하우스홀더 변환법을 이용한 토플리즈 행렬의 빠른 QR 인수분해 알고리즘

최 재 영†

요 약

본 논문에서 m×n (m≥n)인 토플리즈 행렬의 빠른 QR 인수분해 알고리즘들을 제안한다. 본 알고리즘들은 위치가 변환되어도 불변하는 (shift-invariance) 토플리즈 행렬의 특성을 효과적으로 이용하였다. 알고리즘들의 주요 변환 도구로 안정된 하우스홀더 변환과 하이퍼볼릭 하우스홀더 변환을 사용하였다. 본 알고리즘들은 O(mn)의 연산을 필요로하며, 분산메모리 병렬 컴퓨터에서 쉽게 구현될 수 있다.

Fast QR Factorization Algorithms of Toeplitz Matrices based on Stabilized/Hyperbolic Householder Transformations

Jaeyoung Choi†

ABSTRACT

We present fast QR factorization algorithms for an m×n (m≥n) Toeplitz matrix. These QR factorization algorithms are determined from the shift-invariance properties of underlying matrices. The major transformation tool is a stabilized/hyperbolic Householder transformation. The algorithms require O(mn) operations, and can be easily implemented on distributed-memory multiprocessors.

1. Introduction

An $m \times n$ $(m \geq n)$ matrix $\boldsymbol{T}$ is Toeplitz if all elements on each diagonal are equal,

$$\boldsymbol{T} = \begin{pmatrix} t_n & t_{n-1} & \cdots & t_1 \\ t_{n+1} & t_n & \cdots & t_2 \\ \vdots & \vdots & \ddots & \vdots \\ t_{m+n-1} & t_{m+n-2} & \cdots & t_m \end{pmatrix}.$$

Overdetermined Toeplitz systems arise in engineering and scientific applications, including linear prediction and discretization of integral equations. The solution to such systems is often determined via least square criteria. For the full rank system the most numerically desirable methods are based on the QR factorization of the matrix of the underlying problem [9].

Several authors have developed QR factorization algorithms for an $m \times n$ $(m \geq n)$ Toeplitz matrix in only $O(mn)$ operations [1, 7, 12]. Bojanczyk, Brent and De Hoog (BBH algorithm for short) developed a Toeplitz QR

† 종신회원 : 숭실대학교 정보과학대학 컴퓨터학부

논문접수 : 1997년 11월 5일, 심사완료 : 1998년 2월 18일

factorization algorithm based on the shift invariance property of Toeplitz matrices [1]. Cybenko developed another QR factorization of Toeplitz matrices [7] from the lattice algorithm [6]. This algorithm computes an inverse orthogonal factorization $\boldsymbol{T}\cdot\boldsymbol{R}^{1} = \boldsymbol{Q}$ instead of computing $\boldsymbol{T} = \boldsymbol{Q}\cdot\boldsymbol{R}$. Cybenko and Berry [8] described how to compute a triangular decomposition of Hermittian matrices with small displacement structure using hyperbolic Householder transformation [11]. For a Toeplitz matrix $\boldsymbol{T}$, the displacement rank of $\boldsymbol{A} = \boldsymbol{T}^T\cdot\boldsymbol{T}$ is bounded by four and the method presented in [8] can be used to compute the Cholesky factor of $\boldsymbol{A}$.

In this paper, we present new QR factorization algorithms of Toeplitz matrices. Our derivation uses the same partitioning presented in [1], but the new algorithm is based on hyperbolic Householder transformations [11]. And the proposed algorithms can be easily implemented on distributed-memory multiprocessors.

2. Hyperbolic Householder Transformations

Let $\boldsymbol{\Phi}$ be a diagonal matrix with diagonal entries +1 and -1. A transformation $\boldsymbol{W}$ is called hyperbolic with respect to $\boldsymbol{\Phi}$ if and only if

$$\boldsymbol{W}\cdot\boldsymbol{\Phi}\cdot\boldsymbol{W}^T = \boldsymbol{\Phi}. \qquad (1)$$

A hyperbolic Householder matrix $\boldsymbol{H}$ is a matrix of the form,

$$H = \boldsymbol{\Phi} - 2\,\boldsymbol{v}\cdot\boldsymbol{v}^T / (\boldsymbol{v}^T\cdot\boldsymbol{\Phi}\cdot\boldsymbol{v})$$

where $\boldsymbol{v}$ is any vector for which $\boldsymbol{v}^T\cdot\boldsymbol{\Phi}\cdot\boldsymbol{v} \neq 0$. Such a matrix is Hermitian and hyperbolic with respect to $\boldsymbol{\Phi}$.

Let $\boldsymbol{u}$ be a vector such that

$$\|\boldsymbol{u}\|_{\Phi}^2 = \boldsymbol{u}^T\cdot\boldsymbol{\Phi}\cdot\boldsymbol{u} \neq 0.$$

Then the choice $\boldsymbol{v} = \boldsymbol{u} \pm \|\boldsymbol{u}\|_{\Phi}\cdot\boldsymbol{e}_i (1 \leq i \leq n)$ guarantees $\boldsymbol{H}\cdot\boldsymbol{u} = \mp\|\boldsymbol{u}\|_{\Phi}\boldsymbol{e}_i$, i.e., $\boldsymbol{H}$ can compress the hyperbolic norm of the vector $\boldsymbol{u}$ into the i-th component of $\boldsymbol{u}$.

The problem of rank-p updating and rank-q downdating is defined as follows: Given a $p \times n$ matrix $\boldsymbol{Y}$, $q \times n$ matrix $\boldsymbol{Z}$, and an upper triangular matrix $\boldsymbol{R}$ such that $\boldsymbol{A}^T\cdot\boldsymbol{A} = \boldsymbol{R}^T\cdot\boldsymbol{R} + \boldsymbol{Y}^T\cdot\boldsymbol{Y} - \boldsymbol{Z}^T\cdot\boldsymbol{Z}$ is positive definite, find the Cholesky factor of $\boldsymbol{A}$, which can be computed by transforming the matrix $(\boldsymbol{R}^T\ \boldsymbol{Y}^T\ \boldsymbol{Z}^T)^T$ into the upper triangular matrix by multiplying it by a sequence of hyperbolic Householder transformations with respect to $\boldsymbol{\Phi}$. All subsequent algorithms presented in the paper are derived from this property. For properties of the hyperbolic Householder transforms, [11].

3. QR Factorization Algorithms for Toeplitz Matrices

This section presents the new algorithms based on the stabilized Householder transformation [4] and hyperbolic Householder transformations as well as Bojanczyk, Brent, and de Hoog's algorithm (BBH algorithm for short) [1]. The initial partitionings of $\boldsymbol{T}$ and $\boldsymbol{R}$ are the same for both algorithms, but the procedures to compute $\boldsymbol{R}$ and $\boldsymbol{Q}$ depend on transformations. We call the algorithm with one Givens rotations and one rank-2 stabilized Householder transformation *SHT algorithm*, and one rank-3 hyperbolic Householder reflection *HHT algorithm*.

3.1 BBH Algorithm

Let $\boldsymbol{T}$ be a full rank $m \times n$ $(m \geq n)$ Toeplitz matrix. $\boldsymbol{T}$ can be partitioned in two ways using the shift invariance property of Toeplitz matrices:

$$\boldsymbol{T} = \begin{pmatrix} T_{1,1} & \boldsymbol{T}_{1,2:n} \\ \boldsymbol{T}_{2:m,1} & \boldsymbol{T}_{-1} \end{pmatrix} = \begin{pmatrix} T_{1,1} & \boldsymbol{y}^T \\ \boldsymbol{T}_{2:m,1} & \boldsymbol{T}_{-1} \end{pmatrix} \qquad (2)$$

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{T}_{-1} & \boldsymbol{T}_{1:m-1,n} \\ \boldsymbol{T}_{m,1:n-1} & T_{m,n} \end{pmatrix} = \begin{pmatrix} \boldsymbol{T}_{-1} & \boldsymbol{T}_{1:m-1,n} \\ \boldsymbol{x}^T & T_{m,n} \end{pmatrix} \qquad (3)$$

where we use the MATLAB notation: $\boldsymbol{T}_{1,2:n} = (t_{1,2}, t_{1,3}, \ldots, t_{1,n})^T$. $\boldsymbol{T}_{2:m,1} = (t_{2,1}, t_{3,1}, \ldots, t_{m,1})^T$. $\boldsymbol{T}_{-1}$ is an $(m-1)\times(n-1)$ principal submatrix of $\boldsymbol{T}$ and $\boldsymbol{x}$, $\boldsymbol{y}$ are $(n-1)$ vectors such that $\boldsymbol{x}^T = \boldsymbol{T}_{m,1:n-1}$, $\boldsymbol{y}^T = \boldsymbol{T}_{1,2:n}$, respectively. Let $\boldsymbol{R}$ be an upper triangular matrix from the QR factorization of $\boldsymbol{T}$. $\boldsymbol{R}$ is also partitioned in two ways:

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,2:n} \\ 0 & \boldsymbol{R}_b \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{z}^T \\ 0 & \boldsymbol{R}_b \end{pmatrix}, \quad (4)$$

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}_t & \boldsymbol{R}_{1:n-1,n} \\ \boldsymbol{0}^T & \boldsymbol{R}_{n,n} \end{pmatrix} \quad (5)$$

where $\boldsymbol{R}_t$ and $\boldsymbol{R}_b$ are $(n-1)\times(n-1)$ principal top and bottom submatrices of $\boldsymbol{R}$, respectively, and $\boldsymbol{z}$ is an $(n-1)$ vector such that $\boldsymbol{z}^T = \boldsymbol{R}_{1,2:n}$.

From the following relation of $\boldsymbol{T}$ and $\boldsymbol{R}$

$$\boldsymbol{T}^T \cdot \boldsymbol{T} = \boldsymbol{R}^T \cdot \boldsymbol{R},$$

we can get the main relationship of $\boldsymbol{R}_t$ and $\boldsymbol{R}_b$ by replacing $\boldsymbol{T}$ and $\boldsymbol{R}$ with Eq.(2) and Eq.(4), and by substituting $\boldsymbol{T}_{-1}$ by Eq.(3) and Eq.(5).

$$\boldsymbol{R}_b{}^T \cdot \boldsymbol{R}_b = \boldsymbol{R}_t{}^T \cdot \boldsymbol{R}_t + \boldsymbol{y} \cdot \boldsymbol{y}^T - \boldsymbol{x} \cdot \boldsymbol{x}^T - \boldsymbol{z} \cdot \boldsymbol{z}^T \quad (6)$$

where

$$\begin{aligned} \boldsymbol{z}^T &= (T_{1,1}\, \boldsymbol{y} + \boldsymbol{T}_{2:m,1}\, \boldsymbol{T}_{-1}) \,/\, (\, \boldsymbol{T}_{1,1}{}^2 + \boldsymbol{T}_{2:m,1}{}^T\, \boldsymbol{T}_{2:m,1})^{\frac{1}{2}} \\ &= \boldsymbol{T}_{:,1}{}^T\, \boldsymbol{T}_{:,2:n} \,/\, (\, \boldsymbol{T}_{:,1}{}^T\, \boldsymbol{T}_{:,1})^{\frac{1}{2}} \quad (7) \end{aligned}$$

Eq.(6) says that matrix $\boldsymbol{R}_b$ is obtained from the matrix $\boldsymbol{R}_t$ following three rank-1 modifications. When implemented as a rank-1 update followed by two rank-1 downdates, Eq.(6) gives us a means of computing the k-th row of $\boldsymbol{R}_b$ from the k-th row of $\boldsymbol{R}_t$. Since the first row of $\boldsymbol{R}_t$ is defined by Eq.(7) and the k-th row of $\boldsymbol{R}_b$ is identical to the $(k+1)$-st row of $\boldsymbol{R}_t$, we have a recursion for calculating the rows of $\boldsymbol{R}$.

An $(m+1)\times(n-1)$ augmented matrix $\overline{\boldsymbol{T}}$ is formed to compute the orthogonal factor $\boldsymbol{Q}$. For details of the algorithm, see [1]. This algorithm needs $mn + 6n$^2 $+ O(n)$ multiplications to compute $\boldsymbol{R}$ and $13mn + 6n$^2 $+ O(n)$ multiplications to compute both $\boldsymbol{R}$ and $\boldsymbol{Q}$.

3.2 SHT Algorithm

This algorithm uses one Givens rotations and one rank-2 stabilized Householder reflections instead of three Givens rotations, and it starts with the same partitionings of $\boldsymbol{T}$ and $\boldsymbol{R}$ as the BBH algorithm. We rewrite Eq.(6) as follows:

$$\boldsymbol{R}_1{}^T \cdot \boldsymbol{R}_1 = \boldsymbol{R}_t{}^T \cdot \boldsymbol{R}_t + \boldsymbol{y}^T \cdot y$$

$$\boldsymbol{R}_b{}^T \cdot \boldsymbol{R}_b = \boldsymbol{R}_1{}^T \cdot \boldsymbol{R}_1 - \boldsymbol{x}^T \cdot x - \boldsymbol{z}^T \cdot z,$$

Define G and H by the following relations

$$G \cdot \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{R}_t \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0}^T \end{pmatrix} \quad (8)$$

$$H \cdot \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{z} \\ \boldsymbol{R}_b \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0}^T \\ \boldsymbol{0}^T \end{pmatrix} \quad (9)$$

where G is a series of products of plane rotations to produce the upper triangular matrix form of $\boldsymbol{R}_1$, and H is a Householder reflection to make the upper triangular matrix form $\boldsymbol{R}_b$.

Assume we know an $m\times n$ matrix $\boldsymbol{Q}$. Define $\overline{\boldsymbol{T}}$ and two $(n+1)\times(m+1)$ orthogonal matrices $\widehat{Q}_2$ and $\widetilde{Q}_2$ as:

$$\overline{\boldsymbol{T}} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{T}_{-1} \\ \boldsymbol{x} \end{pmatrix} = \begin{pmatrix} \boldsymbol{y} \\ \\ \boldsymbol{C} \end{pmatrix} = \begin{pmatrix} \boldsymbol{D} \\ \\ \boldsymbol{x} \end{pmatrix},$$

$$\widehat{\boldsymbol{Q}}_2 = \begin{pmatrix} 1 & \boldsymbol{0}^T \\ 0 & \boldsymbol{Q}^T \end{pmatrix},$$

$$\widehat{\boldsymbol{Q}}_2 = \begin{pmatrix} \boldsymbol{0}^T & 1 \\ \boldsymbol{Q}^T & \boldsymbol{0} \end{pmatrix}.$$

The products of $\widehat{Q}_2\, \overline{\boldsymbol{T}}$ and $\widehat{Q}_2\, \overline{\boldsymbol{T}}$ are

$$\widehat{\boldsymbol{Q}}_2\, \overline{\boldsymbol{T}} = \widehat{\boldsymbol{Q}}_2 \cdot \begin{pmatrix} \boldsymbol{y} \\ \\ \boldsymbol{C} \end{pmatrix} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{R}_t \\ \boldsymbol{0}^T \end{pmatrix}, \quad (10)$$

$$\tilde{\boldsymbol{Q}}_2\,\overline{\boldsymbol{T}} = \tilde{\boldsymbol{Q}}_2 \cdot \begin{pmatrix} \boldsymbol{D} \\ \\ \boldsymbol{x} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{z} \\ \boldsymbol{R}_b \end{pmatrix}. \qquad (11)$$

Define $\hat{\boldsymbol{G}}$ as

$$\hat{\boldsymbol{G}} = \begin{pmatrix} \boldsymbol{G} & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 \end{pmatrix}.$$

From the relations of Eq.(8) and Eq.(10),

$$\hat{\boldsymbol{G}} \cdot \hat{\boldsymbol{Q}}_2\,\overline{\boldsymbol{T}} = \begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0}^T \\ \boldsymbol{0}^T \end{pmatrix}, \qquad (12)$$

and from the relations of Eq.(9) and Eq.(11),

$$\boldsymbol{H} \cdot \tilde{\boldsymbol{Q}}_2\,\overline{\boldsymbol{T}} = \begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0}^T \\ \boldsymbol{0}^T \end{pmatrix} \qquad (13)$$

By comparing Eq.(12) with Eq.(13), the following relationship is acquired between the first *(n-1)* rows of $\widehat{G}\widehat{Q}_2$ and $H\widetilde{Q}_2$:

$$\hat{\boldsymbol{G}}\ \hat{\boldsymbol{Q}}_2 = \boldsymbol{H}\ \tilde{\boldsymbol{Q}}_2. \qquad (14)$$

Eq.(8) and Eq.(9) show a way to compute the upper triangular matrix $\boldsymbol{R}$ and Eq.(14) forms a base for the algorithm which calculates the orthogonal matrix $\boldsymbol{Q}$. This algorithm needs $mn + 5n^2 + O(n)$ multiplications to compute $\boldsymbol{R}$ and $11mn + 5n^2 + O(n)$ multiplications to compute both $\boldsymbol{R}$ and $\boldsymbol{Q}$.

3.3 HHT Algorithm

This algorithm uses one rank-3 hyperbolic Householder reflection, and it starts with the same partitionings of $\boldsymbol{T}$ and $\boldsymbol{R}$ as the BBH algorithm. Let

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{x}^T \\ \boldsymbol{z}^T \\ \boldsymbol{R}_b \\ \boldsymbol{y}^T \end{pmatrix} \text{ and } \boldsymbol{\Phi} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \boldsymbol{I}_{n-1} & \\ & & & -1 \end{pmatrix}$$

We can rewrite Eq.(6) as

$$\boldsymbol{R}_t^T\ \boldsymbol{R}_t = \boldsymbol{R}_b^T\ \boldsymbol{R}_b + \boldsymbol{x}\ \boldsymbol{x}^T - \boldsymbol{y}\ \boldsymbol{y}^T = \boldsymbol{B}^T\ \boldsymbol{\Phi}\ \boldsymbol{B} \qquad (15)$$

Then,

$$\boldsymbol{F} \cdot \boldsymbol{B} = \boldsymbol{F} \cdot \begin{pmatrix} \boldsymbol{x}^T \\ \boldsymbol{z}^T \\ \boldsymbol{R}_b \\ \boldsymbol{y}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_t \\ \boldsymbol{0}^T \\ \boldsymbol{0}^T \\ \boldsymbol{0}^T \end{pmatrix}, \qquad (16)$$

where $\boldsymbol{F}$ is a product of n hyperbolic Householder reflections F_i, $1 \le i \le n$, with respect to $\boldsymbol{\Phi}$ that transforms $\boldsymbol{B}$ into the upper triangular matrix form $\boldsymbol{R}_t$.

Let the matrix $\boldsymbol{T}$ be partitioned as follows:

$$\boldsymbol{T} = (\,\boldsymbol{C} \quad \boldsymbol{T}_{:,n}) = \begin{pmatrix} \boldsymbol{T}_{-1} & \boldsymbol{T}_{:,n} \\ \boldsymbol{x}^T & \end{pmatrix},$$

$$\boldsymbol{T} = (\,\boldsymbol{T}_{:,1} \quad \boldsymbol{D}\,) = \begin{pmatrix} \boldsymbol{T}_{:,1} & \boldsymbol{y}^T \\ & \boldsymbol{T}_{-1} \end{pmatrix}.$$

We want to find an $m \times n$ matrix $\boldsymbol{Q}$, with orthogonal columns, such that

$$\boldsymbol{Q}^T \cdot \boldsymbol{T} = \boldsymbol{Q}^T \cdot (\,\boldsymbol{C} \quad \boldsymbol{T}_{:,n}) = \begin{pmatrix} \boldsymbol{R}_t & \boldsymbol{R}_{1:n-1,n} \\ \boldsymbol{0}^T & \boldsymbol{R}_{n,n} \end{pmatrix}, \qquad (17)$$

$$\boldsymbol{Q}^T \cdot \boldsymbol{T} = \boldsymbol{Q}^T \cdot (\,\boldsymbol{T}_{:,1} \quad \boldsymbol{D}\,) = \begin{pmatrix} R_{1,1} & \boldsymbol{z}^T \\ 0 & \boldsymbol{R}_b \end{pmatrix}. \qquad (18)$$

Assume for a moment that we know an $m \times n$ matrix $\boldsymbol{Q}$ with orthonormal columns such that

$$\boldsymbol{Q}^T \boldsymbol{C} = \begin{pmatrix} \boldsymbol{R}_t \\ \boldsymbol{0}^T \end{pmatrix}. \qquad (19)$$

$$\boldsymbol{Q}^T \boldsymbol{D} = \begin{pmatrix} \boldsymbol{z}^T \\ \boldsymbol{R}_b \end{pmatrix}. \qquad (20)$$

Define an *(m+2)×(n-1)* augmented matrix $\check{\boldsymbol{T}}$ and two *(n+2)×(m+2)* hypernormal matrices $\hat{\boldsymbol{H}}$ and $\tilde{\boldsymbol{H}}$ as

$$\check{\boldsymbol{T}} = \begin{pmatrix} \boldsymbol{y}^T \\ \boldsymbol{T}_{-1} \\ \boldsymbol{x}^T \\ \boldsymbol{y}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}^T \\ \boldsymbol{C} \\ \\ \boldsymbol{y}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{D} \\ \\ \boldsymbol{x}^T \\ \boldsymbol{y}^T \end{pmatrix}.$$

$$\hat{\boldsymbol{H}} = \begin{pmatrix} \boldsymbol{f} & \boldsymbol{Q}^T & -\boldsymbol{f} \\ \boldsymbol{h} & \boldsymbol{g}^T & -\boldsymbol{h} \\ -\boldsymbol{h} & \boldsymbol{g}^T & \boldsymbol{h} \end{pmatrix},$$

$$\tilde{\boldsymbol{H}} = \begin{pmatrix} \boldsymbol{0}^T & 1 & 0 \\ \boldsymbol{Q}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 & 1 \end{pmatrix},$$

where $\hat{\boldsymbol{H}}$ and $\tilde{\boldsymbol{H}}$ satisfy the property of hypernormality of Eq.(1). Form two products

$\hat{\boldsymbol{H}}\check{\boldsymbol{T}}$ and $\widetilde{\boldsymbol{H}}\check{\boldsymbol{T}}$ from the definition of Q in Eq.(19) and Eq.(20),

$$\hat{\boldsymbol{H}}\cdot\check{\boldsymbol{T}} = \hat{\boldsymbol{H}}\cdot\begin{pmatrix} \boldsymbol{y}^T \\ \boldsymbol{C} \\ \\ \boldsymbol{y}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{Q}^T\boldsymbol{C} \\ \\ \boldsymbol{0}^T \\ \boldsymbol{0}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_t \\ \boldsymbol{0}^T \\ \boldsymbol{0}^T \\ \boldsymbol{0}^T \end{pmatrix}, \qquad (21)$$

$$\widetilde{\boldsymbol{H}}\cdot\check{\boldsymbol{T}} = \widetilde{\boldsymbol{H}}\cdot\begin{pmatrix} \boldsymbol{D} \\ \\ \boldsymbol{x}^T \\ \boldsymbol{y}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}^T \\ \boldsymbol{Q}^T\boldsymbol{D} \\ \\ \boldsymbol{y}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}^T \\ \boldsymbol{z}^T \\ \boldsymbol{R}_b \\ \boldsymbol{y}^T \end{pmatrix} \qquad (22)$$

From Eq.(21) and Eq.(22) with Eq.(16), the following relation is satisfied between the first (n-1) rows.

$$\hat{\boldsymbol{H}}\,\check{\boldsymbol{T}} = \boldsymbol{F}\cdot\widetilde{\boldsymbol{H}}\,\check{\boldsymbol{T}}, \qquad (23)$$

that is,

$$\begin{aligned} (\ \boldsymbol{I}_n\ \ \boldsymbol{0}\ \ \boldsymbol{0}\ )\cdot\begin{pmatrix} \boldsymbol{f} & \boldsymbol{Q}^T & -\boldsymbol{f} \\ \boldsymbol{h} & \boldsymbol{g}^T & -\boldsymbol{h} \\ -\boldsymbol{h} & \boldsymbol{g}^T & \boldsymbol{h} \end{pmatrix} \\ = (\ \boldsymbol{I}_n\ \ \boldsymbol{0}\ \ \boldsymbol{0}\ )\cdot\boldsymbol{F}\cdot\begin{pmatrix} \boldsymbol{0}^T & 1 & 0 \\ \boldsymbol{Q}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 & 1 \end{pmatrix} \end{aligned}$$

At the *j*-th step, the *j*-th row of $\hat{\boldsymbol{H}}\,\check{\boldsymbol{T}}$ is compared with the *j*-th row of $\widetilde{\boldsymbol{H}}\,\check{\boldsymbol{T}}$ after being multiplied by the *j*-th hyperbolic Householder matrix $\boldsymbol{F}$ in order to get the *(j+1)*-st row of $\boldsymbol{Q}^T$. Because the first element in the *j*-th row of $\hat{\boldsymbol{H}}\,\check{\boldsymbol{T}}$, f_j (i.e. the *j*-th element of *f*) is unknown, the first element in the *(j+1)*-st row of $\boldsymbol{Q}^T$ can't be calculated. We use a primitive way to compute the first column of $\boldsymbol{Q}^T$.

The following relation is applied to get the first row of $\boldsymbol{Q}$ in each step:

$$\boldsymbol{T}_{1,j} = \boldsymbol{Q}_{1,1:j}\cdot\boldsymbol{R}_{1:j,j},$$

then,

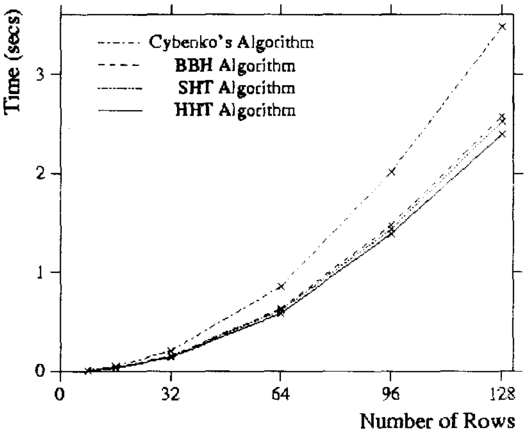
$$\boldsymbol{f}_j = \boldsymbol{Q}_{1,j} = (\ \boldsymbol{T}_{1,j} - \boldsymbol{Q}_{1,1:j-1}\cdot\boldsymbol{R}_{1:j-1,j}\ )\ /\ \boldsymbol{R}_{j,j}. \quad (24)$$

The triangular matrix $\boldsymbol{R}$ is directly computed from Eq.(16) based on the hyperbolic Householder transformations. Eq.(23) and Eq.(24) form a basis for the algorithm which computes the orthogonal matrix $\boldsymbol{Q}$. The first row of $\boldsymbol{Q}$ is obtained from Eq.(24), and the rest are computed by Eq.(23). This algorithm requires $mn + 4n^2 + O(n)$ multiplications to compute $\boldsymbol{R}$ and $9mn + 4.5n^2 + O(n)$ multiplications to compute both $\boldsymbol{R}$ and $\boldsymbol{Q}$.

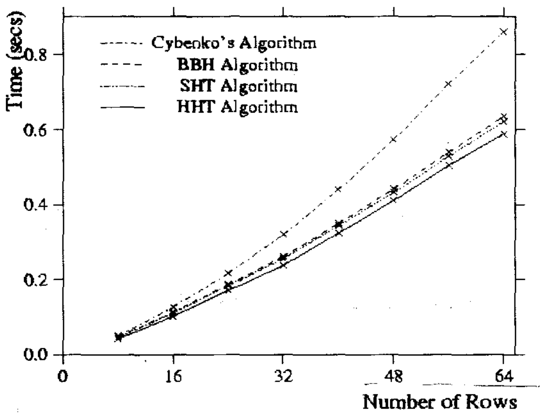
4. Implementation and Results

We implemented the QR algorithms of Toeplitz matrices based on N.Tsao's modified version of Householder transformations [13]. For implementation details, see [5]. We compared the SHT algorithm and the HHT algorithm with the BBH algorithm and the Cybenko's algorithm. The Cybenko's algorithm needs $9mn + 10.5n^2 + O(n)$ multiplications to obtain both $\boldsymbol{R}^{-1}$ and $\boldsymbol{Q}$, while BBH algorithm, SHT algorithm, and the HHT algorithm need $13mn + 6n^2 + O(n)$, $11mn + 5n^2 + O(n)$, and $9.5mn + 4.5n^2 + O(n)$ multiplications, respectively, to compute both $\boldsymbol{R}$ and $\boldsymbol{Q}$.

We measured the execution times on one node of the i860 processor. Figures 1 and 2 compare the execution times of the algorithms for square matrices and for rectangular



(Figure 1) Algorithm Comparisons for Square Matrices



(Figure 2) Algorithm Comparisons for Rectabular Matrices (No. of rows are fixed to 64)

matrices, respectively, with the number of columns fixed at $m = 64$.

For square matrices, the HHT algorithm always has the best performance while Cybenko's algorithm displays the worst performance. The HHT algorithm is about 30% faster than Cybenko's algorithm. Both the SHT algorithm and The HHT algorithm is superior to the BBH algorithm since it uses the hyperbolic Householder reflections instead of Givens rotations. The SHT algorithm can save approximately 2% of the time, and the HHT algorithm can save about 7% of the time compared to the BBH algorithm.

The HHT algorithm is always the fastest. As n grows, the HHT algorithm saves more time compared with Cybenko's algorithm, and it saves about 7% of time compared to the BBH algorithm.

5. Analysis and Conclusion

We tested those four algorithms of Toeplitz matrices using an example in Luk and Qiao's paper [10] in order to compare the accuracies of QR decomposition, orthogonalities of $\boldsymbol{Q}$ and triangularities of $\boldsymbol{R}$. These are defined as,

$$\text{Accur}(\boldsymbol{QR}) = \| \boldsymbol{T} - \boldsymbol{Q}\cdot\boldsymbol{R} \|_F \; / \; \| \boldsymbol{T} \|_F,$$
$$\text{Ortho}(\boldsymbol{Q}) = \| \boldsymbol{Q}^T\cdot\boldsymbol{Q} - \boldsymbol{I} \|_F \; / \; \| \boldsymbol{I} \|_F,$$
$$\text{Trian}(\boldsymbol{T}) = \| \boldsymbol{T}^T\cdot\boldsymbol{T} - \boldsymbol{R}^T\cdot\boldsymbol{R} \|_F \; / \; \| \boldsymbol{T}^T\cdot\boldsymbol{T} \|_F,$$

and for Cybenko's algorithm the accuracies of QR decomposition and triangularities of $\boldsymbol{R}^{-1}$ are defined by:

$$\text{Accur}'(\boldsymbol{QR}) = \| \boldsymbol{T}\cdot\boldsymbol{R}^{-1} - \boldsymbol{Q} \|_F \; / \; \| \boldsymbol{T}\cdot\boldsymbol{R}^{-1} \|_F,$$
$$\text{Trian}'(\boldsymbol{T}) = \| - \boldsymbol{I} \|_F \; / \; \| \boldsymbol{I} \|$$

As long as the test matrices are well-conditioned, the numerical errors of accuracies, orthogonalities and triangularities are negligible. The test matrix,

<Table 1> Accuracy comparisons of algorithms

Cond No.	Criteria	BBH	SHT	HHT	Cybenko
5.67e+02 (t = 1e-01)	Accuracy of QR	2.4008e-16	1.3107e-16	1.0933e-16	4.6699e-14
	Orthogonality of Q	7.8635e-12	3.3701e-12	5.5585e-12	1.4077e-14
	Triangularity of T	1.1315e-16	6.7969e-17	6.7969e-17	6.2578e-14
5.68e+04 (t = 1e-03)	Accuracy of QR	1.5924e-16	1.4679e-16	1.3802e-16	1.5882e-12
	Orthogonality of Q	4.1585e-08	4.1581e-08	8.6553e-08	1.4782e-12
	Triangularity of T	1.1772e-16	1.3931e-16	1.4420e-16	3.5950e-12
5.68e+06 (t = 1e-05)	Accuracy of QR	1.4335e-16	2.8505e-16	6.3715e-16	1.1282e-10
	Orthogonality of Q	1.3453e-03	2.2381e-03	1.3453e-03	4.7105e-10
	Triangularity of T	1.5823e-16	2.3856e-16	1.8638e-16	4.7625e-10
5.68e+08 (t = 1e-07)	Accuracy of QR	1.7716e-16	2.1937e-16	8.3823e-16	4.0752e-08
	Orthogonality of Q	4.4440e-12	4.9206e-01	4.8147e-01	2.9617e-08
	Triangularity of T	1.4149e-16	3.3337e-16	1.6729e-16	7.0308e-08

$$T = \frac{1}{27}\begin{pmatrix} 27 & 9 & 3 & -23+t \\ 9 & 27 & 9 & 3 \\ 3 & 9 & 27 & 9 \\ -23+t & 3 & 9 & 27 \end{pmatrix}$$

is positive definite but ill-conditioned, where t is a small number. The test results for the matrices are included in Table 1.

In the cases of ill-conditioned matrices, the large condition numbers of the test matrices cause the orthogonalities of $\boldsymbol{Q}$ to be lost. Since the algorithms are based on the equations $\boldsymbol{T}^T \cdot \boldsymbol{T} = \boldsymbol{R}^T \cdot \boldsymbol{R}$ and $\boldsymbol{T} = \boldsymbol{Q} \cdot \boldsymbol{R}$, the triangularities of $\boldsymbol{R}$ and accuracies of $\boldsymbol{Q} \cdot \boldsymbol{R}$ are kept in spite of the large condition numbers.

Another advantage of those SHT and HHT algorithms is that $\boldsymbol{R}$ can be computed without computing $\boldsymbol{Q}$. The required computations to find $\boldsymbol{R}$ ($mn + 5n^2 + O(n)$ for the SHT algorithm, and $mn + 4.5n^2 + O(n)$ for the HHT algorithm) for square matrices are not even half of the total computations of $\boldsymbol{R}$ and $\boldsymbol{Q}$ ($11mn + 5n^2 + O(n)$ for the SHT algorithm and $9mn + 4.5n^2 + O(n)$ for the HHT algorithm).

Bojanczyk and Choi [3] have implemented and compared the BBH algorithm and the Cybenko's algorithms on the iPSC/2 hypercube machine. The results showed that BBH algorithm is always superior to the Cybenko's algorithm, and Cybenko's algorithm is not appropriate for parallel machines because it is based on the Gram-Schmidt orthogonalization procedure, which has a large communication overhead. Since both SHT and HHT algorithms are based on the same partitioning as the BBH algorithm, they can be effectively implemented on parallel machines.

References

[1] A. W. Bojanczyk, B. T. Brent, and F. R. de Hoog, "QR Factorization of Toeplitz Matrices," Numer. Math., 49 (1986), pp.81-94.

[2] A. W. Bojanczyk and A. O. Steinhardt, "Matrix Downdating Techniques for Signal Processing," SPIE, Vol.975, Advanced Algo. & Arch. for Signal Proc.III, 1988, pp.68-75.

[3] A. W. Bojanczyk and J. Choi, "Implementations of Three QR Algorithms of the Toeplitz Matrix on a Hypercube," 4th Annual Conference of Hypercube (HCCA4), 1989.

[4] A. W. Bojanczyk and A. O. Steinhardt, "Stabilized Hyperbolic Householder Transformations," IEEE Trans. Acoust. Speech, Signal Processing, 1990.

[5] J. Choi, "Matrix-Based Signal Processing Algorithms on Distributed-Memory Multiprocessors," Ph.D. Thesis, School of Electrical Engineering, Cornell University, 1992.

[6] G. Cybenko, "A General Technique with Application to Time Series Anaysis and Signal Processing," Math. Comput., 40 (1983), pp.323-336.

[7] G. Cybenko, "Fast Toeplitz Orthogonalization Using Inner Products," SIAM, J. Sci. Statist. Comput., 8 (1987), pp.734-740.

[8] G. Cybenko and M. Berry, "Hyperbolic Householder Algorithm for Factoring Structured Matrices," SIAM, J. Matrix Anal. Appl., 11 (1990), pp.499-520.

[9] G. H. Golub and C. F. Van Loan, Matrix Computations, 2nd Edition, The Johns Hopkins University Press, Baltimore, MD, 1989.

[10] F. T. Luk and S. Qiao, "A Fast but Unstable Orthogonal Triangularization Technique for Toeplitz Matrices," J. Linear Algebra Appli., 88/89 (1987), pp.495-506.

[11] C. M. Rader and A. O. Steinhardt, "Hyperbolic Householder Transforms," IEEE Trans. Acoust. Speech, Signal Processing,

34 (1986), pp.1589-1602.

[12] D. R. Sweet, "Fast Toeplitz Orthogonalization," Numer. Math., 43 (1984), pp.1-21.

[13] N. K. Tsao, "A Note on Implementing the Householder Transformation," SIAM Journal of Numerical Analysis, 12 (1975), pp.53-58.

최 재 영

1984년 서울대학교 제어계측공학과(학사)

1986년 미국 남가주대학교 전기공학과(석사)

1991년 미국 코넬대학교 전기공학부(박사)

1992년 1월~1994년 2월 미국 국립 오크리지연구소 연구원

1994년 3월~1995년 2월 미국 테네시 주립 대학교 연구교수

1995년 3월~현재 숭실대학교 정보과학대학 컴퓨터학부 조교수

관심분야 : 병렬/분산처리, 병렬알고리즘, 초고속계산론, 시스템 소프트웨어