FEM Homework 4

Shakil Rafi

April 17, 2022

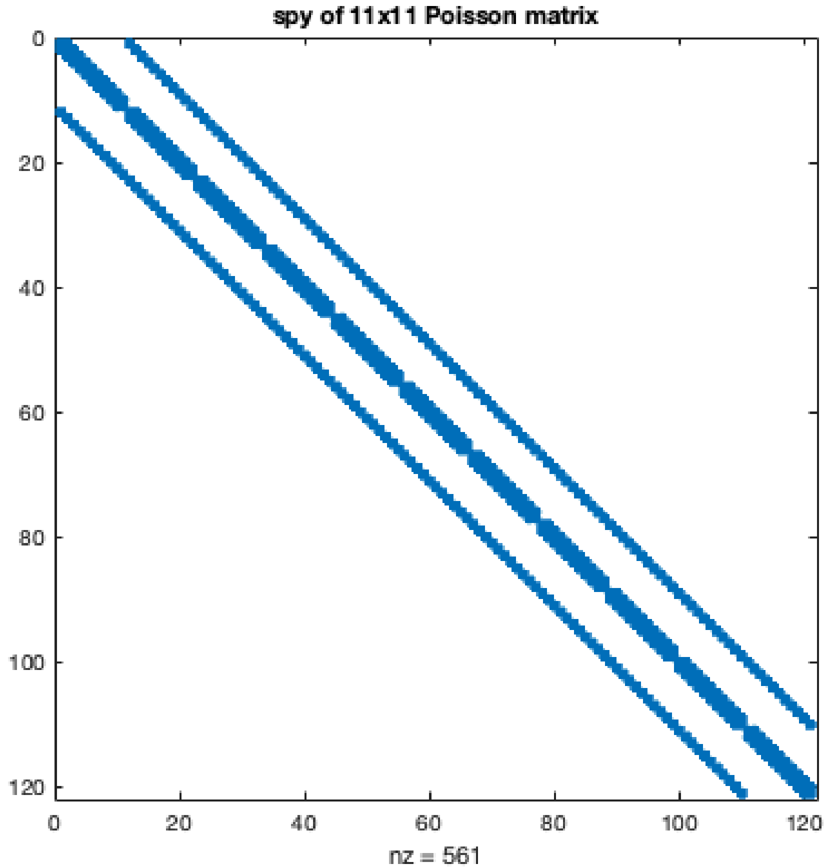
Q1 The MATLAB files are attached below.
Q2 The MATLAB files are attached below.
Q3 The MATLAB files are attached below.
Q3e AMD methods result in a smaller elimination tree than both RCM and the original matrix. AMD is also drastically faster for cholsky, backsub, and forsub.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Finite Element Methods HW4.
% This is the answer to Question 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure();
    A = gallery('poisson',11);
    subplot(1,1,1),spy(A),title('spy of 11x11 Poisson matrix');


methods = ["Jacobi";
    "Block Jacobi";
    "Gauss-Siedel";
    "Block Gauss-Siedel";
    "Symmetric Gauss-Siedel";
    "Block Symmetric Gauss-Siedel";
    "SOR, omega = 1.6";
    "Block SOR, omega = 1.5"];

iterations = [k_J;
    k_BJ;
    k_GS;
    k_BGS;
    k_SGS;
    k_BSGS;
    k_SOR;
    k_BSOR];


for i = [11, 31, 63]

    A = gallery('poisson',i);
    I = eye(size(A,1));
    b = ones(size(A,1),1);
    x = zeros(size(A,1),1);
    tol = 10^-6;

    %Jacobi
    M = diag(diag(A));
    [x_J,k_J] = statit(A,M,[], b, x,tol);
    D = M;

    %Block Jacobi

    M = triu(tril(A,1),-1);
    D_B = M;

    [x_BJ,k_BJ] = statit(A,M,[], b, x,tol);

    %Gauss-Siedel

    M = tril(A);
```

```
    [x_GS,k_GS] = statit(A,M,[], b, x,tol);

    %Block Gauss-Siedel

    M = tril(A,1);

    [x_BGS,k_BGS] = statit(A,M,[], b, x,tol);

    %Symmetric Gauss-Siedel

    M_1 = tril(A)/sqrt(D);
    M_2 = transpose(M_1);
    M = M_1*M_2;

    [x_SGS,k_SGS] = statit(A,M,M_2, b, x,tol);

    %Block symmetric Gauss-Siedel

    M_1 = tril(A,1)/chol(D_B);
    M_2 = transpose(M_1);
    M = M_1*M_2;

    [x_BSGS,k_BSGS] = statit(A,M,M_2, b, x,tol);

    %SOR (omega = 1.6)

    omega = 1.6;
    M = D/omega + tril(A,-1);

    [x_SOR,k_SOR] = statit(A,M,[], b, x,tol);

    %Block SOR (omega = 1.5)

    omega = 1.5;
    M = D_B/omega + tril(A,-3);

    [x_BSOR,k_BSOR] = statit(A,M,[], b, x,tol);

    %Final output
    disp("Iterations for Poisson matrix on an " + i + " by "+i+" grid is:")

    table(methods, iterations)
end
```

```
Iterations for Poisson matrix on an 11 by 11 grid is:

ans =

  8×2 table

             methods               iterations
    ______________________________    __________
```

```
    "Jacobi"                           5000
    "Block Jacobi"                     5000
    "Gauss-Siedel"                     5000
    "Block Gauss-Siedel"               2828
    "Symmetric Gauss-Siedel"           2833
    "Block Symmetric Gauss-Siedel"     1004
    "SOR, omega = 1.6"                 1404
    "Block SOR, omega = 1.5"            937

Iterations for Poisson matrix on an 31 by 31 grid is:

ans =

  8×2 table

               methods                iterations
    ______________________________    __________

    "Jacobi"                           5000
    "Block Jacobi"                     5000
    "Gauss-Siedel"                     5000
    "Block Gauss-Siedel"               2828
    "Symmetric Gauss-Siedel"           2833
    "Block Symmetric Gauss-Siedel"     1004
    "SOR, omega = 1.6"                 1404
    "Block SOR, omega = 1.5"            937

Iterations for Poisson matrix on an 63 by 63 grid is:

ans =

  8×2 table

               methods                iterations
    ______________________________    __________

    "Jacobi"                           5000
    "Block Jacobi"                     5000
    "Gauss-Siedel"                     5000
    "Block Gauss-Siedel"               2828
    "Symmetric Gauss-Siedel"           2833
    "Block Symmetric Gauss-Siedel"     1004
    "SOR, omega = 1.6"                 1404
    "Block SOR, omega = 1.5"            937
```



Published with MATLAB® R2022a

```
function [x,k] = statit(A,M,M2,b,x,tol)
%STATIT Stationary Iteration
%
%       x^{k+1} = x^{k} + M \ r^{k},   r^{k} = b - A x^{k}
%       for solving A x = b
%
%       [x,k] = statit(A,M1,M2,b,x,tol)
%       Input:  A  system matrix
%               M1,M2  M = M1*M2 `preconditioner'
%                      (M2 = [] indicates M2=identity)
%               b  right hand side
%               x  initial vector x^{0} (default x = 0)
%               tol (default tol = eps)
%       Output: x approximate solution
%               k number of iteration until convergence
%       convergence criterion:
%       norm(b - A*x) <= tol*norm(b - A*x0)
% number of function input arguments
if (nargin < 6), tol = eps; end
if (nargin < 5), x = zeros(size(A,1),1); end

r = b - A*x;
rnrm0 = norm(r);  rnrm = rnrm0;
for k=1:5000
   if isempty(M2),
      x = x + M\r;
   else
      x = x + M2\(M\r);
   end
   r = b - A*x;
   rnrm = norm(r);
   if rnrm < tol*rnrm0, return, end
end
```

```
Not enough input arguments.

Error in statit (line 20)
if (nargin < 5), x = zeros(size(A,1),1); end
```

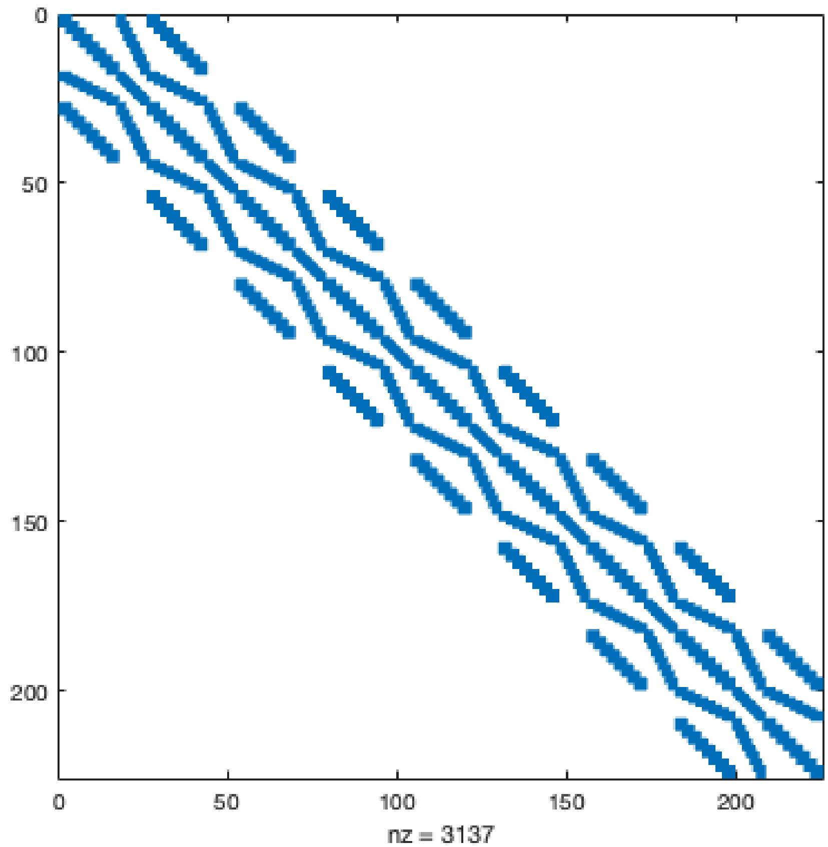
Published with MATLAB® R2022a

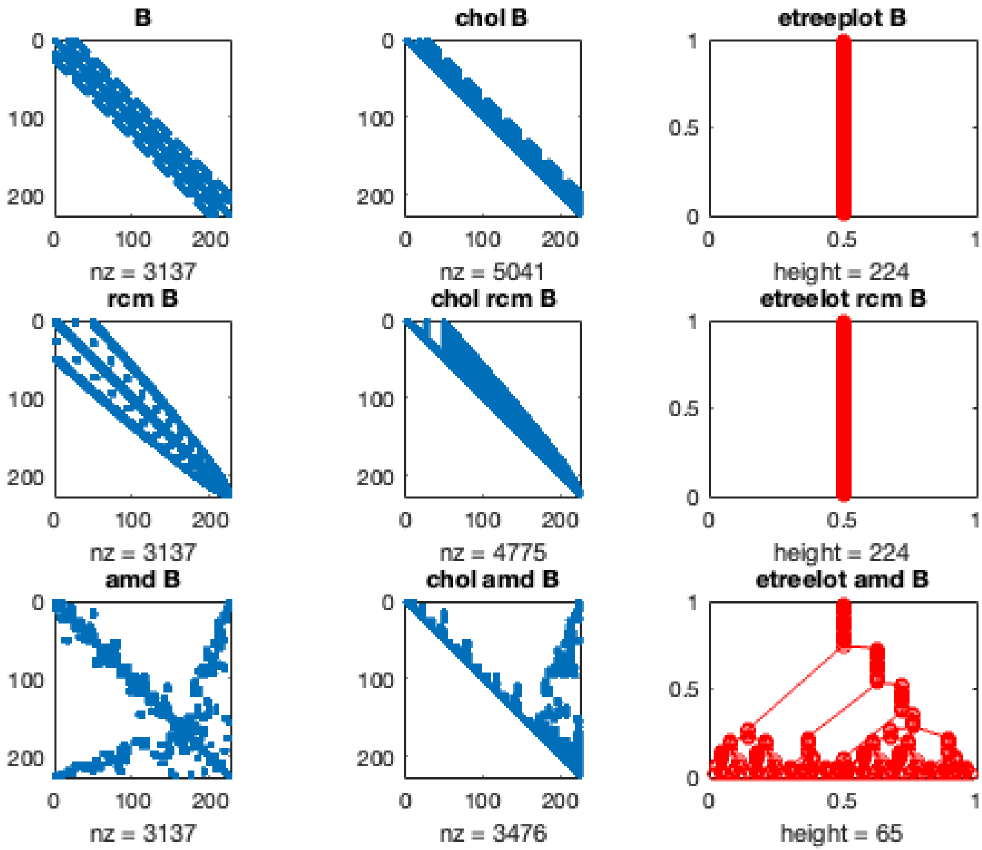
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the answer to question 2 a function for converting
% and mxn matrix into skyline storage, i.e. two vectors
% pointers and values.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [pointers,values] = skylinestorage(A)

m = size(A,1);
pointers = NaN(1, m+1);
values = [];

for i=1:m
    pointers(i) = size(values,2)+1;
    start_point = find(A(i,:),1);   %we sill slice a matrix at row i, from the
 first non-zero elemnt to the diagonal
    end_point = i;
    values = [values, A(i,start_point:end_point)];
end

end
```

```
Not enough input arguments.

Error in skylinestorage (line 10)
m = size(A,1);
```

Published with MATLAB® R2022a

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the first part of question 3
% A Wathen matrix with rcm and amd applied with appropriate etreeplots
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

B = gallery('wathen',8,8);

spy(B);

p = symrcm(B);
rcmB = B(p,p);

p = symamd(B);
amdB = B(p,p);

figure();
    subplot(3,3,1),spy(B),title('B')
    subplot(3,3,2),spy(chol(B)),title('chol B')
    subplot(3,3,3),etreeplot(B),title('etreeplot B')

    subplot(3,3,4),spy(rcmB),title('rcm B')
    subplot(3,3,5),spy(chol(rcmB)),title('chol rcm B')
    subplot(3,3,6),etreeplot(rcmB),title('etreelot rcm B')

    subplot(3,3,7),spy(amdB),title('amd B')
    subplot(3,3,8),spy(chol(amdB)),title('chol amd B')
    subplot(3,3,9),etreeplot(amdB),title('etreelot amd B')
```


0
50
100
150
200
0
50
100
150
200
nz = 3137



Published with MATLAB® R2022a

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This is the second part of question 3
% where we time the cholesky decomp, forsub, and backsub
% of B, B with rcm and B with amd
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

chol_time = zeros(3,1);             %time it takes to chol various matrices
forsub_time = zeros(3,1);           %time it takes to forsub various matrices
backsub_time = zeros(3,1);          %time it takes to backsub various matrices

labels = ["original"; "rcm";"amd"];

for N = [16 32 64 128 256];

    matrix = gallery('wathen',N,N);
    amd_matrix = matrix(symamd(matrix),symamd(matrix));
    rcm_matrix = matrix(symrcm(matrix),symrcm(matrix));

    tic
    chol_B = chol(matrix);
    chol_time(1,1) = toc;

    tic
    chol_rcm = chol(rcm_matrix);
    chol_time(2,1) = toc;

    tic
    chol_amd = chol(amd_matrix);
    chol_time(3,1) = toc;

    b = rand(size(matrix,1),1);

    tic
    y = chol_B'\b;
    forsub_time(1,1) = toc;

    tic
    chol_B\y;
    backsub_time(1,1) = toc;

    tic
    y = chol_rcm'\b;
    forsub_time(2,1) = toc;

    tic
    chol_rcm\y;
    backsub_time(2,1) = toc;

    tic
    y = chol_amd'\b;
    forsub_time(3,1) = toc;
```

```
    tic
    chol_amd\y;
    backsub_time(3,1) = toc;

    disp("For a Wathen matrix of size " + N + " the timings are:")
    table(labels,chol_time, forsub_time, backsub_time)

end
```

```
For a Wathen matrix of size 16 the timings are:

ans =

  3×4 table

      labels       chol_time     forsub_time    backsub_time
    __________    __________    ___________    ____________

    "original"       0.01267      0.0014125      0.00073075
    "rcm"         0.00092287      0.0001505      4.3292e-05
    "amd"          0.0010767     3.9334e-05      3.3959e-05

For a Wathen matrix of size 32 the timings are:

ans =

  3×4 table

      labels      chol_time    forsub_time    backsub_time
    __________    _________    ___________    ____________

    "original"    0.0098026     0.0002895      0.00026042
    "rcm"          0.010899    0.00032042      0.00018575
    "amd"         0.0027617    0.00011917      9.3958e-05

For a Wathen matrix of size 64 the timings are:

ans =

  3×4 table

      labels      chol_time    forsub_time    backsub_time
    __________    _________    ___________    ____________

    "original"     0.12099      0.0051725       0.0015352
    "rcm"          0.13087      0.0046642       0.0018777
    "amd"         0.023939     0.00099208      0.00049812

For a Wathen matrix of size 128 the timings are:

ans =

  3×4 table
```

labels	chol_time	forsub_time	backsub_time
"original"	0.81112	0.061393	0.011748
"rcm"	1.1554	0.031885	0.012329
"amd"	0.092994	0.0043535	0.0022061

For a Wathen matrix of size 256 the timings are:

ans =

3×4 table

labels	chol_time	forsub_time	backsub_time
"original"	13.037	2.9336	1.6057
"rcm"	14.836	2.5037	1.2069
"amd"	0.87747	0.17305	0.018949

Published with MATLAB® R2022a