EE 613: Non-linear Dynamical Systems

Solving the generalized Sylvester's equation $AXB + CXD = E$

Dimple Kochar
16D070010

20th March, 2020

Abstract

Sylvester's equation is introduced defining all the terms in it and its solvability conditions are discussed. Definitions and algorithms to reduce it to $AX + XB = C$ and then solve it have been taken from well-accepted papers and accordingly referenced. Since all the literature on this topic is decades old and most innovations on algorithms are complete, my assignment suggests variations at its implementation in today's software. In the end, an example is solved using the algorithm to depict an understanding of the topic.

1 Introduction

Sylvester matrix equation is a special case of the general linear equation $\sum_{i=1}^{p} A_i X B_i = E$ which is given as

$$AXB + CXD = E \tag{1}$$

in which all matrices are real (can be extended to complex valued case), A and C are $m \times m$ and B and D are $n \times n$ matrices. E is $m \times n$ and the desired solution X is $m \times n$. Sylvester equations are a part of many systems and control theory problems. These equations have important applications in stability analysis, observer design, output regulation problems and eigenvalue assignment.[1]

Many algorithms to solve this equation have been proposed in literature. This home paper assignment while looking at this equation's solvability, also looks at different algorithms and suggests some variations.

2 Solvability of $AXB + CXD = E$

In linear algebra, if $A_0, A_1, \ldots, A_l$ are $n \times n$ complex matrices for some non-negative integer l and $A_l \neq 0$ (the zero matrix), then the matrix pencil of degree l is the matrix-valued function defined on the complex numbers $L(\lambda) = \sum_{i=0}^{l} \lambda^i A_i$. A particular case is a linear matrix pencil is $A - \lambda B$, where $\lambda \in \mathbb{C}$ (or $\mathbb{R}$), A and B are complex (or real) $n \times n$ matrices. A pencil is called regular if there is at least one value of λ such that $\det(A - \lambda B) \neq 0$. We call eigenvalues of a matrix pencil all complex numbers λ for $\det(A - \lambda B) = 0$ The set of the eigenvalues is called the spectrum of the pencil and is denoted $\sigma(A, B)$.[2]

Now, it is proved that Equation (1) has a unique solution if and only
(i) if the matrix pencils $A + \lambda C$ and $D - \lambda B$ are regular
(ii) the spectrum of $A + \lambda C$ is disjoint from the spectrum of $D - \lambda B$, i.e.,
$\sigma(A, -C) \ \cap \ \sigma(D, B) = \emptyset$. [3]

The statement can be thought of in a short and concise way. We consider the equations

$$(\lambda_1 A + \lambda_2 C)XB + CX(\lambda_1 D - \lambda_2 B) = G \tag{2}$$

$$(\lambda_1 A + \lambda_2 C)XD - AX(\lambda_1 D - \lambda_2 B) = -F \tag{3}$$

for some real λ_1, and λ_2, which are not both zero. One of the equations (2) or (3) is equivalent to Equation (1). If conditions (i) and (ii) are satisfied, $\lambda_i' s$ can be found so that the matrices involving the $\lambda_i' s$ are non-singular. Thus, solving equation (2) or (3) is equivalent to solving an equivalent Sylvester equation, which yields a unique solution.

If either (or both) of the conditions (i) and (ii) are violated, some $\lambda_i' s$ can be found such that the matrices $(\lambda_1 A + \lambda_2 C)$ and $(\lambda_1 D - \lambda_2 B)$ are singular. Let $y \neq 0$ and $z \neq 0$ be such that $(\lambda_1 A + \lambda_2 C)y = 0$ and $z^H(\lambda_1 D - \lambda_2 B) = 0$. Then cyz^H, for any nonzero constant c, will be a nontrivial solution of the homogeneous equation related to equation (2) or (3). As a result, a solution cannot be unique, if it exists at all.

There are many algorithms to solve the equation. We will first look at the brute force method.

3 Brute Force

The Kronecker product matrix $M \otimes N$ is the block matrix whose (i, j) block $m_{ij}N$. To solve Equation (1) by Brute Force[4], we rewrite it in the standard form as a linear matrix-vector system:

$$Py = q \tag{4}$$

where

$$P = B^T \otimes A + D^T \otimes C$$
$$y = (x_{1,1}, x_{2,1}, \ldots, x_{m,1}, x_{1,2}, \ldots, x_{m,n})^T$$
$$q = (e_{1,1}, e_{2,1}, \ldots, e_{m,1}, e_{1,2}, \ldots, e_{m,n})^T$$

Equation (4) can be solved by Gaussian elimination. However, since matrix P has dimensions $mn \times mn$, this approach becomes time consuming and impractical except for small systems. Considering today's space or control systems, this algorithm is not put to use much. So, instead let's modify the equation a bit. If B and C are non-singular matrices, we can pre-multiply both sides by C^{-1} and post-multiply by B^{-1}, we get

$$C^{-1}AX + XDB^{-1} = C^{-1}EB^{-1} \tag{5}$$

So, now it is in the standard form $FX + XG = H$ and can be solved using standard algorithms.[5][6] However, this works only when both B and C are non-singular, otherwise we cannot define B^{-1} and C^{-1}. Hence, a more general method is desirable.

4 Generalized Bartels-Stewart Method

The Bartel-Stewart method is a transformation method developed for solving equations of the form $AX + XB = C$.[5]. However, for our equation we need to generalize it which we do so by dividing in 3 main steps:

- Transformation to the reduced form
- Solution of the reduced equation
- Back transformation of the solution

To understand what the above steps mean, first let's extend it to a more general case by rewriting equation (1) as:

$$(Q_1AZ_1)(Z_1^TXZ_2)(Z_2^TBQ_2^T) + (Q_1CZ_1)(Z_1^TXZ_2)(Z_2^TDQ_2^T) = Q_1EQ_2^T \tag{6}$$

where Q_1, Z_1, Q_2, and Z_2 are orthogonal matrices. Moler and Stewart's QZ algorithm states that for a matrix eigenvalue problem $Ax = \lambda Bx$, there are unitary matrices Q and Z so that QAZ and QBZ are both upper-triangular.[7] Applying QZ to matrix pairs (A, C) and (D, B), we can have $Q_1AZ_1 = P$ and $Q_2D^TZ_2 = T$ as quasi-upper-triangular and $Q_1CZ_1 = S$

and $Q_2 B^T Z_2 = R$ as upper-triangular matrices. A quasi-upper-triangular matrix is a block upper-triangular matrix where the blocks on the diagonal are 1×1 or 2×2. Complex eigenvalues are found as the complex eigenvalues of those 2×2 blocks on the diagonal. 1×1 blocks corresponding to the real eigen values.[8] Defining $Z_1^T X Z_2 = Y$ and $Q_1 E Q_2^T = F$, we get:

$$PYR^T + SYT^T = F \tag{7}$$

Since this equation now has a special triangular structure, it can be solved by back substitution technique. Let a_k denote the kth column of the matrix A. By this notation, for kth column on each side of equation (7), we get:

$$P\sum_{j=k}^{n} r_{k,j} y_j + S\sum_{j=k-1}^{n} t_{k,j} y_j = f_k \tag{8}$$

The summation ranges reflect the upper-triangular nature of R and quasi-upper-triangular nature of T. This can be rewritten as:

$$(r_{k,k}P + t_{k,k}S)y_k + t_{k,k-1}Sy_{k-1} = f_k - \sum_{j=k+1}^{n} (r_{k,j}P + t_{k,j}S)y_j = f_k^{n-k} \tag{9}$$

where f_k^J can be defined by recursion as:

$$\begin{aligned} f_k^0 &= f_k \\ f_k^J = f_k^{J-1} &- r_{k,J} P y_J - t_{k,J} S y_J \end{aligned}$$

Equation (7) can thus be solved columnwise by starting from the last (nth) column and moving back to the first. Finally, X can be obtained by $X = Z_1 Y Z_2^T$.
Due to its lower complexity, this method can be used for solving equations with large order matrices. $\mathcal{O}(m^3+n^3)$ is the complexity of this method. It can also be implemented parallelly, thus giving fast results.

5 Modified Hessenburg-Schur Method

Another method to solve the equation $AX + XB = C$ is the Hessenburg-Schur method.[6] In this method, the larger of the two matrices A or D, say A is bigger as an example, is reduced only to upper Hessenburg form, while D is again reduced to quasi-upper-triangular form. As in the Bartels-Stewart Method, B and C are reduced to upper-triangular matrices. An upper Hessenberg matrix has zero entries below the first subdiagonal. Basically, the major difference between the two methods is that only the first of QZ algorithm is applied on (A, C), i.e., A is reduced to upper Hessenburg only while C is reduced to upper-triangular. For (D, B) the algorithm runs till end like the previous method. The rest of the steps are same as the Bartels-Stewart Method, except the number of subdiagonal elements on the LHS of equation (9) changes accordingly.

The resulting savings in computation time on the first QZ call to more than offset the subsequent complication of the back substitution step.[4] Although the complexity is the same as previous method, the number of flops is lesser. In fact, the subroutines required for the Hessenberg-Schur variant of the Bartels-Stewart algorithm are used in the MATLAB control system toolbox.

6 Example

Let's consider a small example.

$$\begin{pmatrix}1 & 2\\ 2 & 1\end{pmatrix} X \begin{pmatrix}1 & 2\\ 1 & 2\end{pmatrix} + \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} X \begin{pmatrix}-1 & 2\\ 3 & 0\end{pmatrix} = \begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}$$

By doing QZ factorization of A and C, we get

$$Q_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & -1\\ 1 & 1\end{pmatrix} \qquad Z_1 = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ -1 & 1\end{pmatrix}$$

(since the example is 2×2 not much effort is needed to make both upper triangular. For higher order matrices, we stop at quasi so as to save computation.)
By doing QZ factorization of D and B, we get

$$Z_2 = \frac{1}{\sqrt{13}}\begin{pmatrix}3 & -2\\ 2 & 3\end{pmatrix} \qquad Q_2 = \frac{1}{\sqrt{5}}\begin{pmatrix}1 & 2\\ 2 & -1\end{pmatrix}$$

Multiplying, we obtain

$$P = Q_1 A Z_1 = \begin{pmatrix}-1 & 0\\ 0 & 3\end{pmatrix} \qquad S = Q_1 C Z_1 = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

$$R^T = Z_2^T B Q_2^T = \frac{1}{\sqrt{65}}\begin{pmatrix}25 & 0\\ 5 & 0\end{pmatrix} \qquad T^T = Z_2^T D Q_2^T = \frac{1}{\sqrt{65}}\begin{pmatrix}15 & 0\\ 3 & 26\end{pmatrix}$$

Similarly,

$$F = Q_1 E Q_2^T = \frac{1}{\sqrt{10}}\begin{pmatrix}1 & 2\\ 5 & 0\end{pmatrix}$$

Now, moving to next step in the method, i.e., solving for nth, or 2nd column of Y:

$$\begin{aligned}(r_{2,2}P + t_{2,2}S)y_2 &= f_2\\ y_2 &= \frac{1}{\sqrt{26}}\begin{pmatrix}1 & 0\end{pmatrix}^T\\ (r_{1,1}P + t_{1,1}S)y_1 &= f_1 - (r_{1,2}P + t_{1,2}S)y_2\\ y_1 &= \frac{1}{18\sqrt{26}}\begin{pmatrix}-27 & 13\end{pmatrix}^T\end{aligned} \tag{10}$$

Finally,

$$X = Z_1 Y Z_2^T = \frac{1}{18}\begin{pmatrix}-3 & 1\\ 6 & 1\end{pmatrix}$$

References

[1] Y. Feng, M. Yagoubi, "Robust Control of Linear Descriptor Systems"

[2] https://en.wikipedia.org/wiki/Matrix_pencil

[3] E. Chu, "The solution of the matrix equations $AXB - CXD = E$ and $(YA - DZ, YC - BZ) = (E, F)$," Linear Algebra and its Applications, August, 1987

[4] J. Gardiner, A. Laub, J. Amato, C. Moler, "Solution of the Sylvester Matrix Equation $AXB^T + CXD^T = E$," ACM Transactions on Mathematical Software, June, 1992

[5] R. H. Bartels, G. W. Stewart, "Solution of the equation $AX + XB = C$", Communications of the ACM, September, 1972

[6] G. H. Golub, S. Nash, C. F. Van Loan, "A Hessenburg-Schur method for the problem $AX + XB = C$", IEEE Transactions on Automatic Control, December, 1979

[7] C. B. Moler, G. W. Stewart "An algorithm for generalized matrix eigen value problems", SIAM Journal on Numerical Analysis, April, 1973

[8] https://pdfs.semanticscholar.org/906e/406b81ee623ea132e085652ef93c7bb4d12f.pdf