

Practical assignment 10. Comparison of Direct and Iterative Methods for different sparse systems

Consider two different linear sparse systems from the files `pe0sym.mat` and `mxw3d.mat`. Each of these files contains both matrix of the system and exact solution. The right-hand side vector is computed as $b = A \cdot x_{\text{exact}}$. Matrix from the system `pe0sym.mat` is a symmetric banded matrix. Matrix from the system `mxw3d.mat` has typical structure for 3D discretized PDEs.

Apply two direct and two iterative sparse solvers to solve both systems, in each case reporting the CPU time required to find the solution, the residual and the absolute error. Here we shall use sparse iterative solvers without preconditioners. For iterative methods, to reach convergence, try different values for number of iterations. However, remember that iterative methods do not always converge. Investigate two systems separately (you can write two different files for each system), and then fill in the table below.

Use the following Matlab code in your assignment. Commented lines in bold font indicate the parts you need to write on your own.

- 1) Start with loading the system. Then calculate the right-hand side vector and plot the sparse pattern of the matrix A .

```
clc
clear
% load the system (matrix and exact solution)
load pe0sym.mat
%load mxw3d.mat
whos %show info about the system
b = A*x_exact; % compute the right-hand side vector
%check here whether the matrix is symmetric

figure(1)
%show sparse pattern of A
```

- 2) The first direct method to use (referred as DM1) is Matlab backslash operator `\`. Write the code for computing solution by this method and then report the residual and relative residual norm, as well as absolute error.

```
% Direct Method 1: backslash operator
disp('Time required for DM1')
%find solution x1 using direct method DM1 and calculate cpu
%time required to find it
x1=...;
disp('Residual norm for DM1')
norm(b-A*x1,2)
disp('Relative residual norm for DM1')
norm(b-A*x1,2)/norm(b,2)
disp('Absolute error norm for DM1')
norm(x1-x_exact,2)
```

- 3) The second direct method (referred as DM2) computes lu-factorization of the matrix in the form $[L, U, P, Q] = \text{lu}(A)$ and then applies forward and backward substitutions. Here `lu` command computes a sparse factorization $PAQ = LU$, where P and Q are permutation matrices chosen to minimize the fill-ins in the triangular parts L and U . Write the code for computing solution by this method and then report the residual and relative residual norm, as well as absolute error. When computing time required to find the solution, consider two cases: a) when cpu time to find lu-factorization is counted; b) when cpu time to find lu-factorization is not taken into account. Plot the sparse pattern of the reordered matrix PAQ and compute the fill-in factor of the system (number of nonzero entries in L and U parts divided by the number of nonzero entries in the original matrix).

```
% Direct Method 2: factorization PAQ=LU
[L,U,P,Q]=lu(A);
figure(2)
%show sparse pattern of PAQ
%calculate the fill-in factor
fill_in=...
disp('Time required for DM2-1')
%find solution x2 using direct method DM2 and calculate cpu
%time DM2-1 required to find it (including time of lu-
%factorization)
[L,U,P,Q]=lu(A);
x2=Q\'(U\'(L\'(P\'b)));
disp('Time required for DM2-2')
%find solution x2 using direct method DM2 and calculate cpu
%time DM2-2 required to find it (NOT including time of lu-
%factorization)
x2=Q\'(U\'(L\'(P\'b)));
disp('Residual norm for DM2')
%calculate residual norm
disp('Relative residual norm for DM2')
%calculate relative residual norm
disp('Absolute error norm for DM2')
%calculate absolute error norm
```

For iterative solvers we shall use MINRES and GMRES methods. Note that there are also other sparse iterative methods in Matlab, but not all of them converge for the given systems. See Matlab help for the list of available sparse iterative methods.

- 4) The first iterative solver to try is MINRES method. Use it in the form `minres(A,b,tol,maxit)`. Set the tolerance `tol` to default value: `tol=[]`, vary the maximum number of iterations `maxit` from 20 to 5000, and obtain the minimum number of iterations when the method converges. If the method does not converge for the given range of iterations, report “no convergence”.

```
% Iterative Method 1: MINRES
disp('Time required for ItM MINRES')
%find solution x3 using MINRES iterative method and calculate
%cpu time required to find it
```

```

x3=...;
disp('Number of iterations for MINRES')
%calculate the minimum number of iterations required for
%convergence or report "no convergence"
disp('Residual norm for MINRES ')
%calculate residual norm
disp('Relative residual norm for MINRES ')
%calculate relative residual norm
disp('Absolute error norm for MINRES ')
%calculate absolute error norm

```

- 5) The second iterative solver to try is GMRES method. Use it in the form `gmres(A,b,rest)`. Vary the number of restarts `rest` from 10 to 100 and determine when GMRES converges. If the method does not converge for the given range of restarts, report “no convergence”.

```

% Iterative Method 2: GMRES
disp('Time required for ItM GMRES')
%find solution x4 using GMRES iterative method and calculate
%cpu time required to find it
X4=...;
disp('Number of restarts for GMRES')
%calculate the minimum number of restarts required for
%convergence or report "no convergence"
disp('Residual norm for GMRES')
%calculate residual norm
disp('Relative residual norm for GMRES ')
%calculate relative residual norm
disp('Absolute error norm for GMRES ')
%calculate absolute error norm

```

- 6) Write a small report on your observations. Show the patterns of the matrices A and PAQ for both systems from `pe0sym.mat` and `mxw3d.mat`. Indicate the values of fill-in factors for each system. Fill in the following table with results of your investigation. If iterative method does not converge, write “no convergence” in the respective cell. **Make conclusions based on your results.**

	Methods			
	Direct		Iterative	
Results	DM1	DM2	MINRES	GMRES
System: pe0sym				
Cpu time		DM2-1 (includes time of lu-factorization): DM2-2 (does not include time of lu-factorization):		
Residual norm				
Relative residual norm				
Absolute error				

Number of iterations	-	-		-
Number of restarts	-	-	-	
System: mxw3d				
Cpu time		DM2-1 (includes time of lu-factorization): DM2-2 (does not include time of lu-factorization):		
Residual norm				
Relative residual norm				
Absolute error				
Number of iterations	-	-		-
Number of restarts	-	-	-	