To appear in Proceedings of Workshop on Applied Parallel Computing in Physics, Chemistry and Engineering (PARA95), Lyngby, Denmark, August 22–24, 1995, Lecture Notes in Computer Science, Springer-Verlag.

# Reduction of a Regular Matrix Pair (A, B) to Block Hessenberg-Triangular Form

Krister Dackland and Bo Kågström

Department of Computing Science, University of Umeå, S–901 87 Umeå, Sweden.
Email addresses: dacke@cs.umu.se and bokg@cs.umu.se.

**Abstract.** An algorithm for reduction of a regular matrix pair $(A, B)$ to block Hessenberg-triangular form is presented. This condensed form $Q^T(A, B)Z = (H, T)$, where $H$ and $T$ are block upper Hessenberg and upper triangular, respectively, and $Q$ and $Z$ orthogonal, may serve as a first step in the solution of the generalized eigenvalue problem $Ax = \lambda Bx$. It is shown how an elementwise algorithm can be reorganized in terms of blocked factorizations and higher level BLAS operations. Several ways to annihilate elements are compared. Specifically, the use of Givens rotations, Householder transformations, and combinations of the two. Performance results of the different variants are presented and compared to the LAPACK implementation DGGHRD, which indeed is unblocked.

## 1 Introduction

A reliable way to solve the generalized eigenvalue problem $Ax = \lambda Bx$, where $A$ and $B$ are square real $n \times n$ matrices, is to reduce $(A, B)$ to generalized Schur ($GS$) form $(S, T)$, where $S$ is upper quasi-triangular and $T$ is upper triangular. A quasi-triangular matrix is block upper triangular with $1 \times 1$ and $2 \times 2$ blocks on the diagonal. The $2 \times 2$ blocks on the diagonal of $S - \lambda T$ correspond to pairs of complex conjugate eigenvalues. The real eigenvalues are given by $(s_{ii}, t_{ii}) \neq (0, 0)$ (i.e., the $1 \times 1$ diagonal blocks of $(S, T)$). The finite eigenvalues are $s_{ii}/t_{ii}$, where $t_{ii} \neq 0$. If $(s_{ii}, t_{ii}) \neq (0, 0)$ for all $i$, then $(A, B)$ is a regular matrix pair.

Typically, the $GS$ form is computed in two steps. First, $(A, B)$ is reduced to condensed form $(H, T)$ by using orthogonal transformations such that $H$ is upper Hessenberg and $T$ is upper triangular (e.g. see [5]). In the second step the $QZ$ algorithm [9] is iteratively applied to the condensed pencil $(H, T)$ to obtain the desired $GS$ form $(S, T)$.

One way to achieve high performance on modern advanced architectures is to restruct well-known elementwise algorithms in terms of level 2 (matrix-vector) and level 3 (matrix-matrix) operations. These blocked algorithms permit reuse of data in the higher levels of a memory hierarchy, which make them run closer to peak machine speed.

In this contribution we focus on the first step in the process of solving the generalized eigenvalue problem. An implementation of an elementwise algorithm for the Hessenberg-triangular reduction (called DGGHRD for double precision real data) is included in LAPACK [1]. We show how the elementwise algorithm can be reorganized in terms of blocked factorizations and higher level (matrix-matrix) operations to reduce $A$ to upper *block* Hessenberg form and $B$ to upper triangular form. In order to make the algorithm portable and efficient most operations are expressed in terms of BLAS [8], [4], [6], [7] and LAPACK.

Several ways to annihilate elements are evaluated. We compare the use of Givens rotations, Householder transformations, and combinations of the two. Performance results of the different variants are presented and compared to the unblocked LAPACK implementation DGGHRD.

The rest of the paper is organized as follows. Section 2 outlines the elementwise and level 3 (blocked) algorithms. In Section 3 different variants of the blocked algorithm are presented. Section 4 presents some measured performance results of the implementations for an IBM RS6000 workstation. Finally, in Section 5 we give a brief summary and outline future work.

## 2 Reduction of $(A, B)$ to Hessenberg-Triangular Form

We present algorithms for reducing a regular matrix pair $(A, B)$ to condensed form in terms of an orthogonal equivalence transformation such that

$$Q^T A Z = H, \quad Q^T B Z = T,$$

where $H$ is block upper Hessenberg, $T$ is upper triangular and $Q$ and $Z$ are orthogonal. First, we review the elementwise algorithm that reduces $A$ to upper Hessenberg form. The second algorithm is based on level 3 operations and reduces $A$ to block upper Hessenberg form. In both cases the final form of $B$ is upper triangular.

### 2.1 Elementwise Reduction to Hessenberg-Triangular Form

The elementwise algorithm reduces a regular matrix pair $(A, B)$ to upper Hessenberg-triangular form [5]. In the first step of the reduction an orthogonal matrix $U$ is determined such that $U^T B$ is upper triangular. To preserve the generalized eigenvalues of the matrix pair $U$ is also applied to $A$: $A \leftarrow U^T A$. Now $A$ is dense and $B$ is triangular. To further reduce $A$ to upper Hessenberg form while preserving $B$ upper triangular, Givens rotations or $2 \times 2$ Householder reflections are used. Orthogonal transformations applied from left annihilate elements in $A$ but destroy the structure of $B$. The triangular form of $B$ is restored by right hand side orthogonal transformations. We illustrate one step of the reduction using Givens rotations ($A$ and $B$ are of size $4 \times 4$):

$$Q_{34}^T A = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot \end{bmatrix}, \quad Q_{34}^T B = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot \\ & & \cdot & \cdot \\ & & \star & \cdot \end{bmatrix},$$

$$
BZ_{34} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot \\ & & \cdot & \cdot \\ & & & \cdot \end{bmatrix}, \quad AZ_{34} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot \end{bmatrix}
$$

The rotation $Q_{34}$ eliminates $a_{41}$ and when applied to $B$ introduces a non-zero element $b_{43}$. This fill-in is set to zero by a right rotation $Z_{34}$. Since $Z_{34}$ only operates on columns 3 and 4, no new non-zero elements are introduced in $A$. To annihilate the remaining elements below the sub-diagonal similar two-sided rotations are used.

### 2.2 Level 3 Reduction to Block Hessenberg-Triangular Form

We reorganize the elementwise algorithm to a blocked variant that mainly perform level 3 (matrix-matrix) operations and some level 2 (matrix-vector) and level 1 (vector-vector) operations. The final forms of $H$ and $T$, the transformed $A$ and $B$, are block upper Hessenberg and upper triangular, respectively.

Initially, $B$ is reduced to upper triangular form using a blocked (level 3) $QR$ factorization algorithm [1], [3]. The transformations are also applied to $A$, resulting in an equivalence transformation of $(A, B)$. ¿From now on we focus on the reduction of $A$ to upper block Hessenberg form, while preserving $B$ upper triangular.

The matrices $A$ and $B$ are partitioned into $M \times M$ square blocks $A_{i,j}$ and $B_{i,j}$, respectively, of size $nb \times nb$, i.e., $M = n/nb$ and the block size $nb$ is chosen with respect to the underlying architecture. Without loss of generality, we assume that $nb$ is a multiple of $n$.

The following example used to present the algorithm shows $A$ and $B$ in block form after one iteration of the block reduction algorithm. The symbol $\square$ shows a dense block and the symbol ◹ shows an upper triangular block. All blocks are of size $nb \times nb$ and $M = 6$.

$$
A^{(1)} = \begin{bmatrix} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \text{◹} & \square & \square & \square & \square & \square \end{bmatrix}, \quad B^{(1)} = \begin{bmatrix} \text{◹} & \square & \square & \square & \square & \square \\ 0 & \text{◹} & \square & \square & \square & \square \\ 0 & 0 & \text{◹} & \square & \square & \square \\ 0 & 0 & 0 & \text{◹} & \square & \square \\ 0 & 0 & 0 & 0 & \text{◹} & \square \\ 0 & 0 & 0 & 0 & 0 & \text{◹} \end{bmatrix}
$$

This form is obtained by first reducing $A_{M,1}$ to triangular form. The resulting (left) orthogonal transformation $U$ (of size $nb \times nb$), where $U^T A_{M,1}$ is upper triangular, is also applied to $B$, which destroys the structure of $B_{M,M}$. The triangular form of $B_{M,M}$ is restored via an $RQ$ factorization of the diagonal block, giving the right hand side transformation $V$ (of size $nb \times nb$). The application of $U$ and $V$ onto the remaining blocks in the last block row of $A$ and the last block columns of $A$ and $B$ are performed using the level 3 BLAS routine DGEMM for general matrix multiply and add: $A_{M,2:M} \leftarrow U^T A_{M,2:M}$, $A_{:,M} \leftarrow A_{:,M} V$, $B_{1:M-1,M} \leftarrow B_{1:M-1,M} V$.

Below follows a sequence of illustrations that describe the operations performed on different blocks of $A$ and $B$ during the remaining $M-2(=4)$ iterations to reduce the first block column to block upper Hessenberg form. The symbols $\triangleleft$, $\triangleright$ and $\bowtie$ illustrate level 3 updates from the right, left and both hand sides, respectively. Lower level reductions and updates are shown by the symbols $\otimes$ (two sided) and $\textcircled{$\triangleright$}$ (left hand side). $A^{(i)}$ and $B^{(i)}$ are the reduced $A$ and $B$ after iteration $i(=1:M-1)$.

$$A^{(2)} = \begin{bmatrix} \square & \square & \square & \square & \triangleleft & \triangleleft \\ \square & \square & \square & \square & \triangleleft & \triangleleft \\ \square & \square & \square & \square & \triangleleft & \triangleleft \\ \square & \square & \square & \square & \triangleleft & \triangleleft \\ \textcircled{$\triangleright$} & \triangleright & \triangleright & \triangleright & \bowtie & \bowtie \\ 0 & \triangleright & \triangleright & \triangleright & \bowtie & \bowtie \end{bmatrix}, \quad B^{(2)} = \begin{bmatrix} \diagdown & \square & \square & \square & \triangleleft & \triangleleft \\ 0 & \diagdown & \square & \square & \triangleleft & \triangleleft \\ 0 & 0 & \diagdown & \square & \triangleleft & \triangleleft \\ 0 & 0 & 0 & \diagdown & \triangleleft & \triangleleft \\ 0 & 0 & 0 & 0 & \otimes & \otimes \\ 0 & 0 & 0 & 0 & 0 & \otimes \end{bmatrix}$$

The second iteration starts by annihilating the triangular block $A_{M,1}$ and reducing $A_{M-1,1}$ to upper triangular form. When applied to $B$, these (left) orthogonal transformations destroy the structure of the diagonal blocks $B_{M-1,M-1}$ and $B_{M,M}$ and introduce a non-zero subdiagonal block $B_{M,M-1}$. The triangular structure of $B$ is restored by computing an $RQ$ factorization of

$$\begin{bmatrix} B_{M-1,M-1} & B_{M-1,M} \\ B_{M,M-1} & B_{M,M} \end{bmatrix}.$$

The accumulated left and right orthogonal transformations $U$ and $V$ (now both of size $2nb \times 2nb$) from these two steps are applied to the remaining parts of block columns and rows $M-1$ and $M$: $A_{M-1:M,2:M} \leftarrow U^T A_{M-1:M,2:M}$, $A_{:,M-1:M} \leftarrow A_{:,M-1:M}V$, $B_{1:M-2,M-1:M} \leftarrow B_{1:M-2,M-1:M}V$.

Similar operations are performed in the remaining iterations as illustrated below. In iteration 3, $A_{5,1}$ and $A_{4,1}$ are annihilated and triangularized, respectively. Notice that some $B$-blocks are updated from the left hand side using level 3 routines. Similarly, in iteration 4, $A_{4,1}$ and $A_{3,1}$ are annihilated and triangularized, respectively.

$$A^{(3)} = \begin{bmatrix} \square & \square & \square & \triangleleft & \triangleleft & \square \\ \square & \square & \square & \triangleleft & \triangleleft & \square \\ \square & \square & \square & \triangleleft & \triangleleft & \square \\ \textcircled{$\triangleright$} & \triangleright & \triangleright & \bowtie & \bowtie & \triangleright \\ 0 & \triangleright & \triangleright & \bowtie & \bowtie & \triangleright \\ 0 & \square & \square & \triangleleft & \triangleleft & \square \end{bmatrix}, \quad B^{(3)} = \begin{bmatrix} \diagdown & \square & \square & \triangleleft & \triangleleft & \square \\ 0 & \diagdown & \square & \triangleleft & \triangleleft & \square \\ 0 & 0 & \diagdown & \triangleleft & \triangleleft & \square \\ 0 & 0 & 0 & \otimes & \otimes & \triangleright \\ 0 & 0 & 0 & 0 & \otimes & \triangleright \\ 0 & 0 & 0 & 0 & 0 & \diagdown \end{bmatrix}$$

$$A^{(4)} = \begin{bmatrix} \square & \square & \triangleleft & \triangleleft & \square & \square \\ \square & \square & \triangleleft & \triangleleft & \square & \square \\ \textcircled{$\triangleright$} & \triangleright & \bowtie & \bowtie & \triangleright & \triangleright \\ 0 & \triangleright & \bowtie & \bowtie & \triangleright & \triangleright \\ 0 & \square & \triangleleft & \triangleleft & \square & \square \\ 0 & \square & \triangleleft & \triangleleft & \square & \square \end{bmatrix}, \quad B^{(4)} = \begin{bmatrix} \diagdown & \square & \triangleleft & \triangleleft & \square & \square \\ 0 & \diagdown & \triangleleft & \triangleleft & \square & \square \\ 0 & 0 & \otimes & \otimes & \triangleright & \triangleright \\ 0 & 0 & 0 & \otimes & \triangleright & \triangleright \\ 0 & 0 & 0 & 0 & \diagdown & \square \\ 0 & 0 & 0 & 0 & 0 & \diagdown \end{bmatrix}$$

After $M-1$ iterations the first block column of $A$ is reduced to block upper Hessenberg form with $A_{2,1}$ upper triangular, while preserving $B$ upper triangular.

$$A^{(5)} = \begin{bmatrix} \square & \triangleleft & \triangleleft & \square & \square & \square \\ \textcircled{\triangleright} & \bowtie & \bowtie & \triangleright & \triangleright & \triangleright \\ 0 & \bowtie & \bowtie & \triangleright & \triangleright & \triangleright \\ 0 & \triangleleft & \triangleleft & \square & \square & \square \\ 0 & \triangleleft & \triangleleft & \square & \square & \square \\ 0 & \triangleleft & \triangleleft & \square & \square & \square \end{bmatrix}, \quad B^{(5)} = \begin{bmatrix} \diagdown & \triangleleft & \triangleleft & \square & \square & \square \\ 0 & \otimes & \otimes & \triangleright & \triangleright & \triangleright \\ 0 & 0 & \otimes & \triangleright & \triangleright & \triangleright \\ 0 & 0 & 0 & \diagdown & \square & \square \\ 0 & 0 & 0 & 0 & \diagdown & \square \\ 0 & 0 & 0 & 0 & 0 & \diagdown \end{bmatrix}$$

To complete the block reduction similar operations are performed on the trailing $M-2$ block columns.

# 3 Annihilation of elements

There are several ways to annihilate block-subdiagonal elements of $A$ and fill-in elements in $B$. We describe five variants, where two of them use Givens rotations, two of them use Householder reflections and one uses a combination of Givens and Householder transformations. All five are based on the level 3 reduction algorithm outlined in Section 2.2.

## 3.1 Givens 1

In the implementation called Givens 1, elements are annihilated by use of Givens rotations following the elementwise algorithm in Section 2.1. All the other blocks involved are updated by use of the level 3 BLAS DGEMM.

Specifically, when an element of $A$ is annihilated by a left hand side rotation, the diagonal block of $B$ is immediately updated. The resulting (single) fill-in element is then annihilated by a right hand side rotation. This rotation, when applied to $A$ do not affect the $A$-block under annihilation and may therefore be accumulated and applied later.

## 3.2 Givens 2

The variant Givens 2 differs from Givens 1 in the following way. Instead of eliminating the fill-in immediately all rotations from the left hand side (to kill one block of $A$) are accumulated and applied to the whole current block row of $B$ in one single DGEMM operation. This will destroy the structure of the involved diagonal blocks of $B$ as illustrated in the following example:

$$U^T B^{(2)}_{M-1:M,M-1:M} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ f_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ f_2 & f_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ f_3 & f_2 & f_1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ f_4 & f_3 & f_2 & f_1 & \cdot & \cdot & \cdot & \cdot \\ & f_4 & f_3 & f_2 & f_2 & \cdot & \cdot & \cdot \\ & & f_4 & f_3 & f_3 & f_3 & \cdot & \cdot \\ & & & f_4 & f_4 & f_4 & f_4 & \cdot \end{bmatrix}.$$

The example shows the fill-in introduced in $B$ resulting from iteration 2 of the example in Section 2.2. The fill-in elements are labeled $f_j$, where $j$ tells what column of $A$ introduced the fill-in. To restore $B$ to upper triangular form $U^T B^{(2)}_{M-1:M,M-1:M}$ is $RQ$-factorized using a routine based on Givens rotations that takes the new structure of $B^{(2)}_{M-1:M,M-1:M}$ into account.

### 3.3 Householder 1

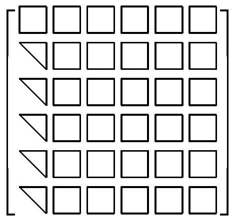
Householder 1 follows the same operation pattern as Givens 2, but here Householder reflections are used in all annihilations. Refering to the last example, $U^T B^{(2)}_{M-1:M,M-1:M}$ will now be a full matrix. The $RQ$ factorization that restores $B$ to triangular form is also based on Householder reflections.

### 3.4 Householder 2

In all previous described implementations the orthogonal transformations $U$ and $V$ have been explicitly constructed before applying the DGEMM updates (which require a temporary array of size $2nb \times n$). In Householder 2, the compact WY representation of Householder matrices is used to reduce the amount of extra storage and copying [11] . We compute $Y$ (of size $2nb \times nb$) and $S$ (of size $nb \times nb$) such that $U$ (or $V$) $= I - YSY^T$. When calling the LAPACK routines for computing a $QR$ or $RQ$ factorization, the Householder vectors $Y$ are returned as the lower part of the factorized block. To extract the triangular matrix $S$ the LAPACK routine DLARFT is called and to apply the reflections to the remaining part of the matrices $A$ and $B$ the LAPACK routine DLARFB is used.

### 3.5 Givens-Householder

In this implementation both Givens and Householder transformations are used. We extend techniques for reducing a non-symmetric matrix $A$ to condensed form [10], [2] to matrix pairs. First the current block column of $A$ to be reduced is $QR$-factorized block by block using Householder reflections. This step results in a shark fin shaped block column.

$A =$ 

Next all triangular blocks (except the subdiagonal block) are annihilated using Givens rotations. The lower of two subsequent upper triangular blocks is annihilated column by column from left to right as follows: to kill an element $a_{i,j}$, $a_{i-1,j}$ is used as pivot. If $a_{i-1,j}$ refer to an element in the upper of the two triangles, then $a_{j,j}$ becomes the pivot element. We illustrate the elimination of the first two columns of two $4 \times 4$ subsequent triangular blocks.

$$\begin{bmatrix} p & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet \\ & & \bullet & \bullet \\ & & & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet \\ & & \bullet & \bullet \\ & & & \bullet \end{bmatrix}, \quad \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet \\ & & \bullet & \bullet \\ & & & \bullet \\ & p & \bullet & \bullet \\ & \bullet & \bullet & \bullet \\ & & \bullet & \bullet \\ & & & \bullet \end{bmatrix}, \quad \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ & p & \bullet & \bullet \\ & & \bullet & \bullet \\ & & & \bullet \\ & \bullet & \bullet & \bullet \\ & & \bullet & \bullet \\ & & \bullet & \bullet \\ & & & \bullet \end{bmatrix}, \quad \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet \\ & & \bullet & \bullet \\ & & & \bullet \\ & & \bullet & \bullet \\ & & p & \bullet \\ & & \bullet & \bullet \\ & & & \bullet \end{bmatrix}.$$

The transformations are of course applied to the associated remaining blocks of $A$ and $B$. The update of $B$ with respect to the $QR$ factorizations of the first block column and the annihilation (with Givens rotations) of the bottom triangular block $A_{M,1}$ results in the following structure ($M = 6$):

$$B = \begin{bmatrix} \diagdown & \square & \square & \square & \square & \square \\ 0 & \diagdown & \square & \square & \square & \square \\ 0 & 0 & \diagdown & \square & \square & \square \\ 0 & 0 & 0 & \diagdown & \square & \square \\ 0 & 0 & 0 & 0 & \diagdown & \square \\ 0 & 0 & 0 & 0 & 0 & \diagdown \end{bmatrix} \rightarrow \begin{bmatrix} \square & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square & \square \\ 0 & 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & 0 & \square & \square \\ 0 & 0 & 0 & 0 & \square & \square \end{bmatrix},$$

i.e. all diagonal blocks of $B$ are full matrices. Moreover, one non-zero subdiagonal block of $B$ is introduced. To restore $B_{M-1:M,M-1:M}$ to triangular form the same operations as in Householder 1 are performed.

## 4 Performance Results

We present performance results of our implementations of the five variants of the level 3 reduction algorithm. The target machine is an IBM RS/6000, model 25T, 66MHz. The results are compared to the elementwise implementation DGGHRD from LAPACK [1]. DGGHRD is modified to compute the same condensed form as the level 3 implementations, i.e., $H$ is block upper Hessenberg and $T$ is upper triangular.

The results in Table 1 show that the implementation Householder 1 outperforms the Givens-based implementations independent of the block size $nb$. Moreover, for small block sizes it is advantageous to explicitly construct the orthogonal matrices $Q$ and $Z$ while for larger block sizes the compact WY approach is to prefer. An explanation to this is that in the WY approach DLARFB is called to perform the update [1]. In DLARFB the update of a block is split into two parts one using the level 3 BLAS DTRMM (triangular matrix multiply) and one using DGEMM. If $Q$ and $Z$ are small the gain in this strategy is lost resulting in poorer performance.

Note that, when the the block size increases, the amount of work decreases and thereby the execution time of DGGHRD.

**Table 1.** Performance results in seconds on IBM RS/6000, model 25T, 66MHz

| $n$ | $nb$ | DGGHRD | Gvns 1 | Gvns 2 | Hsh 1 | Hsh 2 | Gvn-Hsh |
|---|---|---|---|---|---|---|---|
| 512 | 8 | 152.8 | 68.9 | 69.6 | **63.9** | 101.4 | 100.0 |
| 512 | 16 | 144.2 | 52.2 | 52.8 | **44.4** | 64.9 | 62.2 |
| 512 | 32 | 135.4 | 47.9 | 47.7 | **39.6** | 47.3 | 49.1 |
| 512 | 64 | 123.3 | 47.6 | 47.7 | 40.9 | **39.8** | 43.8 |
| 512 | 128 | 93.0 | 46.5 | 45.4 | 41.0 | **27.6** | 35.3 |
| 1024 | 8 | 1232.1 | 570.0 | 563.9 | **499.7** | 882.1 | 849.6 |
| 1024 | 16 | 1199.6 | 408.0 | 411.1 | **341.9** | 553.0 | 524.6 |
| 1024 | 32 | 1167.1 | 365.9 | 364.9 | **296.0** | 397.9 | 391.1 |
| 1024 | 64 | 1089.8 | 382.1 | 384.5 | **322.8** | 374.1 | 379.3 |
| 1024 | 128 | 975.8 | 449.9 | 404.6 | 364.4 | **303.7** | 365.5 |

## 5 Summary

Five variants of a level 3 algorithm for reducing a regular matrix pair $(A, B)$ to block Hessenberg-triangular form have been presented. It has been shown through computational experiments on a high-performance workstation that the new level 3 based algorithms reduce the execution time by a factor 2–4 compared to an elementwise algorithm. Furthermore, our computational experiments show that algorithms based on Householder reflections achieve the best performance among the blocked (level 3) algorithms. Notably, for large enough block sizes the compact WY approach gives the best performance.

The results reported here is ongoing work and we currently extend different blocked variants to parallel algorithms for scalable MIMD computers.

## References


1. E. Anderson, Z. Bai, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. LAPACK Users' Guide, Second Edition. *SIAM Press, Philadelphia*, 1995.
2. M. Berry, J. Dongarra, and Y. Kim. A Highly Parallel Algorithm for the Reduction of a Nonsymmetric Matrix to Block Upper-Hessenberg Form. *Technical Report, LAPACK Working Note 68, University of Tennessee, CS-94-221*, 1994.
3. K. Dackland, E. Elmroth, and B. Kågström. Parallel Block Matrix Factorizations on the Shared Memory Multiprocessor IBM 3090 VF/600J. *The International Journal of Supercomputer Applications*, 6:69–97, 1992.
4. J. Dongarra, J. Du Croz, S. Hammarling, and Richard J. Hanson. An Extended Set of FORTRAN Basic Linear Algebra Subroutines. *ACM Transactions on Mathematical Software*, 14(1):1–17, March 1988.
5. Gene H. Golub and Charles F. Van Loan. Matrix Computations, 2nd ed. *The John Hopkins University Press, Baltimore, Maryland*, 1989.
6. S. Hammarling J. Dongarra, J. Du Croz and I. Duff. A Set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. on Mathematical Software*, 16:1–17, 1990.

7. B. Kågström and C. Van Loan. GEMM-Based Level 3 BLAS. *Tech. rep. CTC91TR47, Department of Computer Science, Cornell University*, 1989.
8. C. Lawson, R. Hanson, D. Kincaid, and F. Krogh. Basic Linear Algebra Subprograms for Fortran Usage. *ACM Transactions on Mathematical Software*, 5:308–323, 1979.
9. C. B. Moler and G. W. Stewart. An Algorithm for Generalized Matrix Eigenvalue Problems. *SIAM J. Num. Anal.*, 10:241–256, 1973.
10. A. Pothen and P. Raghavan. Distributed Orthogonal Factorizations: Givens and Householder Algorithms. *SIAM J. Sci. Statist. Comput.*, 10:1113–1134, 1989.
11. R. Schreiber and C. Van Loan. A Storage Efficient WY Representation for Products of Householder Transformations. *SIAM J. Sci. Statist. Comput.*, 10:53–57, 1989.

This article was processed using the LaTeX macro package with LLNCS style