Sorbonne Université

UE 4IN910 - « Algorithmique numérique »
Master Informatique - M1 SFPN

Rapport de TME
TME n°2 - Introduction à l'optimisation

Calle Viera Andersson

Enseignant
GRAILLAT Stef

Table des matières

Introduction

Dans ce T.M.E nous allons nous réaliser une introduction à l'optimisation, surtout en dimension 2. Pour cela nous allons réaliser une implémentation de l'algorithme de Newton sur la fonction de Rosenbrock. Par la suite nous utilserons la méthode de Wolfe pour trouver un pas optimal.
Finalement on analysera une méthode heuristique.

Le code est réalisé en MatLab et fournit avec le rapport.
On lance les script **exo6.m** et **exo7.m** pour les deux premiers exercices et le code du dernier est dans le fichier **NelderMeade.m**.

1 Exercice 6 (Fonction de Rosenbrock et méthode de Newton) :

Soit $x \in \mathbb{R}^2$, on définit $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$. On peut faire tourner **minimumRosenbrock.m** pour calculer le gradient g et la hessienne H de f grâce à la Symbolic Math Toolbox.

Question 1 :

Grâce à notre script on obtient les équations suivantes :

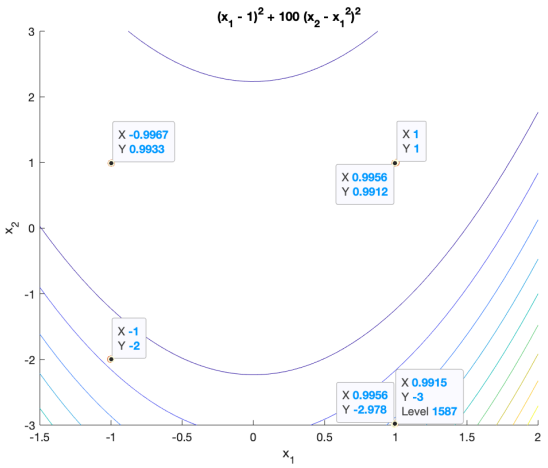
$$g(x) = \begin{pmatrix} 2x_1 - 400x_1(x_2 - x_1^2) - 2 \\ 200x_2 - 200x_1^2 \end{pmatrix} \qquad H(x) = \begin{pmatrix} 2 - 400x_2 + 1200x_1^2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}$$

Question 2 :

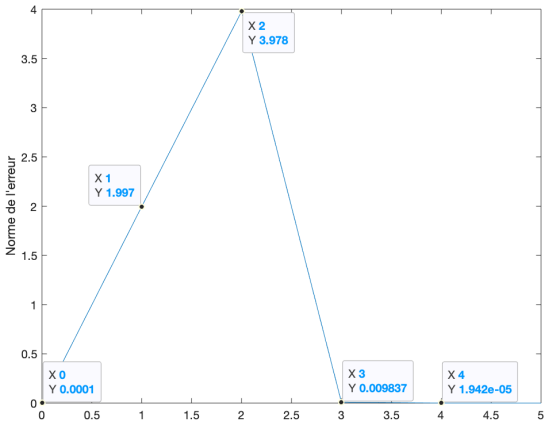
Soit $x^* = [1, 1]^T$, vérifions que c'est bien un minimum de f. Cela est montré dans la fonction MatLab dont une description est faite ici.
En résolvant $g(x) = 0$, on trouve que x^* est le seul point critique de f. Pour savoir si c'est un point selle ou un extremum on s'intéresse au signe de $H(x^*)$, or $H(1, 1) = 400 > 0$, donc x^* est bien un extremum. Pour savoir si c'est un maximum ou un minimum on s'intéresse au signe de $tr(H(x^*))$, or $tr(H(1, 1)) = 1002 > 0$, donc c'est bien un minimum local.

Question 3 et 4 :

On va partir du point $x_0 = [-1, -2]^T$ pour réaliser les 5 itérations de l'algorithme de Newton. Grâce à notre algorithme nous obtenons : $x^* = [-1.2206845009935043, -2]^T$.



Lignes de niveau de f entre [-1.5 ;2 ;-3 ;3] avec itérés.



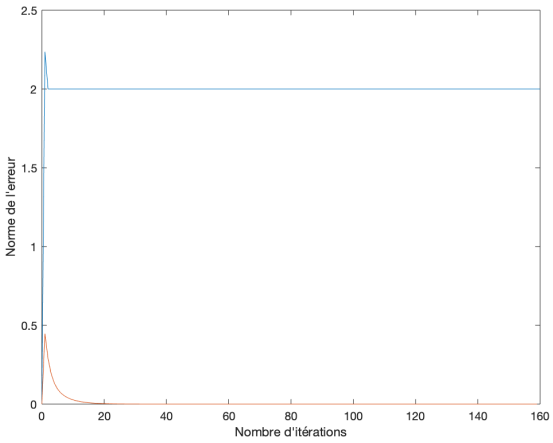
Erreur de convergence.

On peut observer qu'au bout de 5 itérations on se retrouve très vite dans un état où l'on va osciller entre des points très "proches" par rapport aux coordonnées x_2. Sur ce nombre d'itérations réduits on observe néanmoins que la convergence semble être quadratique.

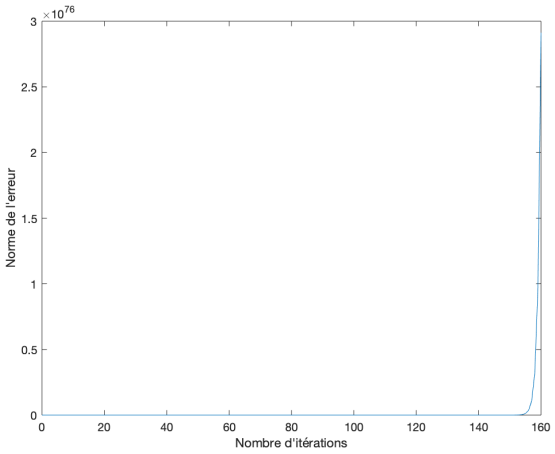
2 Exercice 7 : Méthode de gradient à pas optimal et méthode de Wolfe

Question 1 :

Soit $f(x) = x_1^2 + 2x_2^2$, nous allons réaliser des itérations de la méthode du gradient sur f avec $\alpha \in \{0.1,\ 0.5,\ 1\}$ et $x_0 = [-1, -1]$.



Convergence pour $\alpha = 0.1$ (orange) et $\alpha = 0.5$ (bleu)



Convergence pour $\alpha = 1$

On peut observer qu'il n'y a que pour $\alpha = 0.1$ qu'on converge vers une solution. Pour $\alpha = 0.5$, l'erreur de convergence tend vers une valeur constante de 2. Cela pourrait correspondre à un état "puit" dans lequel l'algorithme reste coincé. Pour $\alpha = 1$ c'est pire, en effet l'erreur reste constante au début mais explose vers la fin des calculs.

Question 2 :

Ici même pour un pas de l'ordre de 10^{-1} les calculs sont très longs et prennent trop de temps à s'effectuer.
Cela pose donc la question du choix d'un α optimal et nous conduit vers l'algorithme étudié dans la question 3.

Question 3 :

La méthode de Wolfe nous fournit un pas optimal noté t ici et un interval de confiance $T = [t_g,\ t_d]$. Après avoir fait tourner **Wolfe.m** on obtiet que $t = 0.00097656$ et $T = [0,\ 0.0020]$.
Maintenant que nous avons un pas optimal nous pouvons faire tourner l'algorithme du gradient avec $\alpha = t$. Après avoir attendu 18 heures sur Macbook Pro 2015 et en ayant choisit une erreur $\epsilon = 10^{-16}$ je n'ai toujours pas eu de résultat. Néanmoins j'ai atteint les 16 itérations en ce lapse de temps, ce qui dépasse de loin les 10 itérations avec les autres valeurs de α.
D'aucun pourrait remarquer que ce résultat est dû au temps attendu mais on peut aussi remarquer qu'en 19 minutes on atteint 13 itérations avec $\alpha = t$ alors qu'avec les autres nous n'étions qu'à 10.

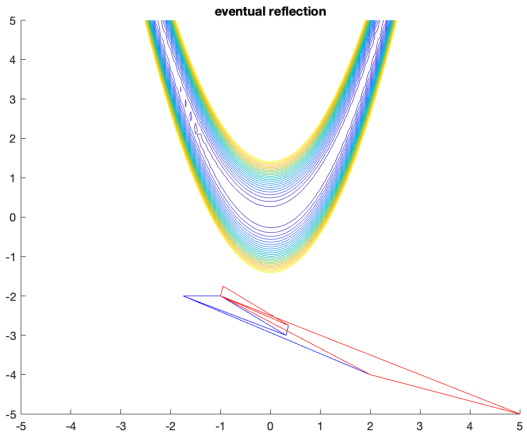
3 Exercice 8 :

Question 1 et 2 :

L'algorithme de Nelder Mead est une heuristique basée sur le simplexe. Le choix de ce dernier influe donc sur le résultat de l'algorithme. Pour améliorer nos chances de converger vers un minimum local il faut donc choisir des points linéairement indépendants.
Pour cela on peut former X_0 comme un ensemble de trois points, le premier correspond à notre point de départ x_0, ensuite on sonstruit $x_i = x_0 + h * e_i$ où e_i est le vecteur i de la base canonique et h choisit.
Par exemple pour $X_0 = [5-5;\ 2-4;\ -1-2]$ on obtient au bout de 250 itération le résultat $[1;\ 1]$ qui est bien le minimum de la fonction de Rosenbrock.

Question 3 :

Avec la classe nous avons utilisé la fonction présente dans Nelder_mead.m qui elle donne le résultat $[1.0006;\ 1.0012]$ sûrement à cause d'erreurs d'arrondis mais qui a l'avantage d'afficher l'évolution du polytope.



Évolution du polytope sur 4 itérations

La fonction fminsearch quant à elle renvoie bien $[1;\ 1]$ et ce bien plus rapidement.