ON PARALLEL FAST JACOBI ALGORITHM FOR THE EIGENPROBLEM OF REAL SYMMETRIC MATRICES*

Deng Jian-xin
(Computing Center, Academia Sinica, Beijing, China)

Abstract

Jacobi algorithm has been developed for the eigenproblem of real symmetric matrices, singular value decomposition of matrices and least squares of the overdetermined system on a parallel computer. In this paper, the parallel schemes and fast algorithm are discussed, and the error analysis and a new bound are presented.

§1. Introduction

The accuracy of the classical Jacobi method [7] for the eigensystem of a real symmetric matrix is comparable with that of QR. Especially, the computed eigenvectors are almost exactly orthogonal and span a correct subspace. It is independent of the separation of the eigenvalues. For eigenvectors, QR is incomparable to the Jacobi method. If the matrix is close to the diagonal form, QR loses its advantage [8]. But in speed, the Jacobi method is too slow.

The algorithm has had several interesting developments, since Givens transformations without square roots were presented [3]. The Jacobi algorithm using Gentleman's technique has been developed, and an experimental code [1] produces a more accurate solution than the classical version. The fast algorithm requires only $4n$ multiplications for each two-sided transformations; apparently 50% of work can be reduced. But it is still not suitable for systems of which the order is larger than ten.

With the development of parallel computers, the parallel Jacobi algorithm has been given, and used to solve singular value decomposition problems and to find the least squares solution for the overdetermined system [4]-[6]. The parallel fast algorithm is nearly n times faster than the classical Jacobi method. We have reason to believe that the algorithm may be efficiently used for the parallel computing of larger order matrices. The error analysis of the Jacobi algorithm has been given by Wilkinson [7], and the error analysis of Givens transformations has been given by Gentleman [2], but they cannot be applied to the new algorithm directly, because there are some differences between the two algorithms. In fact, although their transformations are essentially orthogonal similar, the computed transformations are not in fast algorithm. In this paper, an error analysis and a new bound are presented.

* Received December 25, 1987.

§2. Fast Rotation Transformations

In the fast Jacobi algorithm a real symmetric $n \times n$ matrix A is reduced to a diagonal form by a sequence of fast plane rotations R. Let A and R be the product of three matrices respectively:

$$A = DBD, \qquad R = \tilde{D}BD^{-1} \tag{2.1}$$

where $\tilde{D} =$ diag $(\tilde{d}_i)$, $D =$ diag (d_i), $d_i > 0$, $i = 1, 2, \cdots, n$. A rotation similarity transformation in plane (p, q) of the classical method can be modified as

$$\tilde{A} = \tilde{D}\tilde{B}\tilde{D}, \qquad \tilde{B} = HBH^T \tag{2.2}$$

where the nonzero elements of H are defined by

$$h_{pq} = \alpha, \qquad h_{qp} = \beta, \quad h_{ii} = 1, \quad i = 1, 2, \cdots. \tag{2.3}$$

B differs from $\tilde{B}$ only in the rows and columns p, q, as shown in (2.2). α, β can be computed by the following equations:

$$1 = d_p^2 b_{pp} - d_q^2 b_{qq},$$

$$g = 2b_{pq}/(1 + \text{sign}(1)(1^2 + 4d_p^2 d_q^2 b_{pq}^2)^{1/2}, \tag{2.4}$$

$$\alpha = gd_q^2, \qquad \beta = -gd_p^2. \tag{2.5}$$

The diagonal elements in $\tilde{D}$ can be defined by

$$\tilde{d}_p^2 = d_p^2/(1 - \alpha\beta), \qquad \tilde{d}_q^2 = d_q^2/(1 - \alpha\beta). \tag{2.6}$$

The transformation defined by (2.1)-(2.6) saves 50% of work by comparison with the classical algorithm. Notice that $\tilde{A} = \tilde{D}\tilde{B}\tilde{D}$ is never produced before completion of the process; therefore the iteration requires only $\tilde{B}$ and $\tilde{D}$. The transformation $\tilde{B} = HBH^T$ in (2.2) is no longer orthogonal similar; because the matrix H is not orthogonal but $\tilde{D}HD^{-1}$ is orthogonal.

§3. Parallel Scheme

In the Jacobi method each transformation affects only two rows and two columns. Consider a sweep consisting of $n(n-1)/2$ transformations, which should annihilate each off-diagonal elements only once. The transformations in a sweep can be divided into $n-1$ (or n) groups. Since all transformations in each group are disjoint plane rotations, they can be simultaneous when implemented. Hence the algorithm is $n/2$ times faster than the common one. Furthermore, an improved error bound can be obtained [2]. Some partition