Suggestion Answer for Homework 3, CS421

October 24, 1999

1.

```
function value=norm_1(A)
[ m,n ] = size(A);
value = 0;
for j = 1:n
    sum = 0;
    for i = 1:m
       sum = sum + abs(A(i,j));
    end
    if sum > value
       value = sum;
    end
end
```

```
function value=norm_inf(A)
[m,n] = size(A);
value = 0;
for i = 1:m
    sum = 0;
    for j = 1:n
       sum = sum + abs(A(i,j));
    end
    if sum > value
       value = sum;
    end
end
```

test for 1 norm

size	My result	Matlab result	error
2 * 3	1.719548e+00	1.719548e+00	0.000000e+00
3 * 4	2.244618e+00	2.244618e+00	0.000000e+00
4 * 5	2.458338e+00	2.458338e+00	0.000000e+00
5 * 6	3.047336e+00	3.047336e+00	0.000000e+00
6 * 7	4.096330e+00	4.096330e+00	0.000000e+00

test for ∞ norm

size	My result	Matlab result	error
2 * 3	1.742419e+00	1.742419e+00	0.000000e+00
3 * 4	2.502426e+00	2.502426e+00	0.000000e+00
4 * 5	2.752163e+00	2.752163e+00	0.000000e+00
5 * 6	3.935501e+00	3.935501e+00	0.000000e+00
6 * 7	4.754859e+00	4.754859e+00	0.000000e+00

2. See the reference paper.

3. The details about the algorithm is on *Matrix Computations*, Van Loan.

```
function BL = bchol(BA)
[m,n] = size (BA);
beta = m-1;
for j = 1:n
   for k = max(1,j-beta):j-1
      lambda = min(k+beta, n);
      BA(1:lambda-j+1,j) = BA(1:lambda-j+1,j) - BA(j-k+1,k)*BA(j-k+1:lambda-k+1,k);
   end
   lambda = min(j+beta,n);
   BA(1,j)= sqrt(BA(1,j));
   BA(2:lambda-j+1,j) = BA(2:lambda-j+1,j)/BA(1,j);
end
BL = BA;


function x = LTriSol(L,b)
[m,n] = size(L);
beta = m-1;
x = zeros(n,1);
for j = 1:n-1
   x(j) = b(j) / L(1,j);
   for i = j+1 : min(j+beta, n)
      b(i) = b(i) - L(i-j+1,j) * x(j);
   end
```

```
end
x(n) = b(n) / L(1,n);


function x = UTriSol(U,b);
[m,n] = size(U);
beta = m-1;
x = zeros(n,1);
for j = n:-1:2
   x(j) = b(j) / U(beta+1 ,j );
   for i = max(1,j-beta):j-1
      b(i) = b(i) - U(i-j+beta+1,j) * x(j);
   end
end
x(1) = b(1) / U(beta+1,1);
```

n	beta	residue
20	0	1.4358e-18
20	1	7.3018e-18
20	3	1.4406e-17
20	5	2.4985e-17
20	8	3.8144e-18
20	10	7.1778e-18
20	20	1.2649e-17
25	0	1.0454e-19
25	1	4.4253e-17
25	3	1.0002e-17
25	5	7.6069e-18
25	8	6.5694e-18
25	10	7.0866e-18
25	25	3.1254e-18

flops $= n(beta)^2 + O(n * beta)$.

4(a).
Let $v = \alpha s + w$ where α is a scalar and $s^T w = 0$. Then

$$\alpha = r/s^T s$$

Becasue $s^T w = 0$,

$$\|v\|_2^2 = (\alpha s + w)^T(\alpha s + w) = \alpha^2 s^T s + w^T w$$

The minimal value of $\|v\|_2^2$ can be achieved when $w = 0$. Thus the solution for the original problem is $v = \frac{rs}{s^T s}$.

(b)
Let

$$\begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

where E_i, $i = 1, 2, \cdots, n$ are row vectors of E. Then

$$\min \|E\|_F^2 = \min(\sum_{i=1}^{n} \|E_i\|_2^2) = \sum_{i=1}^{n}(\min(\|E_i\|_2^2).$$

From (a) ,

$$E_i = \frac{r_i s^T}{s^T s}$$

where r_i is ith component of r. Thus

$$E = \frac{r s^T}{s^T s}.$$

Let $y_k - B_k s_k = r$, the Broyden updates is exactly the solution of our problem.