Steepest descent

Iniziamo con una routine per l'ottimizzazione non vincolata.

Vogliamo minimizzare la funzione $f(\underline{x})$ per $\underline{x} \in |R$.

Il metodo della discesa del gradiente (Steepest descent) consiste nello scegliere una funzione di tentativo x_0 in modo arbitrario e di procedere in modo iterativo come segue:

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k p_k$$

Dove:

- $p_k = -\nabla f(\underline{x}_k)$
- α_k appartenente ai reali positivi. α_k è il passo di discesa e, per il momento verrà supposto costante.

Codice Matlab (Copiare e incollare le istruzioni seguenti)

```
function [xstar, vstar, counter, tolerance, xks, fks]
= ...
    steepestdescent(f, x0, epsilon, maxiterations)
x = x0;
counter = 0; tolerance = 1e30;
xks = x; fks = feval(f, x);
fprintf('function \t counter \t tolerance\t\n');

while tolerance > epsilon && counter < maxiterations
    fid=fopen('log_1.m','w');
    d = -grad(f, x);
    alpha=0.1;
    x = x + alpha * d;
    v = feval(f, x);
    xks = [xks, x];
    fks = [fks, v];
    tolerance = norm(d, 2);
    counter = counter + 1;
    fprintf('%f \t\t %f \t\t
%f\n',v,counter,tolerance);
    for m=1:length(x)
       fprintf(fid,'Z(%i) = \t %4.6f \t \n',m,x(m));
    end
    fclose(fid);
end
xstar = x;
vstar = v;
end
```

```
function g = grad(fun, x0)
delta = x0 / 1000000;
for i = 1 : length(x0)
    if x0(i) == 0
        delta(i) = 1e-12;
    end
    u = x0;
    u(i) = x0(i) + delta(i);
    f1 = feval ( fun, u );
    u(i) = x0(i) - delta(i);
    f2 = feval ( fun, u );
    g(i,1) = (f1 - f2) / (2 * delta(i));
end
end
```

Esempio pratico:

```
Command Window
>> f=@(x)(x.^2-2.*x)
x0=3
epsilon=0.001;
maxiterations=5;
[xstar, vstar, counter, tolerance, xks, fks] =steepestdescent(f, x0, epsilon, maxiterations);

f =

  function_handle with value:

    @(x)(x.^2-2.*x)


x0 =

     3

function      counter      tolerance
1.560000          1.000000        4.000000
0.638400          2.000000        3.200000
0.048576          3.000000        2.560000
-0.328911         4.000000        2.048000
-0.570503         5.000000        1.638400
>> xstar

xstar =

    1.6554

fx >> |
```

Nella figura vogliamo trovare la x per cui è minima la

$$f(x) = x^2 - 2x$$

La soluzione, in forma chiusa, è

$$x = 1$$

$$f(1) = -1$$

Il risultato, con sole 5 iterazioni è abbastanza deludente fornendo come soluzione:

$$x = 1.6554$$

$$f(1) = -0.570503$$

Aumentando a 10 le iterazioni otteniamo

```
Command Window
x0=3
epsilon=0.001;
maxiterations=10;
[xstar, vstar, counter, tolerance, xks, fks] =steepestdescent(f, x0, epsilon, maxiterations);
xstar
vstar

f =

  function_handle with value:

    @(x)(x.^2-2.*x)


x0 =

     3

function      counter      tolerance
1.560000          1.000000         4.000000
0.638400          2.000000         3.200000
0.048576          3.000000         2.560000
-0.328911         4.000000         2.048000
-0.570503         5.000000         1.638400
-0.725122         6.000000         1.310720
-0.824078         7.000000         1.048576
-0.887410         8.000000         0.838861
-0.927942         9.000000         0.671089
-0.953883         10.000000        0.536871

xstar =

    1.2147


vstar =

   -0.9539

fx >>
```

$$x = 1.2147$$

$$f(1) = -0.9539$$

Ovviamente aumentando le iterazioni si raggiunge la soluzione entro la tolleranza richiesta (epsilon=0.001)