

UNIVERSITÉ DE RENNES 1

Optimisation de paramètres

TP 1

Enzo Le Bouëdec
9 février 2018
enzo.lebouedec@etudiant.univ-rennes1.fr

Exercice 1 : Méthodes à pas constant

Gradient à pas constant

Dans un premier temps on cherche à programmer l'algorithme de descente du gradient à pas constant. On se propose de vérifier les résultats obtenus lors de séances de TD en étudiant la fonction $f(x)$ (1).

$$f(x) = \sum_{i=1}^n (x_i - i)^2 \quad (1)$$

On sait que pour cette fonction le minimum théorique est $x_i^* = i, i = 1, \dots, n$. D'après la figure 1 on voit que l'on converge vers le minimum théorique pour des pas inférieurs à 1. Ceci correspond aux résultats obtenus analytiquement. En effet, pour un pas de 1, on bouclait autour de la position initiale, et pour un pas supérieur, il y avait divergence. D'autre part, on constate aussi que le pas optimal est 0.5 et que le résultat exact est alors atteint. Comme il n'y a plus convergence pour des pas supérieurs à 1, la partie droite du graphique de gauche n'est donc pas à interpréter mathématiquement. Elle représente le nombre d'itérations avant d'atteindre un NaN en divergent du résultat. On observe exactement les mêmes résultats pour $n = 1$ et $n = 2$.

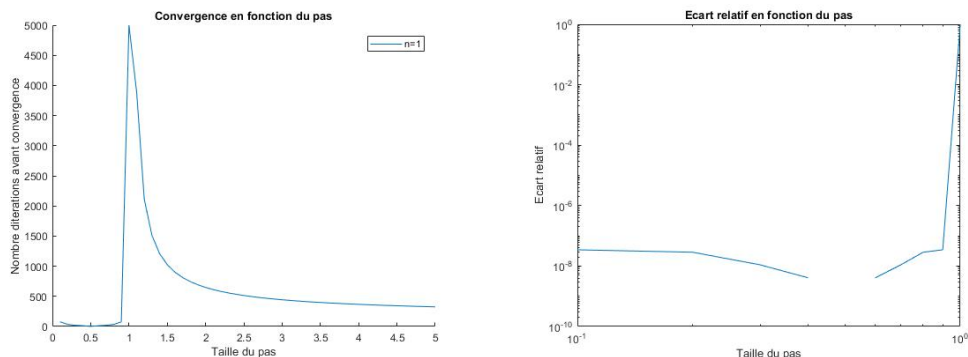


FIGURE 1 – Convergence et écart relatif de la descente du gradient à pas constant pour $f(x) = \sum_{i=1}^n (x_i - i)^2$.

Gradient conjugué à pas constant

On se propose maintenant d'étudier la même fonction mais avec l'algorithme de descente du gradient conjugué de Dai Yuan à pas constant (GC-DYCST). Comme on peut le voir sur la figure 2, l'algorithme converge en au plus deux itérations. Ces résultats numériques sont en accord avec les

résultats analytiques obtenus en séance de TD. On constate encore une fois que si le pas optimal (0.5) est choisi, on obtient une convergence en une itération, comme avec la descente de gradient à pas constant (GCST).

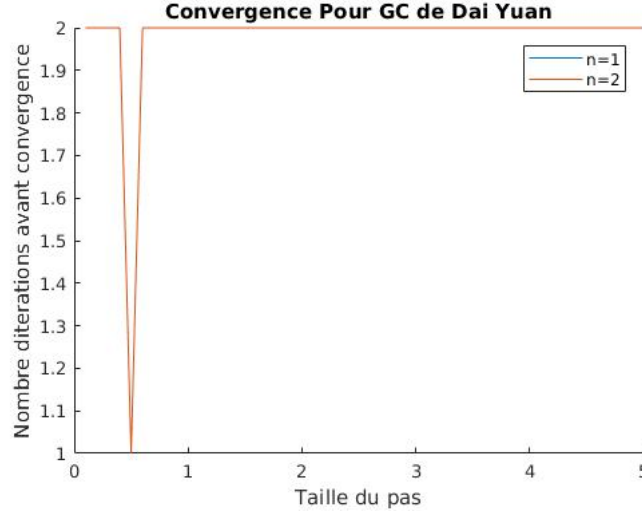


FIGURE 2 – Convergence GCDYCST pour $f(x) = \sum_{i=1}^n (x_i - i)^2$

Question 4

On étudie maintenant une autre fonction : $g(x)$ (2).

$$g(x) = \sum_{i=1}^n \frac{1}{i} (x_i - i)^2 \quad (2)$$

On souhaite comparer les performances des deux algorithmes sur cette fonction pour différentes valeurs du pas. On constate sur la figure 3 que le gradient conjugué (GCDYCST) possède aussi un pas maximal au dessus duquel il ne converge plus vers la solution. Cependant cette valeur seuil est supérieure à celle du gradient à pas constant. Le graphique représentant l'erreur en fonction du pas confirme que au dessus du pas "critique" l'algorithme ne converge plus vers la solution exacte (et même diverge).

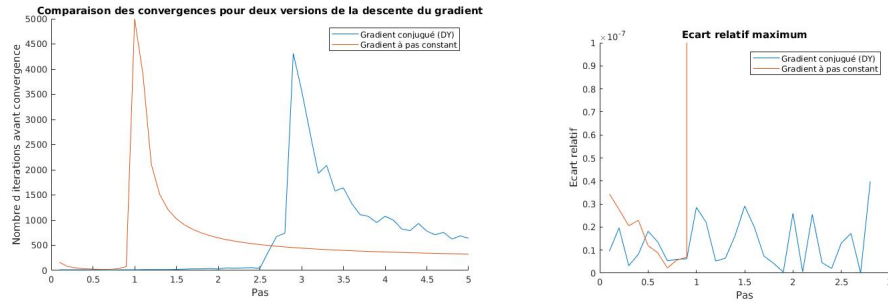


FIGURE 3 – Convergence et écart relatif de la descente du gradient conjugué (DY) et du gradient à pas constant pour $g(x) = \sum_{i=1}^n \frac{1}{i}(x_i - i)^2$.

Question 5

On cherche à déterminer le minimum de la fonction oscillante définie en (3).

$$s(x) = (x - 1)^2 + \sin\left(4\pi\left(x - \frac{9}{8}\right)\right) \quad (3)$$

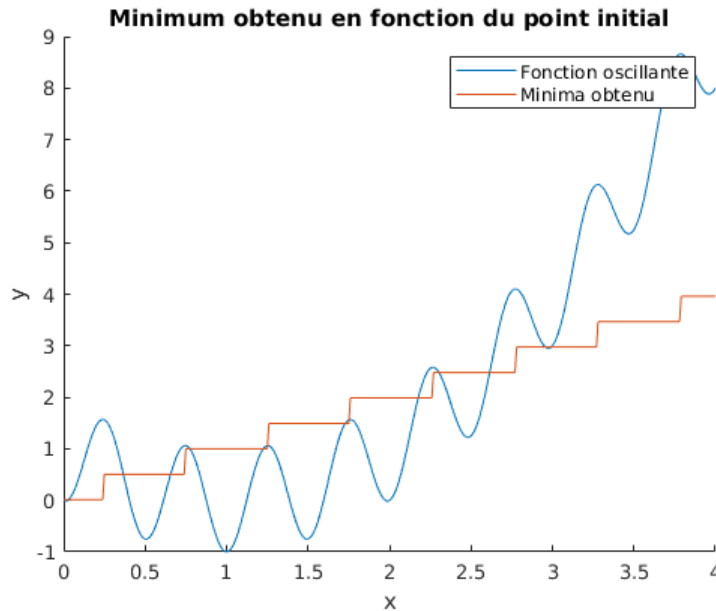


FIGURE 4 – Minimisation de la fonction s (3) en fonction du point initial

On constate sur la figure 4 que le minimum obtenu par la descente du

gradient dépend du point initial. La méthode de la descente du gradient converge donc vers un minimum local, mais elle n'assure pas qu'il soit le minimum global. Il est donc possible de faire plusieurs essais d'initialisation pour pallier à ce problème, ou encore de faire une analyse de la fonction à minimiser.

Exercice 2 : Méthodes à pas optimal

Question 1

On cherche dans un premier temps à valider les méthodes à pas optimaux qui ont été implémentées. Pour ce faire on peut s'appuyer sur des résultats analytiques obtenus en TP. En effet, les deux méthodes doivent converger en exactement une itération dans le cas de la minimisation de la fonction $f(x)$ (1). De plus, dans le cas de la fonction $g(x)$ (2) on s'attend à ce que le gradient conjugué converge en exactement deux itérations. On retrouve bien numériquement ces comportements. Par exemple, pour la fonction $g(x)$ on obtient les résultats présentés dans le tableau 1.

TABLE 1 – Nombres d'itérations avant convergence avec une tolérance de $1e^{-7}$ pour $g(x)$ avec $n = 2$ et $pas = 0.5$

Algorithme	Nombre d'itérations
Gradient à pas constant (GCST)	29
Gradient à pas optimal (GOPT)	16
Gradient conjugué à pas optimal (GCDYOPT)	2

Question 2

On cherche maintenant à étudier l'impact de la taille de la dimension de l'espace n pour la minimisation de $f(x)$ et $g(x)$. Dans le cas de la fonction $f(x)$ GOPT et GCDYOPT convergent vers la solution en une itération quelque soit la dimension de l'espace. Dans le cas de $g(x)$ la dimension de l'espace semble avoir un fort impact, comme on peut le constater dans la figure 5. En effet on constate, pour les deux méthodes, que le nombre d'itérations avant de converger vers la solution augmente avec la dimension de l'espace. On constate aussi que GCDYOPT nécessite significativement moins d'itérations que GOPT. Il en est de même pour le nombre d'évaluation de la fonction et de son gradient comme en témoigne la figure 6. Il est intéressant de noter que, si GOPT paraît ne plus converger vers la solution à partir d'un certain rang, c'est parce que le nombre maximal d'itérations a été atteint.

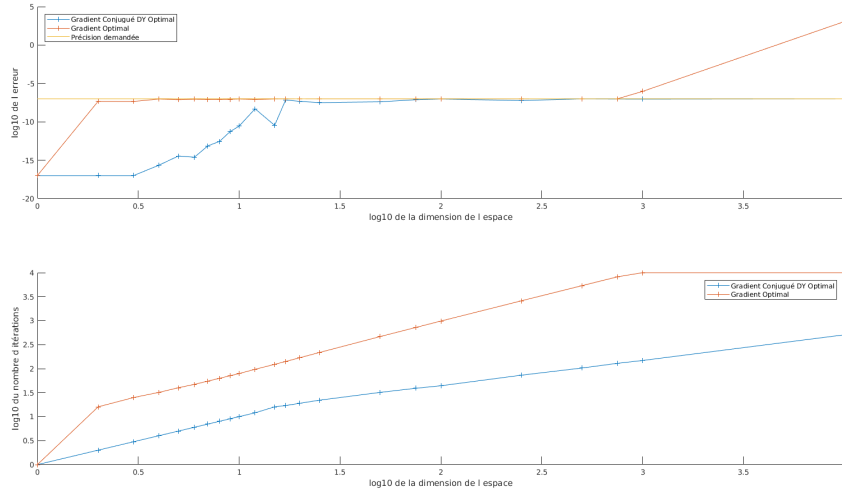


FIGURE 5 – Descente de GOPT et GCDYOPT sur $g(x)$ pour différents n

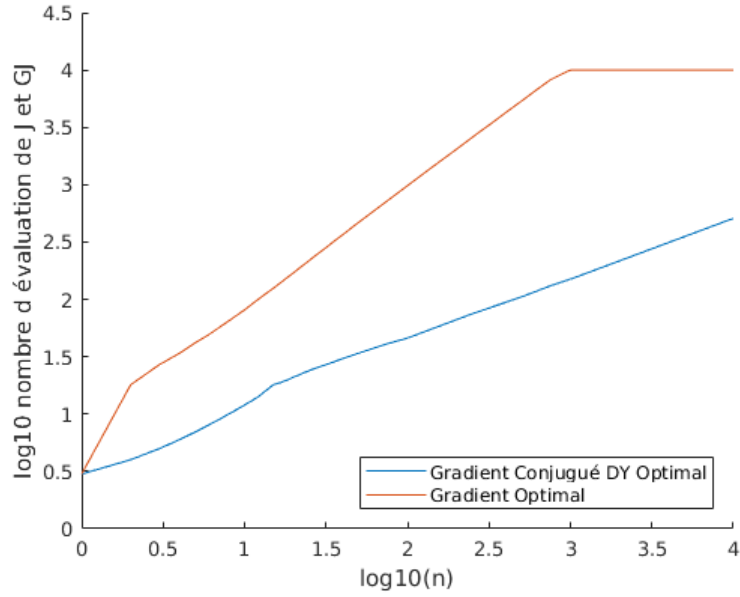


FIGURE 6 – Nombre d'évaluation de $g(x)$ et de son gradient pour GOPT et GCDYOPT en fonction de n

Pour avoir une meilleur idée de la différence entre ces deux algorithmes, il est intéressant de tracer leur évolution vers le minimum. Ceci est présenté dans la figure 7. On constate que le gradient à pas optimal (GOPT) à du

"mal" à descendre au fond de la vallée car il suit la direction locale du gradient. A l'inverse le gradient conjugué (GCDYOPT) trouve une direction de descente plus efficace vers le minimum en gardant en mémoire le gradient précédent et en le combinant au gradient local. Ainsi le GCDYOPT converge en 3 itérations contre 16 pour le GOPT pour une tolérance fixée de $1e^{-7}$.

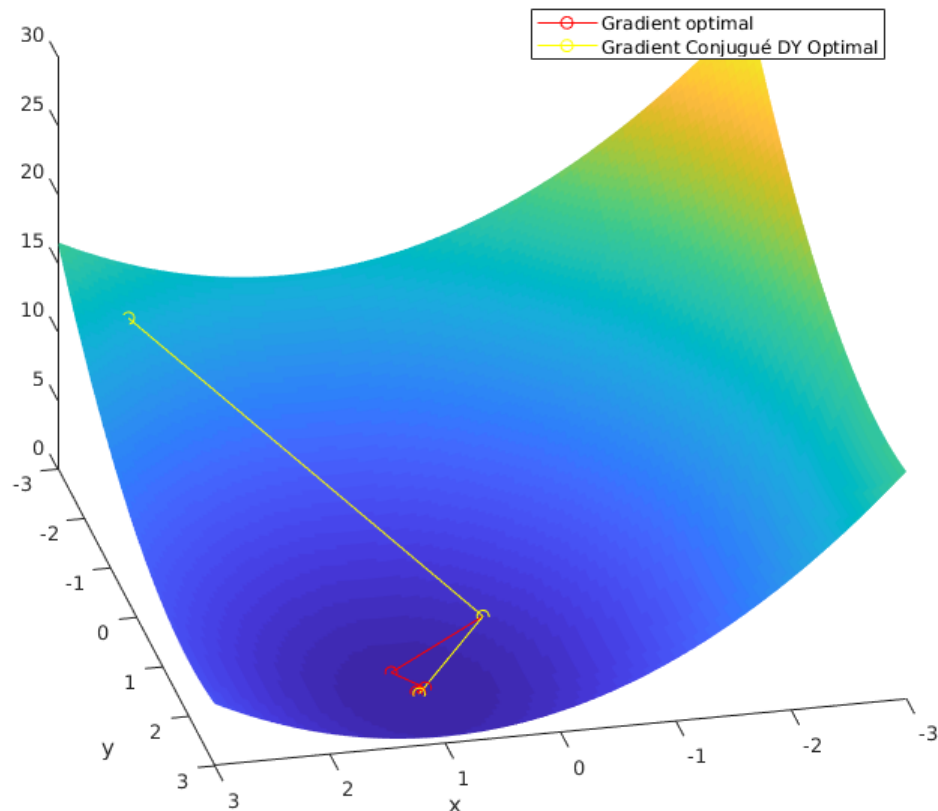


FIGURE 7 – Descente de GOPT et GCDYOPT sur $g(x)$ avec $n = 2$

Exercice 3 : Méthodes disponibles dans Matlab

Matlab possède une variété de fonctions de minimisations déjà implémentées. On peut notamment citer `fminsearch`. On cherche alors à comparer les performances de cette fonction par rapport aux algorithmes de descente de gradient implémentés au cours de ce TP. On peut commencer par comparer sur un cas testé précédemment. D'après le tableau 2 on voit que `fminsearch`

paraît bien moins efficace que les algorithmes du gradient pour arriver à la solution, quels qu'ils soient. Elle ne semble pas non plus réussir à minimiser la fonction $g(x)$ quand $n = 100$, dans le sens où en 10000 itérations elle n'a toujours pas convergé. Quand on l'utilise pour minimiser $s(x)$, elle s'arrête aussi sur le problème des minimas locaux.

TABLE 2 – Nombres d'itérations avant convergence avec une tolérance de $1e^{-7}$ pour $g(x)$ avec $n = 2$, $pas = 0.5$ et $x_0 = [-15, 15]$

Algorithme	$g(x)$	$f(x)$
Gradient à pas constant (GCST)	29	2
Gradient à pas optimal (GOPT)	16	1
Gradient conjugué à pas optimal (GCDYOPT)	2	1
fminsearch	72	76

Exercice 4 : Minimisation de la fonction de Rosenbrock

On cherche ici à comparer toutes les méthodes précédemment introduites sur une fonction qui a une vallée circulaire : la fonction de Rosenbrock, notée $h(x_1, x_2)$ (4) qui est représentée sur la figure 8.

$$h(x_1, x_2) = (1 - x_1^2) + 100(x_2 - x_1^2)^2 \quad (4)$$

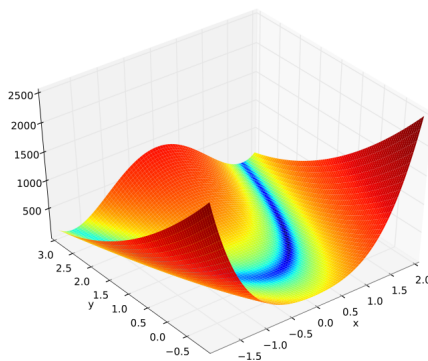


FIGURE 8 – Représentation graphique de la fonction de Rosenbrock

Comme il n'est pas aisé de déterminer la formule analytique du pas optimal pour cette fonction, les algorithmes GOPT et GCDYOPT ne seront pas testés sur cette fonction. D'après les conclusions des précédentes parties, ils seraient pourtant à priori plus efficaces que leurs homologues GCST et

GCDYCST. Les résultats pour différentes tailles de pas pour ces deux algorithmes sont présentés dans les figures 9 et 10. On constate que pour des pas supérieurs à $1e^{-3}$ il y a divergence des résultats (il est donc impossible de représenter les résultats sur la figure 10). On constate encore que GCDYCST nécessite moins d'itérations et converge vers une meilleure solution que GCST.

Il est intéressant de noter que les résultats divergent aussi si la solution initiale choisie est trop éloignée de la solution exacte. Ce comportement ne se retrouve pas avec `fminsearch` qui semble plus robuste pour la minimisation de la fonction de Rosenbrock. En effet elle converge vers le bon minimum même avec une point de départ $x_0 = [15, 35]$, ce qui n'est pas le cas de GCST et GCDYCST.

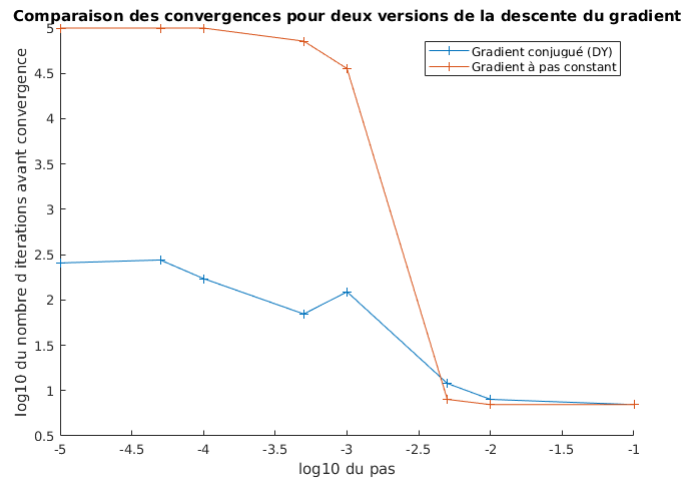


FIGURE 9 – Nombre d'itérations en fonction du pas

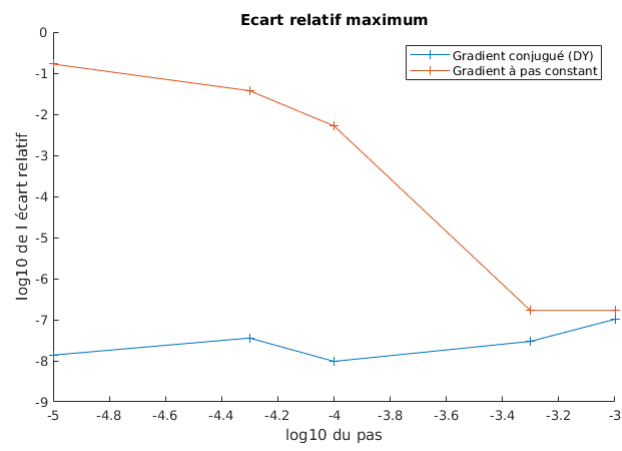


FIGURE 10 – Écart relatif en fonction du pas