Profa. Cristiane - 08/04/2006
Aula 9 - Métodos de Otimização Irrestrita (continuação)

3.6 - Aplicação Prática de Métodos de Quase-Newton em Alguns Problemas de Otimização Irrestrita

Método de Newton Modificado: newt-mod.m (Alteração no laço do **While**)

```
        ...
while (erro > epsilon & k < limax),
     k = k+1;
    xv = x;
    d1 = grad2(Q,b,xv);
     d = -S*d1;
    fx = funcao2(Q,b,xv);
   solucao(k-1,1) = fx;
   lam = passoN(S,Q,b,xv);
     p = lam*d;
     x = xv + p;
  erro = norm(x-xv);
end
        ...

function y = passoN(S,Q,b,x)
         g = grad2(Q,b,x);
         y = (g'*S*g)/(g'*S*Q*S*g);
```

Método de Davidon-Fletcher e Powell (DFP) - Resolução do Exercício 1 da Aula 7

ALG. 6 Seja $x_k \in \Re^n$ tal que $\nabla f(x_k) \neq 0$. Defina $H_0 \in \Re^{nxn}$ simétrica e definida positiva. Os passos para determinar x_{k+1} são:

Passo 1: Calcular $d_k = -H_k \, \nabla f(x_k)$;

Passo 2: (Busca Linear Exata) Determinar λ_k, minimizador de $f(x_k + \lambda \, d_k)$ sujeito a $\lambda \geq 0$, e definir $x_{k+1} = x_k + \lambda_k d_k$

Passo 3: Definir $p_k = \lambda_k d_k = x_{k+1} - x_k$, $\; q_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ e calcular $H_{k+1} = H_k + \dfrac{(p_k \, p_k^t)}{(p_k^t \, q_k)} - \dfrac{(H_k \, q_k \, q_k^t \, H_k)}{(q_k^t \, H_k \, q_k)}$

Teorema 1 : Se o método DFP é aplicado para minimizar uma função quadrática com hessiana definida positiva fazendo busca linear exata, então:
(a) Se H_k é definida positiva então H_{k+1} também o é;
(b) $\{d_0, d_1, \cdots, d_{n-1}\}$ é um conjunto LI;
(c) $H_k \, q_j = p_j$ para todo $j \leq k$;
(d) $x_n = x^*$; (e) $H_n = Q^{-1}$.

dpf.m (Alteração no laço do **While**)

```
disp('      Minimizacao Irrestrita - Metodos de Quase-Newton')
disp('           Metodo de Davidon-Fletcher-Powell')
        ...
ltil = 0.99;
%H = inv(Q);
H = eye(4);
while (erro > epsilon & k < limax),
   k = k+1;
   xv = x;
   Hv = H;
   d1 = grad2(Q,b,xv);
    d = -Hv*d1;
   fx = funcao2(Q,b,xv);
   solucao(k-1,1) = fx;
   lam = passo1(ltil);
   p= lam*d;
   x = xv + p;
   q = grad2(Q,b,x) - d1;
   H = Hv + ((p*p')/(p'*q)) - ((Hv*q*q'*Hv)/(q'*Hv*q));
   erro = norm(x-xv);
end
figure(1), plot(solucao,'+-'), xlabel('numero de iteracoes'), ylabel('valor da funcao');
```