Introduction à l'optimisation avec Scilab

Fabien Navarro

1 Optimisation linéaire-quadratique avec Scilab

Pour résoudre un problème d'optimisation sans contraintes on dispose de la commande `optim` qui recherche un minimum (local) d'une fonction dont on connaît le gradient. La syntaxe d'appel est la suivante : `[Fmin,xopt] = optim(F,x0)`
où F correspond à la fonction à minimiser et x_0 le point de départ de la méthode (par défaut, le solveur `optim` fait appel à l'algorithme du gradient conjugée, mais il est possible d'utiliser d'autres méthodes de résolution). La valeur retournée est la valeur du minimum et un x qui réalise ce minimum. La fonction F doit être de la forme suivante : `function [y,dy,ind] = F(x,ind)`
Elle doit retourner, pour un x donné, la valeur y de la fonction F évaluée en x et la valeur dy du gradient de F évaluée en x . Dans les cas simples il est préférable de calculer la valeur de ∇F, mais on peut également utiliser la fonction `derivative` (ou encore la fonction `numdiff`) qui calcul une approximation du gradient et de la hessienne en un point. L'argument d'entrée `ind` est un message envoyé au solveur de telle sorte à lui indiquer ce qu'il doit calculer. Une autre syntaxe d'appel possible de la fonction `optim` combiné avec la fonction `NDcost` permet d'éviter le calcul du gradient, elle correspond au cas où la fonction F est de la forme suivante : `function [y] = F(x)`
Dans ce cas, l'appel de la fonction `optim` doit être de la forme suivante :
`[Fmin,xopt] = optim(list(NDcost,F),x0)`
où la fonction `NDcost` calcul une approximation du gradient.

Pour plus de détails regarder l'aide de `NDcost`, celle de `derivative` et celle d'`optim`. Et tester

```
deff("[y,yprim,ind]=f(x,ind)","y=sin(%pi*x),yprim=%pi*cos(%pi*x)")
[fxopt,xopt]=optim(f,0.2)
[fxopt,xopt]=optim(f,0.50000000000000001)
[fxopt,xopt]=optim(f,0.5000000000000001)
[fxopt,xopt]=optim(f,3)
...
x=linspace(-4,4,200);fplot2d(x,f);plot(xopt,fxopt,'r.')
```

Vous pouvez egalement tester les exemples présents dans l'aide de Scilab (scilab 5).

2 Exercices

Exercice 1.
On considère la fonction :

$$\begin{aligned} g : \mathbb{R}^3 &\to \mathbb{R} \\ (x,y,z) &\mapsto x^2 + (x-y)^2 + 3(y-z)^2 - 2x + 6y - 6z \end{aligned}$$

1) Calculer ∇g et $\nabla^2 g$.

2) Ecrire une fonction `costg` prenant en entrée un vecteur (colonne) `x` à trois composantes et ayant comme paramètres de sortie `y` et `dy` qui calculent respectivement la fonction g ci dessus et son gradient ∇g (`function [y,dy,ind] = costg(x,ind)`).
3) Résoudre le problème $\min_{\mathbb{R}^3} g(x)$ en utilisant `optim`, avec $(x_0, y_0, z_0) = (0, 0, 0)^T$.
4) Ecrire le problème de minimisation sous la forme d'un système linéaire de type $Ax = b$.
5) Vérifier que la matrice A est définie positive.
6) Résoudre le système $Ax = b$, (en utilisant par exemple `inv`, `linsolve` ou`\`)
7) Conclure sur la nature de la solution.

Exercice 2. (Problèmes de gestion de stocks)
Une firme produit deux biens $p_1 = 80$ et $p_2 = 84$ en quantités q_1 et q_2. Le revenu issu de la vente est donc

$$R = p_1 q_1 + p_2 q_2,$$

La fonction de coût total (ou le critère, ou coût, ou fonction-objectif, ou fonction-coˆut,. . .) est donnée par

$$C = 8q_1^2 + 6q_2^2 - 2q_1q_2 - 40q_1 - 42q_2 + 180$$

et le bénéfice réalisé est :

$$B = R - C.$$

(Rq : vous pouvez aussi choisir de faire l'exo avec $C = q_1^2 + q_1q_2 + q_2^2$ en prenant $p_1 = 52$ et $p_2 = 44$).

On souhaite trouver les quantités $q_1^\star$ et $q_2^\star$ **maximisant** le bénéfice.

1) a) (facultatif) Calculer ("à la main") le gradient et la matrice hessienne de la fonction à minimiser.
   b) (facultatif) Á partir de la question précédente et des prix p_1 et p_2, trouver les quantités $q_1^\star$ et $q_2^\star$ maximisant le bénéfice.
   c) Ecrire (en *Scilab*) la fonction à minimiser.
   d) Résoudre le problème en utilisant la fonction `optim`.
   e) Représenter graphiquement (en utilisant par exemple `fplot3d` ou `plot3d` et `feval`) la fonction à minimiser sur le pavé $[-20q_1^\star, 20q_1^\star] \times [-20q_2^\star, 20q_2^\star]$.
   (Rq : pour la représentation graphique, il faut redéfinir la fonction à minimiser pour qu'elle prenne les arguments q_1 et q_2)
   f) En utilisant la fonction `fcontour` (ou `contour`), dessiner une suite de 50 lignes de niveau sur le pavé $[-20q_1^\star, 20q_1^\star] \times [-20q_2^\star, 20q_2^\star]$. Pour plus de clarté du dessin on pourra supprimer l'affichage de la valeur des lignes de niveau (avec la commande `xset("fpf",string=" ")`).

2) Même problème avec des prix adaptatifs, i.e. variant en fonction de la quantité de produits :

$$\begin{cases} p_1 = q_1 - q_2 \\ p_2 = q_1 + q_2 \end{cases} \qquad \left(\text{ou avec } \begin{cases} p_1 = 256 - 3q_1 - q_2 \\ p_2 = 222 + q_1 - 5q_2 \end{cases} \right)$$

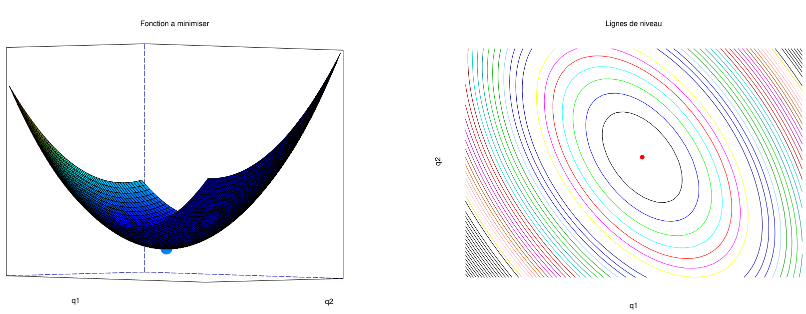

FIGURE 1 – Fonction à minimiser et lignes de niveau.