

TP5 : Master1

Spécialité : G01: Commande Electrique+ G02:Réseaux Electriques + G03:Energies
Renouvelables + Robotique

I. Objectif

L'objectif de ce TP est d'écrire un programme sous MATLAB qui permet d'appliquer quelques méthodes d'optimisation sans contraintes et de comparer ces dernières afin de bien comprendre les différences entre eux.

1_Méthode de gradient

Considérons le problème d'optimisation suivant :

$$f(x_1, x_2) = x_1^2 + x_1x_2 + 3x_2^2$$

$$x_0 = (3,3); \epsilon = 0.000001$$

Taper le programme suivant :

%%%

```
function [xopt, fopt, niter, gnorm, dx] = grad_descent(varargin)
if nargin==0
% define starting point
x0 = [3 3]';
elseif nargin==1
% if a single input argument is provided, it is a user-defined starting
% point.
x0 = varargin{1};
else
error('Incorrect number of input arguments.')
end
% termination tolerance
tol = 1e-6;
% maximum number of allowed iterations
maxiter = 1000;
% minimum allowed perturbation
dxmin = 1e-6;
% step size ( 0.33 causes instability, 0.2 quite accurate)
alpha = 0.1;
% initialize gradient norm, optimization vector, iteration counter,
perturbation
gnorm = inf; x = x0; niter = 0; dx = inf;
% define the objective function:y
f = @(x1,x2) x1.^2 + x1.*x2 + 3*x2.^2;
% plot objective function contours for visualization:
figure(1); clf; ezcontour(f, [-5 5 -5 5]); axis equal; hold on
% redefine objective function syntax for use with optimization:
f2 = @(x) f(x(1),x(2));
% gradient descent algorithm:
while and(gnorm>=tol, and(niter <= maxiter, dx >= dxmin))
% calculate gradient:
g = grad(x);
gnorm = norm(g);
% take step:
xnew = x - alpha*g;
% check step
if ~isfinite(xnew)
```

```
display(['Number of iterations: ' num2str(niter)])
error('x is inf or NaN')
end
% plot current point
plot([x(1) xnew(1)], [x(2) xnew(2)], 'ko-')
refresh
% update termination metrics
niter = niter + 1;
dx = norm(xnew-x);
x = xnew;
end
xopt = x;
fopt = f2(xopt);
niter = niter - 1
% define the gradient of the objective
function g = grad(x)
g = [2*x(1) + x(2)
x(1) + 6*x(2)];
```

Questions :

1- Exécuter le programme.

Conclure.

2- Exécuter le programme pour différents pas

$\alpha = 0.1, 0.2, 0.32$ et 0.33 .

Conclure.

3- Comparer les différentes trajectoires obtenues. Commentaires (nombres d'itérations, temps de calcul, ...).