Practical assignment 4

1) Discretization of 1D Poisson's equation. Consider discretization of Poisson's equation on a unit segment [0,1] with $u(x)|_{\Gamma} = 0$, where Γ is the boundary of the unit segment [0,1].

1) Generate the sparse matrix of the linear system for the discretized Poisson's equation

```
D = sparse(1:n,1:n,-2*ones(1,n),n,n);
E = sparse(2:n,1:n-1,ones(1,n-1),n,n);
S = E+D+E'
```

2) Show the pattern of the resulting matrix (spy command in Matlab).
3) Convert the matrix to dense format (`full` command) and compare the memory used for the storage of dense and sparse matrix (`whos` command) for different matrix size.

2) Matrix pattern and permutations. In Matlab, show the pattern for given matrix. Apply symmetric permutation of rows and columns of *A*, based on the permutation π. Show new pattern.

$\pi = \{1,3,4,2,5,6\}$

$$A = \begin{pmatrix} x & x & 0 & 0 & x & 0 \\ x & x & x & 0 & 0 & x \\ 0 & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & 0 \\ x & 0 & 0 & x & x & x \\ 0 & x & 0 & 0 & x & x \end{pmatrix}$$

Note. Matlab command for symmetric permutation: `A(p,p)`, if p is a permutation vector. The given matrix can be defined by the following commands:

```
a6=ones(6,1)
a5=ones(5,1)
a2=ones(2,1)
A=diag(a6)+diag(a5,1)+diag(a5,-1)+diag(a2,4)+diag(a2,-4)
```

3) Reordering and fill-ins. Try example from Matlab. Reorder a sparse matrix using several reordering methods (available in current version of Matlab) and compare the fill-ins incurred by the LU decomposition of the reordered matrices. Obtain the number of nonzero elements in the resulting matrices. The example below considers two symmetric reordering algorithms: symmetric approximate minimum degree algorithm and reverse Cuthill-McKee. It loads the west0479 matrix, which is a real-valued 479-by-479 sparse matrix with both real and complex pairs of conjugate eigenvalues.

Plot patterns for L and U parts. Do the same with the bucky ball matrix (the 60-by-60 sparse adjacency matrix of the connectivity graph of the Buckminster Fuller geodesic dome, **`A=bucky`**; command).

The fill-in factor is the number of nonzero elements in L and U parts divided by the number of nonzero elements in the matrix (can be calculated using **`nnz`** command). When do you observe minimal and maximal fill-in factors for west0479 matrix? When do you observe minimal and maximal fill-in factors for bucky matrix?

```
load west0479.mat
A = west0479;
%Calculate several different permutations of the matrix columns
p1 = symamd(A);% approximate minimum degree algorithm
p2 = symrcm(A); % reverse Cuthill-McKee reordering algorithm
%Compare the sparsity structures of the LU decomposition of A
% using the different ordering methods.

%you can combine all subplots in one figure
figure (1)
subplot(1,2,1)
spy(A)
title('Original Matrix')
subplot(1,2,2)
spy(lu(A))
title('LU Decomposition')
[L,U,P]=lu(A);
%Plot patterns of L and U separately, add subplots and titles

figure (2)
subplot(1,2,1)
spy(A(p1,p1))
title('Approximate Minimum Degree')
subplot(1,2,2)
spy(lu(A(p1,p1)))
title({'LU decomposition with reordering','by Approximate Minimum
Degree'})
[L1,U1,P1]=lu(A(p1,p1))
%Plot patterns of L and U separately, add subplots and titles

figure (3)
subplot(1,2,1)
spy(A(p2,p2))
title('Reverse Cuthill-McKeee')
subplot(1,2,2)
spy(lu(A(p2,p2)))
title({'LU decomposition with reordering','by Reverse Cuthill-McKee'})
[L2,U2,P2]=lu(A(p2,p2));
%Plot patterns of L and U separately, add subplots and titles
```

4) **Sparse storage**. Develop an algorithm that transfers a matrix from a dense format to a given sparse format and vice versa. Write a Matlab program for this algorithm and (optionally) for an alrorithm of matrix-by-vector multiplication, when a matrix is stored in a given sparse format (according to the variants of individual project):
   1) COORD (coordinate format)
   2) CSR (Compressed Sparse Row)

3) CSC (Compressed Sparse Column)
4) MSR (Modified Sparse Row)
5) MSC (Modified Sparse Column)
6) Ellpack-Itpack