06-606 Computational Methods

Assignment #2

1. Given the system of equations

$$f_1 = 2x_1^2 + x_2^2 - 6 = 0$$
$$f_2 = x_1 + 2x_2 - 3.5 = 0$$

a. Solve with Newton's method (unit step size) using as a starting point $x_1 = 2.0,\ x_2 = 1.0$

b. Using as a starting point $x_1 = x_2 = 0$, solve the system with Newton's method with the Armijo line search. What problems are likely to occur from this starting point and what must you do to get the method to converge to a tolerance of $f^T f < 10^{-8}$?

2. Given is the system of linear equations in n variables x

$$f(x) = b + Ax = 0$$

where A is a non-singular matrix. Show the convergence properties of Newton's method. What properties does Broyden's method have on such a system (some literature search is required here) ?

3. Reformulate the following equations so they do not have poles. Why is this necessary?

$$f_1 = \exp\,(x/(y^2 - 6))/z + 6 = 0$$
$$f_2 = 6\ \ln(1/z^2)/t + 6 = 0$$

4. For the system of equations in problem 1, apply Broyden's method to find the solution with the starting point given in 1a Use the update formula for the inverse of the jacobian and estimate the initial jacobian J^0 by using a finite difference perturbation of 10^{-3}. The criterion for convergence is the same as in problem 1.

5. Research the MINPACK library on NETLIB and describe which subroutines would be appropriate to solve problems 1a,b and 4.

6. Derive the quadratic rule for stepsize adjustment (α_q) that is used in step d. of the Armijo linesearch.

7. For Broyden's method:

a) Assume that B^0 is symmetric. Derive a symmetric Broyden updating formula of the form: $B^{k+1} = B^k + u\,u^T$ which satisfies the secant relation.

b) Derive the analogous symmetric inverse update formula without using B^k in the final formula.

c) Verify that $B^{k+1} = B^k + (y - B^k s)c^T/c^T s$ satisfies the secant relation for an arbitrary vector c.

7

Solution Hwk 2

1) $$f_1 = 2x_1^2 + x_2^2 - 6 = 0$$
$$f_2 = x_1 + 2x_2 - 3.5 = 0$$

a) Newton's Method

$$x^{k+1} = x^k - (J^k)^{-1} f(x^k)$$

$$J^k = \begin{bmatrix} 4x_1 & 2x_2 \\ 1 & 2 \end{bmatrix} \qquad J^{-1} = \frac{1}{8x_1 - 2x_2} \begin{bmatrix} 2 & -2x_2 \\ -1 & 4x_1 \end{bmatrix}$$

$$x_1^{k+1} = x_1^k - \left\{ \frac{2f_1^k - 2x_2 f_2^k}{8x_1^k - 2x_2^k} \right\}$$

$$x_2^{k+1} = x_2^k - \left\{ \frac{-f_1^k + 4x_1^k f_2^k}{8x_1^k - 2x_2^k} \right\}$$

Method converges from $x^0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ in 3 iterations to $x^* = \begin{bmatrix} 1.595865 \\ 0.95207 \end{bmatrix}$

b) from $x^0 = 0$, $J^0 = \begin{bmatrix} 0 & 0 \\ 1 & 2 \end{bmatrix}$ (singular)

$$f^0 = \begin{bmatrix} -6 \\ -3.5 \end{bmatrix} \quad p^{sd} = -J^T f = \begin{bmatrix} 3.5 \\ 7 \end{bmatrix}$$

$$\beta = \frac{\| p_{sd} \|^2}{\| J p_{sd} \|^2} = 0.2 \qquad x^1 = x^0 + \beta p^{sd} = \begin{bmatrix} 0.7 \\ 1.4 \end{bmatrix}$$

1st step gives $\alpha = 0.3$. All others $\alpha = 1$. Converges in 4 iterations

```
      implicit none
      double precision h(2,2), s(2), y(2), x(2), f(2),
     u shy, hs(2), hy(2), f0(2), ff
      integer i, j, n, jj

      data x/2.d0, 1.d0/, n/2/

      data h/0.1427d0, -0.071427, -0.1427, 0.57143/


      f(1) = 2.d0*x(1)**2.d0 + x(2)**2.d0 - 6.d0
      f(2) = x(1) + 2.d0*x(2).- 3.5d0

      do jj = 1,10
      write(6,*) ' iteration ' , jj

 10   do i = 1,n
         s(i) = 0.d0
         do j = 1,n
         s(i) = s(i) -(h(i,j))*f(j)
         enddo
      x(i) = x(i) + s(i)
      f0(i) = f(i)
      enddo

      f(1) = 2.d0*x(1)**2.d0 + x(2)**2.d0 - 6.d0
      f(2) = x(1) + 2.d0*x(2) - 3.5d0

      ff= 0.d0
      do i = 1,n
         ff = ff + f(i)*f(i)
         y(i) = f(i) - f0(i)
         hs(i) = 0.d0
         do j = 1,n
            hs(i) = hs(i) + h(i,j)*s(j)
         enddo
         enddo


       shy = 0.d0

      do i = 1,n
         hy(i) = 0.d0
         do j = 1,n
            hy(i) = hy(i) + h(i,j)*y(j)
         enddo
            shy = shy + s(i)*hy(i)
         enddo

         do i = 1,n
            do j = 1,n
               h(i,j) = h(i,j) + (s(i)-hy(i))*hs(j)/shy
               enddo
               enddo

         write(6,*) ' f = ', f
         write(6,*) ' ff = ', ff
         write(6,*) ' x = ', x
         write(6,*) ' s = ', s
         write(6,*) ' y = ', y
         write(6,*) ' h = ', h

         if(ff.lt.1.d-8) stop

         enddo
         end
```

```
 iteration    1
 f =     0.26277594135600     3.8200000000010D-04
 ff =      6.9051341279532D-02
 x =      1.6432500000000    0.92856600000000
 s =    -0.35675000000000    -7.1434000000000D-02
 y =     -2.7372240586440   -0.49961800000000
 h =     0.15536822922146    -7.7703329240747D-02  -0.13792764123459
   0.56906558944640
 iteration    2
 f =      3.6015620481194D-02     1.0020533595334D-05
 ff =      1.2971250190565D-03
 x =      1.6024756556685    0.94876718243256
 s =     -4.0774344331533D-02     2.0201182432564D-02
 y =    -0.22676032087481    -3.7197946640477D-04
 h =     0.18393726835310    -9.1965791162658D-02  -0.18385721844147
   0.59199491664513
 iteration    3
 f =    -6.3286178078492D-05    -2.0611469020437D-07
 ff =      4.0051828190481D-09
 x =      1.5958528831665    0.95207345535938
 s =    -6.6227725019186D-03     3.3062729268165D-03
 y =     -3.6078906659273D-02    -1.0226648285538D-05
 h =     0.18355125932165    -9.1776221664718D-02  -0.18331471606090
   0.59172849307589
```

$$f(x) = Ax + b = 0$$
$$x^* = -A^{-1}b$$

<u>Newton's method</u>

$$x^1 = x^0 - J^{-1}f(x^0)$$
$$= x^0 - A^{-1}(Ax^0 + b)$$
$$x^1 = x^0 - x^0 - A^{-1}b = x^*$$

solves in one iteration

<u>Broyden's method</u> - this is much more complicated

- if $n=1$, then secant method will converge in one iteration
- if $B^0 = A$ then the result is same as Newton.
- if $B^0 \neq A$ then there is a long complex analysis by Gay (1979, SIAM J. Num. Anal, 16, 4, p. 623) that shows of lin. indep steps, Broyden converges in $\leq 2n$ iterations for linear system.

Reformulate poles

$$f_1 = \frac{\exp\left(\frac{x}{(y^2-6)}\right)}{z} + 6 = 0$$

$$f_2 = 6 \ln(1/z^2)/t + 6 = 0$$

multiply by denominators

$$\exp\left(\frac{x}{y^2-6}\right) + 6z = 0$$

$$6 \ln(1/z^2) + 6t = 0$$

replace terms that can be undefined.

$$\left.\begin{array}{l} x - v(y^2-6) = 0 \\ w z^2 - 1 = 0 \\ \exp u - w = 0 \end{array}\right\} \text{introduce 3 variables \& eqns.}$$

$$\exp(v) + 6z = 0$$

$$6u + 6t = 0$$

) Apply Broyden's method to

$$2x_1^2 + x_2^2 - 6 = 0 \qquad x^0 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$
$$x_1 + 2x_2 - 3.5 = 0$$

$$H^{k+1} = H^k + \frac{(s - H^k y) s^T H^k}{s^T H y}$$

$$B^0 = \begin{bmatrix} 8 & 2 \\ 1 & 2 \end{bmatrix} \qquad H^0 = \begin{bmatrix} 1/7 & -1/7 \\ -1/14 & 4/7 \end{bmatrix}$$

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

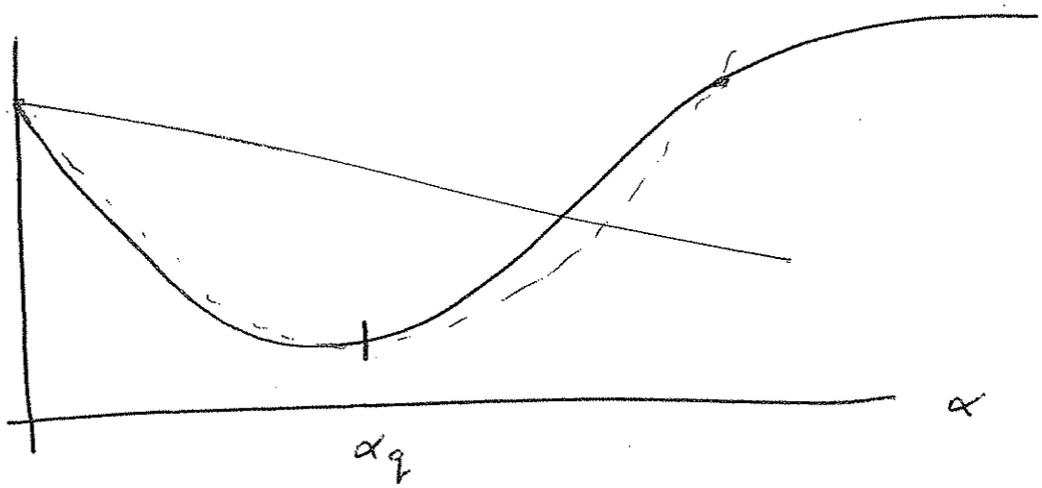
see attached file -

converges in 3 iterations

to $x^* = \begin{bmatrix} 1.5958 \\ 0.95207 \end{bmatrix}$

5) Research MINPACK routines ...

Quadratic derivation - Let $\phi(\alpha) = a\alpha^2 + b\alpha + c$



$$\phi_0 = c$$

$$\phi_\alpha = a\alpha^2 + b\alpha + c$$

$$\phi_0' = b = -2\phi_0 \text{ for Newton}$$

$$a = \left(\phi_\alpha - \phi_0'\alpha - \phi_0\right)/\alpha^2 = \frac{(\phi_\alpha - \phi_0(1-2\alpha))}{\alpha^2}$$

$$\alpha_q = \frac{-b}{2a} = \frac{2\phi_0\alpha^2}{2(\phi_\alpha - \phi_0(1-2\alpha))} = \frac{\phi_0\alpha^2}{-\phi_0(1-2\alpha)+\phi_\alpha}$$

7.

a) $B^{k+1} = B^k + uu^T$

$B^{k+1}s = B^k s + uu^T s = y$

$u = \frac{y - B^k s}{u^T s} = \frac{y - B^k s}{k}$

$\left(B^{k+1} = B^k + \frac{(y - B^k s)(y - B^k s)^T}{k^2} \right) s = y$

$k^2 = (y - B^k s)^T s$

$B^{k+1} = B^k + uu^T = B^k + \frac{(y - B^k s)(y - B^k s)^T}{(y - B^k s)^T s}$

b) SMW formula

$(A + xy^T)^{-1} = A^{-1} - \frac{A^{-1} x y^T A^{-1}}{1 + y^T A^{-1} x}$

$(B^k + uu^T)^{-1} = (B^k)^{-1} - \frac{(B^k)^{-1} u u^T B^{k^{-1}}}{1 + u^T B^{k^{-1}} u}$

$H^{k+1} = (B^{k+1})^{-1} = H^k - \frac{H^k u u^T H^k}{1 + u^T H^k u}$

$u = \frac{(y - B^k s)}{((y - Bs)^T s)^{1/2}}$

$H^{k+1} = H^k - \frac{H^k (y - B^k s)(y - B^k s)^T H^k}{(y - Bs)^T s + (y - B^k s)^T H^k (y - B^k s)}$

$$(y - B^k s)^T H^k = y^T H^k - s^T \cancel{(B^k)^T H^k}$$
$$= y^T H^k - s^T = (H^k y - s)^T$$

this follows only because B & s are symmetric

<u>also</u> $y^T s - s^T B s + (y - B^k s)^T H^k (y - B^k s)$

$$= y^T s - s^T B s + (y - B^k s)^T (H y - s)$$
$$= (y - B^k s)^T H y = y^T H y - s^T y$$

Substituting leads to:

$$H^{k+1} = H^k - \frac{(H^k y - s)(H^k y - s)^T}{(H y - s)^T y}$$

c) $$B^{k+1} = B^k + \frac{(y - B^k s) c^T}{c^T s}$$

$$B^{k+1} s = y = B^k s + (y - B^k s) \frac{\cancel{c^T s}}{\cancel{c^T s}}$$