Scientific Computing Homework 3

Michael Nelson

Problem 1

Exercise 1. Implement a banded version of Gauss Elimination (without pivoting) that takes an additional argument k denoting the number of off-diagonal entries in each direction that are possibly non-zero ($a_{ij} = 0$ if $i > j + k$ or $j > i + k$) and takes advantage of this fact. Create BandedGaussElimination.m with the function function x = BandedGaussElimination(A,b,k). Test your algorithm with the matrix A=diag(2*ones(1,n))+diag(-1*ones(1,n-1),1)+diag(-1*ones(1,n-1),-1); and RHS b=ones(n,1); by comparing against MATLAB's backslash for example for $n = 10$.

Solution 1. The code is given below

```
function [x] = BandedGaussElimination(A,b,k)

n = length(b);

for j=1:n-1
        % Check to see if the pivot is zero
        if abs(A(j,j)) < 1e-15
                error('A has diagonal entries of zero')
        end

        % Apply transformation to remaining submatrix and RHS vector
        for i=j+1:j+k
                m = A(i,j)/A(j,j); % multipliyer for current row i
                for j1=j:j+k % loop to update row i
                        A(i,j1) = A(i,j1) - m*A(j,j1);
                end
                b(i) = b(i) - m*b(j); % update RHS
        end
end

% A is now upper triangular. Use backsubstitution to solve the transformed problem.
x = BackSubstitution(A,b);
```

Now we test this algorithm on the matrix A and b defined above. We write this in matlab as follows:

```
A = @(n)diag(2*ones(1,n))+diag(-1*ones(1,n-1),1)+diag(-1*ones(1,n-1),-1);
b = @(n)ones(n,1);
k = 1;
% first we check backslash
A(10)\b(10)
ans =
    5.0000
        9.0000
        12.0000
        14.0000
        15.0000
        15.0000
        14.0000
        12.0000
```

```
        9.0000
        5.0000
% now we check our algorithm
BandedGaussElimination(A(10),b(10),k)
ans =
    5.0000
        9.0000
        12.0000
        14.0000
        15.0000
        15.0000
        14.0000
        12.0000
        9.0000
        5.0000
% our algorithm works!
```

Problem 2

Exercise 2.

Solution 2. We consider $n = 2^7, \ldots, 2^{12}$ since $n = 2^{13}$ was taking too long for partial pivoting. The code is given below

```
A = @(n) diag(2*ones(1,n))+diag(-1*ones(1,n-1),1)+diag(-1*ones(1,n-1),-1);
B = @(n) sparse(A(n))
b = @(n) ones(n,1);
k = 1;

ns = 2.^(7:12);
backslashes = [];
partialpivots = [];
bandedgausses = [];
backslashsparses = [];
for n = ns
        tic;
        A(n)\b(n);
        toc;
        backslashes = [backslashes toc];
end
for n = ns
        tic;
        GaussPartialPivoting(A(n),b(n));
        toc;
        partialpivots = [partialpivots toc];
end
for n = ns
        tic;
        BandedGaussElimination(A(n),b(n),k);
        toc;
        bandedgausses = [bandedgausses toc];
end
for n = ns
        tic;
        B(n)\b(n);
        toc;
        backslashsparses = [backslashsparses toc];
```

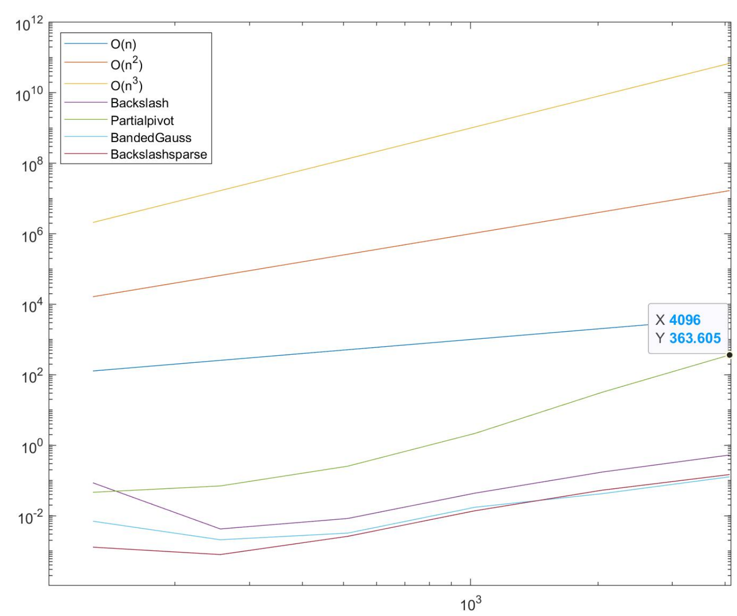
```
end

for i=1:6
        disp([backslashes(i) partialpivots(i) bandedgausses(i) backslashsparses(i)]);
end

    0.0856    0.0462    0.0070    0.0013
    0.0042    0.0699    0.0021    0.0008
    0.0083    0.2538    0.0032    0.0026
    0.0436    2.1627    0.0173    0.0138
    0.1734   31.8712    0.0422    0.0530
    0.5277  363.6051    0.1268    0.1472

ns2 = 2.^(14:2:24);
ns3 = 2.^(21:3:36);
loglog(ns,ns,ns,ns2,ns,ns3,ns,backslashes,ns,partialpivots,ns,bandedgausses,ns,backslashsparses )
legend('O(n)','O(n^2)','O(n^3)','Backslash','Partialpivot','BandedGauss','Backslashsparse'
,'Location','northwest')
```

The plot is given below



The results match our expectations.

Problem 3

Exercise 3. What is the LU factorization of the matrix

$$A = \begin{pmatrix} 1 & a \\ c & b \end{pmatrix}$$

and under what condition is the matrix A singular?

Solution 3. Let $\widetilde{L} = \left(\begin{smallmatrix} 1 & 0 \\ -c & 1 \end{smallmatrix}\right)$. Note that

$$\begin{aligned} \widetilde{L}A &= \begin{pmatrix} 1 & 0 \\ -c & b \end{pmatrix} \begin{pmatrix} 1 & a \\ c & b \end{pmatrix} \\ &= \begin{pmatrix} 1 & a \\ 0 & b-ac \end{pmatrix}, \end{aligned}$$

so setting $U = \left(\begin{smallmatrix} 1 & a \\ 0 & b-ac \end{smallmatrix}\right)$, we see that $\widetilde{L}A = U$. In particular, setting $L = \widetilde{L}^{-1} = \left(\begin{smallmatrix} 1 & 0 \\ c & 1 \end{smallmatrix}\right)$, we see that $A = LU$ is the LU factorization of A. Note that

$$\begin{aligned} A \text{ is singular} &\iff \det A = 0 \\ &\iff \det(LU) = 0 \\ &\iff \det L \det U = 0 \\ &\iff \det U = 0 \\ &\iff b = ac. \end{aligned}$$

Problem 4

Exercise 4. Compute the LU factorization of the matrix

$$A = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

Solution 4. Let $\widetilde{L} = \left(\begin{smallmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$. Note that

$$\begin{aligned} \widetilde{L}A &= \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}, \end{aligned}$$

so setting $U = \left(\begin{smallmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{smallmatrix}\right)$, we see that $\widetilde{L}A = U$. In particular, setting $L = \widetilde{L}^{-1} = \left(\begin{smallmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$, we see that $A = LU$ is the LU factorization of A.