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

Generalized conjugate gradient method for solving multilinear systems

Eisa Khosravi Dehdezi* and Saeed Karimi

Department of Mathematics, Faculty of Intelligent Systems Engineering and Data Science, Persian Gulf University, Bushehr, Iran

Abstract

Let $\mathcal{L}$ be a real linear operator with a positive definite symmetric part $\mathcal{M}$. In certain applications, several problems of the form $\mathcal{M} \star_N \mathcal{Y} = \mathcal{G}$ can be solved with less human or computational effort than the original equation $\mathcal{L} \star_N \mathcal{U} = \mathcal{F}$. In this paper, the generalized conjugate gradient method of Concus and Golub [Lecture Notes in Economics and Mathematical Systems 134, Springer-Verlag, New York, 1976] and Widlund [SIAM J. Numer. Anal., 15 (1978), pp. 801-812] is extended for solving some tensor equations via Einstein product. An example is also provided to show the efficiency of the proposed method. Finally, some concluding remarks are given.

Keywords: Generalized conjugate gradient method, Tensor, Multilinear systems

Mathematics Subject Classification [2010]: 15A10, 15A69, 15A72

1 Introduction

The generalized conjugate gradient method of Concus and Golub [2] and Widlund [3] is an iterative method for solving a system of linear equations $\mathbf{Ax} = \mathbf{b}$ when the coefficient matrix $\mathbf{A}$ is real and has positive definite symmetric part $\mathbf{M} = (\mathbf{A} + \mathbf{A}^{\mathsf{T}})/2$. This method can be summarized as follows:

Algorithm 1. Generalized Conjugate Gradient (GCG)

1. Let $\mathbf{x}_0$ be given and set $\mathbf{x}_{-1} = 0$
2. For $j = 0, 1, ...$ until convergence solve $\mathbf{Mv}_j = \mathbf{b} - \mathbf{Ax}_j$ and compute $\rho_j =< \mathbf{Mv}_j, \mathbf{v}_j >$
3. If $j = 0$ set $\omega_{j+1} = 1$ else compute $\omega_{j+1} = [1 + \frac{\rho_j}{\rho_{j-1}} \frac{1}{\omega_j}]^{-1}$
4. Compute $\mathbf{x}_{j+1} = \mathbf{x}_{j-1} + \omega_{j+1}(\mathbf{v}_j + \mathbf{x}_j - \mathbf{x}_{j-1})$,

where $< \mathbf{y}, \mathbf{x} >$ denotes the Euclidean inner product.

*Speaker. Email address: esakhosravidehdezi@gmail.com

Herein, from [4] some definitions and notations are collected. Tensors are written in calligraphic font, e.g., $\mathcal{A}$. Let N be a positive integer, an order N tensor $\mathcal{A} = (a_{i_1...i_N}) = (\mathcal{A})_{i_1...i_N}(1 \leq i_j \leq I_j,\ j = 1,2,...,N)$ is a multidimensional array with $I(I = I_1...I_N)$ entries. Each entry of $\mathcal{A}$ is denoted by $a_{i_1 i_2 ... i_N}$. $\mathcal{O}$ with all entries zero denotes the zero tensor. With this definition of tensors, matrices are tensors of order two where signified by bolded capital letters, e.g., $\mathbf{A}$. As usual, $\mathbb{R}$ and $\mathbb{C}$ denotes the real and complex number field, respectively. Let $\mathbb{R}^{I_1\times...\times I_N}$ and $\mathbb{C}^{I_1\times...\times I_N}$ be the set of order N, dimension $I_1 \times I_2 \times ... \times I_N$ tensors over $\mathbb{R}$ and $\mathbb{C}$, respectively. Let N, M, L be the positive integers, $\mathcal{A} \in \mathbb{C}^{I_1\times...\times I_N\times K_1\times...\times K_M}$ and $\mathcal{B} \in \mathbb{C}^{K_1\times...\times K_M\times J_1\times...\times J_L}$. The Einstein product of $\mathcal{A}$ and $\mathcal{B}$ is defined by the operation $\star_M$ via

$$(\mathcal{A} \star_M \mathcal{B})_{i_1...i_N j_1...j_L} = \sum_{k_M=1}^{K_M} ... \sum_{k_1=1}^{K_1} a_{i_1...i_N k_1...k_M} b_{k_1...k_M j_1...j_L}.$$

Let $\mathcal{A} = (a_{i_1...i_N j_1...j_N}) \in \mathbb{C}^{I_1\times...\times I_N\times I_1\times...\times I_N}$, then $\mathcal{A}^{i+1} = \mathcal{A} \star_N \mathcal{A}^i,\ i = 1,2,....$. Let $\mathcal{B} = (b_{i_1...i_M j_1...j_N}) \in \mathbb{C}^{J_1\times...\times J_M\times I_1\times...\times I_N}$ be the conjugate transpose of $\mathcal{A}$, where $b_{i_1...i_M j_1...j_N} = \bar{a}_{j_1...j_M i_1...i_N}$. The tensor $\mathcal{B}$ is denoted by $\mathcal{A}^*$. When $b_{i_1...i_M j_1...j_N} = a_{j_1...j_M i_1...i_N}$, $\mathcal{B}$ is called the transpose of $\mathcal{A}$, denoted by $\mathcal{A}^\mathsf{T}$. Trace of $\mathcal{A}$ is defined by $tr(\mathcal{A}) = \sum_{i_N=1}^{I_N} ... \sum_{i_1=1}^{I_1} a_{i_1...i_N i_1...i_N}$. Inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{C}^{I_1\times...\times I_N\times J_1\times...\times J_M}$ is defined by

$$< \mathcal{X}, \mathcal{Y} >= tr(\mathcal{Y}^* \star_N \mathcal{X}) = \sum_{j_M=1}^{J_M} ... \sum_{j_1=1}^{J_1} \sum_{i_N=1}^{I_N} ... \sum_{i_1=1}^{I_1} x_{i_1...i_N j_1...j_M} \bar{y}_{j_1...j_M i_1...i_N},$$

so the tensor norm that generated by this inner product is

$$||\mathcal{X}|| = \sqrt{< \mathcal{X}, \mathcal{X} >} = \sqrt{\sum_{j_M=1}^{J_M} ... \sum_{j_1=1}^{J_1} \sum_{i_N=1}^{I_N} ... \sum_{i_1=1}^{I_1} |x_{i_1...i_N j_1...j_M}|^2},$$

which is the tensor Frobenius norm. $\mathcal{A} \in \mathbb{R}^{I_1\times...\times I_N\times I_1\times...\times I_N}$ is said to be a diagonal tensor if $a_{i_1...i_N j_1...j_N} = 0$ for $i_l \neq j_l$ and $l = 1,...,N$. A diagonal tensor $\mathcal{I} \in \mathbb{R}^{I_1\times...\times I_N\times I_1\times...\times I_N}$ is an identity tensor if $i_{i_1...i_N j_1...j_N} = \prod_{k=1}^{N} \delta_{i_k j_k}$, where $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases}$

2 Main results

In this paper we extend the GCG algorithm, named GCG-BTF, for solving the following tensor equations (or multilinear systems) via Einstein product:

$$\mathcal{A} \star_N \mathcal{X} = \mathcal{B}, \tag{1}$$

where $\mathcal{A} \in \mathbb{R}^{I_1\times...\times I_N\times I_1\times...\times I_N}$ and $\mathcal{X}, \mathcal{B} \in \mathbb{R}^{I_1\times...\times I_N}$.

Algorithm 2. Generalized Conjugate Gradient Based Tensor Form (GCG-BTF)

1. Let $\mathcal{X}_0$ be given and set $\mathcal{X}_{-1} = \mathcal{O}$.
2. For $j = 0, 1, ...$ until convergence solve $\mathcal{M} \star_N \mathcal{V}_j = \mathcal{B} - \mathcal{A} \star_N \mathcal{X}_j$ and compute $\rho_j =< \mathcal{M} \star_N \mathcal{V}_j, \mathcal{V}_j >$
3. If $j = 0$ set $\omega_{j+1} = 1$ else compute $\omega_{j+1} = [1 + \dfrac{\rho_j}{\rho_{j-1}} \dfrac{1}{\omega_j}]^{-1}$

4. Compute $\mathcal{X} = \mathcal{X}_{j-1} + \omega_{j+1}(\mathcal{V}_j + \mathcal{X}_j - \mathcal{X}_{j-1})$.

Let $\mathcal{A} = \mathcal{M} - \mathcal{N}$, whence $-\mathcal{N} = (\mathcal{A} - \mathcal{A}^{\mathsf{T}})/2$ is the skew-symmetric part of $\mathcal{A}$, and let $\mathcal{K} = \mathcal{M}^{-1} \star_N \mathcal{N}$. Then the iterate $\mathcal{X}_j$ can be characterized as the unique element in the affine Krylov-type subspace

$$\mathcal{X}_0 + Span\{\mathcal{V}_0, \mathcal{K} \star_N \mathcal{V}_0, \mathcal{K}^2 \star_N \mathcal{V}_0, ..., \mathcal{K}^{j-1} \star_N \mathcal{V}_0\} \equiv \mathcal{X}_0 + \varphi_j,$$

satisfying the Galerkin condition

$$< \mathcal{Z}, \mathcal{A} \star_N \mathcal{E}_j >= 0 \qquad for\ all \quad \mathcal{Z} \in \varphi_j, \tag{2}$$

where $\mathcal{E}_j = \mathcal{X} - \mathcal{X}_j$. Moreover, it can be shown that

$$\mathcal{X}_j = \mathcal{X} + P_j(\mathcal{K}) \star_N \mathcal{E}_0, \tag{3}$$

where $P_j(\mu)$ is an even (odd) polynomial of degree at most j for j even (odd) and $P_j(1) = 1$.

Notation. $< \mathcal{Y}, \mathcal{Z} >_{\mathcal{M}}$ denotes the $\mathcal{M}$-inner product $< \mathcal{M} \star_N \mathcal{Y}, \mathcal{Z} >$ and $||\mathcal{Z}||_{\mathcal{M}}$ denotes the corresponding norm. Note that

$$\begin{aligned} < \mathcal{K} \star_N \mathcal{Y}, \mathcal{Z} >_{\mathcal{M}} &=< \mathcal{N} \star_N \mathcal{Y}, \mathcal{Z} >= - < \mathcal{Y}, \mathcal{N} \star_N \mathcal{Z} > \\ &= - < \mathcal{M} \star_N \mathcal{Y}, \mathcal{M}^{-1} \star_N \mathcal{N} \star_N \mathcal{Z} >= - < \mathcal{Y}, \mathcal{K} \star_N \mathcal{Z} >_{\mathcal{M}}, \end{aligned}$$

so that $\mathcal{K}$ is skew-symmetric with respect to $< .,. >_{\mathcal{M}}$ and $< \mathcal{K} \star_N \mathcal{Z}, \mathcal{Z} >_{\mathcal{M}}= 0$ for all $\mathcal{Z}$.

We note that the Krylov-type sequence is completely specified by its first element $\mathcal{V}_0$. We have worked exclusively with $\mathcal{V}_0 = \mathcal{M}^{-1} \star_N \mathcal{R}_0$, where $\mathcal{R}_j = \mathcal{B} - \mathcal{A} \star_N \mathcal{X}_j$. This is a very natural choice especially when the norm of the operator $\mathcal{K}$ is small.

Remark. It can be shown that the iterate $\mathcal{X}_j$ generated by the GCG-BTF method is the best approximation to $\mathcal{X}$ with respect to a certain j-dimensional affine subspace, but not with respect to the affine Krylov-type subspace $\mathcal{X}_0 + \varphi_j$ (unless $\mathcal{X}_j = \mathcal{X}$).

Error bounds. It is not difficult to see that, use the best approximation property of the iterates $\mathcal{X}_j$, the error bound for the GCG-BTF method is as follows:

$$||\mathcal{X}_j - \mathcal{X}||_{\mathcal{M}} \leq ||Q_j(\mathcal{K}) \star_N (\mathcal{X}_0 - \mathcal{X})||_{\mathcal{M}},$$

for any real polynomial $Q_j(\mu)$ of degree at most j satisfying $Q_j(1) = 1$ and $Q_j(-1) = (-1)^j$.

3 Numerical results

In this section, we give a numerical example to show the performance of the proposed algorithm. All tests were carried out in double precision with a MATLAB code and initial tensor $\mathcal{X}_0 = \mathcal{O}$, when the computer specifications are Microsoft Windows 10 Intel(R), Core(TM)i7-7500U, CPU 2.70 GHz, with 8 GB of RAM. All used codes came from the MATLAB tensor toolbox developed by Bader and Kolda [1]. We compared the proposed methods with CG-BTF, CGS-BTF and Bi-CGSTAB-BTF algorithms, where the stopping criterion is $\|\mathcal{R}_j\| < 10^{-8}$.

We consider two-dimensional (2D) Poisson problem

$$\begin{aligned} -\nabla^2 v = f, \quad in\ \Omega = [0,1] \times [0,1], \\ v = 0, \quad on\ \partial\Omega, \end{aligned} \tag{4}$$

where f is a known function,

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}, \tag{5}$$

and v is unknown function.

Several problems in physics and mechanics are modeled by (4), where v represents, for example, temperature, electromagnetic potential, or displacement of an elastic membrane fixed at the boundary. We compute an approximation of the unknown function $v(x, y)$ in (4). The mesh points are obtained by discretizing the unit square domain with step sizes, Δx in the x-direction and Δy in the y-direction. If we assume that $\Delta x = \Delta y = h = \dfrac{1}{n+1}$, after the standard central difference approximations, we obtain the difference formula

$$4v_{ij} - v_{i-1j} - v_{i+1j} - v_{ij-1} - v_{ij+1} = h^2 f_{ij}, \qquad i, j = 1, 2, \cdots, n. \tag{6}$$

The higher order tensor representation of the 2D discretized Poisson problem (4) is

$$\mathcal{A}_n \star_2 \mathcal{V} = \mathcal{F}, \tag{7}$$

where $\mathcal{A}_n \in \mathbb{R}^{n\times n\times n\times n}$ and $\mathcal{V}, \mathcal{F} \in \mathbb{R}^{n\times n}$ are discretized on the unit square. The nonzeros entries of the tensor block $(\mathcal{A}_n^{(2,4)})_{k=\alpha,l=\beta}$ are in the following five-point stencil

$$\begin{cases} (\mathcal{A}^{(2,4)}_{n_{\alpha,\beta}})_{\alpha,\beta} = \dfrac{4}{h^2}, \\ (\mathcal{A}^{(2,4)}_{n_{\alpha,\beta}})_{\alpha-1,\beta} = (\mathcal{A}^{(2,4)}_{n_{\alpha,\beta}})_{\alpha,\beta-1} = \dfrac{-1}{h^2}, \\ (\mathcal{A}^{(2,4)}_{n_{\alpha,\beta}})_{\alpha+1,\beta} = (\mathcal{A}^{(2,4)}_{n_{\alpha,\beta}})_{\alpha,\beta+1} = \dfrac{-1}{h^2}, \end{cases} \tag{8}$$

for $\alpha, \beta = 2, ..., n-1$ and $\mathcal{F} = 10 * tenrand(n, n) \in \mathbb{R}^{n\times n}$. The numerical results are depicted in Figure 1 for $n = 30$, where in GCG-BTF method, we choose $\mathcal{X}_1 = tenones(n, n)$ and the obtained approximation of the inversion of $\mathcal{M}^{-1}$ using RAPID algorithm [4] is obtained for solving $\mathcal{M} \star_2 \mathcal{V}_j = \mathcal{B} - \mathcal{A} \star_2 \mathcal{X}_j$.

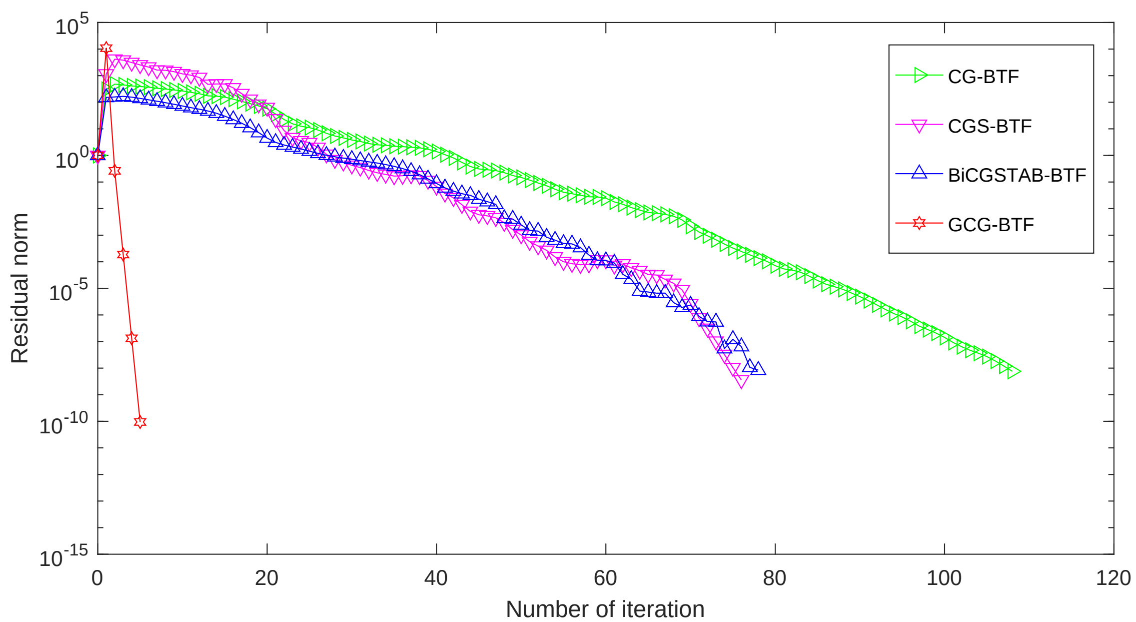

Figure 1: Residual curves.

4 Conclusion

In this paper, the generalized conjugate gradient method is extended for solving tensor equation $\mathcal{A} \star_N \mathcal{X} = \mathcal{B}$. The proposed numerical example provided the efficiency of the GCG-BTF method.

References

[1] B.W. Bader and T.G. Kolda, MATLAB tensor toolbox, version 2.6, Available online at https://www.tensortoolbox.org 2010.

[2] P. Concus and G.H. Golub, A generalized conjugate gradient method]: or nonsymmetric systems of linear equations, *Lecture Notes in Economics and Mathematical Systems, R. Glowinski and J. L. Lions, eds., Springer-Verlag, New York,* 134 (1976), 56–65.

[3] O. Widlund, A Lanczos method for a class of nonsymmetric systems of linear equations, *SIAM J. Numer. Anal.,* 15 (1978), 801–812.

[4] E. Khosravi Dehdezi and S. Karimi, A rapid and powerful iterative method for computing inverses of sparse tensors with applications, *Appl. Math. Comput.* DOI: 10.1016/j.amc.2021.126720