Journal of Computational Mathematics, Vol.25, No.5, 2007, 583–594.

FAST PARALLELIZABLE METHODS FOR COMPUTING INVARIANT SUBSPACES OF HERMITIAN MATRICES *

Zhenyue Zhang
(*Department of Mathematics, Zhejiang University, Hangzhou, China*
Email: zyzhang@zju.edu.cn)

Hongyuan Zha
(*College of Computing, Georgia Institute of Technology Atlanta, GA 30332, USA*
Email: zha@cc.gatech.edu)

Wenlong Ying
(*Department of Mathematics, Zhejiang University, Hangzhou, China*)

Abstract

We propose a *quadratically* convergent algorithm for computing the invariant subspaces of an Hermitian matrix. Each iteration of the algorithm consists of *one* matrix-matrix multiplication and *one* QR decomposition. We present an accurate convergence analysis of the algorithm without using the big O notation. We also propose a general framework based on implicit rational transformations which allows us to make connections with several existing algorithms and to derive classes of extensions to our basic algorithm with faster convergence rates. Several numerical examples are given which compare some aspects of the existing algorithms and the new algorithms.

Mathematics subject classification: 15A18, 65F05, 65F35.
Key words: Eigenvalue, Invariant subspace, Hermitian matrix, QR method, Parallelizable method.

1. Introduction

In [15] we proposed a cubically convergent algorithm for computing the two invariant subspaces of an Hermitian matrix A corresponding to the eigenvalues of A inside and outside the unit interval $[-1, 1]$, respectively. There we also presented a detailed convergence analysis which proved the cubic convergence of the algorithm. The derivation of the algorithm is inspired by the work in [1, 2, 3, 4, 6, 10, 11, 12] and the algorithm only uses matrix-matrix multiplications and QR decompositions as building blocks which are highly parallelizable primitive operations in libraries such as ScalaPack [14]. In this paper, we continue along the same line of research and concentrate on deriving new algorithms that can substantially reduce the amount of storage and the number of matrix-matrix multiplications. By exploiting the symmetry of the eigenvalue problem, we succeeded in deriving a new algorithm that employs only *one* matrix-matrix multiplication and *one* QR decomposition in each iteration. The presentation of the algorithm is the topic of Section 2. The structure of the new algorithm is extremely simple which allows us to give a much refined convergence analysis of the algorithm in Section 3. In particular, we were able to remove all of the big O expressions which were heavily used in [15]. The resulting bounds are cleaner and more concise. In Section 4, we analyze our proposed algorithm from the point of view of implicit rational transformations. This approach allows us to propose classes

* Received December 15, 2005; final revised September 1, 2006; accepted October 1, 2006.

of extensions of our basic algorithm which have higher convergence rates. To test the power of the implicit rational transformation framework, we will derive a simple version of the matrix sign function scheme from the general framework. We then discuss the relations of our new algorithms with Algorithm ISDA proposed in [1, 3, 10] and Algorithm CUBIC proposed in [15]. We focus on the accuracy of the invariant subspaces that are computed by those algorithms for a variety of numerical examples.

Remark. We want to emphasize that when the matrix A is non-Hermitian, then all the algorithms proposed in the sequel can be converted into algorithms for computing the singular subspaces of A.

2. The Algorithms

Our focus is to derive new algorithms which use as few matrix-matrix multiplications as possible in each iteration for computing an invariant subspace $\mathcal{V}_{(a,b)}$ of an Hermitian matrix $A \in \mathcal{C}^{n\times n}$ corresponding to the eigenvalues inside a preassigned interval $(a,\, b)$.[2)]

Theoretically, such an invariant subspace of A can be obtained by the following three steps. First, construct a function f that maps the complement of interval $[a,\, b]$ to zero and keep the image of $[a,\, b]$ far from zero. Second, compute the matrix function $f(A)$. Finally, compute the range space of $f(A)$ using QR algorithm column-pivoting to obtain the invariant subspace as required.

However, it is difficult to design such a function explicitly A feasible approach is to construct a function that has such properties approximately. We consider a sequence of functions $\{f_k\}$ that converges to an ideal f. One of the approaches for designing f_k is that we use a multiple composite of a fixed function g together with a scaling function ℓ,

$$f_k = \underbrace{g \circ \cdots \circ g}_{k \text{ times}} \circ \ell \equiv g^{(k)} \circ \ell \quad \text{with} \quad g^{(k)} = \underbrace{g \circ \cdots \circ g}_{k \text{ times}}, \tag{2.1}$$

where the iterative function g should be chosen such that 1) it has two invariant intervals I_1 and I_2 that cover the real space, i.e., $R = I_1 \cup I_2$, and 2) it shrinks one of the intervals, say I_1, as k increases, i.e., $g^{(k)}(I_1) \to \{\alpha\}$ as $k \to \infty$. The scaling function ℓ maps the inside of (a, b) into I_2 and the outside to I_1. This approach leads to an iterative method for computing the invariant subspace as follows.

Basic Iteration for Computing an Invariant Subspace.

1. Initial scaling. Set $B_0 = \ell(A)$.
2. Iteration. For $k = 0, \ldots,$ compute $B_{k+1} = g(B_k)$ until convergence.
3. Column-pivoting QR. Compute an orthogonal basis matrix of the range space of B_p for a convergent iterator B_p.

Obviously, $B_k = f_k(A)$ with f_k defined in (2.1). For ease of computation, the iterative function g should be chosen such that the matrix function $g(A)$ can be computed easily. In

[2)] We assume that a and b are not eigenvalues of A.