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

A new preconditioning technique for SOR algorithm for solving multi-linear systems

Afsaneh Hasanpour[1] and Maryam Mojarrab[2*]

[1]Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran

[2]Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran

Abstract

Preconditioning techniques are useful procedures in increasing the rate of convergence of iterative methods and in some cases in eliminating possible stagnation in solving multi-linear systems with nonsingular $\mathcal{M}$-tensors. In this paper, we propose a novel preconditioner obtained by minimizing the norm of the iteration tensor. We also consider a preconditioned SOR iterative method for solving tensor equations whose coefficient tensor is an $\mathcal{M}$-tensor. Numerical examples, and comparison results are given to show the efciency of the preconditioner.

Keywords: Preconditioner, Multi-linear systems, $\mathcal{M}$-tensor, SOR method

Mathematics Subject Classification [2010]: 65F08, 65F10

1 Introduction

Tensor equations have many applications in engineering, and scientic computing [2], such as evolutionary game dynamics [5], partial differential equations, and data mining [1], and image processing [3].

Consider the following tensor equation of the form

$$\mathcal{A}x^{m-1} = b, \tag{1}$$

where $\mathcal{A} \in \mathbb{R}^{[m,n]}$ is an order m dimension n tensor, x, and b are vectors in $\mathbb{C}^n$. The tensor-vector product is a vector where the entries are defined by

$$(\mathcal{A}x^{m-1})_i = \sum_{i_2 i_3 \cdots i_n = 1}^{n} a_{i i_2 i_3 \cdots i_n} x_{i_2} x_{i_3} \cdots x_{i_n}, \qquad i = 1, 2, \cdots, n,$$

where x_i denotes the ith component of x. It can be seen that multi-linear systems are made up of a series of non-linear equations.

Many theoretical analyses, and algorithms were presented for solving (1). It is proofed that (1) will have a unique positive solution if $\mathcal{A}$ be a nonsingular $\mathcal{M}$-tensor, and b be a positive vector. In addition, some conditions were presented for the existence and uniqueness of the solution of (1).

*Speaker. Email address: ma_mojarrab@math.usb.ac.ir

The role of the preconditioning technique is clear in solving linear and multi-linear systems, which can improve the convergence rate of the method if a suitable preconditioner is chosen. Lots of efficient preconditioners were proposed to solve linear systems. Although a little research introduced preconditioned methods for solving multi-linear systems. Li et al. [6] proposed the preconditioned tensor splitting method for solving the following preconditioned multi-linear systems (1):

$$P\mathcal{A}x^{m-1} = Pb,$$

where P is a preconditioner and the iterative scheme is as follows:

$$x_k = (M(\mathcal{E}_P)^{-1}\mathcal{F}_P x_{k-1}^{m-1} + M(\mathcal{E}_P)^{-1}Pb)^{[\frac{1}{m-1}]}, \qquad k = 1, 2, \cdots,$$

so that $P\mathcal{A} = \mathcal{E}_P - \mathcal{F}_P$ is a tensor splitting of $P\mathcal{A}$. A modified preconditioned Gauss-Seidel method was proposed [4].

In this paper, we proposed a diagonal preconditioner for the SOR method to solve multi-linear systems. We apply the new preconditioned SOR method to some Numerical examples and compare the new method to the original SOR method. Numerical experiments and comparison results confirm the power of the preconditioner.

This paper is organized as follows. In Section 2, a new preconditioner is introduced, and the preconditioned SOR method is constructed. Section 3 consist of some numerical examples which demonstrate the efficiency of the presented preconditioned iterative method. The final section consists of conclusion.

2 Main results

In this section, we propose a diagonal preconditioner by minimizing the Frobenius norm of $\mathcal{I}_m - P\mathcal{A}$. Consider the following multi-linear system:

$$\mathcal{A}x^{m-1} = b, \tag{2}$$

where $\mathcal{A} \in \mathbb{R}^{[3,n]}$, x and b are vectors in $\mathbb{C}^n$. Applying a nonsingular matrix P as a preconditioner, we have

$$P\mathcal{A}x^{m-1} = Pb. \tag{3}$$

We can add x^{m-1} to both sides of (3). So

$$x^{m-1} = (\mathcal{I}_m - P\mathcal{A})x^{m-1} + Pb,$$

which implies

$$x_{k+1}^{m-1} = (\mathcal{I}_m - P\mathcal{A})x_k^{m-1} + Pb.$$

When we choose $P = \alpha I$, a constant diagonal matrix will appear. So the iteration transforms to the Richardson iteration.

$$x_{k+1}^{m-1} = (\mathcal{I}_m - \alpha\mathcal{A})x_k^{m-1} + \alpha b,$$

where α is the value of the diagonal entries.

We use the Frobenius norm, and we want to make a diagonal preconditioning matrix P. We denote the set of a diagonal matrix of size n by Δ_n. The optimization problem will be as

$$\min_{P\in\Delta_n} \|\mathcal{I}_m - P\mathcal{A}\|_F, \qquad or \qquad \min_{P\in\Delta_n} \|\mathcal{I}_m - P\mathcal{A}\|_F^2.$$

Finding the entries of P, we define the preconditioner by

$$P = \begin{pmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_n \end{pmatrix},$$

then

$$\min_{\alpha_1,\alpha_2,\cdots,\alpha_n} \|\mathcal{I}_m - P\mathcal{A}\|_F.$$

The tensor $P\mathcal{A}$ is a tensor where the ith row of each frontal slice of $\mathcal{A}$ is multiplied by α_i, because P is diagonal. We call it row scaling.

By the definition of the Frobenius norm and t-product, we have

$$\begin{aligned} \|\mathcal{I}_m - P\mathcal{A}\|_F^2 &= tr\{[(\mathcal{I}_m - P\mathcal{A}) * (\mathcal{I}_m - P\mathcal{A})^T]_{(:,:,1)}\} \\ &= tr\{[\mathcal{I}_m * \mathcal{I}_m^T - \mathcal{I}_m * (P\mathcal{A})^T - (P\mathcal{A}) * \mathcal{I}_m + (P\mathcal{A}) * (P\mathcal{A})^T]_{(:,:,1)}\} \\ &= 1 - 2tr(\mathcal{I}_m * (P\mathcal{A})^T)_{(:,:,1)} + tr((P\mathcal{A}) * (P\mathcal{A})^T)_{(:,:,1)} \end{aligned}$$

where

$$tr(\mathcal{I}_m * (P\mathcal{A})^T)_{(:,:,1)} = \sum_{i=1}^{n} \alpha_i a_{iii},$$

and

$$tr((P\mathcal{A}) * (P\mathcal{A})^T)_{(:,:,1)} = \sum_{i=1}^{n} \alpha_i^2 \|\mathcal{A}_{(i,:,:)}\|_2^2.$$

Therefore

$$\|\mathcal{I}_m - P\mathcal{A}\|_F^2 = 1 - 2\sum_{i=1}^{n} \alpha_i a_{iii} + \sum_{i=1}^{n} \alpha_i^2 \|\mathcal{A}_{(i,:,:)}\|_2^2.$$

$\|\mathcal{I}_m - P\mathcal{A}\|_F^2$ is a convex function in the α_i, and we can find the minimize. We can set the partial derivatives with respect to $\alpha_i,\ i = 1, 2, \cdots, n$ equal to zero. Thus

$$\frac{d}{d\alpha_i}\|\mathcal{I}_m - P\mathcal{A}\|_F^2 = -2\sum_{i=1}^{n} a_{iii} + 2\sum_{i=1}^{n} \alpha_i \|\mathcal{A}_{(i,:,:)}\|_2^2 = 0.$$

Since α_i is the only varibale of each equation, then

$$\alpha_i = \frac{a_{iii}}{\|\mathcal{A}_{(i,:,:)}\|_2^2}, \quad i = 1, 2, \cdots, n.$$

Accordingly, the diagonal preconditioner which is optimal in the Frobenius norm can be defined as follows

$$P = \begin{pmatrix} \frac{a_{111}}{\|\mathcal{A}_{(1,:,:)}\|_2^2} & & & \\ & \frac{a_{222}}{\|\mathcal{A}_{(2,:,:)}\|_2^2} & & \\ & & \ddots & \\ & & & \frac{a_{nnn}}{\|\mathcal{A}_{(n,:,:)}\|_2^2} \end{pmatrix}.$$

We only consider that the coefcient tensor of (1) is a strong $\mathcal{M}$-tensor. Without loss of generality, we assume that each diagonal entry of the tensor $\mathcal{A}$ is 1. We have the following preconditioned multi-linear system with our new preconditioner:

$$\hat{\mathcal{A}}x^{m-1} = \hat{b},$$

where $\hat{\mathcal{A}} = P\mathcal{A}$ and $\hat{b} = Pb$. Consider

$$\hat{\mathcal{A}} = \hat{\mathcal{D}} - \hat{\mathcal{L}} - \hat{\mathcal{F}}, \quad or \quad \hat{\mathcal{A}} = \hat{\mathcal{I}}_m - \hat{\mathcal{L}} - \hat{\mathcal{F}},$$

with $\hat{\mathcal{D}} = \hat{D}\mathcal{I}_m$, $\hat{\mathcal{L}} = \hat{L}\mathcal{I}_m$, where D is the positive diagonal matrix, $-L$ is the strictly lower triangle matrix of $M(\hat{\mathcal{A}})$. We take the preconditioned SOR method as:

$$x_k = (\mathcal{T}_p x_{k-1}^{m-1} + q_p)^{[\frac{1}{m-1}]}, \qquad k = 1, 2, \cdots,$$

where

$$\begin{aligned}
\mathcal{T}_p &= M(\hat{\mathcal{E}}_p)^{-1}\hat{\mathcal{F}}_p,\\
\hat{\mathcal{E}}_p &= \frac{1}{\omega}(\hat{\mathcal{D}} - \omega\hat{\mathcal{L}}),\\
\hat{\mathcal{F}}_p &= (1-\omega)\hat{\mathcal{D}} - \omega\hat{\mathcal{F}},\\
q_p &= M(\hat{\mathcal{E}}_p)^{-1}b.
\end{aligned}$$

Theorem 2.1. *Let $\mathcal{A} \in \mathbb{R}^{[3,n]}$ be a strong $\mathcal{M}$-tensor. Then for the new preconditioner P, $\hat{\mathcal{A}} = P\mathcal{A}$ is a strong $\mathcal{M}$-tensor.*

3 Numerical results

n this section, we use some numerical experiments to show the effectiveness and superiority of the preconditioned SOR method. The stopping criterion $\| \mathcal{A}x^{m-1} - b \| \leq 10^{-10}$ is used and a maximum of 1000 iterations is allowed. In all the examples, we take the starting vector x_0, and the right hand-side vector b equal to $ones(n, 1)$. Finding the optimal parameter ω, we search from 0.01 to 2 in the interval of 0.01. All the examples were executed in double precision in MATLAB R2014a.

We show the number of iterations by "*Iter*", the norm of $\mathcal{A}x_k^{m-1} - b$ (x_k is the kth approximate solution) in seconds by "*Error*" and the CPU time in third by "*time*" for the preconditioned SOR (PSOR) and the SOR methods, respectively.

The product $\mathcal{A}x^{m-1}$ denoted in (1) can be computed by transforming into the following matrix-vector product:

$$\mathcal{A}x^{m-1} = \mathcal{A}\underbrace{(x \otimes x \otimes \cdots \otimes x)}_{m-1},$$

where $\otimes$ shows the Kronecker product. Also the matrix-tensor product $B\mathcal{A}$ is defined in (2).

Example 3.1. Consider $\mathcal{B} \in \mathbb{R}^{[3,10]}$ as a nonnegetive tensor with

$$b_{ijk} = | \, tan(i + j + k) \, | \, .$$

We have $\rho(\mathcal{B}) = 1450.3$. Thus, $\mathcal{A} = 1500\mathcal{I} - \mathcal{B}$ is a symmetric nonsingular $\mathcal{M}$-tensor.

We take $b = ones(10, 1)$ and initial vector $x_0 = ones(10, 1)$. For different ω, we compare the presented PSOR method and SOR method for solving a nonsingular tensor equation. The results are shown in Table 1. We show the results for different amounts of ω. In this example, we could understand that the $\omega = 1$ is the optimal value of this parameter, which means that preconditioned Jacobian method performs better than the preconditioned SOR method. The comparison results demonstrate that the preconditioned method could be more efcient than the original method.

Table 1: *Numerical results of Example 3.1.*

	PSOR			SOR		
ω	Iter	Error	time	Iter	Error	time
0.25	86	9.32e-11	0.019	112	8.42e-11	0.028
0.5	37	5.76e-11	0.017	47	9.00e-11	0.032
0.75	19	6.11e-11	0.016	24	9.70e-11	0.024
1	6	2.16e-11	0.016	7	2.24e-11	0.028
1.25	19	3.49e-11	0.017	24	4.83e-11	0.021
1.5	36	7.49e-11	0.018	47	5.26e-11	0.021
1.75	85	8.79e-11	0.021	110	9.87e-11	0.024

Example 3.2. In this example, we consider $\mathcal{B} \in \mathbb{R}^{[3,n]}$ as a nonnegative tensor with

$$b_{ijk} = \mid sin(i+j+k) \mid .$$

and solve the 3rd-order $\mathcal{M}$-tensor system $\mathcal{A}x^2 = b$ where $\mathcal{A} = s\mathcal{I} - \mathcal{B}$. We can set $s = n^2$, since

$$\rho(\mathcal{B}) \leq \max_{1\leq i\leq n} \sum_{j,k=1}^{n} b_{ijk} \leq \max_{1\leq i\leq n} \sum_{j,k=1}^{n} 1 = n^2.$$

Hence $\mathcal{A}$ is a strong $\mathcal{M}$-tensor.

We apply preconditioned and un-preconditioned SOR methods to solve (1) with different amounts of n. The numerical results are reported in Table 2 which confirm that the preconditioned SOR method performs better in both CPU times and iterative steps than the SOR method.

Table 2: *Numerical results of Example 3.2 with* $\omega_{opt} = 1.5$.

	PSOR			*SOR*		
n	Iter	Error	time	Iter	Error	time
100	32	7.87e-11	0.148	44	6.09e-11	0.202
200	33	5.27e-11	1.125	46	7.41e-11	1.270
300	33	6.70e-11	4.423	47	9.61e-11	5.238
400	33	7.87e-11	29.262	48	9.15e-11	62.974

Figure 1 illustrates the relationship between the number of iterations and the norm $\mathcal{A}x^{m-1} - b$ and confirms the efficiency of the preconditioner in reducing the error.

4 Conclusion

In this paper, we proposed a diagonal preconditioner by minimizing the Frobenius norm of $\mathcal{I}_m - P\mathcal{A}$ to solve multi-linear systems. We apply the SOR method and PSOR method to solve

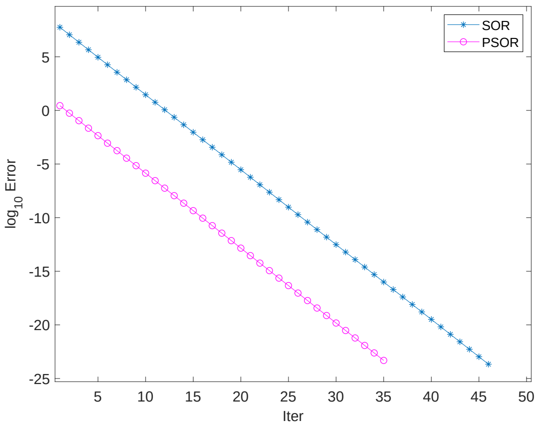

Figure 1: Performance of the SOR and PSOR in reducing residual norm for Example 3.2.

some numerical examples. Analyzing the comparison results shows that the preconditioner improves the method especially in reducing the number of iterations and CPU time.

References

[1] M. Che, L. Qi and Y. Wei, Positive-definite tensors to nonlinear complementarity problems, *J Optim Theory Appl.*, 168 (2016), 475–487.

[2] L. Cui, C. Chen, W. Li and M. Ng, An eigenvalue problem for even order tensors with its applications, *Linear Multilinear Algebra*, 64 (2016), 602–621.

[3] L. Cui, W. Li and M. Ng, Primitive Tensors and Directed Hypergraphs, *Linear Algebra Appl.*, 471 (2015), 96–108.

[4] L. Cui, M. Li and Y. Song, Preconditioned tensor splitting iterations method for solving multi-linear systems, *Appl. Math. Lett.*, 96 (2019) 89–94.

[5] J. Hofbauer, K. Sigmund, Evolutionary game dynamics, *Bull Am Math Soc.*, 40 (2003), 479–519.

[6] W. Li, D. Liu, S.W. Vong, Comparison results for splitting iterations for solving multi-linear systems, *Appl. Numer. Math.*, 134 (2018), 105–121.