HW4

Yilun Zhang

October 26, 2015

Exercise 1

Method 1, let $f(x) = dx - 1$, so the solution of $f(x) = 0$ is $1/d$. The matlab code is

```
function [x l] = newroot(x0,a, maxiter, tol)
fold = a*x0-1;
fnew = fold-1;
x=x0;
iter=0;
while( abs(fnew-fold)>tol && iter< maxiter)
fold = fnew;
iter=iter+1;
x=x-(a*x-1)/a;
fnew=a*x+1;
end
l=iter;
end
```

Start value of both 0.3 and 1 both converge in 1 step, this is because

$$x_{k+1} = x_k - \frac{ex_k - 1}{e} = \frac{1}{e}$$

Method 2, let $g(x) = 1/x - d$, the update steps is

$$x_{k+1} = x_k + (1/x_k - d)x_k^2 = 2x_k - dx_k^2$$

Then if the start value is 0.3 it will converge to the true value, if the start value is 1, it will not converge.

Exercise 2

a

$$\begin{aligned} x_{k+1} &= x_k - \frac{x_k^q}{q x_k^{q-1}} \\ &= x_k(1 - \frac{1}{q}) \end{aligned}$$

Therefore, it's Q-linearly converge.

b

Let $f(x) = ||x||_2^\beta$

$$\nabla f = \beta ||x||_2^{\beta-2} x$$

$$\nabla^2 f = \beta ||x||_2^{\beta-4} (||x||_2^2 I + (\beta-2) x x')$$

Because $(A + CBC')^{-1} = A^{-1} - A^{-1}C(B^{-1} + C'A^{-1}C)^{-1}C'A^{-1}$

$$(\nabla^2 f)^{-1} = \frac{1}{\beta ||x||_2^{\beta-2}} (I - \frac{\beta-2}{\beta-1} \frac{1}{||x||_2^2} x x')$$

The pure Newton method:

$$\begin{aligned} x_{k+1} &= x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k) \\ &= x_k - \frac{1}{\beta ||x_k||_2^{\beta-2}} (I - \frac{\beta-2}{\beta-1} \frac{1}{||x_k||_2^2} x_k x_k') \beta ||x_k||_2^{\beta-2} x_k \\ &= \frac{\beta-2}{\beta-1} x_k \end{aligned}$$

Then, x_k converges Q-linearly when $\beta > 3/2$ globally. When $\beta = 2$, it converges in 1 step. When $\beta \leq 1$, it will not converge for start point $x_0 \neq 0$.

c

The Newton method:

$$\begin{aligned} x_{k+1} &= x_k - \alpha_k (\nabla^2 f(x_k))^{-1} \nabla f(x_k) \\ &= x_k - \alpha_k \frac{1}{\beta ||x_k||_2^{\beta-2}} (I - \frac{\beta-2}{\beta-1} \frac{1}{||x_k||_2^2} x_k x_k') \beta ||x_k||_2^{\beta-2} x_k \\ &= (1 - \frac{\alpha_k}{\beta-1}) x_k \end{aligned}$$

The Armijo condition: $f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k [\nabla f(x_k)] p_k$. This implies

$$\left[1 - |1 - \frac{\alpha_k}{\beta-1}|^{\beta}\right] ||x_k||_2^{\beta} \leq c_1 \alpha_k \frac{\beta}{\beta-1} ||x_k||_2^{\beta}$$

as long as $||x_k||_2$ is not zero, we can see α_k doesn't dependent on k, then let $\alpha_k \equiv \alpha$ where α satisfies

$$\left[1 - |1 - \frac{\alpha_k}{\beta-1}|^{\beta}\right] \leq c_1 \alpha_k \frac{\beta}{\beta-1}$$

Then, x_k converges Q-linearly when $|1 - \frac{\alpha}{\beta-1}| < 1$ globally. When $\frac{\alpha}{\beta-1} = 1$, it converges in 1 step. When $\beta \leq 1$, it will not converge for start point $x_0 \neq 0$, because $|1 - \frac{\alpha_k}{\beta-1}|$ always larger than 1. when $\alpha > 0$ and $\beta > 1$, it will Q- linearly converge.

Excercise 3

Prove this by induction.

For $k = 0, i = 0$, need to show $D^1 q^0 = p^0$

$$D^1 q^0 = (D^0 + \frac{y^0 y^{0'}}{q^{0'} y^0}) q^0 = D^0 q^0 + \frac{y^0 y^{0'} q^0}{q^{0'} y^0} = p^0$$

Suppose k-1 hold,

$$D^{k+1} q^i = (D^k + \frac{y^k y^{k'}}{q^{k'} y^k}) q^i = D^k q^i + \frac{y^k y^{k'} q^k}{q^{k'} y^k} = p^i$$

if $i < k$

$$LHS = D^k q^i + \frac{y^k y^{k'} q^k}{q^{k'} y^k} = p^i + \frac{1}{q^{k'} y^k} (p^{k'} q_i - q^{k'} p^i)$$

since $Qp^i = q^i$, and Q is positive definit thus invertible. $p^{k'} q_i - q^{k'} p^i = 0$. So the LHS=$p^i$

if $i = k$

$$D^{k+1}q^k = (D^k + \frac{y^k y^{k'}}{q^{k'} y^k})q^k = D^k q^k + \frac{y^k y^{k'} q^k}{q^{k'} y^k} = p^k$$

therefore , we have $D^{k+1}q^i = p^i$

Since $D^n q^i = p^i$ and $Q^{-1}q^i = p^i$ for $i = 0, 1, \cdots, n-1$. Let $X = [q^0, q^1, \cdots, q^{n-1}]$. Because $\{q^i\}$ are linearly independent. X is full rank. $(D^n - Q^{-1})X = 0$ implies $D^n = Q^{-1}$.

Exercide 4

I already implemented BFGS in last homework, the matlab code:

```
function [x,fc,itc] = newton(obj,i,maxit,tol,qusi,eps)
c=0.0001;
x0=p00_start(i,p00_n(i));
n=p00_n(i);
[fc,gc,hc]=obj(i,x0);
[P,D]=eig(hc);
if(min(diag(D))<eps)
D=max(D,eps*eye(n));
hc=P*D*P';
end
g0=gc;
xc=x0;
itc=1;
H=inv(hc);
while(norm(gc) > tol*norm(g0) & itc <= maxit)
p=-H*gc;
alpha=1.0; xt=xc+alpha*p; ft=obj(i,xt);
fg= fc + c*alpha*(p'*gc);
cout=1;
while(ft > fg) % check Armijo condition
alpha=alpha*0.9;
```

```
fg= fc + c*alpha*(gc'*p);
xt=xc+alpha*p;
ft=obj(i,xt);
cout=cout+1;
if(cout>20)
break
end
end
xc=xt;
go=gc;
[fc,gc,hc]=obj(i,xc);
itc=itc+1;
if(qusi)
s=alpha*p;
y=gc-go;
pho=1/(y'*s);
H=(eye(n)-pho*s*y')*H*(eye(n)-pho*y*s')+pho*s*s';

else
[P,D]=eig(hc);
if(min(diag(D))<eps)
D=diag(D);
D(D<=1e-8)=eps;
D=diag(D);
hc=P*D*P'
H=inv(hc);
end
end
end
x=xc;
```

prob ID	f^*	number of iter
1	4.46500818473627	20001
2	0.242677406804875	412
3	1.12793276962393e-08	4
4	0.0032230846252832	20001
5	3.97089247824700e-14	6
6	6.37118858804940e-12	45
7	0.00228767657879530	3076
8	2.49997654797500e-06	59
9	0.489395214700814	16
10	25364766502.5965	20001
11	85822.2019034065	251
12	8.39437213309053e-20	13
13	8.42718143579277e-16	13
14	4.93038065763132e-32	2
15	1.63898347747199e-06	14859
16	1.81352665939531e-08	20001
17	7.79452679362206	20001
18	5.93799296903267e-14	13

Table 1: result of Newton method

```
end
```

Table 1 is the result of pure Newton method(with eigen value correction), Table 2 shows result of quasi newton.

We can see all problems converge but number 2, 5, 10, 16 return a NA. I guess this is because $\rho = 1/y's$, when y and s are both very small, it will cause overflow in calculation in Matlab.

prob ID	f^*	number of iter
1	4.75669872116614e-07	36
2	NaN	3
3	1.12793283165719e-08	3
4	8.23648565011059e-10	203
5	NaN	3
6	3.52702079374637e-18	7
7	0.00228767029059872	20
8	2.49997507553623e-06	8
9	0.489395214700777	8
10	NaN	8
11	85822.2162472798	18
12	32.8349996345640	7
13	1.80370537607437e-13	5
14	4.93038065763132e-32	2
15	1.67963844460425e-05	15
16	NaN	5
17	7.87479283486014	13
18	0.444444444430489	203

Table 2: result of quasi Newton method