# Suggestion Answer for Homework 2, CS421

October 4, 1999

1. (a)

$$f(\alpha)^2 = (x - \alpha y)^T (x - \alpha y) = x^T x - 2\alpha x^T y + \alpha^2 y^T y$$

When $\alpha = \frac{x^T y}{y^T y}$, $f(\alpha)^2$ achieve the minimum value.

(b) Because $rank(A) \leq m < n$, the column vectors $A(:,1)$, $A(:,2)$, ..., $A(:,n)$ are linearly dependent. Thus there exists $c_1$, $c_2$, ..., $c_n$ where $c_i$ $i = 1, 2, ..., n$ are not all zeros, s.t.

$$\sum_{i=1}^{n} c_i A(:,i) = 0$$

Thus there exists nontrival solution for the system $Ax = 0$.

2.

```
matrix_size = [1, 2, 5, 10, 15, 20];
for s = 1: size(matrix_size,2)
   n = matrix_size(s);
   A = rand(n,n) + 0.5*eye(n,n);
   T = triu(A);
   b = rand(n,1);
   y = T\b;
   x = zeros(n,1);
   for i = n:-1:1
      x(i) = b(i)/T(i,i);
      b(1:i) = b(1:i) - x(i) * T(1:i,i);
   end
   error = norm(x - y);
   disp(sprintf('n = %4i',n))
   disp(sprintf('error = %13.6e', error))
end
```

3.

```
matrix_size = [1, 2, 5, 10, 15, 20];
for s = 1: size(matrix_size,2)
```

```
    n = matrix_size(s);
    A = rand(n,n) + 0.5*eye(n,n);
    T = tril(A);
    b = rand(n,1);
    y = T\b;
    x = zeros(n,1);
    for i = 1:n
       x(i) = b(i)/T(i,i);
       b(i:n) = b(i:n) - x(i) * T(i:n,i);
    end
    error = norm(x − y);
    disp(sprintf('n = %4i',n))
    disp(sprintf('error = %13.6e', error))
end
```

4.

```
function [L, U, P] = ppLU(A)
N = size(A,1);
P = 1:N;
for k = 1:N
    [pivot,index] = max(abs(A(k : N, k)));
    index = index + k - 1;
    temp = A(k,:);
    A(k,:)= A(index,:);
    A(index,:) = temp;
    temp = P(k);
    P(k) = P(index);
    P(index) = temp;
    if abs(A(k,k)) ¡ 1.0e-07
       disp('error in LU factorization')
    end
    A(k+1:N,k) = A(k+1:N,k)/A(k,k);
    A(k+1:N,k+1:N) = A(k+1:N,k+1:N) - A(k+1:N,k)*A(k,k+1:N);
end
L = tril(A);
U = triu(A);
for k = 1:N
    L(k,k) = 1;
end
```

```
matrix_size = [1, 2, 5, 10, 15, 20];
for s = 1: size(matrix_size,2)
    n = matrix_size(s);
```

$A$ = rand(n,n);
$b$ = rand(n,1);
$y = A\backslash b$;
$[L, U, P]$ = ppLU($A$);
$x = L\backslash b(P)$;
$x = U\backslash x$;
error = norm($x - y$);
residu1 = norm($A * x - b$) / (norm($A$) * norm($x$));
residu2 = norm($A * y - b$) / (norm($A$) * norm($y$));
disp(sprintf('n = %4i', n))
disp(sprintf('norm($x - y$) = %13.6e', error));
disp(sprintf('residual of backslash = %13.6e', residu1))
disp(sprintf('residual of my algorithm = %13.6e', residu2))
end

n = 1
norm(x-y) = 0.000000e+00
residual of backslash = 0.000000e+00
residual of my algorithm = 0.000000e+00
n = 2
norm(x-y) = 0.000000e+00
residual of backslash = 6.086617e-17
residual of my algorithm = 6.086617e-17
n = 5
norm(x-y) = 0.000000e+00
residual of backslash = 4.506300e-17
residual of my algorithm = 4.506300e-17
n = 10
norm(x-y) = 2.108145e-15
residual of backslash = 5.309895e-17
residual of my algorithm = 5.590941e-17
n = 15
norm(x-y) = 3.701467e-12
residual of backslash = 2.537621e-17
residual of my algorithm = 4.858496e-17
n = 20
norm(x-y) = 1.485829e-14
residual of backslash = 3.716460e-17
residual of my algorithm = 3.413342e-17

5.

$matrix\_size = [5, 10, 15, 20]$;
for s = 1: size($matrix\_size$,2)

```
    n = matrix_size(s);
    A = hilb(n);
    b = rand(n,1);
    y = A\b;
    [L, U, P] = ppLU(A);
    x = L\b(P);
    x = U\x;
    error = norm(x - y);
    cdnA = cond(A);
    residu1 = norm(A * x - b) / (norm(A) * norm(x));
    residu2 = norm(A * y - b) / (norm(A) * norm(y));
    disp(sprintf('n = % 4i', n))
    disp(sprintf('condition number = % 13.6e',cdnA))
    disp(sprintf('norm(x - y) = % 13.6e', error));
    disp(sprintf('residual of backslash = % 13.6e', residu1))
    disp(sprintf('residual of my algorithm = % 13.6e', residu2))
  end
```

```
n = 5
condition number = 4.766073e+05
norm(x-y) = 1.332620e-07
residual of backslash = 1.032944e-17
residual of my algorithm = 1.492218e-17
error in LU factorization
error in LU factorization
error in LU factorization
n = 10
condition number = 1.602442e+13
norm(x-y) = 1.457386e+08
residual of backslash = 1.720703e-17
residual of my algorithm = 1.022058e-17
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 2.038014e-18.
> In /home/brew/hech/TA/hw2/sc5.m at line 6
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
n = 15
condition number = 2.285273e+17
```

norm(x-y) = 4.758684e+15
residual of backslash = 9.990620e-18
residual of my algorithm = 8.232010e-18
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 3.557198e-19.
> In /home/brew/hech/TA/hw2/sc5.m at line 6
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 5.633360e-17.
> In /home/brew/hech/TA/hw2/sc5.m at line 9
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
error in LU factorization
n = 20
condition number = 8.491696e+17
norm(x-y) = 3.442156e+16
residual of backslash = 9.807539e-18
residual of my algorithm = 8.859259e-18

The system warnings indicate that the solution solved by 'backslash' is inaccurate. There are also warnings on our algorithm which indicate that our matrix is near singular. So both of the two solutions are inaccurate. Although the norm of residual is very small, it does not mean the norm of $A * x - b$ is small. Furthermore, we can find the big norm of $x$ causes the small residual norm. Thus we draw the conclusion that both the 'backslash' and our algorithm are not fit for solving hilbert system.

6. (a) Because

$$C^T = (A^T A)^T = A^T A = C$$

$\Longrightarrow$ C is symmetric.

$\forall x \in R^n$,

$$x^T C x = x^T A^T A x = y^T y \geq 0$$

where $y = Ax$.

If $x^T C x = 0$, then $Ax = 0$. Because $rank(A) = n$, the column vectors of A are linearly independent. Thus

$$x_1 A(:,1) + \cdots + x_n A(:,n) = 0$$

only has trival solution where $x_1, \cdots, x_n$ are components of $x$.
$\Longrightarrow$ $x = 0$

(b)

```
function L=mychol(A)
N = size(A, 1);
if N == 1
   L = sqrt(A(1, 1));
else
   L = mychol(A(1 : N − 1, 1 : N − 1));
   l = L\A(1 : N − 1, N);
   d = sqrt(A(N, N) − l' ∗ l);
   L = [L, zeros(N − 1, 1); l', d];
end
```

```
matrix_size = [5, 10, 15, 20];
for s = 1: size(matrix_size,2)
   n = matrix_size(s);
   A = rand(n,n);
   C = A' ∗ A;
   b = rand(n,1);
   y = C\b;
   L = mychol(C);
   x = L\b;
   x = L'\x;
   error = norm(x − y);
   residu1 = norm(A ∗ x − b) / (norm(A) * norm(x));
   residu2 = norm(A ∗ y − b) / (norm(A) * norm(y));
   disp(sprintf('n = %4i', n))
   disp(sprintf('norm(x − y) = %13.6e', error));
   disp(sprintf('residual of backslash = %13.6e', residu1))
   disp(sprintf('residual of my algorithm = %13.6e', residu2))
end
```

```
n = 5
norm(x-y) = 1.342590e-14
residual of backslash = 9.832714e-02
residual of my algorithm = 9.832714e-02
```

n = 10
norm(x-y) = 1.137322e-10
residual of backslash = 4.951153e-03
residual of my algorithm = 4.951153e-03
n = 15
norm(x-y) = 1.703751e-13
residual of backslash = 3.477350e-02
residual of my algorithm = 3.477350e-02
n = 20
norm(x-y) = 3.918056e-11
residual of backslash = 5.635413e-03
residual of my algorithm = 5.635413e-03