Division of Scientific Computing
Department of Information Technology
Uppsala University

Numerical Linear Algebra
Feb-March 2021

Project - Solving linear systems with complex symmetric matrices

1 Aim of the project

The project deals with large scale linear systems with complex symmetric matrices and their solution using iterative methods. Consider the system

$$C\boldsymbol{z} = \boldsymbol{c}, \tag{1}$$

where $\boldsymbol{z} \in \mathbb{C}^n, \boldsymbol{c} \in \mathbb{C}^n$ and $C \in \mathbb{C}^{n\times n}$.
The matrix is called *complex symmetric* if $C = C^T$. We assume also that the real part of C is symmetric positive definite.
The theory is valid also in more general cases, where the system to be solved has the structure in the case when the symmetric part of the imaginary part of C, i.e., the matrix $1/2(imag(C) + (imag(C))^T)$, is positive semidefinite.

2 Solution approaches

Approach 1:

Solve the system as it is given (i.e., as a complex system) using a suitable iterative solution method. This is possible in Matlab and in various other packages.

Approach 2:

Use preconditioned iterative solution methods to solve the system using two techniques to construct a preconditioner, referred to as 'PMHSS' and 'PRESB'. As is seen below PMHSS is based on a matrix splitting. PRESB is based on rewriting the complex matrix in the so-called *complex-to-real* form. The complex system (1) is solved after rewriting it in a real form as a twice larger real system of equations. This enables us to use the rich experience in solving real systems of equations with iterative methods,
We describe the *complex-to-real* idea in more detail. The matrix C and the vector $\boldsymbol{c}$ have complex entries. The solution $\boldsymbol{z}$ is a complex vector itself.

We rewrite (1) and separate the equations for the real and the complex part of the solution vector $\boldsymbol{x}$. Let $\boldsymbol{z} = \boldsymbol{x} + i\,\boldsymbol{y}$ and similarly, $C = A + i\,B$ and $\boldsymbol{c} = \boldsymbol{a} + i\,\boldsymbol{b}$. We substitute the latter in (1) and obtain the following real linear system of equations which, as already mentioned, is twice larger:

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{bmatrix} \tag{2}$$

The matrix $\mathcal{A} = \begin{bmatrix} A & -B \\ B & A \end{bmatrix}$ is skew-symmetric, i.e., $\mathcal{A}^T = -\mathcal{A}$. (It has purely imaginary eigenvalues.)
The system in (2) is now real and can be solved using standard iterative solution methods, such as `GMRES` and others. Note that the blocks are square.
Systems of the above form arise in many contexts, for instance, from complex-to-real formulation of complex symmetric matrices, distributed optimal control problems, constrained by partial differential equations, multiphase flow problems solved as Kahn-Hilliard formulation and others.

Methods to solve systems with such matrices have been studied much in the scientific literature. As stated, we consider two preconditioning methods for this class of problems. The first one is referred to as the *Preconditioned Modified Hermitian Skew-Hermitian Splitting* (PMHSS) method ([3, 4] etc.) and the second one is referred to as *PREconditioning for matrices with Square Blocks* (PRESB) ([1, 2] etc.).

- PMHSS
  Let $C = a + iB$ and $C\boldsymbol{x} = \boldsymbol{b}$. Consider an alternating direction type of method, described in [**?**] and [4]. It has the form of a stationary (fixed point) iteration method,

$$\begin{array}{l} (\alpha V + A)\boldsymbol{x}^{k+1/2} = (\alpha V - iB)\boldsymbol{x}^k + \boldsymbol{b} \\ (\alpha V + B)\boldsymbol{x}^{k+1} = (\alpha V + iA)\boldsymbol{x}^{k+1/2} - i\boldsymbol{b},\ k = 0, 1, \dots . \end{array} \tag{3}$$

  Here V and α are method parameters. The matrix V is a preconditioner, chosen as a symmetric and positive definite (spd) matrix if A and B are symmetric and positive semidefinite (spsd).

  It can be shown that $C = F - G$, where $F(V, \alpha) = \frac{1+i}{2\alpha}(\alpha V + A)V^{-1}(\alpha V + B)^{-1}$ and $G(V, \alpha) = \frac{1+i}{2\alpha}(\alpha V + A)V^{-1}(\alpha V - iB)^{-1}$.

  It is also possible to use F as a preconditioner for a GMRES-type of method. This method still involves some complex arithmetics, but to a lesser extent than if a method is applied directly for (1).

  In the numerical examples we apply PMHSS in (3) with $\alpha = 1$ and $V = A$, as well as a preconditioner for the GMRES method. Letting the initial approximation to be $\boldsymbol{x}^0 = \boldsymbol{0}$, for this particular choice of the method parameters, the application of the preconditioner simplifies significantly and becomes as in Algorithm 1. Here $\boldsymbol{q}$ is the current residual in the iterative solution method. Thus, PMHSS requires only one solution with $A + B$ and complex arithmetic, while PRESB requires two solutions with $A + B$ and real arithmetic.

Algorithm 1 PMHSS preconditioner

```
1: Solve $(A + B)\boldsymbol{z} = \boldsymbol{q}$
2: Set $\boldsymbol{x} = 0.5 * (1 - i)\boldsymbol{z}$
```

- PRESB

  Assume that A is symmetric and positive definite and B is positive semidefinite. The PRESB preconditioner has the following form:

$$\mathcal{P}_{PRESB} \equiv \begin{bmatrix} A & -B \\ B & A+2B \end{bmatrix} = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix} \begin{bmatrix} A+B & 0 \\ B & A+B \end{bmatrix} \begin{bmatrix} I & I \\ 0 & I \end{bmatrix} \tag{4}$$

  It is shown that all eigenvalues of the preconditioned system $\mathcal{P}_{PRESB}^{-1}\mathcal{A}$ belong to the interval $[0.5, 1]$. Thus, when solving the system (2) iteratively, using $\mathcal{P}_{PRESB}$ as a preconditioner, the number of iterations is bounded, independently of the size of the system. This property guarantees high numerical efficiency of the preconditioner.

  The factorization of $\mathcal{P}_{PRESB}$, shown in (4) leads to a very efficient algorithm to solve systems with it. Since the PRESB preconditioner is applicable to more general class of problems, where the off-diagonal blocks can be non-symmetric (but their symmetric part is positive semi-definite), and even complex, we define the algorithm in a more general form. Let

$$\mathcal{A} = \begin{bmatrix} A & -B_2 \\ B_1 & A \end{bmatrix}.$$

  Then

$$\mathcal{P}_{PRESB} \equiv \begin{bmatrix} A & -B_2 \\ B_1 & A+B_1+B_2 \end{bmatrix} = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix} \begin{bmatrix} A+B_1 & 0 \\ B_1 & A+B_2 \end{bmatrix} \begin{bmatrix} I & I \\ 0 & I \end{bmatrix} \tag{5}$$

  The steps to perform the action $\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \mathcal{P}_{PRESB}^{-1} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{bmatrix}$ are given in Algorithm 2.

Algorithm 2 PRESB preconditioner

```
1: Let $H_1 = A + B_1$, $H_2 = A + B_2$
2: Solve $H_1\boldsymbol{h} = \boldsymbol{p} + \boldsymbol{q}$
3: Solve $H_2\boldsymbol{y} = \boldsymbol{q} - B_1\boldsymbol{h}$
4: Compute $\boldsymbol{x} = \boldsymbol{h} - \boldsymbol{y}$
```

  Apart from two vector operations, the computational cost of applying the PRESB preconditioner is the cost to solve two systems with H_1 and H_2 and one matrix-vector multiplication with B_1. Only real arithmetic is required. It is expected that when solving with H_1 and H_2, we can use some standard out-of-the-box preconditioning technique.

3 Tasks

1. You have at your disposal two sets of complex symmetric matrices, described in Problem 1 and Problem 2.

   Using both sets. You could be curious and
   (i) check the sparsity of the above matrices (`spy`) and their structure,
   (ii) for some smaller-sized matrices compute the complete spectra by using the `MATLAB` function `eig` and see how the spectrum changes with size.

2. The *Quasi-Minimal Residual (QMR)* method is one of the iterative solution methods, recommended for solving complex linear systems. Read some theory about QMR and describe it briefly. What are the main features of this method? Is it computationally cheaper than GMRES, for instance?

3. Solve Problem (1) using unpreconditioned `QMR`. Plot the convergence. (Please use `semilogy` and not `plot`.)

   For $n = 50$ you see that it takes exactly 50 iterations for QMR to converge. Is this an illustration of the *final termination property of the method*?

4. Solve the problem in a real form (2) using a preconditioned `GMRES` or .

   Implement the action of the preconditioner as follows:

   Implement the preconditioners $\mathcal{P}_{PRESB}$ and $\mathcal{P}_{PMHSS}$ as defined via the algorithms. You have to write a Matlab routine, called, say, `blkprec.m`, which implements the preconditioners.

   The way to use the preconditioner in MATLAB is as follows

   ```
   [x,flag,relres,iter,resvec] = gmres(A,rhs,restart,tol,maxit,...
                                   @blkprec,[],[],A,A)
   ```

   The MATLAB function

   ```
    function w = blkprec(v,A,B)
   ```

   should implement the solution of the system $\mathcal{P}_{xx}\boldsymbol{w} = \boldsymbol{v}$ as sketched above. To solve systems with $A + B$ use MATLAB backslash operator.

   Remark: Of course, one can use OMR, preconditioned by $\mathcal{P}_{PRESB}$. However, QMR requires the action of the transposed of the preconditioner and some special attention should be paid to that issue when implementing the preconditioner in `blkprec.m`.

5. As a theoretical exercise, include in the report the derivation of the spectral properties of $\mathcal{P}_{PRESB}^{-1}\mathcal{A}$ or of $\mathcal{P}_{PMHSS}^{-1}\mathcal{A}$.

4 Test matrices

You have at your disposal two sets of complex symmetric test-matrices.

Problem 1 *Consider the so-called* $R_{22}-$Padé approximation *systems*

$$A = \Big(I + \Big(1 + i\Big)\ L\Big) \tag{6}$$

which arise in Padé type integration schemes for parabolic problems. The matrices are generated by `Pade_parabolic_matrix.m`*, available in the Dropbox directory.*
Create consecutively matrices of increasing size: 50, 100, 500, 5000 $\cdots$.

Problem 2 *You have at hand six matrices, arising in simulating electromagnetic time-harmonic Eddy current problems. The matrices are complex and parameter-dependent, namely, generated for two values of a problem parameter* ω*, which denotes frequency and for three problem sizes,* 293*,* 2903 *and* 25602*. The PDE problem is in 3D, discretized by Nedelec finite elements.*

```
Matrix_Eddy_0.0001_293.mat      Matrix_Eddy_10000_293.mat
Matrix_Eddy_0.0001_2903.mat     Matrix_Eddy_10000_2903.mat
Matrix_Eddy_0.0001_25602.mat    Matrix_Eddy_10000_25602.mat
```

The files are uploaded in the dropbox directory.

5 Writing a report on the results

The report has to have the following issues covered:

1. Brief description of the problem and the methods used and the computational complexity of the preconditioner $\widehat{P}$.
2. Upon your choice, a theoretical derivation of the properties of one of the two preconditioners.
3. Numerical experiments

   Describe the experiments (iteration counts, plots of the residual history, timing) for representative cases. How does the number of iterations grow with the size? Which method is to recommended for the given test problems and why? Please note, that iteration counts should be reported preferably in a table and not as a graph.
4. Conclusions.

A printout of the Matlab code must be attached to the report.
Success!

References

[1] O. Axelsson, Optimality properties of a square block matrix preconditioner with applications, Computers & Mathematics with Applications, 80 (2020), 286-294

[2] O. Axelsson, M. Neytcheva, B. Ahmad, A comparison of iterative methods to solve complex valued linear algebraic systems. Numer. Algorithms, 66 (2014), 811-841.

[3] M. Benzi, D. Bertaccini, Block preconditioning of real–valued iterative algorithms for complex linear systems, *SIAM J. Numer. Anal.* 28 (2008), 598–618.

[4] Z.-Z.Bai, M. Benzi, F. Chen, On preconditioned MHSS iteration methods for complex symmetric linear systems, *Numer. Alg.*, 56 (2011), 297–317.