Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Applied Numerical Mathematics 47 (2003) 237–253

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

Modified ST algorithms and numerical experiments

Cosmo D. Santiago, Jin-Yun Yuan[1,*]

Departamento de Matemática, Universidade Federal do Paraná, Centro Politécnico,
CP: 19.081, CEP: 81.531-990, Curitiba, Paraná, Brazil

Abstract

Recently Golub and Yuan [BIT 42 (2002) 814] proposed the ST decomposition for matrices. However, its numerical stability has not been discussed so far. Here we present preliminary investigations on the numerical behavior of the ST decomposition. We also propose modifications (modified algorithm) to improve the algorithm's numerical stability. Numerical tests of the Golub–Yuan algorithm and our modified algorithm are given for some famous test matrices. All tests include comparisons with the LU (or Cholesky) decomposition without pivoting. These numerical tests indicate that the Golub–Yuan algorithm and its modified version possess reasonable numerical stability. In particular, the modified algorithm is stable for sparse matrices. Moreover, it is more stable than the Golub–Yuan algorithm in the case of dense matrices.

Keywords: ST decomposition; Golub–Yuan algorithm; Modified Golub–Yuan algorithm; LU factorization; Cholesky decomposition; Numerical stability

1. Introduction

Matrix decomposition is a very important tool in matrix computation and its applications. The LU decomposition is a well-known dynamic process to transform a matrix A into a product of triangular matrices, that is, A is numerically factored into lower and upper triangular matrices L and U such that

$$A = LU. \tag{1}$$

We can simplify the decomposition significantly by taking into account the symmetry in the case of symmetric and positive definite matrices. This special decomposition is called the *Cholesky*

[*] Corresponding author.
E-mail address: jin@mat.ufpr.br (J.-Y. Yuan).

[1] The work of this author was supported by CNPq, FINEP and FUNPAR, Brazil.

0168-9274/$30.00 © 2003 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/S0168-9274(03)00059-X

decomposition. The Cholesky decomposition constructs a lower triangular matrix L such that

$$A = LL^T. \tag{2}$$

Golub and Yuan [6] recently introduced a new matrix decomposition, called the ST decomposition (hereafter referred to as ST). It amounts to expressing a nonsingular matrix A as the product of a triangular matrix and a symmetric positive definite matrix. Specifically speaking, it finds lower triangular matrices T and L such that $TA = LL^T$. The numerical stability of the LU and Cholesky decompositions has been extensively studied. However, very little work has appeared on the numerical stability of the ST. It is our purpose in this work to investigate such properties. We test the ST on several types of matrices with different sizes and compare the results with the LU (or Cholesky) decomposition without pivoting, as we shall not discuss the pivoting process for the ST here. From our test results, ST presents reasonable numerical stability, specially for sparse matrices.

The remainder of this paper is arranged as follows. In Section 2, the main idea of the ST is reviewed. We propose some modifications of the original Golub–Yuan algorithm to improve its numerical stability. This new version is henceforth called the modified algorithm, or MST, in short. All tests include the modified algorithm alongside the other three algorithms already mentioned. In Sections 3 and 4, we present numerical test results for selected well-conditioned and ill-conditioned dense matrices respectively. In Section 5, deals with numerical tests for some special matrices. Short comments about the test matrices are included in the exposition. Finally, in the last section, we briefly discuss the numerical results.

2. The ST decomposition

The ST decomposition consists in the construction of two lower triangular matrices T and L. The matrix T transforms the matrix A into a symmetric and positive definite (SPD) matrix, that is,

$$TA = LL^T = S.$$

Theorem 2.1, established in [6], asserts the conditions for the existence of this decomposition.

Theorem 2.1. *For every nonsingular and nonsymmetric $n \times n$ matrix A, whose leading principal submatrices are nonsingular, there exist triangular matrices T and L such that*

$$TA = LL^T. \tag{3}$$

Properties of the matrices in (3) imply

$$A = \left[\begin{array}{c|c} A_k & a_{k+1} \\ \hline \tilde{a}_{k+1}^T & \alpha \end{array}\right], \quad T = \left[\begin{array}{c|c} T_k & 0 \\ \hline t_{k+1}^T & \beta \end{array}\right] \quad \text{and} \quad L = \left[\begin{array}{c|c} L_k & 0 \\ \hline l_{k+1}^T & \tau \end{array}\right],$$

where $\alpha \neq 0$, $\beta \neq 0$, $\tau \neq 0$, L_k and T_k are lower triangular, and A_k is nonsingular. It follows from (3) that

$$\begin{aligned} A_k &= T_k^{-1} L_k L_k^T, \\ l_{k+1} &= L_k^{-1} T_k a_{k+1}, \\ t_{k+1} &= T_k^T L_k^{-T} \left(l_{k+1} - \beta L_k^{-1} \tilde{a}_{k+1} \right), \\ \tau &= \sqrt{\beta(\alpha - \tilde{a}_{k+1}^T L_k^{-T} L_k^{-1} T_k a_{k+1})}. \end{aligned}$$

The above relations lead to the following algorithm:

Algorithm 2.1 (*Golub–Yuan ST*).

```
Set $t_{11}$ such that $t_{11}a_{11} > 0$ and $l_{11} = \sqrt{t_{11}a_{11}}$;
For $k = 1, \ldots, n-1$
    $l_{k+1} = L_k^{-1} T_k a_{k+1}$,
    $\hat{l}_{k+1} = L_k^{-1} \tilde{a}_{k+1}$,
    $s = a_{k+1,k+1} - \hat{l}_{k+1}^T l_{k+1}$,
    Choose $t_{k+1,k+1} \neq 0$ such that $\gamma = t_{k+1,k+1}s > 0$ (or large enough)
    $l_{k+1,k+1} = \sqrt{\gamma}$
    $t_{k+1} = T_k^T L_k^{-T} \big(l_{k+1} - t_{k+1,k+1}\hat{l}_{k+1}\big)$
End
```

Note that the diagonal elements $\beta \neq 0$ of the matrix T are arbitrary. Thus we can choose β such that $\beta(\alpha - \tilde{a}_{k+1}^T L_k^{-T} L_k^{-1} T_k a_{k+1}) > 0$. Also, L's diagonal elements τ should be larger than zero for all k. However sometimes the elements β or τ can be very small, leading to serious numerical instability. In Algorithm 2.1, the elements $t_{k+1,k+1}$ ($\beta = t_{k+1,k+1}$) are arbitrary, apart from the sign requirement. From the numerical point of view, we found that, for most dense matrices, the elements $l_{k+1,k+1}$ approach zero in Algorithm 2.1, which is one of the sources of instability. To avoid this difficulty, a possible alternative is to minimize the rounding errors by controlling the difference $s = a_{k+1,k+1} - \hat{l}_{k+1}^T l_{k+1}$ using some tolerance δ (here we choose $\delta = 10^{-18}$). Since the decomposition is not unique, we may choose values for $\beta = t_{k+1,k+1}$ (or $l_{k+1,k+1} = \sqrt{\gamma}$) that improve the stability. With such considerations in mind, we obtain the modified algorithm MST as follows. We shall not discuss the convergence of the modified algorithm here. This will be the subject of a future paper.

Algorithm 2.2 (*Modified ST*).

```
Set $t_{11}$ such that $t_{11}a_{11} > 0$, $l_{11} = \sqrt{t_{11}a_{11}}$, $\delta = 10^{-18}$ and choose, as initial guess $\eta = 1$;
For $k = 1, \ldots, n-1$
    $l_{k+1} = L_k^{-1} T_k a_{k+1}$,
    $\hat{l}_{k+1} = L_k^{-1} \tilde{a}_{k+1}^T$,
    $s = a_{k+1,k+1} - \hat{l}_{k+1}^T l_{k+1}$,
    if $|s| < \delta$  $t_{k+1,k+1} = 1$,  else $t_{k+1,k+1} = \operatorname{sign}(s)\eta$
    update $\eta$ such that $l_{k+1,k+1} > 0$
    $\gamma = t_{k+1,k+1}s$
    $l_{k+1,k+1} = \sqrt{\gamma}$
    $t_{k+1} = T_k^T L_k^{-T} (l_{k+1} - t_{k+1,k+1}\hat{l}_{k+1})$
End
```

Remark. In fact, our modification consists in fixing some special characteristics of the determination of $t_{k+1,k+1}$ for the Golub–Yuan algorithm. We also found that instead of $t_{k+1,k+1}$, the value of $l_{k+1,k+1} \neq 0$ ($\tau = l_{k+1,k+1}$) can be arbitrary. In the following sections, we shall test Algorithms 2.1 and 2.2 on dense and sparse matrices with some structure. Some test matrices are provided by Gregory and David (see [7]), others by the collection of test matrices "The Test Matrix Toolbox" [10]. The sparse matrices used in our tests come from the Harwell-Boeing collection. The ST algorithms are implemented using MATLAB.

The MATLAB M-file for LU decomposition was obtained from [4]. All tests were run on a single processor Intel Pentium III 700 MHz with 128 MB of RAM memory. CPU time, in seconds, reported in the experiments is the elapsed time. The relative error was measured according to

$$\operatorname{res}\bigl(T, L, L^T\bigr) = \frac{\|A - T^{-1}LL^T\|_F}{\|A\|_F},$$

for ST Algorithms 2.1 and 2.2 and

$$\operatorname{res}(L, U) = \frac{\|A - LU\|_F}{\|A\|_F},$$

for the LU decomposition without pivoting.

3. Well-conditioned dense matrices

3.1. Diagonally dominant matrix

The diagonally dominant matrices used were built as follows: all entries were generated randomly using the MATLAB function *randn* and then the value of the diagonal entry of each given row is set as the sum of all entries, in absolute values, of that row. The performance of the algorithms is summarized in Table 1 and Fig. 1. Here we set $\eta = \|L_{k+1}\|_2$. Note that the ST decomposition is stable for this type of matrix. The relative error of the modified algorithm is smaller than that of the Golub–Yuan algorithm.

3.2. Random matrix

We also considered random matrices generated by the MATLAB function *randn* because they are well-conditioned and always considered favorite test matrices. The test results and relative error curves are displayed in Table 2 and Fig. 2, respectively, with the choice $\eta = \|L_{k+1}\|_2$.

3.3. Symmetric positive definite matrix

In this batch of tests we used symmetric positive definite Moler matrices [8] defined by $A_n(\theta) = C_n(\theta)^T C_n(\theta)$, where $C_n(\theta)$ is unit upper triangular with all $c_{i,j} = \theta$ for $i \neq j$. We set $\theta = -2$ as in [2,3] and $\eta = \|L_{k+1}\|_\infty$ in Algorithm 2.2. The idea here is to control the elements $t_{k+1,k+1}$ in order

Table 1
Performance for diagonally dominant matrices

	LU		ST		MST
n	CPU time	$\operatorname{res}(L, U)$	CPU time	$\operatorname{res}(T, L, L^T)$	$\operatorname{res}(T, L, L^T)$
100	0.33	2.21e−16	0.33	4.66e−16	3.32e−16
200	1.65	2.71e−16	4.12	6.14e−16	3.59e−16
300	4.56	3.29e−16	16.3	8.26e−16	4.06e−16
400	12.1	3.86e−16	40.7	8.75e−16	4.49e−16
500	22.2	4.33e−16	66.7	1.02e−15	4.76e−16

Table 2
Performance for matrices with random entries

	LU		ST		MST
n	CPU time	res(L, U)	CPU time	res(T, L, L^T)	res(T, L, L^T)
100	0.28	2.45e−14	0.28	2.47e−08	5.86e−09
200	1.43	2.30e−13	3.52	2.49e−07	1.17e−07
300	4.28	4.20e−13	15	5.4e−07	2.19e−07
400	8.46	5.28e−13	35.2	1.02e−05	3.08e−06
500	32.7	6.53e−13	66.6	1.39e−05	4.98e−06

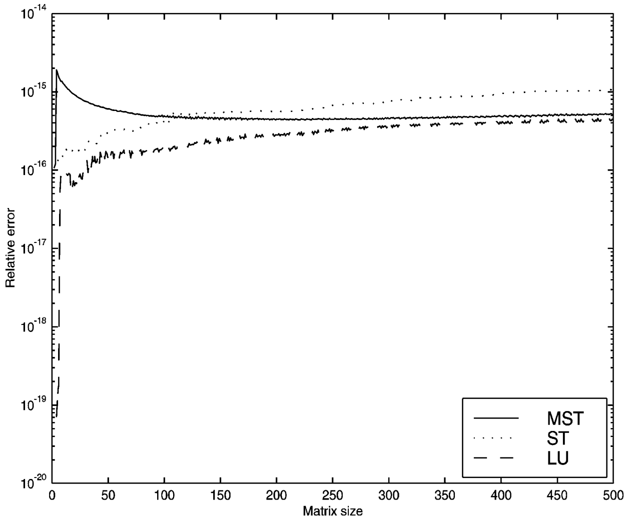

Fig. 1. Relative error for diagonally dominant matrices.

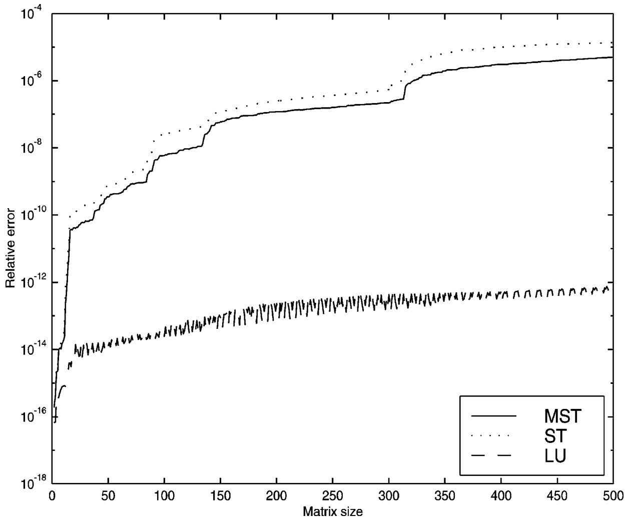

Fig. 2. Relative error for random matrices.

to obtain stability. We tested the modified algorithm and Golub–Yuan algorithm with several negative integer values for θ. In this case, T is the identity matrix, and L is unit triangular. The relative error obtained is zero and the CPU time is 66,7 seconds. Therefore, the decomposition for Moler matrices is stable for negative integer θ. When we choose the parameter $\theta > 0$ or non integer, the relative errors of the modified algorithm and Golub–Yuan algorithm are similar to that of the LU decomposition, but not better. The Cholesky method is stable, finding the associated decomposition with relative error zero in 49 seconds.

4. Ill-conditioned dense matrices

4.1. *Hilbert matrix*

The Hilbert matrix is one of the most famous ill-conditioned test matrices. Its condition number and numerical stability are very bad even for small n, as shown in Table 3. The condition number for $n = 20$, for instance, already reaches the impressive figure 1.0675e+019 [13].

It follows from the results in Table 4 that, for $n = 500$, the relative error is in the order of 0.000197 for the ST decomposition and 7.92e$-$17 for the LU decomposition, respectively. Although the relative error of the ST decomposition is bigger than that of the LU decomposition, the ST decomposition improves the condition number of the factors obtained. For instance, $\text{cond}(L) = 3.12\text{e}+08$ in the case of ST, and $\text{cond}(L) = 2.08\text{e}+12$ and $\text{cond}(U) = 8.16\text{e}+20$ for the LU decomposition. This means that the ST decomposition produces better conditioned triangular factors. Numerical results are presented in Table 4 and Fig. 3 with the choices $\eta = \|L_{k+1}\|/2k$, $k = 1, 2, \ldots, n-1$. The modified algorithm improves the numerical stability significantly.

4.2. *Lotkin matrix*

The Lotkin matrix is a special case of the Hilbert matrix whose first row consists only of ones. The matrix is ill-conditioned and nonsymmetric, with many negative eigenvalues whose magnitude is small. Its inverse has integer entries [9]. The behavior of the ST algorithms on Lotkin matrices are given in Table 5 and Fig. 4. The updating of the elements $t_{k+1,k+1}$, used $\eta = \|L_{k+1}\|/2k$. Note that, in this case, our modification improves the relative error significantly compared with the Golub–Yuan algorithm. From these results, it follows that the LU decomposition is still better. But the results obtained for the modified algorithm are reasonably acceptable for real applications.

Table 3
Condition number of the Hilbert matrix

n	2	3	4	5	20
$\text{cond}(H_n)$	19.2815	524.0568	1.5514e+04	4.7661e+05	1.0675e+19

Table 4
Performance of the methods for Hilbert matrices

	LU		ST		MST
n	CPU time	res(L, U)	CPU time	res(T, L, L^T)	res(T, L, L^T)
100	0.38	5.83e−17	0.28	0.0452	3.31e−07
200	1.37	6.66e−17	3.57	379	3.43e−06
300	6.7	7.2e−17	14.1	5.92e+03	5.04e−05
400	14.6	7.6e−17	35.3	1.51e+04	0.000126
500	29.3	7.92e−17	65.4	1.69e+05	0.000197

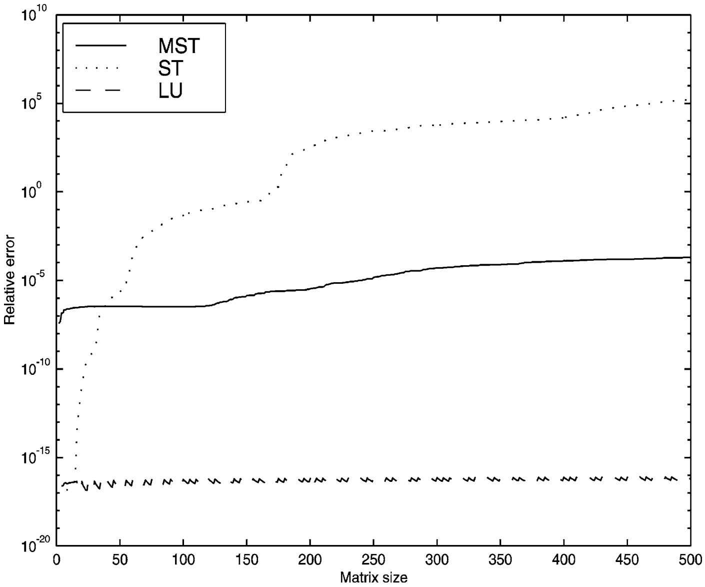

Fig. 3. Relative residual for Hilbert matrices.

5. Special matrices

For the sparse matrices (Poisson, Wathen, Dorr and Toeplitz Tridiagonal), we fix the value $\eta = 2$ in the determination of the diagonal elements of T. Therefore, all elements $t_{k+1,k+1}$ are equal to 2 except t_{11} and t_{22}. It follows from our numerical experiments that the sparsity of the ST decomposition is the same as that of the LU decomposition for all test sparse matrices in this section. Hence, we shall not give the details of their sparsity structure.

5.1. Poisson's matrix

Here we consider the block tridiagonal (sparse) matrix of order n^2 resulting from discretizing *Poisson's equation*

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = f(x, y),$$

Table 5
Relative error for Lotkin matrices

	LU		ST		MST
n	CPU time	res(L, U)	CPU time	res(T, L, L^T)	res(T, L, L^T)
100	0.11	3.92e−17	0.29	0.000797	6.7e−07
200	1.59	4.04e−17	3.46	0.445	1.07e−06
300	5.99	4.09e−17	13.6	461	2.28e−06
400	13.8	4.15e−17	33.7	4.29e+04	1.04e−05
500	26.8	4.18e−17	66.9	2.36e+06	6.31e−05

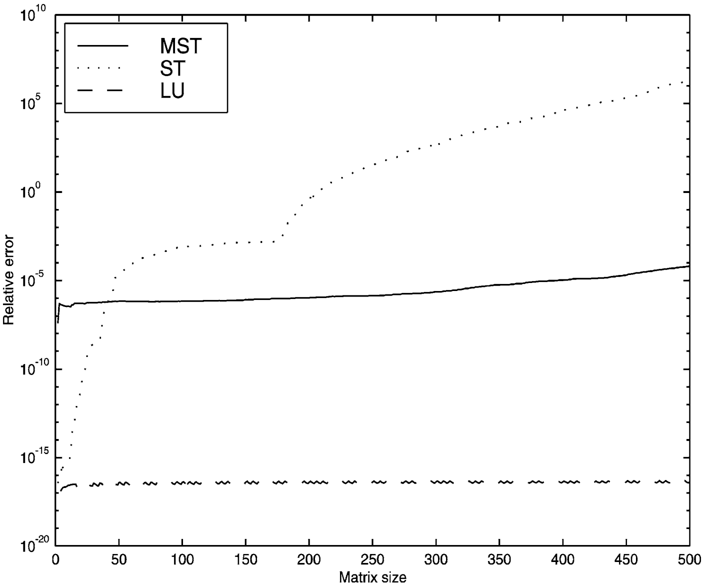

Fig. 4. Relative error for Lotkin matrices.

where $(x, y) \in \Omega = (0, 1) \times (0, 1)$. The five point finite difference method generates a well-conditioned block tridiagonal SPD coefficient matrix [1, Chapter 6] and [4,5,11]. Test results are given in Table 6 and Fig. 5. Note that the method is stable for this type of matrices. It follows from the results in Table 6 that the relative error of two ST algorithms is smaller than that of the LU decomposition for this class of test matrices.

5.2. Wathen matrix

Wathen's matrix is a matrix (sparse, random entries) arising from two-dimensional finite element discretization on an uniform Cartesian mesh with l nodes in the x-direction and m nodes in the y-direction, where $n = 3lm + 2l + 2m + 1$. A is the precisely consistent mass matrix for a regular l by m grid of 8-node (serendipity) elements in the plane. In particular, with $D = \text{diag}(A)$, the eigenvalues of $D^{-1}A$ lie in $(0.25, 4.5)$ for positive integers l and m [15]. In our tests the matrix A is constructed with $l = m$. The performance of the algorithms is given in Table 7 and in Fig. 6. Here the ST algorithms obtain

Table 6
Relative error for Poisson's matrices

	LU		ST		MST
n	CPU time	res(L, U)	CPU time	res(T, L, L^T)	res(T, L, L^T)
10	0.28	9.17e−17	0.28	1.35e−17	1.22e−16
14	2.37	1.31e−16	3.24	1.25e−16	1.33e−16
18	11	1.56e−16	17.7	1.36e−16	1.41e−16
22	37.3	1.65e−16	58.9	1.36e−16	1.41e−16
23	48.9	1.64e−16	73.5	1.48e−16	1.44e−16

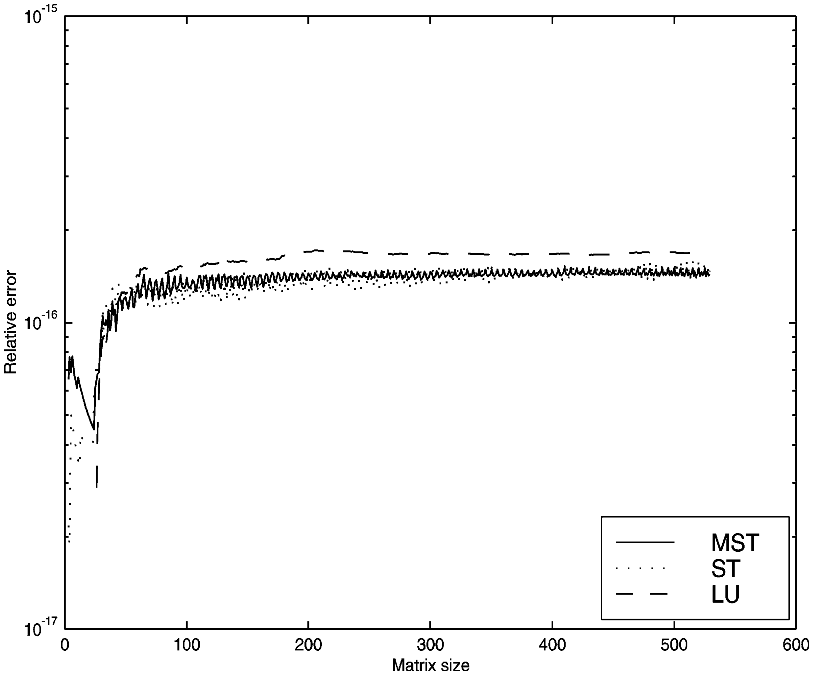

Fig. 5. Curve of residual error for Poisson's matrices.

better results than the LU decomposition. For the Cholesky decomposition, the relative error and CPU time are 1.51e−016 and 0.66 seconds, respectively. Note that the relative error of the modified algorithm is smaller than that of Cholesky decomposition.

5.3. CDDE1—Matrix Market

The CDDE1 test matrix comes from the following constant-coefficient convection diffusion equation, which is widely used for testing and analyzing numerical methods for the solution of linear system of equations. The equation is

$$\begin{aligned} -\Delta u + 2p_1u_x + 2p_2u_y - p_3u &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega, \end{aligned}$$

Table 7
Relative error for Wathen matrices

	LU		ST		MST
n	CPU time	res(L,U)	CPU time	res(T,L,L^T)	res(T,L,L^T)
96	0.22	1.42e−16	0.22	1.16e−16	1.02e−16
225	1.87	1.32e−16	5.22	1.37e−16	1.27e−16
341	5.5	1.38e−16	20.7	1.40e−16	1.32e−16
481	13.4	1.34e−16	58.9	1.61e−16	1.42e−16
560	23.4	1.27e−16	93.2	1.48e−16	1.30e−16

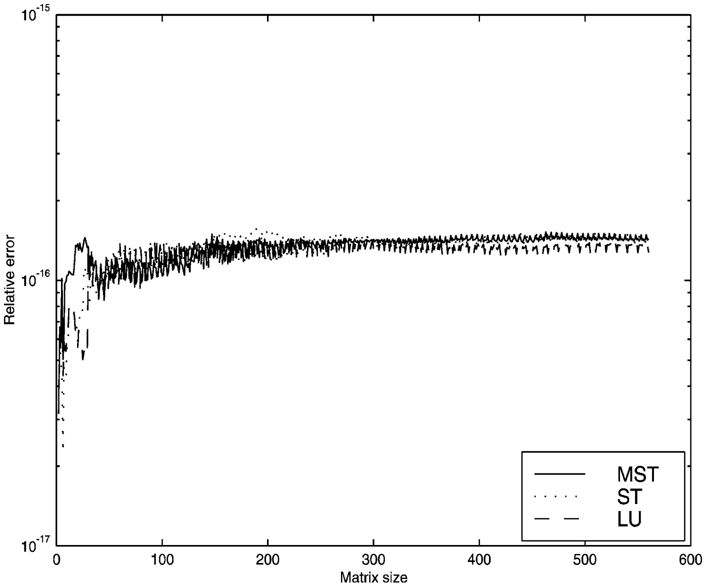

Fig. 6. Relative error for Wathen matrices.

where p_1, p_2 e p_3 are positive constants. The discretization of $-\Delta u$ by finite difference schemes with a 5-point stencil on a uniform $m \times m$ grid gives a sparse linear system of equations

$$Au = b,$$

where A is of order m^2, and u, b are m^2-dimensional vectors. Centered differences are used for the first derivatives. If the grid points are numbered with the row-wise natural ordering, then A is a block tridiagonal matrix of the form

$$A = \begin{bmatrix} T & (\beta+1)I & & & \\ (-\beta+1)I & T & (\beta+1)I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & (\beta+1)I \\ & & & (-\beta+1)I & T \end{bmatrix}$$

Table 8
Relative error for CDDE1—Matrix Market

	LU		ST		MST
n	CPU time	res(L, U)	CPU time	res(T, L, L^T)	res(T, L, L^T)
100	0.44	1.53e−16	0.44	1.52e−16	1.49e−16
200	3.18	1.82e−16	3.46	1.84e−16	1.78e−16
300	10.6	1.88e−16	13.3	1.94e−16	1.86e−16
400	24.3	1.87e−16	33.2	1.97e−16	1.91e−16
500	46	1.93e−16	66.2	1.97e−16	1.97e−16

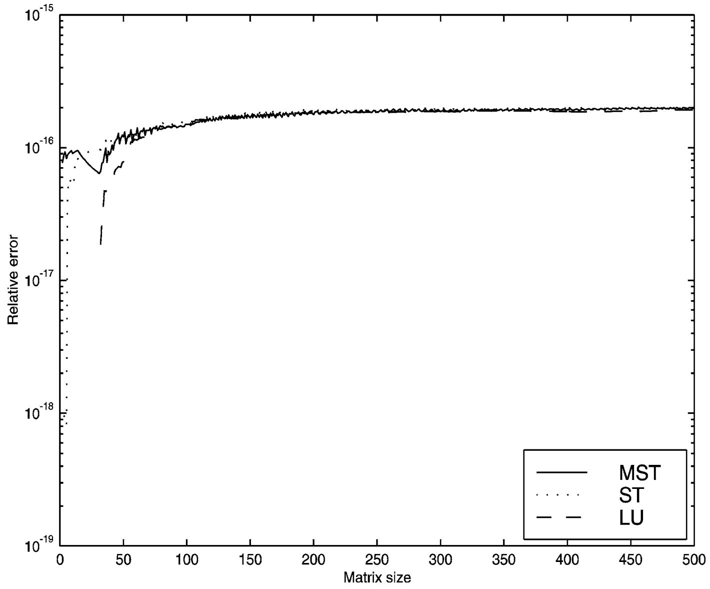

Fig. 7. Relative error for CDDE1—Matrix Market.

with

$$T = \begin{bmatrix} 4-\sigma & \gamma-1 & & & \\ -\gamma-1 & 4-\sigma & \gamma-1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma-1 \\ & & & -\gamma-1 & 4-\sigma \end{bmatrix},$$

where $\beta = p_1 h$, $\gamma = p_2 h$, $\sigma = p_3 h^2$ and $h = 1/(n+1)$. This real nonsymmetric matrix is extracted from the Harwell-Boeing Collection [12]. In our experiments, A has 2404 nonzero elements for $n = 500$, with $p_1 = 1$, $p_2 = 2$, $p_3 = 30$. The two ST algorithms obtain the same results as the LU decomposition. This illustrates the efficiency of the method for these structured matrices. The results of the methods are shown in Table 8 and Fig. 7, respectively.

5.4. Diagonally dominant Dorr matrix

The Dorr matrix $D_n(\alpha)$ [2,3,8,9], is a nonsymmetric, row diagonally dominant, tridiagonal M-matrix[2]. $D_n(\alpha)$ has diagonal dominance factors

$$\gamma_i = \begin{cases} |d_i| - |d_{i,i-1} - d_{i,i+1}| = (n+1)^2\alpha & \text{for } i = 1, 2, \ldots, n, \\ 0 & \text{otherwise}. \end{cases}$$

This matrix is ill-conditioned for small values of the parameter $\alpha > 0$. Fig. 8 illustrates the relative error for Dorr matrices. Table 9 gives the numerical behavior of the algorithms. The LU decomposition works very well for these matrices. Here we use $\eta = \|L_{k+1}\|$ in the formula for $t_{k+1,k+1}$.

Table 9
Convergence of the decomposition for Dorr matrices

	LU		ST		MST
n	CPU time	res(L, U)	CPU time	res(T, L, L^T)	res(T, L, L^T)
100	0.27	0	0.28	3.20e−13	3.97e−13
200	1.43	0	3.4	7.40e−13	7.04e−13
300	4.17	0	13.8	8.33e−13	8.85e−13
400	8.9	0	41.4	9.58e−13	8.72e−13
500	16.2	0	69.2	9.52e−13	9.49e−13

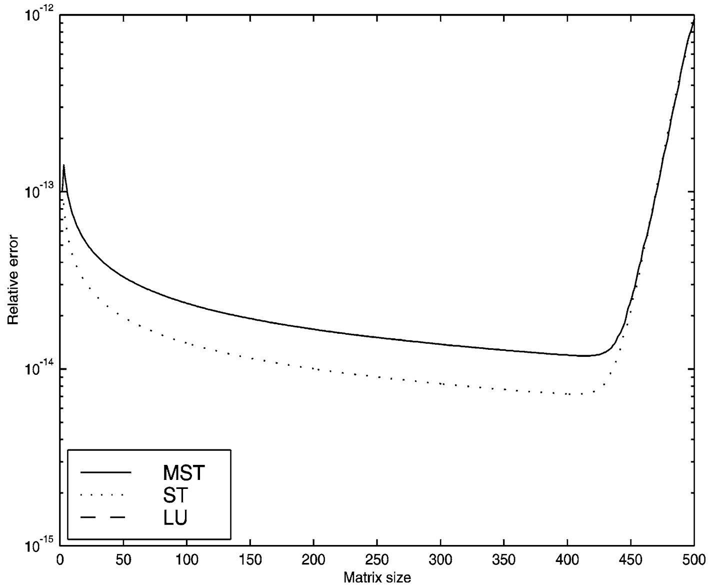

Fig. 8. Curve of relative error of the Dorr matrices.

[2] We say that a matrix is a M-matrix if $a_{ij} \leqslant 0$ for all $i \neq j$ and all the eigenvalues of A have nonnegative real part. Equivalently, a matrix is a M-matrix if $a_{ij} \leqslant 0$ for all $i \neq j$ and all the elements of A^{-1} are nonnegative.

5.5. *Prolate matrix*

The Prolate matrix $A(\alpha)$ is a symmetrically ill-conditioned Toeplitz matrix given by

$$a(1) = 2\alpha \quad \text{and} \quad a(2:k+1) = \frac{\sin(2\pi\alpha(1:k))}{\pi(1:k)}, \quad k = 2, \ldots, n-1,$$

where a_k is the first row of A. This is the typical case arising in signal processing applications. It was first studied by Slepian in the 1950s at Bell Labs [14]. With $0 < \alpha < 1/2$, A is positive definite with distinct eigenvalues in $(0,1)$, which tend to cluster around 0 and 1. For $\alpha = 0.125$, the Prolate matrix is very ill-conditioned ($\mathrm{cond}(A_{500}(0.125)) \approx 2.20\text{e}{+}18$). The diagonal elements of the matrix T are updated using $\eta = \|L_{k+1}\|$. The numerical results of the ST algorithms are given in Table 10 for Prolate matrices. Fig. 9 shows the relative error curve. Although the modified algorithm and the Golub–Yuan algorithm are not good for these matrices, the modified algorithm improves the numerical stability.

Table 10
Relative error for Prolate matrix

	LU		ST		MST
n	CPU time	res(L,U)	CPU time	res(T,L,L^T)	res(T,L,L^T)
100	0.27	7.21e−16	0.28	0.353	2.23e−05
200	1.59	1.02e−15	3.52	0.794	0.000202
300	4.11	9.89e−15	14	5.02	0.0003
400	8.78	9.57e−15	33.6	22.1	0.000942
500	15.3	9.13e−15	66.6	31.5	0.000969

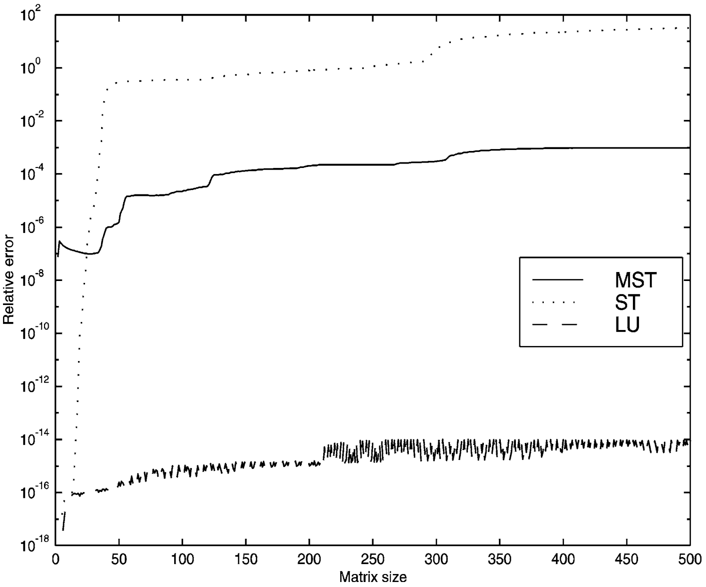

Fig. 9. Curve of relative error for Prolate matrix.

5.6. Circulant matrix

Circulant matrices are special Toeplitz matrices where the diagonals wrap around. Our test matrices are given by

$$A=\begin{bmatrix} 1 & 2 & \dots & \dots & n \\ n & 1 & 2 & \dots & n-1 \\ n-1 & n & 1 & \ddots & n-2 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 2 & 3 & \dots & n & 1 \end{bmatrix},$$

and the condition number is $\mathrm{cond}(A_n)=n+1$. The numerical results for the ST algorithms are given in Table 11 and Fig. 10.

Table 11
Relative error for Circulant matrix

	LU		ST		MST
n	CPU time	$\mathrm{res}(L,U)$	CPU time	$\mathrm{res}(T,L,L^T)$	$\mathrm{res}(T,L,L^T)$
100	0.16	2.95e−15	0.28	2.16e−12	3.43e−13
200	1.59	6.01e−15	3.51	1.17e−11	1.56e−12
300	6.42	9.22e−15	14.1	3.41e−11	4.45e−12
400	15.4	1.23e−14	35	6.87e−11	9.37e−12
500	30.5	1.57e−14	67.7	1.23e−10	1.58e−11

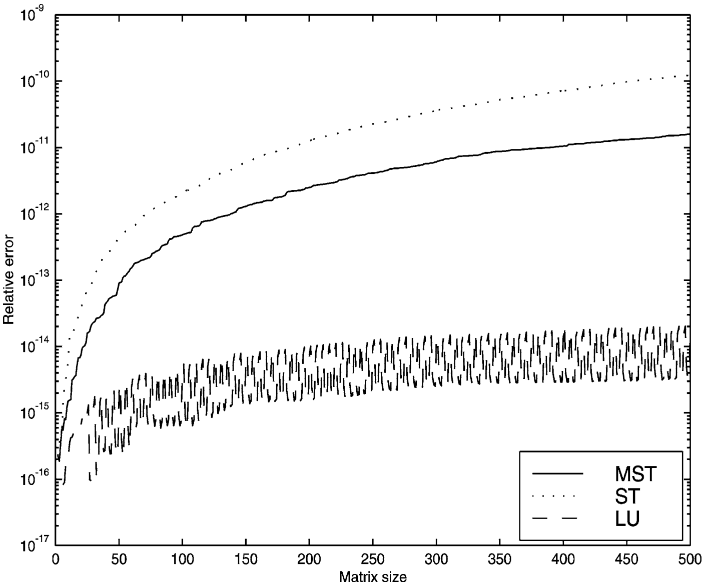

Fig. 10. Residual error for Circulant matrix.

5.7. Toeplitz tridiagonal matrix

Consider the tridiagonal Toeplitz matrix

$$T=\begin{bmatrix} d & -c & & \\ -c & d & \ddots & \\ & \ddots & \ddots & -c \\ & & -c & d \end{bmatrix}, \tag{4}$$

where $c=-1$, $d=2$ and $e=3$. The ST algorithms yield very good results, although the LU decomposition still has smaller error. For the symmetric positive definite Toeplitz matrix $T_{500}(-1,2,-1)$, the relative errors are 1.22e−16 in 0 seconds, 6.18e−17 in 68.9 seconds and 0 in 26.5 seconds, respectively, for the Cholesky[3], ST and LU decompositions. The performance of the algorithms is given

Table 12
Relative error for Tridiagonal Toeplitz matrix (sparse)

	LU		ST		MST
n	CPU time	res(L,U)	CPU time	res(T,L,L^T)	res(T,L,L^T)
100	0.17	1.03e−17	0.27	6.97e−16	1.15e−15
200	1.54	7.28e−18	3.46	5.69e−16	1.7e−15
300	5.54	5.94e−18	13.9	5.19e−16	2.13e−15
400	13.4	5.14e−18	34.7	4.93e−16	2.51e−15
500	26.5	4.6e−18	67.3	4.76e−17	2.79e−15

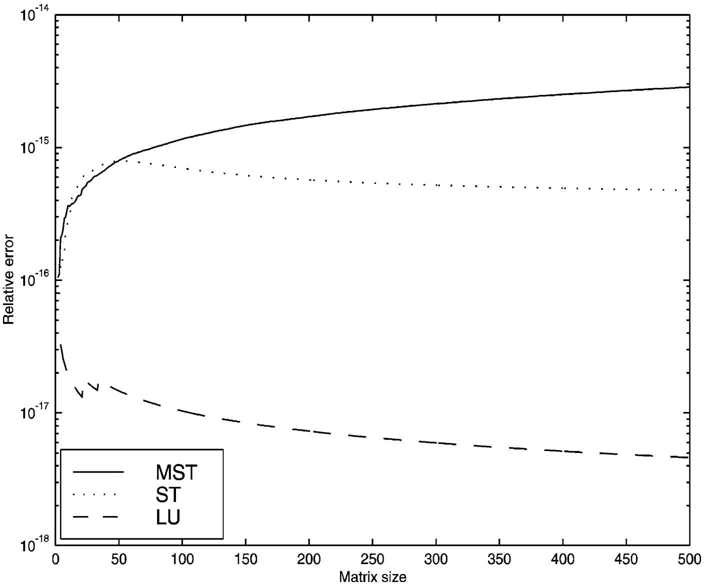

Fig. 11. Curve of relative error for Tridiagonal Toeplitz matrix (sparse).

[3] For Cholesky decomposition, the MATLAB function *chol* was used.

in Table 12 and Fig. 11. Note that for this matrix our modification has a poorer performance than the Golub–Yuan algorithm, which compares favorably with the LU decomposition.

6. Conclusions

The numerical tests in this paper indicate the efficiency and numerical stability of the ST decomposition. For the sake of numerical stability, we introduced a modification in the Golub–Yuan algorithm. We tested both algorithms, as well as the LU and Cholesky decompositions, on several types of matrices. Algorithms ST and MST do not exhibit better numerical stability than the LU decomposition for most of our test matrices. In some cases, especially when the matrix is sparse and structured, as in Section 5, the algorithms can produce errors as small as those corresponding to the LU (or Cholesky) decomposition. According to our tests, the modified algorithm improves the numerical stability of the Golub–Yuan Algorithm, specially for very ill-conditioned matrices such as Hilbert, Lotkin and Prolate matrices. Our numerical experiments also suggest that the ST decomposition is stable, in some sense, for many real applications, although it is not as good as the LU (or Cholesky) decomposition for some types of matrices. Of course, the decomposition still needs further improvements, for instance to attain better numerical stability, which is one of our future research topics.

Acknowledgements

The authors thank Professor Plamen Yalamov for his suggestion on the issue of the paper. The first author would like to give his thanks to Mr. Celso Cordeiro for his helpful discussions. Both authors earnestly thank the referees for their constructive comments and helpful suggestions which greatly improved the quality of this paper.

References

[1] B.N. Datta, Numerical Linear Algebra and Applications, Brooks and Cole, Pacific Grove, CA, 1994.

[2] J.W. Demmel, N.J. Higham, Stability of block algorithms with fast level-3 BLAS, ACM Trans. Math. Softw. 18 (1992) 274–291.

[3] J.W. Demmel, N.J. Higham, R.S. Schreiber, Stability of block LU factorization, Numer. Linear Algebra Appl. 2 (1995) 173–190.

[4] L.V. Fausett, Applied Numerical Analysis Using Matlab, Prentice-Hall, Englewood Cliffs, NJ, 1999.

[5] G.H. Golub, C.F. Van Loan, Matrix Computation, 3rd Edition, Johns Hopkins University Press, Baltimore, MA, 1996.

[6] G.H. Golub, J.-Y. Yuan, ST: Symmetric-triangular decomposition and its applications. Part I: Theorems and algorithms, BIT 42 (2002) 814–822.

[7] R.T. Gregory, D.L. Karney, Collection of Matrices for Testing Computational Algorithms, Wiley Interscience, New York, 1969.

[8] N.J. Higham, Algorithm 694: A collection of test matrices in MATLAB, ACM Trans. Math. Softw. 17 (1991) 289–305.

[9] N.J. Higham, The Test Matrix Toolbox for MATLAB, Numerical Analysis Report No. 237, Manchester Centre for Computational Mathematics, Manchester, December 1993.

[10] N.J. Higham, Accuracy and Stability of Numerical Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.

[11] C. Keller, N.I.M. Gould, A.J. Wathen, Constraint preconditioning for indefinite linear system, Rutherford Appleton Laboratory, Chilton, Didcot, February 1999.

[12] I.S. Duff, R.G. Grimes, J.G. Lewis, Users' Guide for the Harwell-Boeing sparse matrix collection (release 1), Research and Technology Division, Boeing Computer Services, Report RAL-92-086, Atlas Centre, Rutherford Appleton Laboratory, Chilton, Didcot, December 1992.

[13] J. Todd, The condition of the finite segments of the Hilbert matrix, in: O. Taussky (Ed.), Contributions to the Solution of Systems of Linear Equations in the Determination of Eigenvalue, in: Applied Mathematics Series, Vol. 39, National Bureau of Standards, United States Department of Commerce, Washington, DC, 1954, pp. 106–116.

[14] J.M. Varah, The prolate matrix, Linear Algebra Appl. 187 (1993) 269–278.

[15] A.J. Wathen, Realistic eigenvalue bounds for the Galerkin mass matrix, IMA J. Numer. Anal. 7 (1987) 449–457.