

Équilibre d'une chaîne articulée - TP5

Aristote Koen
Thomas Chatrefou

Février 2020

Prise en compte de contraintes d'inégalité

On prend à présent en compte la présence d'un plancher convexe linéaire défini dans le plan (x,y) au moyen de la fonction affine :

$$\phi(x) = r + xs \quad r \in \mathbb{R}^p \quad s \in \mathbb{R}^p$$

Tous les points doivent se trouver dans l'ensemble convexe :

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

Avec $e := (1, 1, \dots, 1)^T \in \mathbb{R}^p$

La chaîne étant affine entre ses noeuds, elle sera entièrement contenue dans C si tous les noeuds s'y trouvent, c'est à dire pour tout $i \in \llbracket 1, n_n \rrbracket$ et tout $j \in \llbracket 1, p \rrbracket$ on a :

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

On obtient donc un problème d'optimisation sous la forme :

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

On obtient alors un problème avec n_b contraintes d'égalité et $n_n p$ contraintes d'inégalité.

On modifie donc le simulateur décrit auparavant de sorte à lui donner un argument `lmi` en plus : un vecteur colonne de taille n_n contenant les multiplicateurs de lagrange associés aux contraintes d'inégalité.

Le simulateur retourne alors deux éléments de sortie en plus :

1. `ci` : un vecteur c_I de taille $n_n p$ contenant la valeur des contraintes d'inégalité
2. `ai` : la jacobienne des contraintes d'inégalité, une matrice c'_I de $n_n p$ lignes et $2n_n$ colonnes.

Ainsi on a :

$$c_I(x) = \begin{bmatrix} r_1 \\ \vdots \\ r_p \\ r_1 \\ \vdots \\ r_p \end{bmatrix} + \begin{bmatrix} x_1 s_1 \\ \vdots \\ x_1 s_p \\ \vdots \\ x_{n_n} s_1 \\ \vdots \\ x_{n_n} s_p \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_1 \\ \vdots \\ y_{n_n} \\ \vdots \\ y_{n_n} \end{bmatrix} \in \mathbb{R}^{pn_n}$$

dont on peut écrire la Jacobienne de la façon suivante :

$$c'_I = \begin{pmatrix} M_1 & A_1 \\ M_2 & A_2 \\ \vdots & \vdots \\ M_{n_n} & A_{n_n} \end{pmatrix}$$

Avec

$$M_1 = \begin{pmatrix} s_1 & 0 & \dots & 0 \\ s_2 & 0 & \dots & 0 \\ \vdots & & & \\ s_p & 0 & \dots & 0 \end{pmatrix} \quad M_k = \begin{pmatrix} 0 & \dots & s_1 & 0 & \dots & 0 \\ 0 & \dots & s_2 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & s_p & 0 & \dots & 0 \end{pmatrix} \quad M_{n_n} = \begin{pmatrix} 0 & \dots & 0 & s_1 \\ 0 & \dots & 0 & s_2 \\ \vdots & & & \vdots \\ 0 & \dots & 0 & s_p \end{pmatrix}$$

M_k des matrices à p lignes et n_n colonnes et $k \in \llbracket 1, n_n \rrbracket$ la k -ième colonne de la matrice

$$A_1 = \begin{pmatrix} -1 & 0 & \dots & 0 \\ -1 & 0 & \dots & 0 \\ \vdots & & & \\ -1 & 0 & \dots & 0 \end{pmatrix} \quad A_k = \begin{pmatrix} 0 & \dots & -1 & 0 & \dots & 0 \\ 0 & \dots & -1 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & -1 & 0 & \dots & 0 \end{pmatrix} \quad A_{n_n} = \begin{pmatrix} 0 & \dots & 0 & -1 \\ 0 & \dots & 0 & -1 \\ \vdots & & & \vdots \\ 0 & \dots & 0 & -1 \end{pmatrix}$$

A_k des matrices à p lignes et n_n colonnes et $k \in \llbracket 1, n_n \rrbracket$ la k -ième colonne de la matrice
Les contraintes étant affines, il n'y a pas besoin de modifier le hessien de lagrangien.

Le problème étant maintenant un problème d'optimisation avec contraintes d'égalité et d'inégalité de la forme

$$(P_{EI}) \quad \begin{cases} \min f(x) \\ c_E(x) = 0 \\ c_I(x) \leq 0 \end{cases}$$

l'optimiseur a également besoin de modifications : on se propose donc d'améliorer le précédent en remplaçant la méthode de Newton par un algorithme d'optimisation quadratique successive. Une itération de cet algorithme remplace la résolution du système linéaire de la méthode de Newton par la résolution d'un problème d'optimisation quadratique, on calcule alors une solution primale-duale $(d_k, \lambda_k^{\text{PQ}})$ du problème quadratique osculateur

$$\begin{cases} \min_{d \in \mathbb{R}^n} \nabla f(x_k)^T d + \frac{1}{2} d^T M_k d \\ c_E(x_k) + c'_E(x_k) d = 0 \\ c_I(x_k) + c'_I(x_k) d \leq 0 \end{cases}$$

avec M_k le hessien du lagrangien ou une approximation de celui-ci. Le nouvel itéré est alors

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

Dans la suite, nous prenons pour M_k une approximation définie positive de $H_k := \nabla_{xx}^2 \ell(x_k, \lambda_k)$. Les deux méthodes les plus couramment utilisées pour cela sont la factorisation de Cholesky modifiée et les techniques de quasi-Newton.

Factorisation de Cholesky modifiée

Questions théoriques

(4.1) Supposons que M_k est l'identité, alors le problème quadratique osculateur se réécrit

$$\begin{cases} \min_{d \in \mathbb{R}^n} \nabla f(x_k)^T d + \frac{1}{2} \|d\|_2^2 \\ c_E(x_k) + c'_E(x_k)d = 0 \\ c_I(x_k) + c'_I(x_k)d \leq 0 \end{cases}$$

Le terme en $\|d\|$ va simplement pénaliser la direction de descente sur sa norme, on peut s'intéresser au terme $\nabla f(x_k)^T d$. D'après Cauchy-Schwarz :

$$|\nabla f(x_k)^T d| \leq \|\nabla f(x_k)\| \|d\|$$

donc

$$\nabla f(x_k)^T d \geq -\|\nabla f(x_k)\| \|d\|$$

avec égalité si d est colinéaire à $-\nabla f(x_k)$. Donc à chaque itération, la résolution de ce problème va donner une solution proche d'une simple descente de gradient qui, bien que convergente à l'ordre 1, est excessivement lente pour un problème d'optimisation sous contraintes. En pratique, nous n'arrivons même pas à observer la convergence de l'algorithme pour $M_k = I$ dans le cas-test 4.a. car le nombre d'itérations nécessaires est trop élevé.

Au contraire, en prenant $M_k = H_k + E_k$ une approximation de la Hessienne, nous avons un algorithme qui converge à l'ordre 2, donc il suffit de quelques dizaines d'itérations pour observer la convergence.

(4.2) On suppose que l'utilisateur connaît une solution x_* où les contraintes sont qualifiées, alors d'après les conditions de KKT, il existe λ_* tel que :

$$\begin{cases} \nabla_x \ell(x_*, \lambda_*) = \nabla f(x_*) + \nabla c(x_*) \lambda_* = 0 \\ c_E(x_*) = 0 \\ 0 \leq (\lambda_*)_I \perp c_I(x_*) \leq 0 \end{cases}$$

donc en donnant un λ_0 tel que $0 \leq (\lambda_*)_I \perp c_I(x_*) \leq 0$, c'est-à-dire dont les coefficients correspondant aux contraintes d'inégalité actives sont nuls, et le reste des coefficients non nuls, on peut directement calculer le multiplicateur optimal λ_* en inversant la matrice $\nabla_{c_{E \cup I^0}}(x_*)$ (qui est inversible car les contraintes sont qualifiées en x_*) :

$$\lambda_{*i} = \begin{cases} -(\nabla_{c_{E \cup I^0}}(x_*))^{-1} \nabla f(x_*)_i & \text{si } i \in E \cup I^0 \\ 0 & \text{sinon} \end{cases}$$

(4.3) Si (x_*, λ_*) vérifie les conditions d'optimalité du second ordre pour le problème avec contraintes d'égalité (CS2), c'est-à-dire

$$\begin{cases} \nabla_x \ell(x_*, \lambda_*) = 0 \\ c_E(x_*) = 0 \end{cases}$$

et $\nabla_{xx}^2 \ell(x_*, \lambda_*)$ définie positive sur le noyau de $\nabla_{c_E}(x_*)$, alors x_* est un minimum local si on a de plus que toutes les contraintes d'inégalité sont non-actives. En d'autres termes, on peut dire

que le point limite obtenu est un minimum local s'il vérifie les CS2 du problème avec contraintes d'égalité et qu'il n'est pas au bord de l'ensemble admissible pour les contraintes d'inégalité.

(4.4) Le comportement différent du nouveau solveur sur le cas-test 4.a par rapport au solveur sans contraintes d'inégalité provient directement de ce qui précède et de l'approximation M_k : l'algorithme converge vers un point qui vérifie les CS2 du problème avec contraintes d'égalité car on approxime le hessien du lagrangien par une matrice définie positive (ce qui n'était pas le cas auparavant), qui de plus assure que la solution d_k du problème osculateur est une *bonne* direction de descente (section (3.2)), et comme il n'y a pas de plancher, toutes les contraintes d'inégalité sont non-actives, on converge donc vers un minimum global.

(4.5) La convergence n'est plus quadratique car en modifiant H_k pour forcer la factorisation de Cholesky, on perd l'information sur les dérivées secondes.

Résultats numériques

Testons ce nouveau solveur sur différents cas-tests :

Dans le cas-test 4a, le solveur du problème quadratique osculateur converge vers le minimum global contrairement au précédent qui ne trouvait au mieux qu'un point stationnaire.

Si on prend simplement $M_k = I$ au lieu de $M_k = H_k + E_k$, on perd la convergence de l'algorithme,

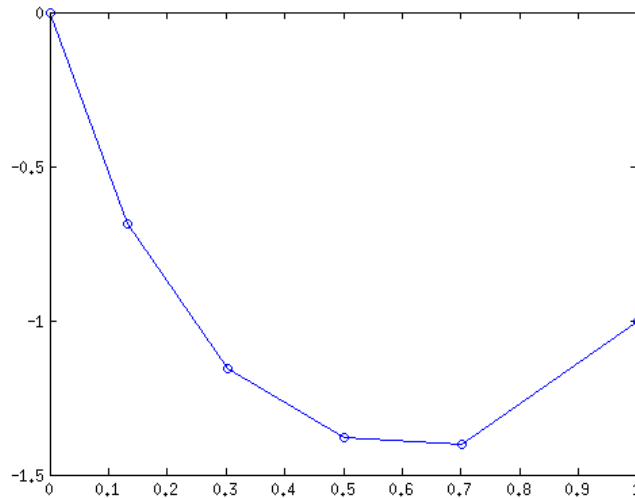
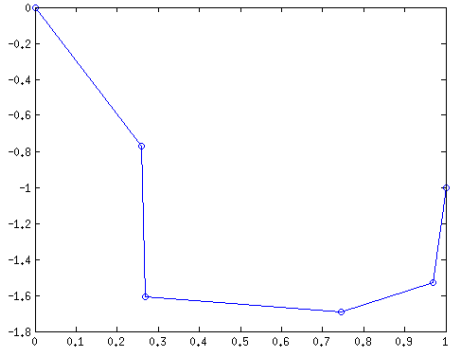
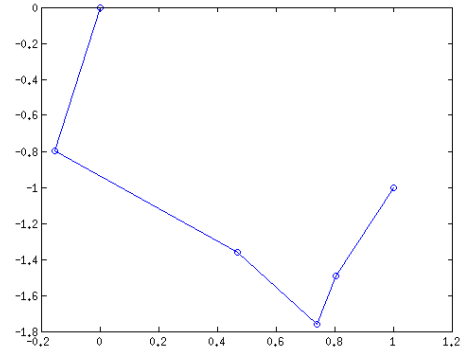


FIGURE 1 – Cas-test 4a avec optimisation quadratique successive

et en particulier dans le cas 4a, il oscille entre les deux positions de la Figure 8.



(a) position 1



(b) position 2

FIGURE 2 – Cas-test 4a avec $M_k = I$

Sur la Figure 3 est représentée la solution calculée pour le cas-test 4b, on voit que cette solution minimise l'énergie potentielle tout en restant au dessus du plancher.

Sur le cas 4c, l'algorithme converge bien vers le minimum et on peut observer sur la Figure 4

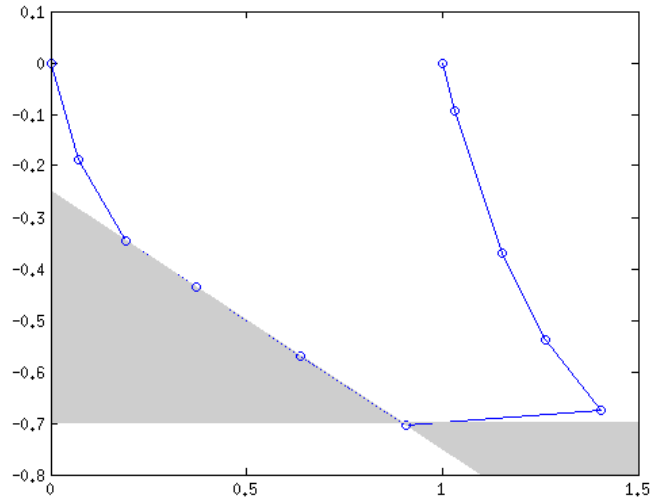


FIGURE 3 – Cas-test 4b

que les contraintes d'inégalité sont bien satisfaites. En effet la chaîne reste toujours au pire sur le plancher.

On peut aussi le vérifier numériquement en faisant tourner le solveur qui en mode bavard retournera un 1 si les contraintes d'inégalité sont satisfaites. La convergence de l'algorithme OQS est rapide, dans tous les cas on converge entre 18 et 30 itérations.

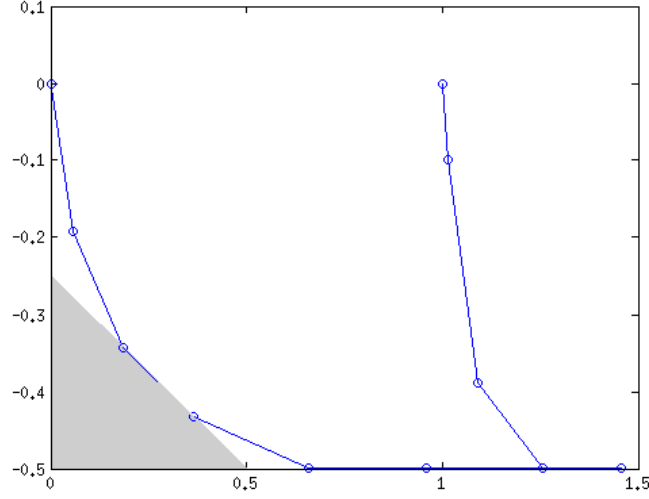


FIGURE 4 – Cas-test 4c

Méthode de quasi-Newton

Questions théoriques

(5.1) Pour évaluer la vitesse de convergence d'une suite $(z_k = (x_k, \lambda_k))_{k \in \mathbb{N}}$ vers une solution $z_* = (x_*, \lambda_*)$, on s'intéresse au quotient

$$\frac{\|z_{k+1} - z_*\|}{\|z_k - z_*\|}$$

où $\|\cdot\|$ est une norme sur \mathbb{R}^{2n_n} , par exemple la norme infinie.

Comme on ne connaît pas la solution, on cherche à exprimer la vitesse de convergence en terme d'une quantité dont la limite est connue : on pose alors

$$F : z_k \mapsto \begin{pmatrix} \nabla_x \ell(x_k, \lambda_k) \\ c_E(x_k) \\ \lambda_I^T c_I(x_k) \end{pmatrix}$$

où λ_I sont les multiplicateurs de Lagrange associés aux contraintes d'inégalité.

D'après les conditions de KKT, on a $F(z_*) = 0$ et pour k suffisamment grand, pourvu que les contraintes soient différentiables en z_* et $\nabla F(z_*)$ inversible

$$F(z_k) = F(z_*) + \nabla F(z_*)(z_k - z_*) + o(\|z_k - z_*\|)$$

donc

$$\|F(z_k)\| \leq C_1 \|z_k - z_*\| \quad \text{où} \quad C_1 = \|\nabla F(z_*)\|$$

et

$$z_k - z_* = \nabla F(z_*)^{-1} \nabla F(z_*)(z_k - z_*) = \nabla F(z_*)^{-1} (F(z_k) + o(\|z_k - z_*\|))$$

donc

$$\|z_k - z_*\| \leq C_2 \|F(z_k)\| + o(\|z_k - z_*\|) \quad \text{où} \quad C_2 = \|\nabla F(z_*)^{-1}\|$$

On a alors $F(z_*) = 0$ et $F(z_k) \sim (z_k - z_*)$ donc les vitesses de convergence seront déterminées à partir du quotient

$$\frac{\|F(z_{k+1})\|_\infty}{\|F(z_k)\|_\infty}$$

dont on prendra le logarithme de manière à vérifier numériquement le type de convergence.

(5.2) Comme M_k est définie positive, le problème quadratique osculateur est beaucoup plus facile à résoudre et a au plus une solution, cependant le PQO peut encore avoir des contraintes linéaires incompatibles empêchant de calculer une solution.

(5.3) Si le hessien du Lagrangien n'est pas définie positive, alors les exigences sur M_k rentrent en contradiction car imposer la définie positivité de M_k dégrade l'approximation du hessien du lagrangien.

(5.4.i) Si on multiplie l'équation (5.2) par δ_k , alors on obtient

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

Idéalement, pour approcher la Hessienne du Lagrangien, on devrait avoir

$$\nabla_x \ell(x_k, \lambda_{k+1}) \simeq \nabla_x \ell(x_{k+1}, \lambda_{k+1}) + \nabla_{xx}^2 \ell(x_{k+1}, \lambda_{k+1})(x_k - x_{k+1})$$

et comme M_k joue le rôle de $\nabla_{xx}^2 \ell(x_k, \lambda_k)$,

$$\gamma_k^\ell \simeq \nabla_{xx}^2 \ell(x_{k+1}, \lambda_{k+1})(x_{k+1} - x_k) \simeq M_{k+1}\delta_k$$

donc γ_k devrait être égal à γ_k^ℓ .

(5.4.ii) Si $\gamma_k^T \delta_k > 0$, alors pour un vecteur $v \in \mathbb{R}^n$ arbitraire, la formule (5.2) donne

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

or d'après Cauchy-Schwarz

$$(\delta_k^T M_k v)^2 \leq \|M_k^{1/2} \delta_k\|_2^2 \|M_k^{1/2} v\|_2^2 = (\delta_k^T M_k \delta_k)(v^T M_k v)$$

donc

$$v^T M_{k+1} v \geq v^T M_k v - \frac{(\delta_k^T M_k \delta_k)(v^T M_k v)}{\delta_k^T M_k \delta_k} + \frac{(\gamma_k^T v)^2}{\gamma_k^T \delta_k} = \frac{(\gamma_k^T v)^2}{\gamma_k^T \delta_k} \geq 0$$

avec égalité seulement si v est orthogonal à γ_k et colinéaire à δ_k , ce qui n'est vrai que si $v = 0$ car $\gamma_k^T \delta_k > 0$.

(5.4.iii) Maintenant si $\gamma_k = \gamma_k^\ell$, alors comme $H_k = \nabla_{xx}^2 \ell(x_{k+1}, \lambda_{k+1})$ n'est pas forcément définie positive

$$\gamma_k^T \delta_k \simeq \delta_k^T H_k^T \delta_k \not\geq 0$$

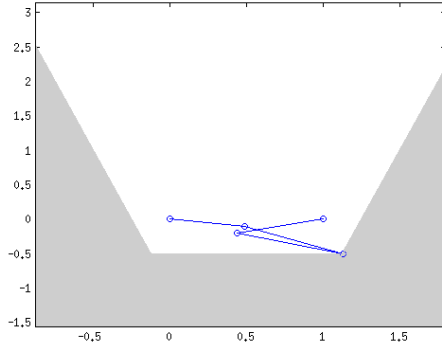
donc on ne peut plus appliquer le raisonnement précédent sur M_{k+1} .

Résultats numériques

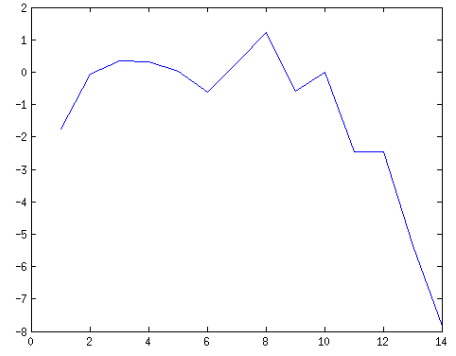
Dans le cas 5a notre algorithme converge en 15 itérations et on observe bien la convergence superlinéaire de l'algorithme, en effet

$$\log \left(\frac{\|F(z_{k+1})\|_\infty}{\|F(z_k)\|_\infty} \right) \rightarrow -\infty$$

Dans le cas 6a nous avons rencontré un problème. En effet, l'algorithme semble bien retourner la solution mais la norme du gradient de notre lagrangien explose et donc l'algorithme continue

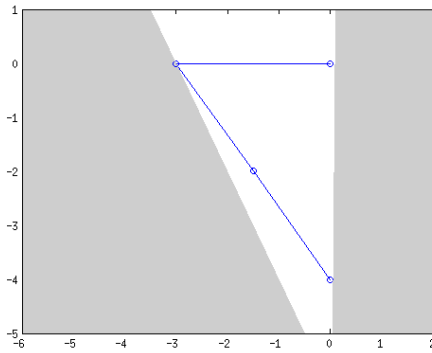


(a) Cas 5a solution

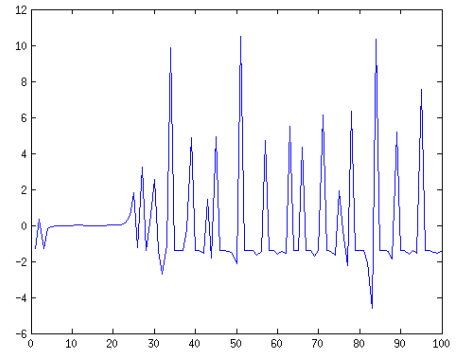


(b) Erreur-cas 5a

FIGURE 5 – Cas-test 5a

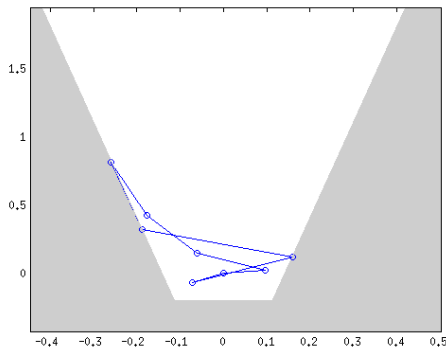


(a) Cas 6a solution

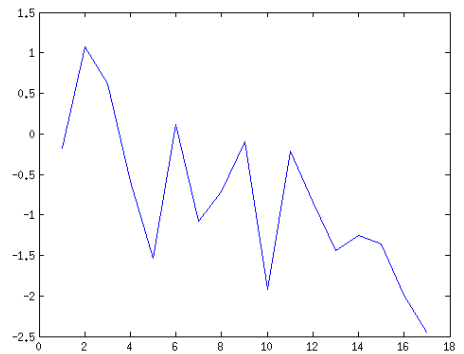


(b) Erreur-cas 6a

FIGURE 6 – Cas-test 6a



(a) Cas 6b solution



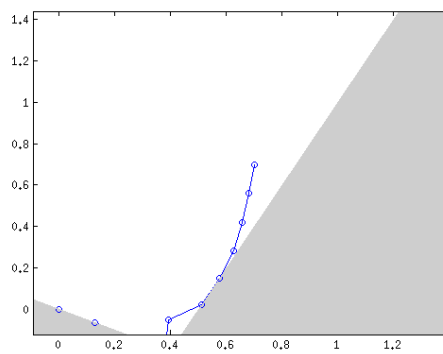
(b) Erreur-cas 6b

FIGURE 7 – Cas-test 6b

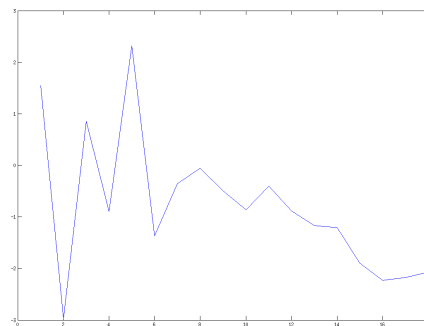
à itérer. Il est important de noter que nous n'avons pas implémenté de recherche linéaire dans cet algorithme et avons conservé un pas fixe de valeur 1.

Dans le cas 6b nous observons bien la convergence de l'algorithme. Nous avons choisi la chaîne initiale en lançant l'algorithme du TP précédent et choisi un point proche de la solution.

Dans le cas 6c nous avons choisi un plancher avec un saut et avons choisi une chaîne avec



(a) Cas 6c solution



(b) Erreur-cas 6c

FIGURE 8 – Cas-test 6c

des longueurs équidistantes (cf au scripte pour avoir les valeurs de xy , R,S). Voici le résultat obtenu. Nous observons une convergence en 19 itérations vers une solution .