HEWLETT PACKARD

Work Notes on Elementary Matrices

Augustin A. Dubrulle
Computer Systems Laboratory
HPL-93-69
July, 1993

dubrulle@hpl.hp.com

matrix computations, matrix block algorithms, linear algebra

Elementary matrices are studied in a general framework where the Gauss and Householder types are particular cases. This compendium includes an analysis of characteristic properties, some new derivations for the representation of products, and selected applications. It is the condensation of notes useful for the development and test of matrix software, including high-performance block algorithms.

Internal Accession Date Only

Work Notes on
ELEMENTARY MATRICES

AUGUSTIN A. DUBRULLE
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304
dubrulle@hpl.hp.com

July 1993

ABSTRACT

Elementary matrices are studied in a general framework where the Gauss and Householder types are particular cases. This compendium includes an analysis of characteristic properties, some new derivations for the representation of products, and selected applications. It is the condensation of notes useful for the development and test of matrix software, including high-performance block algorithms.

1 Introduction

Elementary transformations [7, 8, 13] are the basic building blocks of numerical linear algebra. The most useful transformations for the solution of linear equations and eigenvalue problems are represented by two types of elementary matrices. Householder matrices (or reflectors), which constitute the first type, preserve the euclidean norm and are unconditionally stable. Gauss matrices, which make up the second type, are economical of computation, but are not generally stable. The product of a Gauss matrix and a well-chosen, data-dependent, transposition matrix (another elementary matrix) defines a transformation the instability of which is contained, and which sometimes constitutes a practical alternative to Householder reflections. Such "stabilized" elementary transformations are the basis for algorithms using techniques of pivoting in the solution of linear equations and eigenvalue problems [13].

Real Householder and Gauss transformations are members of a larger class represented by the generic matrix

$$\mathbf{T} = \mathbf{I} + \mathbf{v}\mathbf{w}^T, \qquad \mathbf{v}, \mathbf{w} \in \mathcal{R}^n, \qquad \mathbf{T} \in \mathcal{R}^{n \times n}, \tag{1.1}$$

where the vectors $\mathbf{v}$ and $\mathbf{w}$ are arbitrary. Typically, elementary transformations are used for the annihilation of selected elements of a vector or a matrix. The transformation of a nonzero vector

$$\mathbf{y} = \mathbf{T}\mathbf{x}, \qquad \mathbf{x} \neq \mathbf{0}, \qquad \mathbf{x}, \mathbf{y} \in \mathcal{R}^n,$$

defines $\mathbf{T}$ from equation (1.1) by the following relations:

$$\mathbf{v} = \frac{\mathbf{y} - \mathbf{x}}{\mathbf{w}^T\mathbf{x}}, \qquad \mathbf{w}^T\mathbf{x} \neq 0.$$

The elementary matrix so specified is not unique, and additional constraints have to be placed on $\mathbf{w}$ to lift the uncertainty. Yet, selected components of $\mathbf{x}$ can be annihilated by $\mathbf{T}$ with

$$v_i = -\frac{x_i}{\mathbf{w}^T\mathbf{x}} \quad \Leftrightarrow \quad y_i = 0, \qquad i \in \mathcal{I}.$$

Conversely, the components of $\mathbf{x}$ corresponding to null components of $\mathbf{v}$ are invariant under $\mathbf{T}$:

$$v_j = 0 \quad \Leftrightarrow \quad y_j = x_j, \qquad j \in \mathcal{J}.$$

These features have made the elementary matrix a powerful instrument of algorithm design and implementation. This report is a repertory of properties, formulas, and other information useful for the development of matrix software. Some of the material is new or hitherto unpublished.

2 Basics

We first derive some simple norms. Starting with the basic expressions

$$\mathbf{T}\mathbf{e}_j = \mathbf{e}_j + w_j\mathbf{v}, \qquad \mathbf{e}_i^T\mathbf{T} = \mathbf{e}_i^T + v_i\mathbf{w}^T,$$

we get

$$\|\mathbf{T}\|_1 = \max_{1 \leq j \leq n} \|\mathbf{e}_j + w_j\mathbf{v}\|_1, \qquad \|\mathbf{T}\|_\infty = \max_{1 \leq i \leq n} \|\mathbf{e}_i + v_i\mathbf{w}\|_1. \tag{2.1}$$

The inverse of an elementary matrix, if it exists, is another elementary matrix defined by

$$\mathbf{T}^{-1} = \mathbf{I} - \gamma^{-1}\mathbf{v}\mathbf{w}^T, \qquad \gamma = 1 + \mathbf{w}^T\mathbf{v}, \qquad \gamma \neq 0, \tag{2.2}$$

for which we have the norms

$$\|\mathbf{T}^{-1}\|_1 = \max_{1 \leq j \leq n} \|\mathbf{e}_j - \frac{w_j}{\gamma}\mathbf{v}\|_1, \qquad \|\mathbf{T}^{-1}\|_\infty = \max_{1 \leq i \leq n} \|\mathbf{e}_i - \frac{v_i}{\gamma}\mathbf{w}\|_1. \tag{2.3}$$

Condition numbers for the above norms are obtained from equations (2.1),(2.3), and the definition

$$\kappa_p(\mathbf{T}) = \|\mathbf{T}\|_p \|\mathbf{T}^{-1}\|_p.$$

A transformation is involutory if it coincides with its inverse. From equations (1.1) and (2.2), an elementary matrix is involutory if $\gamma = -1$, that is,

$$\mathbf{T} = \mathbf{T}^{-1} \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{v} = -2.$$

Using the property that the inverse and the transpose of an orthogonal matrix are equal, we derive the condition for an elementary matrix to be orthogonal,

$$\mathbf{T}^T = \mathbf{T}^{-1} \quad \Leftrightarrow \quad \mathbf{w} = -2\|\mathbf{v}\|^{-2}\mathbf{v},$$

where $\|.\|$ designates the $\ell_2(n)$ norm $\|.\|_2$. Orthogonal elementary matrices are Householder reflectors, which are symmetric and involutory:

$$\mathbf{H} = \mathbf{I} + \alpha\mathbf{u}\mathbf{u}^T, \qquad \alpha = -2\|\mathbf{u}\|^{-2}, \qquad \mathbf{u} \in \mathcal{R}^n.$$

The Householder transformation $\mathbf{H}_k$ that annihilates all components of a vector $\mathbf{x}$ other than x_k,

$$\mathbf{H}_k\mathbf{x} = \beta\mathbf{e}_k, \qquad |\beta| = \|\mathbf{x}\|,$$

is specified by

$$\mathbf{u} = \mathbf{x} - \beta\mathbf{e}_k, \qquad \alpha = \frac{1}{\beta(x_k - \beta)}, \qquad \beta = -\mathrm{sgn}(x_k)\|\mathbf{x}\|.$$

For implementation purposes, this matrix is often expressed in the following equivalent forms,

$$\left.\begin{aligned} \mathbf{H}_k &= \mathbf{I} + \mathbf{v}\mathbf{w}^T, \\ \mathbf{H}_k &= \mathbf{I} + \frac{1}{v_k}\mathbf{v}\mathbf{v}^T, \\ \mathbf{H}_k &= \mathbf{I} + v_k\mathbf{w}\mathbf{w}^T, \end{aligned}\right\} \quad \mathbf{v} = \frac{\mathbf{x}}{\beta} - \mathbf{e}_k, \qquad \mathbf{w} = \frac{1}{v_k}\mathbf{v},$$

where the vectors are normalized for trouble-free computation.

A transposition matrix is a particular reflector defined by

$$\mathbf{\Theta}_{k,m} = \mathbf{I} - (\mathbf{e}_k - \mathbf{e}_m)(\mathbf{e}_k - \mathbf{e}_m)^T,$$

which exchanges the k^{th} and m^{th} components of a vector.

Gauss (or Jordan) matrices are defined by the choice of a vector of the canonical basis for $\mathbf{w}$:

$$\mathbf{G}_k = \mathbf{I} + \mathbf{v}\mathbf{e}_k^T.$$

The most common use for a Gauss (or Jordan) matrix[1] is to annihilate all the components of a vector $\mathbf{x}$ other than x_k, $x_k \neq 0$:

$$\mathbf{G}_k\mathbf{x} = x_k\mathbf{e}_k, \qquad \mathbf{v} = \mathbf{e}_k - \frac{\mathbf{x}}{x_k}.$$

For that matrix, $\gamma = 1$, and the inverse satisfies

$$\mathbf{G}_k^{-1} = \mathbf{I} - \mathbf{v}\mathbf{e}_k^T, \qquad \text{that is,} \qquad \mathbf{G}_k + \mathbf{G}_k^{-1} = 2\mathbf{I}.$$

There is another type of Gauss matrix,

$$\tilde{\mathbf{G}}_k = \mathbf{I} + \tilde{\mathbf{v}}\mathbf{e}_k^T,$$

which also reduces $\mathbf{x}$ to a stretching of $\mathbf{e}_k$. It is involutory, and is defined by

$$\tilde{\mathbf{G}}_k\mathbf{x} = -x_k\mathbf{e}_k, \qquad \tilde{\mathbf{v}} = -\mathbf{e}_k - \frac{\mathbf{x}}{x_k}.$$

Note that, for a same vector $\mathbf{x}$, the vectors $\mathbf{v}$ and $\tilde{\mathbf{v}}$ of $\tilde{\mathbf{G}}_k$ and $\mathbf{G}_k$ differ only by their k^{th} components. While the effects on $\mathbf{x}$ of $\tilde{\mathbf{G}}_k$ and $\mathbf{G}_k$ are very much the same in most practical applications (the sign change in the result is seldom important), these two matrices have a fundamental difference that we shall discuss later. Although $\tilde{\mathbf{G}}_k$ is computationally equivalent to $\mathbf{G}_k$ for most algorithms, it is never used in practice.

From equations (2.1) and (2.3), we get the norms

$$\|\mathbf{G}_k\|_p = 1 + \|\mathbf{v} - v_k\mathbf{e}_k\|_p, \qquad p = 1, \infty.$$

For a same vector $\mathbf{x}$, the corresponding $\ell_1(n)$ and $\ell_\infty(n)$ norms of $\mathbf{G}_k$ and $\tilde{\mathbf{G}}_k$ take the same values. These matrices also have the same norms as their inverses.

The transformation defined by the pairing

$$\mathbf{G}_k\mathbf{\Theta}_{k,m}\mathbf{x} = x_m\mathbf{e}_k, \qquad |x_m| \geq |x_i|, \qquad i = 1, 2, \ldots, n,$$

is the basic operation of Gaussian elimination methods using partial pivoting. It is referred to as a stabilized elementary transformation [13], as the components of the vector $\mathbf{v}$ generated for $\mathbf{G}_k$ by $\Theta_{k,m}\mathbf{x}$ are bounded by unity. A stabilized elementary transformation can be similarly built from an involutory Gauss matrix, but it must be noted that the product $\tilde{\mathbf{G}}_k\mathbf{\Theta}_{k,m}$ is not necessarily involutory. In the following, we refer to a Gauss matrix as stabilized if its associated $\mathbf{v}$ vector is bounded by unity in the infinity norm.

[1] Some authors [7] distinguish between Gauss and Jordan elementary matrices as follows: (1) a Jordan transformation annihilates the elements of a vector *other than* a designated element, and (2) a Gauss transformation annihilates the elements of a vector *below* a specified element. For simplicity, noting that a Gauss transformation coincides with a Jordan transformation in a subspace, we do not make such a distinction, and we use the appellation "Gauss matrix" for either type.

3 Proper and singular elements

We first look at the proper vectors and values of $\mathbf{T}$. We note that the subspace $\mathcal{P}$ orthogonal to $\mathbf{w}$ is an invariant subspace of dimension $(n-1)$ for the proper value unity:

$$\mathbf{T}\mathbf{x} = \mathbf{x} \quad \Leftrightarrow \quad \mathbf{x} \in \mathcal{P}, \qquad \mathcal{P} \perp \mathbf{w}.$$

We also have

$$\mathbf{T}\mathbf{v} = \gamma\mathbf{v}.$$

If $\mathbf{v}$ is not in $\mathcal{P}$, $\mathbf{v}$ and $\mathcal{P}$ define a full set of proper vectors associated with the proper values γ and unity, the latter with multiplicity $n-1$. This case is exemplified by the matrices $\mathbf{H}_k$ and $\tilde{\mathbf{G}}_k$ of the previous section. If $\mathbf{v}$ is in $\mathcal{P}$, $\gamma = 1$, and $\mathbf{T}$ is defective: all the proper values are unity, and there are only $(n-1)$ proper vectors. The above Gauss matrix $\mathbf{G}_k$ is of this type.

We now turn to the determination of the singular values and the right singular vectors. A singular vector $\mathbf{x}$ and the associated singular value σ satisfy

$$\mathbf{T}^T\mathbf{T}\mathbf{x} = \sigma^2\mathbf{x}, \qquad \mathbf{T}^T\mathbf{T} = \mathbf{I} + \mathbf{v}\mathbf{w}^T + \mathbf{w}\mathbf{v}^T + \|\mathbf{v}\|^2\,\mathbf{w}\mathbf{w}^T, \tag{3.1}$$

where $\|.\|$ denotes $\|.\|_2$.

We first assume that $\mathbf{v}$ and $\mathbf{w}$ are not collinear. Let $\mathcal{S}$ be the subspace orthogonal to $\{\mathbf{v},\mathbf{w}\}$. $\mathcal{S}$ is a singular subspace associated with $(n-2)$ unit singular values, and any orthogonal basis of $\mathcal{S}$ forms a set of $(n-2)$ right singular vectors. Since neither of the two remaining right vectors can be parallel to $\mathbf{v}$ or $\mathbf{w}$, we must have the generic representation

$$\mathbf{x} = \frac{\mathbf{v}}{\|\mathbf{v}\|} + \mu\frac{\mathbf{w}}{\|\mathbf{w}\|}, \qquad \mu \neq 0, \tag{3.2}$$

to a scaling constant. Substituting this expression in equation (3.1) and separating the terms in $\mathbf{v}$ and $\mathbf{w}$, we get

$$\sigma^2 = \gamma + \mu\|\mathbf{v}\|\,\|\mathbf{w}\|, \tag{3.3}$$

$$\mu\sigma^2 = \gamma\mu + \gamma\|\mathbf{v}\|\,\|\mathbf{w}\| + \mu\|\mathbf{v}\|^2\|\mathbf{w}\|^2.$$

The elimination of σ^2 from the second of these equations yields

$$\mu^2 - \mu\|\mathbf{v}\|\,\|\mathbf{w}\| - \gamma = 0. \tag{3.4}$$

Using the Cauchy-Schwartz inequality, it is easy to show that the discriminant of this quadratic equation in μ is non-negative. The roots are

$$\mu_1 = \frac{1}{2}\left(\|\mathbf{v}\|\,\|\mathbf{w}\| + \sqrt{\|\mathbf{v}\|^2\|\mathbf{w}\|^2 + 4\gamma}\right), \qquad \mu_2 = -\frac{\gamma}{\mu_1},$$

which define the two right singular vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ by formula (3.2). The determination of the associated singular values immediately follows. From equations (3.4) and (3.3), we have

$$\sigma = |\mu|,$$

after taking a square root. Hence, the nonunit singular values

$$\sigma_1 = \frac{1}{2}\left(\|\mathbf{v}\|\,\|\mathbf{w}\| + \sqrt{\|\mathbf{v}\|^2\|\mathbf{w}\|^2 + 4\gamma}\right), \qquad \sigma_2 = \frac{|\gamma|}{\sigma_1},$$

are associated with the (unnormalized) right singular vectors

$$\mathbf{x}_1 = \frac{\mathbf{v}}{\|\mathbf{v}\|} + \sigma_1\frac{\mathbf{w}}{\|\mathbf{w}\|}, \qquad \mathbf{x}_2 = \frac{\mathbf{v}}{\|\mathbf{v}\|} - \mathrm{sgn}(\gamma)\sigma_2\frac{\mathbf{w}}{\|\mathbf{w}\|}.$$

Using the squares of the singular values and the Cauchy-Schwartz inequality, it can be shown that σ_1 and σ_2 are the extreme singular values:

$$\sigma_2 \le 1 \le \sigma_1, \qquad \sigma_2 \le |\gamma| \le \sigma_1.$$

The corresponding (unnormalized) left singular vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ are defined by the basic equation

$$\mathbf{T}\mathbf{x}_i = \sigma_i\mathbf{y}_i,$$

which yields

$$\mathbf{y}_1 = \sigma_1\frac{\mathbf{v}}{\|\mathbf{v}\|} + \frac{\mathbf{w}}{\|\mathbf{w}\|}, \qquad \mathbf{y}_2 = \sigma_2\frac{\mathbf{v}}{\|\mathbf{v}\|} - \mathrm{sgn}(\gamma)\frac{\mathbf{w}}{\|\mathbf{w}\|}.$$

The other left singular vectors coincide with their right homologues in the subspace $\mathcal{S}$ orthogonal to $\{\mathbf{v}, \mathbf{w}\}$. Since σ_1 and σ_2 are the extreme singular values, we have

$$\kappa_2(\mathbf{T}) = \frac{\sigma_1^2}{|\gamma|}, \quad \text{or} \quad \kappa_2(\mathbf{T}) = \frac{\gamma + \sigma_1\|\mathbf{v}\|\,\|\mathbf{w}\|}{|\gamma|}.$$

When $\mathbf{v}$ and $\mathbf{w}$ are collinear,

$$\mathbf{w} = \rho\mathbf{v}, \qquad \gamma = 1 + \rho\|\mathbf{v}\|^2,$$

and $\mathcal{S}$ reduces to the subspace orthogonal to $\mathbf{v}$. Any orthonormal basis of $\mathcal{S}$ is a set of $(n-1)$ right (and left) singular vectors for the singular value unity with corresponding multiplicity. The remaining singular value and vectors are

$$\sigma_1 = |\gamma|, \qquad \mathbf{x}_1 = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \qquad \mathbf{y}_1 = \mathrm{sgn}(\gamma)\mathbf{x}_1.$$

The Householder reflector is in that class of matrices, with $\gamma = -1$.

From the above, we get the nonunit singular values and condition number of the Gauss matrix $\mathbf{G}_k$ for $\mathbf{w} = \mathbf{e}_k$ and $\gamma = 1$:

$$\sigma_1 = \frac{1}{2}\left(\|\mathbf{v}\| + \sqrt{\|\mathbf{v}\|^2 + 4}\right), \qquad \sigma_2 = \frac{1}{\sigma_1}, \qquad \kappa_2(\mathbf{G}_k) = 1 + \sigma_1\|\mathbf{v}\|.$$

Identical results are obtained for the involutory Gauss matrix $\tilde{\mathbf{G}}_k$. From here, it is easy to show that the $\ell_2(n)$ condition number of a stabilized Gauss transformation is bounded by $(n+1)$.

4 The Bischof-Van Loan expressions of products

Most methods for the numerical solution of the standard problems of linear algebra rely on sequences of elementary transformations to reduce a matrix to some special form. In a serial implementation of these methods, r transformations cause some elements of the matrix to be fetched and stored r times. Since the processing times of memory references and arithmetic operations are about the same for high-performance computers, much effort has been devoted to reducing the occurrence of the former for better efficiency. One successful approach to that end consists of representing a sequence of r elementary transformations by a single operator, which, when applied to a matrix, generates substantially fewer than r memory references per matrix element. The construction of that operator may create some computational overhead that can be controlled by the value assigned to r. Algorithms using that device are the basis for the design of recent high-performance libraries [1]. They are rich of matrix operations, and are often referred to as block algorithms. The derivations below follow the work by Bischof and Van Loan (B-VL) on the representation of products of Householder matrices [2].

Consider the product of r elementary matrices of order m with $r < m$,

$$\mathbf{P}^{(r)} = \mathbf{T}_r\mathbf{T}_{r-1}\ldots\mathbf{T}_1, \qquad \mathbf{T}_k = \mathbf{I} + \mathbf{v}_k\mathbf{w}_k^T, \qquad \mathbf{v}_k, \mathbf{w}_k \in \mathcal{R}^m,$$

where $\mathbf{v}_i$ and $\mathbf{w}_i$ can be arbitrary. We show that any such product can be expressed in the form

$$\mathbf{P}^{(k)} = \mathbf{I} + \mathbf{X}^{(k)}\mathbf{W}^{(k)T}, \qquad \mathbf{X}^{(k)}, \mathbf{W}^{(k)} \in \mathcal{R}^{m\times k}, \qquad \mathbf{W}^{(k)} \equiv \sum_{i=1}^{k}\mathbf{w}_i. \quad (4.1)$$

Noting that this representation is obvious for $r = 1$ with $\mathbf{X}^{(1)} = \mathbf{v}_1\mathbf{e}_1^T$, let us assume that identity (4.1) is true, for which we have

$$\mathbf{P}^{(k+1)} = (\mathbf{I} + \mathbf{v}_{k+1}\mathbf{w}_{k+1}^T)\left(\mathbf{I} + \mathbf{X}^{(k)}\mathbf{W}^{(k)T}\right).$$

This last equation immediately yields the expression

$$\mathbf{X}^{(k+1)} = (\mathbf{I} + \mathbf{v}_{k+1}\mathbf{w}_{k+1}^T)\mathbf{X}^{(k)} + \mathbf{v}_{k+1}\mathbf{e}_{k+1}^T, \tag{4.2}$$

which verifies the representation

$$\mathbf{P}^{(k+1)} = \mathbf{I} + \mathbf{X}^{(k+1)}\mathbf{W}^{(k+1)T}.$$

Similarly, it can be proved that there exists a formula

$$\mathbf{P}^{(r)} = \mathbf{I} + \mathbf{V}^{(r)}\mathbf{Y}^{(r)T}, \qquad \mathbf{V}^{(k)}, \mathbf{Y}^{(k)} \in \mathcal{R}^{m\times k}, \qquad \mathbf{V}^{(k)} \equiv \sum_{i=1}^{k} \mathbf{v}_i,$$

for which $\mathbf{Y}^{(r)}$ is defined by the recurrence

$$\mathbf{Y}^{(1)} = \mathbf{w}_1\mathbf{e}_1^T, \qquad \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \left(\mathbf{I} + \mathbf{Y}^{(k)}\mathbf{V}^{(k)T}\right)\mathbf{w}_{k+1}\mathbf{e}_{k+1}^T.$$

This type of representation leads to block algorithms with the best time performance [5].

5 The Schreiber-Van Loan variants

The Schreiber-Van Loan (S-VL) representation is a modification of the B-VL expression of products that trades the $\mathbf{X}^{(r)}$ (or $\mathbf{Y}^{(r)}$) matrix for a triangular matrix of order r and thereby leads to implementations more economical of storage.

Starting from

$$\mathbf{x}_j^{(j)} = \mathbf{v}_j, \qquad j \le k,$$

the recurrence (4.2) generates the equations

$$\mathbf{x}_j^{(i)} = \mathbf{x}_j^{(i-1)} + \left(\mathbf{w}_{j+i}^T\mathbf{x}_j^{(i-1)}\right)\mathbf{v}_i, \qquad i = j+1,\ldots,k,$$

which, after summation with respect to i, yield

$$\mathbf{x}_j^{(k)} = \mathbf{v}_j + \left(\mathbf{w}_{j+1}^T\mathbf{x}_j^{(j)}\right)\mathbf{v}_{j+1} + \ldots + \left(\mathbf{w}_k^T\mathbf{x}_j^{(k-1)}\right)\mathbf{v}_k.$$

This expression reveals $\mathbf{x}_j^{(k)}$ as a linear combination of $\{\mathbf{v}_j, \mathbf{v}_{j+1}, \ldots, \mathbf{v}_k\}$, and implies the existence of a lower-triangular matrix $\mathbf{\Gamma}^{(k)}$ with unit diagonal such that

$$\mathbf{X}^{(k)} = \mathbf{V}^{(k)}\mathbf{\Gamma}^{(k)}, \qquad \mathbf{\Gamma}^{(k)} \in \mathcal{R}^{k\times k}.$$

$\mathbf{\Gamma}^{(k)}$ is the "middle matrix" of the Schreiber-Van Loan representation [11]

$$\mathbf{P}^{(k)} = \mathbf{I} + \mathbf{V}^{(k)}\mathbf{\Gamma}^{(k)}\mathbf{W}^{(k)T},$$

which is more economical of storage than the alternatives of the previous section for the case of Householder matrices. The identity

$$\mathbf{I} + \mathbf{V}^{(j)}\mathbf{\Gamma}^{(j)}\mathbf{W}^{(j)T} = (\mathbf{I} + \mathbf{v}_j\mathbf{w}_j^T)\left(\mathbf{I} + \mathbf{V}^{(j-1)}\mathbf{\Gamma}^{(j-1)}\mathbf{W}^{(j-1)T}\right)$$

readily generates the recurrence

$$\mathbf{\Gamma}^{(j)} = \mathbf{\Gamma}^{(j-1)} + \mathbf{e}_j\left(\mathbf{w}_j^T\mathbf{V}^{(j-1)}\mathbf{\Gamma}^{(j-1)} + \mathbf{e}_j^T\right), \tag{5.1}$$

which can be used to build $\mathbf{\Gamma}^{(k)}$ row by row, starting from $\mathbf{\Gamma}^{(1)} = \mathbf{e}_1\mathbf{e}_1^T$. This is the approach taken in [11] for the case of Householder matrices. Note that equation (5.1) is equivalent to

$$\mathbf{\Gamma}^{(j)} = \mathbf{\Gamma}^{(j-1)} + \mathbf{e}_j\left(\mathbf{w}_j^T\mathbf{X}^{(j-1)} + \mathbf{e}_j^T\right),$$

where $\mathbf{X}^{(j-1)}$ is the matrix of the B-VL representation. This matrix, however, does not have to be computed.

Dropping superscripts for simplicity, let

$$\mathbf{P} = \mathbf{I} + \mathbf{V}\mathbf{\Gamma}\mathbf{W}^T, \qquad \mathbf{V},\mathbf{W} \in \mathcal{R}^{m\times r}, \qquad \mathbf{\Gamma} \in \mathcal{R}^{r\times r},$$

represent the product of r elementary matrices. For $k \leq r$, any matrix $\mathbf{\Gamma}^{(k)}$ of the previous section is a leading principal submatrix of $\mathbf{\Gamma}$, while $\mathbf{V}^{(k)}$ and $\mathbf{W}^{(k)}$ derive from the restriction of $\mathbf{V}$ and $\mathbf{W}$ to their leading k columns. Left multiplication of equation (5.1) by $\mathbf{e}_j^T$ yields

$$\mathbf{e}_j^T\mathbf{\Gamma}^{(j)} = \mathbf{e}_j^T + \mathbf{w}_j^T\mathbf{V}^{(j-1)}\mathbf{\Gamma}^{(j-1)} \qquad j = 1,2,\ldots,r, \tag{5.2}$$

that is, the j^{th} row of $\mathbf{\Gamma}$. In that equation, the term crucial to closed-form expression is the vector

$$\mathbf{w}_j^T\mathbf{V}^{(j-1)},$$

for which we introduce the additive triangular decomposition

$$\mathbf{W}^T\mathbf{V} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{U}}, \qquad \ell_{ij} = \mathbf{w}_i^T\mathbf{v}_j, \;\; i > j, \qquad \ell_{ij} = 0, \;\; i \leq j. \tag{5.3}$$

With this definition, we have

$$\mathbf{w}_j^T\mathbf{V}^{(j-1)} = \mathbf{e}_j^T\boldsymbol{\mathcal{L}},$$

for which equation (5.2) generates the matrix expression

$$\mathbf{\Gamma} = \mathbf{I} + \boldsymbol{\mathcal{L}}\mathbf{\Gamma}$$

and the following closed form[2] of the middle matrix:

$$\mathbf{\Gamma} = (\mathbf{I} - \boldsymbol{\mathcal{L}})^{-1}. \tag{5.4}$$

Still making no particular assumption about $\mathbf{V}$ and $\mathbf{W}$ other than $\mathbf{I} + \boldsymbol{\mathcal{U}}$ be not singular, we now derive the inverse of

$$\mathbf{P} = \mathbf{I} + \mathbf{V}(\mathbf{I} - \boldsymbol{\mathcal{L}})^{-1}\mathbf{W}^T. \tag{5.5}$$

The Sherman-Morrison formula applied to $\mathbf{P}$ in its original form yields

$$(\mathbf{I} + \mathbf{V}\mathbf{\Gamma}\mathbf{W}^T)^{-1} = \mathbf{I} - \mathbf{V}(\mathbf{\Gamma}^{-1} + \mathbf{W}^T\mathbf{V})^{-1}\mathbf{W}^T,$$

in which we use the decomposition (5.3) to get

$$\mathbf{P}^{-1} = \mathbf{I} - \mathbf{V}(\mathbf{I} + \boldsymbol{\mathcal{U}})^{-1}\mathbf{W}^T, \tag{5.6}$$

an expression similar in form to that of equation (5.5).

We now briefly consider the particular case of Householder matrices,

$$\mathbf{w}_j = -\frac{2}{\|\mathbf{v}_j\|^2}\mathbf{v}_j, \qquad j = 1, 2, \ldots, r,$$

which we recast in the form

$$\mathbf{W} = \mathbf{V}\mathbf{D}^{-1}, \qquad d_{ij} = 0, \qquad i \neq j, \qquad d_{ii} = -\frac{\|\mathbf{v}_i\|^2}{2}.$$

The representation of the product reduces to

$$\mathbf{P} = \mathbf{I} + \mathbf{V}\mathbf{\Delta}^{-1}\mathbf{V}^T, \qquad \mathbf{\Delta} = \mathbf{D}\mathbf{\Gamma}^{-1}, \tag{5.7}$$

where $\mathbf{\Delta}$ is lower-triangular. Using the orthogonality property of $\mathbf{P}$ and the expression (5.6) of the inverse, we get the relation

$$\mathbf{\Delta} + \mathbf{\Delta}^T = -\mathbf{V}^T\mathbf{V}, \tag{5.8}$$

which provides the algorithm for the computation of $\mathbf{\Delta}$ from $\mathbf{V}$ alone.

The representation of the middle matrix by its inverse $\mathbf{I} - \boldsymbol{\mathcal{L}}$ has interesting computational implications that we explore below. The construction of $\mathbf{\Gamma}$ with the recurrence (5.1) requires about

$$f_1 \approx mr^2 + \frac{r^3}{3}$$

[2]This expression was independently developed by Puglisi [10] for Householder matrices.

floating-point operations, while the computation of $\mathbf{\Gamma}^{-1}$ with formula (5.4) takes

$$f_2 \approx mr^2$$

operations. The difference in those numbers reflects the fact that the recurrence (5.1) actually combines the computation of $\mathbf{\Gamma}^{-1}$ and its inversion, as

$$f_1 - f_2 \approx \frac{r^3}{3}$$

is about the number of operations required for the inversion of a triangular matrix of order r. Under the reasonable assumption that multiplication by a lower-triangular matrix requires the same amount of work and machine time as the multiplication by the inverse (a simple forward substitution), the implementation of block methods based on the S-VL representation should benefit from using $\mathbf{\Gamma}^{-1}$ instead of $\mathbf{\Gamma}$. At each block step of the QR factorization of a matrix of order n with blocking parameter r, $r^3/3$ operations can be avoided, for an approximate total saving of $nr^2/3$ floating-point operations. What is more important is that the construction of $\boldsymbol{\mathcal{L}}$ can be carried out with some level-3 BLAS for the rank-r update of a symmetric matrix.

6 Remarks on implementation

Implementations of elementary transformations (and products thereof) for high-performance machines are far from being uniquely defined, even with the use of tuned BLAS [9, 3, 4]. In this section, we briefly review some of the possible choices with a definite bias for fast scalar architectures with hierarchical storage (including a cache). The focus is on Householder matrices, which are computationally more complex than their Gauss relatives. To limit the scope of the discussion, we exclude the cases where the transformations have small dimensions (e.g., plane reflections, or transformations of dimension three of the LR or QR algorithms for real Hessenberg matrices). Likewise, the assumption that the fast memory (proximal cache) can accomodate more than just a few columns of the matrix being transformed eliminates the case of very large dimensions. Questions of stride of reference to the matrix elements assume a FORTRAN organization of two-dimensional arrays in which consecutive elements of a column are stored in contiguous memory cells. Parentheses in equations below indicate computational blocks and order of operation.

We start with the left multiplication of a matrix $\mathbf{A} \in \mathcal{R}^{n\times n}$ by an elementary matrix,

$$\left(\mathbf{I} + \mathbf{v}\mathbf{w}^T\right)\mathbf{A} = \mathbf{A} + \mathbf{v}\left(\mathbf{A}^T\mathbf{w}\right)^T, \tag{6.1}$$

which, in this form, can be considered as a rank-1 update of $\mathbf{A}$. A natural way to organize the computation is based on a partition of the matrix in strips of s columns (perhaps fewer for the last strip) such that a strip can be contained in the cache. As soon as the segment of $\mathbf{w}^T\mathbf{A}$ corresponding to a strip is computed, it can be used for the update of the strip, which at that point may be overwritten in storage (*in situ* transformation). Computing and keeping the segment in s registers (the accumulators of the scalar products) for immediate use in the matrix update minimizes storage references and efficiently exploits the data present in the fast memory. The determination of the optimum value of s usually requires further information on machine architecture. Note that a finer performance analysis of this computation is likely to require some additional partitioning of the matrix for the efficient calculation of each segment of $\mathbf{w}^T\mathbf{A}$ (this remark in fact applies to all the transformations considered in this section). We shall not discuss this level of detail, which is usually handled by some efficient BLAS – or in-line substitute code.

Using a similar approach, and assuming that the cache can hold s rows of $\mathbf{A}$, we express the right multiplication by

$$\mathbf{A}\left(\mathbf{I}+\mathbf{w}\mathbf{v}^T\right) = \mathbf{A} + (\mathbf{A}\mathbf{w})\mathbf{v}^T,$$

which suggests for $\mathbf{A}$ a partitioning in strips of s rows, with a corresponding computation of $\mathbf{A}\mathbf{w}$ segment by segment. As each segment becomes available, it can be used for the update of the associated matrix strip.

The elementary orthogonal similarity transformation of a matrix,

$$\mathbf{C} = \left(\mathbf{I}+\mathbf{v}\mathbf{w}^T\right)\mathbf{A}\left(\mathbf{I}+\mathbf{w}\mathbf{v}^T\right),$$

naturally lends itself to a wider variety of algorithms. The first consists of a sequence of two one-sided transformations,

$$\mathbf{C} = \mathbf{B} + \mathbf{v}\left(\mathbf{B}^T\mathbf{w}\right)^T, \qquad \mathbf{B} = \mathbf{A} + (\mathbf{A}\mathbf{w})\mathbf{v}^T, \tag{6.2}$$

which is entirely executed by level-2 (matrix-vector) operations. Another algorithm expresses the transformation as two combined matrix updates of rank one:

$$\mathbf{C} = \mathbf{A} + \mathbf{v}\mathbf{f}^T + \mathbf{g}\mathbf{v}^T,$$

$$\mathbf{f} = \mathbf{A}^T\mathbf{w} + \tfrac{1}{2}\left(\mathbf{w}^T\mathbf{A}\mathbf{w}\right)\mathbf{v}, \qquad \mathbf{g} = \mathbf{A}\mathbf{w} + \tfrac{1}{2}\left(\mathbf{w}^T\mathbf{A}\mathbf{w}\right)\mathbf{v}.$$

In general, this scheme is not as efficient as the one prescribed by equation (6.2), as the construction of $\mathbf{f}$ and $\mathbf{g}$ requires level-1 (vector) operations,

namely, one scalar product and two elementary linear combinations. When $\mathbf{A}$ is symmetric, however, the equality of $\mathbf{f}$ and $\mathbf{g}$ reduces complexity, and symmetry can be preserved throughout the computation.

While the use of BLAS in the implementation of these operations is beneficial, it prevents the kind of optimization outlined for the left transformation (6.1), where the registers containing partial results of $\mathbf{Aw}$ are immediately used for the update of a strip of matrix. At best, for re-use of the cache contents (a strip of matrix), stripping can be made an explicit part of the program performing the operation, and the BLAS for matrix-vector multiplication and matrix update can be called in sequence for each strip. We do not consider this approach to be satisfactory, and not because of the minor performance loss due to nonoptimal use of registers. What is more serious is the burden placed on the user for some optimization that can be better achieved with tuned subprograms implementing Householder transformations. Such software, which is included in LAPACK in the form of obscure auxiliary routines (the _LARF_ set), fully deserves inclusion in the BLAS.

The above comments carry over to products of Householder transformations, for which best performance is achieved with B-VL representations. This level of efficiency is matched by the S-VL representation for one-sided transformations only if modified to use the inverse of the middle matrix (5.7) and the construction (5.8) . For similarity transformations, the B-VL implementation is still slightly faster. The construction of its $\mathbf{X}^{(r)}$ or $\mathbf{Y}^{(r)}$ matrices (Section 4) is best performed by using the inverse $\mathbf{I} - \boldsymbol{\mathcal{L}}$ or $\boldsymbol{\Delta}^{-1}$ of the S-VL middle matrix and appropriate level-3 BLAS. In the following, we outline sample implementations of block methods that illustrate these points (the operations are expressed for execution by level-3 BLAS).

Let $\mathbf{V}$ be the matrix in $\mathcal{R}^{n \times r}$ whose columns are vectors associated with a sequence of Householder transformations. Using formula (5.8), $\boldsymbol{\Delta}$ is constructed by forming the lower-triangular part of $-\mathbf{V}^T\mathbf{V}$ and multiplying its diagonal by one half. These two matrices define the matrix of the product of the transformations:

$$\mathbf{P} = \mathbf{I} + \mathbf{V}\boldsymbol{\Delta}^{-1}\mathbf{V}^T,$$

which we apply to a matrix $\mathbf{A}$. We first consider the case of a left transformation *in situ*,

$$\mathbf{A} := \mathbf{PA},$$

for which the S-VL implementation requires an ancillary array of $(r \times n)$

cells for the intermediate matrices represented by $\mathbf{Z}$:

$$\mathbf{Z} := \mathbf{A}^T\mathbf{V},$$

$$\mathbf{Z} := \mathbf{Z}\boldsymbol{\Delta}^{-T}, \qquad \text{(forward substitution)}$$

$$\mathbf{A} := \mathbf{A} + \mathbf{V}\mathbf{Z}^T.$$

Two arrays of $(r \times n)$ cells are needed for the intermediate matrices $\mathbf{Y}$ and $\mathbf{Z}$ in the B-VL implementation below:

$$\mathbf{Y} := \mathbf{V}\boldsymbol{\Delta}^{-T}, \qquad \text{(forward substitution)}$$

$$\mathbf{Z} := \mathbf{A}^T\mathbf{Y},$$

$$\mathbf{A} := \mathbf{A} + \mathbf{V}\mathbf{Z}^T.$$

It is clear that the additional expense of storage for $\mathbf{Y}$ matrix of the the B-VL representation cannot be justified, since the two schemes differ only by the order of operation. The same comment applies to right-hand transformations. Hence, the S-VL implementation should be preferred for one-sided transformations.

The case of a similarity transformation

$$\mathbf{A} := \mathbf{P}\mathbf{A}\mathbf{P}^T,$$

is somewhat different. Its S-VL implementation is represented by:

$$\mathbf{Z} := \mathbf{A}\mathbf{V},$$

$$\mathbf{Z} := \mathbf{Z}\boldsymbol{\Delta}^{-T}, \qquad \text{(forward substitution)}$$

$$\mathbf{A} := \mathbf{A} + \mathbf{Z}\mathbf{V}^T,$$

$$\mathbf{Z} := \mathbf{A}^T\mathbf{V},$$

$$\mathbf{Z} := \mathbf{Z}\boldsymbol{\Delta}^{-T}, \qquad \text{(forward substitution)}$$

$$\mathbf{A} := \mathbf{A} + \mathbf{V}\mathbf{Z}^T.$$

Note that this procedure uses only one ancillary array and performs two forward substitutions. The B-VL approach uses two arrays and performs

only one forward substitution:

$$\begin{aligned}
&\mathbf{Z} := \mathbf{AV}, \\
&\mathbf{Y} := \mathbf{V}\boldsymbol{\Delta}^{-1}, \qquad \text{(forward substitution)} \\
&\mathbf{A} := \mathbf{A} + \mathbf{ZY}^T, \\
&\mathbf{Z} := \mathbf{A}^T\mathbf{V}, \\
&\mathbf{A} := \mathbf{A} + \mathbf{YZ}^T.
\end{aligned}$$

The gain in performance is very slight if r is a small fraction of n, a condition usually satisfied in practice. This consideration should make the S-VL scheme based on the inverse of the middle matrix the preferred design for library software[3].

7 Transformations in two dimensions

Transformations in two dimensions commonly appear in larger computations such as the solutions of standard and generalized eigenvalue problems, from which the instances discussed below are borrowed.

We first consider orthogonal transformations. In most applications, these are practically interchangeable with Givens rotations [7] represented by the matrix

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \qquad c^2 + s^2 = 1,$$

which is not of the Householder type (it is the product of a Householder transformation and a transposition). While reflectors in two dimensions can be used in the general form of Section 2

$$\mathbf{H} = \mathbf{I} - 2\mathbf{uu}^T, \qquad \|\mathbf{u}\| = 1,$$

they can also be represented by

$$\mathbf{H} = \begin{bmatrix} c & s \\ s & -c \end{bmatrix}, \qquad c^2 + s^2 = 1. \tag{7.1}$$

The equivalence of the two representations follows from letting u_1 and u_2 be the sine and cosine of an arc, which defines s and c of equations (7.1) as the sine and the cosine of the double arc. A matrix (7.1) is usually referred to as a plane reflector or a Givens reflector.

[3]The current S-VL implementations in LAPACK do not use the inverse of the middle matrix, more likely by an accident of timing rather than by design.

The simplest use of a Givens reflector is found in the annihilation of a component of a vector, as illustrated by the transformation

$$\mathbf{Hx} = \|\mathbf{x}\|\mathbf{e}_1,$$

which, as for a plane rotation, defines

$$c = \frac{x_1}{\|\mathbf{x}\|}, \qquad s = \frac{x_2}{\|\mathbf{x}\|}.$$

A more complicated problem is the similarity reduction of a matrix $\mathbf{A}$ of order two to some special form $\mathbf{B}$:

$$\begin{bmatrix} c & s \\ s & -c \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} c & s \\ s & -c \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}. \tag{7.2}$$

Assuming that $c \neq 0$ and letting

$$d = a_{11} - a_{22}, \qquad e = a_{21} + a_{12}, \qquad c = \frac{1}{\sqrt{1+t^2}}, \qquad s = tc,$$

we transform equation (7.2) into the equivalent form

$$\mathbf{B} = \frac{1}{1+t^2} \begin{bmatrix} a_{22}t^2 + et + a_{11} & a_{21}t^2 + dt - a_{12} \\ a_{12}t^2 + dt - a_{21} & a_{11}t^2 - et + a_{22} \end{bmatrix}. \tag{7.3}$$

This expression is the basis for the derivation of special transformations, including the reduction of $\mathbf{A}$ to standard upper Schur form, which we consider now. The standard Schur form $\mathbf{B}$ of $\mathbf{A}$ is defined as follows: if $\mathbf{A}$ has real eigenvalues, $\mathbf{B}$ is upper-triangular, while if $\mathbf{A}$ has complex eigenvalues, b_{11} and b_{22} are equal and coincide with the common real part of the eigenvalues. Assuming that $a_{21} \neq 0$, we first handle the simple case $a_{12} = 0$, for which the assignments $c = 0$ and $s = 1$ produce the desired result through an exchange of the rows and columns of $\mathbf{A}$. In the general case, the condition $b_{21} = 0$ and equation (7.3) lead to

$$a_{12}\,t^2 + d\,t - a_{21} = 0, \tag{7.4}$$

This equation has real roots if

$$\delta \geq 0, \qquad \delta = d^2 + 4a_{21}a_{12}.$$

As expected, this is the condition for which $\mathbf{A}$ has real eigenvalues. It is easy to check that the simple change of variable

$$\lambda = a_{22} - a_{12}\,t.$$

tranforms equation (7.4) into **A**'s characteristic equation. Instead of solving equation (7.4) to determine the reflection that triangularizes **A** (when it exists), it is numerically better to perform (1) a similarity reflection that produces a matrix with equal diagonal elements, followed by (2) a similarity reflection that triangularizes that matrix when the eigenvalues are real. This technique is used by the LAPACK auxiliary routine _LANV2 for a reduction to Schur form by plane rotations. Some of the reasons for the preferablility of this approach are easily seen in equation (7.3). In the following, we treat the two phases of the above computation as two separate problems.

From equation (7.3), the reflection for which $b_{11} = b_{22}$ is defined by

$$d\,t^2 - 2e\,t - d = 0 \qquad d \neq 0.$$

This equation has real roots since its discriminant Δ is always positive,

$$\Delta = d^2 + e^2, \qquad \Delta \geq 0.$$

Choosing the root of smaller magnitude to maximize c,

$$t = -\mathrm{sgn}(e)\frac{d}{|e| + \sqrt{\Delta}},$$

we obtain the formulas

$$c = \frac{1}{\sqrt{2}}\left(1 + \frac{|e|}{\sqrt{\Delta}}\right)^{1/2}, \qquad s = -\frac{1}{2}\mathrm{sgn}(e)\frac{d}{c\sqrt{\Delta}},$$

which prescribe safe computations for the solution of our first problem.

For the second problem, we use again equation (7.3) with the assumptions

$$a_{11} = a_{22}, \qquad a_{21}a_{12} > 0,$$

for which the triangularization formulas reduce to

$$a_{12}t^2 - a_{21} = 0, \qquad c = \sqrt{\frac{a_{12}}{e}}, \qquad s = \sqrt{\frac{a_{21}}{e}}, \qquad e = a_{21} + a_{12}.$$

In principle, the problem of similarity triangularization can also be solved with the use of Gauss transformations[4], albeit not as satisfactorily from a viewpoint of numerical stability. We briefly look at such an approach in the remaining part of this section.

[4]Such a triangularization is no longer a reduction to Schur form since Gauss transformations are not orthogonal.

To parallel the above discussion, we consider the similarity transformation of **A** by an involutory Gauss matrix,

$$\begin{bmatrix} 1 & 0 \\ g & -1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ g & -1 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \tag{7.5}$$

which leads to the expression

$$\mathbf{B} = \begin{bmatrix} a_{12}g + a_{11} & -a_{12} \\ a_{12}g^2 + dg - a_{21} & -a_{12}g + a_{22} \end{bmatrix}, \qquad d = a_{11} - a_{22}. \tag{7.6}$$

Predictably, the triangularization of **A** is predicated on the existence of a real solution g of the same equation as in the orthogonal case (7.4):

$$a_{12}\, g^2 + d\, g - a_{21} = 0,$$

If real roots exist, the root of smaller magnitude should be chosen to minimize the condition of the elementary matrix (see Section 3):

$$g = \mathrm{sgn}(d)\frac{2a_{21}}{|d| + \sqrt{\delta}}, \qquad \delta = d^2 + 4a_{21}a_{12}, \qquad \delta \geq 0.$$

In addition, the elementary matrix will be stabilized under the following condition,

$$|a_{21}| \leq |a_{12}| \quad \Rightarrow \quad |g| \leq 1,$$

which can be proved using the inequality $\delta \geq 0$.

When the eigenvalues of **A** are complex, the specification $b_{11} = b_{22}$ is realized with

$$g = -\frac{d}{2a_{12}}, \qquad \delta < 0, \qquad |a_{21}| \leq |a_{12}| \quad \Rightarrow \quad |g| \leq 1.$$

Finally, triangularization in the case where $a_{11} = a_{22}$ leads to

$$g = \sqrt{\frac{a_{21}}{a_{12}}}, \qquad a_{21}a_{12} > 0,$$

to which the usual stability condition applies. To satisfy this condition, a similarity transposition may be needed first to bring the off-diagonal element of smaller absolute value into sub-diagonal position. Preferably, this effect is achieved in practice by explicitly exchanging the columns of the Gauss matrix for the left transformation in equation (7.5), and by exchanging its rows for the right transformation (the scrambled matrices are no longer involutory). The similarity transformation (7.6) then becomes

$$\mathbf{B} = \begin{bmatrix} 0 & 1 \\ -1 & g \end{bmatrix} \mathbf{A} \begin{bmatrix} g & -1 \\ 1 & 0 \end{bmatrix},$$

which generates an alternate set of formulas to be used when exchanges are required:

$$\mathbf{B} = \begin{bmatrix} a_{21}g + a_{22} & -a_{21} \\ a_{21}g^2 - dg - a_{12} & -a_{21}g + a_{11} \end{bmatrix}, \qquad d = a_{11} - a_{22}.$$

8 Conclusion

General elementary matrices were found useful for the construction of test data, which can be made less costly of computer time with the use of block formulations. The modification described in this report for the Schreiber-Van Loan representation has proved to be efficient and easy to implement in test and production programs. Its combination with Stewart's method [12] for the generation of random orthogonal transformations is strongly recommended.

Subprograms for the implementation of Householder and Schreiber-Van Loan transformations would be a welcome addition to the BLAS. The main advantages over implementations explicitly based on BLAS include opportunities for superior tuning and cleaner coding. This addition would also be consistent with the presence of four routines for plane rotations in the set of level-1 BLAS.

Section 7 on elementary transformations in two dimensions was motivated by the reduction of a matrix of order two to standard Schur form used in LAPACK, which represents a sophisticated use of plane reflections (or rotations) for the solution of a problem that is more difficult than it may seem.

References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia PA, 1992.

[2] C. Bischof and C. Van Loan, *The WY representation for products of Householder matrices*, SIAM J. Sci. Stat. Comp., 8:s2-s13, 1987.

[3] J. Dongarra, J. DuCroz, S. Hammarling,, and R. Hanson, *An extended set of FORTRAN basic linear-algebra subprograms*, ACM Trans. Math. Soft., 14:1-17 and 18-32, 1988.

[4] J. DONGARRA, J. DUCROZ, S. HAMMARLING,, AND I. DUFF, *A set of level-3 basic linear-algebra subprograms*, ACM Trans. Math. Soft., 16:1-17 and 18-28, 1988.

[5] A. DUBRULLE, *On block Householder algorithms for the reduction of a matrix to Hessenberg form*, in Supercomputing'88: Vol. II, Science and Applications, Martin and Lundstrom eds., IEEE Computer Society Press, Washington DC, 1989.

[6] A. DUBRULLE, *On FORTRAN matrix software and vector computing*, Proc. Third IMSL Users North America Conf., Monterey CA, 1990.

[7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore MD, 1989.

[8] A. S. HOUSEHOLDER, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York NY, 1964.

[9] C. LAWSON, R. HANSON, R. KINCAID, AND F. KROGH, *Basic linear-algebra subprograms for FORTRAN usage*, ACM Trans. Math. Soft., 5:308-323, 1979.

[10] C. PUGLISI, *Modification of the Householder method based on the compact WY representation*, SIAM J. Sci. Stat. Comp., 13:723-726, 1992.

[11] R. SCHREIBER AND C. VAN LOAN, *A storage-efficient WY representation for products of Householder transformations*, SIAM J. Sci. Stat. Comp., 10:53-57, 1989.

[12] G. W. STEWART, *The efficient generation of orthogonal matrices with an application to condition estimators*, SIAM J. Num. Anal., 17:403-409, 1980.

[13] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.