Master 2 RODECO Octobre 2017

Optimisation non-linéaire

Exercice 1

On cherche à minimiser la fonction

$$f(x) = (x_1 - 2)^2 + x_1(x_2 - 1) + 3(x_2 + 1)^2 + e^{\frac{x_1+x_2}{10}}.$$

L'objectif est ici d'implémenter une ou plusieurs méthodes de recherche d'un optimum local.

1. Énoncer les conditions d'optimalité du premier ordre pour la fonction f.

2. Programmer sous Matlab une méthode de gradient pour rechercher un optimum de la fonction f. On pourra sortir du programme si l'amélioration de la fonction objectif est inférieure à 10^{-6}. Les conditions de sortie seront définies par un nombre d'itérations de l'algorithme supérieur à 1000 ou une variation de la fonction objectif inférieure à 10^{-6}.

Rappel : Méthode du gradient

$$x_{k+1} = x_k - \alpha_k f'(x_k).$$

Aide : Squelette de la méthode du gradient fourni en annexe 1.

▷ On choisira un coefficient $\alpha_k = 0.4$. Que remarquez-vous ? Quel coefficient semble plus approprié ? Comment rendre la méthode plus robuste en cas de mauvais choix de α_k ?

3. En utilisant les résultats de la question 1, et à l'aide de la fonction `fzero`, vérifier les résultats obtenus par votre algorithme.

Aide : `fzero` est une méthode qui permet de trouver les zéros d'une fonction (méthode de bissection et méthode de la sécante).

4. Comment vérifier analytiquement que l'on obtient bien un minimum local ? La commande matlab

```
ezsurf('(x1-2)^2+x1*(x2-1)+3*(x2+1)^2+exp((x1+x2)/10)',[-10 10 -10 10]);
```

permet de visualiser la fonction f.

5. Implémenter la méthode de Newton

Exercice 2

Un jardinier souhaite installer un système d'arrosage automatique sur un terrain. Il a réussi à délimiter un emplacement où la terre est assez meuble pour recevoir le système d'arrosage qui est de type circulaire. Cet emplacement est borné par le système d'inéquations présenté ci-dessous (les valeurs sont en mètre).

$$\begin{cases} x & \geq & 0 \\ y & \leq & 8 \\ y & \leq & 1.4x \\ y & \leq & -2.5x + 47.5 \end{cases}$$

Le système est composé de deux arroseurs circulaires identiques. Leur portée (ajustable) est de 2m au minimum mais ne peut excéder 35m. Le jardinier dispose seulement de 20m de tuyau sécable pour alimenter les arroseurs, le robinet source se trouvant aux coordonnées $(0, 0)$. De plus, pour éviter des excès d'irrigation, les zones arrosées ne doivent pas se chevaucher. Compte-tenu de ces contraintes, le jardinier souhaite arroser la plus grande surface possible surface.

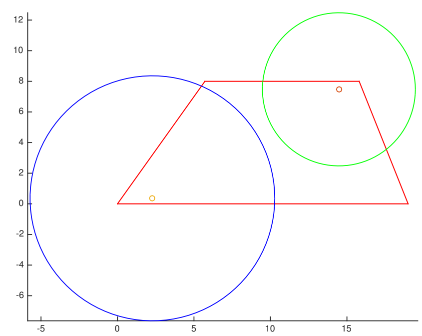

Figure 1 – Une solution réalisable pour le problème du jardinier.

1. Proposer une solution réalisable et calculer sa surface.
2. Proposer un modèle de ce problème d'optimisation.
3. A l'aide de la toolbox `optimization` de Matlab, programmer la fonction `fmincon` pour résoudre le problème du jardinier. Comment interpréter les résultats rendus par la fonction `fmincon` ?
4. Identifier les contraintes actives.

Annexe 1

La fonction `opt_gradient` prend en paramètres un point de départ, un nombre maximum d'itérations et une variation minimale de l'objectif. Elle fournit en retour, l'optimum trouvé par la méthode du gradient, la valeur du critère à ce point et le nombre d'itérations exécutées.

```
function [xopt,fopt, niter]= opt_gradient(x0,maxiter,varmin)

f=@(x1,x2) (x1-2)^2+x1*(x2-1)+3*(x2+1)^2+exp((x1+x2)/10);

% initialisations

...
clf;
ezcontour('(x1-2)^2+x1*(x2-1)+3*(x2+1)^2+exp((x1+x2)/10)',[-10 10 -10 10]);
hold on;

while (niter <= maxiter) & (abs(varf) >= varmin)

    % calcul du gradient

    % methode de descente

    plot ([x(1) xnew(1)], [x(2) xnew(2)],'-ko');

end

xopt=xnew
fopt=f(xnew)
```