21 octobre 2022

Rapport Bureau d'Etude 1
SXS314

Giuliano Vinci, Paul Calot

Question 1

Supposons que d_k minimise :

$$\langle \nabla_x L(x_k, \lambda_k), d\rangle + \frac{1}{2}\left\langle \nabla_x^2 L(x_k, \lambda_k)d, d\right\rangle$$

Puisque $\nabla_x L(x_k, \lambda_k) = \nabla_x J(x_k) + D_c^\top(x_k)\lambda_k$, nous pouvons réécrire la quantité précédente comme :

$$\langle \nabla_x J(x_k), d\rangle + \left\langle D_c^\top(x_k)\lambda_k, d\right\rangle + \frac{1}{2}\left\langle \nabla_x^2 L(x_k, \lambda_k)d, d\right\rangle$$

Or en supposant que $Dc(x_k)d = -c(x_k)$:

$$\left\langle D_c^\top(x_k)\lambda_k, d\right\rangle = \lambda_k^\top D_c(x_k)d = -\lambda_k^\top c(x_k)$$

Si x_k est une solution du problème quadratique qui approxime le problème réel, alors les contraintes sont vérifiées, ce qui devrait être le cas en théorie par construction. Cependant, avec les erreurs numériques, nous pouvons imaginer que les contraintes ne soient pas vérifiées. Au quel cas, nous pouvons simplement remarquer que $\langle \lambda_k, c(x_k)\rangle$ ne dépend pas de d et donc ne joue pas dans le problème de minimisation.

Ainsi, dans tous les cas, nous avons d_k qui minimise la quantité transformée précédente et qui est donc solution du problème quadratique (QP_k).

Question 2

On note :

$$L_{QP_k} : \begin{array}{ccc} (d,\lambda) & \mapsto & \langle \nabla_x J(x_k), d\rangle + \frac{1}{2}\left\langle \nabla_x^2 L(x_k, \lambda_k)d, d\right\rangle + \lambda^\top(D_c(x_k)d + c(x_k)) \\ R^n \times R^p & \to & R \end{array}$$

D'après l'énoncé, λ_{k+1} est le multiplicateur optimal de Lagrange associé au point solution d_k pour le Lagrangien L_{QP_k}. C'est-à-dire : $\nabla L_{QP_k}(d_k, \lambda_{k+1}) = 0_{R^{n+p}}$.

Or, on a pour $(d, \lambda) \in R^n \times R^p$:

$$\nabla L_{QP_k}(d, \lambda) = \begin{pmatrix} \nabla_x J(x_k) + \nabla_x^2 L(x_k, \lambda_k)d + D_c(x_k)^\top \lambda \\ D_c(x_k)d + c(x_k) \end{pmatrix}$$

D'où :

$$0_{R^{n+p}} = \nabla L_{QP_k}(d_k, \lambda_{k+1}) \Leftrightarrow \begin{pmatrix} \nabla_x^2 L(x_k, \lambda_k) & D_c(x_k)^\top \\ D_c(x_k) & 0_{M_{p\times p}} \end{pmatrix} \begin{pmatrix} d_k \\ \lambda_{k+1} \end{pmatrix} = -\begin{pmatrix} \nabla_x J(x_k) \\ c(x_k) \end{pmatrix}$$

Ce qui donne le résultat.

Question 3

Notons A la matrice de $M_{n+p\times n+p}$ tel que :

$$A = \begin{pmatrix} \nabla_x^2 L(x_k, \lambda_k) & D_c(x_k)^\top \\ D_c(x_k) & 0_{M_{p\times p}} \end{pmatrix}$$

Notons $B = \begin{pmatrix} I_n & -(\nabla_x^2 L(x_k, \lambda_k))^{-1} D_c(x_k)^\top \\ 0_{M_{p\times n}} & I_p \end{pmatrix} \in M_{2n}$, ce qui est une expression valide car $\nabla_x^2 L(x_k, \lambda_k)$ est définie positive et donc inversible.

Nous avons alors :

$$AB = \begin{pmatrix} \nabla_x^2 L(x_k, \lambda_k) & 0_{M_{p\times p}} \\ D_c(x_k) & -D_c(x_k)(\nabla_x^2 L(x_k, \lambda_k))^{-1} D_c(x_k)^\top \end{pmatrix}$$

Puis, on peut calculer le déterminant par bloc :

$$Det(AB) = Det((AB)^\top) = (-1)^p \times Det(\nabla_x^2 L(x_k, \lambda_k)) \times Det(D_c(x_k)(\nabla_x^2 L(x_k, \lambda_k))^{-1} D_c(x_k)^\top)$$

Or, puisque $\nabla_x^2 L(x_k, \lambda_k)$ est définie positive, elle s'écrit D^2, d'où :

$$Det(D_c(x_k)(\nabla_x^2 L(x_k, \lambda_k))^{-1} D_c(x_k)^\top) = Det(D_c(x_k)D) \times Det(DD_c(x_k)^\top)$$

Car pour des matrices carrés : $Det(AB) = Det(A)Det(B)$.

De plus, $rg(AB) \geq rg(A) + rg(B) - n$ pour A avec n colonnes et B avec n lignes (inégalité de Sylvester).

D'où $rg(DD_c(x_k)^\top) \geq rg(D) + rg(D_c(x_k)^\top) - n = p$ car D est inversible et le rang de $D_c(x_k)^\top$ est p par hypothèse. De même, on obtient celle sur le premier facteur en se rappelant que le rang d'une matrice A est égal à celui de sa transposé.

Finalement, en utilisant $Det(B) = 1$, nous avons :

$$Det(A) = (-1)^p \times Det(\nabla_x^2 L(x_k, \lambda_k)) \times Det(D_c(x_k)(\nabla_x^2 L(x_k, \lambda_k))^{-1} D_c(x_k)^\top) \neq 0$$

D'après ce qui précède.

D'où A est inversible et par conséquent le système (LS_k) admet une unique solution.

Question 4

Il est possible d'ajouter en critères d'arrêt les deux suivants :

1. Lorsque la solution x_k stagne (c'est-à-dire que la différence entre deux solutions successives est suffisamment petite), alors nous pouvons considérer que nous avons convergé.
2. Lorsque le gradient du Lagrangien s'approche suffisamment de 0, alors nous pouvons également considérer que nous avons convergé.
3. Une dernière condition qui peut être nécessaire en pratique et que nous avons fini par utiliser a été d'ajouter un critère sur la vérification de la contrainte, c'est-à-dire qu'elle doit être vérifiée à une tolérance près. Nous constations en effet que l'algorithme s'arrêtait sur l'une des deux précédentes conditions tout en ayant une contrainte qui n'était pas vérifiée ou pas "suffisamment".

En pratique, nous observons que vérifier ces trois critères en même temps donne de meilleurs résultats (il faut que les trois soient vérifiés pour que l'algorithme s'arrête).

L'algorithme 1 donne le pseudo-code pour la solution SQP.

Algorithm 1 Algorithme SQP

Initialisation : x_0, λ_0 donnés, $\epsilon > 0$ précision demandée.
Étape courante : x_k, λ_k connus
Calcul de $J(x_k)$, $c(x_k)$, $\nabla_x J(x_k)$, $D_c(x_k)$, $\nabla_x^2 J(x_k)$ et $\nabla_x^2 c(x_k)$ à partir des fonctions exactes.
Calcul du Hessien du Lagrangien : $\nabla_x^2 L(x_k, \lambda_k) \leftarrow \nabla_x^2 J(x_k) + \lambda_k^\top \nabla_x^2 c(x_k)$.
Construction de la matrice $M \leftarrow \begin{pmatrix} \nabla_x^2 L(x_k, \lambda_k) & D_c(x_k)^\top \\ D_c(x_k) & 0_{M_{p\times p}} \end{pmatrix}$
Construction de la matrice $B \leftarrow -\begin{pmatrix} \nabla_x J(x_k) \\ c(x_k) \end{pmatrix}$
Résolution du système $MX = B$
Obtention de $(d_k, \lambda_{k+1}) \leftarrow X^\top$
Mise à jour de $x_{k+1} = x_k + d_k$
Test des critères d'arrêt.

Question 6

L'algorithme 2 donne le pseudo-code pour la solution SQP BFGS.

Algorithm 2 Algorithme SQP BFGS

Initialisation : x_0, $\lambda_0 = 0$, $H_0 = I_n$ donnés, $\epsilon > 0$ précision demandée.
Calcul de $J(x_0)$, $c(x_0)$, $\nabla_x J(x_0)$, $D_c(x_0)$ à partir des fonctions exactes.
Calcul du gradient du Lagrangien : $\nabla_x L(x_0, \lambda_0) \leftarrow \nabla_x J(x_0) + \lambda_0^\top D_c(x_0)$.
Construction de la matrice $M \leftarrow \begin{pmatrix} H_0 & D_c(x_0)^\top \\ D_c(x_0) & 0_{M_{p\times p}} \end{pmatrix}$
Construction de la matrice $B \leftarrow -\begin{pmatrix} \nabla_x J(x_0) \\ c(x_0) \end{pmatrix}$
Résolution du système $MX = B$
Obtention de $(d_0, \lambda_1) \leftarrow X^\top$
Mise à jour de $x_1 = x_0 + d_0$
Étape courante : x_k, x_{k-1}, λ_k, $\nabla_x L(x_{k-1}, \lambda_{k-1})$, H_{k-1} connus
Calcul de $J(x_k)$, $c(x_k)$, $\nabla_x J(x_k)$, $D_c(x_k)$ à partir des fonctions exactes.
Calcul du gradient du Lagrangien : $\nabla_x L(x_k, \lambda_k) \leftarrow \nabla_x J(x_k) + \lambda_k^\top D_c(x_k)$.
Calcul de $s_{k-1} \leftarrow x_k - x_{k-1}$
Calcul de $y_{k-1} \leftarrow \nabla_x L(x_k, \lambda_k) - \nabla_x L(x_{k-1}, \lambda_{k-1})$
If $y_{k-1} \cdot s_{k-1} > 0$
$H_k \leftarrow H_{k-1} + \frac{y_{k-1} y_{k-1}^\top}{y_{k-1}^\top s_{k-1}} - \frac{H_{k-1} s_{k-1} s_{k-1}^\top H_{k-1}}{s_{k-1}^\top H_{k-1} s_{k-1}}$
Else
$H_k \leftarrow H_{k-1}$
Endif
Construction de la matrice $M \leftarrow \begin{pmatrix} H_k & D_c(x_k)^\top \\ D_c(x_k) & 0_{M_{p\times p}} \end{pmatrix}$
Construction de la matrice $B \leftarrow -\begin{pmatrix} \nabla_x J(x_k) \\ c(x_k) \end{pmatrix}$
Résolution du système $MX = B$
Obtention de $(d_k, \lambda_{k+1}) \leftarrow X^\top$
Mise à jour de $x_{k+1} = x_k + d_k$
Test des critères d'arrêt.

Question 7

En pratique, nous avons utilisé un schéma de différences finies centré d'ordre 8, qui peut être trouvé dans le module associé à la méthode BFGS FD. La racine carré de l'erreur machine a été utilisée comme perturbation pour le calcule du gradient par différences finies.

Question 8

Les résultats sont présentés sur les figures 1 (problème 1), 2 (problème 2) et 3 (problème 3). Chaque figure possède quatre sous-figures, une par point de départ. Sur chaque sous-figure sont représentés la fonction J à minimiser, la contrainte c ainsi que les différents algorithmes SQP, BFGS, BFGS avec différences finies.

Les algorithmes peuvent être évalués sur des *benchmarks* simples, ici les fonctions du cercle et de Rosenbrock. On peut alors donner une évaluation qualitative de :

— Complexité
— Stabilité
— Précision
— Robustesse

Ce qui est effectuée dans la suite.

Comparaison des complexités

Il est possible de comparer les résultats des algorithmes également en ce qui concerne leur complexité, qui est l'un des principaux facteurs de choix d'un algorithme par rapport à un autre. Il est évident que l'algorithme SQP peut devenir, en général, le plus complexe des trois à exécuter, en raison de la présence de diverses dérivées de premier et second ordre. Raisonnablement, pour les problèmes d'ingénierie réels et multivariés, on peut soutenir que l'algorithme est inutilisable dans sa forme naïve, ce qui a donné naissance à l'algorithme BFGS. Calculer non seulement le hessian, mais aussi le gradient avec des formules approximatives est une simplification intéressante dans les problèmes généraux à plusieurs variables, où les fonctions de gradient ne sont pas faciles à évaluer analytiquement. Dans ce cas, la méthode BFGS aux différences finies se présente. Néanmoins, comme on peut en juger par le résultat des simulations, les simplifications sont inutiles et le résultat est définitivement de moins bonne qualité que pour l'algorithme original.

Comparaison de stabilités et robustesses

Pour évaluer la stabilité des algorithmes, on pourrait simplement essayer de perturber le point de départ et voir si l'algorithme converge toujours vers la même solution, et s'il converge en général. Cette approche est suggérée par la notion générale de stabilité en physique et dans l'étude numérique des EDOs. Comme on peut le voir sur les graphiques, les algorithmes ne convergent pas en général vers la même solution.

— L'algorithme SQP, comme on le voit, converge vers des solutions similaires quels que soient les points de départ pour les problèmes 1 et 3. Dans le problème 2, au contraire, l'utilisation de [10,10] comme point de départ produit une solution différente des autres points de départ (figure 1).
— L'algorithme BFGS converge toujours vers la même solution dans le problème 1, même si avec une certaine "indécision" par moments. Dans les problèmes 2 et 3, il converge vers la même solution, mais avec de fortes déviations et des erreurs de parcours.
— l'algorithme BFGS - FD converge, dans n'importe quel problème, vers la même solution lorsque le point de départ change, sauf si c'est X0 = [2,5].

Comparaison des précisions

Il n'est pas forcément aisé de définir la précision pour des algorithmes qui s'arrêtent lorsqu'un certain critère est vérifié. Cela dit, pour une tolérance donnée, si l'algorithme termine avec l'un des critères de convergence (c'est-à-dire pas parce qu'on a atteint le maximum d'itérations ou d'appels à une fonction), alors le nombre d'itérations peut permettre une certaine définition de la précision. Cela revient en quelque sorte à prendre les choses à l'envers.

En considérant cette définition, on a les remarques précédentes :

1. Dans le premier problème, le nombre d'itérations est majoritairement le même pour tous les problèmes.
2. Dans le deuxième problème, le nombre d'itérations de la deuxième méthode, à savoir BFGS, est le plus bas, ce qui indiquerait une meilleure précision, dans le sens qu'on s'est donné, et pour

les points de départs utilisés. En effet, cela dépend également de ceux-ci, par conséquent une conclusion générale sur le sujet serait malvenue.

3. Pour le troisième problème, on remarque que la méthode BFGS ne converge pas tout le temps (100 itérations atteints avant). La première méthode (SQP) semble converger tout le temps sur le même point, tandis que la dernière méthode ne donne pas tout le temps les mêmes solutions.

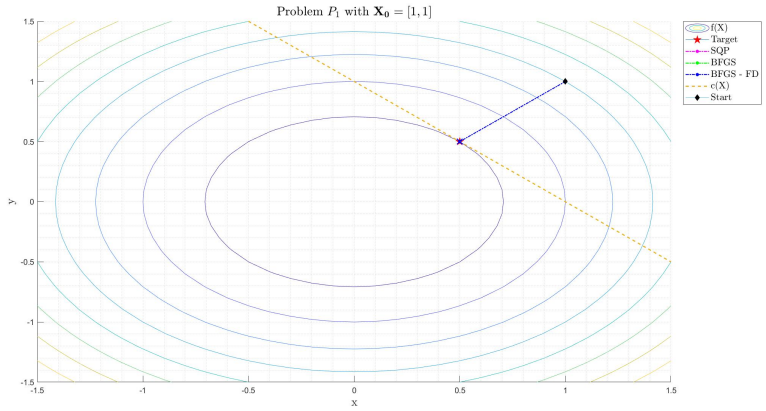
Tout cela est à reconsidérer du point de vue des critères d'arrêt utilisés, à savoir :

1. $\|d_k\| < 10^{-5}$
2. $\|\nabla L_k(x_k, \lambda_k)\| < 10^{-7}$
3. $\|c(x_k)\| < 10^{-6}$

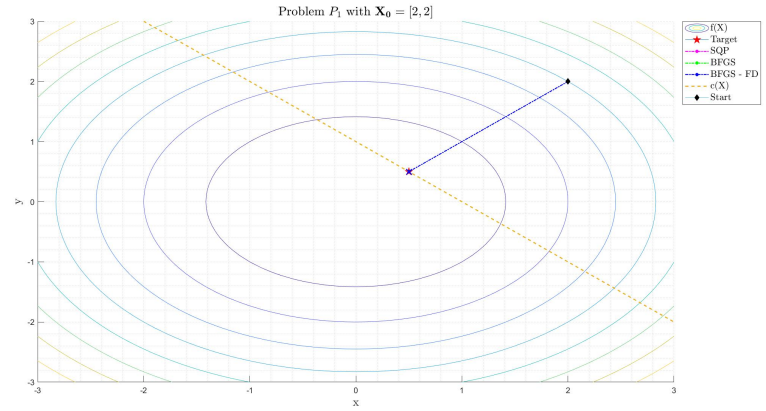
D'autres critères donneront probabZlement d'autres résultats.

Amélioration des méthodes

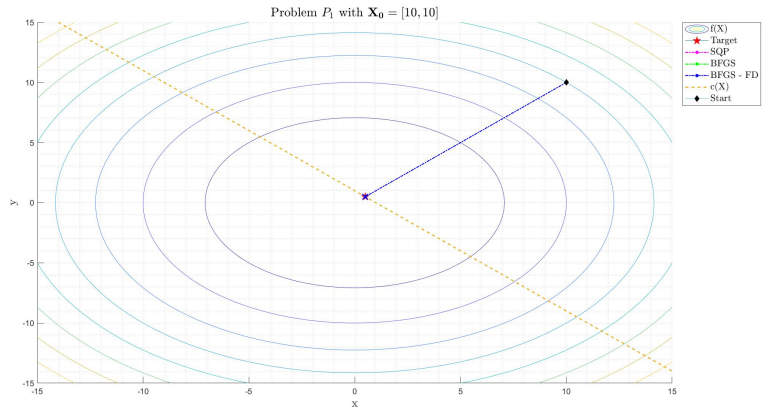
1. Le pas des algorithmes est ici fixe, mais on pourrait envisager des méthodes où l'utilisateur le fournit, possiblement avec une stratégie, ou pour lesquelles une recherche du pas optimal est réalisée. En effet, comme on peut le voir sur les figures 2, nous avons des exemples, notamment pour la méthode BFGS, soit d'explosion ou en tout cas d'éloignement de la solution optimale d'une étape à l'autre de l'algorithme. La recherche du pas optimal peut s'effectuer par exemple avec l'algorithme de *backtracking*.
2. Le calcul de la matrice Hessiennne peut également se faire par différences finies de deuxième ordre (et pas par mise à jour comme c'est le cas pour BFGS).
3. De meilleurs critères d'arrêt pourraient être trouvés, afin d'anticiper certains cas pathologiques et pour rendre plus robuste la méthode. Il a par exemple été constaté qu'un simple critère sur la stagnation de x_k pouvait engendrer des arrêts prématurés. L'ajout d'une certaine patience peut résoudre cela : la condition d'arrêt doit être vérifiée sur p étapes consécutives pour que l'algorithme d'optimisation se termine. Un critère d'arrêt sur la qualité de vérification des contraintes pourraient également être ajouté à côté de ceux déjà existants (c'est-à-dire qu'il faudrait vérifier deux conditions en même temps pour pouvoir terminer l'algorithme), ce qui a été finalement réalisé et constitue ce qui est présenté précédemment. Un risque de fonctionner avec plusieurs critères de convergence, c'est-à-dire de restreindre davantage encore l'ensemble des points de convergence possibles, est de ne jamais converger (au quel cas l'algorithme se termine parce qu'on a atteint le nombre maximal d'itérations), ou de mettre beaucoup de temps à converger, ce qui peut poser des problèmes pour certaines applications industrielles.



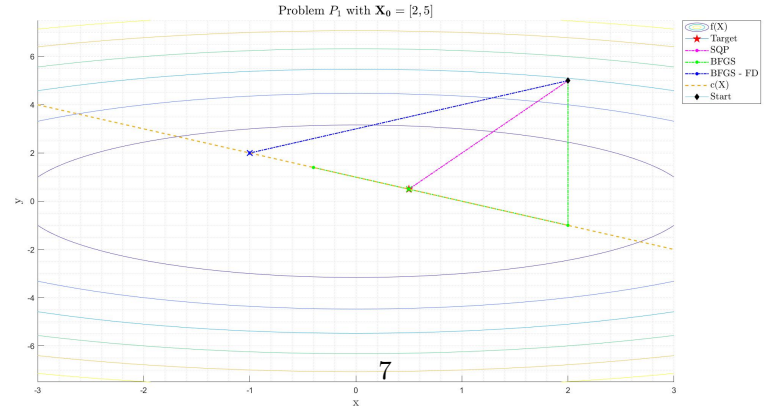
(a) $X_0 = [1, 1]$



(b) $X_0 = [2, 2]$

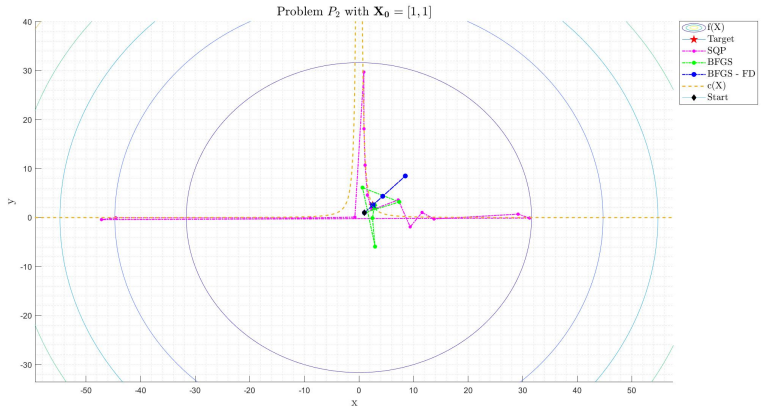


(c) $X_0 = [10, 10]$

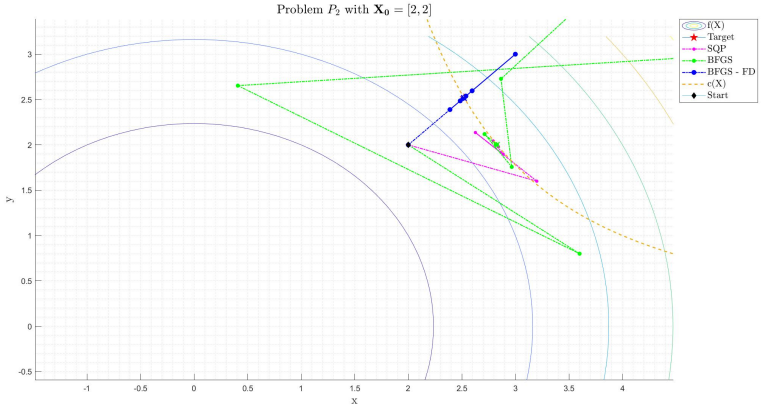


(d) $X_0 = [2, 5]$

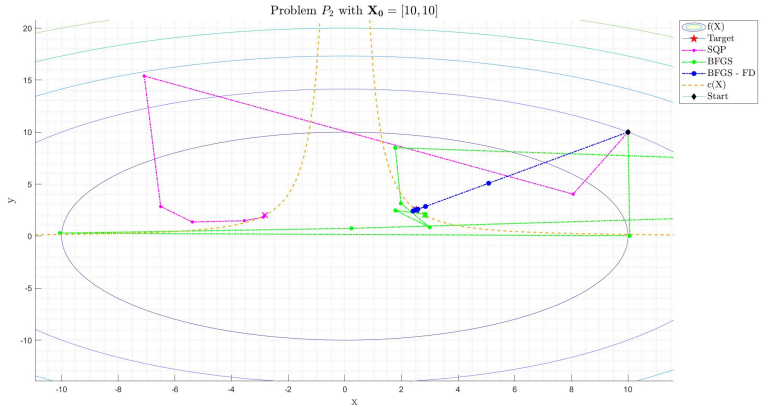
FIGURE 1 – Solutions du Problème 1



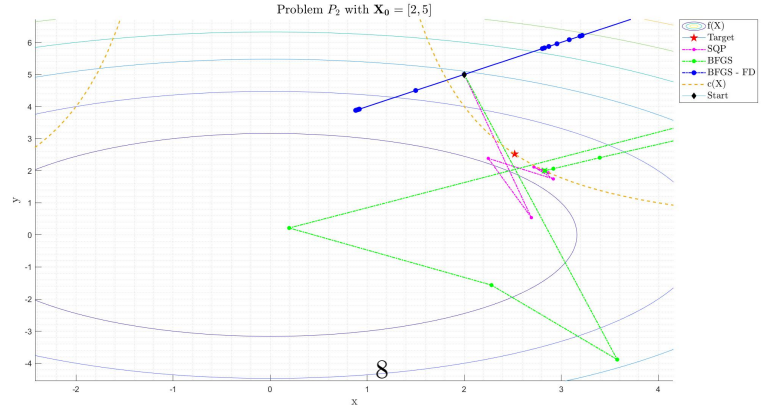
(a) $X_0 = [1, 1]$



(b) $X_0 = [2, 2]$

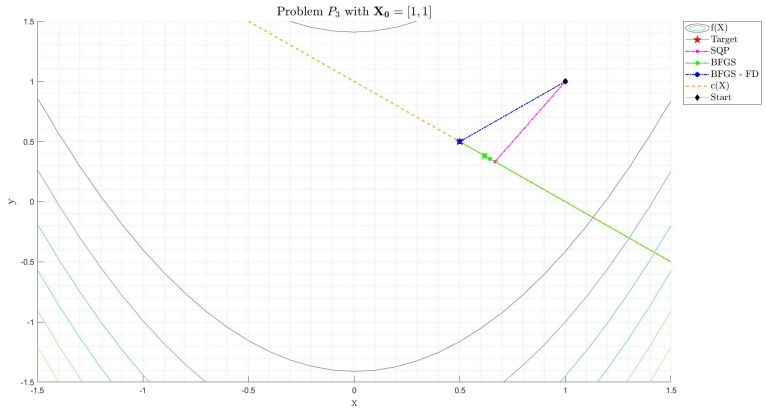


(c) $X_0 = [10, 10]$

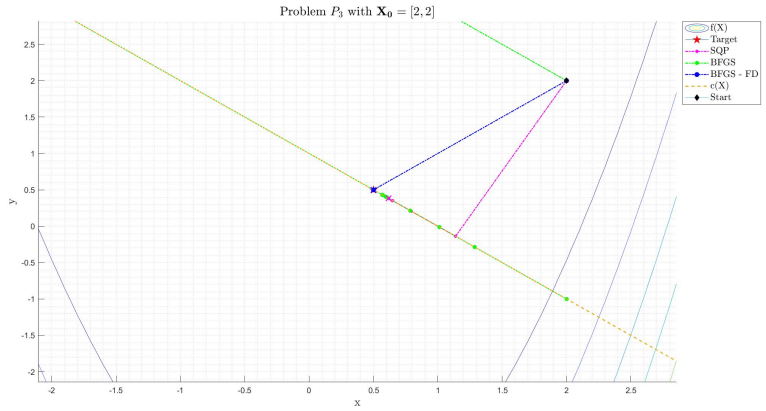


(d) $X_0 = [2, 5]$

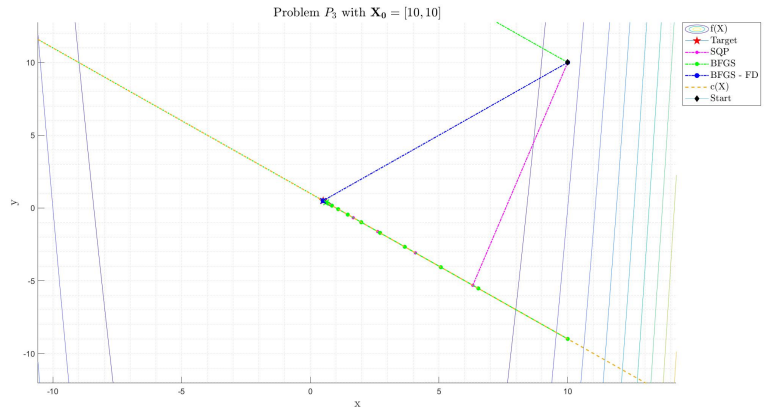
FIGURE 2 – Solutions du Problème 2



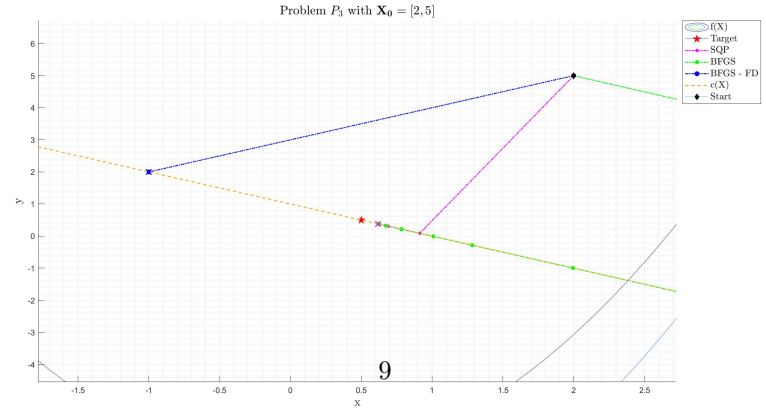
(a) $X_0 = [1, 1]$



(b) $X_0 = [2, 2]$



(c) $X_0 = [10, 10]$



(d) $X_0 = [2, 5]$

FIGURE 3 – Solutions du Problème 3