

Ce rapport est séparé en deux parties. La première concerne le développement et le test d'un optimiseur numérique sur Matlab, la seconde partie concerne l'utilisation de cet outil sur un problème concret d'optimisation d'un réseau routier, proche du problème de Steiner.

Partie 1 : développement et test d'un optimiseur.

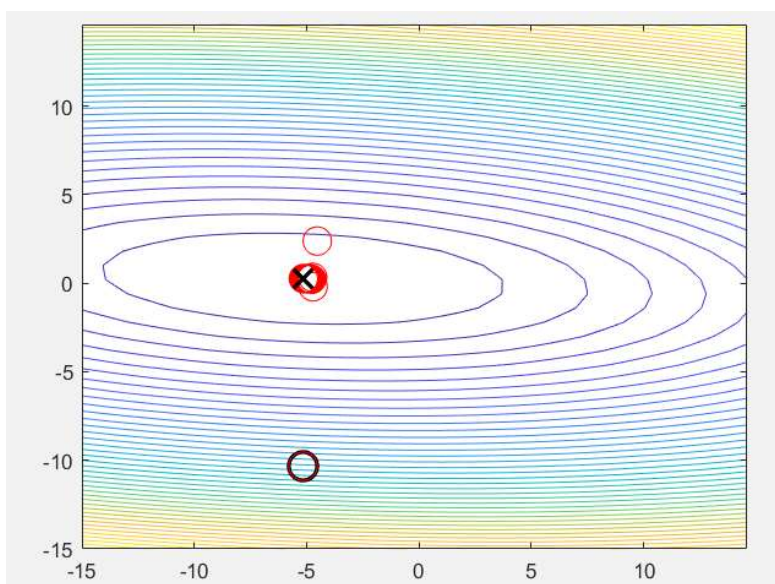
Les méthodes numériques d'optimisation envisagées ici sont principalement les méthodes de descente : les méthodes de gradient pour problèmes non contraints dans \mathbb{R}^n :

- Nelder et Mead (simplexe)
- Gradient à pas constant
- Gradient à pas optimal avec recherche linéaire approchée (Armijo)
- Gradient conjugué avec recherche linéaire approché
- Newton

1) Méthode du gradient à pas constant

On implémente la méthode du gradient à pas constant que l'on teste sur différentes fonctions objectifs proposées . Le calcul du gradient se fait par approximation au premier ordre des différentes dérivées partielles.

Il y a plusieurs possibilités pour la condition d'arrêt. On peut fixer une condition d'arrêt sur le module du gradient jusqu'à ce qu'elle soit en dessous d'un certain seuil (tend vers 0 puisque point critique), ou une condition à base de critère de Cauchy sur la distance entre deux termes x_k consécutifs. Les deux sont envisageables puisqu'aucune ne nécessite de calcul supplémentaire (on calcule dans tous les cas le gradient pour la direction de descente). Comparons les deux conditions, on teste les deux sur la fonction J1 avec un pas $\rho=0.1$ et un seuil $\epsilon=10^{-2}$.



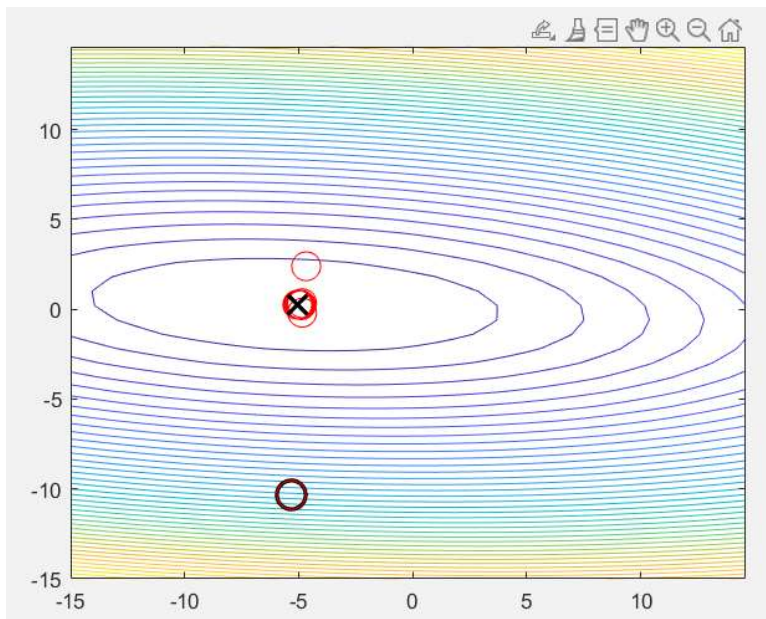
```
méthode utilisée: Pas Constant
nombre de pas d'optim si disponible: 40
nombre d'appels à la fonction objectif: 168
temps calcul CPU: 21.040969 secondes
Gradient J :
Grad =

    0.0094
   -0.0005

J(xoptimal) : -12.886552
coordonnées de x optimal
xopt =

   -5.1449
    0.2572
```

Figure n°1 : Test de la méthode du gradient à pas constant sur la fonction J1 avec condition d'arrêt sur le gradient.



```
méthode utilisée: Pas Constant
nombre de pas d'optim si disponible: 14
nombre d'appels à la fonction objectif: 64
temps calcul CPU: 8.739718 secondes
Gradient J :
Grad =

    0.0943
   -0.0051

J(xoptimal) : -12.881985
coordonnées de x optimal
xopt =

   -5.0571
    0.2524
```

Figure n°2 : Test de la méthode du gradient à pas constant sur la fonction J1 avec condition d'arrêt de Cauchy.

Sur notre exemple, la condition de Cauchy permet une convergence plus rapide. Comme on ne possède pas forcément de solution analytique pour comparer et que l'on n'a pas besoin d'une précision très importante sur les coordonnées, dans la suite, on maintient la seconde condition d'arrêt de Cauchy qui est la plus rapide.

Le réglage du pas joue un rôle décisif dans la convergence de cette méthode : si le pas est petit, on observe une décroissance lente et éventuellement convergence vers un minimum, et si le pas est grand on risque d'avoir divergence. Par exemple, pour notre fonction J1 le pas de 0.1 utilisé pour les exemples est à la limite du domaine de convergence.

On sait montrer que pour le cas où J est quadratique définie positive, le pas doit être inférieur à **deux fois l'inverse de son rayon spectrale** afin d'avoir convergence.

2) Méthode du gradient à pas optimal.

On implémente ensuite le gradient à pas optimal. La recherche du minimum se fait avec la règle d'Armijo.

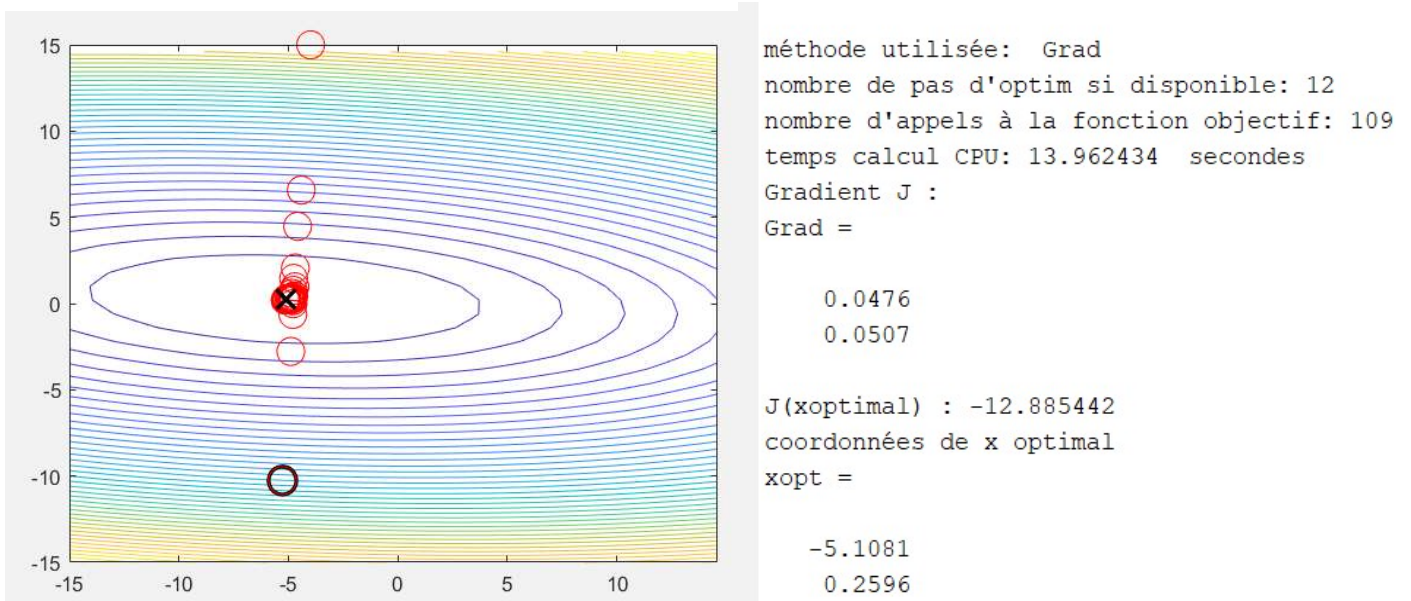


Figure n°3 : Test de la méthode du gradient à pas optimal avec Armijo sur la fonction J1 avec condition d'arrêt de Cauchy.

On diminue bien évidemment le nombre de pas d'optimisation et le nombre d'appel à la fonction objectif par rapport à la méthode à pas constant, néanmoins on remarque que cette méthode augmente le temps de calcul. Ceci est dû au fait qu'à chaque itération on effectue une pseudo-résolution de système linéaire avec Armijo. Néanmoins on remarque que ce n'est pas toujours le cas, suivant le x_0 de départ il se peut que le gradient à pas optimisé prenne moins de temps, car le fait de réduire le nombre de pas d'optimisation peut s'il est assez conséquent compenser la perte de temps induite par la résolution de l'équation linéaire à chaque itération. Cela dépend complètement des conditions initiales et de la distance entre les lignes de niveaux de la fonction objectif.

Comparons le temps de calcul et le nombre de pas d'optimisation des deux méthodes sur des tests identiques avec nos différentes fonctions objectifs afin d'avoir une idée plus précise des gains :

Fonction objectif	Pas constant avec rho adéquat	Pas optimal avec Armijo
FoncJ1	8.73s - rho=0.1 - 14 pas	13.96s – 12 pas
FoncJ2	18.57s - rho=0.2 - 36 pas	6.17s – 4 pas
FoncJ3	8.06s – rho=0.3 – 13 pas	13.66s– 6 pas
FoncJ4	4.04s – rho=0.3 – 7 pas	5.09s – 3 pas
FoncJ5	32.94s – rho =0.5 – 60 pas	9.94s – 4 pas

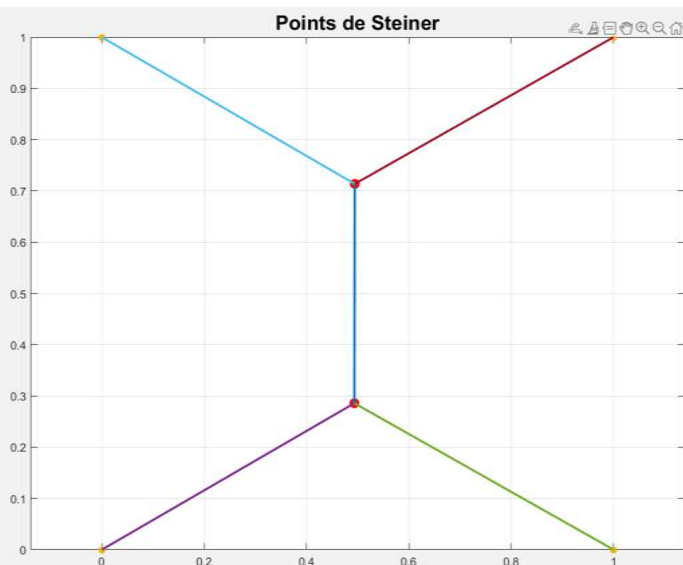
En réglant bien à l'avance notre rho, la méthode du gradient à pas constant parvient parfois (comme pour le test avec J3) à être plus rapide que celle avec pas optimal. Néanmoins le nombre de pas est toujours plus élevé, et lorsque l'on règle mal le rho (test avec J5 par exemple), l'écart se creuse entre les deux méthodes.

D'autres méthodes (gradient conjugué, Newton) ont été développées et sont visibles dans le fichier Matlab, que nous utiliserons dans l'étude du problème de Steiner.

Partie 2 : Application au problème de Steiner

Etant données n villes, on cherche à les relier toutes par un réseau autoroutier de longueur minimale. Les villes sont représentées par des points de \mathbb{R}^2 , et les autoroutes sont des segments de droites. Résoudre le problème de Steiner consiste à trouver le nombre et la position des points de jonctions entre les villes en minimisant la longueur du réseau. On supposera connus le nombre de points de Steiner, ainsi que les liaisons entre les villes. Il restera à calculer numériquement la position des points de jonction minimisant la longueur du réseau (la fonction objectif est la somme des longueurs des segments, l'inconnue est la position des points de Steiner).

On effectue une première résolution avec la méthode du gradient à pas optimal pour le cas $n=4$, voici la solution :



```
méthode utilisée: Grad
nombre de pas d'optim : 3
nombre d'appels à la fonction objectif: 46
temps calcul CPU: 17.164245 secondes
longueur minimale obtenue: 2.732103
coordonnées des points solution:
X =
0.4935
0.2858
0.4942
0.7137
```

Figure n°4 : Résolution du problème de Steiner à 4 villes avec la méthode du gradient à pas optimal.

On obtient la même solution avec la méthode du gradient à pas constant ou bien la méthode de Newton. On vérifie bien la validité de la solution en observant que x est bien un point critique (le gradient s'y annule) et qu'il est un minimum (hessienne positive en x).

Comparons la performance des méthodes sur ce problème pour $n=4$:

Nom de la méthode	Nombre de pas d'optimisations	Temps de calcul
Gradient à pas constant	5 pas – rho = 0.2	15.178426 secondes
Gradient à pas optimal	3 pas	17.164245 secondes
Newton	-	15.010849 secondes
Nelder et Mead	-	53.223083 secondes

La méthode de Nelder est la plus lente. La méthode de Newton étant d'ordre 2, elle converge plus vite. Néanmoins le calcul de la Hessienne rend au final le temps de calcul similaire à la méthode du gradient. On remarque que malgré le gain de 2 pas d'optimisation grâce à la règle d'Armijo, la résolution du système linéaire (dont on a par ailleurs augmenté la dimension par rapport aux tests de la partie précédente) à chaque étape fait au final perdre du temps à la méthode à pas optimisé par rapport à la méthode à pas constant.

Etudions les performances de ces algorithmes lorsque l'on augmente le nombre de villes :

Dimension	Nelder et Mead	Pas constant = 0.2	Pas optimal	Newton
n=5	147s	49s – 12 pas	61s – 11 pas	22s
n=5 bis	59s	22s – 9 pas	23s – 6 pas	9s
n=6	52s	Ne converge pas pour $\rho=0.2$ 47s pour $\rho = 0.1$	47s – 10 pas	19s
n=8	307s	Ne converge pas pour $\rho=0.2$ ou $\rho=0.1$	57s – 7 pas	24s

On remarque que l'il n'y a semble-t-il pas de relation entre la dimension du problème et le temps de résolution des algorithmes. La performance de ces derniers semblent dépendre de la structure du réseau et non de la dimension en elle-même, comme le montre le test n=5 bis en comparaison avec le test n°5.

La méthode du gradient à pas constant ne satisfait plus le critère de convergence lorsque n=6 et n=8 donc on pourrait éventuellement en déduire des conséquences sur la fonction objectif J.

Si J est **alpha-coercive et L-lipschitzienne**, on sait montrer que ρ doit être inférieur à $2 \cdot \alpha / L$ pour que la méthode converge. Si J est **quadratique définie positive**, ρ doit être inférieur à **2 fois l'inverse du rayon spectral**.

Steiner simplifié avec la non convergence de la méthode du gradient à pas constant lorsque la dimension du problème augmente sans pour autant diminuer le pas.