

TP n° 1 : Introduction au Toolbox d'optimisation sous Matlab

1. Rappel sur le graphisme 3D :

Avec la commande mesh on peut aisément avoir une représentation d'une fonction 3D. Cependant il faut construire la grille des coordonnées des points en lesquels on a les valeurs de la fonction à l'aide de la fonction meshgrid.

Exemple 3:

```
x = 0:0.1:2;y = -1:0.1:1;  
z=cos(pi*x).*sin(pi*y) ;  
[X,Y] = meshgrid(x,y);          %créer un maillage 2D(x,y)  
Z=cos(pi*X).*sin(pi*Y) ;  
mesh(X,Y,cos(pi*X).*sin(pi*Y)) % tracer le graphique en3D de Z  
xlabel('x');ylabel('y');zlabel('z');  
title('z=cos(pi*x)*sin(pi*y)')  
on peut aussi utiliser la fonction plot3() :  
plot3(x,y,z)          %tracer z en fonction de x et y en 3D.
```

Exemple 4: t = 0:0.01:20;

```
x = exp(-0.05*t).*cos(t); y = exp(-0.05*t).*sin(t); z = t;  
plot3(x,y,z)  
grid on
```

2. Calcul des points critiques : Pour trouver les points extrêmes (ou points critiques) d'une fonction de deux variables, par exemple : $f(x, y) = x^3 + y^3 + 3x^2 - 3y^2 - 8$, on doit trouver les points qui annulent les dérivées partielles de la fonction.

1-Résoudre analytiquement ce problème.

2-Utiliser les instructions suivantes pour vérifier votre résultat :

```
syms x y ;  
f=x^3+y^3+3*x^2-3*y^2-8;  
fx=diff(f,x)  
fy=diff(f,y)  
S=solve(fx,fy)
```

La commande solve trouve les solutions qui sont égales à zéro simultanément pour les deux fonctions dérivées. S est une structure variable. Pour voir les valeurs de S taper: [S.x, S.y]

Le résultat montre les points critiques pour la fonction analysée $\{(0,0),(0,2),(-2,0),(-2,2)\}$.

Pour visualiser les résultats on peut utiliser la fonction :

```
[x,y]= meshgrid (-3:0.1:3);  
z= x.^3+y.^3+3*x.^2-3*y.^2-8;  
mesh(x,y,z)  
xlabel('x')  
ylabel('y')  
zlabel ('z=f(x,y)')
```

3. Minimisation unidimensionnel : « fminbnd »

Les méthodes d'optimisation pour les fonctions à une variable s'appellent recherche par ligne ('line search'). La fonction **fminbnd** de matlab calcule le min des fonctions à une seule variable.

On crée une fonction externe dans un fichier .m. Travailler avec des fichiers externes permet de simplifier et réduire les erreurs.

```
function y = f(x)
```

```
y = 1./((x-0.3).^2 + 0.01)+ 1./((x - 0.9).^2 + 0.04) -6;
```

Dans l'invite de commande Matlab on peut observer la fonction par :

```
clear all
```

```
fplot('f', [-5 5])
```

```
grid on
```

On fait un zoom pour observer où se trouve notre minimum

```
fplot('f', [-5 5 -10 25])
```

```
grid on
```

Notre minimum se trouve entre 0.3 et 1 .

```
min = fminbnd('f',0.3,1,optimset('Display','iter'));
```

```
fplot('f',[0 2])
```

```
hold on ;
```

```
plot(min,f(min),'r*') ;
```

La fonction **fminbnd** permet de trouver le minimum de la fonction dans un intervalle donné.

Dans les options, on peut voir les approximation successives et l'algorithme que Matlab utilise avec **optimset('Display','iter')**. **fminbnd** trouve les minimums locaux. C'est important de choisir une bonne approximation initiale. **fminbnd** a une convergence lente quand la solution est proche de l'intervalle.

Exercice : Soit la fonction f définie par : $F(x)=2x^3-3x^2+8x-1$

- 1) Donner les points critiques et leur nature (analytiquement)
- 2) Tracer le graphe de la fonction.
- 3) On utilisant la fonction **fminbnd** donner la valeur **min** de la fonction.