IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 11, Issue 2 Ver. III (Mar - Apr. 2015), PP 01-09
www.iosrjournals.org

Factorization of Symmetric Indefinite Matrices

R. Purushothaman Nair*
Advanced Technology Vehicle Project
Vikram Sarabhai Space Centre,Thiruvananthapuram-695 022, India.

Abstract: *A factorization procedure for matrices that satisfy the without row or column exchange condition (**WRC**) is introduced. The strategy is to reduce a column to corresponding column of the identity matrix. Factors so obtained are triangular matrices with same entries in a row or column. These factors and their inverse with simple structures can be constructed using the entries of a given non-zero vector without any computations among the entries. The advantage is that* n^2+2n *flops associated with conventional Gaussian elimination (**GE**) or Neville elimination (**NE**) can be saved using the present approach in solving* n x n *non-homogeneous linear system. The benefits of applying this procedure for decomposing symmetric indefinite matrices are discussed by introducing a tridiagonal reduction procedure. Results on numerical experiments are provided to demonstrate that entries are much less perturbed than GE for typical problems considered here using the proposed approach.* ***AMS classifications:*** *15A04, 15A23*

Keywords: *Symmetric Indefinite matrices; Matrix Factorization; Linear Transformations; Tridiagonal Reduction.*

I. Introduction

Consider factorization of a given $n \times n$ non-singular square matrix $A \in M_n$ as

$$A = LU \tag{1.1}$$

In (1.1) factor L is a lower triangular $n \times n$ matrix and U is an $n \times n$ unit upper triangular matrix. If A is a symmetric positive definite matrix then (1.1) can be represented as in (1.2) below where D is a diagonal matrix.

$$A = U^T D U \tag{1.2}$$

Now consider factorization of symmetric indefinite matrix A given below.

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \\ 3 & 2 & 3 & 4 \end{bmatrix} \tag{1.3}$$

For this matrix A, a factorization process need not be ended with a decomposed form LU or U^TDU. Hence special strategies are devised in factorizing a symmetric indefinite matrix. One such strategy is to involve off diagonal entries in the pivoting process. For maintaining symmetry, transformations such as Gauss, Neville etc. used for introducing zeros in a column are applied as a sequence of conjugate transformations on A. Final target is to represent A as a conjugate transformation of a lower triangular matrix L on a symmetric tridiagonal matrix N as $A = LNL^T$. When Gauss transformations are applied, to avoid element growth in factors, the entry with maximum absolute value is searched in a column and is brought to pivot position. This type of factorization is proposed by Parlette and Reid[1]. Aasen[2] proposed modifications on Parlette and Reid algorithm, which is more stable and economic than [1]. Here, at first, Parlette and Reid algorithm will be modified with use of new transformation matrix in place of Gauss transformation and advantages of such a step will be discussed. Following this, a strategy for decomposing a given symmetric indefinite matrix shall be presented. Advantages of this algorithm over Aasen's algorithm also will be discussed.

II. A Simple Operator Matrix for Transforming a Given Non-Zero Vector to a Column of the Identity Matrix

Let a non-zero vector $x=[x_1\ x_2\ \dots\ x_n]^T$; $x_i \neq 0\ ,i=1,2,\dots,n$ be given. Consider the lower bidiagonal matrix and its inverse defined as below.

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{x}) = [\boldsymbol{\alpha}_{ij}];\ \boldsymbol{\alpha}_{ij} &= 1/\ \boldsymbol{x}_i\ ;\ \boldsymbol{for}\ \boldsymbol{i} = \boldsymbol{j}\ \ ;\ \boldsymbol{i}, \boldsymbol{j} = 1,2,\ldots,\boldsymbol{n}. \\ \boldsymbol{\alpha}_{ij} &= -1/\ \boldsymbol{x}_i\ ;\ \boldsymbol{for}\ \boldsymbol{i} = \boldsymbol{j}+1. \\ \boldsymbol{\alpha}_{ij} &= 0\ ;\ \boldsymbol{i} > \boldsymbol{j}+1\ \boldsymbol{and}\ \boldsymbol{i} < \boldsymbol{j}+1. \end{aligned} \tag{2.1}$$

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{x})^{-1} = [\boldsymbol{\beta}_{ij}];\ \boldsymbol{\beta}_{ij} &= \boldsymbol{x}_i\ ;\ \boldsymbol{for}\ \boldsymbol{i} \ge \boldsymbol{j}\ ;\ \boldsymbol{i}, \boldsymbol{j} = 1,2,\ldots,\boldsymbol{n}. \\ \boldsymbol{\beta}_{ij} &= 0\ ;\ \boldsymbol{for}\ \boldsymbol{i} < \boldsymbol{j}+1. \end{aligned} \tag{2.2}$$

Columns in (2.2) are consisting of the given vector itself and its projections to subspaces of dimension $\boldsymbol{k}=\boldsymbol{n}-1, \boldsymbol{n}-2,\ldots,1$. Clearly, $\boldsymbol{B(x)x}=\boldsymbol{e_1}$ and $\boldsymbol{B(x)^{-1}e_1}= \boldsymbol{x}$. These results (2.1) and (2.2) can be applied to factorize a given $\boldsymbol{n}\ \boldsymbol{x}\ \boldsymbol{n}$ non-singular matrix, say, $\boldsymbol{A}=[\boldsymbol{x_{ij}}]$. **C**onsider $\boldsymbol{A_1}=\boldsymbol{B(x_1)A}$ where $\boldsymbol{x_1}=[\ \boldsymbol{x_{11}}\ \boldsymbol{x_{21}}\ \ldots\ \mathbf{x_{n1}}\ ]^{\boldsymbol{T}}$, first column vector of $\boldsymbol{A}$. Then first column of $\boldsymbol{A_1}$ will be $\boldsymbol{e_1}$. $\boldsymbol{L_1}=\boldsymbol{B(x_1)^{-1}}$ will be the first lower triangular factor. Now consider, first row of $\boldsymbol{A_1}$. Let it be $\boldsymbol{y_1^T}=\boldsymbol{e_1^T A_1}$. $\boldsymbol{U_1}=\boldsymbol{B(y_1)^{-T}}$ will be the first upper triangular factor. $\boldsymbol{A_1^*}= \boldsymbol{AB(y_1)^T}$. So in $\boldsymbol{A_1^*}$, both first row and column will be identical to that of the identity matrix. This procedure can be extended to the second column and row of $\boldsymbol{A_1^*}$ to derive $\boldsymbol{A_2^*}$ and so on, and terminate after $\boldsymbol{n}$-steps, to obtain $\boldsymbol{A_n^*}=\boldsymbol{I}$ and $\boldsymbol{A}= \boldsymbol{L_1L_2\ldots L_nU_nU_{n-1}\ldots U_1}$. It may be noted that $\boldsymbol{L_i}$ are of general structure (2.3) and $\boldsymbol{U_i}$ are that of transposes of (2.3).

$$\boldsymbol{L}_i = \begin{array}{c} \left[\begin{array}{cc} \boldsymbol{I}_i & 0 \\ 0 & \boldsymbol{B}(\boldsymbol{x}_i)^{-1} \end{array}\right]\begin{array}{l} \scriptstyle \boldsymbol{i}-1 \\ \scriptstyle \boldsymbol{n}-\boldsymbol{i}+1 \end{array} \\ \begin{array}{cc} \scriptstyle \boldsymbol{i}-1 & \quad \scriptstyle \boldsymbol{n}-\boldsymbol{i}+1 \end{array} \end{array} \tag{2.3}$$

If in a column or row, some entries are zeros, then corresponding to blocks of non-zero entries, appropriate block submatrices of the matrices (2.1) and (2.2) can be considered. In such matrices, against zero entries of the considered column or row, corresponding rows and columns of the identity matrix can be considered. For example, let $\boldsymbol{x}=[\ \boldsymbol{x_1}\ \boldsymbol{x_2}\ \boldsymbol{0}\ \ \boldsymbol{x_4}\ \boldsymbol{x_5}\ \boldsymbol{0}\ \ \boldsymbol{0}\ \ \ \boldsymbol{0}\ \ \ \ \boldsymbol{x_9}]^{\boldsymbol{T}}$. Then by the above way of considering the non-zero blocks individually, we have

$$\boldsymbol{B}(\boldsymbol{x}) = \begin{bmatrix} 1/\boldsymbol{x}_1 &&&&&&&& \\ -1/\boldsymbol{x}_1 & 1/\boldsymbol{x}_2 &&&&&&& \\ && 1 &&&&&& \\ &&& 1/\boldsymbol{x}_4 &&&&& \\ &&& -1/\boldsymbol{x}_4 & 1/\boldsymbol{x}_5 &&&& \\ &&&&& 1 &&& \\ &&&&&& 1 && \\ &&&&&&& 1 & \\ &&&&&&&& 1/\boldsymbol{x}_9 \end{bmatrix}$$

$$\boldsymbol{B}(\boldsymbol{x})^{-1} = \begin{bmatrix} \boldsymbol{x}_1 &&&&&&&& \\ \boldsymbol{x}_2 & \boldsymbol{x}_2 &&&&&&& \\ && 1 &&&&&& \\ &&& \boldsymbol{x}_4 &&&&& \\ &&& \boldsymbol{x}_5 & \boldsymbol{x}_5 &&&& \\ &&&&& 1 &&& \\ &&&&&& 1 && \\ &&&&&&& 1 & \\ &&&&&&&& \boldsymbol{x}_9 \end{bmatrix}$$

There involves no computations among entries to constitute these matrices as against computation of suitable multiplier for elimination in **GE**. Notably this strategy can work only with matrices that satisfy **WRC** condition. That is, when all leading principle sub-matrices of the given matrix are non-singular.

The strategy of triangularzing A for linear system solution $Ax=b$ can be presented as follows. At i^{th} step, in matrix A_{i-1}^* , divide each row $i,i+1,\ldots,n$ by the respective leading entries $x_{ii}^*, x_{i+1,i}^*,\ldots, x_{n,i}^*$. Replace k^{th} row by subtracting k^{th} row from $(k-1)^{th}$ row for $k=n,n-1,\ldots,i-1$. That is, with k^{th} row R_k, perform $R_k \rightarrow R_k / x_{kk}^*$. Then perform $R_k \rightarrow R_k - R_{k-1}; k=n,n-1,\ldots i-1$. This will reduce its i^{th} column say, C_i to e_i. Note that for $k=1$, $A_0^*=A$. So after n-steps , the system will be reduced to $Ux=c$ where U is an unit upper triangular matrix.

Advantages we have with factors $B(x)^{-1}$ can be presented as follows. These matrices can be easily constructed as in (2.2) from entries of a given column. There involves no computation of multipliers. This will lead to reduce computational cost by n^2+2n operations. A typical example from Golub and Van loan [3] is considered in the numerical illustration and experiments sections.

III. Minimization Of Flops In The Procedure Compared To Gaussian Elimination

A non-homogeneous linear system of the type $Ax=b$ where A is an $n \times n$ non-singular matrix that satisfies **WRC**, can be solved by forward eliminations and backward substitutions using the factorization procedure by avoiding n^2+2n flops of **GE**.

In **GE** in order to eliminate entries of first column of $A=(a_{ij})$ below the pivot entry a_{11}, the $n-1$ multipliers to be computed are $-a_{k1}/a_{11}, k=2,3,\ldots,n.$ Similarly for the second column, it requires to compute $n-2$ multipliers and so on. Thus a total of $n(n-1)/2$ multipliers are to be computed so as to reduce the given linear system to reduced triangular form. Thus computation of multipliers requires $n(n-1)/2$ divisions. It may be noted that reduced upper triangular form in **GE** will be typically a non-unit upper triangular form as below.

$$
\begin{array}{lcl}
a_{11}^* x_1 + a_{12}^* x_2 + a_{13}^* x_2 + \ldots + a_{1n}^* x_n & = c_1^* & \\
\qquad a_{22}^* x_2 + a_{23}^* x_2 + \ldots + a_{2n}^* x_n & = c_2^* & \\
\vdots & \vdots & \\
\qquad\qquad a_{n-1,n-1}^* x_{n-1} + a_{n-1n}^* x_n & = c_{n-1}^* & \\
\qquad\qquad\qquad a_{nn}^* x_n & = c_n^* &
\end{array}
\qquad (3.1)
$$

In the proposed factorization process, eliminations in a column are conducted using its own entries. So the $n(n-1)/2$ divisions required for computing multipliers are not required. As the mapping is to a column of the identity matrix, the reduced upper triangular form will be unit triangular. So after n steps, the reduced upper triangular form typically will be

$$
\begin{array}{lcl}
x_1 + a_{12}^{**} x_2 + a_{13}^{**} x_2 + \ldots + a_{1n}^{**} x_n & = c_1 & \\
\qquad x_2 + a_{23}^{**} x_2 + \ldots + a_{2n}^{**} x_n & = c_2 & \\
\vdots & \vdots & \\
\qquad\qquad x_{n-1} + a_{n-1n}^{**} x_n & = c_{n-1} & \\
\qquad\qquad\qquad x_n & = c_n &
\end{array}
\qquad (3.2)
$$

Now in order to arrive at this form (3.2) out of (3.1) additional row wise divisions by the non-unit leading coefficients are required. This requires another $((n+1)(n+2)/2)-1$ divisions.

So while solving for the unknowns using backward substitutions in (3.2), the additional $((n+1)(n+2)/2)-1$ divisions of reduced upper triangular form of **GE** also are not required. In summary, using the present approach in (3.2), these additional divisions required in **GE** as presented in (3.1) can be avoided and computational load will be reduced by

$$n(n-1)/2+((n+1)(n+2)/2)-1=n^2+2n \qquad (3.3)$$

So a minimization of computational load is achieved in the present approach.

The factorization procedure is discussed in detail in Nair[4].

IV. Parlett-Reid Algorithm with the new Transformation

In order to overcome the difficulties encountered while factorizing a given symmetric indefinite matrix A, several steps have been taken so that A is decomposed to simpler systems. Some of these steps are listed below.

i) $\boldsymbol{A}$ is decomposed to symmetric tridiagonal and lower triangular factors so as to involve the off-diagonal elements in the formation of Gauss matrix $\boldsymbol{M_k}$ and pivoting process.
ii) To maintain symmetry, matrices $\boldsymbol{M_k}$ are applied to $\boldsymbol{A}$ as a sequence of conjugate transformations.
iii) To avoid possible element growth, permutations are applied so that current vector $\boldsymbol{x} = [x_1\ x_2\ \dots\ x_n]^T$ is having an entry, $max\{|x_1|, |x_2|, \dots |x_n|\}$ at the pivot position.

With the above steps, it has been demonstrated by Parlett and Reid[1] that $\boldsymbol{A}$ can be decomposed as

$$\boldsymbol{PAP^T = LNL^T} \tag{4.1}$$

In (4.1) $\boldsymbol{P}$ is a permutation matrix, $\boldsymbol{L}$ is a unit lower triangular matrix and $\boldsymbol{N}$ is a symmetric tridiagonal matrix. The Parlett and Reid algorithm is a straight forward implementation of Gauss transformation matrices $\boldsymbol{M_k}$; $k=1,2,\dots,n$ in accordance with the above three aspects and can be generally represented as

$$\boldsymbol{A_k = M_k(P_kA_{k-1}P_k^T)M_k^T} \quad ; k=1,2,\dots,n-2 \tag{4.2}$$

For matrix $\boldsymbol{A}$ of section-1, if this algorithm is applied, following are the factors.

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1/3 & 1 & 0 \\ 0 & 2/3 & 1/2 & 1 \end{bmatrix} \quad \boldsymbol{N} = \begin{bmatrix} 0 & 3 & 0 & 0 \\ 3 & 4 & 2/3 & 0 \\ 0 & 2/3 & 10/9 & 0 \\ 0 & 0 & 0 & 1/2 \end{bmatrix}$$

It is computationally expensive with $\boldsymbol{n^3/3}$ flops as far as factorization of symmetric matrices is concerned and less efficient when compared to Aasen's[2] algorithm with $\boldsymbol{n^3/6}$ flops. Aasen's algorithm is stable with a simple pivoting strategy.

The sample problem considered here from Golub and Van Loan [3] is a typical example to highlight the advantages of factorization by matrices (2.1). These are demonstrated below by factorization of the symmetric indefinite matrix $\boldsymbol{A}$ in (1.3) by implementing (4.2) of Parlett-Ried factorization where transformation $\boldsymbol{B(x_k)}$ as discussed in section-2 is applied in place of $\boldsymbol{M_k}$, for $k=1,2,\dots,n$, first without pivoting and later with pivoting.

V. Parlett-Reid Algorithm with *B(x)* as Operator

Consider the matrix in (1.3) as $\boldsymbol{A_0}$.

Step-1

Take the first column to construct $\boldsymbol{T_1}$ and $\boldsymbol{T_1^{-1}}$ as defined in (2.1) and (2.2). Obtain $\boldsymbol{A_1} = \boldsymbol{T_1^{-1}A_0\, T_1^{-T}}$. Note that in Step-1, the first column and row of $\boldsymbol{A_1}$ is that of the identity matrix.

Step-2

Take the second column vector of $\boldsymbol{A_1}$ where **we** consider only those entries on the diagonal and below. As in step-1, construct $\boldsymbol{T_2}$ and $\boldsymbol{T_2^{-1}}$, where as defined already, the first two columns of these will be that of the identity matrix. Obtain $\boldsymbol{A_2} = \boldsymbol{T_2^{-1}A_1\, T_2^{-T}}$. In A_2, the first three columns and rows will be that of the identity matrix.

Actually with 2 steps, prescribed by Parlett and Reid, the factors, symmetric tridiagonal matrix $\boldsymbol{N}$ and lower triangular matrix $\boldsymbol{L}$ are ready. The matrix $\boldsymbol{L}$ is given by

$$\boldsymbol{L} = \boldsymbol{T_1T_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 3 & -4 & -1 \end{bmatrix}$$

The matrices $\boldsymbol{L}$ and $\boldsymbol{N}$ satisfy the equation $\boldsymbol{A = LNL^T}$. We shall continue the factorization with one more additional step for reasons cited below in the observations.

Step-3

Construct $\boldsymbol{T_3}$ and $\boldsymbol{T_3^{-1}}$, using third column of $\boldsymbol{A_2}$ and obtain $\boldsymbol{A_3} = \boldsymbol{T_3^{-1}A_2\, T_3^{-T}}$.

$$A_3 = T_3^{-1} A_2 T_3^{-T} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 3/4 & 1 \\ 0 & 0 & 1 & 8 \end{bmatrix} = N$$

$$L = T_1 T_2 T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 3 & -4 & 1/2 \end{bmatrix}$$

In general $A_k = T_k^{-1} A_{k-1} T_k^{-T}$ can be obtained at k^{th} step, $k=1,2,..,n-1$ to decompose a given $n \times n$ symmetric matrix as in (4.1).

If possible pivoting at each step in the above process of decomposing matrix (1.3) so that current column entries are with maximum absolute values than corresponding entries of all other columns towards right of it, after step-3 we get,

$$N = A_3 = T_3^{-1} A_2 T_3^{-T} = \begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 1/2 & 1 & 0 \\ 0 & 1 & 6 & 1 \\ 0 & 0 & 1 & -1/8 \end{bmatrix}$$

The matrix L is given by

$$L = T_1 T_2 T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 3 & -0.5 & 0 \\ 0 & 3 & -1.0 & 2 \end{bmatrix}$$

i. Thus it can be observed that
ii. N is a symmetric tridiagonal matrix with 1 as its off-diagonal entries where these are computed multipliers of each steps in GE.
iii. In the lower triangular component L, as the diagonal entries are avoided for generating the operator T_k at step-k, the first column and row of L will be always e_1 and e_1^T respectively.
iv. It has ended up with unity along the off diagonals of N rather than along the diagonal of matrix L as in Parlett-Reid. This is an added advantage with this factorization. That is, one need not be concerned with the computation and storage of the off-diagonal entries.
v. To complete the factorization in this way, additional one more step has to be conducted. So there are total $n-1$ steps as against $n-2$ steps prescribed by Parlett and Reid. The operator $B(x)$ for this step has its $n-1$ rows and columns identical with that of I. So the concerned matrix multiplications are simple operations solely intended to update the last diagonal entry of N.

VI. Procedure for Factorization of Symmetric Indefinite Matrix

Aasen's algorithm takes advantage of inverses of previous Gauss Transformations, M_i ; $i=1,2,\ldots k-1$, applied to factorize A. For convenience, let permutations P_k applied to A at step-k, $k=1,2,\ldots,n-2$ be ignored to understand crucial activities that lead to diagonal entries of symmetric tridiagonal matrix N, say α_k , $k=1,2,\ldots,n$ and its off-diagonal entries β_{k+1}, $k=2,3,\ldots n$.

At step-k, product of Gauss Transformations M_k applied on A, $\prod_{i=k-1}^{1} M_i^{-1}$ and k^{th} column $[a_{ik}]^T$; $i=1,2,\ldots,n-2$ of A, are used to generate following for the k^{th} step

i) k^{th} Gauss Vector
ii) k^{th} Gauss Transformation M_k

iii) α_k , k^{th} diagonal element of N
iv) β_{k+1} , $k+1^{th}$ off-diagonal element of N.

Keeping the above requirements as targets, a new algorithm shall be introduced to reduce positive indefinite matrices to symmetric tridiagonal matrix N and lower triangular matrix L satisfying $A=LNL^T$. As already observed, because of the feature that factorization by $B(x)$ ends up with a symmetric tridiagonal matrix N which is having off-diagonal entries as 1, one need not be concerned with computation and storage of β_{k+1} , $k=2,\ldots.n$. So here at each step, vector x of the transformation $B(x)$ shall be computed whose $k+1^{th}$ entry will be α_{k+1} ; $k=1,2,\ldots.,n-1$ where α_1 will be entries a_{11} of A. The central concept of the algorithm going to be presented here is to make use of the equation

$$A_k = (T_k^{-1}\ldots.T_2^{-1}T_1^{-1})A(T_k^{-1}\ldots.T_2^{-1}T_1^{-1})^T \qquad (6.1)$$

at step k to generate vector x_{k+1} and there by diagonal entry α_{k+1} of N.
Consider at step k,

$$H_k = (T_k^{-1}\ldots.T_2^{-1}T_1^{-1})^T \qquad (6.2)$$

Let h be the n vector defined by

$$h = (H_k)^T A(H_k e_{k+1}) \qquad (6.3)$$

Now obviously h is the required vector that shall define x_{k+1} for generating

$$T_{k+1}^{-1}=I_{k-2}+B(x_{k+1}) \qquad (6.4)$$

In (6.4) I_{k-2} is an $n \times n$ matrix consisting of the first upper-left $k-2$ columns and rows of the identity matrix and other rows and columns are zeros. $k+1^{th}$ element of h will be the required diagonal element α_{k+1} of N. Thus making use of steps (6.2), (6.3) and (6.4), it can be obtained T_{k+1} and α_{k+1} and after $n-1$ steps, all n diagonal entries of N will be computed. The lower triangular matrix L will be provided by

$$L = T_1T_2\ldots.T_{n-1} \qquad (6.5)$$

The computed L and N will be satisfying equation $A=LNL^T$. In Aasen's algorithm using Gauss Transformation M_k at each step, once a vector h is computed, $max\{ |h_{k+1}|, |h_{k+2}|, \ldots\ldots,|h_n|\}$ is brought to pivot position $k+1$ of h by exchanging entries at positions $k+1$ and $q,k+1\leq q\leq n$ where $|h_q|\geq |h_j|$; $j=k+1,k+2,\ldots,n$. The procedure defined by (6.1) through (6.4) above is a simple one with $O(n^3/6)$ flops for computing vectors, say h_k ; $k=1,2,\ldots n-1$ using (6.3) as in the case of Aasen's algorithm. As presented in Section-3, computations of multipliers are not required with this procedure. Hence $(n-2)(n-1)/2-1$ divisions can also be saved compared to Aasen's algorithm.

VII. Numerical Illustration Of The Procedure For Factorization Of Symmetric Indefinite Matrix

Numerical illustration of the procedure shall be presented below by factorizing the same matrix A used above in the discussion on Parlett-Reid algorithm with $B(x)$ operator.

Step-1

Consider the same matrix (1.3). Using the first column vector, T_1 and T_1^{-1} can be constructed. We have α_1 is 0. Note that the diagonal position is ignored and so first column and row of T_1 and T_1^{-1} are that of the identity matrix.

$$H_1 = (T_1^{-1})^T$$
$$H_1e_2 = [0 \;\; 1 \;\; 0 \;\; 0]^T$$
$$A(H_1)e_2 = [1 \;\; 2 \;\; 2 \;\; 2]^T$$
$$H_1^TA(H_1)e_2 = [1 \;\; 2 \;\; -1 \;\; -1/3]^T$$

So h at step-1 is $[1 \;\; 2 \;\; -1 \;\; -1/3]^T$ and so α_2 **is** 2. Avoid diagonal position, and construct T_2 and T_2^{-1}.

Step-2

Compared to the Parlett-Reid algorithm, T_2 and α_2 are arrived at very easily. Obtain $H_2 =(T_2^{-1}\,T_1^{-1})^T$.

$$H_2e_3 = [0 \;\; 1 \;\; -1/2 \;\; 0]^T$$
$$A(H_2)e_3 = [0 \;\; 1 \;\; 1/2 \;\; 1/2]^T$$
$$H_2^TA(H_2)e_3 = [0 \;\; 1 \;\; 3/4 \;\; -1/2]^T$$

Thus h at step-2 is $[0 \;\; 1 \;\; 3/4 \;\; -1/2]^T$, α_3 is $3/4$ and as before avoiding diagonal entry, T_3 and T_3^{-1} can be constructed.

Step-3

$$\begin{array}{lcl} \text{Obtain } \boldsymbol{H_3} & = & (\boldsymbol{T_3^{-1}T_2^{-1}\ T_1^{-1}})^T. \\ \boldsymbol{H_3e_4} & = & [0 \quad 2 \quad -4 \quad 2\,]^T \\ \boldsymbol{A(H_3)e_4} & = & [0 \quad 0 \quad -2 \quad 0]^T \\ \boldsymbol{H_3^TA(H_3)e_4} & = & [0 \quad 0 \quad 1 \quad 8]^T \end{array}$$

In this last step, as promised, the algorithm successfully computed 4th diagonal entry $\alpha_4 = 8$ from computed $\boldsymbol{h}$ given as $[0 \quad 0 \quad 1 \quad 8]^T$. Computational load is significantly reduced in obtaining diagonal entries α_k ; $k=2,3,\ldots,n$ and operators T_k ; $k=1,2,\ldots n-1$ compared to Parlett-Reid. Thus the procedure is based on generating the diagonal entries of $\boldsymbol{N}$, making use of the matrix $\boldsymbol{H_k}$ at step $\boldsymbol{k}$.

The advantages and uniqueness of the process compared to Aasen's algorithm are

i) No computing and storage of off diagonal entries of symmetric tridiagonal matrix $\boldsymbol{N}$ are required as these will be always unity.
ii) No computing of multipliers for eliminating entries below pivot positions of a column.
iii) No searching for the element with in the vector x_{k+1} having maximum absolute value and exchange with the element at pivot position are involved.
iv) Column and row exchanges may be implemented whenever such exchanges make the current column entries with maximum absolute values compared to corresponding entries of columns towards right of it.
v) The operator involved is $\boldsymbol{B(x)}$ instead of Gauss Transformation $\boldsymbol{M_k}$.
vi) End result will be exactly the same factors computed using Parlett-Reid with $\boldsymbol{B(x)}$ as operator.

It can be observed that equation (6.3) generating successive vectors h_k ; $k=1,2,..,n-1$ replaces multiplication of matrices in (4.2) with much simple matrix-vector multiplication to achieve exactly same result. This also is a minimization of computational costs, resulting in $n^3/6$ operations as against $n^3/3$ operations in (4.2).

VIII. Results of Numerical Experiments conducted using Aasen's Algorithm and Proposed Algorithm

Both algorithms were implemented by coding in Borland C++ Version 3.1 and executed under MS VC++ 6.0 in a Pentium IV PC using double precision arithmetic's. It may be noted that there were searching for maximum absolute valued entry in Gauss vector at each step and corresponding pivoting is also implemented with respect to Aasen's algorithm.

The test conducted was to decompose concerned matrices $\boldsymbol{A}$ to compute triangular factors, lower triangular factor $\boldsymbol{L}$, Symmetric tridiagonal factor $\boldsymbol{N}$, and compute $\boldsymbol{A^*=LNL^T}$. Then absolute forward errors discussed in Higham [5] $\boldsymbol{abs(A- A^*)}$ are computed to verify the effectiveness of the two algorithms.

Test 8.1

The $\boldsymbol{4 \times 4}$ matrix $\boldsymbol{A}$ of section-1 was generated and tested to show effect of floating point arithmetic with the algorithms. Results are tabulated below.

Table 8.1 Absolute forward errors in Decomposing Indefinite Matrix of size $\boldsymbol{4 \times 4}$

	Aasen's Algorithm	Present Algorithm
Minimum	0.00000e+000	0.00000e+000
Maximum	**0.00000e+000**	**4.44089e-016**
Mean	**0.00000e+000**	**5.55112e-017**

Here the matrix is dense and both algorithms reconstructed the matrix in a comparable way.

Test 8.2

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{3} \\ \mathbf{1} & \mathbf{2} & \mathbf{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{2} & \mathbf{3} & \mathbf{3} \\ \mathbf{3} & \mathbf{0} & \mathbf{3} & \mathbf{4} \end{bmatrix}$$

In this test some entries other than those at pivot positions of the $\boldsymbol{4 \times 4}$ matrix of section-1 was replaced with zeros and performed the two decompositions.

Table 8.2: Absolute forward errors : Decomposition of Sparse Indefinite Matrix of size $\boldsymbol{4 \times 4}$

	Aasen's Algorithm	Present Algorithm
Minimum	0.00000e+000	0.00000e+000
Maximum	**2.22045e-016**	**0.00000e+000**
Mean	**2.77556e-017**	**0.00000e+000**

In test 8.2, note that when the matrix become sparse, proposed algorithm takes advantage of this structural property to decompose it in a better way.

Test 8.3

In this test, column entries at non-pivot positions of the ***8 x 8*** matrix ***A*** of section-1 was selectively and programmatically replaced with zeros. The matrix is then decomposed using the algorithms to obtain results below.

Table 8.3.3 Comparison of Forward Relative Errors For Test 8.3

	Aasen's Algorithm	Present Algorithm
Minimum	0.00000e+000	0.00000e+000
Maximum	2.66454e-015	3.55271e-015
Mean	5.78000e-016	3.19189e-016

By executing the programs it can be observed that error increasing tendency has a bias to the final stages of decomposition with the present algorithm where as these are scattered over the entire entries with Aasen's algorithm. In other words, almost all the entries are perturbed more with the Aasen's algorithm where as only the last row and column entries are perturbed with the present algorithm.

Test 8.4

For this test 8.4, the matrix of section –1 of order *50 x 50* is generated and decomposed. For convenience here only results of forward errors are presented in table 8.4.

Table 8.4 Comparison of Relative Errors For decomposing Matrix ***A*** of order ***50 x 50***

	Aasen's Algorithm	Present Algorithm
Minimum	**0.00000e+000**	**0.00000e+000**
Maximum	**1.56319e-013**	**1.13971e-011**
Mean	**1.39726e-014**	**1.49756e-012**

Cleary with the dense structure, Aasen's algorithm is performing in a better way. But it may be noted that present algorithm is a in a much simple way brings out this result compared to Aasen's algorithm.

Test 8.5

For this test 8.5, in matrix of test 8.4 of order **50 x 50,** zeros were introduced using the code presented in test-8.4 to make it sparse and decomposed. The results corresponding to forward relative errors are as below.

Table 8.5 Comparison of Relative Errors For decomposing Sparse Matrix A of order 50 x 50

	Aasen's Algorithm	Present Algorithm
Minimum	**0.00000e+000**	**0.00000e+000**
Maximum	**1.47793e-012**	**3.19744e-014**
Mean	**1.04563e-013**	**3.24736e-015**

This is the salient result out of all these tests. When the matrix is sparse and has nice special structural properties such as symmetry and order property for its column and row entries, this procedure can take advantage of these properties to decompose the matrix in a better way. One can also note that when almost all entries are perturbed in Aasen's algorithm, right-lower boarder entries are perturbed more and the more the entries are towards left-upper boarder, the less the perturbations to these entries by present algorithm. For example, there are zero perturbations for the very first column entries while decomposing using present algorithm.

IX. Conclusions

The procedure presented here involves O(**n^3/6)** flops. It works by simultaneously reducing the columns and rows to decompose a given symmetric indefinite matrix **A** to equivalent symmetric tridiagonal matrix **N** and triangular matrix **L** such that **$A=LNL^T$**. These matrices can be constituted using the entries of a given non-zero vector without any computations among its entries. It transforms the vector to a column of identity matrix. Benefit of such a property is that off-diagonal entries of symmetric tridiagonal matrix **N** are all unity. Consequently successive vectors **h** of transformation **B(h)** are computed where off diagonal entries need not be computed as in Aasen's procedure. These features contribute to saving of **n^2+2n** divisions compared to the Gauss transformation while solving **n x n** non-homogeneous linear system using **B(x)**.

In the numerical experiments section, it has been demonstrated that entries of symmetric and sparse matrices, which have some order property, are much less perturbed while decomposing by the proposed algorithm compared to Aasen's algorithm. In these situations, when almost all the entries are perturbed more quantitatively in Aasen's algorithm, those entries towards right-lower borders are only disturbed in similar scales by the present procedure.

Acknowledgements

Author wishes to acknowledge the VSSC editorial board for making necessary arrangements to review this article, especially to Dr. B. Nageswar Rao, who reviewed this article, prior to its publications. He would like to thank Prof. Dr. G. Gopalakrishnan Nair, for the valuable suggestions while doing this research work. He would also like to thank Dr. G. Madhavan Nair, former Chairman, ISRO, Dr. K. Sivan, Director, LPSC, Dr. S. Somanath, Associate Director(Projects), VSSC, Dr. M.V. Dhekhane, Associate Director(R&D), VSSC and Dr. V.K Sivarama Panicker, Project Director, ATVP, for the encouragements provided to pursue with this work and VSSC editorial board , for giving permission to publish this article.

References

[1]. B.N. PARLETT AND J.K. REID, On the Solution of a System of Linear Equations Whose Matrix is Symmetric but not Definite, BIT Numerical Mathematics, Vol.10, 1970, pp 386-397

[2]. J.O. AASEN, On the Reduction of a Symmetric Matrix to Tridiagonal Form, BIT Numerical Mathematics, Vol.11, 1971,pp 233-242.

[3]. G.H GOLUB AND CHARLES. F VAN LOAN, Matrix Computations, 4th Edition, The John Hpokins University Press, 1985.

[4]. PURUSHOTHAMAN NAIR R, A Simple and Effective Factorization Procedure for Matrices, International Journal of Maths, Game Theory Algebra, 18(2009), No.2, 145-160.

[5]. N.J. HIGHAM, Accuracy and Stability of Numerical Algorithms, 2nd edition, SIAM, Philadelphia, 2002.