Math. Commun., Vol. **15**, No. 1, pp. 281-289 (2010)

An accurate SVD algorithm for 2 by 2 triangular matrices*

Josip Matejaš[1,†]

[1] *Faculty of Economics and Business, University of Zagreb, Kennedyjev trg 6, HR-10 000 Zagreb, Croatia*

Received November 2, 2009; accepted March 20, 2010

Abstract. Using a fine accuracy analysis and the results from [9], a new accurate algorithm for computing the singular value decomposition of 2 by 2 triangular matrices is constructed. It is obtained by combining the new algorithm which is derived in [9] and the algorithm which is coded as an xLASV2 computational routine of LAPACK. Relative error bounds for the output data of the hybrid algorithm are equal to or smaller than the same bounds for any of these two algorithms.
AMS subject classifications: 65F15, 65G05
Key words: triangular matrix of order two, SVD algorithm, error analysis, accuracy

1. Introduction

In [9] we have considered two algorithms for computing the singular value decomposition (SVD) of 2 by 2 triangular matrices. First, we have shown how to modify the Voevodin formulas [18, 6] for rotation angles in order to obtain a new accurate algorithm. Using a subtle rounding error analysis, we have derived sharp accuracy bounds for that algorithm. We have also made a similar rounding error analysis for the algorithm coded as LAPACK computational routine xLASV2. The analysis uses natural assumptions which fully comply with the IEEE floating point standard. It expresses final errors as functions of the errors of some initial or intermediate quantities. This enables us to obtain very sharp error bounds for the case of a general and of almost diagonal 2 by 2 triangular matrix. Although most of the error bounds for the new algorithm (in [9] we named it KOGUL) are better than those for xLASV2, we have noticed that it is possible to define a simple hybrid algorithm whose error bounds will be better than or equal to those for the constituent algorithms.

In this paper we use accuracy results from [9] to construct a new hybrid highly accurate algorithm. In [9] it has been shown that relative error bounds for the most of the output parameters are smaller if KOGUL is used. Only one output parameter has a smaller error bound, in a general case, if xLASV2 is used. The main idea how to construct a new algorithm is simply to choose those parts of each algorithm which compute the output data with a better relative accuracy. Thus, we only need to find where this switching should occur. By simple operation count, we show that the hybrid algorithm is as efficient as the original algorithms.

*Presented at the Sixth Conference on Applied Mathematics and Scientific Computing, Zadar, Croatia, September 2009.

†Corresponding author. *Email address:* josip.matejas@kr.htnet.hr (J. Matejaš)

Note that SVD algorithms for 2 by 2 triangular matrices are mostly used as core algorithms for the Kogbetliantz method which is used for computing SVD of n by n triangular matrices (see [12, 13, 10, 2, 1]). Kogbetliantz method is well understood. Its asymptotic quadratic convergence is studied in [5, 17, 1, 7, 8, 15] while its global convergence is proved in [4] and [3]. Its implementation details and parallelization attempts can be found in [2] and [11]. What still remains to prove is its relative accuracy. The first attempt in this direction was made in [16]. It is obvious that in obtaining sharp accuracy bounds for the Kogbetliantz method, one has to use the sharpest available error bounds for the output data of the core algorithm. And here the results of this paper come into play.

This paper is closely related to [9]. We use the same notation (without introducing it) and the same technique of the proofs. To keep the exposition short, we assume the reader is acquainted with [9].

The paper is organized as follows. In Section 2, we illustrate the new approach to accuracy analysis which is used in [14, 9, 16]. In Section 3 the new hybrid algorithm is constructed and its accuracy is proved.

2. The subtle error analysis

Accuracy results from [9], which are used in this paper, are obtained by a new approach to rounding error analysis. It is based on three assumptions:

- *do not neglect any part of the error.* Thus, we use proper error estimates (note that nonlinear parts of the error $\boldsymbol{\epsilon}$, like the terms of higher order in $\boldsymbol{\epsilon}$, are usually neglected).
- *take into account the signs of the errors,* which is the most important thing in the analysis.
- *assume that the unit round-off* $\mathbf{u}$ *satisfies*

$$\mathbf{u} \in \left\{2^{-23},\ 2^{-24},\ 2^{-52},\ 2^{-53},\ 2^{-64}\right\}. \tag{1}$$

  This assumption follows the trend of contemporary computers. Today they all comply with the IEEE standard.

We shall see that by such approach error bounds can be significantly improved. We use a standard model of the machine arithmetic: the floating point result for the basic arithmetic operations $\circ$ satisfies

$$\begin{aligned} \mathrm{fl}(a \circ b) &= (a \circ b)(1+\varepsilon), \quad |\varepsilon| \ \leq \mathbf{u}, \quad \circ \ \in \{+,-,*,/\}, \quad \varepsilon = \varepsilon(a,b,\circ)\,, \\ \mathrm{fl}(\sqrt{a}) \ &= \sqrt{a}(1+\varepsilon_{\surd}), \quad |\varepsilon_{\surd}| \leq \mathbf{u}, \ \ \varepsilon_{\surd} = \varepsilon_{\surd}(a). \end{aligned} \tag{2}$$

In numerical expressions we frequently meet *suppression* and *cancelation* of the initial errors. The new approach takes these into account. The following example illustrates how suppression of errors takes place.

Example 1. *Let us estimate the error in the evaluation of the expression*

$$y = \frac{1}{1+x}\,, \quad x > 0. \tag{3}$$

Suppose that we have only an approximation of x, $\mathrm{fl}(x) = (1+\boldsymbol{\epsilon})x$ at our disposal. Now, we have

$$\mathrm{fl}(y) = \frac{1+\varepsilon_2}{1+\varepsilon_1} \cdot \frac{1}{1+(1+\boldsymbol{\epsilon})x} = \frac{1+\varepsilon_2}{1+\varepsilon_1} \cdot \frac{1}{1+\frac{\boldsymbol{\epsilon}x}{1+x}} \cdot \frac{1}{1+x}.$$

Here ε_1 comes from addition and ε_2 comes from division. Using the identity

$$\frac{1+a}{1+b} = 1 + a - b + \frac{b^2 - ab}{1+b},$$

we separate linear and nonlinear parts of errors in the above two terms,

$$\frac{1+\varepsilon_2}{1+\varepsilon_1} = 1+\varepsilon_2-\varepsilon_1+\frac{\varepsilon_1^2-\varepsilon_1\varepsilon_2}{1+\varepsilon_1}, \qquad \frac{1}{1+\frac{\boldsymbol{\epsilon}x}{1+x}} = 1 - \underbrace{\frac{\boldsymbol{\epsilon}x}{1+x}}_{\text{suppression}} + \frac{\boldsymbol{\epsilon}^2x^2}{(1+x)(1+x+\boldsymbol{\epsilon}x)}.$$

Thus, we obtain

$$\mathrm{fl}(y) = \left(1+\varepsilon_2-\varepsilon_1-\frac{\boldsymbol{\epsilon}x}{1+x}+\eta_1\right)\cdot y\,, \tag{4}$$

where $|\varepsilon_1|, |\varepsilon_2| \leq \mathbf{u}$ (assumption (2)) and η_1 stands for the nonlinear part of the error,

$$\begin{aligned}\eta_1 = {} & \frac{(\varepsilon_1-\varepsilon_2)\boldsymbol{\epsilon}x}{1+x} + \left(1-\frac{\boldsymbol{\epsilon}x}{1+x}\right)\cdot\frac{\varepsilon_1^2-\varepsilon_1\varepsilon_2}{1+\varepsilon_1} \\ & + \left(1+\varepsilon_2-\varepsilon_1+\frac{\varepsilon_1^2-\varepsilon_1\varepsilon_2}{1+\varepsilon_1}\right)\cdot\frac{\boldsymbol{\epsilon}^2x^2}{(1+x)(1+x+\boldsymbol{\epsilon}x)}.\end{aligned}$$

From relation (4) we can see that the initial error $\boldsymbol{\epsilon}$ is suppressed by the factor $x/(1+x)$ which is less than 1. One may expect that the initial error in the subsequent computations will increase but, in many instances, it is not true. For example, if $x = 0.1$, then the obtained error for $\mathrm{fl}(y)$ is around 11 times smaller than the initial one for $\mathrm{fl}(x)$.

The second example illustrates how cancelation takes place.

Example 2. *We consider the expression*

$$z = \frac{x}{1+x}\,, \quad x > 0\,, \tag{5}$$

and again we assume to have an approximation of x, $\mathrm{fl}(x) = (1+\boldsymbol{\epsilon})x$. Similarly to the above we have

$$\mathrm{fl}(z) = \frac{1+\varepsilon_4}{1+\varepsilon_3}\cdot\frac{(1+\boldsymbol{\epsilon})x}{1+(1+\boldsymbol{\epsilon})x} = \frac{1+\varepsilon_4}{1+\varepsilon_3}\cdot\frac{1+\boldsymbol{\epsilon}}{1+\frac{\boldsymbol{\epsilon}x}{1+x}}\cdot\frac{x}{1+x}.$$

Since $(1+\varepsilon_4)/(1+\varepsilon_3) = 1+\varepsilon_4-\varepsilon_3+(\varepsilon_3^2-\varepsilon_3\varepsilon_4)/(1+\varepsilon_3)$, and

$$\frac{1+\boldsymbol{\epsilon}}{1+\frac{\boldsymbol{\epsilon}x}{1+x}} = 1+\underbrace{\boldsymbol{\epsilon}-\frac{\boldsymbol{\epsilon}x}{1+x}}_{\text{cancelation}} + \frac{\boldsymbol{\epsilon}^2x^2-\boldsymbol{\epsilon}^2x(1+x)}{(1+x)(1+x+\boldsymbol{\epsilon}x)} = 1+\frac{\boldsymbol{\epsilon}}{1+x}-\frac{\boldsymbol{\epsilon}^2x}{(1+x)(1+x+\boldsymbol{\epsilon}x)},$$

we obtain

$$\mathrm{fl}(z) = \left(1 + \varepsilon_4 - \varepsilon_3 + \frac{\boldsymbol{\epsilon}}{1+x} + \eta_2\right) \cdot z\,, \tag{6}$$

where $|\varepsilon_3|, |\varepsilon_4| \leq \mathbf{u}$ and

$$\begin{aligned}\eta_2 = {} & \frac{(\varepsilon_4 - \varepsilon_3)\boldsymbol{\epsilon}}{1+x} + \left(1 + \frac{\boldsymbol{\epsilon}}{1+x}\right) \cdot \frac{\varepsilon_3^2 - \varepsilon_3\varepsilon_4}{1+\varepsilon_3} \\ & - \left(1 + \varepsilon_4 - \varepsilon_3 + \frac{\varepsilon_3^2 - \varepsilon_3\varepsilon_4}{1+\varepsilon_3}\right) \cdot \frac{\boldsymbol{\epsilon}^2 x}{(1+x)(1+x+\boldsymbol{\epsilon}x)}\,.\end{aligned}$$

Here the cancelation takes place: the initial error cancels: $\boldsymbol{\epsilon} - \boldsymbol{\epsilon}x/(1+x)$, and the resulting error $\boldsymbol{\epsilon}/(1+x)$ is again smaller than the initial one. For example, if $x = 100$ and $\boldsymbol{\epsilon} = 10\%$, so that we start with a very bad approximation of x, then $\boldsymbol{\epsilon}/(1+x) < 0.1\%$. This gives an extremely accurate result when compared with the initial data. It is amazing. One may think that, with such bad approximation of the initial data, nothing can be done. However, we have obtained a pretty accurate approximation of z.

In a single Jacobi or Kogbetliantz transformation (see [14, 9]), we compute $\sin\varphi$ and $\cos\varphi$ from $\tan\varphi$ or $\tan 2\varphi$ which is computed from the matrix elements. If we put $x = \tan^2\varphi$ in relations (3) and (5), then we have $y = \cos^2\varphi$ and $z = \sin^2\varphi$. It is obvious from relations (4) and (6) that the initial inaccuracy $\boldsymbol{\epsilon}$ does not blow up if we compute $\sin\varphi$ and $\cos\varphi$ in such way. Moreover, since

$$\frac{\boldsymbol{\epsilon}}{1+x} \to \boldsymbol{\epsilon}, \quad \frac{\boldsymbol{\epsilon}x}{1+x} \to 0 \ \text{ as } \ x \to 0 \qquad \text{and} \qquad \frac{\boldsymbol{\epsilon}}{1+x} \to 0, \quad \frac{\boldsymbol{\epsilon}x}{1+x} \to \boldsymbol{\epsilon} \ \text{ as } \ x \to \infty,$$

we can see that $\cos\varphi$ will have a high relative accuracy in spite of the initial inaccuracy in $\tan\varphi$ provided that the rotation angle is small enough ($x = \tan^2\varphi \to 0$). The same is true for $\sin\varphi$ in the case of a large rotation angle ($x = \tan^2\varphi \to \infty$). This short analysis provides a deeper insight in the nature of relative error bounds given in the table in [9, sec. 5] (see Table 1).

3. The hybrid algorithm

In [9] we have used a fine accuracy analysis of two SVD algorithms for 2 by 2 triangular matrices. The first one is the new algorithm KOGUL (see [9, Sec. 2] for details) and the second one is a LAPACK auxiliary routine xLASV2. Here, we combine these two algorithms to produce a new one which will have better accuracy properties than any of them alone.

Relative error bounds for the two algorithms are summarized in Table 1. Here $s1$, $s2$ and $c1$, $c2$ denote the sines and cosines of the first rotation angle φ and the second angle ψ, respectively, while q denotes the multiplicative correcting factor for updating the diagonal elements. It is important that the error bounds from Table 1 are attainable (or almost attainable in finite arithmetic). The arguments for this fact lie in the analysis applied. Namely, the bounds have been obtained directly from the exact expressions for the errors. The linear form of the errors has been

error	**general**		$\|t1\|, \|t2\| \leq 1/10$	
	KOGUL	xLASV2	KOGUL	xLASV2
$\|\boldsymbol{\epsilon}_{c1}\|$	$2.82\,\mathbf{u}$	$30.01\,\mathbf{u}$	$2.26\,\mathbf{u}$	$11.53\,\mathbf{u}$
$\|\boldsymbol{\epsilon}_{s1}\|$	$10.44\,\mathbf{u}$	$23.01\,\mathbf{u}$	$10.44\,\mathbf{u}$	$20.06\,\mathbf{u}$
$\|\boldsymbol{\epsilon}_{c2}\|$	$14.80\,\mathbf{u}$	$20.01\,\mathbf{u}$	$2.88\,\mathbf{u}$	$3.15\,\mathbf{u}$
$\|\boldsymbol{\epsilon}_{s2}\|$	$14.80\,\mathbf{u}$	$20.01\,\mathbf{u}$	$14.80\,\mathbf{u}$	$17.03\,\mathbf{u}$
$\|\boldsymbol{\epsilon}_{q}\|$	$15.48\,\mathbf{u}$	$6.01\,\mathbf{u}$	$3.56\,\mathbf{u}$	$6.01\,\mathbf{u}$

Table 1. *Relative error bounds*

obtained by replacing nonlinear parts with the bound of the form $\alpha\mathbf{u}$, where α is obtained by replacing $\mathbf{u}$ with 2^{-23} (which corresponds to any mode of rounding in single precision arithmetic, not just the default mode – rounding to the nearest). Therefore, that α is good for all modes and all precisions of the finite arithmetic. Since the inequalities in (2) are attainable, so are the final bounds.

We can see from Table 1 that the error bounds for $s1$, $c1$ and $s2$, $c2$ are smaller if the algorithm KOGUL is used. On the other hand, the relative error bound for q is smaller, in the general case, if an xLASV2 routine is used. The reason lies in the fact that the LAPACK routine computes q first and after that all other quantities. But, if the angles are small, then KOGUL has again better bounds.

Therefore, the idea how to construct a new hybrid algorithm is simple. In the KOGUL algorithm we have to implement part of the xLASV2 routine which computes the factor q, and to activate it when appropriate, for larger rotation angles. So, let us find when this switching should occur.

We use [9, Lemma 3.7. (iii)] which corresponds to the case $|\varphi| < \pi/8$, $|\psi| \leq \pi/4$. We shall express $\boldsymbol{\epsilon}_q$ as a function of the rotation angles and then compare it with the bound $6.01\mathbf{u}$. Note that this bound $6.01\mathbf{u}$ for q is independent of the angles in xLASV2, since q is computed first from the matrix elements. Using the notation from [9, Lemma 3.7. (iii)], we have

$$\boldsymbol{\epsilon}_q = \frac{w}{2}\left(2\vartheta(s2)^2 + w - 1\right)\boldsymbol{\epsilon}_\xi + (\vartheta\eta_{t1} + \eta_{t2})\,(s2)^2 + \frac{\eta_{\mu 2}}{2} + \frac{\eta_\alpha}{2} + \eta_q\,,$$

where $w = \cos 2\varphi$, $\vartheta = h \cdot t1/(h \cdot t1 + g)$ and η_{t1}, η_{t2}, $\eta_{\mu 2}$, η_α and η_q are estimated by the relations [9, A.30, A.35, A.39, A.29] and [9, A.57], respectively. Here f and h are the diagonal elements and g is the off-diagonal element of the initial upper-triangular matrix. As explained in [9] one can assume $f \geq h > 0$ and therefore $0 \leq \vartheta \leq 1$ holds.

Since $w - 1 = -2\sin^2\varphi = -2(s1)^2$, we obtain

$$\begin{aligned}
|\boldsymbol{\epsilon}_q| &\leq \frac{w}{2}\left|2\vartheta(s2)^2 - 2(s1)^2\right| \cdot |\boldsymbol{\epsilon}_\xi| + (\vartheta|\eta_{t1}| + |\eta_{t2}|)\,(s2)^2 + \frac{|\eta_{\mu 2}|}{2} + \frac{|\eta_\alpha|}{2} + |\eta_q| \\
&\leq \max\left\{(s1)^2, (s2)^2\right\}|\boldsymbol{\epsilon}_\xi| + \left[(3.026 + 3.001)(s2)^2 + 0.751 + 1.188 + 1.501\right] \cdot \mathbf{u} \\
&\leq \left[6.01\max\left\{(s1)^2, (s2)^2\right\} + 6.027(s2)^2 + 3.44\right] \cdot \mathbf{u} \\
&\leq \left[12.037\max\left\{(s1)^2, (s2)^2\right\} + 3.44\right] \cdot \mathbf{u}\,, \qquad (7)
\end{aligned}$$

where we have also used [9, Lemma 3.4.] (for $|\boldsymbol{\epsilon}_\xi| \leq 6.01\mathbf{u}$). The obtained bound will be compared with $6.01\mathbf{u}$ (which is the bound for $|\boldsymbol{\epsilon}_q|$ in xLASV2, for the general

case). Thus, we have

$$12.037\max\left\{(s1)^2,(s2)^2\right\}+3.44\le 6.01 \quad\Rightarrow\quad \max\left\{(s1)^2,(s2)^2\right\}\le 0.21351$$

and hence

$$\max\{|s1|,|s2|\}\le\sqrt{0.21351}\approx 0.4621 \quad\text{or}\quad \max\{|\varphi|,\ |\psi|\}\le 0.4804\ . \tag{8}$$

Since [9, Lemma 3.7.(*iii*)] covers the case $|\varphi|<\pi/8$, $|\psi|\le 0.4804\approx 1.22333\cdot(\pi/8)$, it remains to consider the case $\pi/8\le|\varphi|\le 0.4804$, $|\psi|\le 0.4804$ which is covered by [9, Lemma 3.7.(*i*)]. We have

$$\boldsymbol{\epsilon}_q=\frac{z}{2}\left(1-z-2\vartheta(s2)^2\right)\boldsymbol{\epsilon}_\xi+\left(\vartheta\eta_{t1}+\eta_{t2}\right)(s2)^2+\frac{\eta_{\mu 2}}{2}+\frac{\eta_\alpha}{2}+\eta_q,$$

where $z=\cos 2\varphi$, $\vartheta=h\cdot t1/(h\cdot t1+g)$ and η_{t1}, η_{t2}, $\eta_{\mu 2}$, η_α and η_q are estimated by the relations [9, A.26, A.35, A.39, A.21] and [9, A.51], respectively. Since $1-z=2\sin^2\varphi=2(s1)^2$, in the same way as above we obtain

$$\begin{aligned}|\boldsymbol{\epsilon}_q|&\le\max\left\{(s1)^2,(s2)^2\right\}\cdot 6.01\mathbf{u}\\&\quad+\left[(3.026+3.001)(s2)^2+0.751+1.07+1.501\right]\cdot\mathbf{u}\\&\le\left[12.037\max\left\{(s1)^2,(s2)^2\right\}+3.322\right]\cdot\mathbf{u}\,.\end{aligned}\tag{9}$$

Thus, if $12.037\max\left\{(s1)^2,(s2)^2\right\}+3.322\le 6.01$, then $\max\left\{(s1)^2,(s2)^2\right\}\le 0.22332$ or equivalently

$$\max\{|s1|,|s2|\}\le 0.4726 \quad\text{which means}\quad \max\{|\varphi|,\ |\psi|\}\le 0.4922. \tag{10}$$

Now, using the relations (8), (10) and [9, Lemma 3.7.], we make the following conclusions:

- If $|d|\le e$ and $|h\cdot t1+g|\le f$ (that is, $\frac{\pi}{8}\le|\varphi|\le\frac{\pi}{4}$, $|\psi|\le\frac{\pi}{4}$), then
  if $|s1|,|s2|\le 0.46$ (i.e. $|\varphi|,|\psi|\le 0.48$), then KOGUL, otherwise xLASV2 is applied;
- If $|d|\le e$ and $|h\cdot t1+g|>f$ (that is, $\frac{\pi}{8}\le|\varphi|\le\frac{\pi}{4}$, $\frac{\pi}{4}<|\psi|\le\frac{\pi}{2}$), then xLASV2 is applied;
- If $|d|>e$ and $|h\cdot t1+g|\le f$ (that is, $|\varphi|<\frac{\pi}{8}$, $|\psi|\le\frac{\pi}{4}$), then
  if $|s2|\le 0.46$ (i.e. $|\psi|\le 0.48$), then KOGUL, otherwise xLASV2 is applied;
- If $|d|>e$ and $|h\cdot t1+g|>f$ (that is, $|\varphi|<\frac{\pi}{8}$, $\frac{\pi}{4}<|\psi|\le\frac{\pi}{2}$), then xLASV2 is applied.

Hence we can construct the following hybrid algorithm. We use a logical variable "test" to switch from KOGUL to xLASV2 and vice versa. Here is the code of the algorithm.

%%% Algorithm (HYBRID)

test = true %% initialization for KOGUL
if $|g| > f$ **then** $d = g + ((f+h)/g) * (f-h)$; $e = 2*h$
else $d = (f-h)/g + g/(f+h)$; $e = 2*h/(f+h)$
end if

if $|d| \le e$ **then** %%% ξ is now $\cot 2\varphi$
$\xi = d/e$; $\mu = 1+\xi^2$; $\rho = \sqrt{\mu}$; $\alpha = 1 + \frac{|\xi|}{\rho}$
$c1 = \sqrt{\frac{1}{2} * \alpha}$; $s1 = \mathrm{sgn}(\xi)/\sqrt{2*\alpha*\mu}$; $t1 = \mathrm{sgn}(\xi)/(|\xi| + \rho)$
if $|s1| > 0.46$
then test = false %% switch to xLASV2
end if
else %%% ξ is now $\tan 2\varphi$
if $|d| * \mathbf{u} > e$ **then** $c1 = 1$; $t1 = (e/2)/d$; $s1 = t1$; $\xi = 0$; $\alpha = 2$
else
$\xi = e/d$; $\mu = 1+\xi^2$; $\rho = \sqrt{\mu}$; $\alpha = 1 + \frac{1}{\rho}$
$c1 = \sqrt{\frac{1}{2} * \alpha}$; $s1 = \xi/\sqrt{2*\alpha*\mu}$; $t1 = \xi/(1+\rho)$
endif
endif

if $|h * t1 + g| \le f$ **then** %%% $t2$ is $\tan\psi$
$t2 = (h*t1+g)/f$; $\mu 2 = 1 + t2*t2$; $\rho 2 = \sqrt{\mu 2}$; $c2 = 1/\rho 2$; $s2 = t2/\rho 2$
if $|s2| > 0.46$
then test = false %% switch to xLASV2
end if
else %%% $\tau 2$ is $\cot\psi$
test = false %% switch to xLASV2
$\tau 2 = f/(h*t1+g)$; $\mu 2 = 1 + \tau 2 * \tau 2$; $\rho 2 = \sqrt{\mu 2}$
$s2 = sgn(\tau 2)/\rho 2$; $c2 = |\tau 2|/\rho 2$
end if
if (test) **then** $q = \sqrt{\frac{1}{2} * \alpha * \mu 2}$ %% KOGUL
else %% xLASV2
$d = f - h$; $\lambda = d/f$; $m = g/f$; $\tau = 2 - \lambda$
$s = \sqrt{\tau^2 + m^2}$; $r = \sqrt{\lambda^2 + m^2}$; $q = (s+r)/2$
end if
% update the diagonal elements
$f' = f * q$; $h' = h/q$

The algorithm might look complicated but it is not because only one half of the code is active depending on the decisions in the "if" commands. The relative error bounds for this hybrid algorithm are obviously given in Table 1. The bounds for $\boldsymbol{\epsilon}_{s1}$, $\boldsymbol{\epsilon}_{c1}$, $\boldsymbol{\epsilon}_{s2}$ and $\boldsymbol{\epsilon}_{c2}$ are the same as for the KOGUL algorithm. The bound for $\boldsymbol{\epsilon}_q$ is given by relations (7) and (9) for the angles which are defined by (8) and (10), respectively. For all other angles the bound is $6.01\mathbf{u}$, as in the general case for xLASV2 from Table 1. *Thus,* $|\boldsymbol{\epsilon}_q|$ *is never greater than* $6.01\mathbf{u}$.

For example, let $\max\{|t1|, |t2|\} = \eta$. Since $\max\{(s1)^2, (s2)^2\} = \eta^2/(1+\eta^2)$, by using (8) we have $\eta^2/(1+\eta^2) \le 0.21351$ which implies $\eta \le 0.521$. Thus, using the

bound (7), which covers the bound (9), we obtain

$$|\boldsymbol{\epsilon}_q| \;\leq\; \begin{cases} \left(\dfrac{12.037\,\eta^2}{1+\eta^2} + 3.44\right)\mathbf{u} \;<\; 6.01\mathbf{u}, & \text{if } \; 0 < \eta \leq 0.521\,, \\ 6.01\mathbf{u}, & \text{if } \; \eta > 0.521. \end{cases}$$

If, as in Table 1, $\eta = 1/10$, then the above relation yields $|\boldsymbol{\epsilon}_q| \leq 3.5592\mathbf{u}$, which can be found in Table 1. Note again that, for any particular value of rotation angles, [9, Lemma 3.4. and Lemma 3.7.] yield the sharpest possible estimates for the errors $\boldsymbol{\epsilon}_{c1}$, $\boldsymbol{\epsilon}_{s1}$, $\boldsymbol{\epsilon}_{c2}$, $\boldsymbol{\epsilon}_{s2}$ and $\boldsymbol{\epsilon}_q$, sharper than those summarized in Table 1.

Finally, we compare the operation count for each of the three considered algorithms. The sequence of numbers for some operations results from the possible ways how "if" commands can be executed.

operations	**KOGUL**	**xLASV2**	**HYBRID**
$+,\,-$	$8, 9$	11	$8, 9, 13, 14$
$*,\,/$	$16, 17, 18$	18	$16, 17, 19, 20$
$\sqrt{\ }$	5	3	$5, 6$

We can see that all three algorithms have similar operation count.

Modern computers have computational units with long registers containing (by the IEEE standard) at least 78, but usually 80 or more bits. Hence the commands (and maybe all output data) of these algorithms will be computed in extended precision. Therefore, in practical computations the accuracy of the output data to all these algorithms might be alike.

However, using the obtained sharp estimates obtained for the hybrid algorithm, one can prove sharp relative accuracy bounds for the Kogbetliantz method for $n \times n$ triangular matrices. In [16] we have proved it for the case of scaled almost diagonal triangular matrices. The accuracy of that method for general triangular matrices, which numerical experiments clearly confirm, is the topic of our further research.

References

[1] J. P. CHARLIER, P. VAN DOOREN, *On Kogbetliantz's SVD algorithm in the presence of clusters*, Linear Algebra Appl. **95**(1987), 135–160.

[2] J. P. CHARLIER, M. VANBEGIN, P. VAN DOOREN, *On efficient implementation of Kogbetliantz's algorithm for computing the singular value decomposition*, Numer. Math. **52**(1988), 279–300.

[3] K. V. FERNANDO, *Linear convergence of the row–cyclic Jacobi and Kogbetliantz methods*, Numer. Math. **56**(1989), 71–91.

[4] G. E. FORSYTHE, P. HENRICI, *The cyclic Jacobi method for computing the principal values of a complex matrix*, Trans. Amer. Math. Soc. **94**(1960), 1–23.

[5] V. HARI, *On the quadratic convergence of Jacobi algorithms*, Radovi Matematički **2**(1986), 127–146.

[6] V. HARI, *On the quadratic convergence bounds of the serial singular value decomposition Jacobi methods for triangular matrices*, SIAM J. Sci. Stat. Comput. **10**(1989), 1076–1096.

[7] V. HARI, *On sharp quadratic convergence bounds for the serial Jacobi methods*, Numer. Math. **60**(1991), 375–406.

[8] V. Hari, J. Matejaš, *Quadratic convergence of scaled iterates by Kogbetliantz method*, Computing [Suppl.] **16**(2003), 83–105.
[9] V. Hari, J. Matejaš, *Accuracy of two SVD algorithms for 2 × 2 triangular matrices*, Applied Mathematics and Computation **210**(2009), 232–257.
[10] V. Hari, K. Veselić, *On Jacobi methods for singular value decompositions*, SIAM J. Sci. Stat. **8**(1987), 741–754.
[11] V. Hari, V. Zadelj-Martić, *Parallelizing the Kogbetliantz method: a first attempt*, Journal of Numerical Analysis, Industrial and Applied Mathematics, **2**(2007), 49–66.
[12] E. Kogbetliantz, *Diagonalization of general complex matrices as a new method for solution of linear equations*, Proc. Imtern. Congr. Math. Amsterdam **2**(1954), 356–357.
[13] E. Kogbetliantz, *Solutions of linear equations by diagonalization of coeficient matrices*, Quart. Appl. Math. **13**(1955), 123–132.
[14] J. Matejaš, *Accuracy of the Jacobi method on scaled diagonally dominant symmetric matrices*, SIAM J. Matrix Analysis and Applications **31**(2009), 133–153.
[15] J. Matejaš, V. Hari, *Scaled iterates by Kogbetliantz method*, in: *Proceedings of the 1st Conference on Applied Mathematics and Computations*, (M. Rogina, Ed.), Department of Mathematics, University of Zagreb, 2001, 1–20.
[16] J. Matejaš, V. Hari, *Accuracy of the Kogbetliantz method for scaled diagonally dominant triangular matrices*, to appear in Applied Mathematics and Computation.
[17] C. C. Paige, P. Van Dooren, *On the quadratic convergence of Kogbetliantz's algorithm for computing the singular value decomposition*, Linear Algebra and Its Applications **77**(1986), 301–313.
[18] V. V. Voevodin, *Cislennye metody linejnoj algebry*, Nauka, Moscow, 1966.