Test the effects of reordering sparse matrix in MATLAB

Dai bin

Unit: lsec, Email: daibin2020@lsec.cc.ac.cn

There are at least six **commands** to reorder sparse matrix in **MATLAB**: AMD(Approximate minimum degree permutation), SYMAMD(Symmetric approximate minimum degree permutation), COLAMD(Column approximate minimum degree permutation), SYMRCM(Sparse reverse Cuthill-McKee ordering), DISSECT(Nested dissection permutation), COLPERM(Sparse column permutation based on nonzero count).

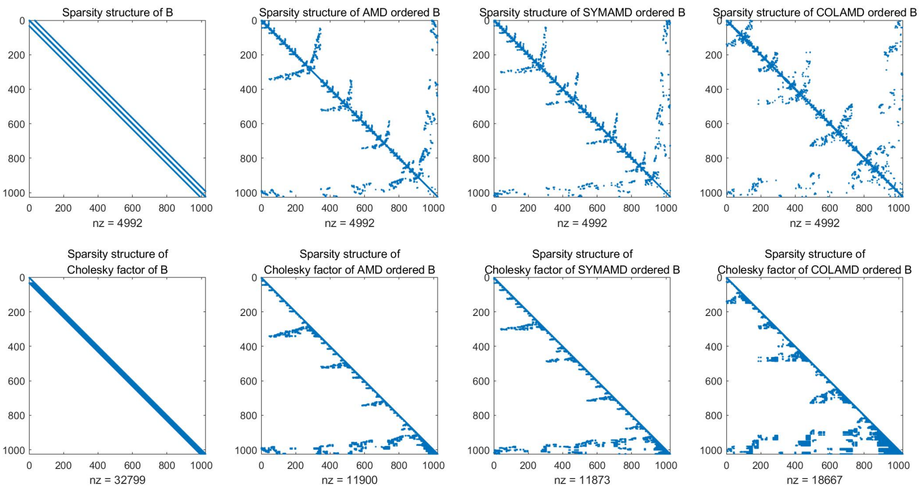
1 Poisson equation

Consider the poisson equation with Dirichlet boundary condition:

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega. \end{aligned}$$

We will get the coefficient matrix by standard five point stencil and test the effects of different ordering of grid points on sparsity and efficiency of solving the equation.

1.1 The effects on cholesky decomposition



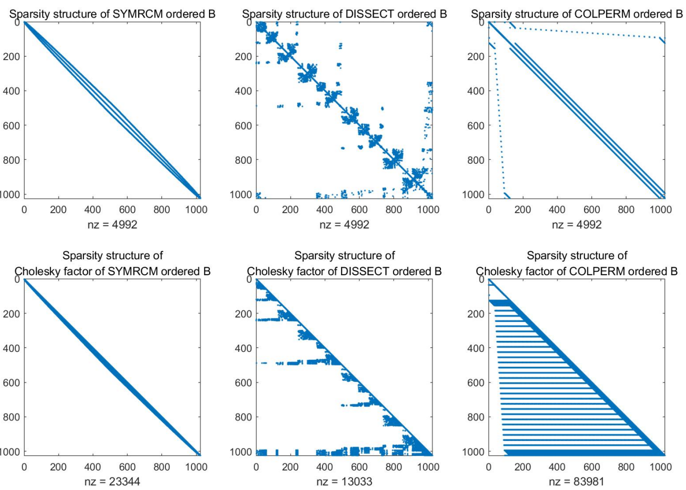


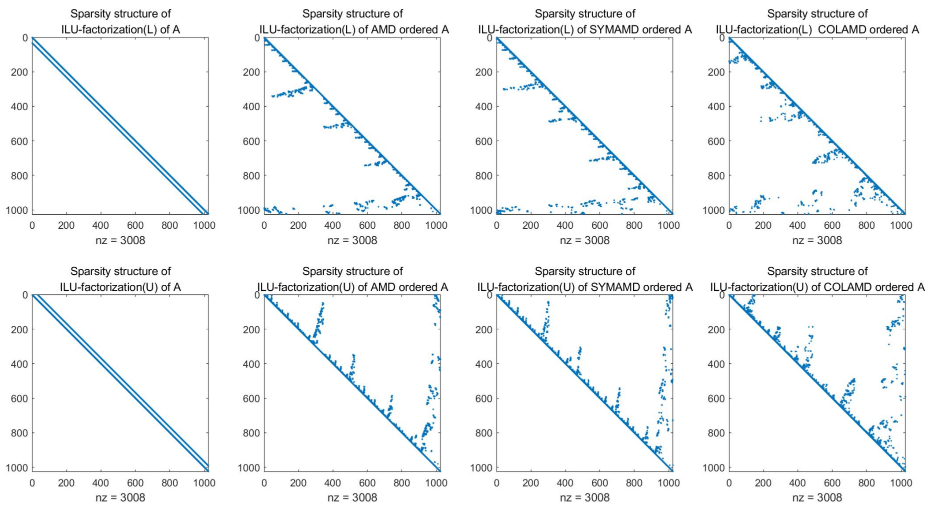
表 1: Elapsed time for reodering algorithms

Algorithms	**Time**
AMD	0.000380
SYMAMD	0.001635
COLAMD	0.001004
SYMRCM	0.000054
DISSECT	0.002660
COLPERM	0.000514

表 2: Elapsed time for cholesky factorization

Order	**Time**
lexicographic	0.001026
AMD-ordered	0.000729
SYMAMD-ordered	0.000681
COLAMD-ordered	0.000940
SYMRCM-ordered	0.000895
DISSECT-ordered	0.000829
COLPERM-ordered	0.003534

1.2 The effects on the sparsity of ILU-factorization



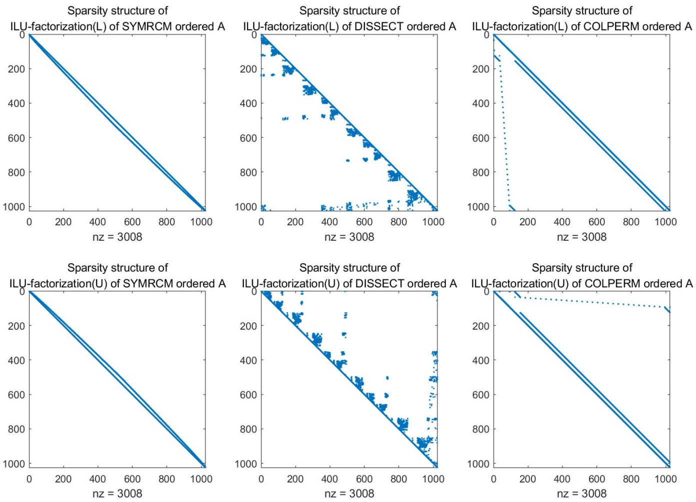


表 3: Elapsed time for ILU factorization

Order	**Time**
lexicographic	0.000972
AMD-ordered	0.000902
SYMAMD-ordered	0.000866
COLAMD-ordered	0.000812
SYMRCM-ordered	0.000392
DISSECT-ordered	0.000921
COLPERM-ordered	0.001022

1.3 The numerical results by ILU-PCG

表 4: mesh size = 32×32; preconditioner: ILU with droptol = 0.1

Method	**Order**	**Size**	**Iteration**	**Relative residual**	**Number: Fill-ins**
cg	lexicographic	1024	84	8.6e-07	\
pcg	lexicographic	1024	65	9.9e-07	4096
pcg	amd-ordered	1024	36	9.3e-07	5768
pcg	symamd-ordered	1024	35	8.9e-07	5760
pcg	colamd-ordered	1024	38	8.3e-07	4412
pcg	symrcm-ordered	1024	61	1e-06	4096
pcg	dissect-ordered	1024	36	7.2e-07	5624
pcg	colperm-ordered	1024	64	9.7e-07	4096

表 5: mesh size = 64×64; preconditioner: ILU with droptol = 0.1

Method	**Order**	**Size**	**Iteration**	**Relative residual**	**Number: Fill-ins**
cg	lexicographic	4096	164	8.6e-07	\
pcg	lexicographic	4096	\	\	16384
pcg	amd-ordered	4096	63	9.9e-07	23896
pcg	symamd-ordered	4096	68	7.4e-07	23868
pcg	colamd-ordered	4096	78	9.6e-07	17806
pcg	symrcm-ordered	4096	\	\	16384
pcg	dissect-ordered	4096	68	8.3e-07	23356
pcg	colperm-ordered	4096	\	\	16384

表 6: mesh size = 128×128; preconditioner:ILU with droptol = 0.1

Method	Order	Size	Iteration	Relative residual	Number: Fill-ins
cg	lexicographic	16384	321	9.9e-07	\
pcg	lexicographic	16384	\	\	65536
pcg	amd-ordered	16384	124	9.6e-07	97232
pcg	symamd-ordered	16384	136	9.2e-07	97076
pcg	colamd-ordered	16384	209	9.5e-07	71728
pcg	symrcm-ordered	16384	\	\	65536
pcg	dissect-ordered	16384	170	9.3e-07	92206
pcg	colperm-ordered	16384	\	\	65536

表 7: mesh size = 128×128; preconditioner: ILU with droptol = 0.05

Method	Order	Size	Iteration	Relative residual	Number: Fill-ins
cg	lexicographic	16384	320	9.6e-07	\
pcg	lexicographic	16384	81	7.9e-07	97794
pcg	amd-ordered	16384	110	9.5e-07	130398
pcg	symamd-ordered	16384	123	8.9e-07	130644
pcg	colamd-ordered	16384	92	8.3e-07	115946
pcg	symrcm-ordered	16384	\	\	97794
pcg	dissect-ordered	16384	113	9.2e-07	130618
pcg	colperm-ordered	16384	79	8.8e-07	97794

The MATLAB code is:

```
A = gallery('poisson',32); % Poisson问题
tic; a = amd(A); toc       %近似最小度置换:Approximate minimum degree permutation
tic; b = symamd(A); toc    %对称近似最小度置换:Symmetric approximate minimum degree permutation
tic; c = colamd(A); toc    %列近似最小度排列:Column approximate minimum degree permutation
tic; d = symrcm(A); toc    %稀疏反向 Cuthill-McKee 排序:Sparse reverse Cuthill-McKee ordering
tic; e = dissect(A); toc   %嵌套剖分置换:Nested dissection permutation
tic; f = colperm(A); toc   %基于非零项计数的稀疏列置换:Sparse column permutation based on nonzero count

tic;L = chol(A,'lower'); toc
tic;L_a = chol(A(a,a),'lower');toc
tic;L_b = chol(A(b,b),'lower');toc
tic;L_c = chol(A(c,c),'lower');toc
tic;L_d = chol(A(d,d),'lower');toc
```

```
tic;L_e = chol(A(e,e),'lower');toc
tic;L_f = chol(A(f,f),'lower');toc

figure;
subplot(2,4,1);     spy(A);
title('Sparsity structure of A');

subplot(2,4,2); spy(A(a,a));
title('Sparsity structure of AMD ordered A');

subplot(2,4,3); spy(A(b,b));
title('Sparsity structure of SYMAMD ordered A');

subplot(2,4,4);     spy(A(c,c));
title('Sparsity structure of COLAMD ordered A');

subplot(2,4,5);     spy(L);
title({'Sparsity structure of';'Cholesky factor of A'});

subplot(2,4,6); spy(L_a);
title({'Sparsity structure of';'Cholesky factor of AMD ordered A'});

subplot(2,4,7); spy(L_b);
title({'Sparsity structure of';'Cholesky factor of SYMAMD ordered A'});

subplot(2,4,8);     spy(L_c);
title({'Sparsity structure of';'Cholesky factor of COLAMD ordered A'});


figure;
subplot(2,4,1);     spy(A(d,d));
title('Sparsity structure of SYMRCM ordered A');

subplot(2,4,2);     spy(A(e,e));
title('Sparsity structure of DISSECT ordered A');

subplot(2,4,3);     spy(A(f,f));
title('Sparsity structure of COLPERM ordered A');

subplot(2,4,5);     spy(L_d);
title({'Sparsity structure of';'Cholesky factor of SYMRCM ordered A'});

subplot(2,4,6); spy(L_e);
title({'Sparsity structure of';'Cholesky factor of DISSECT ordered A'});

subplot(2,4,7); spy(L_f);
title({'Sparsity structure of';'Cholesky factor of COLPERM ordered A'});

%用ILU-PCG求解方程组Ax = rhs
rhs = rand(size(A,1),1);
setup.type = 'ilutp';
setup.droptol = 0.1;  %parameter of ilu: Drop tolerance
tol = 1e-6;           %parameter of pcg: iteration tolerence
maxit = 10000;         %parameter of pcg:the maximum number of iterations

x_cg = pcg(A,rhs,tol,maxit);  % cg

tic; [L_i,U_i] = ilu(A,setup); toc
fill_ins = (nnz(L_i) - (nnz(A) - size(A,1))/2 + size(A,1))*2

tic; [L_i_a,U_i_a] = ilu(A(a,a),setup);toc
fill_ins = (nnz(L_i_a) - (nnz(A(a,a)) - size(A(a,a),1))/2 + size(A(a,a),1))*2

tic; [L_i_b,U_i_b] = ilu(A(b,b),setup);toc
fill_ins = (nnz(L_i_b) - (nnz(A(b,b)) - size(A(b,b),1))/2 + size(A(b,b),1))*2

tic;[L_i_c,U_i_c] = ilu(A(c,c),setup);toc
fill_ins = (nnz(L_i_c) - (nnz(A(c,c)) - size(A(c,c),1))/2 + size(A(c,c),1))*2

tic; [L_i_d,U_i_d] = ilu(A(d,d),setup);toc
fill_ins = (nnz(L_i_d) - (nnz(A(d,d)) - size(A(d,d),1))/2 + size(A(d,d),1))*2
```

```
tic; [L_i_e,U_i_e] = ilu(A(e,e),setup);toc
fill_ins = (nnz(L_i_e) - (nnz(A(e,e)) - size(A(e,e),1))/2 + size(A(e,e),1))*2

tic; [L_i_f,U_i_f] = ilu(A(f,f),setup);toc
fill_ins = (nnz(L_i_f) - (nnz(A(f,f)) - size(A(f,f),1))/2 + size(A(f,f),1))*2


x = pcg(A,rhs,tol,maxit,L_i,U_i);
x_1 = pcg(A(a,a),rhs,tol,maxit,L_i_a,U_i_a);
x_2 = pcg(A(b,b),rhs,tol,maxit,L_i_b,U_i_b);
x_3 = pcg(A(c,c),rhs,tol,maxit,L_i_c,U_i_c);
x_4 = pcg(A(d,d),rhs,tol,maxit,L_i_d,U_i_d);
x_5 = pcg(A(e,e),rhs,tol,maxit,L_i_e,U_i_e);
x_6 = pcg(A(f,f),rhs,tol,maxit,L_i_f,U_i_f);

figure;
subplot(2,4,1);    spy(L_i);
title({'Sparsity structure of';'ILU-factorization(L) of A'});

subplot(2,4,2); spy(L_i_a);
title({'Sparsity structure of';' ILU-factorization(L) of AMD ordered A'});

subplot(2,4,3); spy(L_i_b);
title({'Sparsity structure of';'ILU-factorization(L) of SYMAMD ordered A'});

subplot(2,4,4);    spy(L_i_c);
title({'Sparsity structure of';'ILU-factorization(L)  COLAMD ordered A'});

subplot(2,4,5);    spy(U_i);
title({'Sparsity structure of';'ILU-factorization(U) of A'});

subplot(2,4,6); spy(U_i_a);
title({'Sparsity structure of';'ILU-factorization(U) of AMD ordered A'});

subplot(2,4,7); spy(U_i_b);
title({'Sparsity structure of';'ILU-factorization(U) of SYMAMD ordered A'});

subplot(2,4,8);    spy(U_i_c);
title({'Sparsity structure of';'ILU-factorization(U) of COLAMD ordered A'});

figure;
subplot(2,4,1);    spy(L_i_d);
title({'Sparsity structure of';'ILU-factorization(L) of SYMRCM ordered A'});

subplot(2,4,2);    spy(L_i_e);
title({'Sparsity structure of';'ILU-factorization(L) of DISSECT ordered A'});

subplot(2,4,3);    spy(L_i_f);
title({'Sparsity structure of';'ILU-factorization(L) of COLPERM ordered A'});

subplot(2,4,5);    spy(U_i_d);
title({'Sparsity structure of';'ILU-factorization(U) of SYMRCM ordered A'});

subplot(2,4,6); spy(U_i_e);
title({'Sparsity structure of';'ILU-factorization(U) of DISSECT ordered A'});

subplot(2,4,7); spy(U_i_f);
title({'Sparsity structure of';'ILU-factorization(U) of COLPERM ordered A'});
```

1.4 The numerical results by IC-PCG

表 8: mesh size = 32×32; preconditioner: IC with droptol = 0.1

Method	**Order**	**Size**	**Iteration**	**Relative residual**	**Number: Fill-ins**
cg	lexicographic	1024	84	8.2e-07	\
pcg	lexicographic	1024	27	7.6e-07	4096
pcg	amd-ordered	1024	34	7.1e-06	5692
pcg	symamd-ordered	1024	34	7.3e-07	5676
pcg	colamd-ordered	1024	34	5.2e-07	4320
pcg	symrcm-ordered	1024	27	9.3e-07	4096
pcg	dissect-ordered	1024	37	8.1e-07	5434
pcg	colperm-ordered	1024	26	9.5e-07	4096

表 9: mesh size = 64×64; preconditioner: ILU with droptol = 0.1

Method	**Order**	**Size**	**Iteration**	**Relative residual**	**Number: Fill-ins**
cg	lexicographic	4096	163	8.4e-07	\
pcg	lexicographic	4096	50	7e-07	16384
pcg	amd-ordered	4096	63	7.5e-07	23716
pcg	symamd-ordered	4096	64	7.4e-07	23698
pcg	colamd-ordered	4096	62	8.1e-06	17408
pcg	symrcm-ordered	4096	50	7.5e-07	16384
pcg	dissect-ordered	4096	68	7.6e-07	22662
pcg	colperm-ordered	4096	50	8.4e-07	16384

表 10: mesh size = 128×128; preconditioner:ILU with droptol = 0.1

Method	Order	Size	Iteration	Relative residual	Number: Fill-ins
cg	lexicographic	16384	320	9.4e-07	\
pcg	lexicographic	16384	96	8.1e-07	65536
pcg	amd-ordered	16384	114	9.5e-07	96852
pcg	symamd-ordered	16384	127	9.3e-07	96714
pcg	colamd-ordered	16384	125	9.2e-07	69906
pcg	symrcm-ordered	16384	96	9.5e-07	65536
pcg	dissect-ordered	16384	134	9.4e-07	88252
pcg	colperm-ordered	16384	96	8.4e-07	65536

表 11: mesh size = 128×128; preconditioner:ILU with droptol = 0.05

Method	Order	Size	Iteration	Relative residual	Number: Fill-ins
cg	lexicographic	16384	321	9.6e-07	\
pcg	lexicographic	16384	88	8.8e-07	65790
pcg	amd-ordered	16384	89	7.5e-07	127994
pcg	symamd-ordered	16384	90	7.8e-07	128018
pcg	colamd-ordered	16384	104	8.9e-07	79250
pcg	symrcm-ordered	16384	96	8.3e-07	65790
pcg	dissect-ordered	16384	100	9e-07	114486
pcg	colperm-ordered	16384	96	9.4e-07	65796

The MATLAB code is:

```
A = gallery('poisson',32); % Poisson问题
tic; a = amd(A); toc        %近似最小度置换:Approximate minimum degree permutation
tic; b = symamd(A); toc     %对称近似最小度置换:Symmetric approximate minimum degree permutation
tic; c = colamd(A); toc     %列近似最小度排列:Column approximate minimum degree permutation
tic; d = symrcm(A); toc     %稀疏反向 Cuthill-McKee 排序:Sparse reverse Cuthill-McKee ordering
tic; e = dissect(A); toc    %嵌套剖分置换:Nested dissection permutation
tic; f = colperm(A); toc    %基于非零项计数的稀疏列置换:Sparse column permutation based on nonzero count

opts.type = 'ict';
opts.droptol = 0.1;


L = ichol(A,opts);
```

```
fill_ins = (nnz(L) - (nnz(A) - size(A,1))/2 + size(A,1))*2

L_a = ichol(A(a,a),opts);
fill_ins = (nnz(L_a) - (nnz(A(a,a)) - size(A(a,a),1))/2 + size(A(a,a),1))*2

L_b = ichol(A(b,b),opts);
fill_ins = (nnz(L_b) - (nnz(A(b,b)) - size(A(b,b),1))/2 + size(A(b,b),1))*2

L_c = ichol(A(c,c),opts);
fill_ins = (nnz(L_c) - (nnz(A(c,c)) - size(A(c,c),1))/2 + size(A(c,c),1))*2

L_d = ichol(A(d,d),opts);
fill_ins = (nnz(L_d) - (nnz(A(d,d)) - size(A(d,d),1))/2 + size(A(d,d),1))*2

L_e = ichol(A(e,e),opts);
fill_ins = (nnz(L_e) - (nnz(A(e,e)) - size(A(e,e),1))/2 + size(A(e,e),1))*2

L_f = ichol(A(f,f),opts);
fill_ins = (nnz(L_f) - (nnz(A(f,f)) - size(A(f,f),1))/2 + size(A(f,f),1))*2

rhs = rand(size(A,1),1);
tol = 1e-6;          %parameter of pcg: iteration tolerence
maxit = 1000;        %parameter of pcg:the maximum number of iterations

x_cg = pcg(A,rhs,tol,maxit);  % cg
x_1 = pcg(A,rhs,tol,maxit,L*L');
x_2 = pcg(A(a,a),rhs,tol,maxit,L_a*L_a');
x_3 = pcg(A(b,b),rhs,tol,maxit,L_b*L_b');
x_4 = pcg(A(c,c),rhs,tol,maxit,L_c*L_c');
x_5 = pcg(A(d,d),rhs,tol,maxit,L_d*L_d');
x_6 = pcg(A(e,e),rhs,tol,maxit,L_e*L_e');
x_7 = pcg(A(f,f),rhs,tol,maxit,L_f*L_f');
```