Journal of Computational and Applied Mathematics 252 (2013) 12–20

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

On new algorithms for inverting Hessenberg matrices

J. Abderramán Marrero [a], M. Rachidi [b], V. Tomeo [c,*]

[a] Department of Mathematics Applied to Information Technologies, (ETSIT-UPM) Telecommunication Engineering School, Technical University of Madrid, Avda Complutense s/n. Ciudad Universitaria, 28040 Madrid, Spain

[b] Group DEFA - Department of Mathematics and Informatics, Faculty of Sciences, University Moulay Ismail, B.P. 4010, Beni M'hamed, Meknes, Morocco

[c] Department of Algebra, (EUE-UCM) School of Statistics, University Complutense, Avda de Puerta de Hierro s/n. Ciudad Universitaria, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 5 July 2012
Received in revised form 21 September 2012

MSC:
15A09
15A15
15A23
65F05
65Y20

Keywords:
Computational complexity
Hessenberg matrix
Inverse matrix
Matrix factorization

ABSTRACT

A modification of the Ikebe algorithm for computing the lower half of the inverse of an (unreduced) upper Hessenberg matrix, extended to compute the entries of the superdiagonal, is considered in this paper. It enables us to compute the inverse of a quasiseparable Hessenberg matrix in $O(n^2)$ times. A new factorization expressing the inverse of a nonsingular Hessenberg matrix as a product of two suitable matrices is obtained. Because this allows us the use of back substitution for the inversion of triangular matrices, the inverse is computed with complexity $O(n^3)$. Some comparisons with results obtained using other recent inversion algorithms are also provided.

1. Introduction

The important role of Hessenberg matrices in computational and applied mathematics is well known. In particular they arise in numerical linear algebra, as a result of the application of Givens or Householder orthogonal transformations to a general matrix, when solving the eigenvalue problem, [1,2]. Furthermore the search for fast and simple algorithms for the inversion of such structured matrices is of current interest. The Ikebe algorithm, [3], yields the entries of the upper half of the inverse of any (unreduced) lower Hessenberg matrix with complexity $O(n^2)$. This algorithm provides all the entries of the inverse if the involved matrix is a tridiagonal one. Currently used algorithms utilized specifically for the inversion of tridiagonal matrices are considered in [4].

Two algorithms with complexity $O(n^3)$ have been recently introduced, [5,6], for computing the inverse matrix and the determinant of any (unreduced) nonsingular lower Hessenberg matrix; i.e. with superdiagonal entries $h_{i,i+1} \neq 0$, $(i = 1, 2, \ldots, n-1)$. The method provided in [6] is simpler. Although the procedure described in [5] has a minor flop count, the algorithm given there is more complex because of the way it achieves the inversion of the expanded triangular matrix.

Results about the existence of representations of the inverses of Hessenberg matrices as rank one perturbations of triangular matrices, i.e. in the form $\mathbf{H}^{-1} = \mathbf{T} + \vec{u} \cdot \vec{v}^T$, are known; see e.g. [7,8]. Here the matrix $\mathbf{T}$ represents a particular

* Corresponding author. Tel.: +34 913944055.
E-mail addresses: jc.abderraman@upm.es (J. Abderramán Marrero), mu.rachidi@hotmail.fr (M. Rachidi), tomeo@estad.ucm.es (V. Tomeo).

0377-0427/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2012.11.003

triangular matrix. A constructive example of such a representation for the inverse of a nonsingular lower Hessenberg matrix is given in Theorem 1 of [5].

The aim of this paper is to propose a new characterization of the nonsingular Hessenberg matrices through the factorization

$$\mathbf{H}^{-1} = \mathbf{H}_L \cdot \mathbf{U}^{-1}, \tag{1}$$

for their inverses. A constructive procedure for computing such factorization is also provided.

The matrix $\mathbf{H}_L$ is quasiseparable; i.e. rank $(\mathbf{H}_L(i+1:n,1:i)) \leq 1$, rank $(\mathbf{H}_L(1:i,i+1:n)) \leq 1$, and $i = 1, 2, \ldots, n-1$; see e.g. [9]. The matrix $\mathbf{U}$ is upper triangular, with ones on its main diagonal. Without loss of generality we consider upper Hessenberg matrices. Analogous results can be obtained for the lower Hessenberg case by taking transposes. In addition, we assume that $h_{i+1,i} \neq 0$ $(i = 1, 2, \ldots, n-1)$; i.e. the matrix $\mathbf{H}$ is unreduced. The reduced case can be handled in a similar way by partitioning the matrix in blocks in an appropriate manner.

The nonsingular lower Hessenberg matrix $\mathbf{H}_L$ is obtained directly by a simple extension of the Ikebe algorithm to the entries of the superdiagonal of $\mathbf{H}^{-1}$. The computational complexity of our proposed algorithm for the inversion of unreduced Hessenberg matrices is equivalent to that of back substitution for the entries of $\mathbf{U}^{-1}$, see e.g. [1,2], plus an additional $O(n^2)$ term.

The procedure introduced here can also be used to obtain a factorization

$$\mathbf{H} = \mathbf{U} \cdot \mathbf{H}_U \tag{2}$$

of the original matrix $\mathbf{H}$. The matrix $\mathbf{H}_U = \mathbf{H}_L^{-1}$ is a quasiseparable nonsingular upper Hessenberg matrix.

When $\mathbf{H}$ is also quasiseparable the back substitution stage can be avoided. In this situation, the expanded Ikebe algorithm provides a faster computation of the inverse matrix with complexity $O(n^2)$.

The structure of the paper is as follows. In Section 2, after recalling the Ikebe algorithm, we demonstrate the factorization (1) and show how to compute the inverse matrix $\mathbf{H}^{-1}$ using the algorithm detailed in Appendix A. In Section 3 a customary example and graphical comparisons of the elapsed times are introduced for quasiseparable Hessenberg matrices, and also for matrices associated to the upper Hessenberg form of nonsingular matrices taking on random values in $(-5; 5)$. Some conclusions are outlined at the end.

2. An extension of the Ikebe algorithm for computing the inverses of Hessenberg matrices

2.1. The Ikebe algorithm for the lower half of the inverse of nonsingular Hessenberg matrices

In order to obtain the inverse factorization (1), we begin with the Ikebe algorithm from [3], adapted here to an (unreduced) nonsingular upper Hessenberg matrix of order n. It gives us the lower half of the inverse matrix $\mathbf{H}^{-1}$, i.e. $h_{i,j}^{(-1)}$ with $i \geq j$. Following [3], we have

$$h_{i,j}^{(-1)} = y(i) \cdot x(j); \quad i \geq j, \tag{3}$$

where $y(i)$ and $x(j)$ are the ith and jth components of the vectors $\vec{y}$ and $\vec{x}$, respectively.

The components of the vector $\vec{x}$ were achieved in the following recursive way, with $h_{j,j-1}^{-1} = 1/h_{j,j-1}$,

$$x(1) = \lambda \neq 0 \quad \text{(an arbitrary constant)},$$

$$x(j) = -h_{j,j-1}^{-1} \sum_{k=1}^{j-1} h_{k,j-1} x(k) \quad (j = 2, 3, \ldots, n). \tag{4}$$

The components of the vector $\vec{y}$ were given by the following recurrence,

$$y(n) = \left(\sum_{k=1}^{n} h_{k,n} x(k) \right)^{-1},$$

$$y(i) = -h_{i+1,i}^{-1} \sum_{k=i+1}^{n} h_{i+1,k} y(k) \quad (i = n-1, n-2, \ldots, 1). \tag{5}$$

In addition, we can recover from $y(n)$ the value of $\det \mathbf{H}$, the determinant of $\mathbf{H}$, with the convention $\det \mathbf{H}_0^{(n)} = 1$, by

$$\det \mathbf{H} = (-1)^{n-1} \frac{\left(\prod_{k=2}^{n} h_{k,k-1} \right)}{\lambda \cdot y(n)}. \tag{6}$$

We define now a lower triangular matrix $\mathbf{L}^*$ with its ij entry $(i \geq j)$ given by (3). These entries constitute the lower half of $\mathbf{H}^{-1}$.

Proposition 1. *Let $\mathbf{L}^*$ be a lower triangular matrix obtained from the Ikebe algorithm, given by Eqs. (3)–(5), applied to an (unreduced) nonsingular upper Hessenberg matrix $\mathbf{H}$. Then, the following statements are equivalent:*

1. $\mathbf{L}^*$ *is nonsingular.*
2. $\mathbf{H}^{-1}$ *has a particular LU factorization of the form* $\mathbf{H}^{-1} = \mathbf{L}^* \cdot \mathbf{U}^{*-1}$, *with* $\mathbf{U}^*$ *a nonsingular upper triangular matrix.*

Proof. $1 \Rightarrow 2$. Since the lower triangular matrix $\mathbf{L}^*$ is nonsingular, all its diagonal entries are nonzero. Then the matrix product $\mathbf{U}^* = \mathbf{H} \cdot \mathbf{L}^*$ is also nonsingular. However, the entries $u^*_{i,j}$ with $i > j$ are null, because $\mathbf{L}^*$ gives us the lower half of $\mathbf{H}^{-1}$, and $\mathbf{H} \cdot \mathbf{H}^{-1} = \mathbf{I}_n$. Hence, $\mathbf{U}^*$ is upper triangular, and we have $\mathbf{H}^{-1} = \mathbf{L}^* \cdot \mathbf{U}^{*-1}$.

$2 \Rightarrow 1$. The nonsingularity of $\mathbf{L}^*$ is immediate from its expression, $\mathbf{L}^* = \mathbf{H}^{-1} \cdot \mathbf{U}^*$, as the product of two nonsingular matrices. □

Remark 1. The matrix $\mathbf{L}^*$, obtained by applying the Ikebe algorithm to a nonsingular lower Hessenberg matrix could be a lower singular matrix. In such a case, we cannot obtain the particular *LU* factorization of the inverse matrix using the Ikebe algorithm, as the following example illustrates.

Example 1. Given a nonsingular upper Hessenberg matrix $\mathbf{H}$ of order 4, we obtain a singular lower triangular matrix $\mathbf{L}^*$ using the Ikebe algorithm with $\lambda = 1$:

$$\mathbf{H} = \begin{pmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}; \qquad \mathbf{L}^* = \begin{pmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.50 & 0.00 & 0.00 \\ 0.25 & 0.25 & 0.00 & 0.00 \\ 0.25 & 0.25 & 0.00 & 0.50 \end{pmatrix}.$$

2.2. The expanded Ikebe algorithm

To overcome the previous drawback of the Ikebe algorithm, we compare Eqs. (3)–(5) with the closed form representation for the entries of the inverse of a nonsingular upper Hessenberg matrix; see [10, Corollary 1]. In this way, we can extend the Ikebe algorithm to obtain the entries of the superdiagonal, $h^{(-1)}_{i,i+1}$, $i = 1, \ldots, n-1$, of $\mathbf{H}^{-1}$. First, we introduce two results necessary in the following.

Lemma 1. *A compact representation for the components of the vector $\vec{x}$ from the Ikebe algorithm is given by*

$$x(i) = (-1)^{i-1} \frac{\lambda \cdot \det \mathbf{H}_{i-1}}{\left(\prod_{m=2}^{i} h_{m,m-1}\right)}, \quad 1 \le i \le n, \tag{7}$$

where $\det \mathbf{H}_{i-1}$ *is the determinant of the left principal submatrix of order* $i-1$.

Proof. We take $\det \mathbf{H}_0 = 1$, and the usual conventions about products. Trivially $x(1) = \lambda$, in agreement with the initial condition of the recurrence (4). Also, the representation (7) for $x(i)$ satisfies the recurrence relation (4),

$$\begin{aligned} x(i) &= -h^{-1}_{i,i-1} \sum_{k=1}^{i-1} h_{k,i-1} (-1)^{k-1} \frac{\lambda \cdot \det \mathbf{H}_{k-1}}{\left(\prod_{m=2}^{k} h_{m,m-1}\right)} \\ &= \frac{(-1)^{i-1}\lambda}{\left(\prod_{m=2}^{i} h_{m,m-1}\right)} \sum_{k=1}^{i-1} h_{k,i-1} (-1)^{k+i-1} \det \mathbf{H}_{k-1} \left(\prod_{m=k+1}^{i-1} h_{m,m-1}\right) \\ &= \frac{(-1)^{i-1}\lambda}{\left(\prod_{m=2}^{i} h_{m,m-1}\right)} \sum_{k=1}^{i-1} h_{k,i-1} A^{(i-1)}_{k,i-1}, \end{aligned}$$

where $A^{(i-1)}_{k,i-1}$ is the cofactor of the element $h_{k,i-1}$ in the matrix $\mathbf{H}_{i-1}$. Hence,

$$x(i) = (-1)^{i-1} \frac{\lambda \cdot \det \mathbf{H}_{i-1}}{\left(\prod_{m=2}^{i} h_{m,m-1}\right)}.$$

The proof follows by the uniqueness of the solution of the recurrence (4) with the initial condition $x(1) = \lambda$, an arbitrary constant. □

Lemma 2. *A compact representation for the components of the vector $\vec{y}$ from the Ikebe algorithm is given by*

$$y(i) = (-1)^{i-1} \frac{\left(\prod_{m=2}^{i} h_{m,m-1}\right) \cdot \det \mathbf{H}_{n-i}^{(i)}}{\lambda \cdot \det \mathbf{H}}, \quad 1 \le i \le n, \tag{8}$$

where $\det \mathbf{H}_{n-i}^{(i)}$ *is the determinant of the right principal submatrix of order* $n-i$*, which begins in the* $i+1$*-th row and column and finishes in the* n*-th row and column.*

Proof. For the proof we use the recurrence relation (5) and Lemma 1, the usual conventions about products, and we take $\det \mathbf{H}_0^{(n)} = 1$. Substituting Eq. (7) in the initial condition from (5),

$$\begin{aligned} y(n) &= \left(\sum_{k=1}^{n} h_{k,n} (-1)^{k-1} \frac{\lambda \cdot \det \mathbf{H}_{k-1}}{\left(\prod_{m=2}^{k} h_{m,m-1}\right)} \right)^{-1} \\ &= \frac{(-1)^{n-1} \left(\prod_{m=2}^{n} h_{m,m-1}\right) \det \mathbf{H}_0^{(n)}}{\lambda \cdot \sum_{k=1}^{n} h_{k,n} (-1)^{n+k} \det \mathbf{H}_{k-1} \left(\prod_{m=k+1}^{n} h_{m,m-1}\right)} \\ &= \frac{(-1)^{n-1} \left(\prod_{m=2}^{n} h_{m,m-1}\right) \det \mathbf{H}_0^{(n)}}{\lambda \cdot \sum_{k=1}^{n} h_{k,n} A_{k,n}^{(n)}} \\ &= \frac{(-1)^{n-1} \left(\prod_{m=2}^{n} h_{m,m-1}\right) \det \mathbf{H}_0^{(n)}}{\lambda \cdot \det \mathbf{H}}, \end{aligned}$$

in agreement with Eq. (8) for $i = n$. Also, the representation (8) for $y(i)$ satisfies the recurrence relation (5),

$$\begin{aligned} y(i) &= -h_{i+1,i}^{-1} \sum_{k=i+1}^{n} h_{i+1,k} (-1)^{k-1} \frac{\left(\prod_{m=2}^{k} h_{m,m-1}\right) \cdot \det \mathbf{H}_{n-k}^{(k)}}{\lambda \cdot \det \mathbf{H}} \\ &= \frac{(-1)^{i-1} \left(\prod_{m=2}^{i} h_{m,m-1}\right)}{\lambda \cdot \det \mathbf{H}} \sum_{k=1}^{n} h_{i+1,k} (-1)^{i+1+k} \prod_{m=i+1}^{k} h_{m,m-1} \det \mathbf{H}_{n-k}^{(k)} \\ &= \frac{(-1)^{i-1} \left(\prod_{m=2}^{i} h_{m,m-1}\right)}{\lambda \cdot \det \mathbf{H}} \sum_{k=1}^{n} h_{i+1,k} A_{i+1,k}^{(n-i)}, \end{aligned}$$

where $A_{i+1,k}^{(n-i)}$ is the cofactor of the element $h_{i+1,k}$ in the matrix $\mathbf{H}_{n-i}^{(i)}$. Hence,

$$y(i) = (-1)^{i-1} \frac{\left(\prod_{m=2}^{i} h_{m,m-1}\right) \cdot \det \mathbf{H}_{n-i}^{(i)}}{\lambda \cdot \det \mathbf{H}}.$$

The proof follows by the uniqueness of the solution of the recurrence (5). □

A representation for the entries of the superdiagonal of the matrix $\mathbf{H}^{-1}$ is now introduced by using the two vectors $\vec{x}$ and $\vec{y}$ from the Ikebe algorithm.

Proposition 2. *The entries for the superdiagonal of the inverse of an (unreduced) nonsingular upper Hessenberg matrix* $\mathbf{H}$ *can be represented as*

$$h_{i,i+1}^{(-1)} = y(i) \cdot x(i+1) + h_{i+1,i}^{-1}; \quad 1 \le i \le n-1, \tag{9}$$

where $y(i)$ *and* $x(i+1)$*, given by (5) and (4), respectively, are obtained from the Ikebe algorithm.*

Proof. First, because $\mathbf{H}$ is unreduced, all the $h^{-1}_{i+1,i}$ are well defined. From [10, Eq. (8)], we have, for $1 \le i \le n-1$,

$$h^{(-1)}_{i,i+1} = -\frac{\det \mathbf{H}_i \cdot \det \mathbf{H}^{(i)}_{n-i}}{h_{i+1,i} \cdot \det \mathbf{H}} + \frac{1}{h_{i+1,i}}. \tag{10}$$

Taking the product $y(i) \cdot x(i+1)$ by using the representations (7) and (8), we have

$$y(i) \cdot x(i+1) = -\frac{\det \mathbf{H}_i \cdot \det \mathbf{H}^{(i)}_{n-i}}{h_{i+1,i} \cdot \det \mathbf{H}}. \tag{11}$$

Eq. (9) for the entries of the superdiagonal of $\mathbf{H}^{-1}$ follows by substituting the product (11) in the representation (10). $\square$

Furthermore, since $h^{(-1)}_{i,i+1} = \frac{1}{h_{i+1,i}}\left(1 - \frac{\det \mathbf{H}_i \cdot \det \mathbf{H}^{(i)}_{n-i}}{\det \mathbf{H}}\right)$, the entry $h^{(-1)}_{i,i+1}$ is null $\left(h^{(-1)}_{i,i+1} = 0\right)$ when $\det \mathbf{H}_i \cdot \det \mathbf{H}^{(i)}_{n-i} = \det \mathbf{H}$.

Obtaining $h^{(-1)}_{i,i+1}$ by using (9) requires no additional computational effort. We call this procedure for obtaining the quasiseparable lower Hessenberg matrix $\mathbf{H}_L$ using Eqs. (3)–(9) the *expanded Ikebe algorithm*. Moreover, such a matrix allows us to propose our main result.

Theorem 1. *Let* $\mathbf{H}$ *be a nonsingular matrix of order* n. *Then the following statements are equivalent:*

1. $\mathbf{H}$ *is an upper Hessenberg matrix.*
2. *The inverse matrix* $\mathbf{H}^{-1}$ *has a factorization of the form given in (1), i.e.* $\mathbf{H}^{-1} = \mathbf{H}_L \cdot \mathbf{U}^{-1}$, *where the lower Hessenberg matrix* $\mathbf{H}_L$ *is quasiseparable, and* $\mathbf{U}^{-1}$ *is upper triangular with ones on its main diagonal.*

Proof. $1 \Rightarrow 2$. For unreduced matrices, the lower Hessenberg matrix $\mathbf{H}_L$ is obtained by applying the expanded Ikebe algorithm to the matrix $\mathbf{H}$. In the reduced case, the matrix $\mathbf{H}_L$ can be obtained by partitioning $\mathbf{H}$ into blocks in an appropriate manner. Thus the entries $h^{(-1)}_{i,j}$ ($1 \le i \le n$ and $1 \le j \le i+1$) of $\mathbf{H}_L$ are equal to the corresponding entries of $\mathbf{H}^{-1}$. Note that the matrix $\mathbf{H}_L$ is also quasiseparable. The rank conditions rank $(\mathbf{H}_L(1:i, i+1:n)) \le 1$ are trivially satisfied because $\mathbf{H}_L$ is a lower Hessenberg matrix. The conditions rank $(\mathbf{H}_L(i+1:n, 1:i)) \le 1$ follow by the fact that the lower half of $\mathbf{H}_L$ is identical to the lower half of the rank one matrix $\vec{y} \cdot \vec{x}^T$ from the Ikebe algorithm. In addition, as $\mathbf{H} \cdot \mathbf{H}^{-1} = \mathbf{I}_n$, we have

$$\mathbf{H} \cdot \mathbf{H}_L = \mathbf{U} = \left(\begin{array}{cc|cccc} 1 & 0 & u_{13} & \dots & u_{1,n-1} & u_{1n} \\ 0 & 1 & u_{23} & \dots & u_{2,n-1} & u_{2n} \\ \hline 0 & 0 & 1 & \dots & u_{3,n-1} & u_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & u_{n-1,n} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{array}\right), \tag{12}$$

an upper triangular matrix with ones on its main diagonal. Therefore, $\mathbf{U}$ is nonsingular, $\det \mathbf{U} = 1$. Thus, from (12), $\det \mathbf{H}^{-1} = \det \mathbf{H}_L$, and $\mathbf{H}_L$ is nonsingular. Finally, Eq. (12) yields $\mathbf{H}^{-1} = \mathbf{H}_L \cdot \mathbf{U}^{-1}$.

$2 \Rightarrow 1$. As $\mathbf{H}^{-1} = \mathbf{H}_L \cdot \mathbf{U}^{-1}$, we obtain Expression (2) for the nonsingular matrix $\mathbf{H} = \mathbf{U} \cdot \mathbf{H}_U$. Therefore, $h_{i,j} = (\mathbf{U} \cdot \mathbf{H}_U)_{i,j} = 0$, for $j < i-1$. Thus $\mathbf{H}$ is an upper Hessenberg matrix. $\square$

An equivalent result holds for nonsingular lower Hessenberg matrices.

Remark 2. For an unreduced (respectively reduced) Hessenberg matrix $\mathbf{H}$, nothing is said about the matrix $\mathbf{H}_L$. It could be reduced or unreduced. For an unreduced matrix $\mathbf{H}$, see the comment after the proof of Proposition 2 and Example 2. However, for an unreduced (respectively reduced) Hessenberg matrix $\mathbf{H}$, the matrix $\mathbf{H}_U$ involved in the factorization (2) is also an unreduced (respectively reduced) matrix. Indeed, for $j = i-1$, we have $h_{i,i-1} = (\mathbf{U} \cdot \mathbf{H}_U)_{i,i-1}$. That is, $h_{i,i-1}$ is equal to the $i, i-1$ entry of $\mathbf{H}_U$.

Example 2. In order to obtain $\mathbf{H}_L$ we apply the expanded Ikebe algorithm, taking $\lambda = 1$, to the unreduced Hessenberg matrix given in Example 1. The matrix $\mathbf{H}_L$ contains the lower half plus the superdiagonal of the inverse matrix $\mathbf{H}^{-1}$. The inverse factorization (1) for $\mathbf{H}^{-1}$ is:

$$\mathbf{H}^{-1} = \mathbf{H}_L \cdot \mathbf{U}^{-1} = \begin{pmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.50 & -1.00 & 0.00 \\ 0.25 & 0.25 & 0.00 & -0.50 \\ 0.25 & 0.25 & 0.00 & 0.50 \end{pmatrix} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Note that in this example the quasiseparable matrix $\mathbf{H}_L$ is reduced.

2.3. Inversion procedure for triangular matrices

The complexity of both the method given in [5] and that proposed here is equivalent to that of the inversion of a triangular matrix. We apply a vector approach, similar to that given in [5,6], for the inversion of this class of matrices. Without loss of generality we take $\mathbf{U}$ to be a nonsingular upper triangular matrix of order n.

Denote the columns of $\mathbf{I}_n$ by $(E_1, E_2, \ldots, E_n)$, and those of $\mathbf{U}^{-1}$ by $(C_1, C_2, \ldots, C_n)$.

Since $\mathbf{U}^{-1} \cdot \mathbf{U} = \mathbf{I}_n$, the following vector recurrence relation yields the columns of $\mathbf{U}^{-1}$:

$$C_1 = u_{1,1}^{-1} \cdot E_1$$

$$C_j = u_{j,j}^{-1} \cdot \left(E_j - \sum_{k=1}^{j-1} C_k \cdot u_{k,j} \right) \quad (j = 2, 3, \ldots, n) . \tag{13}$$

This is a particular vectorial back substitution scheme for the entries of the inverse of an upper triangular matrix $\mathbf{U}$, [1,2]. Eq. (13) gives rise to a simpler symbolic inversion procedure, as can be seen from the examples compared in Section 3.

Now we are ready to introduce the proposed algorithm, detailed in Appendix A, for computing and factoring the inverse matrix $\mathbf{H}^{-1}$.

2.4. The case of quasiseparable Hessenberg matrices

If $\mathbf{H}$ is a quasiseparable matrix, our inversion procedure is faster than the algorithms proposed in [5,6]. Suppose that an upper Hessenberg matrix $\mathbf{H}$ satisfies rank $(\mathbf{H}(i+1:n, 1:i)) \le 1$, rank $(\mathbf{H}(1:i, i+1:n)) \le 1$, and $i = 1, 2, \ldots, n-1$. The factorization (2) for $\mathbf{H}$ yields $\mathbf{H} = \mathbf{H}_U$, and the inverse factorization (1) reduces to $\mathbf{H}^{-1} = \mathbf{H}_L$. Therefore, the inverse matrix is obtained directly from the expanded Ikebe algorithm.

3. Numerical examples and elapsed time comparisons

The results given in this section were obtained using the commercial *Matlab*® package on a 1.80 GHz computer.

3.1. Quasiseparable Hessenberg matrices

As an illustration we consider the case of an upper Hessenberg matrix of order $n = 5$. It is the transpose of a standard example given in [5,6]:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \\ 0 & -1 & 1 & 1 & 1 \\ 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} .$$

This upper Hessenberg matrix and also its transpose belong to the class of quasiseparable matrices from Section 2.4, i.e. rank $(\mathbf{H}(i+1:5, 1:i)) \le 1$, rank $(\mathbf{H}(1:i, i+1:5)) \le 1$, and $i = 1, 2, \ldots, 4$. Hence, $\mathbf{H}^{-1}$ can be computed with complexity $O(n^2)$ using the expanded Ikebe algorithm:

```
>> H^(-1) = 0.5000   -0.5000         0         0         0
            0.2500    0.2500   -0.5000         0         0
            0.1250    0.1250    0.2500   -0.5000         0
            0.0625    0.0625    0.1250    0.2500   -0.5000
            0.0625    0.0625    0.1250    0.2500    0.5000
```

The outcomes give the exact values of the inverse matrix. As was expected, the inverse is a quasiseparable lower Hessenberg matrix.

Fig. 1 provides a graphical comparison of the times elapsed in the computation of the inverse of a quasiseparable Hessenberg matrix with entries $h_{i,j} = -1$ for $i = j + 1$, $h_{i,j} = -2.5$ for $i \le j$, and $h_{i,j} = 0$, otherwise. The matrix order n runs from 15 to 155 in steps of size 10.

The algorithm for the Elouafi and Hadj method is given in Appendix B. A simple Ikebe-like procedure for obtaining the last row of $\mathbf{H}^{-1}$ has been introduced as a first step. It gives the initial vector for an Elouafi–Hadj stage to provide the remaining rows.

For comparison, values of *norm*$(\mathbf{H}^{-1}\mathbf{H} - \mathbf{I})$ are given in Table 1, with $\mathbf{I}$ the identity matrix. For large orders, the algorithms given in [5,6] yield inaccurate outcomes in the computation of the inverse matrix of this quasiseparable Hessenberg matrix.

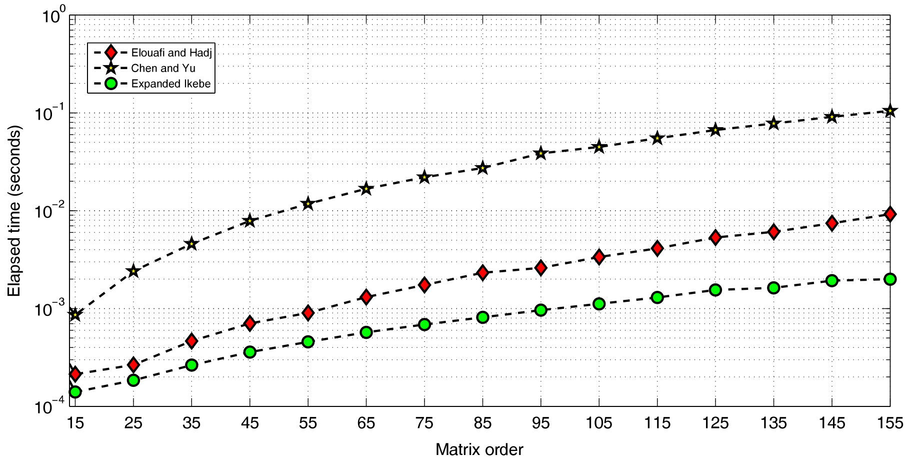

Fig. 1. Elapsed time comparison, in Log scale, for the three algorithms in the computation of the inverse of a quasiseparable Hessenberg matrix.

Table 1
Values of $norm(\mathbf{H}^{-1}\mathbf{H}-\mathbf{I})$ for the three algorithms in the computation of the inverse of a quasiseparable Hessenberg matrix.

Order	Elouafi–Hadj	Chen–Yu	Expanded Ikebe
15	1.75e−13	1.67e−13	1.68e−14
35	7.37e−10	1.54e−10	5.34e−14
55	2.45e−06	1.31e−06	8.65e−14
75	1.36e−02	5.58e−03	2.57e−13
95	4.78e+01	2.62e+01	1.49e−13
115	1.02e+05	6.15e+04	2.57e−13
135	3.09e+08	2.29e+08	7.21e−13
155	2.55e+12	1.02e+12	2.03e−12

3.2. Random Hessenberg matrices

For a comparison using more general unreduced Hessenberg matrices, we consider the associated upper Hessenberg form of nonsingular squared matrices, with random entries uniformly distributed in $(-5; 5)$. Fig. 2 provides a graphical comparison of the mean elapsed times, after 30 trials, required for the inversions of such Hessenberg matrices. In each trial the matrix order runs from 55 to 1005, in steps of 50 units. The numerical outcomes given by the algorithms from [5,6] are now more accurate.

4. Conclusions

An extension of the Ikebe algorithm for the computation and factorization of the inverses of Hessenberg matrices has been proposed. It allows us to provide in Theorem 1 a new characterization of this important class of matrices. An advantage of this procedure, with respect to the usual *LU* inversion methods, is that the introduction of a permutation matrix is not necessary when the Hessenberg matrix contains a singular principal submatrix; see Examples 1 and 2. In addition, our algorithm is faster when the Hessenberg matrix is quasiseparable.

Comparison of flop counts is frequently used to measure computational complexity. Nevertheless, as is remarked in [1], flop counting is just a "quick and dirty" accounting method that captures only one of the several dimensions of the efficiency issue. We must not infer too much from a comparison of flop counts, in particular for algorithms having the same orders of complexity. Hence, for a more precise measure of the computational complexity, we have compared the times elapsed using our algorithm with those elapsed using standard currently used algorithms described in [5,6]. The results show the capabilities of the three algorithms for computing the inverse. The simpler Elouafi–Hadj algorithm, with an Ikebe first step, gives results similar to those given by our algorithm for matrices of low and moderate orders. However, when the orders are increased our proposed algorithm is faster. This algorithm also provides for the inverse matrix $\mathbf{H}^{-1}$ the two matrices appearing in the factorization (1).

The inversion method from [5] can be improved with the introduction, in Algorithm 1 given in [5], of the simpler procedure, from Section 2.3, for the inversion of the involved triangular matrix.

The computation of $\det \mathbf{H}$ has been omitted for brevity, although it can be included in the proposed algorithm without additional computational effort using Expression (6). Thus in the comparisons, the procedure for computing the determinant given in Algorithm 2 from [5] has not been considered.

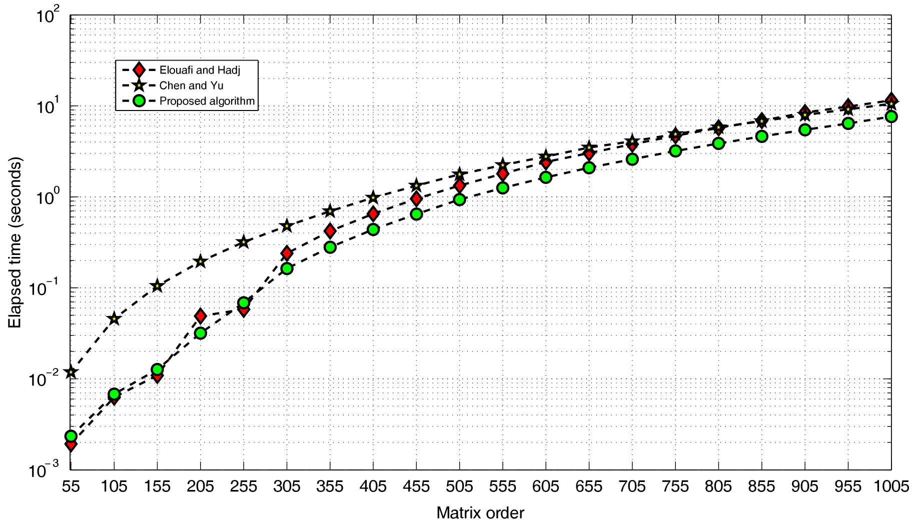

Fig. 2. Comparison of the mean value for elapsed times, in Log scale, for the three algorithms in the computation of the inverse of upper (unreduced) Hessenberg matrices.

Appendix A. An algorithm for computing and factoring H^{-1}

```
function [invH, Ike, invU] = inverseH(H, order)
%%%Preallocating matrices and vectors
Ike=zeros(order); vecx=zeros(order,1); vecy=zeros(order,1);
E=eye(order); invU=zeros(order);
%%%Step1: Computing L_{H} with the expanded Ikebe algorithm
vecx(1)=1.000; %e.g. taking lambda=1
for i=2:order
    for j=1:i-1
        vecx(i)=vecx(i)+H(j,i-1).*vecx(j);
    end
    vecx(i)=-vecx(i)./H(i,i-1);
end
for i=1:order
    vecy(order)=vecy(order)+H(i,order).*vecx(i);
end
    vecy(order)=1./vecy(order);
for i=orden-1:-1:1
    for j=i+1:order
        vecy(i)=vecy(i)+H(i+1,j).*vecy(j);
    end
        vecy(i)=-vecy(i)./H(i+1,i);
end
for j=1:order-1
    Ike(j,1:j)=vecy(j).*vecx(1:j);
    Ike(j,j+1)=vecy(j).*vecx(j+1)+1./H(j+1,j);
end
    Ike(order,1:order)=vecy(order).*vecx;
%%%Step 2: Computing invU using the vector scheme (9)
U=H*Ike; invU(:,1:2)=E(:,1:2); %from Equation (8)
for j=3:order
invU(:,j)=E(:,j)-invU(:,1:j-1)*U(1:j-1,j); %diagonal entries are ones
end
%%%Step 3: Computing invH, factorization (1)
invH=Ike*invU;
end
```

Appendix B. The Elouafi–Hadj algorithm with an Ikebe first step

```
function [invH] = alg_elo_had_ike(H, order)
%%%Preallocating matrices
invH=zeros(order); E=eye(order);
%%% Step 1: Compute the last row using an Ikebe-like procedure
invH(order,1)=1.000;
for i=2:order
    for j=1:i-1
        invH(order,i)=invH(order,i)+H(j,i-1).*invH(order,j);
    end
    invH(order,i)=-invH(order,i)./H(i,i-1);
end
vec_y=0;
for i=1:order
    vec_y=vec_y+H(i,order).*invH(order,i);
end
    invH(order,:)=invH(order,:)./vec_y;
%%%%%Step 2: Computing the remaining rows with the Elouafi-Hadj procedure
for k=1:order-1
    invH(order-k,:)=(E(order-k+1,:)-H(order-k+1,order-k+1:order)*...
                     invH(order-k+1:order,:))./H(order-k+1,order-k);
end
```

References

[1] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, Maryland, USA, 1996.
[2] D.S. Watkins, Fundamentals of Matrix Computations, second ed., Wiley, New York, USA, 2002.
[3] Y. Ikebe, On inverses of Hessenberg matrices, Linear Algebra Appl. 24 (1979) 93–97.
[4] J. Abderramán Marrero, M. Rachidi, V. Tomeo, Non-symbolic algorithms for the inversion of tridiagonal matrices, J. Comput. Appl. Math. 252 (2013) 3–11.
[5] Y.H. Chen, C.Y. Yu, A new algorithm for computing the inverse and the determinant of a Hessenberg matrix, Appl. Math. Comput. 218 (2011) 4433–4436.
[6] M. Elouafi, A.D. Aiat Hadj, A new recursive algorithm for inverting Hessenberg matrices, Appl. Math. Comput. 214 (2009) 497–499.
[7] E. Asplund, Inverses of matrices $\{a_{ij}\}$ which satisfy $a_{ij} = 0$ for $j > i + p$, Math. Scand. 7 (1959) 57–60.
[8] L. Elsner, Some observations on inverses of band matrices and low rank perturbations of triangular matrices, Acta Tech. Acad. Sci. Hung. 108 (1999) 41–48.
[9] R. Bevilaqua, E. Bozzo, G.M. Del Corso, *qd*-type methods for quasiseparable matrices, SIAM J. Matrix Anal. Appl. 32 (2011) 722–747.
[10] J. Abderramán Marrero, V. Tomeo, On the closed representation for the inverses of Hessenberg matrices, J. Comput. Appl. Math 236 (2012) 2962–2970.