Electronic Transactions on Numerical Analysis.
Volume 3, pp. 116-149, December 1995.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

# ON THE CORRECTNESS OF SOME BISECTION-LIKE PARALLEL EIGENVALUE ALGORITHMS IN FLOATING POINT ARITHMETIC*

JAMES W. DEMMEL †, INDERJIT DHILLON‡, AND HUAN REN§

**Abstract.** Bisection is a parallelizable method for finding the eigenvalues of real symmetric tridiagonal matrices, or more generally symmetric acyclic matrices.

Ideally, one would like an implementation that was simultaneously parallel, load balanced, devoid of communication, capable of running on networks of heterogenous workstations, and of course correct. But this is surprisingly difficult to achieve.

The reason is that bisection requires a function Count($x$) which counts the number of eigenvalues less than $x$. In exact arithmetic Count($x$) is a monotonic increasing function of $x$, and the logic of the algorithm depends on this. However, monotonicity can fail, and incorrect eigenvalues may be computed, because of roundoff or as a result of using networks of heterogeneous parallel processors. We illustrate this problem, which even arises in some serial algorithms like the EISPACK routine `bisect`, and show several ways to fix it. One of these ways has been incorporated into the ScaLAPACK library.

**Key words.** symmetric eigenvalue problem, parallel algorithms, monotonicity, correctness, floating point.

**AMS subject classifications.** 65F15, 65Y05.

*Received September 26, 1995. Accepted for publication December 7, 1995. Communicated by J. J. Dongarra.

†Computer Science Division and Department of Mathematics, University of California, Berkeley, California 94720. This material is based in part upon work supported by the Advanced Research Projects Agency contract No. DAAL03-91-C-0047 (via subcontract No. ORA4466.02 with the University of Tennessee), the Department of Energy grant No. DE-FG03-94ER25219, and contract No. W-31-109-Eng-38 (via subcontract Nos. 20552402 and 941322401 with Argonne National Laboratory), the National Science Foundation grant Nos. ASC-9313958, ASC-9005933, CCR-9196022, and NSF Infrastructure Grant Nos. CDA-8722788 and CDA-9401156. The information presented here does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

‡Computer Science Division, University of California, Berkeley, California 94720. The author was supported by DARPA grant DAAL03-91-C-0047 via a subcontract from the University of Tennessee.

§Department of Mathematics, University of California, Berkeley, California 94720. The author was supported by DARPA grant DAAL03-91-C-0047 via a subcontract from the University of Tennessee.