PURDUE UNIVERSITY · CS 51500
MATRIX COMPUTATIONS

HOMEWORK

*Jun Cheng*

*September 25, 2015*

## Problem 0: Homework checklist

✓I didn't talk with any one about this homework.
✓Source-code are included at the end of this document.

## Problem 1:

1. Function backsolve.m

```
function x = backsolve(A, b)
    % Assume matrix A is n by n upper triangular matrix;
    % Vector b is n by 1.
    n = length(b) ;          %% Get the length of the vector b
    x = zeros(n,1);          %% initiate solution Xl
    x(n) = b(n)/A(n,n);      %% For the base case, x(n) = ...
        b(n)/A(n,n);
    for i=n-1:-1:1
        temp = 0;
        for j = i:n
            temp = temp + A(i, j)*x(j);    %% Accumulate all ...
                the other terms
        end
        x(i) = (b(i)-temp)/A(i,i);    %%  solve X(i)
    end
end
```

Function forwardsolve.m:

```
function x = forwardsolve(A, b)
    % Assume matrix A is n by n upper triangular matrix;
    % Vector b is n by 1.
    n = length(b) ;          %% Get the length of the vector b
    x = zeros(n,1);          %% Initiate solution Xl
    x(1) = b(1)/A(1,1);      %% For the base case, x(1) = ...
        b(1)/A(1,1);
    for i=2:n
        temp = 0;
        for j = 1:i
            temp = temp + A(i, j)*x(j); %% Accumulate all ...
                the other terms
        end
        x(i) = (b(i)-temp)/A(i,i);  %%  solve X(i)
    end
end
```

2. I set $n = 1000$ and repeat the procedure for 100 times. The time elapses for Matlab backslash method is around 3.210974 seconds. My backsolve method will spend around 4.733596 seconds. Matlab backslash has better performance than mine.

   For the accuaracy:

$$\frac{\|X_{matlab} - X_{mine}\|}{\|X_{matlab}\|} = 7.2165e - 16 \tag{1}$$

3. My linear solver function:

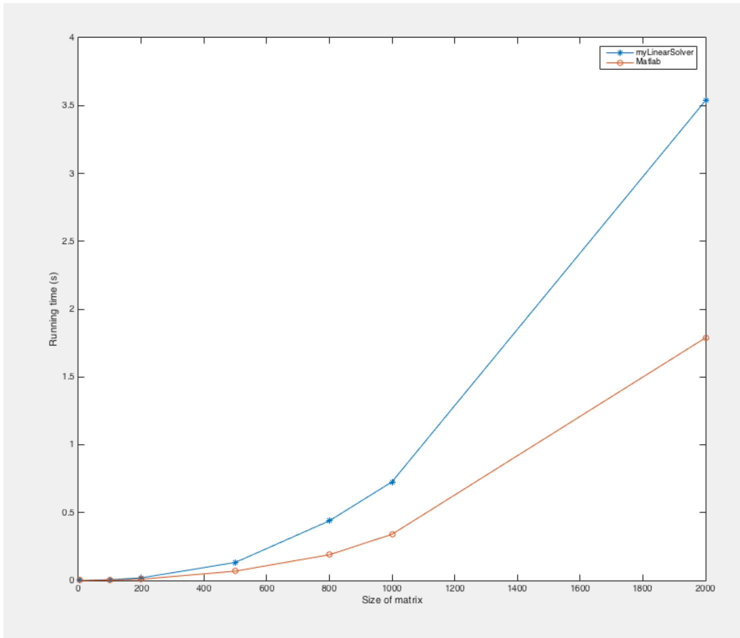


Figure 1: Time consumption for different size of matrix.

```
function x = linearSolver(A, b)
    [L, U, P] = lu(A);
    x1 = forwardsolve(L, P*b);
    x = backsolve(U, x1);
end
```

## Problem 2:

1. Base on

$$\frac{1}{h^2}(u(x_{i-1}, y_j) + u(x_i, y_{j-1}) - 4u(x_i, y_j) + u(x_{i+1}, y_j) + u(x_i, y_{j+1})) = f(x_i, y_j)$$

Then all the linear equations:

$$
\begin{aligned}
(i = 0, j = 0) :& f_{0,0} = 0 \\
(i = 0, j = 1) :& f_{0,1} = 0 \\
(i = 0, j = 2) :& f_{0,2} = 0 \\
(i = 0, j = 3) :& f_{0,3} = 0 \\
(i = 1, j = 0) :& f_{1,0} = 0 \\
(i = 1, j = 1) :& f_{1,1} = 0 \\
(i = 1, j = 2) :& f_{1,2} = 0 \\
(i = 1, j = 3) :& f_{1,3} = 0 \\
(i = 2, j = 0) :& f_{2,0} = 0 \\
(i = 2, j = 1) :& f_{2,1} = 0 \\
(i = 3, j = 0) :& f_{3,0} = 0 \\
(i = 3, j = 1) :& f_{3,1} = 0
\end{aligned}
$$

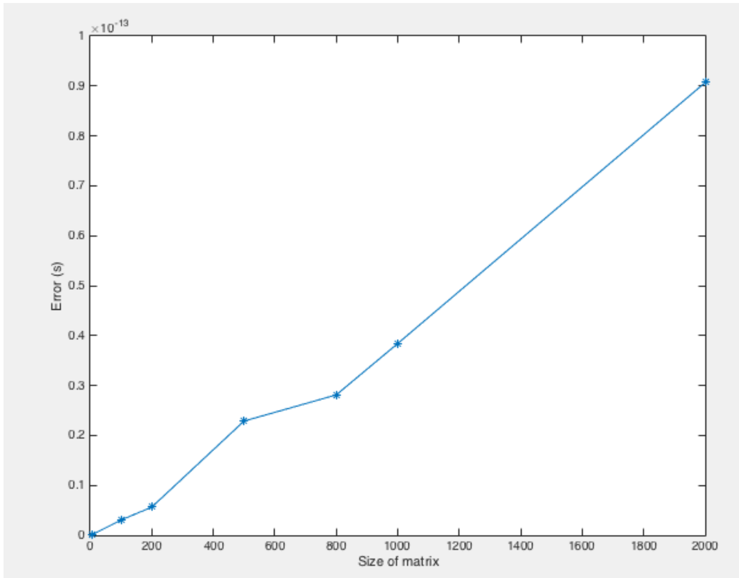


Figure 2: Error plot between Matlab backslash solver and my linear solver.

All the others:

$$(i=2, j=2): f_{2,2} = \frac{1}{h^2}(u(x_1,y_2)+u(x_2,y_1)-4u(x_2,y_2)+u(x_3,y_2)+u(x_2,y_3))$$
$$(i=2, j=3): f_{2,3} = \frac{1}{h^2}(u(x_1,y_3)+u(x_2,y_2)-4u(x_2,y_3)+u(x_3,y_3)+u(x_2,y_3))$$
$$(i=3, j=2): f_{3,2} = \frac{1}{h^2}(u(x_2,y_2)+u(x_3,y_2)-4u(x_3,y_2)+u(x_4,y_2)+u(x_3,y_3))$$
$$(i=3, j=3): f_{3,3} = \frac{1}{h^2}(u(x_2,y_3)+u(x_2,y_2)-4u(x_3,y_3)+u(x_4,y_3)+u(x_3,y_4))$$

2. The code is filled as below:

```
n = 10;
A = zeros((n+1)^2,(n+1)^2);
newA = zeros((n+1)^2-2, (n+1)^2-2);
f = zeros((n+1)^2,1);
G = reshape(1:((n+1)^2), n+1, n+1)';
for i=0:n
 for j=0:n
     row = G(i+1,j+1);
     if i==0 || j == 0 || i==n || j==n
         % we are on a boundary
         f(row) = 0;
         A(row, row) = 1;     % For boundary points, we set ...
             the diagonal term to be 1.

     else
         % we are NOT on a boundary
         f(row) = 1/n^2;
            A(row, G(i,j+1  )) = 1;             % Put the ...
                coefficients to the corresponding entry of ...
                matrix A.
            A(row, G(i+1,  j)) = 1;
            A(row, G(i+1,  j+1  )) = -4;
            A(row, G(i+2,j+1  )) = 1 ;
            A(row, G(i+1,  j+2)) = 1;
        % end

```

```
        end
    end
end
u = A\f;
```

3. Using Matlab's backslash solver, I can get the result for $u$ :

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | -0.0128 | -0.0206 | -0.0254 | -0.0280 | -0.0288 | -0.0280 | -0.0254 | -0.0206 | -0.0128 | 0 |
| 0 | -0.0206 | -0.0343 | -0.0430 | -0.0478 | -0.0493 | -0.0478 | -0.0430 | -0.0343 | -0.0206 | 0 |
| 0 | -0.0254 | -0.0430 | -0.0544 | -0.0608 | -0.0629 | -0.0608 | -0.0544 | -0.0430 | -0.0254 | 0 |
| 0 | -0.0280 | -0.0478 | -0.0608 | -0.0682 | -0.0706 | -0.0682 | -0.0608 | -0.0478 | -0.0280 | 0 |
| 0 | -0.0288 | -0.0493 | -0.0629 | -0.0706 | -0.0731 | -0.0706 | -0.0629 | -0.0493 | -0.0288 | 0 |
| 0 | -0.0280 | -0.0478 | -0.0608 | -0.0682 | -0.0706 | -0.0682 | -0.0608 | -0.0478 | -0.0280 | 0 |
| 0 | -0.0254 | -0.0430 | -0.0544 | -0.0608 | -0.0629 | -0.0608 | -0.0544 | -0.0430 | -0.0254 | 0 |
| 0 | -0.0206 | -0.0343 | -0.0430 | -0.0478 | -0.0493 | -0.0478 | -0.0430 | -0.0343 | -0.0206 | 0 |
| 0 | -0.0128 | -0.0206 | -0.0254 | -0.0280 | -0.0288 | -0.0280 | -0.0254 | -0.0206 | -0.0128 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Problem 3: The Schur Complement

1.

$$\begin{aligned}
\boldsymbol{C}\mathbf{x}_1 + \boldsymbol{D}\mathbf{x}_2 &= \mathbf{b}_2 \\
\boldsymbol{D}\mathbf{x}_2 &= \mathbf{b}_2 - \boldsymbol{C}\mathbf{x}_1 \\
\mathbf{x}_2 &= \boldsymbol{D}^{-1}(\mathbf{b}_2 - \boldsymbol{C}\mathbf{x}_1)
\end{aligned}$$

It is possible when the block matrix $\boldsymbol{D}$ is invertible because we need the inverse of matrix $\boldsymbol{D}$.

2.

$$\begin{aligned}
\boldsymbol{A}\mathbf{x}_1 + \boldsymbol{B}\mathbf{x}_2 &= \mathbf{b}_1 \\
\boldsymbol{C}\mathbf{x}_1 + \boldsymbol{D}\mathbf{x}_2 &= \mathbf{b}_2
\end{aligned}$$

Now we know that $\boldsymbol{D}$ is invertible. So multiply $\boldsymbol{B}\boldsymbol{D}^{-1}$ on the second equation:

$$\begin{aligned}
\boldsymbol{A}\mathbf{x}_1 + \boldsymbol{B}\mathbf{x}_2 &= \mathbf{b}_1 \\
\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\mathbf{x}_1 + \boldsymbol{B}\mathbf{x}_2 &= \boldsymbol{B}\boldsymbol{D}^{-1}\mathbf{b}_2 \\
(\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} - \boldsymbol{A})\mathbf{x}_1 &= (\boldsymbol{B}\boldsymbol{D}^{-1}\mathbf{b}_2 - \mathbf{b}_1)
\end{aligned}$$

So if we can get inverse of $\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} - \boldsymbol{A}$, then we can solve $\mathbf{x}_1$.

3. Then we can solve

$$\mathbf{x}_1 = (\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C} - \boldsymbol{A})^{-1}(\boldsymbol{B}\boldsymbol{D}^{-1}\mathbf{b}_2 - \mathbf{b}_1)$$

## Problem 4:Ranking with Linear Systems

Questions:
I am a little confuse why we need $r_1, r_2, ..., r_n$ to be the indicate the performance of each team, instead of $S_1, S_2, ..., S_n$?

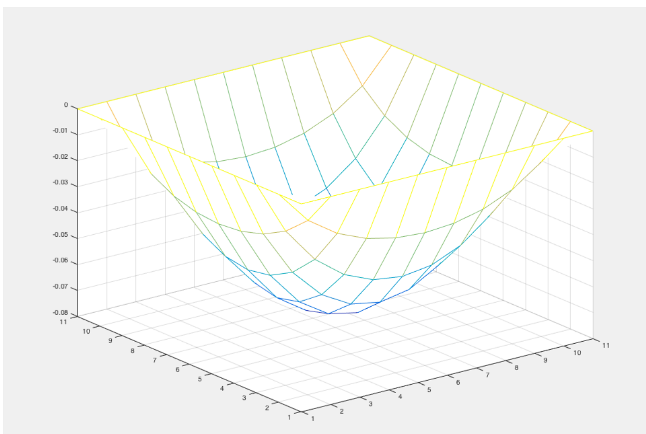


Figure 3: Surface plot of $\mu$.