



Compte rendu projet OPT202

Équilibre d'une chaîne articulée

Réalisé par :

Louay El Melki

Mohamed Issa

Fatma Hachicha

Encadré par :

Mme MARESCAUX Eugénie

2^{ème} Année Techniques avancées

Année universitaire : 2020/2021

1 Introduction

Le problème à résoudre consiste de trouver la position d'équilibre statique d'une chaîne formée de n_b barres rigides, contenue dans un plan vertical (x, y) et devant se trouver au-dessus d'un plancher donné par une courbe convexe linéaire par morceaux. Dans ce TP, on prendra en compte des contraintes d'inégalité. Cette position est obtenue en minimisant l'énergie potentielle de la chaîne sous des contraintes appropriées.

Pour ce faire, on modifiera dans un premier temps le simulateur qui calculera les variables nécessaires à la résolution du problème. Puis, on se propose d'améliorer l'optimiseur en remplaçant la méthode de Newton par un algorithme connu sous le nom *d'optimisation quadratique successive* OQS. On testera par la suite 3 cas tout en analysant les résultats. On terminera ce TP par une autre méthode *La méthode de Quasi Newton* qu'on détaillera par la suite.

2 Présentation du problème

On veut à présent prendre en compte la présence d'un plancher convexe linéaire par morceaux que l'on exprime comme l'enveloppe supérieure d'un nombre fini p de fonctions affines. La chaîne, qui doit rester au-dessus de ce plancher, ne pourra donc pas, en général, prendre sa position d'équilibre précédente. On suppose le plancher infiniment glissant. Il n'y a donc pas lieu d'introduire un modèle décrivant le contact entre la chaîne et le plancher. Ceci se traduit par le fait que l'on cherche toujours la position d'énergie potentielle minimale, la présence du plancher se traduisant simplement par l'ajout de contraintes d'inégalité portant sur la position des nœuds.

Le plancher est supposé décrit dans le plan (x, y) , au moyen d'une fonction affine $\varphi : \mathbb{R} \rightarrow \mathbb{R}^p$ définie par

$$\varphi(x) = r + xs$$

où les vecteurs $r \in \mathbb{R}^p$ et $s \in \mathbb{R}^p$ sont donnés mais pourront varier en dimension et en valeur d'un cas-test à l'autre. Tous les points de la chaîne doivent se trouver dans le convexe

$$C = \{(x, y) \in \mathbb{R} \times \mathbb{R} : ye \geq \varphi(x)\}$$

où, sous forme matricielle, $e := (1 \ 1 \cdots 1)^\top \in \mathbb{R}^p$. On a représenté à la figure 1 la position

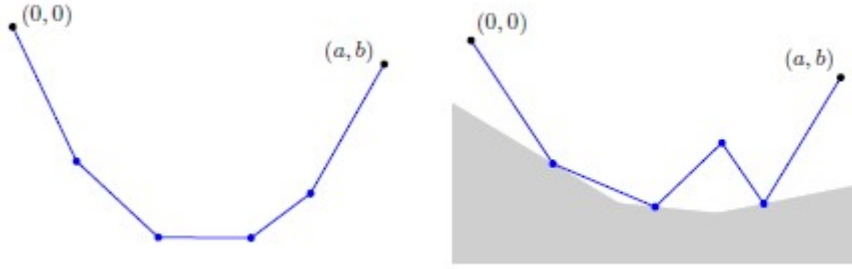


FIGURE 1 – Positions d'équilibre de la chaîne, sans et avec plancher

d'équilibre d'une chaîne, sans (à gauche) et avec (à droite) cette contrainte (dans ce cas, $p = 3$). On supposera que les points de fixation $(0,0)$ et (a,b) se trouvent dans le domaine C , ce qui revient à supposer que les vecteurs r et s (des données) vérifient les inégalités vectorielles

$$r \leq 0 \text{ et } r + as \leq be$$

Comme la chaîne est affine entre ses nœuds et que le plancher est convexe, elle sera entièrement contenue dans C si tous ses nœuds s'y trouvent, c'est-à-dire si pour tout $i \in [1 : n_n]$ et tout $j \in [1 : p]$, on a

$$c_{n_b+(j-1)n_n+i}(x, y) \equiv r_j + x_i s_j - y_i \leq 0$$

Le problème à résoudre est donc de la forme

$$(P_2) \begin{cases} \min e(x, y) \\ c_i(x, y) = 0, & i \in [1 : n_b] \\ c_i(x, y) \leq 0, & i \in [n_b + 1 : n_b + pn_n] \end{cases}$$

où les contraintes d'égalité (les n_b premières) concernent la longueur des barres et les contraintes d'inégalité (les suivantes) expriment la présence du plancher.

3 Le simulateur

```
function [e,ce,ci,g,ae,ai,h1,indic] = chs(indic,xy,lme,lmi)
```

En entrée :

indic : pilote le comportement du simulateur :

= 1: **chs** fait un tracé de la chaîne

= 2: **chs** calcule **e**, **ce** et **ci**

= 4: **chs** calcule **e**, **ce**, **ci**, **g**, **ae** et **ai**

= 5: **chs** calcule **h1**

xy : Le vecteur-colonne à optimiser contenant les abscisses des noeuds et leurs ordonnées $xy = (x_1, \dots, x_{n_n}, y_1, \dots, y_{n_n})$.

lme : vecteur-colonne de dimension n_b contenant les multiplicateurs de Lagrange pour les contraintes d'égalité, $\lambda = \{\lambda_i\}_{i=1}^{n_b}$.

lmi : vecteur-colonne de dimension n_n contenant les multiplicateurs de Lagrange pour les contraintes d'inégalité, $\lambda_I = \{\lambda_i\}_{i=n_b+1}^{n_b+n_n}$.

En sortie :

e : Énergie potentielle, donnée par

$$e = \sum_{i=1}^{n_b} L_i \frac{y_i + y_{i+1}}{2} = \frac{1}{2} \left(L_1, \dots, L_{n_b} \right) \left(y_1 + y_2, \dots, y_{n_b} + y_{n_b+1} \right)^T \quad (1)$$

ce : valeur en **xy** des contraintes sur la longueur des barres.

ci : contient la valeur des $p n_n$ contraintes d'inégalité.

$$C_I = \begin{pmatrix} R_{n_n} + x_1 S_{n_n} - Y_1 \\ R_{n_n} + x_2 S_{n_n} - Y_2 \\ \vdots \\ R_{n_n} + x_{n_n} S_{n_n} - Y_{n_n} \end{pmatrix}$$

$$\text{avec } \begin{cases} R_{n_n} = (r_1, r_2, \dots, r_{n_n})^T \\ S_{n_n} = (s_1, s_2, \dots, s_{n_n})^T \\ Y_i = (y_i, y_i, \dots, y_i)^T \in \mathbb{M}_{1,n_n}, \quad \forall i \in [1, n_n] \end{cases}$$

g : Gradient de l'énergie potentielle.

ae : jacobienne des contraintes d'égalité en **xy**.

ai : jacobienne des contraintes d'inégalité en **xy**.

$$A_I = C'_I = \begin{pmatrix} s_1 \mathbb{1}_{n_n} & -\mathbb{1}_{n_n} \\ s_2 \mathbb{1}_{n_n} & -\mathbb{1}_{n_n} \\ \vdots & \vdots \\ s_p \mathbb{1}_{n_n} & -\mathbb{1}_{n_n} \end{pmatrix}$$

h1 : hessien du lagrangien en **xy**.

$$\nabla_{xx}^2 \ell(\cdot, \lambda) = \nabla^2 e(x) + \sum_{i \in E} \lambda_i \nabla^2 c_i(x) + \sum_{i \in I} \lambda_i \nabla^2 c_i(x)$$

Comme les contraintes sont affines, on obtient :

$$\nabla_{xx}^2 \ell(x, \lambda) = \begin{pmatrix} \lambda_1 + \lambda_2 & -\lambda_2 & 0 & \dots & \dots & 0 \\ -\lambda_2 & \lambda_2 + \lambda_3 & -\lambda_3 & 0 & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -\lambda_{n_n} \\ 0 & \dots & \dots & 0 & -\lambda_{n_n} & \lambda_{n_n} + \lambda_{n_b} \end{pmatrix}$$

où les λ_i sont dans `lm`.

indic : décrit le résultat de la simulation : Il est égal à 1 si l'un des paramètres est incorrect sinon 0.

Il y a deux variables globales en plus :

- **R** : r , vecteur correspondant au plancher $\varphi(x) = r + sx$
- **S** : s , idem.

4 Optimisation quadratique successive

Afin d'améliorer l'optimiseur, on va utiliser l'algorithme d'optimisation quadratique successive. Ceci revient à calculer à chaque itération k une solution primale-duale $(d_k, \lambda_k^{\text{pq}})$ du problème suivant :

$$\begin{cases} \min_{d \in \mathbb{R}^n} \nabla f(x_k)^\top d + \frac{1}{2} d^\top M_k d \\ c_E(x_k) + c'_E(x_k) d = 0 \\ c_I(x_k) + c'_I(x_k) d \leq 0 \end{cases}$$

M_k étant définie comme une approximation du hessien du lagrangien par la factorisation de Cholesky.

Le nouvel itéré est alors

$$x_{k+1} = x_k + d_k \quad \text{et} \quad \lambda_{k+1} = \lambda_k^{\text{pq}}$$

La factorisation de Cholesky de H_k fonctionnera que si $H_k > 0$ puisque $H_k = L_k L_k^\top$. La technique consiste alors à modifier H_k au cours de sa factorisation de Cholesky et d'en déduire une matrice E_k (un écart, une erreur) telle que

$$H_k + E_k = L_k D_k L_k^\top,$$

où E_k est diagonale semi-définie positive, L_k est triangulaire inférieure et D_k est diagonale définie positive.

On prend alors $M_k := L_k D_k L_k^\top$ dans le problème quadratique osculateur.

4.1 Tests

4.1.1 Cas-test 4.a

On prend $L = [0.7 \ 0.5 \ 0.3 \ 0.2 \ 0.5]$

La position initiale des noeuds : $xy = [0.2, \ 0.4, \ 0.6, \ 0.8, \ 1, \ 1.5, \ 1.5, \ 1.3]$

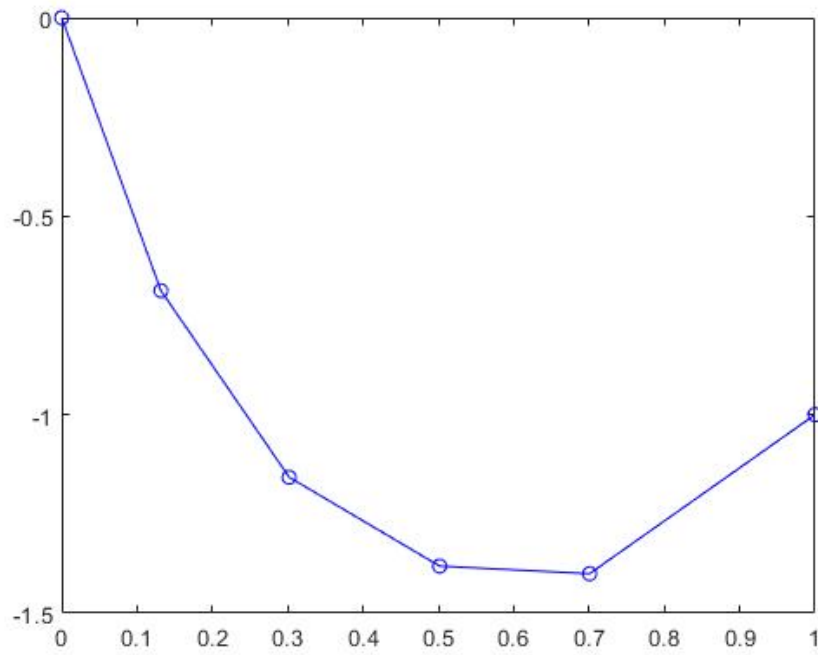


FIGURE 2 – la chaîne à l'équilibre dans le cas test 4-a

4.1.2 Cas-test 4.b

On prend $L=[0.2 \ 0.2 \ 0.2 \ 0.3 \ 0.3 \ 0.5 \ 0.2 \ 0.2 \ 0.3 \ 0.1]$

La position initiale des noeuds :

$xy=[0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5 \ 0.6 \ 0.7 \ 0.8 \ 0.9 \ -0.5 \ -0.9 \ -1.2 \ -1.4 \ -1.5 \ -1.4 \ -1.2$
 $-0.9 \ -0.5]$

plancher :

$R=-0.25$

$S=-0.5$

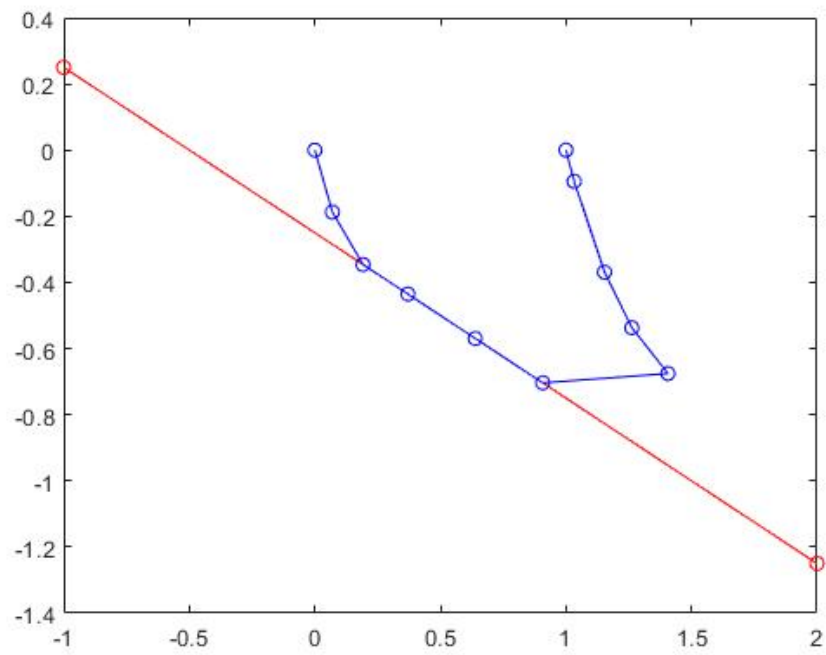


FIGURE 3 – la chaîne à l'équilibre dans le cas test 4-b

4.1.3 Cas-test 4.c

On reprend le même cas 4.b avec :

$$R = [-0.25; -0.5]$$

$$S = [-0.5; 0]$$

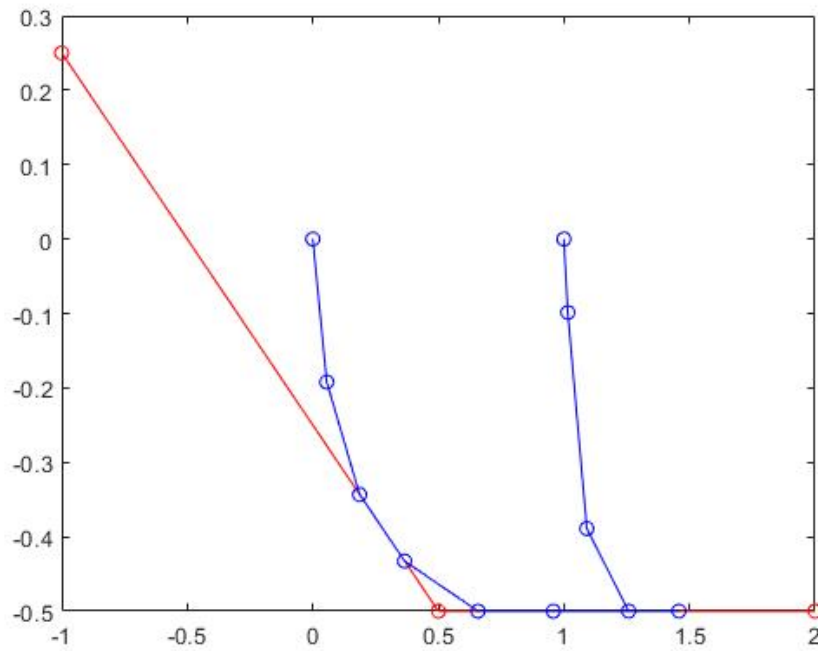


FIGURE 4 – la chaîne à l'équilibre dans le cas test 4-c

4.2 Questions

4.2.1 Question 1 : Cas où $M = I$

Lors de l'étape de modification de l'optimiseur on prend pour M_k une approximation définie positive du hessien du lagrangien afin d'assurer que M_k soit définie positive et la résolution numérique du problème soit plus facile. Alors on peut se demander pourquoi on n'initialise pas directement la matrice M à l'identité. Théoriquement, on se permet de prendre $M = I$ et on trouve une solution. Mais, en ce qui concerne la pratique, pour trouver la solution, il faut un nombre qui n'est pas atteignable d'itérations pour trouver la solution. Il n'est donc pas intéressant de choisir $M = I$.

4.2.2 Question 2 : multiplicateur initial

Dans cette question on est amené à trouver un multiplicateur meilleur que $\lambda_0 = 0$ de telle sorte que si l'utilisateur donne au solveur un point initial primal x_0 qui est une solution, l'algorithme trouve le multiplicateur optimal sans faire d'itération. On a x_0 solution donc minimum local et les contraintes sont affines donc qualifiées en x_0 ainsi on conclut que le multiplicateur optimal serait la solution duale du système (KKT) :

(KKT)

$$\begin{cases} \nabla f(x_0) + c'(x_0)^T \lambda_0 = 0 \\ c_i(x_0) = 0, \quad i \in E \\ c_j(x_0) \leq 0, \quad j \in I \\ (\lambda_0)_j \geq 0, \quad j \in I \\ \sum_{j \in I} (\lambda_0)_j c_j(x_0) = 0 \end{cases}$$

Avec x_0 la solution primale donc pour calculer λ_{opt} sans faire d'itération on peut calculer la solution du problème avec contrainte d'égalité suivant :

$$\begin{cases} \min_{\lambda_0 \geq 0} \|\nabla f(x_0) + c'(x_0)^T \lambda_0\|^2 \\ \sum_{j \in I} (\lambda_0)_j c_j(x_0) = 0 \end{cases}$$

Qui revient à résoudre un système linéaire en λ donc trouve le multiplicateur optimal sans itération.

4.2.3 Question 3 : critère qu'un point limite soit minimum local

Afin de trouver un critère qui permet de vérifier si un point limite soit un minimum local on utilise les conditions d'optimalité du second ordre pour les problèmes avec contraintes d'égalité qui sont :

$$\begin{cases} \nabla f(x_*) + c'(x_*)^T \lambda_* = 0 \\ \forall d \in \mathcal{N}(c'_E(x)) \quad d^T \nabla_{xx} \ell(x_*, \lambda_*) d > 0 \end{cases}$$

Donc si on pose $e = (e_1, \dots, e_{\dim(\mathcal{N}(c'_E(x)))})$ une base de $\mathcal{N}(c'_E(x))$ il suffit de trouver un vecteur de base tel que $e_i^T \nabla_{xx} \ell(x_*, \lambda_*) e_i \leq 0$ pour dire que le point limite x_* n'est pas minimum local.

4.2.4 Question 4 : comportement différent du solveur

On se permet de dire que le comportement différent du solveur est dû à la différence des algorithmes utilisés dans chaque cas pour résoudre le problème car dans le cas du solveur sans contrainte d'inégalité on a utilisé un algorithme de Newton et dans le cas du solveur avec contrainte d'égalité et d'inégalité on a utilisé un algorithme SQP.

4.2.5 Question 5 : convergence quadratique

On observe dans tous les cas la convergence quadratique car si on observe les valeurs de $\frac{\|x_{k+1} - x_{opt}\|}{\|x_k - x_{opt}\|^2}$ on remarque qu'il existe un rang k_{opt} à partir duquel $\frac{\|x_{k+1} - x_{opt}\|}{\|x_k - x_{opt}\|^2} \leq C$ avec C constante qui dépend du cas test.

5 Méthode de Quasi-Newton

Dans cette partie, on va introduire une autre méthode moins "coûteuse" à savoir la méthode de *quasi Newton*. Elle consiste à approximer le hessien par une matrice qu'on note dans la suite \mathbf{M} en utilisant la formule de BFGS.

La version quasi-newtonienne de l'OQS consiste à générer une suite primale-duale $\{(x_k, \lambda_k)\}$ et une suite de matrices symétriques définies positives $\{M_k\}$.

À l'étape k , on calcule la solution primale-duale $(d_k, \lambda_k^{\text{PQ}})$ du problème quadratique osculateur

$$\begin{cases} \min_{d \in \mathbb{R}^n} \nabla f(x_k)^\top d + \frac{1}{2} d^\top M_k d \\ c_E(x_k) + c'_E(x_k) d = 0 \\ c_I(x_k) + c'_I(x_k) d \leq 0 \end{cases} \quad (2)$$

où M_k est une approximation du hessien du lagrangien ℓ $L_k := L(x_k, \lambda_k) := \nabla_{xx}^2 \ell(x_k, \lambda_k)$, $c'_E(x_k)$ et $c'_I(x_k)$ sont respectivement les jacobiniennes des contraintes d'égalité c_E et c_I .

On pose $x_{k+1} := x_k + \alpha_k d_k$, $\lambda_{k+1} := \lambda_k + \alpha_k (\lambda_k^{\text{PQ}} - \lambda_k)$ ($\alpha_k \in]0, 1]$ est le pas calculé par recherche linéaire). M_k est mise à jour par la formule de BFGS qui s'écrit :

$$M_{k+1} = M_k - \frac{M_k \delta_k \delta_k^\top M_k}{\delta_k^\top M_k \delta_k} + \frac{\gamma_k \gamma_k^\top}{\gamma_k^\top \delta_k}$$

On détermine les vecteurs γ_k et δ_k de manière à ce que M_{k+1} reste définie positive et se rapproche de L_{k+1} , ce qui, dans certains cas, peut être contradictoire. Pour cela, on prend pour δ_k le déplacement en x :

$$\delta_k = x_{k+1} - x_k$$

Le vecteur γ_k devrait idéalement être la variation du gradient du lagrangien

$$\gamma_k^\ell = \nabla \ell(x_{k+1}, \lambda_{k+1}) - \nabla \ell(x_k, \lambda_{k+1})$$

Mais, pour conserver la définie positivité de M_k , on doit avoir $\gamma_k^\top \delta_k > 0$, ce qui n'est pas garanti avec $\gamma_k = \gamma_k^\ell$. Si bien que l'on utilisera la correction de Powell (1936-2015) qui consiste à prendre

$$\gamma_k = (1 - \theta) M_k \delta_k + \theta \gamma_k^\ell$$

où θ est pris maximal dans $]0, 1]$ de manière à avoir $\gamma_k^\top \delta_k \geq 0.2 \delta_k^\top M_k \delta_k$. On trouve

$$\theta = \begin{cases} 0.8 \frac{\delta_k^\top M_k \delta_k}{\delta_k^\top M_k \delta_k - (\gamma_k^\ell)^\top \delta_k} & \text{si } (\gamma_k^\ell)^\top \delta_k < 0.2 \delta_k^\top M_k \delta_k \\ 1 & \text{sinon.} \end{cases}$$

Passons maintenant au choix de la matrice initiale. On prendra $M_1 = I$ (matrice identité) à la première itération par manque d'informations. Mais, après le calcul de x_2 et avant le calcul de M_2 , on modifie la valeur de M_1 en $\eta_1 I$, où η_1 a une valeur reflétant l'échelle du problème :

$$\eta_1 := \frac{\|\gamma_1\|_2^2}{\gamma_1^\top \delta_1}$$

Il faut en effet attendre que la première itération soit terminée pour évaluer cette grandeur. Puis on calcule M_k , pour $k \geq 2$, par la formule de BFGS.

5.1 Tests

5.2 Cas-test 5.a

On prend $L=[0.5 \ 0.3 \ 0.4 \ 1.2 \ 0.3 \ 0.3]$

Le deuxième point de fixation de la chaîne : $A=B=0$

La position initiale des noeuds :

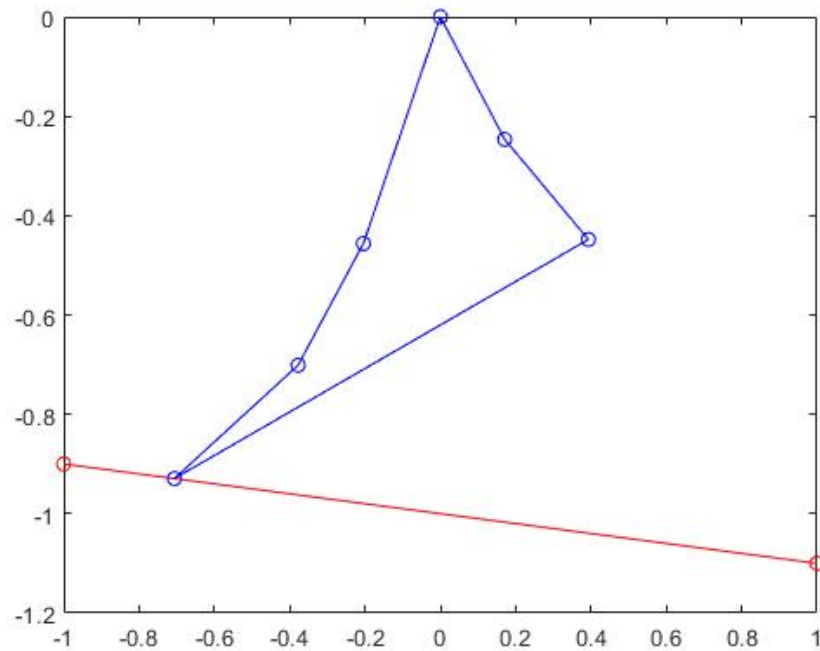


FIGURE 5 – Position d'équilibre de la chaîne au cas test 5-a

$xy=[0.2 \ 0.5 \ 0.8 \ 1.0 \ 1.2 \ -0.4 \ -0.6 \ -0.4 \ -0.2 \ 0.]$

Plancher : $R=[-1]'$; $S=[-0.1]'$;

5.2.1 Cas-test 5.b

On prend $L = [3 \ 2.5 \ 2.5]'$;

Le deuxième point de fixation de la chaîne : $(A=0, B=-4)$

La position initiale des noeuds :

$xy = [-2 \ 0 \ 1 \ -2]'$;

plancher :

$R= [-6 \ -10]'$

$S=[-2 \ 100]'$

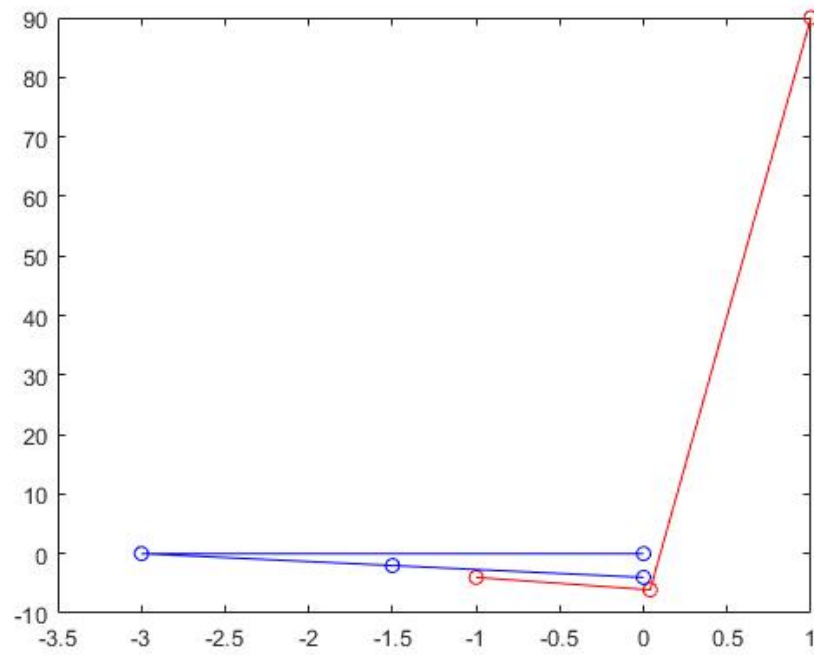


FIGURE 6 – Position d'équilibre de la chaîne au cas test 5-b

5.2.2 Cas-test 5.c

$L = [0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.5 \ 0.4 \ 0.3 \ 0.1]'$;

Deuxième point de fixation de la chaîne : $A=B=0$

On a choisi comme position initiale des noeuds :

$xy = [0.1 \ 0.15 \ 0.2 \ 0.25 \ 0.2 \ 0.15 \ 0.1 \ 0.3 \ 0.5 \ 0.7 \ 0.5 \ 0.4 \ 0.3 \ 0.2]'$;

Plancher :

$R = [-1.0; -0.2; -1.0];$

$S = [-7.0; 0.0; 7.0];$

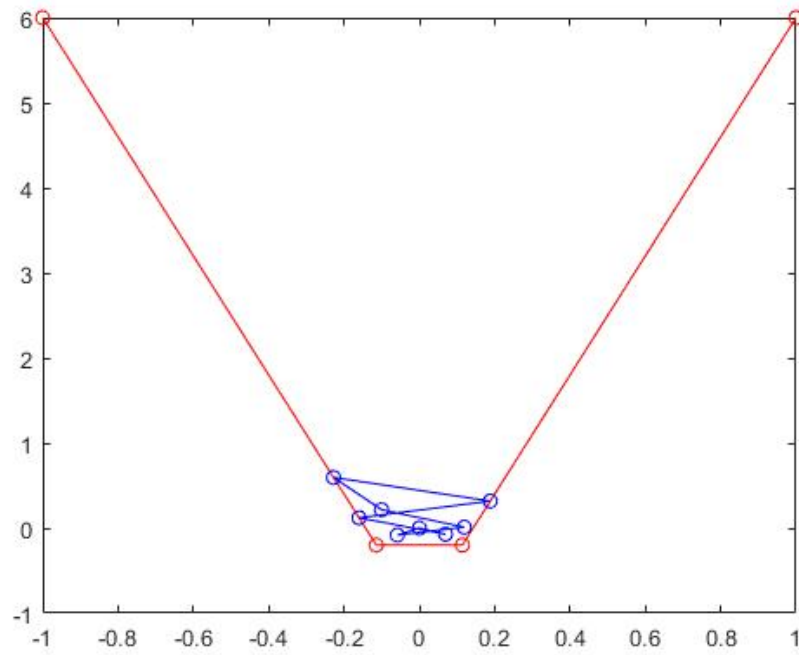


FIGURE 7 – Position d'équilibre de la chaîne au cas test 5-c

5.2.3 Cas-test 5.d

On prend :

$L = [1.5 \ 1.7 \ 0.8 \ 2.2 \ 1]'$;

Deuxième point de fixation de la chaîne : $(A=1, B=-1)$

On a choisi comme position initiale des noeuds :

$xy = [0.2 \ 4 \ 0.6 \ 0.8 \ \dots \ -1 \ -1.5 \ -1.5 \ -1.3]'$;

Plancher :

$R = [-0.8; -1.5]$;

$S = [-1.5; -0.5]$;

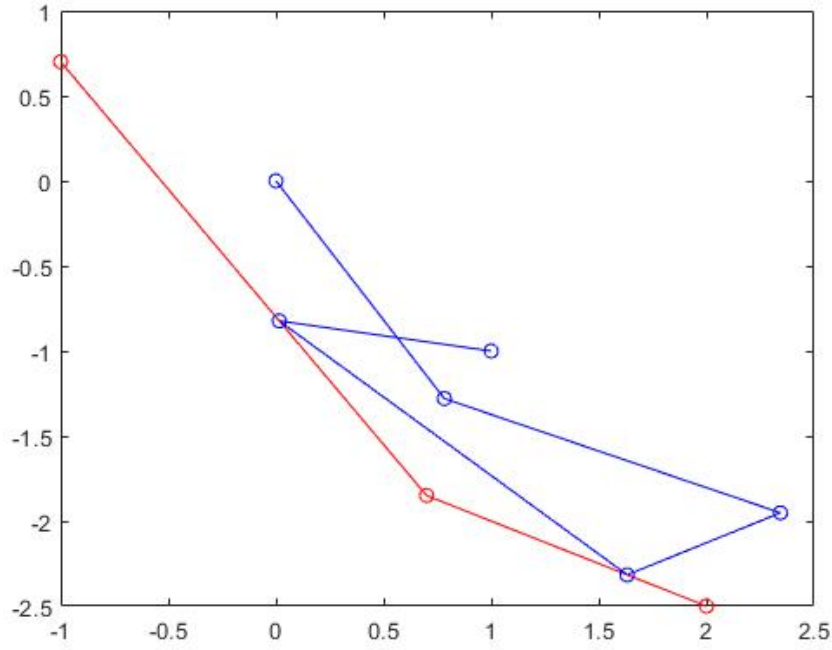


FIGURE 8 – Position d'équilibre de la chaîne au cas test 5-d

5.3 Questions

5.3.1 Question 1

On suppose que la suite (x_k, λ_k) converge vers une solution $z_* = (x_*, \lambda_*)$.
La vitesse de convergence de cette suite d'ordre q est donnée par

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|^q}$$

Comme on ne connaît pas la solution, on étudie le quotient $\frac{\|x_{k+1} - x_k\|}{\|x_k - x_{k-1}\|^q}$.

5.3.2 Question 2

Comme on a $M_k > 0$, l'autre problème qui peut se poser pour empêcher le calcul de la solution du PQO est la linéarisation des contraintes du problème initial qui pourrait conduire probablement au fait que l'ensemble admissible serait vide car les contraintes ne sont plus cohérentes donc on ne peut plus calculer de solution.

5.3.3 Question 3

Lors de l'exécution de l'algorithme par la méthode de Quasi-Newton sur un cas test, l'un des itérés peut se trouver dans un voisinage d'un maximum local strict donc d'après les CN2 le hessien du lagrangien doit être défini négatif donc ça serait contradictoire d'approximer le hessien par une matrice défini positif.

5.3.4 Question 4

Un certain nombre d'affirmations sont faites autour du choix de γ_k dans la formule *BFGS* :

- Le vecteur γ_k pris dans la formule devrait être idéalement γ_k^ℓ :
on veut approximer $L := \nabla_{xx} l(x, \lambda)$ par M donc il serait logique que la variation du gradient du lagrangien intervient dans la formule de mise à jour de M ainsi le vecteur γ_k pris dans la formule devrait être idéalement γ_k^ℓ
- On doit avoir $\gamma_k^T \delta_k > 0$ pour que M_{k+1} soit définie positive :
soit $x \in \mathbb{R}^n$. on a :

$$x^T M_{k+1} x = x^T M_k x + \frac{(\gamma_k^T x)^2}{\gamma_k^T \delta_k} - \frac{(\delta_k^T M_k x)^2}{\delta_k^T M_k \delta_k} = \frac{\delta_k^T M_k \delta_k x^T M_k x - (\delta_k^T M_k x)^2}{\delta_k^T M_k \delta_k} + \frac{(\gamma_k^T x)^2}{\gamma_k^T \delta_k}.$$

si on définit le produit scalaire $\langle x, y \rangle$ comme $x^T M_k y$ l'équation ci-dessus devient :

$$x^T M_{k+1} x = \frac{\langle \delta_k, \delta_k \rangle \langle x, x \rangle - \langle \delta_k, x \rangle^2}{\langle \delta_k, \delta_k \rangle} + \frac{(\gamma_k^T x)^2}{\gamma_k^T \delta_k}$$

Le premier terme est positif par l'inégalité de Cauchy-Schwartz. En ce qui concerne le deuxième terme, comme le pas de l'étape est optimale on a :

$$g_{k+1}^T d_k = 0$$

ce qui implique

$$\gamma_k^T \delta_k = \rho_k (g_{k+1} - g_k)^T d_k = \rho_k g_k^T M_k g_k > 0$$

Donc $x^T M_{k+1} x \geq 0$. Les deux termes étant positifs, la somme est zéro seulement si les deux termes sont 0 . Cela implique que $x = \lambda \delta_k$ avec $\lambda \neq 0$. Dans ce cas, le deuxième terme ne peut pas être égal à zéro car $\gamma_k^T x = \lambda \gamma_k^T \delta_k$. Alors on aura $M_{k+1} > 0$.

donc il faut avoir $\gamma_k^T \delta_k > 0$ pour que M_{k+1} soit définie positive

- Le choix $\gamma_k = \gamma_k^\ell$ ne garantit pas la définie positivité de M_{k+1} :
En effet, d'après le point précédant, pour assurer la définie positivité de la matrice M_k , on doit avoir $\delta_k^T \gamma_k > 0$. Donc,

$$(1-) \delta_k^T M_k \delta_k + \delta_k^T \gamma_k^\ell > 0$$

.

Puisque $M_k > 0$, le premier terme est toujours positif. $\delta_k^T \gamma_k^\ell$ peut être négatif. D'où, le choix $\gamma_k = \gamma_k^\ell$ ne garantit pas la définie positivité de M_k .

5.3.5 Question 5

On choisit \mathbf{M}_1 de façon à approcher au mieux $\nabla^2 f(x_1)$. Or, \mathbf{M}_{k+1} vérifie l'équation Quasi-Newton $y_k = \mathbf{M}_{k+1} s_k$ où $s_k := x_{k+1} - x_k$ et $y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$.

D'autre part, y_1 peut s'écrire $y_1 = \frac{\|y_1\|_2^2}{y_1^T s_1} s_1$

Donc, si on pose $\mathbf{M}_1 = \mu I = \frac{\|y_1\|_2^2}{y_1^T s_1} I$, elle vérifie bien l'équation $y_1 = \mathbf{M}_1 s_1$

6 Conclusion

On a résout notre problème avec différentes méthodes : La méthode de Newton et de Quasi-Newton. On ne trouve pas de solution dans tous les cas tests malgré la simplicité du problème physique mais on peut contourner ce problème en trouvant des approximations des solutions avec des méthodes visant à améliorer ces approximations, et ceci en utilisant des méthodes de recherche linéaire, d'optimisation quadratique successive et la formule de BFGS. Certes, chaque méthode a des avantages et des inconvénients notamment en ce qui concerne la vitesse de convergence de ces algorithmes.