The 11th International Seminar on
Linear Algebra and its Applications
Feb 2022

Three-Term-Tensor Sylvester method for a class of third order tensor linear equations

Mohammad Mahdi Izadkhah[1*]

[1]Department of Computer Science, Faculty of Computer and Industrial Engineering,
Birjand University of Technology, Birjand, Iran

Abstract

In this paper, we present a direct dense method called three-term-tensor Sylvester, to obtain the solution to a class of third order tensor linear equations. The proposed method investigate process of solution without the explicit use of Kronecker form that is desirable for low rank tensor equations. Numerical experiments illustrate the properties of the considered algorithm.

Keywords: Tensors, Schur decomposition, Generalized Sylvester matrix equation

Mathematics Subject Classification [2010]: 65F10, 65F30, 15A24

1 Introduction

The literatures on tensors, actually about their analysis and the associated approximation methods, has grown tremendously in the past twenty years. Numerous different decompositions of tensor equations have allowed the developments of various problem dependent strategies, see [2–4] and references therein.

In this paper, we are interested in the computation of the unique solution $\mathcal{X} \in \mathbb{R}^{n\times n\times n}$ to the nonsingular system in the following tensor form

$$(\mathbf{H} \otimes A_1 \otimes M_1 + \mathbf{H} \otimes A_2 \otimes \mathbf{M} + H_3 \otimes A_3 \otimes \mathbf{M})\text{vec}(\mathcal{X}) = b_3 \otimes b_2 \otimes b_1, \tag{1}$$

where all coefficient matrices are real and have the same $n \times n$ dimensions. Here $\otimes$ denotes the Kronecker product (to be recalled later) and $\text{vec}(\mathcal{X})$ stacks the components of the tensor $\mathcal{X}$ one after the other. In particular, in (1) two terms share the same matrices, either $\mathbf{M}$ or $\mathbf{H}$ (purposely in bold face), while all other matrices $A_i, i = 1, 2, 3$ and H_3, M_1 have no relation to each other.

The numerical solution to (1) can be given in closed form by unfolding the 3-mode tensor in one of the three directions. In particular, a tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ can be written using the mode-1 unfolding as in [2]

$$\mathcal{X}_{(1)} = [X_1, X_2, \ldots, X_{n3}], \quad X_k = \mathcal{X}_{::k} \in \mathbb{R}^{n_1\times n_2}, \quad k = 1, 2, \ldots, n_3$$

*Speaker. Email address: izadkhah@birjandut.ac.ir

each X_k is called a frontal slice of tensor $\mathcal{X}$, and $\mathcal{X}_{(1)}$ is a matrix in $\mathbb{R}^{n_1 \times n_2 n_3}$. Some additional standard notation needs to be recalled. The Kronecker product of two matrices X and Y is defined in the block form as

$$X \otimes Y = \begin{pmatrix} X_{1,1}Y & \cdots & X_{1,n_2}Y \\ \vdots & \ddots & \vdots \\ X_{n_1,1}Y & \cdots & X_{n_1,n_2}Y \end{pmatrix},$$

where $X_{i,j}$ denotes an element of X. Moreover, $\mathrm{vec}(X)$ is the operator stacking all columns of the matrix X one after the other. In the case of third order tensors $\mathcal{X}$, we will apply the $\mathrm{vec}(\mathcal{X})$ operator to the mode-1 unfolding $\mathcal{X}_{(1)}$, that is

$$\mathrm{vec}(\mathcal{X}) := \mathrm{vec}(\mathcal{X}_{(1)}).$$

The reverse operation, for known dimensions of the vector x, will be denoted by $\mathsf{mat}(x, n_1, n_2)$, so that $x = \mathrm{vec}(X)$and $X = \mathsf{mat}(x, n_1, n_2)$. Similarly, $\mathcal{X} = \mathsf{tensor}_{(1)}(x, n_1, n_2, n_3)$ will fold a long vector x into a tensor $\mathcal{X}$ via the mode-1 unfolding.

A standard property of the Kronecker product that will be used repeatedly is the following

$$\mathrm{vec}(AXB) = (B^T \otimes A)\mathrm{vec}(X), \tag{2}$$

where B^T denotes the real transpose of B. The aforementioned equation (2) allows one to go back and forth between the vector and matrix presentation. Other properties used in the sequel are from [1] as

$$i)\ (A \otimes B)^T = A^T \otimes B^T, \quad ii)\ (A \otimes B)(C \otimes D) = (AB \otimes CD), \quad iii)\ (A \otimes B)^{-1} = A^{-1} \otimes B^{-1},$$

Also, Q^* denotes the conjugate transpose of the complex matrix Q, while $H^{-T} = (H^{-1})^T$.

Least approaches try to take *direct* dense methods for low order tensor equations, without the explicit use of the Kronecker product. Here we concentrate on special case of tensor equation (1) motivated from [5], nonetheless appears to be a feasible algebraic formulation of a quite large set of differential problems.

2 Three-Term-Tensor Sylvester method

In this section, we use the mode-1 unfolding related to the specific location of the repeated matrices $\mathbf{H}$ and $\mathbf{M}$ in (1). The following Theorem provides the unique solution to (1) based on the generalized Sylvester matrix equation.

Theorem 2.1. *Let* $(\mathbf{H}^{-1}H_3)^T = QRQ^*$ *be the Schur decomposition of* $(\mathbf{H}^{-1}H_3)^T$*, and* $[\gamma_1, \ldots, \gamma_n] := b_3^T \mathbf{H}^{-T} Q$*. Using the mode-1 unfolding, the solution* $\mathcal{X}$ *to* (1) *is given by*

$$\mathcal{X} = \mathsf{tensor}_{(1)}(\mathrm{vec}([\hat{Z}_1 Q^{-1}, \ldots, \hat{Z}_n Q^{-1}]), n, n, n),$$

where for $j = 1, \ldots, n$*, the matrix* $\hat{Z}_j$ *solves the generalized Sylvester equation*

$$M_1 Z A_1^T + \mathbf{M} Z (R_{j,j} A_3^T + A_2^T) = b_1 \gamma_j b_2^T - \mathbf{M}\mathsf{mat}([\hat{z}_1, \ldots, \hat{z}_{j-1}] R_{1:j-1,j}, n, n) A_3^T,$$

where $R_{j,j}$ *stands for the diagonal entries of the upper triangular matrix* R*, and* $R_{1:j-1,j}$ *denotes the first* $j-1$ *components of* j*-th column of* R*. We set* $\mathsf{mat}([\hat{z}_1, \ldots, \hat{z}_{j-1}] R_{1:j-1,j}, n, n)$ *to be empty array for* $j = 1$*.*

Proof. Using (2) for the unfolded tensor we have

$$(A_1 \otimes M_1 + A_2 \otimes \mathbf{M})\mathcal{X}_{(1)}\mathbf{H}^T + (A_3 \otimes \mathbf{M})\mathcal{X}_{(1)}H_3^T = (b_2 \otimes b_1)b_3^T$$
$$(A_3 \otimes \mathbf{M})^{-1}(A_1 \otimes M_1 + A_2 \otimes \mathbf{M})\mathcal{X}_{(1)} + \mathcal{X}_{(1)}H_3^T\mathbf{H}^{-T} = (A_3 \otimes \mathbf{M})^{-1}(b_2 \otimes b_1)b_3^T\mathbf{H}^{-T}$$
$$(A_3^{-1}A_1 \otimes \mathbf{M}M_1 + A_3^{-1}A_2 \otimes I)\mathcal{X}_{(1)} + \mathcal{X}_{(1)}H_3^T\mathbf{H}^{-T} = (A_3^{-1}b_2 \otimes \mathbf{M}^{-1}b_1)b_3^T\mathbf{H}^{-T}$$

Using $(\mathbf{H}^{-1}H_3)^T = QRQ^*$ and multiplying the equation by Q from the right, we get

$$(A_3^{-1}A_1 \otimes \mathbf{M}M_1 + A_3^{-1}A_2 \otimes I)\mathcal{X}_{(1)}Q + \mathcal{X}_{(1)}QR = (A_3^{-1}b_2 \otimes \mathbf{M}^{-1}b_1)b_3^T\mathbf{H}^{-T}Q$$

Let $\mathcal{X}_{(1)}Q = [\hat{z}_1, \dots, \hat{z}_n]$ and $[\gamma_1, \dots, \gamma_n] := b_3^T\mathbf{H}^{-T}Q$. Thanks to the upper triangular form of R, for the first column $\hat{z}_1$ it holds

$$(A_3^{-1}A_1 \otimes \mathbf{M}^{-1}M_1 + A_3^{-1}A_2 \otimes I)\hat{z}_1 + \hat{z}_1R_{1,1} = (A_3^{-1}b_2 \otimes \mathbf{M}^{-1}b_1)\gamma_1$$

For the subsequent columns $j = 2, \dots, n$, taking into account once again the triangular form of R, we set $w_{j-1} = [\hat{z}_1, \dots, \hat{z}_{j-1}]R_{1:j-1,j}$ so that

$$(A_3^{-1}A_1 \otimes \mathbf{M}^{-1}M_1 + A_3^{-1}A_2 \otimes I)\hat{z}_j + \hat{z}_jR_{j,j} = (A_3^{-1}b_2 \otimes \mathbf{M}^{-1}b_1)\gamma_j - w_{j-1}$$

Let us reshape each $\hat{z}_j$ so that $\hat{Z}_j = \mathsf{mat}(\hat{z}_j, n, n)$. For $j = 1$, we can write

$$\mathbf{M}^{-1}M_1\hat{Z}_1 + \hat{Z}_1(R_{1,1}A_3^TA_1^{-T} + A_2^TA_1^{-T}) = \mathbf{M}^{-1}b_1\gamma_1b_2^TA_1^{-T}$$

Analogously, for $j = 2, \dots, n$ and letting $W_{j-1} = \mathsf{mat}([\hat{z}_1, \dots, \hat{z}_{j-1}]R_{1:j-1,j}, n, n)$, from (2.3) we first obtain

$$\mathbf{M}^{-1}M_1\hat{Z}_j(A_3^{-1}A_1)^T + \hat{Z}_j(R_{j,j}I + (A_3^{-1}A_2)^T) = \mathbf{M}^{-1}b_1\gamma_j(A_3^{-1}b_2)^T - W_{j-1}$$

or equivalently, for $j = 2, \dots, n$

$$\mathbf{M}^{-1}M_1\hat{Z}_j + \hat{Z}_j(R_{j,j}A_3^TA_1^{-T} + A_2^TA_1^{-T}) = \mathbf{M}^{-1}b_1\gamma_jb_2^TA_1^{-T} - W_{j-1}A_3^TA_1^{-T}$$

Multiplying both sides by $\mathbf{M}$ (from the left) and by A_1^T (from the right), the result follows. □

Based on the proof of Theorem 2.1, an explicit constructive way is provided to generate the tensor solution, one slice at the time. The complete procedure is described in the algorithm below, in the following called the Three-Term-Tensor Sylvester ($\mathrm{T}^3 - \textsc{Sylv}$) method.

Algorithm 2.2. $\mathrm{T}^3 - \textsc{Sylv}$

1. **Input:** Matrices $A_1, A_2, A_3, M_1, \mathbf{M}, \mathbf{H}, H_3$ of size $N \times n$, vectors b_1, b_2, b_3 of length n.
2. For $k = 1, \dots, n$

   Compute Q and R such that $(\mathbf{H}^{-1}H_3)^T = QRQ^*$ (Schur decomposition)

   Compute $g = b_3^T\mathbf{H}^{-T}Q$

   Set $F = \mathbf{M}^{-1}b_1g_kb_2^TA_1^{-T}$

   If $k > 1$, Set $W_{k-1} = \mathsf{mat}([\hat{z}_1, \dots, \hat{z}_{k-1}]R_{1:k-1,k}, n, n)$ and $F = F - W_{k-1}A_3^TA_1^{-T}$

   Solve $\mathbf{M}^{-1}M_1\hat{Z} + \hat{Z}(R_{k,k}A_3^TA_1^{-T} + A_2^TA_1^{-T}) = F$ to get $\hat{Z}_k$ and $\hat{z}_k = vec(\hat{Z}_k)$

   end

3. Set $\mathcal{X} = \mathsf{tensor}_{(1)}(\mathrm{vec}([\hat{Z}_1 Q^{-1}, \ldots, \hat{Z}_n Q^{-1}]), n, n, n)$
4. **Output:** $\mathcal{X}$ solution to (1)

In practice, using appropriate transformations, the method is a nested Sylvester solver, which treats one slice at the time, and updates the corresponding coefficient matrix and right-hand side F. The solvability of the Sylvester equations is related to that of the original problem, and in particular to the nonsingularity of $\mathcal{A}$. The algorithm relies on the initial Schur decomposition, which provides robust unitary transformations.

Moreover, for each slice, a matrix Sylvester equation needs to be solved, whose solution also involves the Schur decompositions of the coefficient matrices, as a small scale computation studied in [4]. Indeed, if some of the involved matrices are severely ill-conditioned, the solution may lose accuracy.

3 Numerical experiments

In this section, we report some numerical experiments with the $\mathrm{T}^3 - \mathrm{SYLV}$ method. All experiments were performed using MATLAB R2015a on an Intel Core i7 Laptop with 8G RAM.

Example 3.1. To test the efficiency of the proposed method, we consider dense matrices with random entries (taken from a uniform distribution in the interval $(0, 1)$, MATLAB function `rand`) of increasing size n. Also, the same is used for the vectors b_1, b_2 and b_3. We stress that the Kronecker form of the problem would involve a dense matrix $\mathcal{A}$ of size $n^3 \times n^3$, which could not even be stored.

Without difficulty, we observe that the method is able to solve a (random) structured dense problem of size $n^3 = 16,777,216$ in about $CPU_{Time} = 25$ seconds. The CPU times in Table 1 show that the computational cost of the method approximately grows between six and ten times as the dimension n doubles. However, going from n to $2n$, the problem dimension in the full space would grow from n^3 to $2^3 n^3$. Hence, the actual cost appears to grow linearly with n^3. Since data are dense, Gauss elimination on $\mathcal{A}$ would instead require $\mathcal{O}((n^3)^3)$ floating point operations.

Table 1: CPU times of $\mathrm{T}^3 - \mathrm{SYLV}$ for increasing dimensions of the coefficient matrices, having uniformly distributed random entries

n	CPU Time
8	2.47e-04
16	4.35e-03
32	2.15e-02
64	1.14e-01
128	2.15e+00
256	2.53e+01

4 Conclusion

We have proposed three-term-tensor Sylvester method for solving a class of third order tensor linear equations. The method relied on the Schur decomposition for solving a generalized Sylvester matrix equation to obtain frontal slices od tensor solution, at each time. In fact, the repeated presence of two matrices in the considered third order tensor linear equations make the most of the method, without Kronecker form.

References

[1] G. Golub, C.F. Van Loan, Matrix Computations, 4th edn. The Johns Hopkins University Press, Baltimore (2013).

[2] T.G., Kolda, B.W., Bader, Tensor decompositions and applications. SIAM Rev. 51 (2009) 455–500.

[3] H.G. Matthies, E. Zander, Solving stochastic systems with low-rank tensor compression. Linear Algebra Appl. 436 (2012) 3819–3838.

[4] V. Simoncini, Computational methods for linear matrix equations, SIAM Rev. 58(3)(2016) 377–441.

[5] V. Simoncini, Numerical solution of a class of third order tensor linear equations, Bollet. dell'Union Matematica Italiana 13 (2020) 429-439.