Matrix Equations and Tensor Techniques
IX
Perugia, September 9-10, 2021

A Successive Divide and Conquer Algorithm for the Unilateral Quadratic Matrix Equation

Peter Benner *Martin Köhler*

Max Planck Institute for Dynamics of Complex Technical Systems,
Sandtorstr. 1, D-39106 Magdeburg, Germany

The Unilateral Quadratic Matrix Equation (UQME)

$$AX^2 + BX + C = 0 \tag{1}$$

is a rarely mentioned equation in numerical linear algebra. Only a few publications and algorithms cover the solution of this equation. Beside a direct solved based on the underlying quadratic eigenvalue problem, only the Bernoulli Iteration and Newton's Method are known [2, 3] for general matrices A, B, and C.

In our contribution, we develop an algorithm using the underlying eigenvalue problem

$$Fy = \begin{bmatrix} 0 & I \\ -C & -B \end{bmatrix} y = \lambda \begin{bmatrix} I & 0 \\ 0 & A \end{bmatrix} y = \lambda G y, \quad y = \begin{bmatrix} v \\ \lambda v \end{bmatrix}, \tag{2}$$

and the connection that if X is a solution of the UQME (1), it can be written as

$$X = Z_{21} Z_{11}^{-1}. \tag{3}$$

Thereby, the matrices Z_{21} and Z_{11} originate from the generalized Schur decomposition of the matrix pair (F, G):

$$\left(Q^H F Z, Q^H G Z\right) = \left(\begin{bmatrix} T_{11} & T_{12} \\ & T_{22} \end{bmatrix}, \begin{bmatrix} S_{11} & S_{12} \\ & S_{22} \end{bmatrix} \right) = (T, S) \tag{4}$$

with

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \text{ and } Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}, \tag{5}$$

where each block is of size $m \times m$. Since there is no predefined order of the eigenvalues on the diagonals of (T, S), a variety of solutions X exists. Typically, the *minimal* solution, which corresponds to the m smallest eigenvalues, or the *dominant* solution, derived using the deflating subspace related to the m largest eigenvalues, are of interest. Obtaining these solutions with the computation of the generalized Schur decomposition (4) and a subsequent eigenvalue reordering is a time-consuming procedure due to the complexity and the behavior of the QZ algorithm. Obviously, it is sufficient to compute the block decomposition as shown in Equation (4), *without* having (T_{11}, S_{11}) and (T_{22}, S_{22}) in (quasi) upper triangular form. Only the zero block in the lower left of (4) needs to be of size $m \times m$. Furthermore, if we assume that we want to compute the minimal solution, the blocks of (T, S) have to fulfill the following condition:

$$|\lambda| < \ |\mu| \quad \forall \lambda \in \Lambda \left(T_{11}, S_{11}\right), \, \mu \in \Lambda \left(T_{22}, S_{22}\right). \tag{6}$$

Our Successive Divide and Conquer algorithm (SDC) computes a sequence of unitary transformations (Q_k, Z_k) with

$$Q = Q_1 \cdots Q_p \quad \text{and} \quad Z = Z_1 \cdots Z_p \tag{7}$$

until the block partitioning (4)/(5) fulfills the condition (6) and (T_{11}, S_{11}) are of size $m \times m$. Thereby, the sequence of transformation matrices (Q_k, Z_k) is computed from deflating subspaces, which are computed with the help of the matrix disc function [1] and a dedicated scaling technique [4]. In this way, the algorithm successively transforms the matrix pair (F, G) as long as necessary.

If the UQME (1) is given with real data and the desired solution should be real as well, a minimal solution does not exist if there a complex conjugate eigenvalue pair would be split when computing the block decomposition (4). In order to overcome this issue, we relax this to computing the quasi minimal solution, which differs from the minimal solution in one missing real eigenvalue.

The numerical experiments show that our algorithm is a fast replacement for solving the problem with the QZ algorithm. Even with a pure naive MATLAB implementation of our algorithm, we obtain a speed up of 5.5 to 10 compared to the QZ algorithm. The comparison to the Bernoulli iteration strongly depends on its convergence behavior. Especially, if the gap between the spectra of (T_{11}, S_{11}) and (T_{22}, S_{22}) in (4) gets very small, our algorithms comes up with a much faster solution. Here, we obtain a speed up of 1.25 with a naive MATLAB implementation of the SDC algorithm. Finally, we show an example for the quasi minimal solution, where the Bernoulli iteration does not converge at all.

References

[1] P. Benner. *Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems*. Dissertation, Fakultät für Mathematik, TU Chemnitz–Zwickau, 09107 Chemnitz (Germany), February 1997.

[2] N. J. Higham and H.-M. Kim. Numerical analysis of a quadratic matrix equation. *IMA J. Numer. Anal.*, 20(4):499–519, 2000.

[3] N. J. Higham and H.-M. Kim. Solving a quadratic matrix equation by Newton's method with exact line searches. *SIAM J. Matrix Anal. Appl.*, 23(2):303–316, 1 2001.

[4] M. Köhler. *Approximate Solution of Non-Symmetric Generalized Eigenvalue Problems and Linear Matrix Equation on HPC Platforms*. Dissertation, Department of Mathematics, Otto-von-Guericke University, Magdeburg, Germany, 2021.