CM08/P18/TD - Introduction à l'optimisation : problèmes sans contrainte avec MATLAB

1 Introduction

Ce premier exercice préliminaire a pour but de vous faire démarrer avec un programme MATLAB déjà écrit et qui vous servira de modèle pour les exercices suivants.

1.1 Utilisation basique de `fminunc`

Ouvrez MATLAB et sélectionnez New/Script dans le menu, puis recopiez le programme suivant (n'hésitez pas à poser des questions sur le code) :

```
opt = optimoptions('fminunc');
opt.Display='iter';

[x,fval,flag,out,grad,hess] = fminunc(@rosenbrock,[0.1;0.2],opt);

function f=rosenbrock(x)
  f=[(1-x(1))^2+(10*(x(2)-x(1)^2))^2];
end
```

Sauvez le script sous le nom `rosen.m` puis exécutez le à l'aide du bouton ▷. Vérifiez la solution `x`, la norme du gradient `grad` (fonction `norm`) et les valeurs propres de l'estimation de la matrice hessienne `hess` (fonction `eig`).

1.2 Utilisation du gradient calculé analytiquement

Apportez les modifications suivantes sur votre programme

```
opt = optimoptions('fminunc');
opt.Display='iter';
opt.SpecifyObjectiveGradient=true;

[x,fval,flag,out,grad,hess] = fminunc(@rosenbrock,[0.1;0.2],opt);

function [f,g]=rosenbrock(x)
  f=[(1-x(1))^2+(10*(x(2)-x(1)^2))^2];
  g=[-2*(1-x(1))+200*(-2*x(1))*(x(2)-x(1)^2)
                 200*(x(2)-x(1)^2)];
end
```

et exécutez le à nouveau. Que remarquez-vous ?

Dans la fonction `rosenbrock` introduisez une erreur de signe en remplacant la deuxième ligne par

```
g=[2*(1-x(1))+200*(-2*x(1))*(x(2)-x(1)^2)
```

et exécutez le script. Que pensez-vous du résultat obtenu ?

Dans votre script ajouter avant l'appel à `fminunc` la ligne

```
opt.CheckGradients=true;
```

puis exécutez le script. Proposez quelques remarques conclusives.

2 Optimisation d'un réservoir

On considère le problème d'optimisation de réservoir vu en cours. Montrer qu'en éliminant L grâce à la contrainte de volume il suffit de déterminer le minimum de la fonction

$$f(d) = c_1 \frac{4V}{d} + c_2 \frac{\pi d^2}{2}.$$

1. Expliquer pourquoi la contrainte $d > 0$ est inutile et pourquoi il existe forcément une solution au problème d'optimisation.
2. Montrer que la solution est donnée par $\hat{d} = \left(4 \frac{c_1}{c_2} \frac{V}{\pi}\right)^{1/3}$ et donnez les dimensions (d, L) obtenues pour
   — $V = 330\text{cm}^3$, $c_1/c_2 = 1/2$ (canette de bière),
   — $V = 10\text{m}^3$, $c_1/c_2 = 1$ (cuve en béton).
3. Ecrivez un programme MATLAB permettant de vérifier que `fminunc` vous donne les mêmes solutions.

3 Identification de paramètres

On veut déterminer l'enthalpie et l'entropie de réaction dans un processus de sorption d'hydrogène. Après avoir établi une pression initiale, on fait varier la température par paliers $\{T_1, \dots, T_n\}$ et on observe la pression obtenue à l'équilibre $\{P_1, \dots, P_n\}$

T	-4.75	19.93	49.88	80.04
P	3.51	9.64	22.95	47.21

1. En utilisant `fminunc` déterminer ΔH et ΔS pour ajuster le modèle

   $$P = \exp\left(\frac{\Delta H}{RT} + \frac{\Delta S}{R}\right),$$

   aux données ci-dessus, en déterminant le minimum de la fonction suivante, où $p = 2$

   $$f(\Delta H, \Delta S) = \sum_{i=1}^{n} \left| \exp\left(\frac{\Delta H}{RT_i} + \frac{\Delta S}{R}\right) - P_i \right|^p.$$

2. Dans la fonction f ci-dessus remplacer p par 1 et refaire l'optimisation. Que constatez-vous ?
3. On reprend la fonction f originale ($p = 2$). Calculer son gradient (avec papier et stylo) et mettre à jour votre code MATLAB pour que le gradient exact soit utilisé pendant l'optimisation. Vérifier que l'optimisation est plus rapide.
4. Remplacer f par

   $$h(\Delta H, \Delta S) = \sum_{i=1}^{n} \left(\frac{\Delta H}{RT_i} + \frac{\Delta S}{R} - \ln P_i\right)^2.$$

   et refaire l'optimisation (en laissant MATLAB approcher numériquement le gradient). Que pensez vous des valeurs obtenues pour ΔH et ΔS par rapport à celles obtenues précédemment ?