les algorithmes de gradient conjugué, en particulier pour les fonctions de la forme $f_{A,b}$.

FEUILLE DE TD N°4 : OPTIMISATION, CALCUL DIFFÉRENTIEL, CONVEXITÉ

Convexité.

Exercice 4.1 (Les grands classiques). (1) Inégalité de base Démontrer que

$$\forall (x_1, \dots, x_n) \in \mathbb{R}^n \,, \left(\prod_{k=1}^{n} x_k \right)^{\frac{1}{n}} \leq \frac{\sum_{k=1}^{n} x_k}{n}.$$

(2) Inégalité de Jensen Soit f une fonction continue et positive sur $[0;1]$, et ϕ une fonction continue et convexe. Démontrer que

$$\phi \left(\int_0^1 f \right) \leq \int_0^1 \phi(f).$$

(3) Entropie maximale On considère l'ensemble

$$\Delta := \left\{ x \in \mathbb{R}^d \,, \forall i \,, x_i > 0 \,, \sum_{i=1}^{d} x_i = 1 \right\}$$

et la fonction

$$H : \Delta \ni x \mapsto - \sum_{i=1}^{d} x_i \ln(x_i).$$

Résoudre le problème d'optimisation

$$\sup_{x \in \Delta} H(x).$$

Exercice 4.2 (Convexité et fonctions produits). Soit $f \in C^2(\mathbb{R}; \mathbb{R})$ telle que $\inf_{\mathbb{R}} f = \beta > 0$. On considère la fonction

$$\phi : \mathbb{R}^2 \ni (x, y) \mapsto \frac{1}{2} y^2 f(x).$$

Donner une condition simple sur f pour que ϕ soit convexe.

Calcul différentiel.

Exercice 4.3 (Principe du maximum). Soit Ω un ouvert de $\mathbb{R}^d$ borné et tel que $\partial\Omega$ est une surface de classe C^2. Soit $u \in C^2(\overline{\Omega})$ une fonction telle que

$$\Delta u \leq 0$$

dans Ω. On veut montrer qu'alors

$$\min_{\Omega} u = \min_{\partial\Omega} u.$$

On appelle ce résultat principe du maximum.

(1) Démontrer le principe du maximum en supposant dans un premier temp que $\Delta u < 0$ dans Ω.

(2) Le démontrer quand $\Delta u \leq 0$. *Indication : on pourra utiliser la fonction auxiliaire* $u_\varepsilon : x \mapsto -\varepsilon ||x||^2 + u(x)$.

(3) Ce résultat est-il toujours valable si l'on suppose désormais que pour un certain réel γ on a $\Delta u + \gamma u \leq 0$? Si oui, on fournira une preuve, si non, on fournira un contre-exemple en dimension 1 avec des paramètres bien choisis.

Exercice 4.4 (Une norme régulière est un produit scalaire). Soit $E := \mathbb{R}^2$ muni d'une norme quelconque $\|\cdot\|$. On supppose que l'application $\varphi : E \ni x \mapsto \|x\|^2$ est de classe $\mathscr{C}^2$. Montrer que cette norme est associée à un produit scalaire.

Indication : on pourra utiliser un développement de Taylor en $x = 0$.

Quelques problèmes d'optimisation.

Exercice 4.5. Soit f la fonction définie sur $\mathbb{R}^2$ par

$$f : (x, y) \mapsto x^4 + y^4 - 2(x - y)^2.$$

(1) Montrer qu'il existe $\alpha\,, \beta > 0$ tels que

$$\forall (x, y) \in \mathbb{R}^2\,, f(x, y) \geq \alpha\left(x^2 + y^2\right) - \beta.$$

(2) Montrer que le problème d'optimisation

$$\inf_{(x,y)\in\mathbb{R}^2} f(x, y) \tag{4.5}$$

a une solution.

(3) En déterminant les points critiques et en les classifiant (minima locaux, maxima locaux, points selles), résoudre (4.5).

Exercice 4.6 (Condition de Palais-Smale). Soit $F : \mathbb{R}^d \to \mathbb{R}$ une fonction de classe C^1 et $c \in \mathbb{R}$. On dit que F vérifie la *condition de Palais-Smale* en c s'il existe une suite (x_n) de $\mathbb{R}^d$ telle que

$$\lim_{n\to\infty} F(x_n) = c \quad \text{et} \quad \lim_{n\to\infty} \nabla F(x_n) = 0, \qquad (\mathbf{PS})$$

et si, pour toute suite (x_n) vérifiant (**PS**), (x_n) admet une sous-suite convergente. On note $S_\lambda := \left\{x \in \mathbb{R}^d,\ F(x) \leq \lambda\right\}$.

(1) Montrer que si F vérifie la condition de Palais-Smale en c, alors il existe un point critique $x^* \in \mathbb{R}^d$ de F tel que $F(x^*) = c$.

(2) Les fonctions suivantes vérifient-elles la condition de Palais-Smale en $c = 0$?

$$F_1(x) = x^2, \quad F_2(x) = \exp(x), \quad F_3(x) = \sin^2(x).$$

(3) On suppose que F est bornée inférieurement et on note $m := \inf F$.

(a) Montrer que si (x_n) est une suite telle que $F(x_n) \to m$ (*suite minimisante*), et que $\nabla F(x_n)$ ne converge pas vers 0, alors (x_n) ne converge pas.

(b) Montrer que s'il existe $\lambda > m$ tel que S_λ est compacte, alors m est un minimum de F, et F vérifie la condition de Palais-Smale en m.

Exercice 4.7. Soit $J : \mathbb{R}^2 \ni (x, y) \mapsto y^4 - 3xy^2 + x^2$.

(1) Déterminer les points critiques de la fonction J.

(2) Soit $(a, b) \in \mathbb{R}^2$. Montrer que $\xi = 0$ est un minimum local de l'application $\xi \mapsto J(\xi a, \xi b)$. En particulier, (0,0) est un minimum local le long de toute droite passant par 0.

(3) $(0, 0)$ est-il un minimum local de J le long de la parabole d'équation $y^2 = x$?

Correction de la feuille de TD n°4

Correction 4.1. *(1) Il s'agit simplement de l'inégalité de convexité appliquée à la fonction logarithme.*

(2) Il suffit d'appliquer l'inégalité de convexité aux sommes de Riemann définissant l'intégrale.

(3) La fonction logarithme est une fonction concave ; ainsi, pour tout $x \in \Delta$, on a

$$H(x) = \sum_{i=1}^{d} x_i \ln(x_i^{-1}) \leq \ln\left(\sum_{k=1}^{d} x_i x_i^{-1}\right) = \ln(n)$$

et l'on vérifie que le d-uplet constant $x_i = 1/d$ est bien un maximum.

Correction 4.2. *Un calcul direct montre que la hessienne de la fonction ϕ est donnée par*

$$\nabla^2\phi(x,y) = \begin{pmatrix} \frac{1}{2}y^2 f''(x) & yf'(x) \\ yf'(x) & f(x) \end{pmatrix}.$$

La condition de positivité de la trace donne

$$\forall (x,y) \in \mathbb{R}^2\,, \frac{1}{2}y^2 f''(x) + f(x) > 0.$$

À x fixé, en passant à la limite $y \to \infty$, on en déduit que $f'' \geq 0$ sur $\mathbb{R}$. La condition de positivité du déterminant donne quant à elle

$$\forall (x,y) \in \mathbb{R}^2\,, f''(x)f(x) > \frac{1}{2}\left(f'(x)\right)^2.$$

On en déduit la deuxième condition, mais également que $f'' > 0$ sur $\mathbb{R}$.

Correction 4.3.

Correction 4.4. *Puisque φ est de classe $\mathscr{C}^2$, on peut définir la forme quadratique $q : (x,y) \mapsto \langle \nabla^2\varphi(0)x, y\rangle$, et on a le développement limité*

$$\varphi(x) = q(x,x) + \underset{x\to 0}{o}\left(\|x\|^2\right).$$

Mais là on peut utiliser l'homogénéité de la norme : en fixant $y \in \mathbb{R}^d$ et en posant $x := \lambda y$ pour $\lambda \in \mathbb{R}$ on obtient

$$\lambda^2\|y\|^2 = \lambda^2 q(y,y) + \underset{\lambda\to 0}{o}\left(\lambda^2\right).$$

En passant à la limite $\lambda \to 0$ on en déduit déjà que q est une forme quadratique définie positive, et que

$$\|y\|^2 = q(y,y)$$

pour tout y, ce qui conclut la preuve.

Correction 4.5. *(1) On sait que, pour tout $(x,y) \in \mathbb{R}^2$ on a*

$$2xy \geq -(x^2+y^2).$$

En particulier, ceci implique que

$$f(x,y) = x^4 + y^4 - 2x^2 - 2y^2 + 4xy \geq x^4 + y^4 - 2x^2 - 2y^2 - 2x^2 - 2y^2$$

et donc que

$$f(x,y) \geq \left(x^4 - 4x^2\right) + (y^4 - 4y^2).$$

Or, si l'on considère

$$g : t \mapsto t^4 - 4t^2$$

on vérifie aisément que

$$g(t) \geq \alpha t^2 - \beta$$

pour deux constantes $\alpha\,, \beta$ *choisies de sorte que*

$$t^4 - (4-\alpha)t^2 + \beta$$

soit un polynôme positif. Il suffit de prendre $\alpha \in (0;4)$ *et* $\sqrt{\beta = \frac{(4-\alpha)}{2}}$ *pour obtenir une identité remarquable.*

(2) Par la première question, la fonction est coercive, et (4.5) *admet donc bien une solution.*

(3) Le gradient de f *en* (x,y) *est*

$$\nabla f(x,y) = 4\begin{pmatrix} x^3 - x + y \\ y^3 - y + x \end{pmatrix}.$$

Pour que (x_0, y_0) *soit un point critique on a nécessairement*

$$x_0^3 + y_0^3 = 0$$

et donc

$$y = -x$$

ce qui implique

$$x_0^3 - 2x_0 = 0.$$

Les seuls points critiques sont donc $(0,0)\,, (\sqrt{2}, -\sqrt{2})$ *et* $(-\sqrt{2}, \sqrt{2})$*. Puisque la fonction* f *est de classe* C^2 *on calcule sa Hessienne en ces points. En* $(x,y) \in \mathbb{R}^2$ *on a*

$$\nabla^2 f(x,y) = 4\begin{pmatrix} 3x^2 - 1 & 1 \\ 1 & 3y^2 - 1 \end{pmatrix}.$$

En $(0,0)$ *cette matrice vaut* $\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$ *et on ne peut donc rien conclure. Néanmoins, on peut démontrer à la main que* $(0,0)$ *n'est pas un minimum local, attendu que, pour* $\varepsilon > 0$ *suffisamment petit,* $f(\varepsilon, 0) \sim -2\varepsilon^2 < 0 = f(0,0)$*. Inversement,* $f(\varepsilon, -\varepsilon) = 2\varepsilon^4 > f(0,0)$*.* $(0,0)$ *est donc un point selle.*

En $(\sqrt{2}, -\sqrt{2})$ *la matrice hessienne vaut* $\begin{pmatrix} 5 & 1 \\ 1 & 5 \end{pmatrix}$*. Sa trace est positive, de même que son déterminant ; on a donc affaire à un minimum local strict.*

Correction 4.6. *(1) Soit* (0_n) *une suite telle que* $F(0_n) \to \mu$ *et* $\nabla F(0_n) \to 0$*. D'après la condition de Palais-Smale, il existe* $0^* \in \mathbb{R}^d$ *tel que* (0_n) *converge vers* 0^* *à extraction près. A la limite, on obtient, par continuité de* F *et de* ∇F*,* $F(0^*) = c$ *et* $\nabla F(0^*) = 0$*.*

(2) Pour F_1*, soit* (x_n) *vérifiant* $x_n^2 \to 0$*. Cela implique* $x_n \to 0$*, et la suite converge vers* 0*. Donc* F_1 *vérifie PS en* 0*. Pour* F_2*, la suite* $x_n := -n$ *vérifie* $e^{-x_n} \to 0$*, mais diverge en* $-\infty$*. Donc*F_2 *ne vérifie pas PS en* 0*. Pour* F_3*, la suite* $x_n := n\pi$ *vérifie* $F_3(x_n) = 0$ *et* $F_3'(x_n) = 0$ *(car minimum*

local). Mais x_n diverge, donc F_3 ne vérifie pas PS en 0.

(3) (a) Supposons par l'absurde que (x_n) converge vers x^*. Alors à la limite on a $F(x^*) = m$, et donc m est un minimum. En particulier, c'est un point critique, et $\nabla F(x^*) = 0$, ce qui contredit l'hypothèse que $\nabla F(x_n)$ ne converge pas 0. Donc la suite (x_n) ne converge pas.

(b) On a $\inf_{\mathbb{R}^d} F = \inf_{S_\lambda} F = \min_{S_\lambda} F$ (minimum d'une fonction continue sur un compact) $= \min_{\mathbb{R}^d} F$. Ainsi, m est le minimum de F. Soit (x_n) une suite telle que $F(x_n)$ converge vers m. Alors, à partir d'un certain rang, $x_n \in S_\lambda$, qui est compact. Ainsi, la suite (0_n) converge à extraction près.

Correction 4.7. *(1) Le gradient de J vaut*

$$\nabla J(x, y) = \begin{pmatrix} -3y^2 + 2x \\ 4y^3 - 6xy \end{pmatrix}.$$

On voit aisément que ceci implique que $(0, 0)$ est l'unique point critique de J.

(2) On voit que si l'on pose

$$f : \xi \mapsto J(\xi a, \xi b) = \xi^4 a^4 - 3\xi^3 ab + \xi^2 b^2$$

on a

$$f'(0) = 0\,, f''(0) = 2b^2.$$

Si $b \neq 0$, alors 0 est un minimum local. Si $b = 0$, on a directement $f(\xi) = \xi^4 a^4$ et 0 est donc un minimum global.

(3) En revanche, on a

$$J(y^2, y) = -2y^4$$

et $(0, 0)$ est donc un maximum le long de cette parabole.

5. Descente de gradient II : première analyse en dimensions supérieures

Nous étudions dans ce cours les méthodes de descente de gradient à pas constant, pour la minimisation de fonctions $f : \mathbb{R}^d \to \mathbb{R}$. Toutes les illustrations seront faites dans le cas $d = 2$, qui permet de visualiser ce qui se produit, mais l'étude théorique est valable en toute dimension.

5.1. **Heuristique de la méthode.**

5.1.1. *Choix de la direction de plus grande descente.* Nous l'avons vu dans le cadre de la descente de gradient unidimensionnel, l'idée est de suivre une direction dans laquelle la fonction décroît. En d'autres termes, partant d'un point initial x_0, on veut choisir une direction $v \in \mathbb{S}^{d-1}$ et un pas $\tau > 0$ tel que

$$f(x_0 + \tau v) < f(x_0).$$

Si en dimension 1 on n'avait le choix qu'entre "aller à gauche" et "aller à droite", en dimensions supérieures il est nécessaire de choisir entre d directions possibles. Déterminer cette direction est plutôt aisé : si l'on suppose que l'on part d'un point x_0 et qu'un petit pas $\tau > 0$ est donné, si l'on prend l'approximation

$$f(x_0 + \tau v) \approx f(x_0) + \tau \langle v, \nabla f(x_0) \rangle$$

et que l'on a envie de faire décroître la fonction f le plus rapidement possible, on voit que l'on doit choisir comme direction v la solution du problème d'optimisation

$$\inf_{v \in \mathbb{S}^{d-1}} \langle v \, , \nabla f(x_0) \rangle. \tag{5.1}$$

Si $\nabla f(x_0) = 0$, en d'autres termes, si l'on se trouve en un point critique, alors ce problème est trivial. Si en revanche $\nabla f(x_0) \neq 0$, l'inégalité de Cauchy-Schwarz assure que l'unique solution v^* de (5.1) est

$$v^* := -\frac{\nabla f(x_0)}{\|\nabla f(x_0)\|}.$$

Il semble donc pertinent de choisir cette direction-là et de s'y déplacer. Autrement dit, nous créerons une suite d'itérations de la manière suivante :

$$\forall k \in \mathbb{N} \, , x_{k+1} = x_k - \tau \nabla f(x_k).$$

Attention : dans tout ce cours, nous supposerons la taille du pas fixe, c'est-à-dire indépendante de l'itération. Dans les cours suivants, nous nous intéresserons à des descentes de gradient à pas variables.

5.1.2. *Interprétation géométrique de la direction de plus grande descente.* Pour déterminer la direction dans laquelle nous allons nous déplacer et représenter, géométriquement, cette direction, il nous faut donner une interprétation géométrique des ensembles de niveau, et du gradient de la fonction f.

Définition 5.1 (Surfaces de niveau)**.** Si $\lambda \in \mathbb{R}$ est un réel fixé, la **surface de niveau λ de f** est l'ensemble $\Sigma_\lambda := f^{-1}(\{\lambda\})$. En d'autres termes,

$$\Sigma_\lambda = \{x \in \mathbb{R}^d \, , f(x) = \lambda\}.$$

On a déjà vu en travaux pratiques comment représenter ces surfaces de niveau pour une fonction de deux variables. L'observation qualitative clé est la suivante : **la direction de plus grande descente est orthogonale aux lignes de niveau** de la fonction f. Ceci s'exprime par la proposition suivante :

Proposition 5.1. *Pour tout $\lambda \in \mathbb{R}$, pour tout $x \in \Sigma_\lambda$ tel que $\nabla f(x) \neq 0$, le vecteur $\nabla f(x)$ est perpendiculaire à la surface de niveau Σ_λ.*

Preuve de la proposition 5.1. Il suffit de remarquer que pour toute application $\gamma : \mathbb{R} \to \Sigma_\lambda$ de classe $\mathscr{C}^1$, la différentiation de l'équation $f(\gamma(t)) = \lambda$ implique

$$\langle \nabla f(\gamma(t)), \gamma'(t) \rangle = 0;$$

en particulier, le gradient est orthogonal à tous les vecteurs tangents à la surface, ce qui est la définition de l'orthogonalité. □

5.2. **Description de l'algorithme.** Une fois ces quelques petites idées mises en place, on peut passer à la définition de l'algorithme ; cet algorithme prend en argument une point initial x_0, un pas $\tau > 0$ fixé, un nombre maximal d'itérations `Niter` et une tolérance `tol`.

La structure typique de l'algorithme est donc la suivante : Notez qu'ici on a

Algorithm 2 Structure type de l'algorithme de direction de plus grande descente

```
def algo(f,df,x0,alpha,tol=1e-3,Niter=10)
Initialisation
xn=x0   On définit le premier élément de la liste
L=[]   On définit une liste que l'on augmentera récursivement, et qui contiendra
tous les éléments générés
for n in range(Niter):
if ...< tol : then
  return xn,L
else
  L.append(xn)
  xn=f(xn)-alpha*df(xn)
end if
print("Erreur, l'algorithme n'a pas convergé
après",IterMax,"itérations")
```

implicitement supposé que le gradient de f, noté *df*, était explicitement connu. Dans le TP n°3, nous verrons comment coder le gradient discrétisé quand l'expression explicite du gradient n'est pas accessible.

5.3. **Convergence de l'algorithme : le cas quadratique.** Nous divisons l'étude de la convergence de l'algorithme en deux parties. La philosophie est la suivante : si l'on se rappelle qu'en dimension 1 le caractère **non-dégénéré** des minima locaux de f (en d'autres termes, l'hypothèse que $f''(x^*) > 0$) nous permettait d'obtenir des vitesses et des taux de convergence, on se doute que la non-dégénerescence des minima de f en dimensions d, c'est-à-dire le caractère défini positif de la hessienne $\nabla^2 f(x^*)$ va également être crucial. Localement, au voisinage de x^*, nous pourrons

approcher f par son développement de Taylor à l'ordre 2, c'est-à-dire que nous utiliserons des développements qui disent essentiellement que

$$f(x) \approx \frac{1}{2}\langle \nabla^2 f(x^*)(x-x^*), (x-x^*)\rangle.$$

Il est donc naturel de commencer l'étude de la convergence de l'algorithme de descente de gradient à pas fixe dans le cas d'une fonction quadratique.

5.3.1. *Une première analyse de la convergence : identification des éléments constitutifs.* Dans toute la suite de ce paragraphe, on fixe une matrice $A \in S_d^{++}(\mathbb{R})$, un vecteur $b \in \mathbb{R}^d$ et l'on définit

$$f_{A,b} : x \mapsto \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

Nous avons vu lors du cours précédent que cette fonction était strictement convexe. En outre, son gradient est

$$\nabla f_{A,b}(x) = Ax - b$$

et sa hessienne

$$\nabla^2 f_{A,b}(x) = A.$$

Remarque 5.1 (Descente de gradient et résolution de systèmes linéaires)**.** Un autre intérêt de la minimisation des fonctions de la forme $f_{A,b}$ par descente de gradient est le suivant : le minimiseur x^* de $f_{A,b}$ (qui existe, par coercivité, et qui est unique, par convexité) est l'unique solution de

$$Ax^* = b.$$

Les algorithmes de gradient peuvent en pratique s'avérer beaucoup plus rapide que l'implémentation d'un pivot de Gauss pour la résolution de ce système linéaire.

De la remarque 5.1 on retient que le minimiseur x^* de $f_{A,b}$ est la solution de

$$Ax^* = b \Leftrightarrow x^* = A^{-1}b.$$

Identifions désormais les éléments constitutifs de la preuve de convergence de l'algorithme. On sait que si l'on se fixe un pas $\tau > 0$ la suite des itérés est donnée par

$$x_{k+1} = x_k - \tau A x_k + \tau b$$

de sorte que

$$x_{k+1} - x^* = x_k - \tau A x_k + \tau A x^* - x^* = x_k(\mathrm{I}_d - \tau A) - (\mathrm{I}_d - \tau A)x^* = (\mathrm{I}_d - \tau A)\,(x_k - x^*)\,.$$

Si l'on veut utiliser les critères de convergence linéaire ou quadratique vus lors du premier cours, il est donc nécessaire de pouvoir contrôler la **norme d'opérateur** de la matrice $\mathrm{I}_d - \tau A$.

On rappelle à toutes fins utiles la définition de la norme d'opérateur d'une matrice :

Définition 5.2 (Norme d'opérateur)**.** Soit $M \in M_d(\mathbb{R})$. La norme d'opérateur de M est

$$\|M\|_{\mathrm{op}} := \sup_{x \in \mathbb{R}^d \setminus \{0\}} \frac{\|Mx\|}{\|x\|}.$$

Dans le cas des matrices symétriques réelles, cette norme d'opérateur est contrôlée par les valeurs propres de la matrice M. On rappelle la proposition suivante, qui sera démontrée en travaux dirigés :

Proposition 5.2. *Soit $M \in S_d(\mathbb{R})$ et soient $\lambda_1(M) \leq \lambda_2(M) \leq \cdots \leq \lambda_d(M)$ ses valeurs propres. Alors on a :*

(1) Principe de Courant-Fisher :

$$\lambda_1(M) = \inf_{\|x\|=1} \langle Mx, x\rangle \text{ et } \lambda_d(M) = \sup_{\|x\|=1} \langle Mx, x\rangle.$$

(2) Norme d'opérateur et valeurs propres : la norme d'opérateur de la matrice M est donnée par

$$\|M\|_{\text{op}} = \max\left(|\lambda_1(M)|, |\lambda_d(M)|\right).$$

Le résultat de cette analyse, c'est qu'une maîtrise des rudiments de l'algèbre linéaire est nécessaire avant d'aller plus loin dans l'analyse des algorithmes de minimisation.

Revenons au problème de la convergence de la descente de gradient à pas constant τ. On rappelle que

$$x_{k+1} - x^* = (\mathrm{I}_d - \tau A)(x_k - x^*),$$

de sorte que

$$\|x_{k+1} - x^*\| \leq \max\left(|1 - \tau\lambda_1(A)|, |1 - \tau\lambda_d(A)|\right) \|x_k - x^*\|.$$

On en déduit que la suite $\{x_k\}_{k\in\mathbb{N}}$ converge vers x^* si

$$\max\left(|1 - \tau\lambda_1(A)|, |1 - \tau\lambda_d(A)|\right) < 1,$$

ce qui implique

$$\tau\lambda_d(A) < 2.$$

On a donc établi le théorème suivant :

Théorème 5.1. *[Convergence de la descente de gradient à pas fixe] La descente de gradient à pas fixe converge linéairement vers un minimiseur x^* de $f_{A,b}$ si le pas τ vérifie*

$$\tau\lambda_d < 2.$$

Cette convergence linéaire est à taux

$$\alpha(\tau) := \max\left(|1 - \tau\lambda_1(A)|, |1 - \tau\lambda_d(A)|\right).$$

On a donc obtenu un résultat général, qui donne un taux de convergence explicite. C'est une convergence lente que l'on peut essayer d'optimiser. Pour cela, il suffit de choisir un τ qui minimise $\alpha(\tau)$, en d'autres termes, de déterminer τ^* tel que

$$\alpha(\tau^*) = \min \alpha(\tau).$$

Or, la fonction α est définie comme le maximum de deux fonctions. La première de ces fonctions vaut

$$\beta_1(\tau) = |1 - \tau\lambda_1(A)|$$

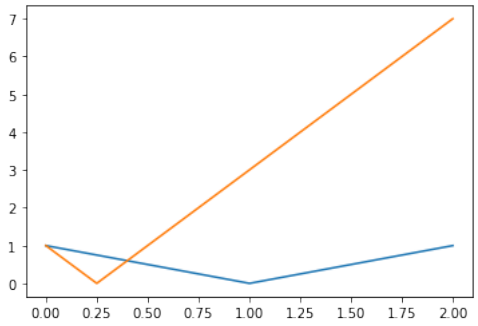
et la seconde

$$\beta_d(\tau) = |1 - \tau\lambda_d(A)|.$$

Pour $k = 1, d$, la fonction β_k est décroissante sur $[0; 1/\lambda_k(A)]$ puis croissante.

Le pas qui assure un taux optimal est

$$\tau^* = \frac{2}{\lambda_1 + \lambda_d}$$



qui correspond à un taux

$$\alpha(\tau^*) = \frac{\lambda_d(A) - \lambda_1(A)}{\lambda_d(A) + \lambda_1(A)}.$$

Représentons graphiquement la fonction $\max(\beta_1, \beta_d)$ dans le cas $\lambda_1(A) = 1$ et $\lambda_d(A) = 4$:

On voit que le minimum du maximum de ces deux fonctions est atteint au point τ^* où

$$\beta_1(\tau^*) = \beta_d(\tau^*) \Leftrightarrow (1 - \tau^*\lambda_1(A))^2 = (1 - \tau^*\lambda_d(A))^2.$$

En développant ces deux polynômes on voit qu'il faut avoir

$$-\frac{2}{\tau^*}(\lambda_1(A) - \lambda_d(A)) = \lambda_d(A)^2 - \lambda_1(A)^2 \Leftrightarrow \tau^* = \frac{2}{\lambda_d(A) + \lambda_1(A)}.$$

Pour ce τ^* optimal on a

$$\alpha(\tau^*) = \frac{\lambda_d(A) - \lambda_1(A)}{\lambda_1(A) + \lambda_d(A)}.$$

Par cette petite analyse nous avons établi le théorème suivant :

Théorème 5.2 (Convergence de la descente de gradient à pas fixe optimal)**.** *Le pas* $\tau^* = \frac{2}{\lambda_1(A)+\lambda_d(A)}$ *est optimal pour la descente de gradient à pas fixe. L'algorithme converge linéairement vers un minimiseur au taux (optimal)*

$$\alpha(\tau^*) = \frac{\lambda_d(A) - \lambda_1(A)}{\lambda_d(A) + \lambda_1(A)}.$$

5.4. **Petit aparté : conditionnement des matrices.** Dans le théorème précédent, nous avons établi le taux de convergence optimal pour une descente de gradient à pas fixe. On voit que, pour que la descente de gradient converge bien, il faut que $\lambda_d(A)$ et $\lambda_1(A)$ soient du même ordre de grandeur, autrement dit que

$$\frac{\lambda_d(A)}{\lambda_1(A)} \approx 1.$$

Le quotient de ces deux valeurs propres porte un nom :

Définition 5.3 (Conditionnement d'une matrice)**.** Soit $A \in S_d^{++}(\mathbb{R})$. Le **nombre de conditionnement** de la matrice A est la quantité

$$\mathrm{cond}(A) := \frac{\lambda_d(A)}{\lambda_1(A)}.$$

Cette notion sera vue sous toutes les coutures dans la feuille de TD n°5 mais retenons qu'en général plus la matrice A est mal conditionnée plus la convergence de l'algorithme de gradient sera mauvaise.

Remarque 5.2. En notant $\|\cdot\|_2$ la norme matricielle induite par la norme euclidienne sur $\mathbb{R}^d$ on a l'expression alternative

$$\mathrm{cond}(A) = \|A\|_2 \cdot \|A^{-1}\|_2.$$

Cette simple réecriture permet de généraliser la notion de conditionnement à celle de "conditionnement induit par une norme sur $\mathbb{R}^d$".

Donnons une inteprétation géométrique du mauvais conditionnement d'une matrice, en fonction des lignes de niveau, en considérant, en deux dimensions, une matrice $A \in S_2^{++}(\mathbb{R})$ et

$$f_{A,0} : \mathbb{R}^2 \ni x \mapsto \frac{1}{2}\langle Ax, x\rangle.$$

On peut, à changement de base orthonormale près, supposer que A est une matrice diagonale :

$$A = \begin{pmatrix} \lambda_1(A) & 0 \\ 0 & \lambda_2(A) \end{pmatrix}.$$

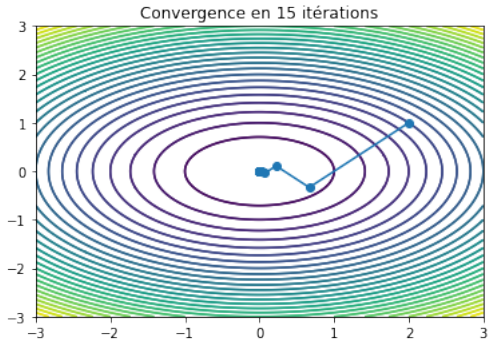
L'ensemble de niveau disons $r_0 > 0$ de la fonction $f_{A,0}$ est donné par

$$\Sigma(A, r_0) = \{(x_1, x_2) \in \mathbb{R}^2\,, \lambda_1(A)x_1^2 + \lambda_2(A)x_2^2 = r_0\}.$$

On reconnaît l'équation d'une ellipse d'excentricité $\sqrt{1 - \frac{\lambda_1(A)^2}{\lambda_d(A)^2}}$. Voyons ce que cela donne géométriquement en représentant les ensembles de niveau de la fonction $f_{A,b}$: dans tous les exemples suivant on prend $b = 0$.

On voit qu'un mauvais conditionnement correspond à des lignes de niveau plus aplaties. Ainsi, on peut interpréter le conditionnement comme le "degré d'applatissement" des ensembles de niveau de la fonction. Plus ils sont homogènes (*i.e.* proche d'un cercle) meilleure est la convergence de l'algorithme. À l'inverse, plus ils sont plats, plus la convergence est désastreuse.

5.5. **Illustration numérique du bon et du mauvais conditionnement d'une matrice.** Numériquement, on observe bien ce phénomène : voici la suite des itérations pour les paramètres $\lambda_1(A) = 1, \lambda_2(A) = 2$ (donc une matrice plutôt bien conditionnée), $b = 0$ et un point d'initialisation $x_0 = (2, 1)$:



Si on prend maintenant $\lambda_1(A) = 1$ et $\lambda_2(A) = 30$ (donc, une matrice très mal conditionnée) on observe un ralentissement dramatique :

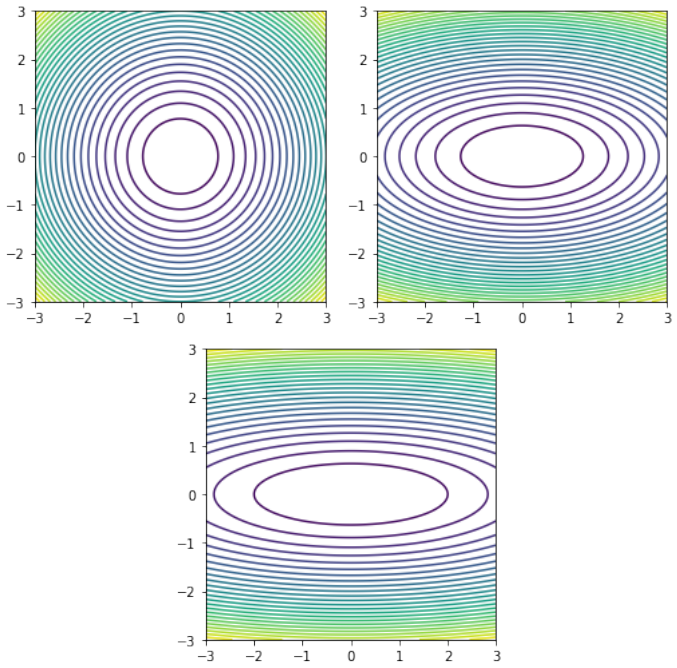
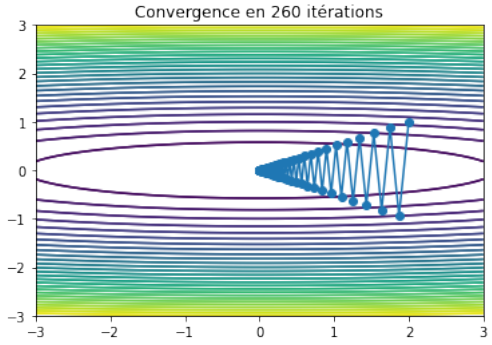


FIGURE 1. De droite à gauche : $\lambda_1(A) = \lambda_2(A) = 1$, $\lambda_1(A) = 1\,, \lambda_2(A) = 4$, $\lambda_1(A) = 1\,, \lambda_2(A) = 10$.



5.6. **Convergence de l'algorithme : le cas général.** Une petite précision : quand nous disons "cas général" nous parlons simplement d'un cas plus général que celui évoqué précédemment, qui est celui d'une fonction non-linéaire, mais dont les minima locaux sont dégénérés au sens vu au cours précédent.

Autrement dit, le problème d'intérêt ici est le suivant : on considère une fonction $f \in \mathscr{C}^3(\mathbb{R}^d; \mathbb{R})$, un point x^* qui est un **minimum local non dégénéré de** f et l'on travaille sur la suite des itérations fournie par la méthode de descente de gradient issue d'une point initial x_0.

On a en particulier supposé que

$$\nabla f(x^*) = 0\,, \nabla^2 f(x^*) \in S_d^{++}(\mathbb{R}).$$

La première idée qui vient (et qui fonctionne) pour établir la convergence de l'algorithme est la suivante (nous énoncerons le théorème recherché au terme de notre analyse) : on écrit que pour tout $x, y \in \mathbb{R}^d$ on a

$$\nabla f(y) = \nabla f(x) + \int_0^1 \nabla^2 f(x + t(y-x)) \cdot (y-x) dt.$$

En particulier, avec $x = x^*$ on obtient

$$\nabla f(y) = \int_0^1 \nabla^2 f(x^* + t(y-x^*)) \cdot (y-x^*) dt.$$

Grâce à cette réécriture, observons que la suite des itérations de la méthode de descente de gradient vérifie, pour tout $k \in \mathbb{N}$,

$$x_{k+1} - x^* = (x_k - x^*) - \tau \nabla f(x_k) = (x_k - x^*) - \tau \int_0^1 \nabla^2 f(x^* + t(x_k - x^*)) \cdot (x_k - x^*) dt.$$

Cette expression se réécrit, matriciellement

$$x_{k+1} - x^* = \left(\mathrm{I}_d - \tau \int_0^1 \nabla^2 f(x^* + t(x_k - x^*)) dt \right) (x_k - x^*) \, .$$

On introduit la matrice

$$M_k := \mathrm{I}_d - \tau \int_0^1 \nabla^2 f(x^* + t(x_k - x^*)) dt.$$

On voudrait utiliser les informations que l'on a sur $\nabla^2 f(x^*)$. On utilise alors le lemme clé suivant :

Lemme 5.1. *Pour tout $A, B \in S_d(\mathbb{R})$, on a*

$$\max(|\lambda_1(A) - \lambda_1(B)|, |\lambda_d(A) - \lambda_d(B)|) \leq \|A - B\|_{\mathrm{op}}.$$

Preuve du Lemme 5.1. Soit x_B un vecteur propre de norme 1 de la matrice B associé à la valeur propre $\lambda_1(B)$. Alors

$$\lambda_1(A) \leq \langle Ax, x \rangle = \langle (A-B)x, x \rangle + \lambda_1(B) \leq \|A - B\|_{\mathrm{op}} \cdot \|x\|^2 + \lambda_1(B),$$

ce qui conclut la preuve. En utilisant, inversement, la formulation de $\lambda_d(B)$ comme un quotient de Rayleigh, on obtient la même conclusion pour la d-ième valeur propre. □

Mais cette continuité pour la norme d'opérateur des valeurs propres est alors cruciale ! En effet, notons

$$\ell_0(x^*) := \frac{\lambda_1(\nabla^2 f(x^*))}{2}, \ell_1(x^*) := 2\lambda_d(\nabla^2 f(x^*)).$$

Alors, par le Lemma 5.1 on en déduit qu'il existe $\varepsilon > 0$ tel que, pour tout $x \in \mathbb{B}(x^*, \varepsilon)$,

$$\ell_0(x^*) \leq \lambda_1(\nabla^2 f(x)) \leq \lambda_d(\nabla^2 f(x)) \leq \ell_1(x^*).$$

En particulier, si l'on pose

$$M(x) := 1 - \tau \int_0^x \nabla^2 f(x^* + t(x - x^*)) dt$$

on récupère que, pour le même $\varepsilon > 0$, et pour tout $x \in \mathbb{B}(x^*, \varepsilon)$, on a

$$1 - \tau \ell_1(x^*) \leq \lambda_1(M(x)) \leq \lambda_d(M(x)) \leq 1 - \tau \ell_0(x^*),$$

et ainsi on en déduit que, si $x_k \in \mathbb{B}(x^*, \varepsilon)$,

$$\|x_{k+1} - x^*\| \leq \max\left(|1 - \tau\ell_1(x^*)|, |1 - \tau\ell_0(x^*)|\right) \|x_k - x^*\|.$$

Ainsi, si $\tau < \frac{1}{\ell(x^*)}$ on récupère bien un taux de convergence linéaire de la descente de gradient.

Bien évidemment, on peut préciser toutes ces informations comme nous l'avons fait auparavant dans le cas d'une fonction quadratique, et l'on s'attend, quand on optimise le pas, à avoir une convergence à taux

$$\frac{\lambda_d(\nabla^2 f(x^*)) - \lambda_1(\nabla^2 f(x^*))}{\lambda_1(\nabla^2 f(x^*)) + \lambda_1(\nabla^2 f(x^*))}.$$

On retrouve encore une fois le problème du conditionnement de la hessienne $\nabla^2 f(x^*)$.

On a donc établi le théorème suivant :

Théorème 5.3. *Soit $f \in \mathscr{C}^2(\mathbb{R}^d; \mathbb{R})$ et $x^* \in \mathbb{R}^d$ un minimum local de f. On suppose que $\nabla^2 f(x^*) \in S_d^{++}(\mathbb{R})$. Il existe $\varepsilon > 0$ tel que, si $x_0 \in \mathbb{B}(x^*, \varepsilon)$ et si $\tau < \frac{2}{\lambda_d(\nabla^2 f(x^*))}$, la descente de gradient à pas τ converge linéairement vers x^*.*

5.7. **La question de la taille des pas et des directions de descente.** Remarquons que, jusqu'ici, nous avons considéré une version plutôt simple de l'idée générale des méthodes de descente : nous avons fixé un pas taux, et nous avons simplement effectué le pas $-\tau\nabla f(x_k)$. Cette méthode n'est peut-être pas la plus intelligente, et il se peut que l'on choisisse, à chaque étape, une nouvelle direction d_k dans la quelle se diriger, et une taille de pas α_k. On peut espérer que ceci donne de meilleures convergences des algorithmes considérés. Pour ces raisons, posons la définition suivante :

Définition 5.4 (Direction de descente)**.** Soit $f \in \mathscr{C}^1(\mathbb{R}^d; \mathbb{R})$. Soit $x \in \mathbb{R}^d$. Un vecteur $d \in \mathbb{R}^d$ est une **direction de descente** de f en x si

$$\langle \nabla f(x), d \rangle < 0.$$

Une fois que l'on est un point x et que l'on veut minimiser la fonction f, il faut choisir une direction de descente et une taille de pas. Il faut bien sûr trouver un compromis : si on prend un pas τ très petit, alors passer de x à $x + \tau d$ améliorera certainement le critère, mais la convergence risque d'être extrêmement lente si l'on n'y prend pas garde, comme nous l'avons montré dans le cas des fonctions quadratiques. À l'inverse, si l'on prend un pas trop grand, on peut ne plus améliorer du tout le critère... Il faut donc trouver un compromis entre ces deux exigences. C'est l'objectif du cours suivant que de présenter une batterie de méthodes permettant de satisfaire à ces deux exigences.

5.8. **Une dernière remarque sur les critères d'arrêt.** Une dernière remarque, qui a son importance, et qui repose également sur le conditionnement de la matrice A : mettons que l'on considère une fonction quadratique de la forme

$$f : x \mapsto \frac{1}{2}\langle Ax, x \rangle - \langle b, x \rangle.$$

Comme toujours, on autorisera notre ordinateur à effectuer un nombre maximal d'itérations mais on aimerait s'assurer que, ou bien la solution trouvée soit "satisfaisante", ou bien que l'algorithme s'arrête s'il a déjà trouvé une solution satisfaisante.

Une idée, pour ce faire, est de dire que "satisfaisant" signifie $\|\nabla f(x)\| < \varepsilon$ où $\varepsilon > 0$ est un (petit) paramètre fixé à l'avance. Mais ce critère, très naturel, répond-il à nos exigences ? En général, **non**, et il faut le manier avec beaucoup de délicatesse. En effet, dans ce cas, le gradient de f vaut

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

Mais, si la matrice A est mal conditionnée, il se peut que $\|Ax - b\|$ soit faible, et que $\|x - x^*\|$ soit, au contraire, très important. On y prendra donc gare, quand on implémentera de tels critères d'arrête en séances de travaux pratiques.

Fiche de TD n°5 : descente de gradient, valeurs propres, conditionnement

Échauffement.

Exercice 5.1 (Retour en dimension 1). Soient $a, b \in \mathbb{R}^2$, $a > 0$, et soit $f : x \mapsto a\frac{x^2}{2} - bx$. On se fixe un pas de descente $\tau > 0$.

(1) Écrire la suite générée par un descente de gradient issue du point $x_0 = 1$.

(2) Montrer à la main que cette suite converge si et seulement si $\tau a < 2$. Quel est le taux de convergence ?

(3) Que se passe-t-il si $\tau = \frac{1}{a}$?

Autour des valeurs propres.

Exercice 5.2 (Principe de Courant-Fisher, caractérisation variationnelle des valeurs propres). Soit $A \in S_d(\mathbb{R})$.

(1) En notant $\mathbb{S}^{d-1}$ la sphère unité de $\mathbb{R}^d$, montrer que les problèmes d'optimisation
$$\inf_{x\in\mathbb{S}^{d-1}} \langle Ax, x\rangle\,, \quad \sup_{x\in\mathbb{S}^{d-1}} \langle Ax, x\rangle$$
ont des solutions.

(2) Montrer que si $\lambda_1(A) \leq \cdots \leq \lambda_d(A)$ est la suite des valeurs propres ordonnées de A, on a
$$\lambda_1(A) = \inf_{x\in\mathbb{S}^{d-1}} \langle Ax, x\rangle\,, \lambda_d(A) = \sup_{x\in\mathbb{S}^{d-1}} \langle Ax, x\rangle.$$

(3) En déduire que
$$\|A\|_{\mathrm{op}} = \max\left(|\lambda_1(A)|, |\lambda_d(A)|\right).$$

Exercice 5.3 (Méthode des puissances itérées). Soit $A \in S_d^{++}(\mathbb{R})$. On suppose que les valeurs propres de A sont toutes distinctes. On les note $0 < \lambda_1 < \cdots < \lambda_d$, et $(u_1, \dots u_d)$ est une base orthonormale de vecteurs propres associés.

(1) Soit $b \in \mathbb{R}^d$ tel que $\langle b, u_d\rangle > 0$. Montrer que
$$\lim_{n\to\infty} \frac{A^n b}{\|A^n b\|} = u_d, \quad \text{et que} \quad \lambda_d = \lim_{n\to\infty} \frac{\|A^{n+1}b\|}{\|A^n b\|}.$$

(2) En déduire un algorithme itératif pour calculer λ_d et u_d.

(3) Soit $b \in \mathbb{R}^d$ tel que $\langle b, u_{d-1}\rangle \neq 0$, et soit $\widetilde{b} := b - \langle b, u_d\rangle u_d$. Montrer que
$$\lim_{n\to\infty} \frac{A^n \widetilde{b}}{\|A^n \widetilde{b}\|} = \pm u_{d-1}, \quad \text{et que} \quad \lambda_{d-1} = \lim_{n\to\infty} \frac{\|A^{n+1}\widetilde{b}\|}{\|A^n \widetilde{b}\|}.$$

Autour du conditionnement. Dans tout le cours, on a utilisé le conditionnement d'une matrice en utilisant ses valeurs propres, et en travaillant sur $\mathbb{R}^d$ avec la métrique euclidienne. On peut sans peine généraliser cette notion à d'autres normes.

Définition 5.5 (Conditionnement d'une matrice pour une norme)**.** Soit N une norme sur $\mathbb{R}^d$ et $\|\cdot\|_N$ la norme matricielle induite :
$$\|M\|_N := \sup_{x\in\mathbb{R}^d} \frac{N(Mx)}{N(x)}.$$

Pour toute matrice inversible $M \in Gl_d(\mathbb{R})$, on appelle conditionnement de M relativement à la norme N la quantité $\text{cond}_N(M)$ définie par

$$\text{cond}_N(M) := \|M\|_N \|M^{-1}\|_N.$$

Exercice 5.4 (Propriétés de base du conditionnement). Dans tout cet exercice, N est une norme fixée sur $\mathbb{R}^d$.

(1) Montrer que pour tout $M \in Gl_d(\mathbb{R})$ on a $\text{cond}_N(M) \geq 1$.

(2) Montrer que si $M \in Gl_d(\mathbb{R})$ et si $\alpha \neq 0$, $\text{cond}_N(\alpha M) = \text{cond}_N(M)$.

(3) Montrer que si $M, N \in Gl_d(\mathbb{R})$ alors $\text{cond}_N(AB) \leq \text{cond}_N(A)\text{cond}_N(B)$.

Exercice 5.5 (Conditionnement et transposition). On travaille dans un premier temps avec la norme euclidienne, et on note le conditionnement relatif à cette norme cond_2. On définit le rayon spectral d'une matrice $A \in M_d(\mathbb{R})$ comme

$$\rho(A) = \sup_{\lambda \in \mathbb{C}, \lambda \text{ valeur propre de } A} |\lambda|.$$

(1) Montrer que pour toute matrice $A \in M_d(\mathbb{R})$ on a $\rho(A^t A) = \|A\|^2$ où $\|\cdot\|$ est la norme matricielle induite par la norme euclidienne sur $\mathbb{R}^d$.

(2) Montrer que pour tout $A, B \in M_d(\mathbb{R})$, si A est inversible, $\rho(AB) = \rho(BA)$.

(3) En déduire que $\text{cond}_2(A) = \text{cond}_2(A^t)$.

(4) On considère désormais la norme ℓ^1 sur $\mathbb{R}^d$ et on note cond_1 le conditionnement induit par cette norme et l'on veut savoir si $\text{cond}_1(A) = \text{cond}_1(A^t)$ si $A \in Gl_d(\mathbb{R})$. En dimension 3, montrer que cette propriété n'est pas valable en travaillant sur la matrice

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

Exercice 5.6 (Propriétés importantes pour le conditionnement relativement à la norme euclidienne). On travaille dans toutes ces questions avec la norme euclidienne sur l'espace $\mathbb{R}^d$.

(1) Invariance du conditionnement par transformation orthogonale Soient $A \in Gl_d(\mathbb{R})$ et $Q \in O_d(\mathbb{R})$. Montrer que $\text{cond}(QA) = \text{cond}(A)$.

(2) Matrices de conditionnement minimal Soit $M \in Gl_d(\mathbb{R})$ une matrice inversible telle que son conditionnement soit égal à 1. Montrer que $M = \alpha Q$ pour un certain $\alpha \in \mathbb{R}^*$ et $Q \in O_d(\mathbb{R})$.

(3) (***)Montrer que si $A, B \in S_d^{++}(\mathbb{R})$, on a

$$\text{cond}(A+B) \leq \max(\text{cond}(A), \text{cond}(B)).$$

Exercice 5.7 (Erreur relative et conditionnement). Dans cet exercice, on veut montrer que le conditionnement permet de contrôler l'erreur relative commise sur la solution d'un système linéaire $Mx = b$, quand on commet une erreur sur le terme de droite b ou sur la matrice M. On travaille avec la norme euclidienne.

(1) Avec une erreur sur le membre de droite : Soient $A \in Gl_d(\mathbb{R})$ et $b_0 \in \mathbb{R}^d \backslash \{0\}$. Soit x_0 la solution de

$$Ax_0 = b_0.$$

Soit $h \in \mathbb{R}^d$ et x_h la solution de

$$Ax_h = b_0 + h.$$

Démontrer que

$$\frac{\|x_0 - x_h\|}{\|x_0\|} \leq \mathrm{cond}(A) \frac{\|h\|}{\|b_0\|}.$$

(2) Avec une erreur sur la matrice : Soient $A \in Gl_d(\mathbb{R})$, $\varepsilon > 0$, $M \in M_d(\mathbb{R})$ et $b \in \mathbb{R}^d \backslash \{0\}$. Soit x_0 la solution de

$$Ax_0 = b.$$

En justifiant que, pour $\varepsilon > 0$ suffisamment petit $A + \varepsilon M \in Gl_d(\mathbb{R})$, et en notant, pour tout $\varepsilon > 0$ suffisamment petit x_ε la solution de

$$(A + \varepsilon M)\, x_\varepsilon = b$$

démontrer que

$$\frac{\|x_\varepsilon - x_0\|}{\|x_\varepsilon\|} \leq \mathrm{cond}(A) \frac{\varepsilon \|M\|}{\|A\|}.$$

Correction de la feuille de TD n°5

Correction 5.1. *(1) On a, pour tout $k \in \mathbb{N}$,*

$$x_{k+1} = x_k - \tau x_k - \tau b.$$

(2) Si la suite converge, elle converge vers l'unique minimiseur $x^* = \frac{b}{a}$ de la f (qui est coercive et strictement convexe). On a alors

$$x_{k+1} - x^* = (1 - \tau a)(x_k - x^*).$$

En particulier, on a convergence si, et seulement si,

$$|1 - \tau a| < 1$$

ce qui est équivalent à $\tau a < 2$.

(3) On voit que si $\tau = \frac{1}{a}$ la méthode converge en une unique itération.

Correction 5.2. *(1) Il s'agit d'un fermé borné de $\mathbb{R}^d$. Puisque l'on travaille en dimension finie, $\mathbb{S}^{d-1}$ est donc compacte.*

(2) Soit $\{u_i\}_{i=1,\dots,d}$ une base orthonormée de vecteurs propres associées à A et telles que, pour tout i, $Au_i = \lambda_i(A)u_i$. Pour tout $x \in \mathbb{R}^d$, on sait que

$$x = \sum_{i=1}^{d} \langle x, u_i \rangle u_i$$

de sorte que

$$\langle Ax, x \rangle = \sum_{i=1}^{d} \lambda_i(A) \langle x, u_i \rangle^2.$$

La conclusion s'ensuit immédiatement : on a, pour tout $x \in \mathbb{R}^d$,

$$\lambda_d(A)\|x\|^2 \geq \langle Ax, x \rangle \geq \lambda_1(A)\|x\|^2.$$

(3) On sait que

$$\|A\|_{\mathrm{op}} = \sup_{x \in \mathbb{S}^{d-1}} \sqrt{\langle Ax, Ax \rangle}.$$

La matrice A étant symétrique,

$$\sqrt{\langle Ax, Ax \rangle} = \sqrt{\langle A^2 x, x \rangle}.$$

On observe également que

$$\lambda_i(A^2) = \lambda_i(A)^2$$

de sorte que, par la question précédente, on obtient

$$\|A\|_{\mathrm{op}} \leq \max(|\lambda_1(A)|, |\lambda_d(A)|).$$

Par ailleurs, on a égalité en choisissant un bon vecteur propre.

Correction 5.3. *(1) On a*

$$A^n b = \lambda_d^n \left(\langle b, u_d \rangle u_d + \left(\frac{\lambda_{d-1}}{\lambda_d} \right)^n \langle b, u_{d-1} \rangle u_{d-1} + \cdots \right).$$

En particulier, comme $\lambda_{d-1} < \lambda_d$, on a $\lim_{n\to\infty} \lambda_d^{-n} A^n b = \langle b, u_d \rangle u_d \neq 0$. En divisant par la norme, on obtient la première égalité. La seconde s'obtient simplement en écrivant

$$\lambda_d = \|\lambda_d u_d\| = \|Au_d\| = \lim_{n\to\infty} \left\| A \frac{A^n b}{\|A^n b\|} \right\| = \lim_{n\to\infty} \frac{\|A^{n+1} b\|}{\|A^n b\|}.$$

(2) On peut calculer λ_d et u_d avec le code suivant.

```
def plusGrandeVAP(A, tol=1e-6, Niter=1000):
d = shape(A,0)
b = 0, rand(d)
u_n = b/norm(b)
lambda_n = norm( dot(A, u_n) )
for n in range(Niter):
if norm(dot(A, u_n) - lambda_n*u_n) < tol:
return lambda_n, u_n
u_n = dot(A, u_n)/norm(A, u_n)
lambda_n = norm(dot(A, u_n))
```

(3) On remarque que $\langle u_d, \widetilde{b}\rangle = \langle u_d, b\rangle - \langle u_d, b\rangle = 0$. Autrement dit, on peut reprendre la question 1) en commençant avec λ_{d-1}. Le reste suit.

Correction 5.4. *(1) On rappelle que toute norme induite est une norme matricielle (i.e. sous-multiplicative). Puisque pour toute matrice inversible M on a*

$$I_d = MM^{-1}$$

on en déduit

$$1 = \|Id\|_N \leq \|M\|_N \|M^{-1}\|_N = \mathrm{cond}_N(M).$$

(2) Il suffit d'observe que $(\alpha M)^{-1} = \frac{1}{\alpha} M^{-1}$.

(3) C'est une conséquence de ce que les normes induites sont des normes matricielles :

$$\mathrm{cond}_N(AB) = \|A^{-1}B^{-1}\|_N \|AB\|_N \leq \|A\|_N \|A^{-1}\|_N \|B\|_N \|B^{-1}\|_N.$$

Correction 5.5. *(1) Par définition de la norme induite on a*

$$\|A\| := \sup_{\|x\|=1} \langle Ax, Ax\rangle = \sup_{\|x\|=1} \langle A^t Ax, x\rangle = \rho(A^t A).$$

La dernière égalité vient de ce que la matrice $A^t A$ est symétrique, et que l'on peut donc appliquer le principe de Courant-Fisher.

(2) Un fait classique de L2 apprend que, quelles que soient les matrices $A, B \in M_d(\mathbb{R})$, et quel que soit $\lambda \in \mathbb{C}$ on a, si A est inversible,

$$\det(AB - \lambda I_d) = \det(BA - \lambda I_d).$$

En effet,

$$\begin{aligned}\det(AB - \lambda I_d) &= \det(ABAA^{-1} - \lambda AA^{-1}) \\ &= \det(A)\det(BA - \lambda I_d)\det(A^{-1}) \\ &= \det(BA - \lambda I_d).\end{aligned}$$

La conclusion est immédiate.

(3) En utilisant les question précédentes, on a

$$\rho(A^t A) = \|A\|^2 = \rho(AA^t)\|A^t\|^2$$

et de la même manière

$$\|A^{-1}\| = \|A^{-t}\|.$$

On en déduit que $\mathrm{cond}(A) = \mathrm{cond}(A^t)$.

(4) On commence par observer que la matrice A est inversible, d'inverse

$$A^{-1} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ -1 & 2 & 0 \\ 0 & -2 & 2 \end{pmatrix}.$$

Par ailleurs, la norme de A relativement à la norme ℓ^1 est la quantité

$$\|A\|_1 = \sup_{|x|+|y|+|z|=1} (4x + 2y + z) = 4.$$

De la même manière,

$$\|A^{-1}\|_1 = 1 = \|A^{-t}\|_1, \|A^t\|_1 = 3$$

et donc cette propriété n'est plus valable.

Correction 5.6. *(1) C'est une simple conséquence de l'invariance des normes induites par composition par une matrice orthogonale.*

(2) Soit $M \in Gl_d(\mathbb{R})$ telle que $\mathrm{cond}(M) = 1$. *Puisque l'on travaille avec la norme euclidienne, l'exercice précédent implique* $\mathrm{cond}(M^t) = \mathrm{cond}(M)$. *Puisque que* $\mathrm{cond}(AB) \leq \mathrm{cond}(A)\mathrm{cond}(B)$ *on en déduit que*

$$\mathrm{cond}(M^t M) \leq \mathrm{cond}(M)^2 = 1$$

et donc que la matrice symétrique définie positive $R = M^t M$ est de conditionnement 1. Puisque R est diagonalisable en base orthonormée réelle, toutes ses valeurs propres sont donc égales. Ainsi,

$$M^t M = \lambda I_d$$

pour un certain réel strictement positif λ. Si l'on pose $Q := \frac{1}{\sqrt{\lambda}} M$ on a ainsi

$$QQ^t = M$$

et donc $Q \in O_d$. Ainsi, $M = \sqrt{\lambda} Q$, ce qui conclue la preuve.

(3) On observe que si M est une matrice symétrique définie positive de valeurs propres $\lambda_1 \leq \cdots \leq \lambda_d$ on a

$$\mathrm{cond}(M) = \frac{\lambda_d}{\lambda_1} = \frac{\sup_{||x||=1} ||Mx||}{\inf_{||x||=1} ||Mx||}$$

de sorte que

$$\mathrm{cond}(A+B) = \frac{\sup_{||x||=1} ||(A+B)x||}{\inf_{||x||=1} ||(A+B)x||}.$$

Soient $\lambda_1 \leq \cdots \leq \lambda_d$ et $\mu_1 \leq \cdots \leq \mu_d$ les valeurs propres de A et B respectivement. On en déduit que

$$\mathrm{cond}(A+B) \leq \frac{\lambda_d + \mu_d}{\lambda_1 + \mu_1}.$$

Mais on sait en outre que

$$\frac{a+b}{c+d} \leq \max\left(\frac{a}{b}, \frac{c}{d}\right),$$

d'où la conclusion.

Correction 5.7. *(1) On pose $y_h := x_0 - x_h$. On a alors*

$$Ay_h = h \Leftrightarrow y_h = A^{-1}h.$$

Ainsi,

$$\|y_h\| \leq \|A^{-1}\| \cdot \|h\|.$$

Par ailleurs, puisque $b_0 = Ax_0$ on a

$$\|b_0\| \leq \|A\| \cdot \|x_0\| \Leftrightarrow \frac{1}{\|x_0\|} \leq \frac{\|A\|}{\|b_0\|}.$$

En multipliant ces deux inégalités terme à terme on obtient l'estimation voulue.

(2) On pose $y_\varepsilon := x_0 - x_\varepsilon$. On obtient l'équation

$$Ay_\varepsilon = \varepsilon M x_\varepsilon,$$

ce qui implique, les normes induites étant sous-multiplicatives, l'estimation

$$\|y_\varepsilon\| \leq \varepsilon \|A^{-1}\| \cdot \|M\| \cdot \|x_\varepsilon\|.$$

On en déduit immédiatement le résultat escompté.

6. Descente de gradient III : quelques règles de détermination des pas de descente

Comme annoncé à la fin du cours précédent, nous nous intéressons dans ce cours à des méthodes qui nous permettent de contourner les difficultés inhérentes aux descentes de gradient à pas constants *i.e.* le fait que la vitesse de convergence soit assez faible, et que l'on doive choisir à l'avance un bon pas. Pour cela, une bonne idée est de chercher, à chaque étape, le "bon" pas τ_k. Mais attention, ce choix ne peut pas être arbitraire : considérons la fonction $f : x \mapsto x^2$. On définit un pas adaptatif de taille

$$\tau_k := \frac{2 + 3 \cdot 2^{-(k+1)}}{2|x_k|}$$

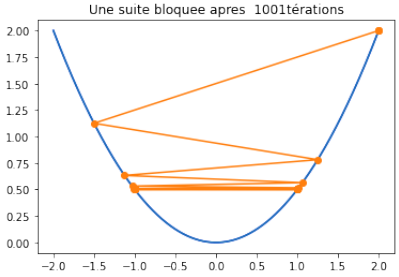
et l'on initialise la descente de gradient en $x_0 = 2$. On peut vérifier à la main que les itérations de la descente de gradient satisfont la relation de récurrence

$$x_{k+1} = x_k - \operatorname{sgn}(x_k) \left(2 + 3 \cdot 2^{-k-1}\right)$$

puis démontrer par récurrence (exercice !) que, pour tout $k \in \mathbb{N}$, on a

$$x_k = (-1)^k (1 + 2^{-k}).$$

On peut voir sur cette expression explicite que la suite reste *in fine* bloquée et oscille perpétuellement entre $+1$ et -1. Pourtant, on a bien choisi une direction de descente, et la valeur de la fonction à optimiser décroît bien à chaque itération, comme on peut le voir sur la figure suivante :



Une analyse possible de ce problème est que le pas est toujours bien trop grand. Voyons à présent comment remédier à cela, en présentant trois règles fondamentales : la recherche du **pas optimal exact**, la **règle de Wolfe** et la **règle d'Armijo**.

6.1. La descente de gradient à pas optimal.

6.1.1. *Présentation de la méthode.* Une première idée (d'ailleurs plutôt bonne) pour améliorer la convergence de la descente de gradient est, une fois une direction de descente d_k en une itération x_k choisie, de résoudre le problème d'optimisation unidimensionnel

$$\min_{\tau \geq 0} f(x_k + \tau d_k). \tag{6.1}$$

La résolution de ce problème, disons τ_k^*, est alors choisie comme taille de pas, et l'on pose

$$x_{k+1} = x_k + \tau_k^* d_k.$$

Dans ce cas, par définition même, on a bien

$$f(x_{k+1}) < f(x_k).$$

Mais (6.1) est un problème unidimensionnel, et l'on peut donc utiliser toutes les méthodes précédemment vues (méthode de la sécante, méthode de la section doré...) pour le calculer. C'est faisable, mais cela peut considérablement alourdir le coût des calculs et le temps que l'ordinateur met à générer une itération.

Une propriété importante de la descente de gradient à pas optimal est la suivante :

Lemme 6.1. *Soit $(x_k)_{k\in\mathbb{N}}$ une suite de points générées par une descente de gradient à pas optimal où, pour tout $k \in \mathbb{N}$, la direction de descente est $d_k \in \mathbb{R}^d$. Alors :*

$$\forall k \in \mathbb{N}\,, \langle \nabla f(x_{k+1}), d_k\rangle = 0.$$

Preuve du Lemme 6.1. On considère le problème d'optimisation (6.1). En un τ^* optimal, la condition d'optimalité d'ordre 1 donne

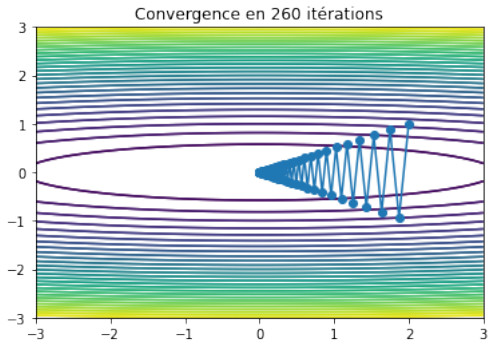
$$\langle \nabla f(x_k + \tau^* d_k), d_k\rangle = 0,$$

ce qui est exactement le propriété d'orthogonalité désirée. □

6.1.2. *Quelques exemples numériques.* On fixe comme choix de direction de descente $d_k = -\nabla f(x_k)$. Pour se convaincre que le choix d'un pas optimal peut effectivement améliorer la convergence de l'algorithme, reprenons le cas, vu au cours précédent, de la minimisation de

$$f : x \mapsto \frac{1}{2}\langle Ax, x\rangle \text{ avec } A = \begin{pmatrix} 1 & 0 \\ 0 & 30 \end{pmatrix},$$

donc, pour une matrice très mal conditionnée. On rappelle que la descente de gradient à pas constant donnait une convergence très mauvaise :



Si maintenant, au lieu de choisir un pas constant, on optimise la taille du pas on obtient

ce qui est une amélioration significative. Mais notons que ceci est fortement lié à la position du point initial, qui n'est pas situé "trop loin" d'un des axes de l'ellipse. Encore une fois, ceci est lié au mauvais conditionnement de la matrice. Pour forcer le trait, si l'on prend un point qui est situé presque exactement sur le plus long des axes de l'ellipse on obtient une convergence en deux itérations !

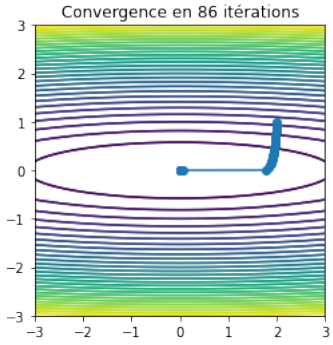


FIGURE 2. Amélioration de la convergence

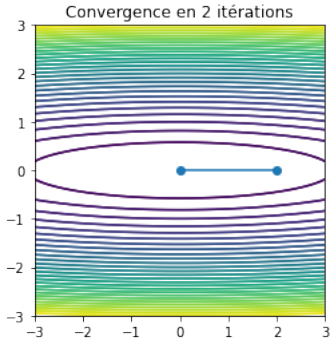


FIGURE 3. Une convergence très rapide si l'on initialise sur le grand axe de l'ellipse

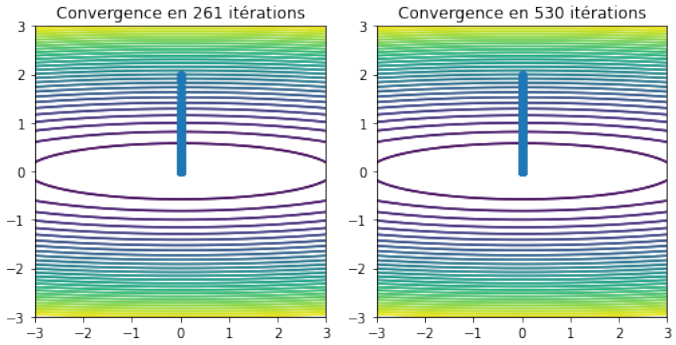


FIGURE 4. À gauche, une descente de gradient à pas constant, à droite une descente de gradient à pas optimal

À l'inverse, si l'on prend une initialisation situé sur le plus petit axe de l'ellipse, on fait presque aussi mauvais qu'avec une descente de gradient à pas constant :

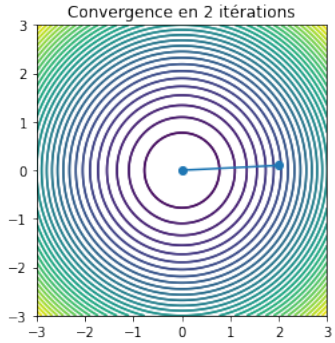


FIGURE 5. Convergence très rapide pour des matrices A colinéaires à l'identité

Observons finalement que, si la matrice A est colinéaire à l'identité alors, quelle que soit l'initialisation choisie, la méthode converge en une ou deux itérations :

6.1.3. *Quelle convergence pour la descente de gradient à pas optimal ?* Il reste à voir si cet algorithme de descente à pas optimal converge effectivement, et s'il fait mieux, ou non, que la descente de gradient à pas constant.

Théorème 6.1 (Convergence de la descente de gradient à pas optimal)**.** *Soit* $A \in \mathcal{S}_d^{++}(\mathbb{R}), b \in \mathbb{R}^d$ *et*

$$f : x \mapsto \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

*La descente de gradient (*i.e. *à chaque itération* k *la direction de descente est* $-\nabla f(x_k)$*) à pas optimal converge linéairement à taux*

$$\left(\frac{\mathrm{cond}(A) - 1}{\mathrm{cond}(A) + 1}\right).$$

On a l'estimation précisée

$$\langle A(x_k - x^*), x_k - x^*\rangle \leq \langle A(x_0 - x^*), x_0 - x^*\rangle \left(\frac{\mathrm{cond}(A) - 1}{\mathrm{cond}(A) + 1}\right)^{2k}.$$

On voit donc que le taux minimal de convergence pour la descente de gradient à pas optimal et le taux optimal pour la descente de gradient à pas constant. En particulier, *la descente de gradient à pas optimal peut être aussi mauvaise que la descente de gradient à pas constant.* Donnons néanmoins une preuve de ce théorème de convergence. Au delà de l'aspect "complétude" du cours, le point positif de cette preuve est qu'elle utilise l'inégalité de Kantorovich, fondamentale dans l'étude des méthodes de descente de gradient.

Proposition 6.1 (Inégalité de Kantorovich)**.** *Soit* $A \in S_d^{++}(\mathbb{R})$ *et* $0 < \lambda_1(A) \leq \cdots \leq \lambda_d(A)$ *ses valeurs propres ordonnées. Alors, pour tout* $x \in \mathbb{R}^d$ *de norme 1, on a*

$$1 \leq \langle Ax, x\rangle \cdot \langle A^{-1}x, x\rangle \leq \frac{1}{4} \cdot \frac{(\lambda_1(A) + \lambda_d(A))^2}{\lambda_1(A)\lambda_d(A)}.$$

La preuve de cette inégalité est un joli exercice de convexité. Nous donnons ici la preuve due à Newman (qui traite d'une généralisation de cette inégalité. Nous renvoyons à la fiche de TD).

Preuve de la Proposition 6.1. La preuve repose sur deux ingrédients : l'inégalité arithmético-géométrique, et l'inégalité de Jensen. Que ce soit pour la borne inférieure ou la borne supérieure, on travaille sous l'hypothèse que A est une matrice diagonale. En toute généralité, il suffit d'utiliser le théorème spectral et de faire un changement de base orthogonale. En notant simplement, pour des raisons de simplicité, λ_k au lieu de $\lambda_k(A)$ pour la k-ème valeur propre de A, on a

$$\langle Ax, x\rangle \cdot \langle A^{-1}x, x\rangle = \sum_{k=1}^{d} x_k^2 \lambda_k \sum_{k=1}^{d} \frac{x_k^2}{\lambda_k}.$$

Commençons par la borne inférieure. Par l'inégalité de Cauchy-Schwarz, on a

$$1 = \left(\sum_{k=1}^{d} x_k^2\right)^2 = \left(\sum_{k=1}^{d} |\sqrt{\lambda_k} x_k| \cdot \left|\frac{x_k}{\sqrt{\lambda_k}}\right|\right)^2 \leq \sum_{k=1}^{d} x_k^2 \lambda_k \sum_{k=1}^{d} \frac{x_k^2}{\lambda_k}.$$

Passons à la borne supérieure. Par l'inégalité arithmético-géométrique on a, pour tout $\delta > 0$,

$$\begin{aligned} 2\left(\sum_{k=1}^{d} x_k^2 \lambda_k \sum_{k=1}^{d} \frac{x_k^2}{\lambda_k}\right)^{\frac{1}{2}} &\leq \sum_{k=1}^{d} \delta x_k^2 \lambda_k + \sum_{k=1}^{d} \frac{x_k^2}{\delta \lambda_k} \\ &\leq \sum_{k=1}^{d} x_k^2 \phi(\delta\lambda_k), \end{aligned}$$

avec $\phi : x \mapsto x + \frac{1}{x}$. Observons à présent que la fonction ϕ est convexe, donc qu'en particulier

$$\sum_{k=1}^{d} x_k^2 \phi(\delta\lambda_k) \leq \max(\phi(\delta\lambda_1), \phi(\delta\lambda_d)).$$

Choisissons δ tel que

$$\phi(\delta\lambda_1) = \phi(\delta\lambda_d).$$

Par des calculs explicites, on obtient

$$\delta = \frac{1}{\sqrt{\lambda_1 \lambda_d}} \text{ et ainsi } \phi(\delta\lambda_1) = \sqrt{\frac{\lambda_1}{\lambda_d}} + \sqrt{\frac{\lambda_d}{\lambda_1}}.$$

En particulier, nous avons démontré

$$2\left(\sum_{k=1}^{d} x_k^2 \lambda_k \sum_{k=1}^{d} \frac{x_k^2}{\lambda_k}\right)^{\frac{1}{2}} \leq \sqrt{\frac{\lambda_1}{\lambda_d}} + \sqrt{\frac{\lambda_d}{\lambda_1}}.$$

En élevant chaque membre de cette inégalite au carré, on obtient la conclusion désirée. □

Enfin, afin de donner une preuve efficace, calculons le pas optimal pour la fonction

$$f : x \mapsto \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

Proposition 6.2. *Si $A \in S_d^{++}(\mathbb{R})$, le pas optimal pour la descente de gradient, partant d'un point x, est*

$$\tau^* = \frac{\|Ax - b\|^2}{\langle A^2x - Ab, Ax - b\rangle}.$$

En particulier, l'algorithme de descente de gradient à pas optimal admet la description explicite suivante :

(1) Initialisation en $x_0 \in \mathbb{R}^d$.

(2) Pour tout $k \in \mathbb{N}$,

$$x_{k+1} = x_k - \frac{\|Ax_k - b\|^2}{\langle A^2x_k - Ab, Ax_k - b\rangle}(Ax_k - b)$$

Preuve de la proposition 6.2. Le gradient de f en x est donné par

$$\nabla f(x) = Ax - b := p_x.$$

Considérons la fonction d'une variable réelle

$$g : \mathbb{R}_+ \ni \tau \mapsto f(x - \tau p_x).$$

En un point de minimum τ^* on a

$$g'(\tau^*) = 0.$$

Puisque

$$g'(\tau) = \langle \nabla f(x - \tau p_x), -p_x\rangle$$

on en déduit qu'en un minimum τ^* (nécessairement unique par stricte convexité et coercivité de la fonction f) on a

$$\langle A(x - \tau p_x) - b, p_x\rangle = 0,$$

ce qui est équivalent à

$$\tau\langle Ap_x, p_x\rangle = \langle Ax - b, p_x\rangle = \langle Ax - b, Ax - b\rangle = \|Ax - b\|^2,$$

ou encore

$$\tau^* = \frac{\|Ax - b\|^2}{\langle A^2x - Ab, Ax - b\rangle}.$$

□

Preuve du Théorème 6.1. Définissons, pour tout $k \in \mathbb{N}$,

$$y_k := A(x_k - x^*).$$

Dans la mesure où x^* est l'unique solution de $Ax = b$ on peut réécrire grâce à la Proposition 6.2 la suite des itérations de la descente de gradient comme

$$x_{k+1} = x_k - \frac{\|y_k\|^2}{\langle Ay_k, y_k\rangle} y_k.$$

Ainsi,

$$\begin{cases} x_{k+1} - x^* = (x_k - x^*) - \frac{\|y_k\|^2}{\langle Ay_k, y_k\rangle} y_k = \left(A^{-1} - \frac{\|y_k\|^2}{\langle Ay_k, y_k\rangle}\mathrm{I}_d\right) y_k\,, \\ y_{k+1} = \left(\mathrm{I}_d - \frac{\|y_k\|^2}{\langle Ay_k, y_k\rangle} A\right) y_k \end{cases}$$

de sorte que

$$
\begin{aligned}
\langle y_{k+1}, x_{k+1} - x^* \rangle &= \left\langle \left(\mathrm{I}_d - \frac{\|y_k\|^2}{\langle Ay_k, y_k \rangle} A \right) y_k, \left(A^{-1} - \frac{\|y_k\|^2}{\langle Ay_k, y_k \rangle} \mathrm{I}_d \right) y_k \right\rangle \\
&= \left\langle y_k, \left(\mathrm{I}_d - \frac{\|y_k\|^2}{\langle Ay_k, y_k \rangle} A \right) \left(A^{-1} - \frac{\|y_k\|^2}{\langle Ay_k, y_k \rangle} \mathrm{I}_d \right) y_k \right\rangle \\
&\quad \text{car } A \text{ est symétrique} \\
&= \left\langle y_k, A \left(A^{-1} - \frac{\|y_k\|^2}{\langle Ay_k, y_k \rangle} \mathrm{I}_d \right)^2 y_k \right\rangle \\
&= \left\langle y_k, \left(A^{-1} - 2\frac{\|y_k\|^2}{\langle Ay_k, y_k \rangle} \mathrm{I}_d + \frac{\|y_k\|^4}{\langle Ay_k, y_k \rangle^2} A \right) y_k \right\rangle \\
&= \langle y_k, A^{-1} y_k \rangle \\
&\quad - 2\frac{\|y_k\|^4}{\langle Ay_k, y_k \rangle} \\
&\quad + \frac{\|y_k\|^4}{\langle Ay_k, y_k \rangle} \\
&= \langle A(x_k - x^*), x_k - x^* \rangle \\
&\quad - \frac{\|y_k\|^4 \langle Ay_k, A^{-1} y_k \rangle}{\langle Ay_k, y_k \rangle \langle A^{-1} y_k, y_k \rangle} \\
&= \langle A(x_k - x^*), x_k - x^* \rangle \left(1 - \frac{\|y_k\|^4}{\langle Ay_k, y_k \rangle \langle A^{-1} y_k, y_k \rangle} \right).
\end{aligned}
$$

En définissant la fonction

$$G(x) := \langle A(x - x^*), x - x^* \rangle$$

nous avons donc établi

$$G(x_{k+1}) = G(x_k) \left(1 - \frac{\|y_k\|^4}{\langle Ay_k, y_k \rangle \langle A^{-1} y_k, y_k \rangle} \right).$$

Par l'inégalité de Kantorovich (Proposition 6.1) on sait que

$$\left(1 - \frac{\|y_k\|^4}{\langle Ay_k, y_k \rangle \langle A^{-1} y_k, y_k \rangle} \right) \leq 1 - 4\frac{\lambda_1(A)/\lambda_d(A)}{\left(\lambda_1(A)/\lambda_d(A) + 1\right)^2} = 1 - 4\frac{\mathrm{cond}(A)}{(\mathrm{cond}(A) + 1)^2},$$

et donc

$$G(x_{k+1}) \leq G(x_k) \left(1 - 4\frac{\mathrm{cond}(A)}{(\mathrm{cond}(A) + 1)^2} \right) = G(x_k) \left(\frac{\mathrm{cond}(A) - 1}{\mathrm{cond}(A) + 1} \right)^2.$$

On en tire la conclusion par une simple récurrence. □

6.2. **Règle d'Armijo, règle de Wolfe.** Comme nous l'avons vu, les algorithmes d'optimisation en dimension 1 peuvent être gourmands en temps. Nous allons donc voir dans la suite de ce cours deux règles pour déterminer des pas qui *sont suffisamment satisfaisants du point de vue de l'optimisation* tout en étant plus faciles à calculer.

6.2.1. *Discussion heuristique.* L'idée des deux règles que nous présentons maintenant est qu'à l'étape k, on choisisse un pas qui ne soit ni trop grand, ni trop petit. Nous l'avons vu, c'est cela qui peut poser problème quand on travaille sur $f : x \mapsto x^2$.

Pour vérifier que l'on ne choisit pas des pas trop grands, qui nous enverraient trop loin de la solution x^* (auquel cas nos approximations de Taylor au premier ordre n'auraient plus aucun sens), on utilise la règle d'Armijo suivante : on sait que, si l'on definit

$$g_k : \tau \mapsto f(x_k + \tau d_k)$$

alors on veut choisir τ_k qui garantisse que $g_k(\tau_k)$ soit "suffisamment" inférieur à $g_k(0) = f(x_k)$. Il nous faut quantifier cela. Fixons donc un paramètre $\gamma > 0$. L'idéal serait d'obtenir un pas de descente $\tau > 0$ tel que

$$f(x_k + \tau d_k) \leq f(x_k) - \gamma\tau.$$

Par ailleurs, il est naturel de quantifier γ en fonction de la pente de la fonction f au point x_k, dans la direction d_k. On choisira donc plutôt

$$\gamma = -\gamma_1 \langle \nabla f(x_k), d_k \rangle$$

où $\gamma_1 \in]0;1[$ sera un paramètre réel à choisir. Ce type de condition va s'appeler **condition d'Armijo**.

À l'inverse, on ne veut pas que les pas soient trop petits. En effet, quelle que soit la fonction f, quelle que soit l'initialisation x_0, si l'on fait des pas microscopiques, on n'a aucune chance de converger vers un minimiseur (il suffit de prendre une suite de taille de pas τ_k telle que $\sum_{k=0}^{\infty} \tau_k \|\nabla f(x_k)\| \leq \frac{\|x_0 - x^*\|}{2}$). Comment quantifier ce critère-là ? L'idée de la **condition de Wolfe** est de travailler sur le gradient de f. En effet, d_k étant une direction de descente, on a

$$\langle \nabla f(x_k), d_k \rangle < 0.$$

Si l'on choisissait pour τ le τ^* du pas optimal exact, on aurait

$$\langle \nabla f(x_k + \tau^* d_k), d_k \rangle = 0. \tag{6.2}$$

La condition de Wolfe, c'est relaxer la condition d'égalité dans (6.2) pour la remplacer par une condition de type "le produit scalaire $\langle \nabla f(x_k + \tau^* d_k), d_k \rangle$ s'est suffisamment rapproché de 0". En choisissant un paramètre $\gamma_2 \in]0;1[$ il faudra donc garantir une condition du type

$$\langle \nabla f(x_k + \tau^* d_k), d_k \rangle \geq \gamma_2 \langle \nabla f(x_k), d_k \rangle.$$

6.2.2. *Formalisation des règles d'Armijo et de Wolfe.* Ces considérations nous amènent à poser la définition suivante :

Définition 6.1. Soit $f \in \mathscr{C}^1(\mathbb{R}^d; \mathbb{R})$, soit $x_k \in \mathbb{R}^d$, d_k une direction de descente en x_k (on suppose en particulier que $\nabla f(x_k) \neq 0$). Soit $\tau > 0$ une longueur de pas.

(1) Si $\gamma_1 \in (0;1)$, on dit que le pas τ satisfait **la condition d'Armijo** en x_k pour γ_1 si

$$f(x_k + \tau d_k) \leq f(x_k) + \gamma_1 \tau \langle \nabla f(x_k), d_k \rangle.$$

(2) Si $0 < \gamma_1 < \gamma_2 < c_2$, on dit que le pas τ vérifie la **condition de Wolfe** en x_k pour (γ_1, γ_2) si τ vérifie la condition d'Armijo en x_k pour γ_1 et si, en outre, on a

$$\langle \nabla f(x_k + \tau d_k), d_k \rangle \geq \gamma_2 \langle \nabla f(x_k), d_k \rangle.$$

Il nous reste à vérifier deux choses : la première, c'est que, si l'on se fixe un couple (γ_1, γ_2) alors, à chaque itération, si l'on se fixe une direction de descente d_k, on peut trouver un pas τ_k qui vérifie la condition de Wolfe en x_k et, en outre, que l'algorithme ainsi modifié a de meilleures propriétés de convergence.

Commençons par le prmeier point, à savoir que, si l'on se fixe un couple (γ_1, γ_2), alors on peut trouver à chaque itération un pas admissible :

Proposition 6.3. *On fixe $0 < \gamma_1 < \gamma_2 < 1$ et $f \in \mathscr{C}^2(\mathbb{R}^d; \mathbb{R})$. Soit $x_k \in \mathbb{R}^d$ et d_k une direction de descente de f en x_k. On suppose que f est bornée inférieurement. Alors il existe $0 < \tau_1 < \tau_2$ tels que, pour tout $\tau \in]\tau_1; \tau_2[$, le pas τ vérifie la condition de Wolfe en x_k pour (γ_1, γ_2). On prendra garde au fait que τ_1 et τ_2 dépendent de l'indice k.*

Le point important dans cette proposition, c'est que les deux paramètres γ_1 , γ_2 sont uniformes, c'est-à-dire qu'ils ne dépendent pas de l'itération.

Preuve de la proposition. Commençons par construire τ_1 , τ_2 tels que tout pas $\tau \in]\tau_1; \tau_2[$ est admissible pour la règle d'Armijo. À cet effet, définissons la fonction

$$\Phi : t \mapsto f(x_k + td_k).$$

Puisque $\Phi'(0) = \langle \nabla f(x_k), d_k \rangle < 0$ et que f est de classe $\mathscr{C}^1$ on en déduit qu'il existe $\varepsilon_1 > 0$ tel que,

$$\forall t \in [0; \varepsilon_1[\,, \Phi'(t) \leq \gamma_1 \Phi'(0).$$

En particulier, pour tout $t \in [0; \varepsilon_1[$, on a

$$\Phi(t) \leq \Phi(0) + t\gamma_1 \Phi'(0).$$

Par la définition de Φ, ceci implique que

$$\forall t \in [0; \varepsilon_1[\,, f(x_k + td_k) \leq f(x_k) + t\gamma_1 \langle \nabla f(x_k), d_k \rangle,$$

ou encore que tout pas $\tau \in]0; \varepsilon_1[$ vérifie la règle d'Armijo en x_k.

Remarque 6.1. De manière cohérente avec l'intuition, on ne voit pas dans cette construction apparaître de borne inférieure sur la taille du pas que l'on construit. Cette borne inférieure, qui garantit que l'on ne fait pas de pas trop petits, vient de la règle de Wolfe.

Maintenant, pour construire un pas qui satisfasse également la règle de Wolfe, saturons l'inégalité précédente ; en d'autres termes, définissons

$$\varepsilon_1^* := \sup \{ \ \varepsilon > 0 \text{ tel que } \forall t < \varepsilon \,, f(x_k + td_k) \leq f(x_k) + t\gamma_1 \langle \nabla f(x_k), d_k \rangle \} \,.$$

Puisque f est bornée inférieurement, ε_1^* est bien défini et on a, en ε_1^*,

$$f(x_k + \varepsilon_1^* d_k) = f(x_k) + \varepsilon_1^* \gamma_1 \langle \nabla f(x_k), d_k \rangle.$$

En reprenant la fonction Φ introduite précédemment, cette identité s'écrit

$$\frac{\Phi(\varepsilon_1^*) - \Phi(0)}{\varepsilon_1^*} = \gamma_1 \Phi'(0) > \gamma_2 \Phi'(0).$$

Or, par le théorème des accroissements finis, il existe $\overline{\tau} \in (0; \varepsilon_1)$ tel que

$$\Phi'(\overline{\tau}) = \frac{\Phi(\varepsilon_1^*) - \Phi(0)}{\varepsilon_1^*} = \gamma_1 \Phi'(0) > \gamma_2 \Phi'(0),$$

et donc, il existe un voisinage $]\tau_1; \tau_2[$ de $\overline{\tau}$ tel que, pour tout $\tau \in]\tau_1; \tau_2[$, on a

$$\gamma_2 \Phi'(0) = \gamma_2 \langle \nabla f(x_k), d_k \rangle \leq \langle \nabla f(x_k + \tau d_k), d_k \rangle,$$

ce qui est exactement la condition de Wolfe. Par construction, on $\overline{\tau} \in]0; \varepsilon_1^*[$ et l'on peut donc choisir $\tau_2 < \varepsilon_1^*$. En particulier, τ_1, τ_2 satisfont aux conclusions du théorème. □

6.2.3. *Peut-on estimer la taille des pas ?* Montrons que la règle de Wolfe garantit effectivement que les pas que l'on construit ne sont pas trop petits. On suppose le paramètre γ_2 fixés dans la suite. On se donne une fonction f, et l'on suppose que ∇f est L-lipschitzien. On suppose que l'on travaille en une itération x_k, que l'on a choisi une direction de descente d_k, non nécessairement égale à $-\nabla f(x_k)$, et que l'on a un pas τ qui vérifie le critère Wolfe.

Puisque d_k est une direction de descente et que $\gamma_2 < 1$ on a

$$\begin{aligned}(\gamma_2 - 1)\langle \nabla f(x_k), d_k\rangle &= \gamma_2 \langle \nabla f(x_k), d_k\rangle \\ &\quad - \langle \nabla f(x_k), d_k\rangle \\ &\leq \langle \nabla f(x_{k+1}), d_k\rangle - \langle \nabla f(x_k), d_k\rangle \\ &\quad \text{car le pas vérifie la règle de Wolfe} \\ &\leq L\|x_{k+1} - x_k\| \cdot \|d_k\| \text{ par Cauchy-Schwarz} \\ &\leq \tau L \|d_k\|^2.\end{aligned}$$

Ainsi, on a établi

$$\tau \geq \frac{1-\gamma_2}{L} \left| \frac{\langle \nabla f(x_k), d_k\rangle}{\|d_k\|^2} \right| \tag{6.3}$$

En particulier, si l'on a choisi la direction de plus grande descente ou, en d'autres termes, si l'on a choisi $d_k = -\nabla f(x_k)$ (c'est-à-dire que l'on fait une descente de gradient standard) on obtient

$$\tau \geq \frac{1-\gamma_2}{L}.$$

6.2.4. *Convergence de l'algorithme pour des pas satisfaisant les critères d'Armijo et de Wolfe.* Nous allons à présent voir que, si le pas τ satisfait à la fois les critères d'Armijo et de Wolfe, alors la méthode converge en un certain sens.

Proposition 6.4 (Zoutendjik)**.** *Soit $f \in \mathscr{C}^1(\mathbb{R}^d; \mathbb{R})$ une fonction minorée et telle que ∇f soit L-lipschitzienne. Soient $0 < \gamma_1 < \gamma_2 < 1$ deux paramètres fixés. Soit $\{x_k\}_{k\in\mathbb{N}}$ la suite définie par $x_0 \in \mathbb{R}^d$ et, pour tout $k \in \mathbb{N}$, $x_{k+1} = x_k + \tau_k d_k$, où d_k est une direction de descente de f en x_k et où τ_k satisfait les critères d'Armijo et de Wolfe pour les paramètres (γ_1, γ_2) en x_k. Alors*

$$\sum_{k=0}^{\infty} \frac{|\langle \nabla f(x_k), d_k\rangle|^2}{\|d_k\|^2} < \infty.$$

En particulier, si $d_k = -\nabla f(x_k)$, on a

$$\nabla f(x_k) \underset{k\to\infty}{\to} 0.$$

Preuve de la proposition. Repartons de l'estimation (6.3) : pour tout $k \in \mathbb{N}$, on a

$$\tau_k \geq \frac{1-\gamma_2}{L} \left| \frac{\langle \nabla f(x_k), d_k\rangle}{\|d_k\|^2} \right|.$$

Rappelons que ceci est valable car τ_k vérifie la règle de Wolfe. Puisque τ_k satisfait également la règle d'Armijo, on a également

$$f(x_k) - f(x_{k+1}) \geq \tau_k \gamma_1 \left|\langle \nabla f(x_k), d_k \rangle\right|.$$

Ainsi,

$$f(x_k) - f(x_{k+1}) \geq \frac{\gamma_1(1-\gamma_2)}{L} \cdot \left| \frac{\langle \nabla f(x_k), d_k \rangle}{\|d_k\|} \right|^2.$$

En somment ces estimations, on obtient l'estimation

$$\frac{\gamma_1(1-\gamma_2)}{L} \sum_{k=0}^{\infty} \frac{|\langle \nabla f(x_k), d_k \rangle|^2}{\|d_k\|^2} \leq f(x_0) - \inf f.$$

Ceci conclut la preuve. □

Mais attention, nous avons bien dit que cette convergence n'avait lieu qu'en un certain sens. En effet, dans le cas $d_k = -\nabla f(x_k)$, on n'obtient "que" l'information

$$\nabla f(x_k) \underset{k\to\infty}{\to} 0.$$

Or, il se peut que ceci se produise, sans toutefois que la suite $\{x_k\}_{k\in\mathbb{N}}$ elle-même ne converge. Pensons par exemple à la fonction e^{-x^2}. Néanmoins, si l'on suppose en outre que la fonction $\|\nabla f\|^2$ est coercive, alors cette information permet de conclure que la suite $\{x_k\}_{k\in\mathbb{N}}$ est bornée, et donc de commencer à utiliser des arguments de compacité.

6.2.5. *Détermination numérique de pas satisfaisants les critères d'Armijo et de Wolfe.* Exposons brièvement comment ces pas peuvent être déterminés numériquement. On se fixe dans toute la suite une fonction f, un point x, une direction de descente d en x, deux paramètres $0 < \gamma_1 < \gamma_2 < 1$. On construit deux suites $\{\tau_{1,k}, \tau_{2,k}\}_{k\in\mathbb{N}}$ telles que :

(1) Pour tout $k \in \mathbb{N}$, $\tau_{1,k} < \tau_{2,k}$.

(2) Pour tout $k \in \mathbb{N}$, $\tau_{1,k}$ vérifie la règle d'Armijo, mais pas $\tau_{2,k}$.

(3) Les deux suites convergent, et ont la même limite τ_∞.

Notons d'abord le fait suivant : on peut bien initialiser l'algorithme **si f est bornée inférieurement**. Pour $\tau_{1,0}$, il suffit de choisir un pas extrêmement petit (on peut le tester à la main sur ordinateur). Pour $\tau_{2,0}$ en revanche il faut s'assurer que l'on peut choisir de sorte à ce qu'il ne satisfasse pas la règle d'Armijo. Mais observons que, f étant bornée inférieurement, la fonction

$$t \mapsto f(x+td) - t\gamma_1 \langle \nabla f(x), d \rangle - f(x)$$

tend vers $-\infty$ quand $t \to \infty$. Maintenant, supposons cette initialisation donnée. On définit l'itération suivante en posant $x_1 := \frac{\tau_{1,0}+\tau_{2,0}}{2}$ et

$$(\tau_{1,1}, \tau_{2,1}) := \begin{cases} (x_1, \tau_{2,0}) \text{ si } x_1 \text{ vérifie la règle d'Armijo,} \\ (\tau_{1,0}, x_1) \text{ sinon.} \end{cases}$$

On itère cette construction, et on l'arrête dès que $\tau_{1,k}$ vérifie également le critère de Wolfe, ce qui nous mène à énoncer l'algorithme sous la forme suivante :

Il faut nous assurer que cet algorithme converge en un nombre fini d'étapes. C'est le cas, comme on peut le voir par l'argument (élémentaire) suivant : il est clair que, par les mêmes arguments que ceux utilisés pour la méthode de dichotomie, les deux

Algorithm 3 Recherche d'un pas vérifiant à la fois Armijo et Wolfe

```
def pasWolfe(F, dF, x, h, tau0, Tau0, gamma1=0.2, gamma2=0.7,
Niter=1000):
 Fx, dFx = F(x), dF(x)
Initialisation
taun, Taun, xn = tau0, Tau0, (tau0 + Tau0)/2
taun satisfait Wolfe
 if dot(dF(x + taun*h), h) >= gamma2*dot(dFx,h):
return   taun
taun satisfait Armijo
if F(x + xn*h) <= Fx + gamma1*xn*dot(dFx,h):
 taun, Taun = taun, xn
else:
 taun,Taun=xn,Taun
print("Erreur, l'algorithme n'a pas convergé
après",IterMax,"itérations")
```

suites $\{\tau_{1,k}\}_{k\in\mathbb{N}}$ et $\{\tau_{2,k}\}_{k\in\mathbb{N}}$ sont adjacentes. En particulier, elles convergent vers une même limite τ^*. Mais on a, d'une part,

$$f(x+\tau_{2,k}d)-f(x)>\gamma_1\tau_{2,k}\langle\nabla f(x),d\rangle$$

et d'autre part

$$f(x+\tau_{1,k}d)-f(x)<\gamma_1\tau_{1,k}.$$

Formant le quotient différentiel $(f(x+\tau_{2,k}d)-f(x+\tau_{1,k}d))/(\tau_{2,k}-\tau_{1,k})$ il apparaît que

$$\langle\nabla f(x+\tau^*d),d\rangle\geq\gamma_1\langle\nabla f(x),d\rangle>\gamma_2\langle\nabla f(x),d\rangle.$$

En d'autres termes, τ^* vérifie la condition de Wolfe. Ainsi, pour k suffisamment grand, $\tau_{1,k}$ satisfait également la condition de Wolfe.

Feuille de TP n°4 : Points selles et algorithme de l'élastique

On présente un algorithme d'optimisation permettant de déterminer les points selles d'une fonction de plusieurs variables. La recherche de ces points est très importante en chimie : grossièrement, un système chimique cherche toujours à minimiser son énergie (états stables = minima). La chimie étudie le passage d'un minimum à un autre (changement d'états = points cols). On n'oubliera pas d'appeler les librairies `numpy` et `matplotlib.pyplot`.

Dans toute cette feuille de TP on considère la fonction F définie par

$$F : \mathbb{R}^2 \ni (x_1, x_2) \mapsto x_1^4 - 2x_1^2 + 2x_2^2 + x_1 + 2x_1^2 x_2.$$

Étude de la fonction F.

Exercice 6.1. Définir la fonction F (qui prend en argument un `array x` de taille 2). Représenter 100 lignes de niveau de F sur $[-2, 2] \times [-2, 1]$. Identifier graphiquement les minima locaux et les éventuels points selles.

Exercice 6.2. Ecrire la fonction `dF(x)` qui prend un `array x` de taille 2, et qui renvoie le gradient de F sous forme d'un `array` de taille 2.

Nous allons maintenant déterminer les minima locaux de F en utilisant une descente de gradient à pas fixe. Nous rappelons le code établi dans le troisième TP (vous pouvez également utiliser le vôtre) :

```
1   def descenteGradient(dF, x0, tau=0.05, tol=1e-6, Niter
        =1000):
2     xn,L = x0,[]
3     for n in range(Niter):
4         dFxn = dF(xn)
5         if linalg.norm(dFxn) < tol:
6             return xn, L
7         L.append(xn)
8         xn = xn - tau*dF(xn)
9     print("Probleme, la descente de gradient n'a pas convergé
          e après", Niter,'itérations')
10    return xn, L
```

Exercice 6.3. Que fait le code ci-dessous ?

```
1  xA,LA = descenteGradient(dF, array([-1, 0]))
2  xB,LB = descenteGradient(dF, array([1,-1]))
3
4  print('Point xA found in %d iterations'%len(LA))
5  print('Point xB found in %d iterations'%len(LB))
6
7  xx1 = linspace(-2,2,100)
8  xx2 = linspace(-2,1,100)
9  Z = array([[F(array([x1,x2])) for x1 in xx1 ] for x2 in xx2])
10
11 contour(xx1,xx2,Z,100)
```

```
12 plot([x[0] for x in LA], [x[1] for x in LA], '.-r')
13 plot([x[0] for x in LB], [x[1] for x in LB], '.-b')
```

À la recherche des points selles. Dans la section précédente nous avons pu déterminer numériquement les minima x_A et x_B de la fonction F. On va essayer de déterminer le point selle de la fonction F *via* la *méthode de l'élastique*. Heuristiquement, l'idée est la suivante : on rejoint les deux points x_A et x_B par un élastique que l'on tire. La trajectoire dessinée par l'élastique ainsi tirée est la trajectoire qui monte "le moins haut" possible. On peut démontrer que ce point culminant est un point selle de la fonction f.

Pour rendre cela plus rigoureux on a besoin de préciser comment notre élastique est modélisé. Rappelons d'abord que pour un élastique linéique de constante de raideur k, de longueur au repos ℓ, l'énergie de l'élastique quand il est tendu entre un point x_0 et un point x_1 est donnée par l'expression

$$k\left(\|x_0 - x_1\| - \ell\right)^2.$$

Dans notre contexte, un élastique est une approximation d'un élastique courbe par une succession de "petits" élastiques linéiques.

On considère un N-uplet `[x1,...,xN]` de points de $\mathbb{R}^2$. On pose

$$\texttt{x0} = x_A\,, \texttt{x(N+1)} = x_B$$

et l'on définit

$$\texttt{X:=} \begin{pmatrix} \texttt{x0} \\ \texttt{x1} \\ \vdots \\ \texttt{x(N+1)} \end{pmatrix}.$$

`X` modélise l'élastique.

L'énergie totale de l'élastique constitué de petits élastiques linéiques de constante de raideur k et de longueur au repos ℓ, est donnée par l'expression

$$E_{k,\ell}(\texttt{X}) = \sum_{i=0}^{N} k(N+1)\left(\|\texttt{x(i+1)-xi}\| - \ell\right)^2 + F(\texttt{xi}).$$

Attention à la normalisation !

Exercice 6.4. En supposant que k et l sont des variables globales que l'on choisira plus tard, coder la fonction `E(X)`, qui prend un `array X` de taille $(N+2)2$ et qui renvoie l'énergie associée. Noter que l'on peut à partir de `X` obtenir N via la commande `np.size(X,0)-2` et pour tout i obtenir `xi` via la commande `X[i,:]`.

Exercice 6.5. On rappelle que x_A et x_B sont fixés. Montrer que le gradient de E est donné par

$$\nabla E = \begin{pmatrix} 0 \\ \nabla E(\mathbf{x}_1) \\ \vdots \\ \nabla E(\mathbf{x}_N) \\ 0 \end{pmatrix}$$

avec

$$\nabla E(\mathbf{x}_i) = \nabla F(\mathbf{x}_i) + 2k(N+1)\left(\|\mathbf{x}_{i+1}-\mathbf{x}_i\| - \ell\right)\frac{\mathbf{x}_{i+1}-\mathbf{x}_i}{\|\mathbf{x}_{i+1}-\mathbf{x}_i\|} + 2k(N+1)\left(\|\mathbf{x}_{i-1}-\mathbf{x}_i\| - \ell\right)\frac{\mathbf{x}_i-\mathbf{x}_{i-1}}{\|\mathbf{x}_i-\mathbf{x}_{i-1}\|}$$

En utilisant cette expression, coder la fonction `dE(X)` qui prend un `array` de taille $(N+2)\times 2$, et qui renvoie un `array` de taille $(N+2)\times 2$.

Exercice 6.6. Minimiser la fonction E à l'aide d'une descente de gradient à pas constant. On choisira comme initialisation des points équidistants entre `xA` et `xB`. On prendra $N = 100$ points, $k = 0.02$ et $\ell =$ `np.linalg.norm(xA-xB)/N`. On représentera également l'évolution de la valeur de la fonction E au cours de l'évolution. Combien de variables a la fonction que l'on essaie de minimiser ?

Exercice 6.7. Afficher les lignes de niveau de F et l'élastique obtenu.

Exercice 6.8. Donner une approximation numérique du point col obtenu. Est-ce cohérent avec le dessin ?

Exercice 6.9. Afficher la ligne de niveau de F qui passe par le point col.

Pour affiner l'approximation. La méthode de l'élastique que nous venons de voir permet de trouver une première approximation grossière du point col. On peut raffiner cette approximation en utilisant ensuite une méthode de Newton. Dans le cas de la méthode de Newton multi-dimensionnelle les itérations sont définies par

$$x_{k+1} = x_k - \nabla^2 F(x_k)^{-1}\nabla F(x_k).$$

Exercice 6.10. Expliquer pourquoi cette méthode, initialisée suffisamment proche du point col, permet de trouver des points critiques qui ne sont pas des minima.

Exercice 6.11. Coder une fonction `HessF(X)` qui prend un `array` de taille 2 et renvoie un `array` de taille 2×2.

Exercice 6.12. Écrire un algorithme de Newton `Newton(df,HessF,x0,tol=1e-6,Niter=100)`.

Exercice 6.13. À l'aide de ce code déterminer une meilleure approximation du point col en utilisant pour initialisation l'approximation du point col obtenu par la méthode de l'élastique.

7. Descentes de gradient IV : gradient conjugué

7.1. **Première discussion et principe de la méthode.** Passons au dernier aspect de la descente de gradient en discutant la méthode du **gradient conjugué**. Le problème de la descente de gradient, même à pas optimal, c'est qu'elle converge très lentement si elle n'est pas bien initialisée. Même si c'est un exemple que nous connaissons désormais bien, considérons le cas d'une matrice

$$A := \begin{pmatrix} a_0 & 0 \\ 0 & a_1 \end{pmatrix}$$

avec $a_0 \neq a_1$, et posons

$$f : \mathbb{R}^2 \ni x = (x_0, x_1) \mapsto \frac{1}{2} a_0 x_0^2 + \frac{1}{2} a_1 x_1^2.$$

Pour trouver la solution en disons une itération, il faudrait, si l'on choisit de suivre la direction de descente du gradient, qu'il existe un réel $\tau \in \mathbb{R}$ tel que

$$\tau \nabla f(x_0) = x_0.$$

Or ceci se réécrit

$$x_{0,0} = \tau a_0 x_{0,0}\,, x_{0,1} = \tau a_1 x_{0,1}.$$

On voit donc que si $x_{0,0} x_{0,1} \neq 0$, il est impossible de remplir cette condition. Est-il néanmoins possible, si l'on itère suffisamment de fois l'algorithme de descente de gradient à pas optimal, d'arriver à un point x_k tel que x_{k+1} soit égal à 0 ? Reprenons la proposition 6.2 : on voit que les itérations de la descente de gradient à pas optimal, que nous noterons $\{x_k = (x_{0,k}, x_{1,k})\}_{k \in \mathbb{N}}$ sont données de manière itérative par

$$x_{0,k+1} = \frac{a_1^2 (a_1 - a_0) x_{0,k} x_{1,k}^2}{a_0^3 x_{0,k}^2 + a_1^3 x_{1,k}^2}\,, x_{1,k+1} = \frac{a_0^2 (a_0 - a_1) x_{0,k}^2 x_{1,k}}{a_0^3 x_{0,k}^2 + a_1^3 x_{1,k}^2}.$$

En particulier, si $x_{0,0} x_{0,1} \neq 0$ alors, pour tout $k \in \mathbb{N}$, $x_{k,0} x_{k,1} \neq 0$, et donc la descente de gradient à pas optimal ne converge pas en un nombre fini d'itérations.

Observons néanmoins qu'il n'y a *a priori* aucune raison pour que ce choix de direction de descente soit le meilleur. Arrivant en effet au point x_0, il faudrait plutôt choisir comme direction de descente celle qui nous emmène directement de x_1 à 0, c'est-à-dire choisir

$$d_1 = \frac{x_1}{\|x_1\|}.$$

Observons alors (ceci sera systématisé) que les vecteurs $d_0 = \nabla f(x_0)$ et d_1 sont A-orthogonaux :

$$\langle A d_0, d_1 \rangle = 0.$$

En particulier, dans ce cas très particulier où l'on cherche en fait à calculer 0, le choix de deux directions de descentes A-orthogonales permet d'obtenir un algorithme qui converge en deux itérations. Isolons cette notion d'orthogonalité :

Définition 7.1. Soit $A \in S_d^{++}(\mathbb{R})$. Deux vecteurs $x\,, y \in \mathbb{R}^d$ sont dits *A-orthogonaux* si, et seulement si,

$$\langle Ax, y \rangle = 0.$$

Il existe plusieurs manières d'attaquer la construction des méthodes de gradient conjugué. L'une, par laquelle nous allons commencer, est la plus naturelle mais ne fait pas forcément apparaître la nature hilbertienne de la méthode. La seconde, par projection, est plus absconse mais à de nombreux égards très instructive.

Dans toute la suite, $A \in S_d^{++}(\mathbb{R})$ et $b \in \mathbb{R}^d$ sont fixés, et l'on définit

$$f : \mathbb{R}^d \ni x \mapsto \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

7.2. **Une première approche de la méthode du gradient conjugué.** Une première manière d'étudier le gradient conjugué est de renforcer le principe de la descente de gradient à pas optimal. Rappelons que cette méthode, déjà étudiée lors de l'un des cours précédents, consiste à choisir, une itération x_k étant donnée (telle que $\nabla f(x_k) \neq 0$), x_{k+1} par la formule

$$x_{k+1} = x_k - \tau^* \nabla f(x_k) \text{ avec } \tau^* \text{ défini par } \tau^* := \underset{\tau>0}{\operatorname{argmin}} f(x_k - \tau \nabla f(x_k)).$$

Ce faisant, on a déjà complètement oublié tout ce qui avait été calculé aux étapes précédentes. Il est *a priori* bien meilleur de plutôt choisir x_{k+1} sous la forme

(7.1)
$$x_{k+1} = x_k - \sum_{i=0}^{k} \tau_i^* \nabla f(x_i) \text{ avec } \tau_0^*, \dots, \tau_k^* = \underset{\tau_0,\dots,\tau_n>0}{\operatorname{argmin}} f\left(x_k - \sum_{i=0}^{k} \tau_i \nabla f(x_i)\right).$$

Si l'on définit ainsi les itérations, on voit que les conditions d'optimalité s'écrivent

$$\forall i \in \{0, \dots, k\}, \langle \nabla f(x_{k+1}), \nabla f(x_i)\rangle = 0. \tag{7.2}$$

La condition (7.2) est bien plus forte que celle de la ,simple descente de gradient à pas optimal, qui ne faisait que générer des gradients qui étaient successivement orthogonaux.

Une conséquence immédiate de cette simple observation est la suivante :

Lemme 7.1. *L'algorithme décrit ci-dessus par la relation* (7.1) *converge en au plus d itérations.*

Démonstration. Notons $k^* := \max\{k \in \mathbb{N}, \nabla f(x_k) \neq 0\} \in \mathbb{N} \cup \{\infty\}$. Évidemment, la fonction f étant strictement convexe on sait que, pour tout $i \leq k^*$ on a

$$\nabla f(x_i) \neq 0.$$

Si, par l'absurde, on a

$$d < k^*$$

alors la famille $\{\nabla f(x_i)\}_{i=0,\dots,k^*}$ est une famille de vecteurs non-nuls et deux à deux orthogonaux. En particulier, c'est une famille libre. Or, en dimension d, les familles libres sont de cardinal au plus d, une contradiction. □

Le problème malgré tout, c'est qu'il n'est pas tout à fait évident de savoir si l'on a vraiment amélioré notre approche, en remplaçant un problème d'optimisation à d variables en une suite de problèmes d'optimisation en k variables. Montrons que cette méthode peut naturellement se réecrire comme une méthode itérative simple. Pour cela, il nous faut commencer par analyser les directions de descente générées par (7.1).

Une première propriété des directions de descente dans (7.1) Introduisons, pour $k \in \mathbb{N}$ tel que $\nabla f(x_k) \neq 0$, la direction de descente d_k choisie par la règle (7.1), c'est-à-dire

$$d_k := x_{k+1} - x_k.$$

Puisque pour tout $x \in \mathbb{R}^d$ on a

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

on en déduit que

$$\nabla f(x_{k+1}) = \nabla f(x_k) + Ad_k.$$

Par ailleurs, par définition même de x_{k+1}, il existe $(k+1)$-coefficients $\{\alpha_{k,j}\}_{j=0,\dots,k}$ tels que

$$d_k = \sum_{j=0}^{k} \alpha_{k,j} \nabla f(x_j).$$

Isolons ces deux informations :

$$\begin{cases} d_k = \sum_{j=0}^{k} \alpha_{k,j} \nabla f(x_j)\,, \\ \nabla f(x_{k+1}) = \nabla f(x_k) + Ad_k. \end{cases} \tag{7.3}$$

Par la condition d'orthogonalité (7.2) on en déduit que, pour tout $i \leq k-1$,

$$0 = \langle \nabla f(x_{k+1}), \nabla f(x_i) \rangle = \langle \nabla f(x_k), \nabla f(x_i) \rangle + \langle Ad_k, \nabla f(x_i) \rangle = \langle Ad_k, \nabla f(x_i) \rangle.$$

Mais, puisque, si $j \leq k-1$, on peut écrire

$$d_j = \sum_{i=1}^{j} \alpha_{j,i} \nabla f(x_i)$$

c'est-à-dire comme une somme de vecteurs A-orthogonaux à d_k, on en déduit la propriété cruciale du gradient conjugué :

$$\boxed{\forall k \neq j\,, \langle Ad_j, d_k \rangle = 0.} \tag{7.4}$$

Cette propriété de A-orthogonalité donne son nom à la méthode du gradient conjugué. Notons que la morale de ce que cet algorithme se termine bien en un nombre fini d'étapes, c'est qu'il n'était pas pertinent de choisir la plus profonde direction de descente pour le produit scalaire euclidien, mais plutôt pour le produit scalaire induit par la matrice A. Nous retrouverons ce point de vue dans notre présentation alternative de la méthode du gradient conjugué.

Par ailleurs, en prenant cette fois le produit scalaire avec $\nabla f(x_k)$ dans l'expression de $\nabla f(x_{k+1})$ on obtient

$$-\|\nabla f(x_k)\|^2 = \langle Ad_k\,, \nabla f(x_k) \rangle.$$

En particulier, si l'algorithme n'a pas terminé en x_k, $\langle Ad_k, \nabla f(x_k) \rangle \neq 0$. Mais on obtient en fait mieux !

Calcul des coefficients $\alpha_{k,i}$ Mais toutes ces considérations ne nous avancent pas plus sur le calcul effectif des coefficients $\{\alpha_{k,j}\}_{k,j}$, calcul qui nous permettrait d'exhiber une structure itérative simple.

Commençons par le lemme suivant :

Lemme 7.2. *Tant que l'algorithme du gradient conjugué n'a pas convergé, on a*

$$\forall k\,, \langle d_k, \nabla f(x_k) \rangle \neq 0.$$

Preuve du Lemme 7.2. On démontre cette propriété par récurrence. Pour $k = 1$, d_1 est, par définition, colinéaire à $\nabla f(x_1)$. Supposons la propriété vraie au rang k. Supposons par l'absurde, l'algorithme n'ayant pas terminé à l'itération $k+1$, que l'on ait

$$\langle d_{k+1}, \nabla f(x_{k+1})\rangle = 0.$$

Par construction, on obtient

$$d_{k+1} = \sum_{j=1}^{k} \alpha_{k+1,j} \nabla f(x_j).$$

Néanmoins, on sait également que

$$(d_i)_{i\leq K} = (\nabla f(x_i))_{i\leq K} \cdot \big(h\big)\,,$$

système que l'on peut inverser. On obtient ainsi que $d_{k+1} \in \mathrm{Vect}(d_1, \ldots, d_k)$, en contradiction avec la propriété (7.4) qui garantit la liberté de la famille $(d_1, \ldots, d_{k+1})$. □

Ainsi, quitte à multiplier d_k par une constante, on peut supposer que $\alpha_{k,k} = 1$, et l'on adoptera donc la convention

$$d_k = \sum_{j=1}^{k-1} \alpha_{k,j} \nabla f(x_j) + \nabla f(x_k).$$

Mais utilisons de nouveau la relation d'orthogonalité

$$\forall k \neq j\,, \langle Ad_k, d_j\rangle = 0.$$

Puisque $Ad_j = \nabla f(x_{j+1}) - \nabla f(x_j)$ il apparaît que

$$\langle d_k, \nabla f(x_{j+1})\rangle = \langle d_k, \nabla f(x_j)\rangle.$$

Si $j = k-1$ on en tire

$$\|\nabla f(x_k)\|^2 = \langle d_k, \nabla f(x_k)\rangle = \alpha_{k,k-1}\|\nabla f(x_{k-1})\|^2.$$

Si $j < k-1$ on en tire

$$\alpha_{k,j+1}\|\nabla f(x_{j+1})\|^2 = \alpha_{k,j}\|\nabla f(x_k)\|^2.$$

En particulier, on obtient, pour tout $j \leq k-1$,

$$\alpha_{k,j} = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_j)\|^2},$$

et, ainsi, la décomposition explicite

$$d_k = \sum_{j=1}^{k} \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_j)\|^2} \nabla f(x_k),$$

de sorte que

$$d_{k+1} = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2} d_k + \nabla f(x_{k+1}). \tag{7.5}$$

Nous sommes désormais en bonne voie pour réussir à mettre la méthode du gradient conjugué sous forme itérative. Pour conclure, il nous reste à calculer le pas optimal τ_k^* à faire dans la direction d_k.

On rappelle que τ_k^* est défini comme le minimiseur de la fonction

$$\tau \mapsto f(x_k - \tau d_k).$$

Par les mêmes arguments que ceux qui nous ont permis de calculer le pas optimal dans un cours précédent, on obtient immédiatement pour τ_k^* l'expression

$$\tau_k^* = \frac{\langle \nabla f(x_k), d_k \rangle}{\langle Ad_k, d_k \rangle}.$$

7.3. **Résumé de la méthode du gradient conjugué.** Si l'on résume toute l'analyse que nous venons de mener, la méthode du gradient conjugué peut être décrite de la manière suivante :

(1) Initialisation en un point $x_0 \in \mathbb{R}^d$. La première direction de descente est $d_0 = \nabla f(x_0)$, le premier pas est $\tau_0^* = \frac{\langle \nabla f(x_0), d_0 \rangle}{\langle Ad_0, d_0 \rangle}$ et $x_1 = x_0 - \tau_0^* d_0$.

(2) Pour tout $k \in \mathbb{N}^*$, on définit

$$d_k = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2} d_{k-1} + \nabla f(x_k)\,, \tau_k^* := \frac{\langle \nabla f(x_k), d_k \rangle}{\langle Ad_k, d_k \rangle}$$

et on pose

$$x_{k+1} = x_k - \tau_k^* d_k.$$

(3) L'algorithme converge en au plus d itérations.

Fiche de TD n°6 : Descente de gradient, gradient conjugué

Exercice 7.1 (Descente de gradient à pas optimal, le cas quadratique). On considère $A \in S_d^{++}(\mathbb{R})$ et la fonction

$$f : x \mapsto \frac{1}{2}\langle Ax, x\rangle.$$

On considère la descente de gradient à pas optimal. Une itération x_k étant fixée, déterminer le pas t_k tel que

$$f(x_k - t_k \nabla f(x_k)) = \inf_{t \geq 0} f(x_k - t\nabla f(x_k)).$$

Exercice 7.2 (Une variante de la descente de gradient). On rappelle que la norme $\|\cdot\|_\infty$ sur $\mathbb{R}^d$ est définie de la manière suivante : si $(e_1, \ldots, e_d)$ est la base canonique de $\mathbb{R}^d$ et si $x = \sum_{i=1}^d x_i e_i$ alors

$$\|x\|_\infty = \max_i |x_i|.$$

On note $\|\cdot\|$ la norme euclidienne usuelle.

On travaille avec une fonction f de classe $\mathscr{C}^2$, coercive, convexe, et on suppose qu'il existe $\ell, L > 0$ tels que

$$\forall x \in \mathbb{R}^d\,, \forall h \in \mathbb{R}^d\,, \ell\|h\|^2 \leq \langle \nabla^2 f(x)h, h\rangle \leq L\|h\|^2.$$

(1) Montrer que le problème d'optimisation

$$\inf_{x \in \mathbb{R}^d} f(x)$$

a une unique solution x^*.

(2) Démontrer que, pour tout $x \in \mathbb{R}^d$, on a

$$\frac{\ell}{2}\|x - x^*\| - \|\nabla f(x)\| \leq 0$$

et que pour tout $x \in \mathbb{R}^d$

$$f(x) - f(x^*) \leq \frac{1}{2\ell}\|\nabla f(x)\|^2.$$

(3) On étudie l'algorithme itératif suivant : une initialisation x_0 étant donnée on choisit, à chaque étape k la direction

$$d_k := -\frac{\partial f}{\partial x_{i(k)}}(x_k)e_i$$

où $i(k)$ est le premier indice tel que

$$\left|\frac{\partial f}{\partial x_{i(k)}}\right| = \|\nabla f(x_k)\|_\infty.$$

Le pas de descente est le réel $t_k \geq 0$ défini par

$$t_k = \text{argmin}_{t \geq 0} f(x_k + td_k)$$

et on pose

$$x_{k+1} = x_k + t_k d_k.$$

Montrer que le pas t_k est bien défini et que d_k est une direction de descente.

(4) Montrer que pour tout $t \geq 0$

$$f(x_{k+1}) \leq f(x_k + t_k d_k) \leq f(x_k) - t\|\nabla f(x_k)\|_\infty^2 + \frac{Lt^2}{2}\|\nabla f(x_k)\|_\infty^2.$$

(5) En déduire que

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|_\infty^2.$$

(6) En déduire que

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\ell}{dL}\right)(f(x_k) - f(x^*)).$$

(7) Quel est le taux de convergence de la suite des valeurs renvoyées par l'algorithme ? Est-il meilleur ou moins bon que l'algorithme de descente classique ?

Exercice 7.3 (Changement de produit scalaire). On note $\langle\cdot,\cdot\rangle$ le produit scalaire usuel de $\mathbb{R}^d$. Soit $A \in S_d^{++}(\mathbb{R})$ de valeurs propres $0 < \lambda_1 \leq \cdots \leq \lambda_d$. On définit la forme bilinéaire sur $\mathbb{R}^d$

$$\forall x, y \in \mathbb{R}^d, \quad \langle x, y\rangle_A := \langle xA, y\rangle = x^T A y.$$

(1) Montrer que $\langle\cdot,\cdot\rangle_A$ est un produit scalaire.

(2) Montrer que la norme associée $\|\cdot\|_A$ est équivalente à la norme usuelle $\|\cdot\|$, et plus précisément,

$$\forall x \in \mathbb{R}^d, \quad \lambda_1||x||^2 \leq ||x||_A^2 \leq \lambda_d\|x\|^2.$$

(3) Soit $B \in M_d(\mathbb{R})$. Montrer que

$$\langle x, By\rangle = \langle A^{-1}B^T Ax, y\rangle_A.$$

On dit que le dual de B est $A^{-1}B^T A$ pour le produit scalaire $\langle\cdot,\cdot,\rangle_A$.

(4) Pour les fonctions suivantes, calculer le gradient, **pour le produit scalaire** $\langle\cdot,\cdot\rangle_A$:

$$F_1(x) := \|x\|_A^2, \quad F_2(x) := \|x\|^2, \quad F_3(x) := \frac{1}{2}x^T Ax - b^T x.$$

On remarquera que le gradient dépend du produit scalaire...

Exercice 7.4 (Résolution de l'équation de Poisson). On veut dans cet exercice résoudre l'équation différentielle

$$\begin{cases} -u'' = f & \text{dans } [0;1], \\ u(0) = u(1) = 0. \end{cases} \tag{7.6}$$

Ici la fonction f est continue sur l'intervalle. Pour résoudre cette équation numériquement, on va mettre en œuvre des algorithmes de type descente de gradient ou gradient conjugué pour l'analyse d'un système discrétisé approché.

(1) Soit g une fonction de classe $\mathscr{C}^4$ sur $\mathbb{R}$, avec $\|g^{(4)}\|_{L^\infty} < \infty$. On se fixe un pas de discrétisation $\delta > 0$. Démontrer qu'il existe une constante C_0 telle que

$$\left| g''(x) - \frac{g(x+\delta) - 2g(x) + g(x-\delta)}{\delta^2} \right| \leq C_0\delta^2.$$

(2) On se fixe un entier $d > 1$ et on pose $\delta := \frac{1}{d+1}$. On veut approcher la solution u de (7.6) par un vecteur $U := (u_i)_{i=0,\dots,d+1}$ (noter que $u_0 = u_1 = 0$) en approchant le terme source f par $F = (F(i\delta))_{i=0,\dots,d+1}$. Est-il raisonnable de choisir U comme la solution de

$$\Delta_\delta U = F$$

où Δ_δ est la matrice définie par

$$\Delta_\delta = \frac{1}{\delta^2} \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & -1 & 2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -1 & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{pmatrix} ?$$

(3) Démontrer que Δ_δ est une matrice symétrique définie positive. En déduire que le système $\Delta_\delta U = F$ est bien posé (*i.e.* qu'il admet une unique solution).

(4) Calculer les valeurs propres (et les vecteurs propres associés) de la matrice Δ_δ. Que vaut $\mathrm{cond}(\Delta_\delta)$? En donner un équivalent quand $\delta \to 0$. *Indication : on pourra commencer par déterminer les réels λ et les fonctions $\phi \in \mathscr{C}^2(\mathbb{R};\mathbb{R})$ non identiquement nulles sur $[0;1]$ telles que $-\phi'' = \lambda\phi$.*

(5) Proposer une méthode de résolution numérique de (7.6).

Correction de la feuille de TD n°6

Correction 7.1. *La condition d'optimalité au premier ordre s'écrit, au pas optimal t_k (qui, nécessairement, est différent de 0),*

$$\langle \nabla f(x_k - t_k \nabla f(x_k)) \, , \nabla f(x_k) \rangle = 0.$$

Puisque, pour tout vecteur $y \in \mathbb{R}^d$, on a

$$\nabla f(y) = Ay$$

on en déduit, après quelques manipulations élémentaires, que

$$t_k = \frac{\langle Ax_k, Ax_k \rangle}{\langle A^2 x_k, Ax_k \rangle}.$$

Correction 7.2. *(1) La fonction f est coercive, elle admet donc un minimiseur x^*. Puisque la fonction est strictement convexe, ce minimiseur est unique.*

(2) On observe que pour tout $x \in \mathbb{R}^d$ on a, par le théorème des accroissements finis en x,

$$f(x^*) \geq f(x) + \langle \nabla f(x), x^* - x \rangle + \frac{\ell}{2} \|x - x^*\|^2.$$

Ainsi, par l'inégalité de Cauchy-Schwarz, on obtient la minoration

$$f(x^*) \geq f(x) + \|x - x^*\| \left(\frac{\ell}{2} \|x - x^*\| - \|\nabla f(x)\| \right),$$

de sorte que

$$\frac{\ell}{2} \|x - x^*\| - \|\nabla f(x)\| \leq 0.$$

Si d'un autre côte on utilise plutôt la minoration générale : pour tout $x, y \in \mathbb{R}^d$ on a

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\ell}{2} \|x - y\|^2$$

et que l'on minimise le membre de droite en $y \in \mathbb{R}^d$ on voit que le minimiseur y^* est donné par

$$y^* = x - \frac{1}{\ell} \nabla f(x)$$

et le membre de droite vaut alors $f(x) - \frac{1}{2\ell} \|\nabla f(x)\|^2$. En prenant $y = x^*$ on obtient la conclusion souhaitée.

(3) La fonction f étant coercive et strictement convexe, il en va de même de sa restriction à la droite $x_k + \mathbb{R} d_k$, de sorte que t_k est bien défini (il existe et il est unique). Par ailleurs,

$$\langle d_k, \nabla f(x_k) \rangle = -\|\nabla f(x_k)\|_\infty^2 < 0$$

et donc d_k est bien une direction de descente.

(4) Par définition de t_k on a

$$\forall t \in \mathbb{R} \, , f(x_{k+1}) \leq f(x_k + t d_k).$$

Or, pour tout $t \in \mathbb{R}$, il existe $z \in [x_k; x_k + td_k]$ tel que

$$f(x_k + td_k) = f(x_k) + t\langle \nabla f(x_k), d_k\rangle + \frac{t^2}{2}\langle \nabla^2 f(z)d_k, d_k\rangle$$
$$\leq f(x_k) - t\|\nabla f(x_k)\|_\infty^2 + \frac{Lt^2}{2}\|\nabla f(x_k)\|_\infty^2.$$

(5) On choisit un pas t^* qui minimise l'expression

$$-t\|\nabla f(x_k)\|_\infty^2 + \frac{Lt^2}{2}\|\nabla f(x_k)\|_\infty^2$$

et un calcul direct montre que $t^* = \frac{1}{L}$. Dans ce cas là, on obtient la majoration

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|_\infty^2.$$

(6) Commençons par observer que

$$\|x\|_\infty^2 \geq \frac{\|x\|^2}{d}.$$

Ainsi on observe que

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|_\infty^2 \leq -\frac{1}{2dL}\|\nabla f(x_k)\|^2.$$

Mais on sait en outre, par la première question, que

$$0 \leq f(x_k) - f(x^*) \leq \frac{1}{2\ell}\|\nabla f(x_k)\|^2$$

et donc

$$f(x_{k+1}) - f(x^*) \leq f(x_k) - f(x^*) - \frac{1}{2L}\|\nabla f(x_k)\|_\infty^2 \leq (1 - \frac{\ell}{dL})(f(x_k) - f(x^*)).$$

On en déduit par le critère de convergence linéaire que cette méthode converge linéairement (et moins rapidement que la méthode de descente de gradient classique).

Correction 7.3. *(1) L'application est clairement bilinéaire et symétrique (car A est symétrique). De plus, pour tout $x \in \mathbb{R}^d$, on a*

$$\langle x, x\rangle_A = \langle x, Ax\rangle \geq \lambda_1\|x\|^2,$$

qui est positif, et qui ne s'annule que pour $x = 0$. Donc $\langle\cdot,\cdot\rangle_1$ est un produit scalaire.

(2) C'est la caractérisation des valeurs propres par le min-max.

(3) On rappelle que pour tout matrice C, on a $\langle x, Cy\rangle = \langle C^T x, y\rangle$ (pour le produit scalaire usuel). On obtient donc, en utilisant que A est symétrique,

$$\langle x, By\rangle_A = \langle x, ABy\rangle = \langle B^T A^T x, y\rangle = \langle A^{-1}B^T Ax, Ay\rangle = \langle A^{-1}B^T Ax, y\rangle_A.$$

(4) On revient à la définition du gradient à partir de la différentielle. On a

$$F_1(x + h) = \|x + h\|_A^2 = \|x\|_A^2 + \|h\|_A^2 + 2\langle x, h\rangle_A = F_1(x) + 2\langle x, h\rangle_A + o(h).$$

Le terme linéaire est déjà sous la forme d'un produit scalaire, donc $\nabla F_1(x) = 2x$. Pour F_2, on obtient

$$F_2(x + h) = \|x + h\|^2 = \|x\|^2 + 2\langle x, h\rangle + o(h).$$

Pour trouver le gradient associé au nouveau produit scalaire, il faut réécrire le terme linéaire comme un produit scalaire $\langle \cdot, \cdot \rangle_A$. On trouve que $\langle 2x, h\rangle = \langle 2A^{-1}x, h\rangle_A$, donc $\nabla F_2(x) = 2A^{-1}x$.
Enfin, pour $F_3(x)$, on obtient

$$F_2(x+h) = F_3(x) + \frac{1}{2}h^T Ax + \frac{1}{2}x^T Ah - b^T h + o(h) = F_3(x) + x^T Ah - (A^{-1}b)^T Ah + o(h).$$

Ainsi, on trouve $\nabla F_3(x) := x - A^{-1}b$.

Correction 7.4. *(1) On fait un développement limité à l'ordre 4 de la fonction g : pour tout x et pour tout h il existe $\xi_\pm \in [x; x \pm h]$ tels que*

$$g(x \pm h) = g(x) \pm hg'(x) + \frac{h^2}{2}g''(x) \pm \frac{h^3}{3!}g^{(3)}(x) + \frac{h^4}{4!}g^{(4)}(\xi_\pm).$$

On en déduit que

$$g(x+\delta) + g(x-\delta) - 2g(x) = \delta^2 g''(x) + \frac{\delta^4}{4!}\left(g^{(4)}(\xi_+) + g^{(4)}(\xi_-)\right).$$

On en déduit immédiatement l'estimation voulue.

(2) On observe que pour tout $x \in \mathbb{R}^d$ on a

$$\begin{aligned}\delta^2\langle U_\delta x, x\rangle &= 2x_1^2 - x_1x_2 - \sum_{i=1}^{d-1}(x_ix_{i-1} + 2x_i^2) - \sum_{i=3}^{d} x_ix_{i-1} + 2x_d^2 - x_{d-1}x_d \\ &= \sum_{i=2}^{d}(x_i - x_{i-1})^2 + x_1^2 + x_d^2.\end{aligned}$$

On en déduit la stricte positivité de U_δ. En particulier, le système proposé à l'étude est bien posé et l'on a existence et unicité d'une solution.

(3) Calculons les valeurs propres et les vecteurs propres du laplacien (continu). On voit par un calcul classique que les valeurs propres sont les $k^2\pi^2$ (pour $k \in \mathbb{N}$) et que les fonctions propres associées sont les $\sin(k\pi\cdot)$. Définissons, par analogie, les vecteurs

$$\Phi_k := (\sin(k\pi i\delta))_{i=0,\dots,d+1}.$$

Alors on voit que la i-ième composante de $U_\delta\Phi_k$ est

$$\delta^2(U_\delta\Phi_k)_i = -\sin(k\pi(i-1)\delta) + 2\sin(k\pi\delta) - \sin(k\pi(i+1)\delta).$$

Néanmoins, puisque $\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b)$ on en déduit que

$$(U_\delta\Phi_k)_i = \mu_k(\Phi_k)_i$$

avec

$$\mu_k = \frac{2}{\delta^2}(1 - \cos(k\pi\delta)) = \frac{2}{\delta^2}\left(1 - \cos\left(\frac{k\pi}{d+1}\right)\right)$$

pour $k = 1, \dots, d$. On a trouvé d valeurs propres distinctes, on les a donc toutes trouvées. En particulier, le conditionnement de la matrice vaut

$$\frac{\left(1 - \cos\left(\frac{d\pi}{d+1}\right)\right)}{\left(1 - \cos\left(\frac{\pi}{d+1}\right)\right)} \sim_{\delta\to 0/d\to\infty} \frac{4}{\pi}(d+1)^2 = \frac{4}{\pi\delta^2}.$$

Feuille de TP n°5 : Différences finies et gradient conjugué

7.4. **Résolution d'équations différentielles.** Dans la lignée de la feuille de TD n°6, voyons comment l'on peut résoudre grâce à des algorithmes d'optimisation certaines équations différentielles. Dans la première partie de ce TP, on cherche à trouver une solution 1-périodiques de l'équation différentielle ordinaire :

$$\dot{x}(t) + x(t) = e^{\sin(2\pi t)}.$$

Pour cela, nous découpons le segment $[0,1]$ en L points régulièrement espacés $[t_0, t_2, \dots, t_{L-1}] \in [0,1)$ avec $t_0 = 1$ et $t_L = 1$ (attention aux indices), et nous représentons une fonction $x : [0,1] \to \mathbb{R}$ par le vecteur

$$x := \begin{pmatrix} x(t_0) \\ \vdots \\ x(t_{L-1}) \end{pmatrix} \in \mathbb{R}^L.$$

De plus, nous approchons la dérivée seconde par des différences finies (avec les conditions de périodicité)

$$x''(t_i) \approx Dx := L^2\big(x(t_{i+1}) + x(t_{i-1}) - 2x(t_i)\big).$$

Exercice 7.5. Montrer que D peut-être vu comme une matrice de $\mathcal{M}_L(\mathbb{R})$, puis écrire une fonction `getD(L)` qui renvoie cette matrice. Pour ce faire, on pourra utiliser la fonction modulo de python (`a%b` pour `a` modulo `b`).

En pratique, on prendra des très grandes valeurs de L. Il n'est pas donc pas pratique de stocker la matrice D (presque tous ses coefficients sont nuls). On préfère utiliser une fonction "multiplier par D".

Exercice 7.6. Écrire une fonction `dd(x)` qui prend un array `x` de taille L, et renvoie l'array D`x` de taille L, sans construire D. Autrement dit `dd(x)` renvoie directement le vecteur `x`$''$. Pour ce faire, on se renseignera sur la fonction `roll` de Python (qui décale les éléments d'un `array`).

Exercice 7.7. Compiler la cellule suivante, et commenter le résultat :

```
L = 10000
x = random.rand(L)

print("\nCalcul de ddx en construisant la matrice :")
%time v1 = np.dot(getD(L),x)

print("\nCalcul de ddx directement :")
%time v2 = dd(x)

print("\nErreur entre les deux calculs : ", np.linalg.norm(v1
      - v2))
```

Exercice 7.8. Montrer que l'équation initiale peut s'écrire sous la forme $Ax = b$, avec $A = D + Id_L$ et b le vecteur de taille L qui contient les valeurs $\exp(\sin(2\pi t_i))$.

Exercice 7.9. Écrire une fonction `getb(L)` qui renvoie le vecteur b (Attention : dans la fonction `linspace(0,1,K)` de Python, le dernier élément de la liste créée est 1).

Exercice 7.10. Écrire une fonction `A(x)` qui renvoie le résultat de la multiplication de x par $A = D + Id_L$.

On passe à la résolution de l'équation. Nous venons de voir qu'il suffisait de résoudre le système $Ax = b$. Pour cela, on utilise l'algorithme du gradient conjugué. On rappelle que le gradient conjugué est défini par l'initialisation

$$x_0 = 0,\ p_0 = b,\ r_0 = b$$

puis

$$\begin{aligned}
\alpha_{n+1} &= \frac{r_n^T r_n}{p_n^T A p_n}, \\
x_{n+1} &= x_n + \alpha_{n+1} p_n, \\
r_{n+1} &= r_n - \alpha_{n+1} p_n, \\
\beta_{n+1} &= \frac{r_{n+1}^T r_{n+1}}{r_n^T r_n}, \\
p_{n+1} &= r_{n+1} + \beta_{n+1} p_n.
\end{aligned}$$

De plus, l'agorithme s'arrête dès que $n > L + 2$, ou que $\|r_n\|$ est plus petit qu'une certaine tolérance.

Exercice 7.11. Ecrire une fonction `solveGC(A,b,tol=1e-6)` qui trouve la solution de $Ax = b$ avec l'algorithme du gradient conjugué. On fera attention au fait que `A` est ici une fonction, et non une matrice.

Exercice 7.12. Résoudre l'équation initiale avec l'algorithme du gradient conjugué avec $L = 1000$. Vérifier que votre solution `x` vérifie bien $dd(x) + x = b$.

7.5. **Reconstruction de sources musicales : l'algorithme MUSIC.** Dans l'algorithme MUSIC, on cherche la position de N sources placés en les points s_i $(i = 1, \dots, N)$, en utilisant les signaux captés par M récepteurs placés en les points r_j $(j = 1, \dots, M)$. Dans cet exercice, on regarde le problème dans le plan $\mathbb{R}^2$, et les positions des sources et des récepteurs seront sous la forme d'array de taille $N \times 2$ et $M \times 2$ respectivement.

Exercice 7.13. Ecrire une fonction `getConfiguration(N,M)` qui renvoie les `arrays` sources et recepteurs. On pourra choisir les sources aléatoirement dans le carré $[-1, 1]^2$ et les récepteurs disposés régulièrement sur le cercle de centre $(0, 0)$ et de rayon 2. Vérifier que le code fonctionne en l'utilisant pour représenter visuellement les emplacements des sources et des capteurs pour des valeurs particulières de M et N.

On suppose que chaque source émet un signal indépendant dans le temps. Ici, on prendra des signaux aléatoires entre -1 et 1. De plus, comme il n'y a pas d'échelle de temps dans le problème, on peut supposer sans perte de généralité que les signaux émettent un signal à fréquence 1, et les signaux émettent pendant un temps $T \in \mathbb{N}^*$.

Exercice 7.14. Ecrire une fonction `getInitialSignals(N,T=1000)` qui renvoie un `array` F de taille $N \times T$ de signaux aléatoires indépendants.

En pratique, on n'a pas accès à $F(t)$, mais seulement à ce qu'enregistrent les récepteurs. Le récepteur j enregistre

$$g_j(t) = \sum_{i=1}^{N} f_i(t) \frac{1}{\|r_j - s_i\|}.$$

ou encore

$$G(t) = \Phi(s_1; \dots; s_N)F(t) \quad \text{avec } \Phi(x_1; \dots; x_N) = (\Phi(x_1)\Phi(x_2)\dots\Phi(x_N))$$
$$\text{où} \quad \Phi(x) := (\frac{1}{\|r_1 - x\|}, \dots, \frac{1}{\|r_M - x\|})^T.$$

Exercice 7.15. : Ecrire la fonction `Phi(X, recepteurs)` qui renvoie l'`array` $\Phi(x_1, \dots, x_d)$ de taille $M \times d$, où X est un array de taille $d \times 2$, et où les (x_i) sont les lignes de X. On pourra calculer d avec `d = np.shape(X)[0]`, et M avec `M = shape(recepteurs)[0]`.

Exercice 7.16. Ecrire une fonction `getRecordedSignals(sources, recepteurs, F)` qui renvoie $G(t)$.

On suppose maintenant qu'on n'a pas accès à l'array source, et qu'on ne connait seulement la position des récepteurs et G. On a donc la configuration fixée suivante :

```
N,M,T = 4,30,1000
sources, recepteurs = getConfiguration(N,M)
F = getInitialSignals(N,T)
G = getRecordedSignals(sources, recepteurs, F)
```

Exercice 7.17. Calculer la matrice de corrélation $C = GG^T$, et vérifier que C n'a que N grandes valeurs propres.

Exercice 7.18. Calculer la matrice de projection P^* sur les N vecteurs propres de C correspondants aux plus grandes valeurs propres de C. On rappelle que si $Cu_i = \lambda_i u_i$, alors la matrice de projection sur $Vect(u_i)$ est $u_i u_i^T$.

Exercice 7.19. Ecrire une fonction `distanceMUSIC(x)` qui prend un point $x \in \mathbb{R}^2$, et renvoie la distance $\|P^*(\Phi(x)) - \Phi(x)\|$

Exercice 7.20. Tracer les courbes de niveau de la fonction $log(distanceMUSIC)$ sur $[-1, 1] \times [-1, 1]$, ainsi que la vrai position des sources.

8. Révisions & Exercices classiques

Exercice 8.1. Soit $a > 0$, $F : x \mapsto ax^2$ et $\tau \in \mathbb{R}$. On pose $x_0 = 1$ puis $x_{n+1} = x_n - \tau F'(x_n)$.

(1) Comment se comporte la suite (x_n) pour les différentes valeurs de τ ?

(2) On pose maintenant $x_{n+1} = x_n - \tau \frac{1}{\varepsilon}[F(x_n + \varepsilon) - F(x_n)]$ (différence finie). Que se passe-t-il maintenant ?

(3) Que se passe-t-il avec des différences finies centrées ?

Exercice 8.2. On considère la fonction

$$f : \mathbb{R}^2 \ni (x, y) \mapsto x^2 + 2xy + 3y^2 - x - y.$$

(1) Mettre f sous la forme $\frac{1}{2}\langle A\cdot, \cdot\rangle - \langle b, \cdot\rangle$ avec A symétrique.

(2) Le problème d'optimisation $\inf_{\mathbb{R}^2} f$ a-t-il une solution ? Cette solution est-elle unique ?

(3) Si un minimiseur x^* existe, calculer la conditionnement de f en ce minimiseur ?

(4) Proposer une méthode de résolution numérique du problème et la coder en Python (à faire chez vous).

(5) Mêmes questions pour la fonction

$$f : \mathbb{R}^2 \ni (x, y) \mapsto x^2 + 2xy - y^2 + 3x.$$

Exercice 8.3. Soit $J : \mathbb{R}^2 \ni (x, y) \mapsto y^4 - 3xy^2 + x^2$.

(1) Déterminer les points critiques de la fonction J.

(2) Soit $(a, b) \in \mathbb{R}^2$. Montrer que 0 est un minimum local de l'application $\xi \mapsto J(\xi a, \xi b)$. En particulier, (0,0) est un minimum local le long de toute droite passant par 0.

(3) $(0, 0)$ est-il un minimum local de J le long de la parabole d'équation $y^2 = x$?

Exercice 8.4 (Méthode des puissances itérées). Soit $A \in S_d^{++}(\mathbb{R})$. On suppose que les valeurs propres de A sont toutes distinctes. On les note $0 < \lambda_1 < \cdots < \lambda_d$, et $(u_1, \dots u_d)$ est une base orthonormale de vecteurs propres associés.Soit $b \in \mathbb{R}^d$ tel que $\langle b, u_d\rangle > 0$. Montrer que

$$\lim_{n\to\infty} \frac{A^n b}{\|A^n b\|} = u_d, \quad \text{et que} \quad \lambda_d = \lim_{n\to\infty} \frac{\|A^{n+1} b\|}{\|A^n b\|}.$$

Email address: mazari@ceremade.dauphine.fr