Automne 2014

MAT-2410 : optimisation
séance de travaux pratiques 2

Dans cette séance de travaux pratiques, nous allons voir comment utiliser le logiciel Matlab pour résoudre des problèmes d'optimisation avec contraintes ayant la forme générique

$$\begin{array}{cc} \min & f(x) \\ \boldsymbol{g}(x) \leq 0 & \\ \boldsymbol{h}(x) = 0 & \end{array}$$

où $\boldsymbol{g} = (g_1, g_2, \ldots, g_m)$ sont les contraintes d'inégalité et $\boldsymbol{h} = (h_1, h_2, \ldots, h_p)$ les contraintes d'égalité.

Pour les problèmes posés en dimension 2, le logiciel Maple peut être utile afin de visualiser le problème. On consultera les instructions Maple de la séance 1.

Pour calculer la solution du problème d'optimisation avec contraintes $\min_{x \in U} f(x)$, nous allons utiliser la commande fmincon de la librairie Optimization Toolkit de Matlab. Voici la syntaxe d'appel.

```
[x,fval,exitflag,output] = fmincon(@fopt,x0,B,b,Beq,beq,lb,ub,@gcontr,options);
```

- @*fopt* est la fonction Matlab à optimiser,
- x_0 est le point de départ de l'algorithme,
- B et b sont liés aux contraintes linéaires de type inégalité $B\,x \leq b$,
- *Beq* et *beq* sont liés aux contraintes linéaires de type égalité $B_{eq}\,x = b_{eq}$,
- @*gcontr* est la fonction Matlab qui sert à préciser les contraintes non linéaires de type inégalité et d'égalité. Voir l'exemple ci-dessous.
- *options* sont les options de l'algorithme .

  ```
  options = optimoptions(@fmincon,'Display','iter'); % Display = iter ou notify pour afficher
  options = optimoptions(options,'GradObj','on');   % si on fournit le gradient de f
  options = optimoptions(options,'GradConstr','on');   % si on fournit le gradient des contra
  ```

- x est la solution (si l'algorithme a convergé),
- *fval* est la valeur de f au point x, i.e. $f(x)$,
- *exitflag* est un indice qui indique si l'algorithme a convergé (exitflag = 1),
- *output* fournit des détails sur la convergence de l'algorithme. Il suffit de taper *output*.

- On notera que si un ou plusieurs arguments ci-dessus sont manquants, on doit les préciser par l'ensemble vide [ ].

Voici un exemple de fonction objective à minimiser sous Matlab. Cet exemple correspond à la fonction quadratique $f(x) = \frac{1}{2}(Ax, x) - (b, x) + c$. L'appel se fera avec la commande $@(x)fopt(x, A, b, c)$.

```
function [f,gradf] = fopt(x,A,b,c)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here

f = x'*A*x/2 - b'*x + c;
if nargout > 1
   gradf = A*x- b;    %  gradient de f
end

end
```

Pour préciser l'ensemble des contraintes non linéaires, on doit fournir à Matlab une fonction avec la syntaxe suivante. Voici un exemple: $g(x) = \|x\|^2 - 1 = 0$

```
function [g,geq,gradg, gradgeq ] = gcontr(x)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here

g = [];
geq(1) = norm(x)^2 - 1;
if nargout > 2
   gradg = [];
   gradgeq(:,1) = 2*x;  % vecteur colonne
end
end
```

On notera que si une information est manquante, on écrit l'ensemble vide [ ]. L'ordre est important. On doit toujours commencer par les contraintes de type inégalité même si ce type de contrainte est vide, dans ce cas on pose $g = [\,]$.

Par exemple, si on veut résoudre le problème

$$\min_{B\,x \leq b} f(x),$$

on utilise la commande

```
[x,fval,exitflag,output] = fmincon(@fopt,x0,B,b,[],[],[],[],[],options);
```

1. Résoudre

$$\begin{array}{ll} \min & (x_1-2)^2+(x_2-2)^2 \\ x_1+2x_2 \leq 3 & \\ 3x_1+2x_2 \geq 3 & \\ x_1-2x_2 \leq 2 & \end{array}$$

   Interpréter géométriquement ce problème. Refaire en ajoutant les contraintes $x_1 \geq 0, x_2 \geq 0$.

2. Soit A une matrice symétrique de format $n \times n$. Il s'agit de vérifier numériquement les identités

$$\min_{\|x\|=1} (Ax,x) = \lambda_1 \qquad \text{et} \qquad \max_{\|x\|=1} (Ax,x) = \lambda_n$$

   où λ_1 et λ_n sont respectivement la plus petite et la plus grande valeur propre de A. On utilisera la commande Matlab $e = eig(A)$ pour calculer les valeurs propres. Pour préciser la contrainte, il est préférable de résoudre

$$\min_{\|x\|^2-1=0} (Ax,x) = \lambda_1 \Longleftrightarrow \min_{\|x\|=1} (Ax,x) = \lambda_1$$

   (a) Prendre $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$. Calculer les valeurs propres λ_1 et λ_n. Résoudre à l'aide de fmincon sans préciser le gradient (voir options). Refaire avec l'information du gradient. Vérifier que la solution est bien un vecteur propre, i.e. $Ax = \lambda x$.

   (b) Construire une matrice symétrique 4×4 et refaire a).

3. Résoudre

$$\begin{array}{ll} \max & x_1x_2x_3 \\ 0 \leq x_1+2x_2+2x_3 \leq 72 & \end{array}$$

   en prenant $x_0 = (10,10,10)$ comme point de départ. Interpréter géométriquement ce problème. Résoudre à l'aide de fmincon en calculant le gradient de f.

4. Résoudre

$$\begin{array}{ll} \min & e^{x_1}(4x_1^2+2x_2^2+4x_1x_2+2x_2+1) \\ 1.5+x_1x_2-x_1-x_2 \leq 0 & \\ -x_1x_2 \leq 10 & \end{array}$$

   en prenant $x_0 = (0,0)$ comme point de départ. Résoudre à l'aide de fmincon sans calculer les gradients. Essayer de maximiser! que remarquez-vous?

5. Résoudre

$$\begin{array}{ll} \max & x_1^3 + x_2^3 + x_3^3 \\ x_1^2 + x_2^2 + x_3^2 = 1 & \\ 2x_3^3 - x_2^2 \leq 0 & \\ x_1 \geq 0 & \\ x_3 \leq 0 & \end{array}$$

en prenant $x_0 = (1, 0, -1)$ comme point de départ. Résoudre à l'aide de fmincon en calculant les gradients de f et des contraintes.

6. On considère un ensemble de points du plan $\{x_i\}_{i=1}^m$. Il s'agit de trouver la plus petite boule centrée au point $x_0 = (a, b)$ et de rayon r qui inclut tous les points $\{x_i\}_{i=1}^m$. De manière précise, il faut résoudre

$$\begin{array}{ll} \min & \pi r^2 \\ \|x_1 - x_0\| \leq r & \\ \|x_2 - x_0\| \leq r & \\ \vdots & \\ \|x_m - x_0\| \leq r & \end{array}$$

Attention à la notation, les inconnues du problème sont (a, b, r). Résoudre à l'aide de fmincon sans calculer les gradients. Vous pouvez prendre 5 points ou plus. Fixer le point initial selon les points choisies.