Ministère de l’enseignement supérieur et de la recherche scientifique

**UNIVERSITÉ DE BATNA2 - FACULTÉ DE TECHNOLOGIE - DÉPARTEMENT D'ELECTRONIQUE**

Master 1 : Automatique et Systèmes

**Matière : Optimisation TP 2 – La méthode de Newton et quasi-Newton**

**Manipulation I :**

Par l’application de la méthode de Newton pour minimiser la fonction objectif *f (x,y)* ci- dessous :

*f(x, y) = x2 + y2;*

En prenant le point de départ *x0=* 3 et *y0=3,* avec ε = 10^-3.

Avec l’algorithme du Newton est donné par la forme itérative suivante :

*XK+1=XK- [∇2F(XK)]-1\*∇F(XK)*

1. Ecrire un programme Matlab qui permet de calculer la valeur x\* et y\* qui minimiser la fonction objective *f (x,y)*.
2. Tracer la variation de la fonction *f(x,y)* en fonction de nombre d’itérations pour 𝑛 = 20;
3. Expliquer le phénomène de convergence de l’algorithme.
4. Donner l’interprétation de chaque résultat obtenu.

 **Manipulation II :**

Par l’application de la méthode de quasi-Newton pour minimiser la fonction objectif *f (x,y)* ci- dessous :

*f(x, y) = x2 + y2;*

En prenant le point de départ *x0=* 3 et *y0=3,* avec ε = 10^-3.

Avec l’algorithme du quasi-Newton est donné par la forme itérative suivante :

S0=*[∇2F(X0)]-1*

*XK+1=XK- SK\*∇F(XK)*

*Avec : SK+1=SK+*$\frac{(dk-sk\*yk)(dk-sk\*yk)^{T}}{(dk-sk\*yk)^{T}\*yk}$

*dk=xk+1- xk*

*YK=∇F(XK)- ∇F(XK-1)*

1. Ecrire un programme Matlab qui permet de calculer la valeur x\* et y\* qui minimiser la fonction objective *f (x,y)*.
2. Tracer la variation de la fonction *f(x,y)* en fonction de nombre d’itérations pour 𝑛 = 20;
3. Expliquer le phénomène de convergence de l’algorithme.
4. Donner l’interprétation de chaque résultat obtenu.

Révision générale pour le gradient et la matrice Hessienne :

% Function Definition:

syms X Y; % TAKE X, Y AS SYMBOLS

f =X^2 +Y^2;

x(1) = 2;

y(1) = 2;

% Gradient Computation :

df\_dx = diff(f, X);

df\_dy = diff(f, Y);

J = [subs(df\_dx,[X,Y], [x(1),y(1)]) subs(df\_dy, [X,Y], [x(1),y(1)])];

% Gradient by values

% subs: returns a copy of s replacing symbolic variables in s with their values obtained from the calling

function and the MATLAB® workspace, and then evaluating s. Variables with no assigned values remain as variables.

% Hessian Computation:

ddf\_ddx = diff(df\_dx,X);

ddf\_ddy = diff(df\_dy,Y);

ddf\_dxdy = diff(df\_dx,Y);

ddf\_dydx = diff(df\_dy,X);

ddf\_ddx\_1 = subs(ddf\_ddx, [X,Y], [x(1),y(1)]);

ddf\_ddy\_1 = subs(ddf\_ddy, [X,Y], [x(1),y(1)]);

ddf\_dxdy\_1 = subs(ddf\_dxdy, [X,Y], [x(1),y(1)]);

ddf\_dydx\_1 = subs(ddf\_dydx, [X,Y], [x(1),y(1)]);

H = [ddf\_ddx\_1, ddf\_dxdy\_1; ddf\_dydx\_1, ddf\_ddy\_1] % Hessian matrix

S=INV (H) ; INERSE OF THE HESSIAN MATRIX

X=X-S\*J’