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

A R T I C L E I N F O

Article history:
Available online 3 October 2016

Keywords:
Indefinite symmetric matrix
Antitriangular matrices
Inertia

A B S T R A C T

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.

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

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

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/

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

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, $i = 2, \dots, n$, 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, $i = 2, \dots, n$.

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

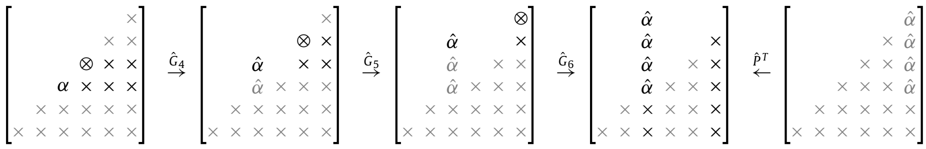

Fig. 1. Displacement and modification of the entry α from position $(j, i_1 - j + 1)$ to $(1, i_1)$ by means of a sequence of Givens rotations applied to the left $(\rightarrow)$ followed by a multiplication by a permutation matrix $\hat{P}^T$ to the right $(\leftarrow)$. In each matrix, the nonzero entries are denoted with the symbol $\times$ in gray, the entry to be annihilated by the multiplication of the Givens rotation is denoted with $\otimes$, the entries smaller than α in absolute value are denoted by $\hat{\alpha}$, and the rows to be modified are depicted in black, while the columns swapped by the application of the permutation matrix $\hat{P}^T$ to the right are depicted in black.

$$M_1^{(i)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & Y^T \\ \mathbf{0} & X & Z^T \\ Y & Z & W \end{bmatrix} \begin{matrix} \}i_1 \\ \}i_2 \\ \}i_1 \end{matrix},$$

where $2i_1 + i_2 = i$.

In the S-BAT algorithm, for a fixed threshold $\tau > 0$, $M_1^{(i)}$ is considered "τ-numerically" singular if one of the entries in the main antidiagonal of Y is, in absolute value, less than τ, i.e., if

$$|Y(j, i_1 - j + 1)| < \tau, \quad \text{for a } j \in \{1, \ldots, i_1\}. \tag{2}$$

In the sequel we will show that the matrix $M_1^{(i)}$ can be τ-numerically singular even if condition (2) is not satisfied and we propose a procedure to recover the τ-inertia.

To this end, let us suppose that (2) holds for a $j \in \{1, \ldots, i_1\}$ and let $\alpha = Y(j, i_1 - j + 1)$.

We proceed in the following way.

Let $\hat{G}_k \in \mathbb{R}^{i_1 \times i_1}$, $k = j+1, j+2, \ldots, i_1$, be the sequence of Givens rotations such that, for $\hat{Q}_1 = \hat{G}_{i_1} \cdots \hat{G}_{j+2}\hat{G}_{j+1}$,

$$\hat{Q}_1 Y \hat{P}^T = \begin{bmatrix} & \hat{\alpha} \\ \hat{Y} & \hat{\mathbf{a}}_1 \end{bmatrix} \begin{matrix} \}1 \\ \}i_1 \end{matrix} \tag{3}$$

with $\hat{\alpha} \in \mathbb{R}$, $|\hat{\alpha}| \le \alpha$, $\hat{\mathbf{a}}_1 \in \mathbb{R}^{i_1 - 1}$, $\hat{Y} \in \mathbb{R}^{(i_1-1)\times(i_1-1)}$ nonsingular antitriangular and $P^T \in \mathbb{R}^{i_1 \times i_1}$ is a permutation matrix swapping the columns $i_1 - j + 1$ and i_1.

The transformation (3) is depicted in a graphical way in Fig. 1 for a matrix Y of order 6.

Let

$$Q_1 = \begin{bmatrix} \hat{P} & & \\ & I_{i_2} & \\ & & \hat{Q}_1 \end{bmatrix}.$$

Then

$$M_2^{(i)} = Q_1 M_1^{(i)} Q_1^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \hat{Y}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \hat{\alpha} & \hat{\mathbf{a}}_1^T \\ \mathbf{0} & \mathbf{0} & X & \hat{\mathbf{a}}_2 & \hat{Z}^T \\ \mathbf{0} & \hat{\alpha} & \hat{\mathbf{a}}_2^T & \hat{\gamma} & \hat{\mathbf{a}}_3^T \\ \hat{Y} & \hat{\mathbf{a}}_1 & \hat{Z} & \hat{\mathbf{a}}_3 & \hat{W} \end{bmatrix} \begin{matrix} \}i_1 - 1 \\ \}1 \\ \}i_2 \\ \}1 \\ \}i_1 - 1 \end{matrix}, \tag{4}$$

with

$$\begin{bmatrix} \hat{\alpha} \\ \hat{\mathbf{a}}_1 \end{bmatrix} = \hat{Q}_1 \begin{bmatrix} \mathbf{0} \\ \alpha \\ Y_{j+1:i_1, i_1 - j + 1} \end{bmatrix}, \quad \begin{bmatrix} \hat{\mathbf{a}}_2^T \\ \hat{Z} \end{bmatrix} = \hat{Q}_1 Z, \quad \begin{bmatrix} \hat{\gamma} & \hat{\mathbf{a}}_3^T \\ \hat{\mathbf{a}}_3 & \hat{W} \end{bmatrix} = \hat{Q}_1 W \hat{Q}_1^T.$$

Therefore $|\hat{\alpha}| \le |\alpha|$. Let us consider the submatrices

$$K = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \hat{\alpha} \\ \mathbf{0} & X & \hat{\mathbf{a}}_2 \\ \hat{\alpha} & \hat{\mathbf{a}}_2^T & \hat{\gamma} \end{bmatrix}, \quad K_1 = \begin{bmatrix} X & \hat{\mathbf{a}}_2 \\ \hat{\mathbf{a}}_2^T & \hat{\gamma} \end{bmatrix}.$$

We observe that, in exact arithmetic, if $\hat{\alpha} > 0$, then $\text{Inertia}(K) = (1, 0, i_2 + 1)$ if $\theta > 0$, $\text{Inertia}(K) = (i_2 + 1, 0, 1)$ if $\theta < 0$. Nevertheless, we will see that, in floating point arithmetic, K can be numerically singular, therefore implying that $\text{Inertia}(K) = (1, 1, i_2)$ if K_1 is indefinite or $\text{Inertia}(K) = (0, 1, i_2 + 1)$ if K_1 is definite.

Let $\hat{P}_2 \in \mathbb{R}^{(i_2+2)\times(i_2+2)}$ be a permutation matrix such that

$$K_2 = \hat{P}_2 K \hat{P}_2^T = \begin{bmatrix} X & \hat{\mathbf{a}}_2 & \mathbf{0} \\ \hat{\mathbf{a}}_2^T & \hat{\gamma} & \hat{\alpha} \\ \mathbf{0} & \hat{\alpha} & \mathbf{0} \end{bmatrix},$$

and let

$$P_2 = \begin{bmatrix} I_{i_1-1} & & \\ & \hat{P}_2 & \\ & & I_{i_1-1} \end{bmatrix}.$$

Hence K_2 can be decomposed as

$$K_2 = \theta \hat{L}\hat{L}^T + K_3,$$

where

$$\hat{L} = \begin{bmatrix} L \\ \mathbf{w}^T \\ \mathbf{0} \end{bmatrix} \text{ and } K_3 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\gamma} & \hat{\alpha} \\ \mathbf{0} & \hat{\alpha} & \mathbf{0} \end{bmatrix},$$

and $\mathbf{w} = \theta L^{-1}\hat{\mathbf{a}}_2$, $\tilde{\gamma} = \hat{\gamma} - \theta \mathbf{w}^T\mathbf{w}$.

Hence K, and thus $M_1^{(i)}$, are either numerically definite, indefinite or singular if this is also the case for the submatrix

$$\hat{K}_3 = \begin{bmatrix} \tilde{\gamma} & \hat{\alpha} \\ \hat{\alpha} & 0 \end{bmatrix}.$$

The eigenvalues of $\hat{K}_3$ are respectively

$$\lambda_{1,2}(\hat{K}_3) = \frac{\tilde{\gamma}}{2} \pm \sqrt{\frac{\tilde{\gamma}^2}{4} + \hat{\alpha}^2}.$$

Therefore, if

$$\left|\frac{\hat{\alpha}}{\tilde{\gamma}}\right| < \frac{\sqrt{\varepsilon}}{2} \tag{5}$$

then $\hat{K}_3$ is numerically singular with numerical eigenvalues $\lambda_1(\hat{K}_3) = 0$ and $\lambda_2(\hat{K}_3) = \tilde{\gamma}$.

In this case, we can consider the Givens rotation $\tilde{G}_2$ such that

$$\tilde{G}_2 \hat{K}_3 \tilde{G}_2^T = \begin{bmatrix} 0 & \\ & \lambda_2(\hat{K}_3) \end{bmatrix}.$$

Let

$$\hat{Q}_2 = \begin{bmatrix} I_{i_2} & \\ & \tilde{G}_2 \end{bmatrix} \text{ and } Q_2 = \begin{bmatrix} I_{i_1-1} & & \\ & \hat{Q}_2 & \\ & & I_{i_1-1} \end{bmatrix}.$$

Then

$$\hat{Q}_2 K_2 \hat{Q}_2^T = \theta \begin{bmatrix} L \\ \mathbf{w}_1^T \\ \mathbf{w}_2^T \end{bmatrix} \begin{bmatrix} L^T & \mathbf{w}_1 & \mathbf{w}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_2(\hat{K}_3) \end{bmatrix},$$

with

$$\begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \end{bmatrix} = \tilde{G}_2 \begin{bmatrix} \mathbf{w}^T \\ \mathbf{0} \end{bmatrix}.$$

Let $\hat{Q}_3 \in \mathbb{R}^{(i_2+2)\times(i_2+2)}$ be the product of the Givens rotations $\hat{Q}_3 = G_{i_2-1}G_{i_2-2}\cdots G_2G_1$ such that

$$\hat{Q}_3 \begin{bmatrix} L \\ \mathbf{w}_1^T \\ \mathbf{w}_2^T \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \hat{L} \\ \hat{\mathbf{w}}_2^T \end{bmatrix} \begin{matrix} \}1 \\ \}i_2 \\ \}1 \end{matrix}, \tag{6}$$

with $\hat{L}$ nonsingular lower triangular and $\hat{\mathbf{w}}_2 \in \mathbb{R}^{i_2}$. The transformation (6) is depicted in a graphical way in Fig. 2 for $i_2 = 4$. Then

$$\begin{bmatrix} \times &&& \\ \times & \times && \\ \times & \times & \times & \\ \times & \times & \times & \otimes \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{G_4} \begin{bmatrix} \times &&& \\ \times & \times && \\ \times & \times & \otimes & \\ \times & \times & \times & \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{G_3} \begin{bmatrix} \times &&& \\ \times & \otimes && \\ \times & \times && \\ \times & \times & \times & \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{G_2} \begin{bmatrix} \otimes &&& \\ \times &&& \\ \times & \times && \\ \times & \times & \times & \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{G_1} \begin{bmatrix} &&& \\ \times &&& \\ \times & \times && \\ \times & \times & \times & \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}$$

Fig. 2. Annihilation of the main diagonal of the lower trapezoidal matrix by a sequence of Givens rotations described in (6). The same notation of Fig. 1 is adopted.

$$\hat{Q}_3\hat{Q}_2\hat{P}_2K\hat{P}_2^T\hat{Q}_2^T\hat{Q}_3^T = \theta \begin{bmatrix} \mathbf{0} \\ \hat{L} \\ \hat{\mathbf{w}}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \hat{L}^T & \hat{\mathbf{w}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda_2(\hat{K}_3) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{X} & \tilde{\mathbf{w}}_2 \\ \mathbf{0} & \tilde{\mathbf{w}}_2^T & \tilde{\gamma} \end{bmatrix}$$

where

$$\hat{X} = \theta\hat{L}\hat{L}^T, \quad \tilde{\mathbf{w}}_2 = \theta\hat{L}\hat{\mathbf{w}}_2, \quad \tilde{\gamma} = \theta\hat{\mathbf{w}}_2^T\hat{\mathbf{w}}_2 + \lambda_2(\hat{K}_2). \tag{7}$$

Hence

$$M_3^{(i)} = Q_3Q_2P_2Q_1M_1^{(i)}Q_1^TP_2^TQ_2^TQ_3^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \hat{Y}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{\mathbf{a}}_4^T \\ \mathbf{0} & \mathbf{0} & \hat{X} & \tilde{\mathbf{w}}_2 & \tilde{Z}^T \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{w}}_2^T & \tilde{\gamma} & \tilde{\mathbf{a}}_3^T \\ \hat{Y} & \tilde{\mathbf{a}}_4 & \tilde{Z} & \tilde{\mathbf{a}}_3 & \hat{W} \end{bmatrix},$$

where

$$Q_3 = \begin{bmatrix} I_{i_1+1} && \\ & \hat{Q}_3 & \\ && I_{i_1} \end{bmatrix} \text{ and } \begin{bmatrix} \tilde{\mathbf{a}}_4^T \\ \tilde{Z}^T \\ \tilde{\mathbf{a}}_3^T \end{bmatrix} = \hat{Q}_3\hat{Q}_2\hat{P}_2 \begin{bmatrix} \hat{\mathbf{a}}_1^T \\ \hat{Z}^T \\ \hat{\mathbf{a}}_3^T \end{bmatrix}.$$

Let $\hat{Q}_4 \in \mathbb{R}^{i_1 \times i_1}$ be the orthogonal matrix such that

$$\hat{Q}_4 \begin{bmatrix} \hat{Y}^T \\ \tilde{\mathbf{a}}_4^T \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \tilde{Y}^T \end{bmatrix} \begin{matrix} \}1 \\ \}i_1 - 1 \end{matrix}$$

and let

$$Q_4 = \begin{bmatrix} \hat{Q}_4 & \\ & I_{i_1+i_2} \end{bmatrix}.$$

Then

$$M_4^{(i)} = Q_4M_3^{(i)}Q_4^T = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \tilde{Y}^T \\ \mathbf{0} & \mathbf{0} & \hat{X} & \tilde{\mathbf{w}}_2 & \tilde{Z}^T \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{w}}_2^T & \tilde{\gamma} & \tilde{\mathbf{a}}_3^T \\ \mathbf{0} & \tilde{Y} & \tilde{Z} & \tilde{\mathbf{a}}_3 & \hat{W} \end{bmatrix} \begin{matrix} \}1 \\ \}i_1 - 1 \\ \}i_2 \\ \}1 \\ \}i_1 - 1 \end{matrix}.$$

The last step is to check whether the submatrix

$$\begin{bmatrix} \hat{X} & \tilde{\mathbf{w}}_2 \\ \tilde{\mathbf{w}}_2^T & \tilde{\gamma} \end{bmatrix}$$

is either definite or indefinite and to transform it, by the multiplication of an orthogonal matrix $\hat{Q}_5 \in \mathbb{R}^{(i_2+1)\times(i_2+1)}$, respectively, to

$$\text{either } \theta\tilde{L}\tilde{L}^T \text{ or } \begin{bmatrix} && \beta \\ & \tilde{X} & \tilde{\mathbf{w}}_2 \\ \beta & \tilde{\mathbf{w}}_2^T & \tilde{\gamma} \end{bmatrix} \begin{matrix} \}1 \\ \}i_2 - 1 \\ \}1 \end{matrix}.$$

The details on the last step can be found in [10].

Then

$$M_5^{(i)} = Q_5 M_4^{(i)} Q_5^T$$

is in a BAT form with the corrected numerical inertia, where

$$Q_5 = \begin{bmatrix} I_{i_1} & & \\ & \hat{Q}_5 & \\ & & I_{i_1-1} \end{bmatrix}.$$

Remark 2.1. In many numerical experiments we observed that the bound of (5) should also depend on the size i of the already computed antitriangular submatrix $M_1^{(i)}$ at the i-th iteration. Therefore, to check whether $M_1^{(i)}$ is τ-numerically singular we replace (5) by

$$\left|\frac{\hat{\alpha}}{\tilde{\gamma}}\right| < \frac{i\sqrt{\varepsilon}}{2}.$$

3. The new algorithm

In this section we describe a different way to compute the anti-triangular factorization of a symmetric indefinite matrix $A \in \mathbb{R}^{n\times n}$.

We describe the first and the generic i-th iteration, $i = 1, \ldots, n-1$. Let $A_1 = A$.

At the first step, a Householder matrix $H_1 \in \mathbb{R}^{n\times n}$ is applied to A such that $A_1 = H_1^T \hat{A}_1 H_1$, with

$$\hat{A}_1 = \left[\begin{array}{c|c} T_1 & \begin{matrix}\mathbf{0}\\ \mathbf{v}_2^T\end{matrix} \\ \hline \mathbf{0}\ \mathbf{v}_2 & \tilde{A}_2 \end{array}\right],$$

with $T_1 \in \mathbb{R}^{2\times 2}$, $\mathbf{v}_2 \in \mathbb{R}^{n-2}$, and $\tilde{A}_2 \in \mathbb{R}^{(n-2)\times(n-2)}$. Let $\hat{G}_1 \in \mathbb{R}^{2\times 2}$ be the Givens rotation such that $M_2 = \hat{G}_1 T_1 \hat{G}_1^T$ is symmetric antitriangular (see [10] for details) and let

$$G_1 = \begin{bmatrix} \hat{G}_1 & \\ & I_{n-2} \end{bmatrix}, \quad A_2 = \left[\begin{array}{c|c} M_2 & \mathbf{u}_2\mathbf{v}_2^T \\ \hline \mathbf{v}_2\mathbf{u}_2^T & \tilde{A}_2 \end{array}\right],$$

where $\mathbf{u}_2 \in \mathbb{R}^2$ is the last column of G_1. Then

$$A_1 = Q_1^T A_2 Q_1,$$

with $Q_1 = G_1 H_1$. Hence, after the first step, the submatrix $A_2(1:2, 1:2)$ is in antitriangular form.

At the i-th iteration we proceed in a similar way. Suppose A_i is divided as follows,

$$A_i = \left[\begin{array}{c|c} M_i & \mathbf{u}_i\mathbf{v}_i^T \\ \hline \mathbf{v}_i\mathbf{u}_i^T & \tilde{A}_i \end{array}\right],$$

with $M_i \in \mathbb{R}^{i\times i}$ symmetric antitriangular, $\mathbf{u}_i \in \mathbb{R}^i$, $\mathbf{v}_i \in \mathbb{R}^{(n-i)}$ and $\tilde{A}_i \in \mathbb{R}^{(n-i)\times(n-i)}$ symmetric.

Let $\hat{H}_i \in \mathbb{R}^{(n-i)\times(n-i)}$ be the Householder matrix such that $\hat{H}_i \mathbf{v}_i = \alpha_i \mathbf{e}_i^{(i)}$ and let

$$H_i = \begin{bmatrix} I_i & \\ & \hat{H}_i \end{bmatrix}.$$

Then

$$A_i = H_i^T \hat{A}_i H_i,$$

with

$$\hat{A}_i = \begin{bmatrix} M_i & \hat{\mathbf{u}}_i & \mathbf{0} \\ \hat{\mathbf{u}}_i^T & \gamma_i & \mathbf{v}_{i+1}^T \\ \mathbf{0} & \mathbf{v}_{i+1} & \tilde{A}_{i+1} \end{bmatrix},$$

and $\hat{\mathbf{u}}_i = \alpha_i \mathbf{u}_i$,

$$\begin{bmatrix} \gamma_i & \mathbf{v}_{i+1}^T \\ \mathbf{v}_{i+1} & \tilde{A}_{i+1} \end{bmatrix} = \hat{H}_i^T \tilde{A}_i \hat{H}_i,$$

with $\gamma_i \in \mathbb{R}$, $\mathbf{v}_{i+1} \in \mathbb{R}^{(n-i-1)}$, $\tilde{A}_{i+1} \in \mathbb{R}^{(n-i-1)\times(n-i-1)}$.

Let $\hat{G}_i \in \mathbb{R}^{(i+1)\times(i+1)}$ be the orthogonal matrix [10] such that

$$M_{i+1} = \hat{G}_1 \begin{bmatrix} M_i & \hat{\mathbf{u}}_i \\ \hat{\mathbf{u}}_i^T & \gamma_i \end{bmatrix} \hat{G}_1^T$$

is symmetric antitriangular and let

$$G_i = \begin{bmatrix} \hat{G}_i & \\ & I_{n-i-1} \end{bmatrix}.$$

Then

$$A_{i+1} = G_i \hat{A}_i G_i^T = \left[\begin{array}{c|c} M_{i+1} & \mathbf{u}_{i+1}\mathbf{v}_{i+1}^T \\ \hline \mathbf{v}_{i+1}\mathbf{u}_{i+1}^T & \tilde{A}_{i+1} \end{array}\right],$$

where $\mathbf{u}_{i+1} \in \mathbb{R}^{i+1}$ is the last column of $\hat{G}_i$. Hence

$$A_{i+1} = Q_i A_i Q_i^T,$$

with $Q_i = G_i H_i$.

After the $(n-1)$-st iteration, the matrix A_n is in antitriangular form and the matrix H_{n-1} is the identity matrix.

Remark 3.1. We observe that the antitriangular matrix A_n computed by the proposed algorithm can be obtained computing first the Householder reduction of A to tridiagonal form,

$$T_n = Q_{n-2} Q_{n-1} \cdots Q_1 A_1 Q_{n-2}^T Q_{n-1}^T \cdots Q_1^T, \tag{8}$$

and then applying the sequential algorithm described in [10],

$$A_n = G_{n-1} G_{n-2} \cdots G_1 T_n G_{n-1}^T G_{n-2}^T \cdots G_1^T.$$

Therefore, by (8), the proposed algorithm inherits the properties of the Lanczos algorithm, since the Householder reduction of a symmetric matrix to a tridiagonal one is equivalent to the Lanczos algorithm with starting vector $\mathbf{e}_1^{(n)}$. In particular, if the matrix A has k eigenvalues equal to zero, $0 \le k \le n$, and all the eigenvalues different from zero have multiplicity equal to 1, then the last $k-1$ rows and columns of T_n are zero. Hence, in such a case, applying the described algorithm, all the singularities but 1 of the matrix are handled at the last $k-1$ iterations of the algorithm. Moreover, if the degree of the minimal polynomial of A is $m < n$, the anti-triangular matrix M_m computed after m iterations of the algorithm is not anymore modified by the completion of the algorithm, but only permuted and shifted downward along the main diagonal.

4. Numerical results

In this section we consider three examples. In the first one we emphasize the dependency on $\hat{\alpha}$ and $\tilde{\gamma}$ of (5) in detecting the correct numerical inertia.

In the last two examples we compare the algorithm S-BAT [10] with the algorithm H-BAT described in Section 3, with the LDL^T factorization with partial pivoting [4] denoted by $A = \tilde{P}^T \tilde{L} \tilde{D} \tilde{L}^T \tilde{P}$ and with the LDL^T factorization with rook pivoting [1] denoted $A = \hat{P}^T \hat{L} \hat{D} \hat{L}^T \hat{P}$. The last two factorizations are computed by the `matlab` function `ldlt_symm` available at [8]. The BAT factorizations computed by S-BAT and H-BAT are denoted by $Q_S M_S Q_S^T$ and $Q_H M_H Q_H^T$, respectively. While S-BAT and H-BAT algorithms depend on the tolerance τ fixed a priori, both LDL^T factorizations do not. Having computed the eigenvalues of the block diagonal D of the LDL^T factorization, then Inertia(D,τ) is given by the triple made by the number of eigenvalues of D less than $-\tau$, less or equal to τ in absolute value and greater than τ.

All the experiments were carried out in `matlab` with machine precision $\varepsilon \approx 2.22 \times 10^{-16}$.

Example 1. We consider a matrix $A = QDQ^T$ of order $n = 50$, with $Q \in \mathbb{R}^{n\times n}$ a random orthogonal matrix generated by the `matlab` function $\texttt{gallery}('\texttt{qmult}', n)$, and $D = \texttt{diag}(d)$, with $d = \texttt{randi}([-1,1], n, 1)$, i.e., d is a random vector whose elements are -1, 0, and 1. For this particular example, the number of "-1", "0" and "1" in d are respectively 15, 19 and 16, so, $\texttt{Inertia}(A) = (15, 19, 16)$. It turns out that

$$\frac{\|Q_H M_H Q_H^T - A\|_2}{\|A\|_2} = 1.94 \times 10^{-15}.$$

In Fig. 3 the sparsity structure of the computed antitriangular matrix is displayed (left). This structure was expected [10], since A has only eigenvalues $\{-1, 0, 1\}$. We observe that with the new adapted criterion, the ratios $|\alpha_j/\gamma_j|$ (*) are below the threshold (+) $2j\sqrt{\varepsilon}$ and above the smallest eigenvalue (o) of K at the j-th iteration of the S-BAT algorithm (Fig. 3 (right)).

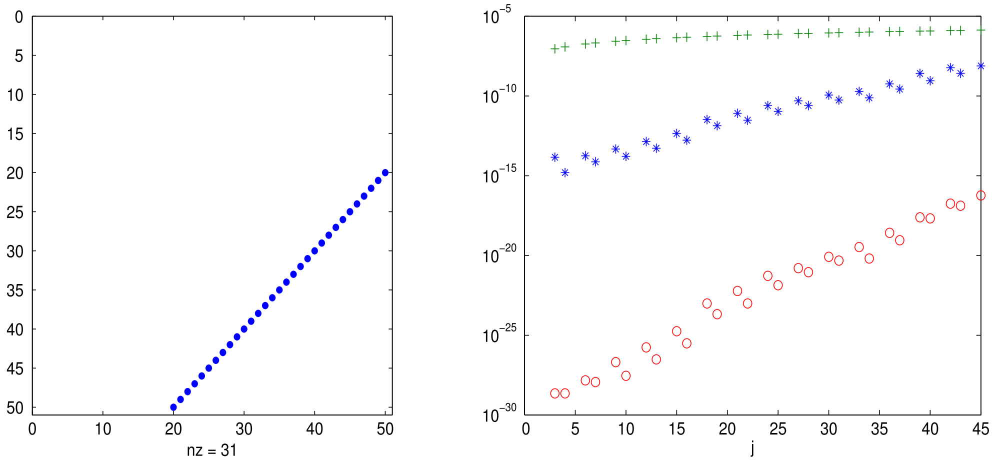

Fig. 3. Left: Computed antitriangular matrix with eigenvalues $\{-1, 0, 1\}$. Right: "o", smallest eigenvalue of K at step j; "*", ratio of the coefficients α_j and γ_j; "+", bounds for the ratios γ_j/α_j.

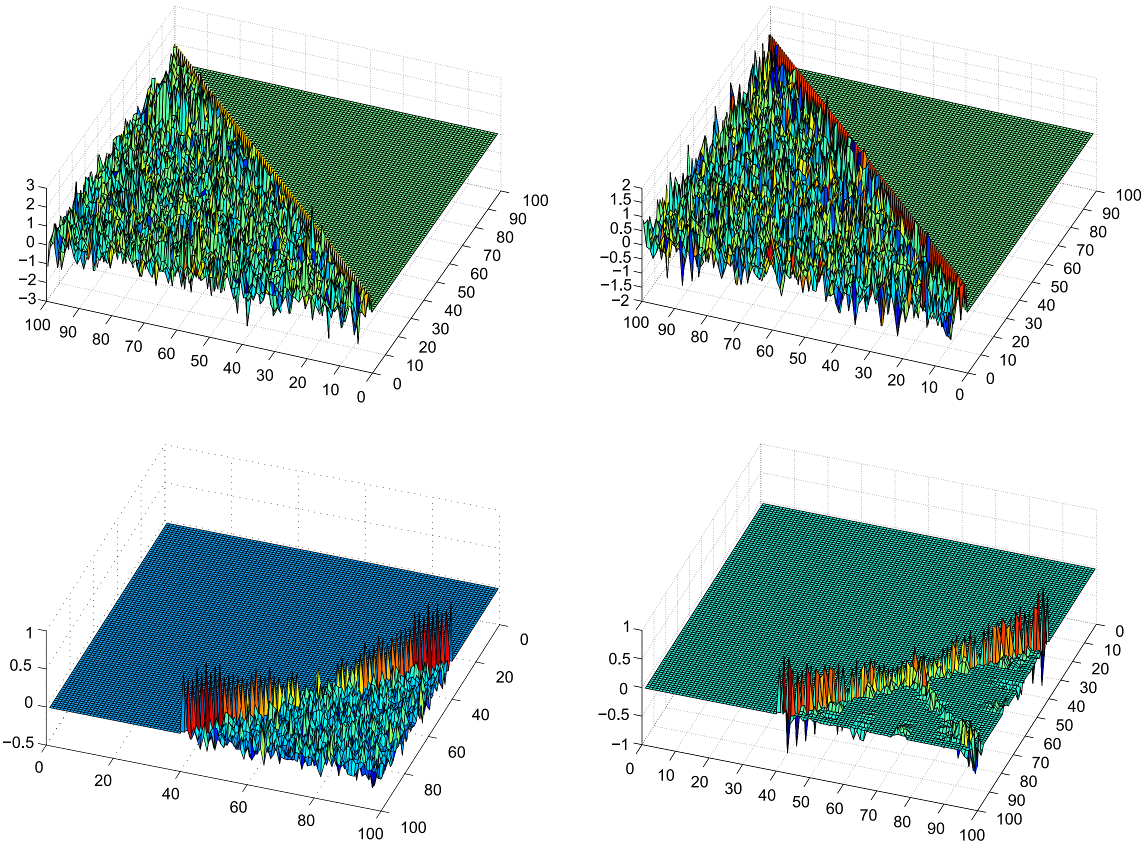

Fig. 4. Entries of the matrices $\tilde{L}$ (top, left), $\hat{L}$ (top, right), M_S (bottom, left), and M_H (bottom, right) in Example 2.

Example 2. Using the vector $d = [\texttt{randn}(60, 1); \texttt{zeros}(40, 1)]$, and a random orthogonal $Q \in \mathbb{R}^{100\times 100}$, the symmetric indefinite matrix $A \in \mathbb{R}^{100\times 100}$ considered in this example is given by $A = Q \times \texttt{diag}(d) \times Q^T$, where Q is computed by the matlab function $\texttt{gallery}('\texttt{qmult}', n)$, with $n = 100$.

The exact Inertia is $(29, 40, 31)$. In Fig. 4 the entries of the $\tilde{L}$ factor of the LDL^T factorization with partial pivoting (first row, left), the entries of the $\hat{L}$ factor of the LDL^T factorization with rook pivoting (first row, right), the entries of the antitriangular matrices M_S and M_H computed respectively by the S-BAT (second row, left) and by the H-BAT algorithm (second row, right) are depicted. We can notice that the $\tilde{L}$, $\hat{L}$ and M_S are full structured matrices, while M_H is sparse.

Table 1
Relative residuals of the factorizations computed by the LDL^T factorization with partial pivoting, by the LDL^T factorization with rook pivoting, by the S-BAT and H-BAT algorithms, respectively, for different values of τ.

τ	$\frac{\|\tilde{P}^T\tilde{L}\tilde{D}\tilde{L}^T\tilde{P}-A\|_2}{\|A\|_2}$	$\frac{\|\hat{P}^T\hat{L}\hat{D}\hat{L}^T\hat{P}-A\|_2}{\|A\|_2}$	$\frac{\|Q_SA_SA_S^T-A\|_2}{\|A\|_2}$	$\frac{\|Q_HA_HA_H^T-A\|_2}{\|A\|_2}$
1.0×10^{-13}	4.99×10^{-16}	3.61×10^{-16}	3.10×10^{-15}	2.39×10^{-15}
1.0×10^{-14}	4.99×10^{-16}	3.61×10^{-16}	3.10×10^{-15}	2.33×10^{-15}
1.0×10^{-15}	4.99×10^{-16}	3.61×10^{-16}	3.14×10^{-15}	2.30×10^{-15}
1.0×10^{-16}	4.99×10^{-16}	3.61×10^{-16}	3.69×10^{-15}	2.56×10^{-15}

Table 2
Inertias of the matrices $\tilde{D}$, $\hat{D}$, M_S and M_H computed by the LDL^T factorization with partial pivoting, by the LDL^T factorization with rook pivoting, by the S-BAT and H-BAT algorithms, respectively, for different values of τ.

τ	Inertia$(\tilde{D},\tau)$	Inertia$(\hat{D},\tau)$	Inertia(M_S,τ)	Inertia(M_H,τ)
1.0×10^{-13}	(29, 40, 31)	(29, 40, 31)	(29, 40, 31)	(29, 40, 31)
1.0×10^{-14}	(29, 40, 31)	(29, 40, 31)	(29, 40, 31)	(29, 40, 31)
1.0×10^{-15}	(34, 31, 35)	(34, 31, 35)	(31, 35, 34)	(30, 39, 31)
1.0×10^{-16}	(52, 1, 47)	(52, 1, 47)	(52, 2, 46)	(49, 10, 41)

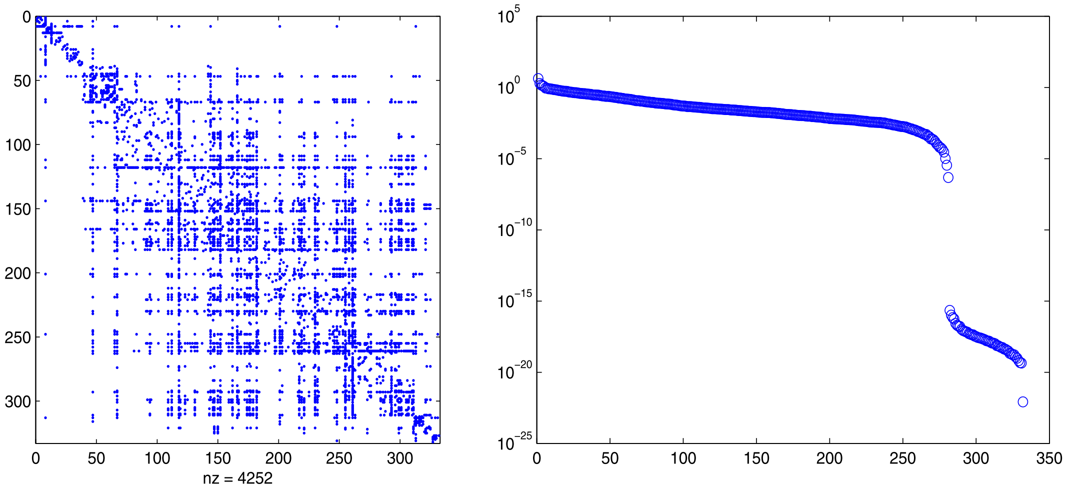

Fig. 5. Sparsity of the matrix of Example 3 (left) and distribution of its eigenvalues (right).

The relative residuals of all considered methods, for different values of τ, are reported in Table 1 (the relative residuals of the LDL^T factorizations are all equal because independent of τ).

The τ-inertias, for all the considered methods and different values of τ, are reported in Table 2. We can observe that all methods fail to compute the exact inertia when the tolerance τ becomes tiny. However, for all values of τ, the inertia computed by H-BAT seems to be the closest to the exact one.

Example 3. In this example we consider the matrix `USAir97` belonging to the group `Pajek` of [5], here denoted by $A \in \mathbb{R}^{332\times332}$. In Fig. 5 the sparsity structure (left) and the singular value distribution (right) of A are displayed. Moreover, Inertia$(A) = [146, 51, 135]$. In Fig. 6 the sparsity structures of $\tilde{L}$, $\hat{L}$, M_S and M_H are depicted. Observe that M_H is sparser than $\tilde{L}$, $\hat{L}$ and M_S.

The relative residuals of all considered methods, for different values of τ, are reported in Table 3 (the relative residuals of the LDL^T factorizations are all equal because independent of τ).

The τ-inertias, for all the considered methods and different values of τ, are reported in Table 4. The inertia computed by H-BAT seems to be the closest to the exact one for all values of τ, also in this example.

5. Conclusions

In this paper a novel criterion to check whether a symmetric antitriangular matrix is singular was introduced. Moreover a different backward stable antitriangular factorization, called as H-BAT algorithm, inheriting the properties of the Lanczos algorithm, is presented. The considered criterion was embedded into the algorithm S-BAT [10] and into H-BAT and their performances in accuracy were compared with those of the LDL factorizations with partial pivoting [4] and with rook pivoting [1]. The numerical examples show that comparable results are obtained in terms of relative residuals. Concerning

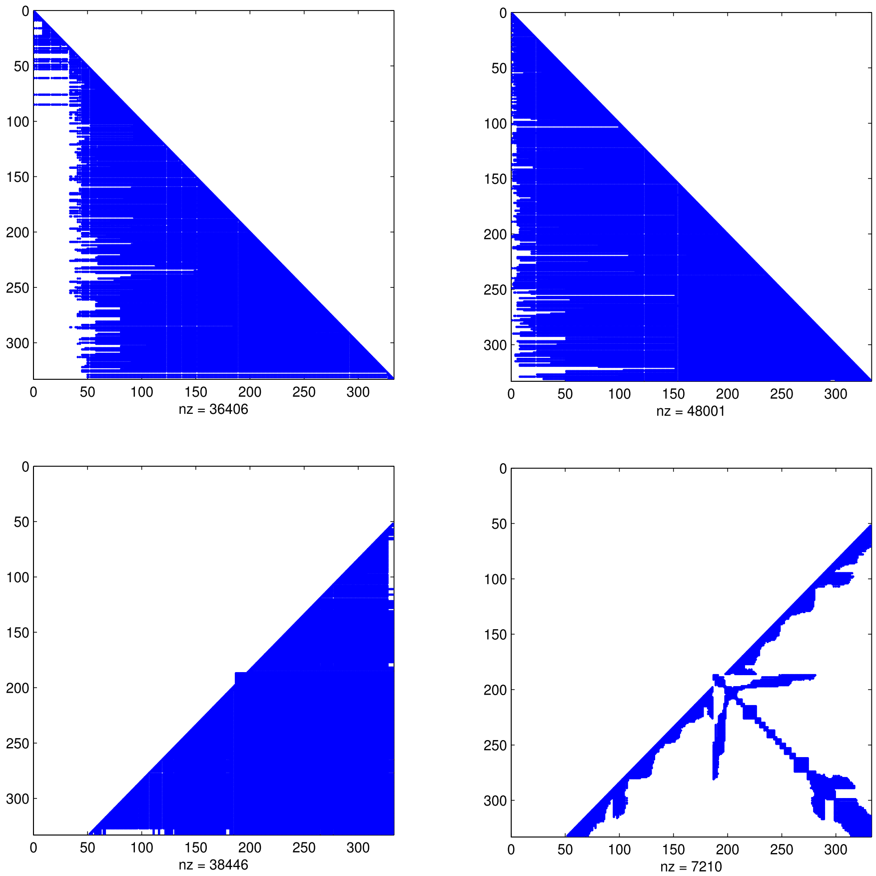

Fig. 6. Sparsity structure of the matrices $\tilde{L}, \hat{L}, M_S$ and M_H in Example 3.

Table 3
Relative residuals of the factorizations computed by the LDL^T factorization with partial pivoting, by the LDL^T factorization with rook pivoting, by the S-BAT and H-BAT algorithms, respectively, for different values of τ.

τ	$\frac{\|\tilde{P}^T\tilde{L}\tilde{D}\tilde{L}^T\tilde{P}-A\|_2}{\|A\|_2}$	$\frac{\|\hat{P}^T\hat{L}\hat{D}\hat{L}^T\hat{P}-A\|_2}{\|A\|_2}$	$\frac{\|Q_S A_S A_S^T-A\|_2}{\|A\|_2}$	$\frac{\|Q_H A_H A_H^T-A\|_2}{\|A\|_2}$
1.0×10^{-13}	2.05×10^{-16}	1.60×10^{-16}	8.47×10^{-15}	1.21×10^{-15}
1.0×10^{-14}	2.05×10^{-16}	1.60×10^{-16}	3.98×10^{-15}	1.21×10^{-15}
1.0×10^{-15}	2.05×10^{-16}	1.60×10^{-16}	1.77×10^{-15}	1.21×10^{-15}
1.0×10^{-16}	2.05×10^{-16}	1.60×10^{-16}	1.94×10^{-15}	1.21×10^{-15}

Table 4
Inertias of the matrices $\tilde{D}, \hat{D}, M_S$ and M_H computed by the LDL^T factorization with partial pivoting, by the LDL^T factorization with rook pivoting, by the S-BAT and H-BAT algorithms, respectively, for different values of τ.

τ	Inertia$(\tilde{D},\tau)$	Inertia$(\hat{D},\tau)$	Inertia(M_S,τ)	Inertia(M_H,τ)
1.0×10^{-13}	(147, 50, 135)	(147, 50, 135)	(146, 51, 135)	(146, 51, 135)
1.0×10^{-14}	(147, 50, 135)	(147, 50, 135)	(146, 51, 135)	(146, 51, 135)
1.0×10^{-15}	(149, 47, 136)	(149, 47, 136)	(146, 51, 135)	(146, 51, 135)
1.0×10^{-16}	(150, 46, 136)	(150, 46, 136)	(146, 51, 135)	(146, 51, 135)

the estimation of the numerical inertia, the considered tests show that H-BAT performs best. Moreover, the computed antitriangular matrix is sparser than the ones provided by the other considered algorithms.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments.

References

[1] C. Ashcraft, R.G. Grimes, J.G. Lewis, Accurate symmetric indefinite linear equation solvers, SIAM J. Matrix Anal. Appl. 20 (2) (1998) 513–561.
[2] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[3] Z. Bujanović, D. Kressner, A block algorithm for computing antitriangular factorizations of symmetric matrices, Numer. Algorithms 71 (1) (2016) 41–57.
[4] J.R. Bunch, L. Kaufman, Some stable methods for calculating inertia and solving symmetric linear systems, Math. Comput. 31 (137) (1977) 163–179.
[5] T.A. Davis, Y. Hu, The university of Florida sparse matrix collection, ACM Trans. Math. Softw. 38 (1) (2011) 1:1–1:25.
[6] P.E. Gill, G.H. Golub, W. Murray, M.A. Saunders, Methods for modifying matrix factorizations, Math. Comput. 28 (126) (1974) 505–535.
[7] G.H. Golub, C.F. Van Loan, Matrix Computations, 4th ed., Johns Hopkins University Press, Baltimore, 2013.
[8] N.J. Higham, The matrix computation toolbox, http://www.ma.man.ac.uk/~higham/mctoolbox.
[9] N. Mastronardi, P. Van Dooren, Recursive approximation of the dominant eigenspace of an indefinite matrix, J. Comput. Appl. Math. 236 (2012) 4090–4104.
[10] N. Mastronardi, P. Van Dooren, The antitriangular factorization of symmetric matrices, SIAM J. Matrix Anal. Appl. 34 (2013) 173–196.
[11] J. Nocedal, S.J. Wright, Numerical Optimization, Springer Series in Operations Research and Financial Engineering, Springer, Berlin, 2006.
[12] J. Pestana, A. Wathen, The antitriangular factorization of saddle point matrices, SIAM J. Matrix Anal. Appl. 35 (2) (2014) 339–353.