Applied Numerical Mathematics 116 (2017) 204–214

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Numerical issues in computing the antitriangular factorization of symmetric indefinite matrices

Teresa Laudadio [a,1], Nicola Mastronardi [a,*,2], Paul Van Dooren [b,3]

[a] Istituto per le Applicazioni del Calcolo "M. Picone", sede di Bari, Consiglio Nazionale delle Ricerche, Via G. Amendola, 122/D, I-70126 Bari, Italy

[b] Catholic University of Louvain, Department of Mathematical Engineering, Avenue Georges Lemaitre 4, B-1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Article history:
Available online 3 October 2016

Keywords:
Indefinite symmetric matrix
Antitriangular matrices
Inertia

ABSTRACT

An algorithm for computing the antitriangular factorization of symmetric matrices, relying only on orthogonal transformations, was recently proposed. The computed antitriangular form straightforwardly reveals the inertia of the matrix. A block version of the latter algorithm was described in a different paper, where it was noticed that the algorithm sometimes fails to compute the correct inertia of the matrix.

In this paper we analyze a possible cause of the failure of detecting the inertia and propose a procedure to recover it. Furthermore, we propose a different algorithm to compute the antitriangular factorization of a symmetric matrix that handles most of the singularities of the matrix at the very end of the algorithm.

Numerical results are also given showing the reliability of the proposed algorithm.

© 2016 IMACS. Published by Elsevier B.V. All rights reserved.

1. Introduction

Given a symmetric indefinite matrix $A \in \mathbb{R}^{n\times n}$ with inertia (n_-, n_0, n_+), where n_-, n_0 and n_+ are the number of eigenvalues of A less, equal and greater than zero, respectively, and defined $n_1 = \min\{n_-, n_+\}$, $n_2 = \max\{n_-, n_+\} - n_1$, there exists (see [10] for details) an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that

$$A = QMQ^T, \quad M = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & Y^T \\ \mathbf{0} & \mathbf{0} & X & Z^T \\ \mathbf{0} & Y & Z & W \end{bmatrix} \begin{matrix} \}n_0 \\ \}n_1 \\ \}n_2 \\ \}n_1 \end{matrix} \tag{1}$$

with $Z \in \mathbb{R}^{n_1\times n_2}$, $Y \in \mathbb{R}^{n_1\times n_1}$ nonsingular lower antitriangular, $W \in \mathbb{R}^{n_1\times n_1}$ symmetric and $X \in \mathbb{R}^{n_2\times n_2}$ symmetric definite if $n_2 > 0$, i.e., $X = \theta LL^T$ with L nonsingular lower triangular and

* Corresponding author.
E-mail addresses: t.laudadio@ba.iac.cnr.it (T. Laudadio), n.mastronardi@ba.iac.cnr.it (N. Mastronardi), paul.vandooren@uclouvain.be (P. Van Dooren).

[1] The work of this author was partially supported by PRIN 2012 N. 2012MTE38N.

[2] The work of this author was partially supported by the GNCS–INdAM project "Metodi di regolarizzazione per problemi di ottimizzazione e applicazioni" and by Nicola Mastronardi.

[3] The work of this author is partly supported by the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Grant number IAP VII/19 and by the Short-Term Mobility program 2016 of Consiglio Nazionale delle Ricerche, Italy.

http://dx.doi.org/10.1016/j.apnum.2016.09.012
0168-9274/© 2016 IMACS. Published by Elsevier B.V. All rights reserved.

$$\theta = \begin{cases} 1, & \text{if } n_+ > n_- \\ -1, & \text{if } n_+ < n_- \end{cases}.$$

When $n_+ = n_-$, θ and X are not defined and the matrix M reduces to

$$M = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Y^T \\ \mathbf{0} & Y & W \end{bmatrix}.$$

Hence, X is symmetric positive definite if $\theta = 1$ and is symmetric negative definite if $\theta = -1$. The matrix M is said to be a Block AntiTriangular (BAT) matrix or, equivalently, M is said to be in a BAT form. In this paper we denote a zero submatrix by $\mathbf{0}$, whose size is trivial to determine and by $\mathbf{e}_j^{(k)}$ the j-vector of the canonical basis of $\mathbb{R}^k$. If $A \in \mathbb{R}^{n\times n}$, we use the `matlab` notation $A(i_1 : i_2, j_1 : j_2)$ to indicate the submatrix of A made by the rows i and j, for $i \in \{i_1, i_i + 1, \cdots, i_2\}$, and $j \in \{j_1, j_1 + 1, \cdots, j_2\}$. Moreover, the machine precision is denoted by ε.

An algorithm for computing the BAT form of the symmetric matrix (1) is described in [10]. The algorithm is backward stable relying only on stable orthogonal transformations. At the i-th iteration of the algorithm, $i = 2, \ldots, n$, we use the BAT form of $A(1 : i-1, 1 : i-1)$ to compute the BAT form of the submatrix $A(1 : i, 1 : i)$ in a recursive way. In this paper we refer to the latter algorithm as the S-BAT algorithm.

A block extension of the S-BAT algorithm is proposed in [3], where at the i-th iteration the BAT form of $A(1 : i+k-1, 1 : i+k-1)$, $k \geq 1$, is computed, yielding the BAT form of $A(1 : i-1, 1 : i-1)$. In this paper it is noticed that, although only stable orthogonal transformations are performed, the algorithm sometimes fails to detect the exact inertia of the matrix.

In this context, the main aim of the present paper is to analyze a possible cause of the loss of accuracy in the computation of the inertia with the S-BAT algorithm and to develop a procedure to retrieve the exact one. Moreover, we propose a different algorithm for computing the BAT form of a symmetric indefinite matrix aimed to overcome this issue. The idea behind the algorithm is to consider a Lanczos-like procedure [7] in order to handle tiny eigenvalues at the end of the algorithm. The new algorithm inherits the nice properties of the Lanczos algorithm, such as the convergence behavior depending on the minimal polynomial of the matrix. We will refer to this algorithm as the H-BAT algorithm.

The BAT form plays an important role in a variety of applications, where it is important to update (downdate) the factorization of a symmetric indefinite matrix modified by a symmetric rank-one matrix in a fast and stable way. For instance, this problem occurs in tracking the dominant eigenspace of a symmetric indefinite matrix [9]. Moreover, a fast procedure for modifying the factorization of an indefinite Hessian is required in optimization problems based on quasi-Newton methods [6,7,11].

In [10] it is shown that the BAT factorization of a symmetric indefinite matrix $A \in \mathbb{R}^{n\times n}$ can be efficiently updated in a stable way when modified by a symmetric rank-one matrix with $O(n^2)$ floating point operations and, hence, such factorization is a good candidate to solve the aforementioned problems. Updating rank-one modifications of factorizations, like the LDL factorization [4] of a symmetric indefinite matrix, in a stable way is not so straightforward due to the block diagonal matrix D requiring, in the worst case, $O(n^3)$ floating point operations.

Finally, symmetric indefinite matrices arise in a block form in numerous saddle point problems [2], so that the BAT factorization can be applied [12].

The paper is organized as follows.

The analysis of the cause of failure in detecting the inertia of the algorithm proposed in [10] and the procedure to retrieve the exact one are described in Section 2. In Section 3 a new algorithm for computing the anti-triangular factorization of symmetric matrices is presented. The numerical examples are described in Section 4 followed by the conclusions.

2. Numerical issues

In this section we analyze a cause of failure of the S-BAT algorithm [10] in detecting the numerical inertia when computing the antitriangular factorization of a symmetric matrix.

Let us define the τ-inertia of a symmetric matrix $A \in \mathbb{R}^{n\times n}$, denoted by Inertia$(A, \tau)$, as the triple $(\hat{n}_-, \hat{n}_0, \hat{n}_+)$, where $\hat{n}_-, \hat{n}_0$ and $\hat{n}_+$ are the number of computed eigenvalues smaller than $-\tau$, smaller or equal to τ in absolute value, and greater than τ, respectively, with $\tau > 0$ a fixed tolerance.

We shortly describe the i-th iteration of the S-BAT algorithm, $i = 2, \ldots, n$.

At the i-th iteration, the computed BAT form of the principal submatrix $A(1 : i-1, 1 : i-1)$ is used to reduce the principal submatrix $A(1 : i, 1 : i)$ to BAT form. At the end of the i-th iteration, we have the following *"partial"* BAT factorization

$$A = Q^{(i)} \begin{bmatrix} M_1^{(i)} & C^{(i)T} \\ C^{(i)} & T^{(i)} \end{bmatrix} Q^{(i)T}$$

where $M_1^{(i)} \in \mathbb{R}^{i\times i}$ is in a BAT form with inertia (i_+, i_0, i_-), $C^{(i)} \in \mathbb{R}^{(n-i)\times i}$, $T^{(i)} \in \mathbb{R}^{(n-i)\times(n-i)}$ symmetric and $Q^{(i)} \in \mathbb{R}^{n\times n}$ orthogonal.

For the sake of simplicity, we assume that $i_0 = 0$. The case $i_0 > 0$ can be handled in a similar way. The submatrix $M_1^{(i)}$ looks like

Download English Version:

https://daneshyari.com/en/article/5776659

Download Persian Version:

https://daneshyari.com/article/5776659

Daneshyari.com