A-Gradient Methods

Gradient Methods form a group of iterative methods to be applied to differentiable functions. These methods share the same methodology.
Let d a vector in IR^n and x^k a point in IR^n such that $\nabla f(x^k) \neq 0$.
For $s \in IR$, we define: $g(s) = f(x^k + sd)$.
The vector d is then a descent direction if $g'(0) < 0$. Deriving g with respect to s gives:
$g'(0) = d^t.\nabla f(x^k)$. This scalar product of two vectors can also be written:
$g'(0)= \cos\theta \| \nabla f(x^k) \|.\| d \|$, with θ angle between $\nabla f(x^k)$ and d.

Suppose that $\| d \| =1$, g'(0) is minimized if $\cos\theta = -1$, thus:

$$d= -\nabla f(x^k) / \| \nabla f(x^k) \|$$

This direction gives the steepest descent direction.
The main difference between gradient methods is in the choice of s_k and d^k.

Steepest Descent Method

This is the most popular gradient method. In this method we consider $d^k= -\nabla f(x^k)$, thus $g(s) = f(x^k - s.\nabla f(x^k))$, and we will try to find $s_k \geq 0$ such that g is minimized. The subproblem to solve is unidimensional.
The algorithm could be described as follows :

Initialization :
Choose a starting point x^0, let $k =0$ and go to the main step.
Main step :

```
Repeat :
- $d^k \leftarrow - \nabla f(x^k)$
- Find $s_k \geq 0$ that solves Minimize $f(x^k + s_k d^k)$.
- $x^{k+1} \leftarrow x^k + s_k d^k$
- $k \leftarrow k + 1$
Do while terminal condition is not satisfied.
```

The terminal condition could be: $\| \nabla f(x^k) \| \leq \varepsilon$, with ε very small and positive.
Or $| f(x^{k+1})- f(x^k) | \leq \varepsilon$.
We can also set a number of terminal conditions.

It is possible to show that if $f(x)$ is differentiable at least one time and takes finite values when x goes to infinity, this algorithm converges to a stationnary point (gradient is zero).

The main weakness of this method is that the convergence speed could be very slow. This is due to the fact that the successive directions are orthogonal.

Example : Minimize $(x_1-2)^4+(x_1-2x_2)^2$ starting from (0,3).

Newton Method (Newton-Raphson)

We consider the second order approximation of a function f in the neighborhood of x^k.
We define : $q(x) = f(x^k) + (x- x^k)^t.\nabla f(x^k) + \frac{1}{2}(x- x^k)^t. H(x^k). (x- x^k)$
A necessary minimum condition for the function q is $\nabla q(x)=0$. This would be written as $\nabla f(x^k) +H(x^k). (x- x^k) =0$. If $|H(x^k)|$ is not zero, the minimum point of $q(x)$ would be *:*

$$x^{k+1}= x^k- H(x^k)^{-1}\nabla f(x^k)$$

This method would guarantee good convergence results only if $|H(x^k)|$ *is not zero,* if $f(x^{k+1})$ is less or equal to $f(x^k)$ et and the starting point is close to a stationnary point.

Example : Minimize $(x_1-2)^4+(x_1-2x_2)^2$ starting from (0,3).

II-Conjugate Gradient Methods

Fletcher and Reeves Method

The Fletcher and Reeves Method could be described as follows:

Initialization :

Choose $\varepsilon > 0$, a positive integer n and a starting point x^1.
Let $y^1 = x^1$, $d_1 = -\nabla f(y^1)$, $k = j = 0$ and go to main step.

Main Step:

1. *If* $\| \nabla f(y^j) \| \leq \varepsilon$, Stop.
   Else, let λ_j be the optimal solution to minimize $f(y^j + \lambda d_j)$ with λ_j positive, and let $y^{j+1} = y^j + \lambda_j d_j$. If $j < n$, go to step 2, else go to step 3.
2. Let $d_{j+1} = -\nabla f(y^{j+1}) + \alpha_j d_j$, *with :* $\alpha_j = \| \nabla f(y^{j+1}) \|^2 / \| \nabla f(y^j) \|^2$
   Replace j by $j+1$ and go to step 1.
3. Let $y^1 = x^{k+1} = y^{n+1}$ and $d_1 = -\nabla f(y^1)$.
   Let $j=1$, $k=k+1$ and go to step 1.

Davidon-Fletcher-Powell Method

The Davidon-Fletcher-Powell Method could be described as follows:

Initialization :

Choose $\varepsilon > 0$, a positive integer n and a starting point x^1.
Choose a symetric matrix D_1 positive semi-definite.
Let $y^1 = x^1$ and $k = j = 1$ and go to main step.

Main Step:

1. If $\| \nabla f(y^j) \| \leq \varepsilon$, Stop.
   Else, let $d_j = -D_j \nabla f(y^j)$, and let λ_j be the optimal solution to minimize $f(y^j + \lambda d_j)$ with λ_j positive, and let $y^{j+1} = y^j + \lambda_j d_j$. If $j < n$, go to step 2.
   If $j = n$, let $y^1 = x^{k+1} = y^{n+1}$, replace k by $k+1$, let $j=1$and repeat step 1.
2. Build $D_{j+1} = D_j + (p_j p_j^t)/(p_j^t q_j) - (D_j q_j q_j^t D_j)/(q_j^t D_j q_j)$
   avec $p_j = \lambda_j d_j$ *et* $q_j = \nabla f(y^j) - \nabla f(y^{j+1})$
   Replace j by $j+1$ and go to step 1.

HomeWork

Minimize $-12x_2 + 4x_1^2 + 4x_1^2 + 4x_1 x_2$

Reference :

NonLinear Programming, Theory and Algorithms, Bazaraa M.S., Sherali H.D and Shetty C.M., second edition, WILEY.