Plateforme Optimisation Statique
Sur machine

Prise en main de la Toolbox Optimization de Matlab

D'une manière générale, pour résoudre un problème d'optimisation sous Matlab avec la toolbox Optimization, il faut :

- Définir une fonction matlab objectif (la fonction à minimiser)

  function [f,G,H] = objfun (x)

- Définir les contraintes :

On distingue les cas où les contraintes sont linéaires ou non et égalités ou non.
Pour les contraintes linéaires, on définit les matrices et vecteurs : A, Aeq, b, beq
Pour les contraintes de type bornes, les vecteurs : l,u
Pour les contraintes non linéaires, une fonction matlab au format

function [c,ceq, gc, gceq] = confun(x)

- Appeler une méthode de résolution

Syntaxe :

[x,fval,exitflag]='methode'(@objfun,x0,A,B,Aeq,Beq,l,u,@confun,**options**)

où 'methode' est la méthode adaptée au problème à traiter (fminunc, fmincon, linprog, quadprog, lsqnonlin, bintprog,…)

Attention : il faut au préalable renseigner le vecteur **options** (sinon ce sont les valeurs par défaut qui seront appliquées). C'est ce vecteur qui permet de gérer l'affichage, la précision des calculs, le nombre d'itérations maximum, le nombre d'évaluations de fonction maximum, le calcul du gradient, etc.

Pour la liste complète des paramètres et leur usage :

```
help optimoptions
help optimset

exemple : optimoptions('fminunc')
```

Remarque 1 : Matlab propose également une interface graphique plus conviviale (commande optimtool).
Remarque 2 : N'hésiter pas à naviguer dans l'aide de Matlab et son tutoriel.

Travail demandé en séance :

1. **Optimisation sans contrainte :** Déterminer le minimum de la fonction

$$F(x)=100(x2-x1^2)^2+(1-x1)^2$$

En partant de (x1,x2)=(-1.9,2).

Comparer les résultats (nombre d'itération, temps de calcul) en utilisant différentes méthodes de résolution :

- En utilisant fminunc et les algorithmes 'trust-region' et 'quasi newton' avec les options 'HessUpdate'= 'bfgs','dfp' ou 'steepdesc' (gradient)
- En utilisant lsqnonlin

Afficher la valeur de la fonction et la progression des algorithmes à chaque itération (options Display='iter').

Attention : Lorsque la procédure se termine vérifier toujours le message de sortie 'exitflag' car il se peut que l'optimisation se finisse sans aboutir à la solution.

2. **Systèmes de grande dimension :** Traiter à présent le problème sans contrainte suivant par la méthode fminunc et l'algorithme 'trust-region':

$$f(x)=\sum_{i=2}^{n} 100(x_i - x^2_{i-1})^2 + (1-x_{i-1})^2$$

avec n= 10000.

   a. Sans calcul explicite du gradient et du Hessien
   b. Ajouter le calcul effectif du gradient dans la fonction objectif et modifier l'option 'GradObj '. Valider votre calcul du gradient avec 'DerivativeCheck'.
   c. Ajouter le calcul du Hessien H

   Comparer les temps de calculs obtenus en a., b. et c. lorsque n =100, 1000,10000

3. **Optimisation avec Contrainte :**

Résoudre numériquement les exercices de la feuille de TD sur Arche et valider vos réponses.

4. **Moindres Carrés**

   Les résultats d'une campagne de mesure sont stockés dans le fichier « indent.mat » . Charger le et afficher cette campagne (load). De bonnes raisons laissent à penser que la relation expliquant cette courbe est de la forme

$$y = a_1 e^{\lambda_1 t} + a_2 e^{\lambda_2 t}$$

   Déterminer les coefficients a1,a2,lamda1 et lamda2 en résolvant un problème au sens des moindres carrés. Utiliser les méthodes de Levenberg-Marquardt et Gauss Newton. Afficher la courbe « optimale ».

5. **Problème du Sac à dos.**

Jo le campeur part en randonnée dans la montagne.
Il ne peut emporter dans son sac à dos qu'un poids limité à P.
Chaque article i = 1,...,n qu'il peut potentiellement emporter pèse pi et lui procure une utilité ui pour sa randonnée.
Question : quels articles emporter pour maximiser l'utilité de son sac sans dépasser la limite de poids ?

Applications: Pour P=25 kg, déterminer le sac idéal

I	Pi	Ui
1	25	40
2	12,5	35
3	11,25	18
4	5	4
5	2,5	10
6	1,25	2

6. **Problème :** Il faut installer des stations d'incendie sur une zone comprenant six villes. On aimerait ne pas en installer 6. On pose le problème suivant : Comment installer le moins possible de stations tout en respectant les contraintes suivantes :
   - Les stations doivent être obligatoirement implantées en ville
   - Chaque ville doit pouvoir être atteinte en moins 15 minutes à partir d'au moins une station..

Table des temps de trajet intervilles:

	1	2	3	4	5	6
1	0	10	20	30	30	20
2		0	25	35	20	10
3			0	15	30	20
4				0	15	25
5					0	14
6						0

7. **Réglage d'un PID.**
   Il n'est généralement pas simple de régler un PID, il existe des méthodes par exemple Broida mais le résultat est loin d'être optimisé, le réglage de l'effet dérivé étant mal maîtrisé.

   Voici une procédure d'optimisation sur la réponse à un échelon. Elle consiste à minimiser l'aire séparant la réponse de la consigne.

   Soit $\min \int_0^T \|y - y_d\|^2$ où y est la réponse et yd la consigne. Comprendre et reprendre l'exemple donné dans le tutorial de Matlab « lsqnonlin with a Simulink Model ».

8 . Bureau d’étude « problème de Schumacher »