COSMO DAMIÃO SANTIAGO

NUMERICAL EXPERIMENTS FOR S & T DECOMPOSITION

CURITIBA
2001

COSMO DAMIÃO SANTIAGO

Numerical Experiments for S&T Decomposition

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciências, pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Setores de Ciências Exatas e de Tecnologia da Universidade Federal do Paraná.

Orientador: Prof. Dr. Jin Yun Yuan.

CURITIBA

2001

COSMO DAMIÃO SANTIAGO

Numerical Experiments for S&T decomposition

Dissertação aprovada como requisito parcial à obtenção do grau de Mestre em Ciências, M.Sc. do Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Setores de Ciências Exatas e de Tecnologia da Universidade Federal do Paraná, pela seguinte banca examinadora:

Coordenador do Curso:

Prof. Dra. Maria Teresinha Arns Steiner.

Orientador:

Prof. Dr. Jin Yun Yuan.
Depto. de Matemática - UFPR

Prof. Dr. Jiahong Yin
Depto. de Matemática - UFPR

Prof. Dra. Yumei Wu
Instituto de Matemática - UFRJ

Curitiba, agosto de 2001.

Paraná - Brasil

For my beloved daughter,
Ana Carolina.
The God always guards her.

Now he that planteth and he that watereth are one: and every man shall receive his own reward according to his own labour.

(Coríntios, 3:8)

Acknowledgements

I would like to thank Professor Jin Yun Yuan, my advisor, for giving me the direction, motivation and many suggestions in the area of numerical analysis, and of course, constant support during my master program. I am also thankful to my friends, Mr. Celso Cordeiro who was always willing to discuss with me about implementations of the S&T algorithm presented in this dissertation, Mr. Ricardo Zanardini, Ms. Waléria Cecílio and Ms. Ingrid Milleo who was giving me valuable suggestions in the beginning of this research and some joint works during whole course.

I like to give my special thanks to Dr. Plamen Yalamov for introducing me to work on the numerical stability of the S&T decomposition which is this dissertation.

Of course, I also thank all employees, specially Mr. Emerson de Castro Firmo da Silva, ex-president of the Sindicato dos Jornalistas Profissionais do Paraná, where I'm working, for giving me all support during my course. A very special gratitude is given to Professor Beatriz Perim de Barros (in memory) for introducing me to the first step of Matlab. I thank Professor Jiahong Yin and Professor Yumei Wu for helping me to correct my English, too.

I express my most grateful thanks to all the individuals who gave me help in different ways.

Contents

Acknowledgements vi

List of Tables ix

Abstract 1

Resumo 2

1 Introduction 3

1.1 *LU* decomposition 4

1.2 Cholesky decomposition 5

1.3 Sparse matrices 5

2 S&T decomposition 8

2.1 Main idea 8

2.2 Algorithms 11

2.3 Test matrices 14

3 Well-Conditioned dense matrices 17

3.1 Diagonally dominant matrix 17

3.2 Random matrix 19

3.3 Symmetric positive definite matrix 19

4 Ill-conditioned dense matrices **21**

4.1 Hilbert matrix 21

4.2 Lotkin matrix 23

5 Special matrices **25**

5.1 Sparse matrices 25

5.1.1 Poisson's matrix 25

5.1.2 Wathen matrix 27

5.1.3 CDDE1 - Matrix Market 28

5.1.4 Diagonally dominant Dorr matrix 31

5.2 Toeplitz Matrices 32

5.2.1 Prolate matrix 33

5.2.2 Circulant matrix 35

5.2.3 Toeplitz tridiagonal matrix 36

6 Conclusions and future work **40**

A Implemetation **42**

Bibliography **45**

List of Tables

x

3.1 Performance for diagonally dominant matrix 18
3.2 Performance for matrix with random entries 19

4.1 Condition number of the Hilbert matrix 21
4.2 Performance of the methods for Hilbert matrix 22
4.3 Relative error for Lotkin matrix 24

5.1 Relative error for Poisson's matrix 26
5.2 Relative error for Wathen matrix 28
5.3 Relative error for CDDE1 - Matrix Market 31
5.4 Convergence of the decomposition for Dorr matrix 34
5.5 Relative error for Prolate matrix 34
5.6 Relative error for Circulant matrix 37
5.7 Relative error for Tridiagonal Toeplitz matrix (sparse) 39

List of Figures

1.1 Sparsity patterns of the nonzero elements of the factor L of S&T decomposition, L and U of LU decomposition for Wathen's matrix . . . 6

2.1 **Manner of S&T decomposition** 15

3.1 Relative error for diagonally dominant matrix 18
3.2 Relative error for Random matrix 20

4.1 Relative residual for for Hilbert matrix 23
4.2 Relative error for Lotkin matrix 24

5.1 Curve of residual error for Poisson's matrix 26
5.2 Sparsity pattern of the Poisson's matrix. 27
5.3 Relative error for Wathen's matrix 29
5.4 Sparsity pattern of the Wathen matrix. 30
5.5 Relative error for CDDE1 - Matrix Market 32
5.6 Sparsity pattern of the CDDE1 - Matrix Market 33
5.7 Curve of relative error of the Dorr matrix 35
5.8 Curve of relative error for Prolate Matrix 36
5.9 Residual error for Circulant matrix 38
5.10 Curve of relative error for Tridiagonal Toeplitz matrix (sparse) 39

Abstract

The new decomposition, *Symmetric and Triangular Decomposition (S&T)*, has been proposed by Golub and Yuan in [12], with the objective of doing decomposition for nonsingular and nonsymmetric matrices. From this decomposition, every nonsingular matrix can be presented by a product of one symmetric matrix and one triangular matrix. Furthermore, the symmetric matrix in this decomposition can be positive definite. They have proposed two numerical algorithms with the same feature for the decomposition. For the sake of numerical stability, we modify the Golub-Yuan algorithm here. We do numerical tests for Golub-Yuan algorithm and our modified algorithm. We display here the numerical performance of one of these algorithms for some famous test matrices. All tests are compared with LU decomposition without pivoting. For the symmetric matrices, the results are compared also with Cholesky decomposition. To compare the results, we emphasize some of the most common matrices recommended by Gregory and Karney in [14], and also by Nicholas in [17, 18].

It follows from our numerical tests that the modified S&T algorithm presented here is stable for sparse matrices whose leading principal submatrices are nonsingular. For dense matrices our modifications give some results much better than original Golub-Yuan algorithm.

Resumo

A nova decomposição, *Decomposição Simétrica e Triangular (S&T)*, foi proposto por Golub e Yuan em [12], com o objetivo de fazer decomposição em matrizes nonsingular e não simétrica. Desta decomposição, toda matriz nonsingular pode ser apresentada pelo produto de uma matriz simétrica e uma matriz triangular. Além disso, a matriz simétrica nesta decomposição pode ser positiva definida. Eles propuseram dois algoritmos numéricos com as mesmas características para fazer esta decomposição. Por causa da instabilidade numérica, nós fizemos modificações em um dos algoritmos. Fizemos vários testes numéricos para o algoritmo do Golub-Yuan e nosso algoritmo modificado. Nós mostramos aqui o desempenho numérico de um destes algoritmos para alguns matrizes testes famosas. Todos os testes foram comparados com a decomposição LU sem pivoteamento. Para as matrizes simétricas, os resultados foram comparados também com a decomposição Cholesky. Para comparar os resultados, nós utilizamos algumas das matrizes mais comuns recomendados por Gregory e Karney em [14], e também por Nicholas em [17, 18].

Os nossos experimentos numéricos mostram que o algoritmo S&T modificado é estável para matrizes esparsas cujos submatrizes principais são nonsingular. Para matrizes densas nossas modificações apresentaram resultados melhores que o Algoritmo proposto por Golub and Yuan.

Chapter 1

Introduction

Our purpose in this dissertation is to investigate the numerical behaviour of the new decomposition for nonsingular, nonsymmetric and especially ill-conditioned matrices, as follows

$$TA = S = LL^T, \tag{1.1}$$

where T and L are the lower triangular matrices and S is the symmetric and positive definite matrix. The transformation (1.1) has the focus in the solution of the linear system

$$Ax = b, \tag{1.2}$$

where A is a $n \times n$ matrix, b is a given column vector of size n and x is the solution, which can be found by iterative or direct method.

This method, (Symmetric and Triangular Decomposition - S&T), has been proposed by Golub and Yuan in [12, 13]. We propose a modified algorithm and compare the modified S&T algorithm with original Golub-Yuan S&T algorithm and LU decomposition without pivoting, and for particular case of symmetric matrix, with the Cholesky decomposition without pivoting, because they are well-known, similar, and very important in several scientific and engineering applications.

The S&T method was created by the idea of transforming a matrix A in a symmetric positive definite matrix (1.1). It is well-known that a symmetric and positive

defined matrix is very nice for matrix computation and has many advantages. In many applications, it is easy to treat with symmetric or symmetric and positive definite systems.

When A is symmetric and positive definite, the Conjugate Gradient (CG) method [15] is an effective and widely used technique. However, when A is nonsymmetric, to find the solution of the linear system (1.2) by iterative methods is considerably more difficult. The convergence behavior of the iterative method depends on the spectral properties of the linear system. For difficult problems, the algorithms may not converge due to accumulated round-off errors [29, pp. 49]. These things motivate us to analyze the numerical stability of the S&T decomposition.

1.1 LU decomposition

The LU decomposition is the most common used method for the matrix decomposition. It is often desirable to adopt a different perspective on processing the matrix A. The LU Decomposition is a dynamic process of transforming the matrix A in the product of two triangular special matrices with very nice properties. That is, A can be numerically factored into two separate triangular matrices, one is lower triangular, L, with unit diagonal elements and an other upper triangular, U, that is

$$A = LU. \tag{1.3}$$

The decomposition of matrices in triangular matrices takes advantages because the triangular matrices have important properties in linear algebra that minimize so much computational cost as the number of operations required by machines in the resolution of the linear system (1.1). This method, however, is unstable for arbitrary matrices. For example, it leads to the breakdown when the diagonal elements of the matrix are near to zero. Our work here is only dealt with numerical behaviour of the S&T decomposition without pivoting. For example, from our tests our work is more useful for totally positive[1] matrices that the exact triangular factors have only

[1]Totally positive matrices are defined as those for which the determinant of every submatrix is

positive entries, considering that pivoting's strategy for S&T decomposition will be our future research issue.

1.2 Cholesky decomposition

When a matrix is symmetric, we can simplify the calculations of the decomposition LU significantly, by taking into advantages of the symmetry. Numerically, the symmetric positive definite matrices are rather special, which occur quite frequently in some applications, so their special factorizations is called *Cholesky decomposition.* Instead of seeking arbitrary lower and upper triangular factors L and U, Cholesky's decomposition constructs a lower triangular matrix L, and L^T can itself serve as the upper triangular part. In other words, we replace equation (1.3) by

$$A = LL^T. \tag{1.4}$$

This factorization is sometimes referred to as "taking the square root" of the matrix A.

The Cholesky algorithm requires about $n^3/6$ executions of the inner loop (consisting of one multiply and one subtract), half of the number of executions required to do the same work by using LU decomposition.

1.3 Sparse matrices

The sparse matrices arise frequently in the discretization of partial differential equations by finite elements and finite difference methods (see, [5, Chapter 6]). Most of the authors consider a sparse matrix whose nonzero elements consist of a relatively small number (about 10%) of total elements. The theory of sparse matrices is widely used and we will not give details here (for more information see [3]). For special sparse matrices such as tridiagonal matrix, pentadiagonal matrix, block matrix etc, the S&T

positive.

decomposition presents better results, (see Chapter 5.1). Since the leading principal submatrices are nonsingular, we can decompose the matrix A in the form (1.1) without harming the sparsity of the matrix too much. We take some test matrices from the Harwell-Boeing Sparse Matrix Collection, which has excellent test sparse matrices, largely drawn from practical problems (see, [22]). Figure (1.1) illustrates the sparsity patterns for matrix L, when the S&T method is applied for Wathen's matrix, see the results in the Chapter 5.2.

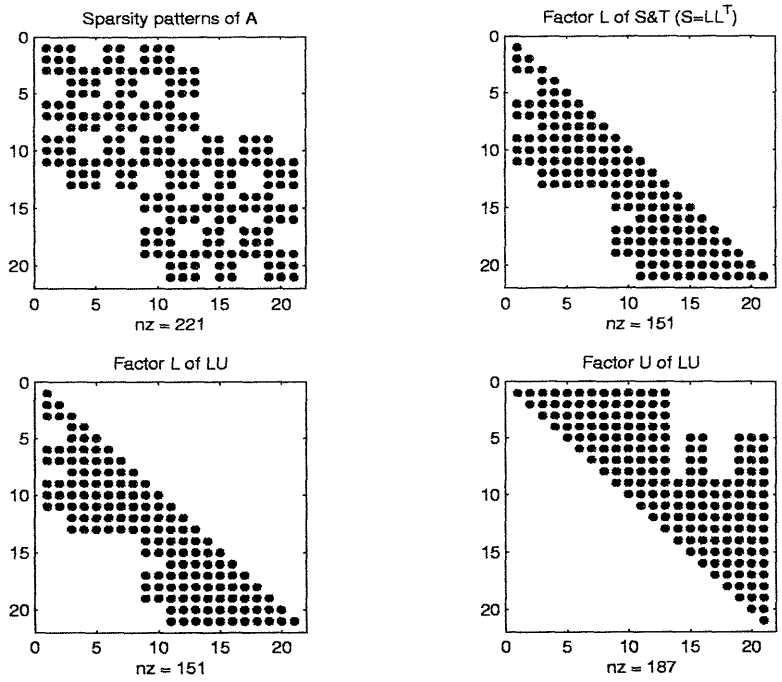

Figure 1.1: Sparsity patterns of the nonzero elements of the factor L of S&T decomposition, L and U of LU decomposition for Wathen's matrix

Note that the sparsity and the amount of nonzero elements from the factor L of S&T is exactly the same as L of LU, but the factor U has more nonzero elements than factor L. For tridiagonal matrices, the factor L of S&T, L and U of LU, yield a bidiagonal matrix.

The tests here for sparse matrices are just to show the numerical performance of the method.

In general, most of the iterative methods work very well for the solution of sparse

linear systems, usually using the preconditioning technique [3], because these systems are almost always sensitive (ill-conditioned), that is, a small variation in the coefficients of the matrix (and sometimes also in the vector b) will cause large variation in the solution of the system. The S&T method can also be used as preconditioner. Golub and Yuan have given some examples in [12].

Direct methods for sparse systems depend crucially on the precise pattern of sparsity of the matrix. For sparse matrices, one usually requires pivoting strategy (interchange of rows or columns) in the stage of the decomposition of the matrix to avoid breakdown. It is important to control the storage of the elements to keep the sparsity, besides of avoid in the possible fill-in. In particular, the LU decomposition and Cholesky decomposition methods have variant versions with pivoting. The direct methods are not recommended to solve large sparse systems, if A becomes large, direct methods typically become very expensive and inefficient; one find the "exact" solution in finite arithmetic operations, but the roundoff errors will increase and the storage requirement will be quite big. On the other hand, there are many techniques for the iterative methods to accelerate the convergence and to reduce the roundoff errors with cheap cost [25, 3]. In Chapter 5 we show the numerical results of the S&T decomposition for some sparse matrices.

The remainder of this dissertation is arranged as follows. In Chapter 2, we give the main idea of the S&T decomposition. We present the theorem and two algorithms: Golub-Yuan algorithm and our modified algorithm. We still do some comments about the test matrices. In Chapters 3 and 4, we present the numerical results for selected well-conditioned dense matrices and ill-conditioned dense matrices respectively. In Chapter 5, we show the results for some special matrices. Finally, in Chapter 6, we give some conclusions and suggestions for future work.

Chapter 2

S&T decomposition

2.1 Main idea

Many problems in numerical mathematics are modeled in terms of a system of linear equations. For instance, numerical methods for partial or ordinary differential equations and integral equations appeared in several problems of Physics, Engineering and the Image processing. Some practical problems are the following:

- airplane wing design;
- radar cross-section studies;
- flow around ships and other off-shore constructions;
- diffusion of solid bodies in a liquid;
- noise reduction; and
- diffusion of light through small particles.

One of the most common problems in real situations is to find the solution of n linear equations with n unknowns in the from of (1.2), and in most of the cases n is very large. In general, to find the solution of these linear systems is not an

easy task, specially if the coefficients matrix is very sensitive. A typical approach to solve such systems is to use LU decomposition which involves the roundoff error in some sense. Nicholas has done a analysis and proposed several ways to calculate and analyze the computational errors in [18] and his several papers can be found in his web site *"http://www.ma.man.ac.uk/~higham/pap-le.html"*. This will help to study the numerical behaviour of the S&T decomposition.

The S&T decomposition is based on the construction by **block-wise** of two matrices T and L (both lower triangulars). The matrix T, generated, has the power to transform the matrix A in a lower triangular matrix L such that the product of L by its transpose is symmetric and positive definite (SPD), that is, given a $n \times n$ nonsymmetric and nonsingular matrix A, we get a sequence of lower triangular matrix L_k whose product by its transpose, L_k^T, gives us a SPD matrix S_k, starting from the construction of a triangular matrix T_k multiplied by matrix A_k, that is, in the end of the process we obtain

$$TA = LL^T$$

where $LL^T = S$, or

$$\begin{bmatrix} t_{11} & & \\ \vdots & \ddots & \\ t_{n1} & \dots & t_{nn} \end{bmatrix} \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & & \\ \vdots & \ddots & \\ l_{n1} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & \dots & l_{n1} \\ & \ddots & \vdots \\ & & l_{nn} \end{bmatrix}$$

The matrix S is positive definite if we can make the decomposition in such a way that the elements of the diagonal of L are all positive, that is, $l_{i,i} > 0$ for all $i = 1, 2, \dots, n$. This idea is similar to the form of work of the algorithm of Cholesky.

THEOREM **2.1.1** *For every nonsingular and nonsymmetric $n \times n$ matrix A, whose leading principal submatrices are nonsingular, there exist a triangular matrix T and a symmetric and positive definite matrix S such that*

$$\boxed{TA = S = LL^T} \tag{2.1}$$

Proof: The proof is given in [12]. □

The S&T decomposition obtains the matrices T and L in block-wise, as follows:

Let A be $n \times n$ nonsingular and nonsymmetric matrix, then

$$A = \left[\begin{array}{c|c} A_k & a_{k+1} \\ \hline \tilde{a}_{k+1}^T & \alpha \end{array}\right] \tag{2.2}$$

where A_k is nonsingular.

Then, we can write the lower triangular T and L as follows

$$T = \left[\begin{array}{c|c} T_k & 0 \\ \hline t_{k+1}^T & \beta \end{array}\right] \tag{2.3}$$

and

$$L = \left[\begin{array}{c|c} L_k & 0 \\ \hline l_{k+1}^T & \tau \end{array}\right] \tag{2.4}$$

From (2.2), (2.3) and (2.4), we write the system

$$\begin{bmatrix} T_k & 0 \\ t_{k+1}^T & \beta \end{bmatrix} \begin{bmatrix} A_k & a_{k+1} \\ \tilde{a}_{k+1}^T & \alpha \end{bmatrix} = \begin{bmatrix} L_k & 0 \\ l_{k+1}^T & \tau \end{bmatrix} \begin{bmatrix} L_k^T & l_{k+1} \\ 0 & \tau \end{bmatrix} \tag{2.5}$$

where

$$\begin{bmatrix} L_k & 0 \\ l_{k+1}^T & \tau \end{bmatrix} \begin{bmatrix} L_k^T & l_{k+1} \\ 0 & \tau \end{bmatrix} = S \tag{2.6}$$

By multiplication of matrices in (2.5), we obtain

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

where $L_k^{-T} L_k^{-1} T_k = A_k^{-1}$.

The element, β, of the diagonal of the matrix T, are arbitrary variables, but we choose $\beta \neq 0$ such that

$$\frac{(\alpha - \tilde{a}_{k+1}^T L_k^{-T} L_k^{-1} T_k a_{k+1})}{\beta} > 0 \tag{2.7}$$

The elements, τ, of the diagonal of matrix L, represented by root square of the expression (2.7) must be larger than zero for all k.

2.2 Algorithms

For the decomposition of the matrix A as a product of a lower triangular matrix T and a symmetric and positive definite matrix $S = LL^T$, we make several numerical tests to evaluate the behavior of the algorithm proposed in [12] and we verify that one of the reasons that can cause instability in the decomposition process is the variation of the diagonal elements of the submatrix L_k or T_k. Golub and Yuan propose that the elements $t_{k+1,k+1}$, $(\beta = t_{k+1,k+1})$, are free, just considering that should be different from zero. However, by making an inversion in the algorithm, we verified that we can leave $l_{k+1,k+1}$, $(\tau = l_{k+1,k+1})$, as a free variables instead of $t_{k+1,k+1}$.

We note that for most dense matrices $l_{k+1,k+1}$ elements approach zero, which is the source of instability of the decomposition. To avoid this, we look for alternative forms to guarantee the continuity of the decomposition and to minimize the rounding errors. For example, we control the difference $s = a_{k+1,k+1} - \hat{l}_{k+1}^T l_{k+1}$, such as $|s| < 10^{-18}$ to avoid the elements $l_{k+1,k+1}$ close to zero. A similar behavior occurs with the matrix T_k. In each step of the algorithm we should choose $l_{k+1,k+1} > 0$ such that T_k is nonsingular. Therefore, since the decomposition is not unique, we can take values $\beta = t_{k+1,k+1}$ (or for $l_{k+1,k+1} = \sqrt{\tau}$, depended on the case) such that the decomposition presents best stability.

Algorithm for S&T decomposition by Golub-Yuan is the following:

ALGORITHM **2.2.1** (Golub-Yuan S&T)

Set t_{11} such that $t_{11}a_{11} > 0$ and $l_{11} = \sqrt{t_{11}a_{11}}$;

For $k = 1, \ldots, n-1$

$l_{k+1} = L_k^{-1} T_k a_{k+1}$,

$\hat{l}_{k+1} = L_k^{-1} \tilde{a}_{k+1}^T$,

$s = a_{k+1,k+1} - \hat{l}_{k+1}^T l_{k+1}$,

Choose $t_{k+1,k+1} \neq 0$ such that $\tau = t_{k+1,k+1} s > 0$ (or large enough)

$l_{k+1,k+1} = \sqrt{\tau}$

$t_{k+1} = T_k^T L_k^{-T} (l_{k+1} - t_{k+1,k+1} \hat{l}_{k+1})$

End

With above considerations, we obtain the modified algorithm for the S&T decomposition.

ALGORITHM **2.2.2** (Modified S&T)

Set t_{11} such that $t_{11}a_{11} > 0$, $l_{11} = \sqrt{t_{11}a_{11}}$, $\delta = 10^{-18}$ and choose guess initial $\eta = 1$;

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

$$t_{k+1,k+1} = sign(s)\eta$$

update η such that $l_{k+1,k+1} > 0$

$$\tau = t_{k+1,k+1} s$$

end

$$l_{k+1,k+1} = \sqrt{\tau}$$

$$t_{k+1} = T_k^T L_k^{-T} (l_{k+1} - t_{k+1,k+1} \hat{l}_{k+1})$$

End

REMARK 2.2.1 In each step of the decomposition, these two algorithms request the resolution of two linear systems and consequently are necessary the calculation of two triangular inverse.

Computational Cost

S&T

These two algorithms require $2n^3/3$ operations. Figure 2.1 shows the way as Algorithms 2.2.1 and 2.2.2 to make the decomposition. The solution of each

triangular system $Ly = b$ and $L^T x = y$ requires $n^2/2$ flops. So, the solution of the positive definite system $Ax = b$ using the S&T algorithm requires $2n^3/3 + O(n^2)$ arithmetic operations.

Cholesky

The Cholesky algorithm requires $n^3/6$ flops to compute L, one-half of the number of flops required to do same job using LU decomposition. Note that the process also requires n square roots. The solution of each triangular system $Ly = b$ and $L^T x = y$ requires $n^2/2$ flops. Thus, the solution of the positive definite system $Ax = b$ using the Cholesky algorithm requires $n^3/6 + O(n^2)$ flops and n square roots.

LU

The LU decomposition without pivoting requires about $n^3/3$ flops and the solution of each triangular system $Ly = b$ and $Ux = y$ needs only $n^2/2$ flops (same as that of Cholesky and S&T), the total flop count for solving the linear system $Ax = b$ by Gaussian elimination is about $n^3/3 + O(n^2)$.

We note that the S&T decomposition is about twice more expensive than LU decomposition and about four time more expensive than Cholesky decomposition. The cost of the S&T decomposition is larger because of construction of the matrix T.

2.3 Test matrices

In many applications in mathematics and engineering, the underlying physical properties of the problem introduce various patterns of structure into arising matrices. As the method here is not sufficiently advanced, in terms of stability, our test problems are still restricted. The chosen test matrices were obtained from the literature. We try to cover dense and sparse matrices with some type of structure. Gregory and David provided a large collection of test matrices (see,[14]). A important collection of test matrices, is "The Test Matrix Toolbox", developed by Nicholas in conjunction

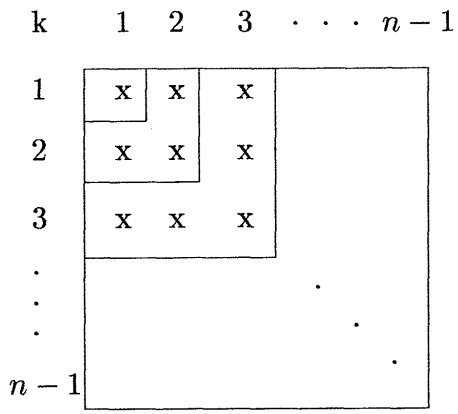

Figure 2.1: **Manner of S&T decomposition**

with his book *Accuracy and Stability of Numerical Algorithms* [18]. This collection has important matrices arising from the practical problems, and gives fundamental information on each matrix. He suggests 58 parametrized test matrices, which are mostly square, dense, nonrandom, and of arbitrary dimension. The collection includes known-inverse, known-eigenvalue; ill-conditioned, rank deficient, symmetric, positive definite, orthogonal, defective, involuntary, and totally positive matrices. We tested many of these matrices. We also tested some sparse matrices of the Horwell-Boing collection that is one of the most important collections of sparse matrices.

The codes for the algorithms presented in this dissertation are implemented in MATLAB which has unit roundoff $u = 2^{-53} \approx 1.11 \times 10^{-16}$, and are given in Appendix A. The MATLAB M-file for *LU* decomposition was obtained from [9]. All tests were run on a single processor Intel Pentium(III) 700 MHz with 128MB of RAM memory. CPU times are collected in the experiments and are used to measure the time of occupations of the machine. For each matrix (except for Moler Matrix) a table is built to show the relative error and the CPU time for each method taking $n = 50, 100, \ldots, 500$, except for Wathen's and Poisson's sparse matrix that have the

order of the matrices. To facilitate the analysis of the convergence of the algorithms, the figures are also given with the curves to describe the relative errors for $n = 1, 2, 3 \ldots, 500$. We measure the relative error as follows

- For S&T decomposition

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

  algorithms (2.2.1) and (2.2.2)

- For LU decomposition

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

- For Cholesky decomposition

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

Where $||.||_F$ denote the Frobenius norm. We choose Frobenius norm, because it is widely used and more suitable for measure approximation of matrices.

Chapter 3

Well-Conditioned dense matrices

3.1 Diagonally dominant matrix

In general, the class of diagonally dominant matrices, for row or column, is well-conditioned. The iterative methods are almost stable and obtain a good approximate solution of the system in a few iterations with respect to the size of the system.

We say that a $n \times n$ matrix A is strongly row diagonally dominant if

$$|a_{ij}| > \sum_{j \neq i}^{n} |a_{ij}|, \text{ for all } i \tag{3.1}$$

A column diagonally dominant matrix is similarly defined.

A column diagonally dominant matrix, like a symmetric and positive definite matrix, possesses the attractive property that row interchange is not necessary at any step during whole Gaussian elimination. The pivot element is already in the right place. Thus, at the first step, a_{11} being the largest in magnitude of all elements in the first column, no row interchange is necessary.

Our test matrix was built by taking random entries, from an uniform distribution with mean zero and variance one, $N(-1, 1)$, generated by MATLAB's **randn** function. Table 3.1 and Figure 3.1 show the performance of the algorithms. The update of $t_{k+1,k+1}$ was made with $\eta = ||L_{k+1}||_2$. Note that the S&T decomposition is stable

	LU		S&T		M-S&T
n	CPUtime	res(L, U)	CPUtime	res(T, L, L^T)	res(T, L, L^T)
50	0.05	1.43e-16	0.05	3.64e-16	3.32e-16
100	0.33	2.21e-16	0.33	4.66e-16	3.32e-16
150	0.88	2.45e-16	1.48	5.72e-16	3.35e-16
200	1.65	2.71e-16	4.12	6.14e-16	3.59e-16
250	3.3	3.01e-16	8.89	7.29e-16	4.01e-16
300	4.56	3.29e-16	16.3	8.26e-16	4.06e-16
350	7.87	3.55e-16	26.7	8.65e-16	4.28e-16
400	12.1	3.86e-16	40.7	8.75e-16	4.49e-16
450	15.5	4.13e-16	58.8	8.96e-16	4.56e-16
500	22.2	4.33e-16	66.7	1.02e-15	4.76e-16

Table 3.1: Performance for diagonally dominant matrix

for this matrix. The modified S&T algorithm gives us residual smaller than S&T algorithm.

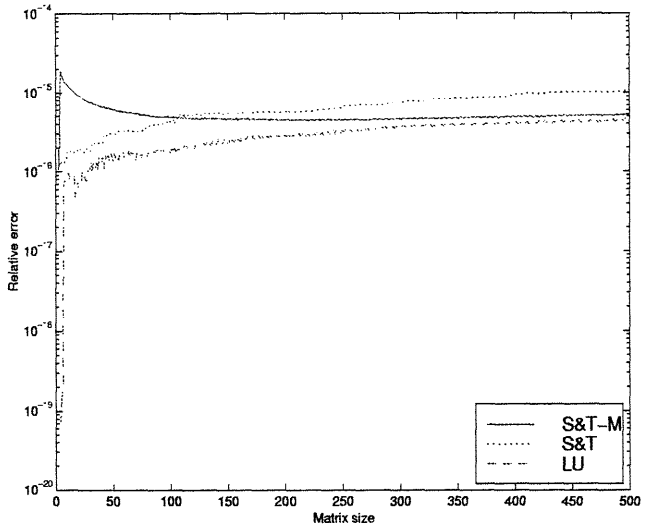

Figure 3.1: Relative error for diagonally dominant matrix

	LU		S&T		S&T
n	CPUtime	res(L,U)	CPUtime	res(T,L,L^T)	res(T,L,L^T)
50	0.11	1.27e-14	0	7.32e-10	3.45e-10
100	0.28	2.45e-14	0.28	2.47e-08	5.86e-09
150	0.66	8.81e-14	1.27	1.15e-07	5.94e-08
200	1.43	2.30e-13	3.52	2.49e-07	1.17e-07
250	3.08	3.31e-13	9.17	3.62e-07	1.58e-07
300	4.28	4.20e-13	15	5.4e-07	2.19e-07
350	6.21	4.68e-13	26.7	6.43e-06	1.94e-06
400	8.46	5.28e-13	35.2	1.02e-05	3.08e-06
450	14.5	5.92e-13	57.8	1.22e-05	3.93e-06
500	32.7	6.53e-13	66.6	1.39e-05	4.98e-06

Table 3.2: Performance for matrix with random entries

3.2 Random matrix

The random matrix chosen here, is a $n \times n$ matrix A with random entries, chosen from an uniform distribution with mean zero and variance one, $N(0,1)$, generated by MATLAB's **randn** function. Random matrices are always the favorite test matrices in test algorithm. Various results are known about the behaviour of matrices with random entries from the normal distribution. Some works have been published covering characteristics and details of this kind matrices, see, for example, the Doctor thesis of Alan Edelman [8] and [18, p. 517]. The results of our tests are in Table 3.2 and the curves of the relative error in Figure 3.2, respectively. The actualization of $t_{k+1,k+1}$ was made with $\eta = ||L_{k+1}||_2$.

3.3 Symmetric positive definite matrix

Here we choose a symmetric positive definite Moler matrix [16] defined by $A_n(\theta) = C_n(\theta)^T C_n(\theta)$, where $C_n(\theta)$ is a unit upper triangular with all $c_{i,j}$ elements, $i \neq j$, equal to θ. To construct this matrix we take $\theta = -2$ as in [6, 7] and to ensure stability in the decomposition, we update $t_{k+1,k+1}$ as follows

$$t_{k+1,k+1} = sign(s)\eta \tag{3.2}$$

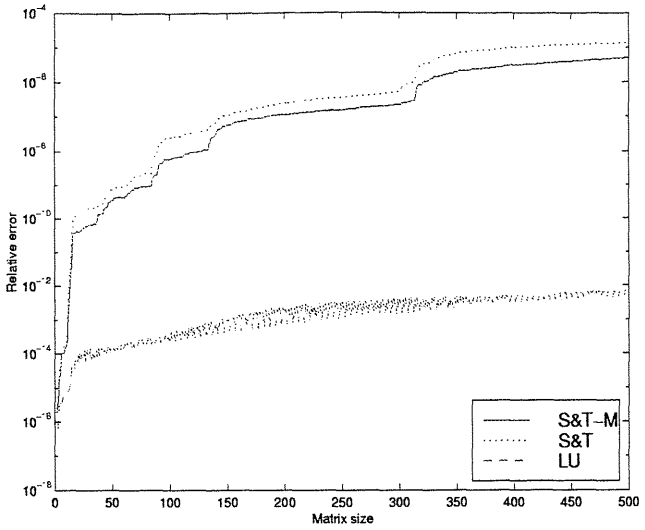

Figure 3.2: Relative error for Random matrix

where η is the infinity norm of the row of L in the step k, that is, $\eta = ||L_{k+1}||_\infty$. The idea here is to control elements $t_{k+1,k+1}$ to assure the stability of the decomposition and to get the possible better choices for L and T. We tested the modified S&T algorithm and original S&T algorithm with several negative integer values θ. In this case, T is identify, and L is unit triangular. The relative error obtained is zero and CPU time is 66,7 seconds. Therefore, the decomposition for Moler matrix is stable for parameter $\theta < 0$ and integer. When we choose the parameter $\theta > 0$ or non integer, the relative errors of modified S&T and Golub-Yuan S&T algorithms are similar to that at in LU decomposition, but not better. The Cholesky method is stable and makes the decomposition with relative error zero in 49 seconds.

Chapter 4

Ill-conditioned dense matrices

4.1 Hilbert matrix

The Hilbert matrix is a classic test matrix and one of the most famous ill-conditioned matrices. The (i, j) elements of this matrix are all in the form $\frac{1}{i+j-1}$, with $(i, j = 1, 2, \ldots, n)$, the elements of the inverse are integers and known explicitly [14]. It was widely used in the 1950s and 1960s for testing inversion algorithms and the solution of linear equation. For this problem, even for small n, both the condition number and numerical stability will be in the very bad situations. For example, Table 4.1 displays that for $n = 20$, cond(H_{20}) already reaches $1.0675e + 019$[1] continuing to grow at an exponential rate, that is, cond$(H_n) \approx e^{3.5n}$, when n is very large [26].

n	$Cond(H_n)$
2	19.2815
3	524.0568
4	1.5514e+004
5	4.7661e+005
20	1.0675e+019

Table 4.1: Condition number of the Hilbert matrix

[1]These numbers were obtained with the MATLAB M-file **cond**

This matrix is common to appear in practical problems of adjustment of curves by a polynomial for the method of the least squares. Forsythe and Moler [10] dedicate a chapter to the Hilbert matrix, to describe the underlying least square problem and discuss numerical computation of the inverse. Although many researchers use this matrix for testing algorithms, Nicholas [19] affirms that the Hilbert matrix is not ideal because of symmetric and positive definite and totally positive. More details about this matrix, including an asymptotic formula for $\text{cond}(H_n)$, can be found in Gregory and Karney [14, pp. 33-38, 66-73], [18, pp. 514], [2, pp. 206], [5, Chapter 6] and [26, 4].

Note that for $n = 500$, the relative error is in the order of 0.000197 for the S&T decomposition and $7.92e{-}17$ for the LU decomposition as seen in Table 4.2. Although relative error of the S&T decomposition is bigger than that of the LU decomposition, for this matrix we obtain a condition number $3.12e+08$ for the factor L of the S&T algorithm, less than $2.08e+12$ for the factor L and $8.16e+20$ for the factor U of LU. The S&T decomposition offers better conditioned triangular approximation matrices.

Here, we update η by $\eta = ||L_{k+1}||/2k$, $k = 1, 2, \ldots, n-1$, to control the diagonal elements of the matrices L and T. Table 4.2 and Figure 4.1 display the convergence of the method. The modified Algorithm 2.2.2 accelerated the convergence of the method significantly.

	LU		S&T		M-S&T
n	CPUtime	res(L,U)	CPUtime	res(T,L,L^T)	res(T,L,L^T)
50	0.06	5.14e-17	0.05	2.27e-06	3.39e-07
100	0.38	5.83e-17	0.28	0.0452	3.31e-07
150	1.04	6.31e-17	1.27	0.274	1.4e-06
200	1.37	6.66e-17	3.57	379	3.43e-06
250	2.47	6.94e-17	7.74	2.74e+03	1.47e-05
300	6.7	7.2e-17	14.1	5.92e+03	5.04e-05
350	10	7.41e-17	23.2	9.41e+03	7.9e-05
400	14.6	7.6e-17	35.3	1.51e+04	0.000126
450	19.9	7.76e-17	50.8	7.13e+04	0.000157
500	29.3	7.92e-17	65.4	1.69e+05	0.000197

Table 4.2: Performance of the methods for Hilbert matrix

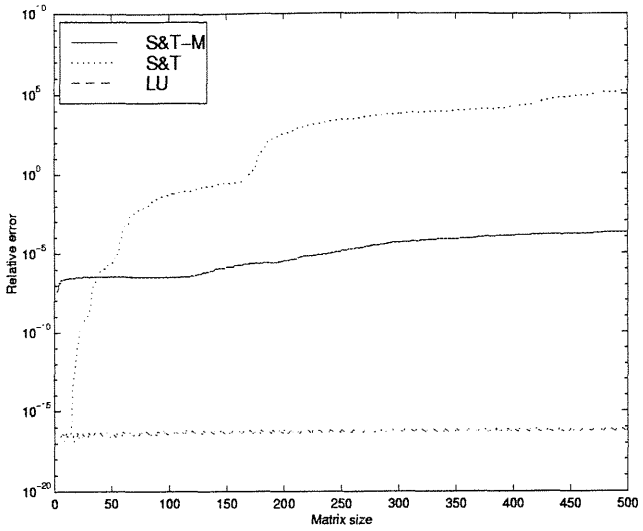

Figure 4.1: Relative residual for for Hilbert matrix

4.2 Lotkin matrix

Lotkin matrix is a special case of the Hilbert matrix whose first row altered to all ones. This matrix is nonsymmetric, ill-conditioned, and has many negative eigenvalues with small magnitude. The inverse has integer entries and is known explicitly, see [17]. Table 4.3 and Figure 4.2 illustrate the behaviour of the algorithms. In this case, the updating of the elements $t_{k+1,k+1}$, is made as in 4.1. Note that also in this case, our modifications improve the relative residual significantly compared with the Golub-Yuan algorithm. From this results, the *LU* decomposition is still better. But the result of the modified S&T algorithm is reasonably accepted for real applications.

	LU		S&T		M-S&T
n	CPUtime	res(L,U)	CPUtime	res(T,L,L^T)	res(T,L,L^T)
50	0.05	3.84e-17	0.06	1.99e-05	6.7e-07
100	0.11	3.92e-17	0.29	0.000797	6.7e-07
150	0.66	4.01e-17	1.37	0.00141	8.12e-07
200	1.59	4.04e-17	3.46	0.445	1.07e-06
250	3.18	4.07e-17	7.47	30.4	1.38e-06
300	5.99	4.09e-17	13.6	461	2.28e-06
350	9.01	4.12e-17	22.2	5.37e+03	5.65e-06
400	13.8	4.15e-17	33.7	4.29e+04	1.04e-05
450	19.5	4.17e-17	50.3	2.16e+05	2.03e-05
500	26.8	4.18e-17	66.9	2.36e+06	6.31e-05

Table 4.3: Relative error for Lotkin matrix

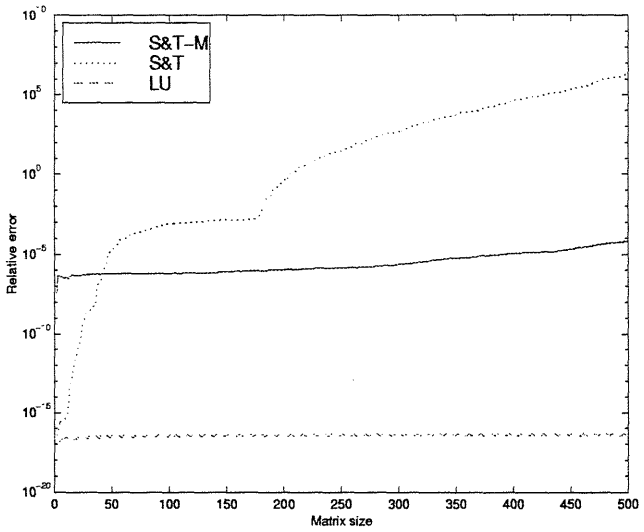

Figure 4.2: Relative error for Lotkin matrix

Chapter 5

Special matrices

5.1 Sparse matrices

For the sparse matrices (Poisson, Wathen, Dorr and Toeplitz Tridiagonal), we allocated ourselves a fixed value, $\eta = 2$, for the elements of the diagonal of the matrix T. Therefore, all elements $t_{k+1,k+1}$ are equal to 2 except t_{11} and t_{22}. This is the best alternative we found to guarantee a relative error as small as that of the LU decomposition for our experiments.

5.1.1 Poisson's matrix

Block tridiagonal matrix from Poisson's equation (sparse) is a kind of matrices that arises in linear equations obtained by discretizing certain elliptic partial differential equations and has the pentadiagonal matrix form. In general, these equations are subject to boundary conditions at the outer boundary of the range. There are no initial conditions, as Wave or Diffusion equations. Hence, they can not be solved by adapting the methods for simple differential equations. Here, we consider the block tridiagonal (sparse) matrix of order n^2 resulting from discretizing **Poisson's equation**

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

where $(x, y) \in \Omega = (0, 1) \times (0, 1)$. The five point finite difference method generates a block tridiagonal SPD coefficient matrix. This matrix is usually in the well-conditioned category. For more details, see [11, 21, 9] and [5, Chapter 6]. The results are given in Table 5.1. Figure 5.1 and Figure 5.2 illustrates the sparsity pattern. We note that the method is stable for this type of matrices. Although the modifications to update the elements $t_{k+1,k+1}$ is not good, the relative error is smaller than that of the LU decompositon when $n = 23^2$.

	LU		S&T		M-S&T
n	CPUtime	res(L, U)	CPUtime	res(T, L, L^T)	res(T, L, L^T)
7	0.05	9.87e-17	0.05	9.35e-17	1.24e-16
10	0.28	9.17e-17	0.28	1.35e-17	1.22e-16
12	0.87	1.19e-16	1.1	1.21e-16	1.20e-16
14	2.37	1.31e-16	3.24	1.25e-16	1.33e-16
16	5.28	1.32e-16	8.07	1.35e-16	1.35e-16
18	11	1.56e-16	17.7	1.36e-16	1.41e-16
20	21	1.58e-16	33.5	1.38e-16	1.53e-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

Table 5.1: Relative error for Poisson's matrix

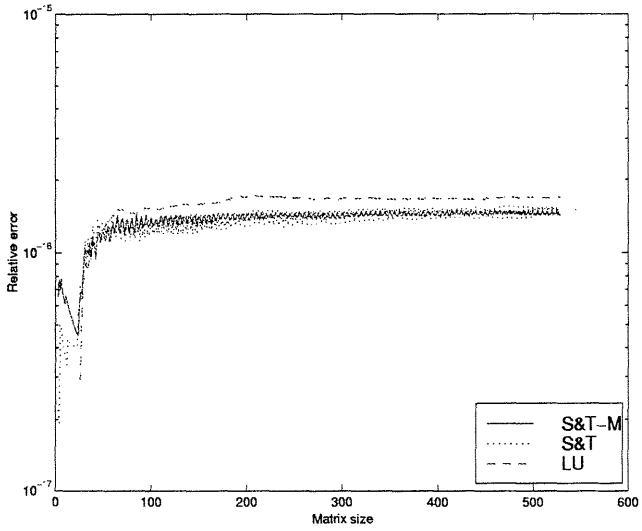

Figure 5.1: Curve of residual error for Poisson's matrix

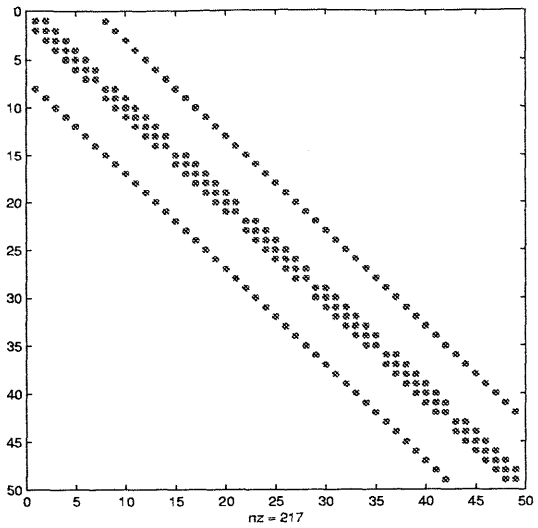

Figure 5.2: Sparsity pattern of the Poisson's matrix.

5.1.2 Wathen matrix

Wathen's matrix is a Finite Element matrix (sparse, random entries) for two-dimensional finite elements discretisation 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 precisely the consistent mass matrix for a regular l by m grid of 8-node (serendipity) elements in the plane. A is symmetric positive definite for any positive values of the density (l, m) which are randomly chosen in our test. In particular, if $D = diag(A)$, then the eigenvalues of $D^{-1}A$ lie in $(0.25, 4.5)$ for any positive integers l and m and any densities, see [28]. In our test we construct the matrix A by taking $l = m$. See the performance of the algorithms in Table 5.2 and in Figure 5.3 and the sparsity pattern in Figure 5.4. Here, we get better results than that by using LU decomposition in the two versions of S&T. For the Cholesky decomposition, the relative error and CPU time $1.51e-016$ and 0.66 seconds, respectively. So, the relative error of the modified S&T decomposition is smaller than that of the Cholesky decomposition.

	LU		S&T		M-S&T
n	CPUtime	res(L,U)	CPUtime	res(T,L,L^T)	res(T,L,L^T)
40	0.06	9.17e-17	0	1.14e-16	1.11e-16
96	0.22	1.42e-16	0.22	1.16e-16	1.02e-16
133	0.49	1.35e-16	0.82	1.31e-16	1.23e-16
225	1.87	1.32e-16	5.22	1.37e-16	1.27e-16
280	3.3	1.27e-16	11.3	1.09e-16	1.27e-16
341	5.5	1.38e-16	20.7	1.40e-16	1.32e-16
408	9.27	1.33e-16	36.5	1.39e-16	1.29e-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

Table 5.2: Relative error for Wathen matrix

5.1.3 CDDE1 - Matrix Market

This test matrix is 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

$$\begin{aligned} -\Delta u + 2p_1 u_x + 2p_2 u_y - p_3 u &= f \quad \text{in} \quad \Omega \\ u &= g \quad \text{on} \quad \partial\Omega \end{aligned}$$

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

$$Au = b$$

where A is of order m^2 and u and b are now vectors of size m^2. Centered differences are used for the first derivatives. If the grid points are numbered using the row-wise

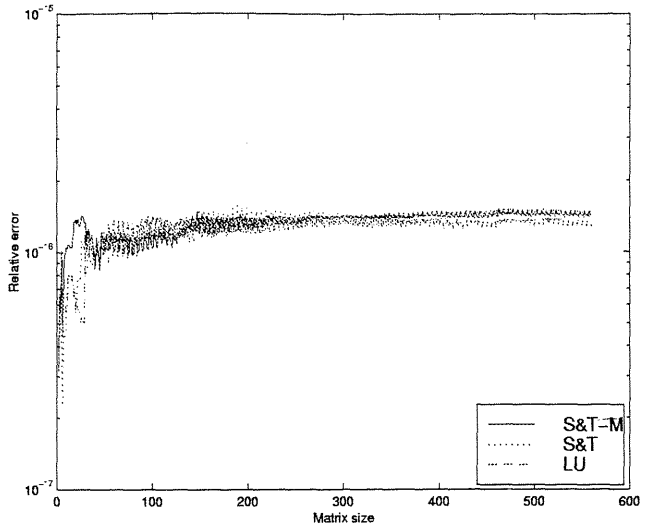

Figure 5.3: Relative error for Wathen's matrix

natural ordering, then A is a block tridiagonal matrix of the form

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

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 is a Model 2-D Convection Diffusion Operator (CDDE1) test and the matrix is extracted from the Harwell Boeing Collection [22], and we use the Matrix Market

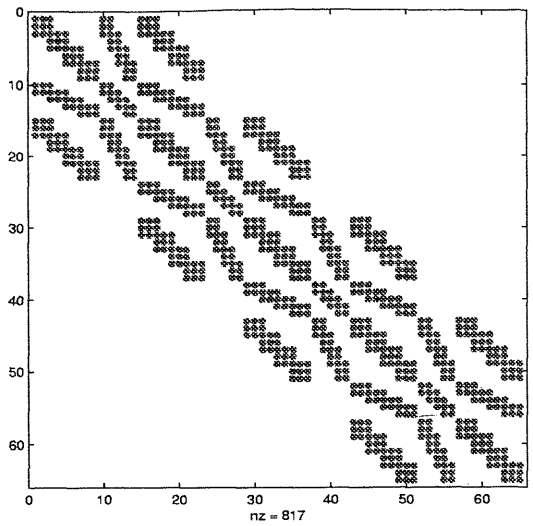

Figure 5.4: Sparsity pattern of the Wathen matrix.

format. It is a real nonsymmetric matrix. For our experiments, we take $n = 500$, so 2404 nonzeros elements arise in the matrix, $p_1 = 1$, $p_2 = 2$, $p_3 = 30$, where

p_1 is the coefficient of $2u_x$ in differential operator;

p_2 is the coefficient of $2u_y$ in differential operator;

p_3 is the coefficient of $-u$ in differential operator

Our modifications do not make a lot of difference with respect to Golub-Yuan algorithm. But the relative error in the decomposition of this matrix made by modified S&T or S&T algorithm is smaller than that of the LU decomposition. This illustrates the efficiency of the method for these structured matrices.

The sparsity pattern is in Figure 5.6 and the results of the methods are shown in Table 5.3 and Figure 5.5, respectively.

	LU		S&T		M-S&T
n	CPUtime	res(L,U)	CPUtime	res(T,L,L^T)	res(T,L,L^T)
50	0.05	7.92e-17	0.06	1.17e-16	1.20e-16
100	0.44	1.53e-16	0.44	1.52e-16	1.49e-16
150	1.53	1.71e-16	1.27	1.75e-16	1.67e-16
200	3.18	1.82e-16	3.46	1.84e-16	1.78e-16
250	6.21	1.85e-16	7.31	1.88e-16	1.85e-16
300	10.6	1.88e-16	13.3	1.94e-16	1.86e-16
350	16.5	1.9e-16	21.8	1.97e-16	1.91e-16
400	24.3	1.87e-16	33.2	1.97e-16	1.91e-16
450	34	1.89e-16	47.8	1.97e-16	1.95e-16
500	46	1.93e-16	66.2	1.97e-16	1.97e-16

Table 5.3: Relative error for CDDE1 - Matrix Market

5.1.4 Diagonally dominant Dorr matrix

The Dorr matrix $D_n(\alpha)$ [17, 16], which was also used in [6, 7], is a nonsymmetric, row diagonally dominant, tridiagonal M-matrix[1]. $D_n(\alpha)$ has diagonal dominance factors

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

This matrix is ill-conditioned for small values of the parameter α when $\alpha > 0$. Figure 5.7 illustrates the relative error for Dorr matrix. The Table 5.4 gives the behaviour to the algorithms. The LU decomposition works very well for this matrix. Here, we use $\eta = ||L_{k+1}||$ for update $t_{k+1,k+1}$. The modified S&T algorithm has residual as good as the Golub-Yuan algorithm.

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

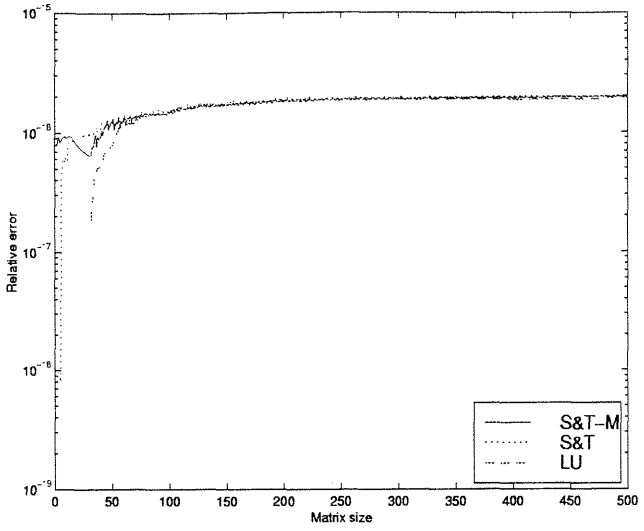

Figure 5.5: Relative error for CDDE1 - Matrix Market

5.2 Toeplitz Matrices

Here, we study the case where A is a Toeplitz matrices. These matrices arise in many applications, such as deconvolution and signal processing, communications engineering, and statistics [1]. In several situations, matrices arise in block structure, too such as diffusion of solid bodies in a liquid, noise reduction, diffusion of light through small particles. More details about Toeplitz matrices can be found in [24].

A Toeplitz matrix has constant diagonals. In other words, A_{ij} depends only on $(i-j)$ and is defined as follows

$$A_n(a) = (a_{i-j})_{i,j=1}^{n}$$

where a_k is the kth Fourier Coefficient of a,

$$a_k = \frac{1}{2\pi}\int_{-\frac{1}{2}}^{\frac{1}{2}} a(e^{i\theta})e^{-ik\theta}d\theta \ , \quad k = 0, \pm 1, \pm 2, \pm 3, \ldots$$

The Toeplitz matrix has the form

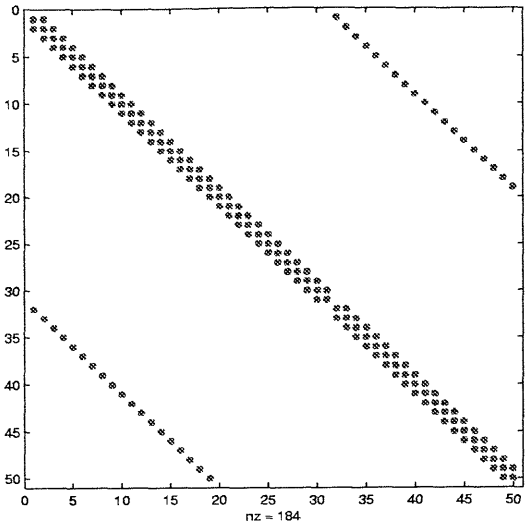

Figure 5.6: Sparsity pattern of the CDDE1 - Matrix Market

$$A_n(a) = (a_{i-j})_{i,j=1}^{n} = \begin{bmatrix} a_0 & a_{-1} & \dots & \dots & a_{1-n} \\ a_1 & a_0 & \ddots & & a_{2-n} \\ \vdots & \ddots & \ddots & a_{-1} & \vdots \\ a_{n-2} & & a_1 & a_0 & a_{-1} \\ a_{n-1} & \dots & \dots & a_1 & a_0 \end{bmatrix}$$

5.2.1 Prolate matrix

Prolate matrix is symmetric, ill-conditioned Toeplitz matrix. This is the typical case arising in signal processing applications. It was first studied by Slepian in the 1950's at Bell Labs. In 1993 Varah also studied this matrix, see [27]. The Prolate matrix is a symmetric Toeplitz matrix given by $A(\alpha)$.

If $0 < \alpha < 1/2$ then

- A is positive definite
- the eigenvalues of A are all distinct, lie in $(0, 1)$, and tend to cluster around 0 and 1.

	LU		S&T		M-S&T
n	CPUtime	res(L,U)	CPUtime	res(T,L,L^T)	res(T,L,L^T)
50	0.05	0	0.06	1.68e-13	1.36e-13
100	0.27	0	0.28	3.20e-13	3.97e-13
150	0.66	0	1.26	5.77e-13	5.85e-13
200	1.43	0	3.4	7.40e-13	7.04e-13
250	2.63	0	8.02	7.98e-13	8.57e-13
300	4.17	0	13.8	8.33e-13	8.85e-13
350	6.27	0	22.7	9.17e-13	9.16e-13
400	8.9	0	41.4	9.58e-13	8.72e-13
450	12.2	0	51.3	9.73e-13	9.38e-13
500	16.2	0	69.2	9.52e-13	9.49e-13

Table 5.4: Convergence of the decomposition for Dorr matrix

In our tests we consider α defaults to 0.125. For this value of α, the Prolate matrix is very ill-conditioned with condition number for $A_{500}(\alpha)$ about $2.20e+18$. The eigenvalues also lie in $(0,1)$. The diagonal elements of the matrix T are updated by $\eta = ||L_{k+1}||$. Table 5.5 shows the convergence of the method when increasing the order of matrix and Figure 5.8 gives the curve of the relative error. Although the modified S&T and Golub-Yuan S&T are not good for the matrices, modified S&T algorithm does improve the numerical stability.

	LU		S&T		M-S&T
n	CPUtime	res(L,U)	CPUtime	res(T,L,L^T)	res(T,L,L^T)
50	0.05	2.45e-16	0.06	0.306	1.57e-06
100	0.27	7.21e-16	0.28	0.353	2.23e-05
150	0.76	1.2e-16	1.32	0.569	0.000132
200	1.59	1.02e-15	3.52	0.794	0.000202
250	2.52	9.28e-15	7.63	1.17	0.000227
300	4.11	9.89e-15	14	5.02	0.0003
350	6.1	9.77e-15	23	16.8	0.00083
400	8.78	9.57e-15	33.6	22.1	0.000942
450	12.2	9.34e-15	48.2	27.2	0.000962
500	15.3	9.13e-15	66.6	31.5	0.000969

Table 5.5: Relative error for Prolate matrix

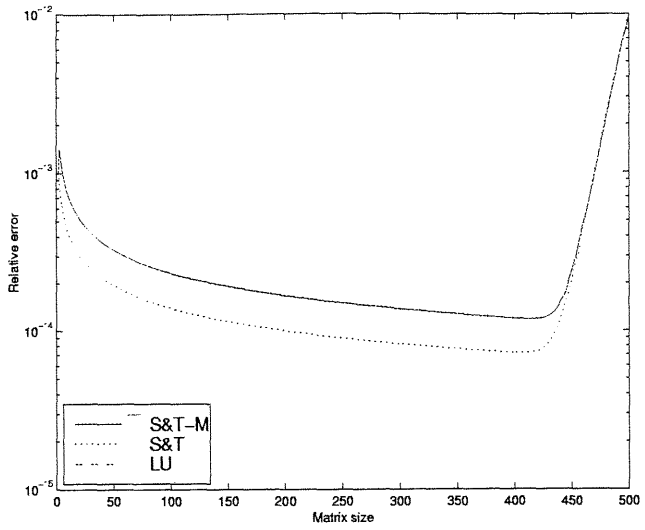

Figure 5.7: Curve of relative error of the Dorr matrix

5.2.2 Circulant matrix

Circulant matrices are special Toeplitz matrices where the diagonals wrap around. This matrix has the property that each row is obtained from the previous one by cyclically permuting the entries one step forward. The circulant linear system also can be resolved by using FFT with good stability (see, [18],p.468).

One application of circulant linear systems is in the proconditioned conjugate gradient method for solving Toeplitz systems. G. Strang [23] was the first to propose circulant preconditioner for solving Toeplitz systems. Huckle [20] does an analysis on iterative method for solving linear Toeplitz equations and shows different ways to get improved preconditioned conjugate gradient algorithms.

The eigensystem of A is explicitly known: If t is a n-th root of unity, then the inner product of v with $w = (1, t, t^2, \ldots, t^n)$ is an eigenvalue of A and $w = (n, n-1, \ldots, 1)^T$ is an eigenvector A, where v is the first row of the matrix A, see [17]. See the results of the methods for this matrix in Table 5.6 and the curve in Figure 5.9.

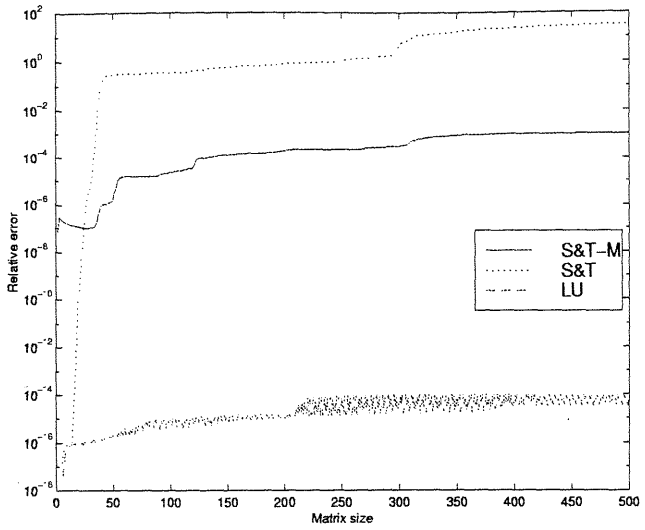

Figure 5.8: Curve of relative error for Prolate Matrix

5.2.3 Toeplitz tridiagonal matrix

The special case of a system of linear equations is tridiagonal, that is, it has nonzero elements only on the diagonal plus or minus one column. It occurs frequently. The common systems are banded diagonal, with nonzero elements only along a few diagonal lines adjacent to the main diagonal (above and below). For tridiagonal sets, the procedures of the LU decomposition, forward and back-substitution each takes only $O(n)$ operations, and the whole solution can be encoded very concisely. Naturally, one does not reserve storage for the full $n \times n$ matrix, but only for the nonzero components, stored as three vectors. The solution using the S&T method is also found in only $O(n)$ operations with forward and back-substitution. The scheme of storage doesn't need to be full, because the decomposition preserve the most sparsity of the original matrix. We can see more information in [17]. The test matrix which we use here is also suggested by Matrix Market [22].

The tridiagonal matrix has the form

	LU		S&T		M-S&T
n	CPUtime	res(L,U)	CPUtime	res(T,L,L^T)	res(T,L,L^T)
50	0.6	1.42e-15	0	3.5e-13	4.83e-14
100	0.16	2.95e-15	0.28	2.16e-12	3.43e-13
150	0.71	4.52e-15	1.26	6.17e-12	7.89e-13
200	1.59	6.01e-15	3.51	1.17e-11	1.56e-12
250	3.57	7.7e-15	7.68	1.99e-11	2.69e-12
300	6.42	9.22e-15	14.1	3.41e-11	4.45e-12
350	10.4	1.09e-14	23	4.89e-11	5.92e-12
400	15.4	1.23e-14	35	6.87e-11	9.37e-12
450	22.3	1.4e-14	53.4	9.11e-11	1.2e-11
500	30.5	1.57e-14	67.7	1.23e-10	1.58e-11

Table 5.6: Relative error for Circulant matrix

$$\begin{bmatrix} d_1 & c_1 & & \\ e_1 & d_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & e_{n-1} & d_n \end{bmatrix} \tag{5.1}$$

where c, d, and e are all scalars. This matrix has feature the tridiagonal Toeplitz matrix of order n with subdiagonal elements c, diagonal elements d, and superdiagonal elements e. The eigenvalues are $d + 2\sqrt{(ce)}\cos(k\pi/(n+1))$, where $k = 1, 2 \ldots n$. A is a symmetric positive definite M-matrix as the form

$$\begin{bmatrix} d & -c & & \\ -c & d & \ddots & \\ & \ddots & \ddots & -c \\ & & -c & d \end{bmatrix} \tag{5.2}$$

To test Algorithms 2.2.1 and 2.2.2, we take the tridiagonal Toeplitz matrix in (5.1), with $e = 3$, $d = 2$ e $c = -1$. The two versions of the S&T method yield very good results, although the LU decomposition still have smaller error. We also test

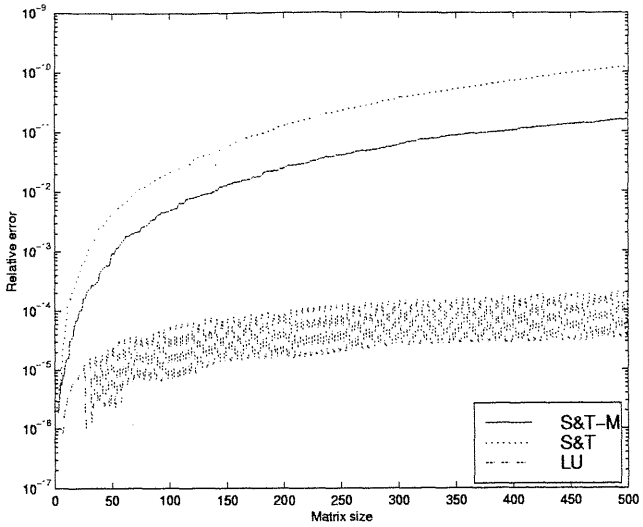

Figure 5.9: Residual error for Circulant matrix

for the case in that T is a symmetric positive definite matrix, that is, $T_{500}(-1, 2, -1)$ and get relative errors $1.22e-16$ in 0 seconds, $6.18e-17$, with 68.9 seconds and 0 with 26.5 seconds, for the Cholesky[2], S&T and LU decomposition, respectively. The performance of the algorithms is in Table 5.7 and Figure 5.10. Note that for this matrix our modification are worse than Golub-Yuan S&T algorithm, but the Golub-Yuan S&T algorithm is as good as the LU decomposition.

[2]For Cholesky were used the MATLAB built-in function *chol*.

	LU		S&T		M-S&T
n	CPUtime	res(L, U)	CPUtime	res(T, L, L^T)	res(T, L, L^T)
50	0.05	1.46e-17	0.06	7.85e-16	7.99e-16
100	0.17	1.03e-17	0.27	6.97e-16	1.15e-15
150	0.61	8.41e-18	1.21	6.15e-16	1.47e-15
200	1.54	7.28e-18	3.46	5.69e-16	1.7e-15
250	3.13	6.51e-18	7.52	5.4e-16	1.93e-15
300	5.54	5.94e-18	13.9	5.19e-16	2.13e-15
350	9.12	5.5e-18	23.5	5.04e-16	2.32e-15
400	13.4	5.14e-18	34.7	4.93e-16	2.51e-15
450	19.3	4.85e-18	50.8	4.84e-15	2.69e-15
500	26.5	4.6e-18	67.3	4.76e-17	2.79e-15

Table 5.7: Relative error for Tridiagonal Toeplitz matrix (sparse)

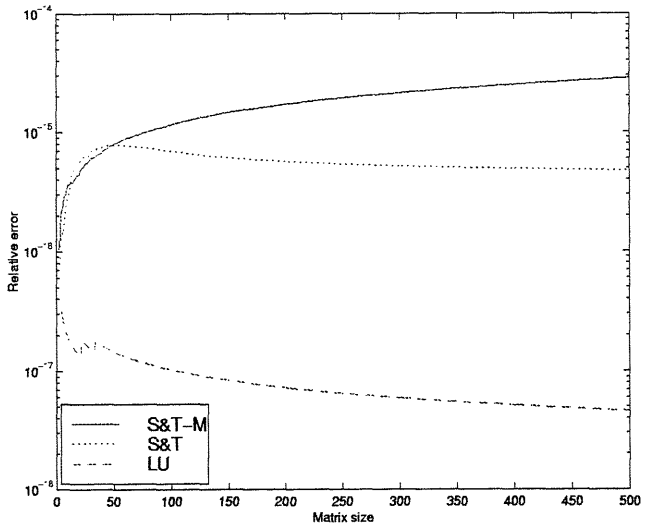

Figure 5.10: Curve of relative error for Tridiagonal Toeplitz matrix (sparse)

Chapter 6

Conclusions and future work

The numerical tests in this dissertation showed the efficiency of the new decomposition of matrices (S&T), proposed by Golub and Yuan in [12]. For the sake of numerical stability, we modify the Golub-Yuan S&T algorithm here. We tested several types of matrices for the Golub-Yuan algorithm and our modified algorithm. The current algorithm does not give the numerical stability as good as LU decomposition for our test matrices. In some cases, especially when the matrix is sparse and possesses structures as in Chapter 5, the algorithm can produce as small error as that by the competitive methods. The modified algorithm does improve the numerical stability by our tests, specially for very ill-conditioned matrices such as Hilbert, Lotkin and Prolate matrix. In terms of our numerical experiments, the S&T decomposition is stable in some sense for many real applications even it is not as good as the LU (or Cholesky) decomposition for some types of matrices. Of course, the algorithm still needs improvement to get better numerical stability, which is our future research issue.

Future work

- To apply the S&T decomposition and to try to improve the solution of the system $Ax = b$ using the iterative refinement.

- To research the possibility to apply the pivoting technique in the S&T method.
- To develop the incomplete decomposition to apply to sparse matrices with strategy of fill-in reduction and to compare with other methods.
- To do tests in eigenvalues and eigenvectors problems.
- Analysis of roundoff errors.
- Numerical Analysis and instability for $n >> 500$

Appendix A

Implementation

Implementation of the Algorithm 2.2.1.

```
function styuan
% S&T Decomposition
%     This routine decomposes the matrix A in the form T*A=S=L*L'
%     Where,
%          T is lower triangular matrix, L is the factor of Cholesky
%          S is symmetric positive definite matrix.
%-------------------------------------------------------------
[A,OP,qsi]=testes; % function with test matrices
n=length(A);
T(1,1)=A(1,1);
L(1,1)=sqrt(T(1,1)*A(1,1));

for k=1:n-1
     L(k+1,1:k) = (L(1:k,1:k)\(T(1:k,1:k)*A(1:k,k+1)))';
     LL = L(1:k,1:k)\A(k+1,1:k)';
     s = A(k+1,k+1) - L(k+1,1:k)*LL;
```

```
     if abs( s ) < 0.1e-18
       T(k+1,k+1) = 1;
     else
      T(k+1,k+1) = sign( s );
   end
   L(k+1,k+1) = sqrt( T(k+1,k+1)*s );
   T(k+1,1:k) = (T(1:k,1:k)'*(L(1:k,1:k)'\(L(k+1,1:k)'-T(k+1,k+1)*LL)))';
end
timeST=toc;
RelativeErrorST=norm(A-T\L*L','fro')/norm(A,'fro');
```

Implementation of the Algorithm 2.2.2.

```
function mSt
% Modified S&T Decomposition
%     This routine decomposes the matrix A in the form T*A=S=L*L'
%     with actualization of the elements of the diagonal of T.
%     Where,
%          T is lower triangular matrix, L is the factor of Cholesky
%          S is symmetric positive definite matrix.
%-------------------------------------------------------------
[A,OP,op,eta]=testes; % Function with test matrices
n=length(A);
T(1,1)=A(1,1);
L(1,1)=sqrt(T(1,1)*A(1,1));
flops(0);
tic for k=1:n-1
     L(k+1,1:k) = (L(1:k,1:k)\(T(1:k,1:k)*A(1:k,k+1)))';
```

```
      LL = L(1:k,1:k)\A(k+1,1:k)';
      s = A(k+1,k+1) - L(k+1,1:k)*LL;

       if abs( s ) < 0.1e-18  % tolerance to avoid breakdown problems
           T(k+1,k+1) = 1;
       else
           T(k+1,k+1) = sign( s )*eta;

             % Take the matrix and makes the appropriate updating
             if op==2 | op==15
                eta = norm(L(k+1,1:k))/2*k;
              elseif op==5 | op==13 | op==14 | op==21
                eta = norm(L(k+1,:),1);
              elseif op==1 | op==18
                eta = norm(L(k+1,:));
              elseif op==8 | op==17 | op==28
                eta = 2;
             end
   end
   L(k+1,k+1) = sqrt( T(k+1,k+1)*s );
   T(k+1,1:k) = (T(1:k,1:k)'*(L(1:k,1:k)'\(L(k+1,1:k)'-T(k+1,k+1)*LL)))';
end
tST=toc;
flpST=flops;
erSTr = norm(A-T\L*L', 'fro')/norm(A,'fro'); % verify the error
```

Bibliography

[1] M. Embree A. Bottcher and V. I. Sokolov. On large Toeplitz band matrices with an uncertain block. *Oxford University Computing Laboratoy, Wolfson Building, Parks Road, Oxford OX1 3QD, UK,* Elsevier, April 2001.

[2] Kendall E. Atkinson. *An Introduction to Numerical Analysis.* John Wiley & Sons, University of Iowa, second edition, 1988.

[3] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition.* SIAM, Philadelphia, PA, 1994.

[4] Man-Duen Choi. Tricks or treats with the Hilbert matrix. *Amer. Math. Monthly,* 90:301–112, 1983.

[5] Biswa Nath Datta. *Numerical Linear Algebra and Applications.* An International Thomson Publishing Company, California, USA, 1994.

[6] James W. Demmel and Nicholas J. Higham. Stability of block algorithms with fast level-3 BLAS. *ACM Trans. Math. Software,* 18(3):274–291, September 1992.

[7] James W. Demmel, Nicholas J. Higham, and Robert S. Schreiber. Stability of block *LU* factorization. *Numerical Linear Algebra with Applications,* 2(2):173–190, 1995.

[8] Alan Edelman. Eigevalues and condition number of random matrices. *Massachusetts Institute of Technology,* Doctor Thesis, May 1989.

[9] Laurene V. Fausett. *Applied Numerical Analysis Using Matlab.* Prentice-Hall, Inc., ISBN: 0-13-319849-9, 1999.

[10] George E. Forsythe and Cleve B. Moler. *Computer Solution of Linear Algebric Systems.* Prentice-Hall, Englewood, Cliffs, NJ, USA, 1967.

[11] Gene H. Golub and Charles F. Van Loan. *Matrix Computation, Third Edition.* Jhons Hopkin Universty Press, Baltimore, Maryland, 1996.

[12] Gene H. Golub and Jin-Yun Yuan. S&t: Symmetric-triangular decomposition and its applications Part I: Theorems and algorithms. *BIT (accepted),* 2001.

[13] Gene H. Golub and Jin-Yun Yuan. S&t: Symmetric-triangular decomposition and its applications Part II: Applications. *BIT 40th. Meeting, Lund, Swedem,* 2000.

[14] Robert T. Gregory and David L. Karney. *Collection of Matrices for Testing Computational Algorithms.* Wiley Interscience, New York, 1969.

[15] M. R. Hestenes and E.Stiefel. Methods of conjugate gradient for solving linear systems. *J. Research of the National Bureau of Standards,* 49(1952) 409-436.

[16] Nicholas J. Higham. Algorithm 694: A collection of test matrices in MATLAB. *ACM Transactions on Mathematical Software,* 17(3):289–305, September 1991.

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

[18] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

[19] Nicholas J. Higham. Testing linear algebra software. In Ronald F. Boisvert, editor, *Quality of Numerical Software: Assessment and Enhancement*, pages 109–122. Chapman and Hall, London, 1997.

[20] Thomas Huckle. Iterative methods for Toeplitz-like matrices. *Universitat Wurzburg, Germany*, 1994.

[21] Carsten Keller, Nicholas I. M. Gould, and Andrew J. Wathen. Constraint preconditioning for indefinite linear system. *Rutherford Appleton Laboratory*, February 1999.

[22] Iain S.Duff, Roger G.Grimes, and John 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, Rotherford Appleton Laboratory, Didcot, Oxon, UK*, December 1992.

[23] Gilbert Strang. A proposal for toeplitz matrix calculations. *Stud. Appli. Math., 74:171-176*, 1986.

[24] Thomas Strohmer. Four short stories about Toeplitz matrix calculations. *Department of Mathematics, University of California, Davis, CA 95616-8633, USA*, October 2000.

[25] M. Tismenetsky. A new preconditioning technique for solving large sparse linear systems. *Linear Algebra and Its Applications*, pages 154–156:331–353, 1991.

[26] John Todd. The condition of the finite segments of the Hilbert matrix. In contributions to the solution of systems of linear equations in the determination of eigenvalue. *Applied Mathematics Series, National Bureau of Standards, United States Department of Commerce, Washington, DC*, (39):106–116, 1954.

[27] J. M. Varah. The prolate matrix. *Linear Algebra and Its Applications*, (187):269–278, 1993.

[28] A. J. Wathen. *Realistic eigenvalue bounds for the Galerkin mass matrix.* IMA J. Numer. Anal., 7, pp. 449-457, 1987.

[29] J. Y. Yuan. *Applied Iterative Analysis. Teaching Note.* Universidade Federal do Paraná, Curitiba, PR, Brasil, *1998.*