

Projet d'optimisation en Matlab

Équilibre d'une chaîne articulée

Séance 5 : Version quasi-newtonienne

Dans cette séance, on poursuit le développement du code d'optimisation quadratique successive (OQS, SQP en anglais, pour Sequential Quadratic Programming) en y introduisant une version quasi-newtonienne. Le problème-test consiste toujours à trouver une position d'équilibre d'une chaîne au repos en présence d'un plancher.

1 Méthode de quasi-Newton

Dans les méthodes de quasi-Newton, on remplace le hessien du lagrangien qui intervient dans le problème quadratique osculateur de l'OQS par une matrice M_k , symétrique *définie positive*, mise à jour par l'algorithme. Dans le cadre de l'OQS, cette approche a plusieurs intérêts : il ne faut pas calculer les dérivées secondes des fonctions intervenant dans la définition du problème d'optimisation à résoudre et le problème quadratique osculateur (PQO) est toujours « mieux » posé (il a au plus une solution). Bien que la convergence soit « légèrement » plus lente qu'avec la méthode de Newton (elle n'est plus « que super-linéaire » ; c'est très bien quand même) et malgré les difficultés conceptuelles rencontrées par cette approche (voir plus loin), le second avantage cité ci-dessus fait que c'est souvent l'approche utilisée dans les codes commerciaux.

La version quasi-newtonienne de l'OQS génère donc une suite primale-duale $\{(x_k, \lambda_k)\}$ et une suite de matrices symétriques définies positives $\{M_k\}$ de la manière suivante. À l'étape k , on calcule d'abord la solution primale-duale $(d_k, \lambda_k^{\text{PQ}})$ du problème quadratique osculateur (voir la question 5.2)

$$\begin{cases} \min_{d \in \mathbb{R}^n} \nabla f(x_k)^T d + \frac{1}{2} d^T M_k d \\ c_E(x_k) + c'_E(x_k) d = 0 \\ c_I(x_k) + c'_I(x_k) d \leq 0, \end{cases} \quad (5.1)$$

dans lequel M_k joue le rôle de $L_k := L(x_k, \lambda_k) := \nabla_{xx}^2 \ell(x_k, \lambda_k)$, le hessien du lagrangien ℓ , $c'_E(x_k)$ et $c'_I(x_k)$ sont les jacobienes des contraintes d'égalité c_E et c_I . Ensuite on prend $x_{k+1} := x_k + \alpha_k d_k$, $\lambda_{k+1} := \lambda_k + \alpha_k (\lambda_k^{\text{PQ}} - \lambda_k)$ ($\alpha_k \in]0, 1]$ est le pas calculé par recherche linéaire) et on met à jour M_k par la formule de BFGS (c'est la formule la plus utilisée).

La formule de BFGS s'écrit :

$$M_{k+1} = M_k - \frac{M_k \delta_k \delta_k^T M_k}{\delta_k^T M_k \delta_k} + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k}. \quad (5.2)$$

Les vecteurs γ_k et δ_k sont déterminés de manière à forcer M_{k+1} à rester définie positive et à se rapprocher de L_{k+1} , ce qui, dans certains cas, peut être contradictoire (voir la question 5.3). 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

$$\gamma_k^\ell = \nabla \ell(x_{k+1}, \lambda_{k+1}) - \nabla \ell(x_k, \lambda_{k+1}). \quad (5.3)$$

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 $\gamma_k = \gamma_k^\ell$ (voir la question 5.4). Si bien que l'on utilisera la *correction de Powell* (1936-2015) qui consiste à prendre

$$\gamma_k = (1 - \theta)M_k\delta_k + \theta\gamma_k^\ell, \quad (5.4)$$

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

$$\theta = \begin{cases} 0.8 \frac{\delta_k^\top M_k \delta_k}{\delta_k^\top M_k \delta_k - (\gamma_k^\ell)^\top \delta_k} & \text{si } (\gamma_k^\ell)^\top \delta_k < 0.2 \delta_k^\top M_k \delta_k \\ 1 & \text{sinon.} \end{cases} \quad (5.5)$$

Il reste à dire ce que l'on doit prendre comme matrice initiale. On prendra $M_1 = I$ (matrice identité) à la première itération (on a pas d'autre information). 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 (ou la « valeur moyenne » de L_1 , voir la question 5.5) :

$$\eta_1 := \frac{\|\gamma_1\|_2^2}{\gamma_1^\top \delta_1}. \quad (5.6)$$

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.

2 Implémentation

2.1 Le paramètre options.deriv

Pour ne pas perdre le travail fait aux séances précédentes, on introduit un nouveau paramètre d'entrée pour la fonction `sqp`, appelé `options.deriv`, permettant de préciser le mode d'exécution désiré de `sqp`. L'optimiseur se présentera donc toujours comme suit

```
function [x, lme, lmi, info] = sqp (simul, x, lme, lmi, options)
```

L'optimiseur interprétera la valeur de `options.deriv` comme suit :

```
options.deriv = 1: methode de quasi-Newton (utilisation des dérivées premières),
               = 2: methode de Newton (utilisation des dérivées secondes),
```

2.2 Soins des sorties

On n'insistera jamais assez sur les informations pertinentes à faire afficher par l'algorithme. En ce qui concerne la version quasi-newtonienne, il est bon de savoir quand est-ce que la correction de Powell entre en jeu. Quelques informations sur le spectre de la matrice mise à jour peut aussi être une information précieuse (mais coûteuse). En mode peu bavard (une ligne par itération), on pourra avoir

```
-----
iter    |gl|    |cel|    (ci,lmi)    |x|    |lm|    Powell    cond(M)
  1    1.13e+00    1.36e+00    0.00e+00    1.2e+00    6.25e-01    4.1e-02    7.5e+01
  2    1.04e+02    8.08e+00    3.74e-09    2.7e+00    1.11e+01                4.8e+01
  3    1.89e+00    2.17e+00    1.43e-08    1.7e+00    3.08e-01                3.8e+01
```

4	9.40e-01	7.97e-01	7.03e-10	1.4e+00	9.81e-01		2.1e+01
5	1.65e+00	5.20e-01	9.20e-09	1.4e+00	2.18e+00		3.7e+01
6	4.25e+00	5.35e-01	3.08e-15	1.1e+00	1.89e+00		3.2e+01
7	1.28e+00	1.62e-01	2.22e-16	1.0e+00	1.98e+00		4.4e+01
8	4.30e-01	5.17e-02	1.56e-19	1.0e+00	2.02e+00		9.7e+01
9	3.20e-01	7.53e-03	1.11e-16	9.9e-01	1.89e+00	6.5e-01	1.9e+02
10	4.16e-01	1.66e-02	4.50e-11	9.7e-01	1.84e+00	6.1e-01	6.6e+02

...

où, en plus des étiquettes décrites dans d'autres séances, **Powell** est le coefficient θ de la formule (5.5) (s'il est différent de 1) et **cond(M)** est le conditionnement de M (reste raisonnable ici).

3 Cas-tests

- **Cas-test 5.a :** 6 barres de longueur

$L = [0.5 \ 0.3 \ 0.4 \ 1.2 \ 0.3 \ 0.3]'$;

Deuxième point de fixation de la chaîne :

$A = 0$;

$B = 0$;

Position initiale des nœuds :

$xy = [\begin{matrix} 0.2 & 0.5 & 0.8 & 1.0 & 1.2 & \dots \\ -0.4 & -0.6 & -0.4 & -0.2 & 0.1 \end{matrix}]'$;

Plancher :

$R = [-1]'$;

$S = [-0.1]'$;

- **Cas-test 5.b :** 3 barres de longueur

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

Deuxième point de fixation de la chaîne :

$A = 0$;

$B = -4$;

Position initiale des nœuds :

$xy = [\begin{matrix} -2 & 0 & \dots \\ 1 & -2 \end{matrix}]'$;

Plancher :

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

$S = [-2 \ 100]'$;

- **Cas-test 5.c :** ranger une chaîne dans un seau...

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

Deuxième point de fixation de la chaîne :

$$A = 0;$$

$$B = 0;$$

Position initiale des nœuds : à trouver.

Plancher :

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

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

- **Cas-test 5.d :** trouvez une configuration avec plancher originale et trouvez la position d'équilibre de la chaîne.

4 Questions

- 5.1.** Comment déterminer la vitesse de convergence d'une suite $\{(x_k, \lambda_k)\}$ convergeant vers une solution $z_* = (x_*, \lambda_*)$ sans connaître cette dernière ? Démontrez mathématiquement que votre test fonctionne et spécifiez dans quelles conditions il en est ainsi. (Réponse longue.)
- 5.2.** *Caractère bien posé du PQO.* Comme $M_k \succ 0$, que peut-il encore se passer, empêchant de calculer une solution du PQO (5.1) ? (Réponse courte.)
- 5.3.** *Approximation du hessien du lagrangien par la formule de BFGS.* Il est dit après (5.2) qu'il peut-être contradictoire d'approcher le hessien du lagrangien par une matrice définie positive. Pourquoi ? (Réponse courte.)
- 5.4.** *Sur le choix de γ_k dans la formule de BFGS.* Un certain nombre d'affirmations sont faites autour de (5.3) et (5.4) :
 - que le vecteur γ_k pris dans la formule (5.2) devrait être idéalement γ_k^ℓ ,
 - que l'on doit avoir $\gamma_k^\top \delta_k > 0$ pour que M_{k+1} soit définie positive,
 - que le choix $\gamma_k = \gamma_k^\ell$ ne garantit pas la définie positivité de M_{k+1} .

Pouvez-vous clarifier ces affirmations et dire en quoi la formule (5.4) de γ_k est préférable ?

- 5.5.** *Initialisation M_1 par $\eta_1 I$.* Pouvez-vous donner un sens à l'initialisation de $M_1 := \eta_1 I$ où η_1 est donné par (5.6) ?