DOI: 10.1016/j.camwa.2013.04.016

Computers and Mathematics with Applications 66 (2013) 56–61

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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

A short note on a generalization of the Givens transformation ☆

R. Biloti [a], L.C. Matioli [b], Jinyun Yuan [b,*]

[a] Department of Applied Mathematics, IMECC/UNICAMP & INCT-GP, Rua Sérgio Buarque de Holanda, 651, 13083-859, Campinas, SP, Brazil
[b] Department of Mathematics, UFPR Centro Politécnico, CP 19.081, 81531-990, Curitiba, PR, Brazil

ARTICLE INFO

Article history:
Received 6 September 2012
Received in revised form 6 March 2013
Accepted 6 April 2013

Keywords:
Householder transformation
Tridiagonalization
LaBudde's transformation
Givens transformation
Generalized Householder transformation
Generalized Givens transformation

ABSTRACT

A new transformation, a generalization of the Givens rotation, is introduced here. Its properties are studied. This transformation has some free parameters, which can be chosen to attain pre-established conditions. Some special choices of those parameters are discussed, mainly to improve numerical properties of the transformation.

1. Introduction

The Givens rotation, like the Householder transformation, plays an important role in scientific computing, especially for least squares problems [1] and QR decomposition [2–4], as well as in computational eigenvalue problems [5–8]. A rotation can be used to eliminate one entry of a given vector with two elements. This is employed successively to map n-dimensional vectors to multiples of the first canonical vector. The orthogonality of the Givens rotation ensures good theoretical properties and numerical performance. For instance, the standard approach to compute eigenvalues is to obtain a tridiagonal matrix, similar to the target one, for which the computational cost of iterative processes is smaller [9–14]. Usually, this is accomplished in two phases: the first is to convert the matrix, by similarity transformations (like Gaussian elimination), to an upper Hessenberg matrix [5–7]; the second is to convert the obtained Hessenberg matrix similarly to a tridiagonal matrix. This last step is usually done by orthogonal transformations such as Householder transformations or Givens rotations [9–14]. It is well known that the Givens rotation is the preferable option for large sparse scientific computing, since it affects only two rows and two columns.

In [15], the authors discussed implementation issues of LaBudde's transformation, which is a generalization of the Householder transformation [16]. They proposed one algorithm (called the GYBR algorithm) to obtain tridiagonalization directly without going via a Hessenberg matrix [15]. In their numerical tests, the GYBR algorithm works well if there is no breakdown for all steps. The GYBR algorithm converts one column and one row simultaneously become similar to multiples of the first canonical vector. Just like the Householder transformation, the GYBR algorithm affects all entries of the matrix

☆ The work of the first and third authors was partially supported by CNPq, Brazil. The work of the third author was also partially supported by CAPES, Brazil.
* Corresponding author. Tel.: +55 41 3361 3400.
E-mail addresses: biloti@ime.unicamp.br (R. Biloti), matioli@ufpr.br (L.C. Matioli), jin@ufpr.br, yuanjy@gmail.com (J. Yuan).

0898-1221/$ – see front matter
http://dx.doi.org/10.1016/j.camwa.2013.04.016

at each step, which destroys the sparsity of the original given matrix. Since Givens rotations are appropriate for sparse problems, due to their surgical changes in the matrix entries, we are motivated to generalize the Givens rotation to eliminate two entries of two given vectors simultaneously. Hence, we propose a new transformation, which we refer to as a *generalized Givens rotation* (see Definition 1 in Section 2). This transformation is not orthogonal in general. We present necessary and sufficient conditions for orthogonality of a generalized Givens transformation.

Since the tridiagonalization is not via a Hessenberg matrix, the new transformation is much cheaper in terms of computational cost (cheaper by almost half of multiplications) than the current tridiagonalization process. For the sparse case, using a generalized Givens transformation is even cheaper than using the GYBR algorithm for tridiagonalization in terms of computational cost [15].

The non-orthogonality constraint results in more flexibility in the choices of the coefficients of the transformation. An important aspect to be highlighted is that such flexibility can be exploited to attain better numerical performance, concerning many different criteria. Based on that, and guided by desired theoretical properties for the transformation, we discuss some particular choices for the free parameters. Our main contribution here is to introduce the new transformation and study its properties.

The organization of this paper is as follows. The generalized Givens rotation is introduced in Section 2. An algorithm and some properties of the new transformation are given in the same section. A necessary and sufficient condition for orthogonality is established. In Section 3, some special choices for parameters of the generalized Givens transformation are discussed. In Section 4, some numerical examples are given to illustrate the numerical performance, especially efficiency and flexibility, of the new transformation, together with some possible applications. In Section 5, we present our conclusions. Note that we simply introduce the generalized Givens transformation here. Further practical applications of the generalized Givens rotation are not the focus of this paper; they will be studied in future work.

2. Generalized Givens transformation

The usual Givens transformation is an orthogonal matrix of a rotation in two dimensions. Upon its usage, it is possible to assemble similarity transformations to tridiagonalize sparse symmetric matrices efficiently.

Suppose that the 3×3 symmetric matrix A has the following structure:

$$\left(\begin{array}{c|c} a_{11} & u^T \\ \hline u & A_{22} \end{array}\right)$$

where u is a two-dimensional vector, and A_{22} is a 2×2 symmetric matrix. Then, the structure of the similarly Givens transformation is

$$\left(\begin{array}{c|c} 1 & 0^T \\ \hline 0 & G \end{array}\right)\left(\begin{array}{c|c} a_{11} & u^T \\ \hline u & A_{22} \end{array}\right)\left(\begin{array}{c|c} 1 & 0^T \\ \hline 0 & G^T \end{array}\right) = \left(\begin{array}{c|c} a_{11} & ke_1^T \\ \hline ke_1 & \hat{A}_{22} \end{array}\right). \tag{1}$$

In this section, a generalized Givens matrix with its computational algorithm is introduced. Its properties are studied as well. Now, we consider non-symmetric matrices. Assume that the 3×3 non-symmetric matrix A is defined by

$$\left(\begin{array}{c|c} a_{11} & v^T \\ \hline u & A_{22} \end{array}\right)$$

where u and v are two two-dimensional vectors, and that A_{22} is a 2×2 non-symmetric matrix. In this case, Eq. (1) would be generalized as

$$\left(\begin{array}{c|c} 1 & 0^T \\ \hline 0 & P \end{array}\right)\left(\begin{array}{c|c} a_{11} & v^T \\ \hline u & A_{22} \end{array}\right)\left(\begin{array}{c|c} 1 & 0^T \\ \hline 0 & P^{-1} \end{array}\right) = \left(\begin{array}{c|c} a_{11} & k_2e_1^T \\ \hline k_1e_1 & \hat{A}_{22} \end{array}\right). \tag{2}$$

Definition 1. The generalized Givens matrix P is defined by

$$P = \alpha \begin{pmatrix} c & d \\ s & c \end{pmatrix}. \tag{3}$$

The inverse of P is given by

$$P^{-1} = \beta \begin{pmatrix} c & -d \\ -s & c \end{pmatrix}, \tag{4}$$

with β satisfying the equation

$$\alpha\beta(c^2 - sd) = 1. \tag{5}$$

Note that the generalized Givens matrix is the Givens matrix if $\alpha = \beta = 1$, $d = -s$, and $s^2 + c^2 = 1$. This is why we refer to P in (3) as a generalized Givens matrix.

The algorithm below computes P to satisfy Eq. (2); that is, given two two-dimensional vectors u and v, P will be determined such that

$$Pu = k_1 e_1, \quad \text{and} \quad P^{-T} v = k_2 e_1. \tag{6}$$

Algorithm 1. Step1. Given vectors $u = (x,\ y)^T$ and $v = (z,\ w)^T$ such that $zx \neq 0$ and $u^T v \neq 0$;
Step2. choose α and β such that $(\alpha\beta\ xz\ u^T v) > 0$;
Step3. compute

$$c = \sqrt{\frac{xz}{\alpha\beta u^T v}}, \tag{7a}$$

$$s = -\frac{y}{x} c, \tag{7b}$$

$$d = \frac{w}{z} c, \tag{7c}$$

$$k_1 = \alpha c \left(x + \frac{yw}{z} \right), \tag{7d}$$

$$k_2 = \beta c \left(z + \frac{yw}{x} \right). \tag{7e}$$

Note that, in Step 2, α and β are free parameters. The choice of these two parameters is discussed later.

The next theorem shows that Algorithm 1 is well defined.

Theorem 1. *Let $u = (x, y)^T$ and $v = (z, w)^T$ be such that $u^T v \neq 0$, $xz \neq 0$, and $xz/(\alpha\beta u^T v) > 0$. If P and P^{-1} are as defined in (3) and (4), respectively, with k_1 and k_2 defined by (7), then $Pu = k_1 e_1$ and $P^{-T} v = k_2 e_1$. Moreover, $\det(P) = \alpha/\beta$.*

Proof. It follows from (6) that

$$(cx + dy) = k_1/\alpha, \tag{8a}$$

$$(sx + cy) = 0, \tag{8b}$$

$$(cz - sw) = k_2/\beta, \tag{8c}$$

$$(cw - dz) = 0, \tag{8d}$$

where $u = (x,\ y)^T$ and $v = (z,\ w)^T$. Writing s and d in terms of c, from (8b) and (8d), we obtain

$$s = -\frac{y}{x} c \quad \text{and} \quad d = \frac{w}{z} c, \tag{9}$$

as stated in (7b) and (7c), respectively.

Under the assumption that $xz/(\alpha\beta u^T v) > 0$, which can be guaranteed by appropriate choices of α and β discussed later on, from (5) and (9), it is straightforward to determine the value of c as in (7a).

Replacing the expressions for c, d, and s in (8a) and (8c), we obtain the expressions (7d) and (7e) for k_1 and k_2, respectively.

Note that, from (7d) and (7e), the ratio x/z is

$$\frac{x}{z} = \frac{\beta}{\alpha} \frac{k_1}{k_2}.$$

From (3), $\det(P) = \alpha^2(c^2 - sd)$. Using (5), simple algebraic manipulation leads to $\det(P) = \alpha/\beta$. □

Remark 1. If $x = z = 0$ but $yw \neq 0$, it is still possible to obtain P such that $Pu = k_1 e_1$ and $P^{-T} v = k_2 e_1$. From (8a)–(8d), $c = 0$, $d = 1$, and $s = -1$. To obtain P with the desired property, it is enough to take $\alpha = \beta = 1$. Then $k_1 = y$ and $k_2 = w$.

From the theorem, it is obvious that $\det(P) = \det(P^{-1}) = c^2 - sd = 1$ if $\alpha = \beta = 1$. The next result follows by imposing that $P^{-1} = P^T$.

Theorem 2. *The generalized Givens matrix P is orthogonal if and only if $\alpha = \beta$ and $d = -s$. For the given non-zero vectors u and v, P is orthogonal if and only if*

$$\frac{y}{x} = \frac{w}{z}$$

whenever $xz \neq 0$.

Proof. The result follows immediately from the fact that $P^{-1} = P^T$ for orthogonal matrices. $\square$

Remark 2. P is orthogonal if $u^T v \neq 0$ and $x = z = 0$, by Remark 1.

Remark 3. Consider the task of tridiagonalizing a matrix. Suppose that the matrix A of order n has the following structure:

$$A = \begin{bmatrix} T & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

where T is a $k \times k$ tridiagonal matrix, $A_{12}^T = [0 \cdots 0 \ p]$, and $A_{21} = [0 \cdots 0 \ q]$, with both p and q being vectors in $\mathbb{R}^{(n-k)}$. To move forward one step in the tridiagonalization process, it is necessary obtain a matrix P_k such that

$$P_k A P_k^{-1} = \begin{bmatrix} T & \sigma_{12} E_{n-k}^T \\ \sigma_{21} E_{n-k} & \hat{A}_{22} \end{bmatrix},$$

where $E_j = [0 \cdots 0 \ e_1^{(j)}]$, and $e_1^{(j)}$ is the first canonical vector of dimension j. The matrix P_k can be composed as a product of generalized Givens matrices, given by

$$P_k = P_{k,n} P_{k,n-1} \cdots P_{k,k+2},$$

where each $P_{k,j}$ is

$$P_{k,j} = \begin{array}{cc}
\left(\begin{array}{ccccccccccc}
1 &&&&&&&&&& \\
&\ddots&&&&&&&&& \\
&&1&&&&&&&& \\
&&&\alpha c&&&&\alpha d&&& \\
&&&&1&&&&&& \\
&&&&&\ddots&&&&& \\
&&&&&&1&&&& \\
&&&\alpha s&&&&\alpha c&&& \\
&&&&&&&&1&& \\
&&&&&&&&&\ddots& \\
&&&&&&&&&&1
\end{array}\right) & \begin{array}{l} \\ \\ \\ \leftarrow \ k \\ \\ \\ \\ \leftarrow \ j \\ \\ \\ \\ \end{array} \\
\begin{array}{ccccccccccc} &&&\uparrow&&&&\uparrow&&& \\ &&&k&&&&j&&& \end{array} &
\end{array}, \tag{10}$$

where $c, d, s, \alpha, \beta, k_1$, and k_2 are given by (7), with $u = (a_{k+1,k}, a_{j,k})^T$ and $v = (a_{k,k+1}, a_{k,j})^T$.

3. Particular choices for α and β

The choice of the two parameters α and β is important for the generalized Givens transformation, because the freedom of choice can be exploited to attain particular computational requirements, such as optimality, or other concerns. In this section, we shall discuss some particular choices for α and β to attain some good properties for the matrix or for applications of the transformation. Note that the hypothesis of Theorem 1 holds for all choices in this section.

3.1. Equalized P and P^{-1}

To keep both P and P^{-1} with same magnitude we have to set $|\alpha| = |\beta|$. To simplify expression (7a), we can define

$$\frac{\alpha}{\operatorname{sign}(xz)} = \frac{\beta}{\operatorname{sign}(u^T v)} = \frac{1}{\sqrt{|u^T v|}}. \tag{11}$$

It is very clear that $xz/(\alpha\beta u^T v)$ is always positive for this particular choice of α and β. Substituting (11) into (7), we obtain the corollary below.

Corollary 1. *For the special choice of α and β in (11), the values for c, d, s, k_1, and k_2 are*

$$c = \sqrt{|xz|}, \qquad d = \frac{w}{z}\sqrt{|xz|}, \qquad s = -\frac{y}{x}\sqrt{|xz|}, \tag{12}$$
$$k_1 = \alpha \frac{\sqrt{|xz|}}{z} u^T v, \qquad k_2 = \beta \frac{\sqrt{|xz|}}{x} u^T v.$$

Finally, the expressions for k_1 and k_2, in terms of the input vectors only, are

$$k_1 = \operatorname{sign}(x)\operatorname{sign}(u^T v)\sqrt{\frac{|x|}{|z|}|u^T v|} \tag{13}$$

and

$$k_2 = \operatorname{sign}(x)\sqrt{\frac{|z|}{|x|}|u^T v|}. \tag{14}$$

3.2. Low computational cost

If the intention is to keep the computational cost as low as possible, an interesting choice for α and β is

$$\alpha = \frac{1}{u^T v}, \quad \text{and} \quad \beta = xz. \tag{15}$$

Clearly, the condition $xz/(\alpha\beta u^T v) > 0$ is satisfied. The expressions of all coefficients of P are simpler than for the previous choices. Hence, this choice leads to lower computational costs.

Corollary 2. *For the special choice of α and β in (15), the values for c, d, s, k_1, and k_2 are*

$$c = 1, \qquad d = \frac{w}{z}, \qquad s = -\frac{y}{x}, \qquad k_1 = \frac{1}{z}, \qquad k_2 = z(u^T v). \tag{16}$$

3.3. (Anti)symmetric output

Instead of asking properties for P, we could also look for good properties for the result of applying P. In this way, we explore choices for α and β for which $|k_1| = |k_2|$. With

$$\alpha = z, \quad \text{and} \quad \beta = \operatorname{sign}(u^T v)x \tag{17}$$

the coefficients in (7) are given by

$$c = \sqrt{1/|u^T v|}, \qquad s = \frac{y}{x}c, \qquad d = \frac{w}{z}c, \qquad k_1 = \frac{k_2}{\operatorname{sign}(u^T v)} = c(u^T v). \tag{18}$$

4. Numerical tests

In this section, we give a numerical example to illustrate the possible application and flexibility of the new transformation. Here, we suppose that there is no breakdown for all steps in the computation. The implementation was done using MATLAB on PCs with 1.8 GHz and enough memory and disk space. Each matrix considered was tridiagonalized with and without pivoting. As usual, the bigger the matrices, the greater the accuracy improvements due to the pivoting. Nevertheless, pivoting is crucial even for small cases, since the situation of $xz = 0$ or $xz + yw = 0$ can break the algorithm down.

To serve as a criterion for the quality of the generalized Givens rotation, for several matrices, we measure the accuracy of the computation of the eigenvalues after the proposed tridiagonalization with respect to those computed directly from the original matrices.

We performed several tests with many different matrices, concerning size, structure, and sparsity. Some matrices were randomly generated, by MATLAB, while others were generated with *LAPACK Test Matrix Generators DLATMR*,[1] and they are suitable for testing an eigenvalue algorithm. The following table shows a comparison of the computed eigenvalues of the original matrices and the computed eigenvalues of the tridiagonalized matrices. In the third and fourth columns we present the relative errors $\|\lambda_T - \lambda\|_2/\|\lambda\|_2$, where λ is a vector containing all eigenvalues of the original matrix and λ_T is that for the tridiagonalized matrix obtained by our algorithm. In all cases, the eigenvalues were computed by the standard MATLAB routine.

[1] This is available at http://math.nist.gov/MatrixMarket/data/misc/xlatmr/.

Problem	Matrix type	n	Relative error	
			Without pivoting	With pivoting
1	Full	6	6.8478e − 16	4.4669e − 16
2	Full	50	3.8864e − 11	5.0242e − 13
3	Sparse	50	–	9.1725e − 14
4	Full	100	1.4004e − 08	5.0773e − 14
5	Sparse	100	–	1.7619e − 15
6	Sparse	200	–	8.1781e − 12
7	Sparse	225	–	3.8786e − 11

The matrices of problems 1 to 6 have no particular structure. Three of them (1, 2, and 4) are full, and the others are sparse. The matrix of problem 7 [5,17] is related to a five-point central finite difference discretization of the two-dimensional variable-coefficient linear elliptic equation

$$-(pu_x)_x - (qu_y)_y + ru_x + (ru)_x + (su)_y + tu = f,$$

where

$$\begin{cases} p \equiv p(x, y) = \exp(-xy) \\ q \equiv q(x, y) = \exp(xy) \\ r \equiv r(x, y) = \beta(x + y) \\ t \equiv t(x, y) = 1/(1 + x + y) \end{cases}$$

and $\beta = 20$. The domain is the unit square $[0, 1] \times [0, 1]$, with Dirichlet boundary conditions. All those matrices are non-symmetric.

All numerical results confirm the good numerical performance of the transformation.

5. Conclusions

We introduce a generalized Givens rotation, which simultaneously eliminates two elements of two vectors respectively. The new transformation has two parameters, whose choice gives us freedom to attain some desired properties. We have discussed some special choices for these two parameters.

Numerical examples given here have illustrated possible applications and the flexibility of the new transformation. The numerical results gave good numerical performance of the transformation. We shall do further research on the transformation.

References

[1] A. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996.
[2] G.H. Golub, C.V. Loan, Matrix Computation, 3rd Edition, The Johns Hopkins University Press, Baltimore and London, 1996.
[3] G.W. Stewart, Introduction to Matrix Computations, Academic Press, New York, 1973.
[4] G.W. Stewart, Matrix Algorithms I: Basic Decompositions, SIAM, Philadelphia, 1998.
[5] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000.
[6] G.H. Golub, Numerical methods for large scale eigenvalue problems, Teaching note, Stanford University (1999).
[7] B. Parlette, The Symmetric Eigenvalue Problem, in: Reprinted as Classics in Applied Mathematics, 20, SIAM, Philadelphia, 1997.
[8] G.W. Stewart, Matrix Algorithms II: Eigensystems, SIAM, Philadelphia, 2001.
[9] E. Gerck, A.B. d'Oliveira, Continued fraction calculation of the eigenvalues of tridiagonal matrices arising from the Schrödinger equation, Journal of Computational and Applied Mathematics 6 (1980) 81–82.
[10] G.H. Golub, N.T. Robertson, A generalized Bairstow algorithm, Communication on Applied and Computational Mathematics 10 (1967) 371–373.
[11] Y. Im, S. Ri, An algorithm for the calculation of eigenvalues of tridiagonal matrices using QD-transformations and the LR (RL) method, Su-hak: Academy of Science of the People's Democratic Republic of Korea 2 (1995) 12–15.
[12] D. Kulkarni, D. Schmidt, S.K. Tsui, Eigenvalues of tridiagonal pseudo-toeplitz matrices, Linear Algebra and its Applications 297 (1999).
[13] L. Pasquini, R. Pavani, Computing the eigenvalues of non-normal tridiagonal matrices, Rendiconti del Seminario Matematico e Fisico di Milano 65 (1995) 109–138.
[14] K. Veselic, On real eigenvalues of real tridiagonal matrices, Linear Algebra and its Applications 27 (1979) 167–171.
[15] G.H. Golub, J.Y. Yuan, R. Biloti, J. Ramos, Optimal generalized Householder transformation with application, Tech. rep., Universidade Federal do Paraná, Brazil (2005).
[16] C.D. LaBudde, The reduction of an arbitrary real square matrix to tridiagonal form using similarity transformations, Mathematics of Computation 17 (1963) 433–437.
[17] A. Stathopolous, Y. Saad, K. Wu, Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods, SIAM Journal on Scientific Computing 19 (1998) 227–245.