Parallel Solution of Generalized Complex Symmetric Eigenproblems

Hannes Schabauer[a] Wilfried N. Gansterer[a]
Andrew G. Sunderland[b] Christoph Pacher[c]

[a]Research Lab Computational Technologies and Applications, University of Vienna, Austria
[b]Computational Science and Engineering Department, Science and Technology Facilities Council, UK
[c]Safety & Security Department, AIT Austrian Institute of Technology GmbH, Austria

Abstract

We investigate a method for efficiently solving a complex symmetric (*non-Hermitian*) generalized eigenvalue problem (EVP) $Ax = \lambda Bx$ in parallel. An evaluation featuring up to 1024 CPU-cores evidences encouraging runtime behavior.

1 Introduction

Our motivation arises in an application from optoelectronics, where reduced accuracy requirements provide an opportunity for trading accuracy for execution time [5, 3]. The conventional approach, as implemented for example in `zgeev` (LAPACK), is to treat complex symmetric problems as general complex and therefore abstain from utilizing the structural symmetry for efficient codes to solve the corresponding EVP. The proposed procedure follows the cardinal steps (a) factorization $B \mapsto LL^\top$, (b) reduction to standard form $My = \lambda y$, (c) tridiagonalization to $Tz = \lambda z$, (d) computing eigenvalues λ_k and eigenvectors z_k (if desired) of the resulting tridiagonal problem, and (e) backtransformation of eigenvectors from z to x if desired (confer, for example, [2]). In the serial case, a full solver has already been developed, while in the parallel case presently only steps (a), (b), and (c) have been evaluated.

2 Methodology and Implementation

The presented paper discusses the cardinal steps factorization, reduction to standard EVP, and non-splitting tridiagonalization (see Section 1) in terms of their runtime behavior. This solver is the consequent development of our work focusing on splitting and non-splitting tridiagonalization of such EVPs [5, 4]. Due to estimated higher parallelization potential, the non-splitting tridiagonalization approach has been elaborated subsequently.

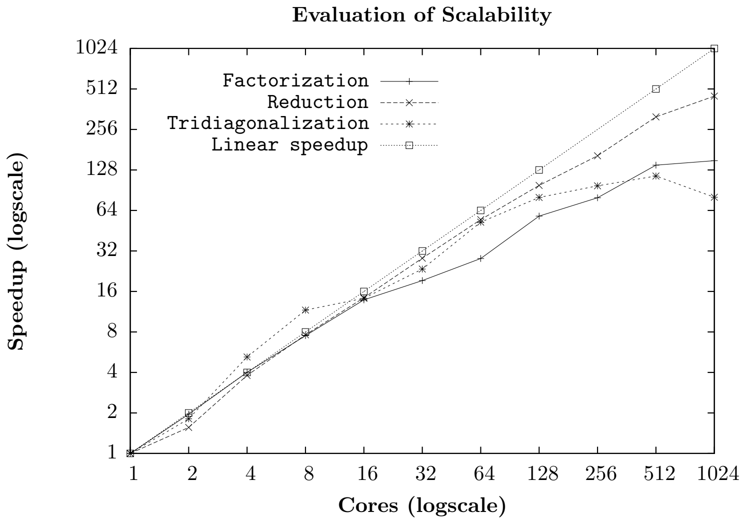

Figure 1: Speedups for order $n = 8192$ on HPCx, featuring 2 to 1024 CPU-cores. Evaluated codes include the symmetric indefinite factorization $B \mapsto LL^\top$, reduction from generalized EVP $Ax = \lambda Bx$ to standard EVP $My = \lambda y$, and non-splitting tridiagonalization to yield $Tz = \lambda z$.

ScaLAPACK-style MPI-based parallel codes have been developed and evaluated. Parallel codes were run on the supercomputer HPCx, which is a cluster featuring 168 IBM eServer 575 logical partitions (LPARs) offering a total of 2560 IBM Power5 processors [1]. All codes were compiled with the IBM xlf compiler and optimized with compiler option O3, BLAS codes linked against IBM ESSL.

3 Results and Conclusions

All parts that have been evaluated, were being tested for matrix sizes up to 8192, running on up to 1024 cores on the STFC machine HPCx. See Figure 1 for the corresponding relative speedups, where speedup $S(n)$ is defined as $\frac{T_1}{T_n}$, where T_1 is the execution time of the parallel code on a single processor or processor core, and T_n is the execution time on n processors or processor cores (confer [6]). In the serial case, the biggest share of measured runtimes is consumed by the tridiagonalization. This observation is evidenced by unpublished work of the authors, but can also be seen in analogy to the parallel studies in [7].

Evaluations reveal that the scalability of the proposed method is reasonably good on the utilized machine. In accordance with the serial version, the part consuming the highest share of the time is the parallel tridiagonalization. The limit in terms of useful utilization of CPU-cores is reached, when the speedup decreases with a higher number of cores. For example, the parallel tridiagonalization on 1024 cores was found to take more than time than the tridiagonalization residing on 512 cores; therefore the utilization of 1024 cores for the parallel non-splitting tridiagonalization of order $n = 8192$ is to be avoided. Depending on the order of the problem, the parallel scaling limits of evaluated parallel codes is about 512 to 1024 where the overall efficiency of the developed codes decreases quickly. All in all, the proposed targets have been reached and the research on complex symmetric eigenproblems has substantially benefited from this research visit.

4 Acknowledgment

The work has been performed under the HPC-EUROPA2 project (project number: 228398) with the support of the European Commission – Capacities Area – Research Infrastructures and has been partly supported by the CPAMMS project (FS397001) in the research focus area "Computational Science" at the University of Vienna.

References

[1] Mike Ashworth, Ian J. Bush, Martyn F. Guest, Andrew G. Sunderland, Stephen Booth, Joachim Hein, Lorna Smith, Kevin Stratford, and Alessandro Curioni. HPCx: towards capability computing. *Concurrency and Computation: Practice and Experience*, 17(10):1329–1361, 2005. DOI 10.1002/cpe.895.

[2] Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, 2000.

[3] Norman Finger, Christoph Pacher, and Winfried Boxleitner. Simulation of Guided-Wave Photonic Devices with Variational Mode-Matching. In *28^{th} International Conference on the Physics of Semiconductors (ICPS)*, volume 893, pages 1493–1494. American Institute of Physics, 2007. DOI 10.1063/1.2730473.

[4] Wilfried N. Gansterer, Andreas R. Gruber, and Christoph Pacher. Non-Splitting Tridiagonalization of Complex Symmetric Matrices. In *9^{th} International Conference of Computational Science (ICCS)*, LNCS, pages 481–490. Springer, 2009. DOI 10.1007/978-3-642-01970-8_47.

[5] Wilfried N. Gansterer, Hannes Schabauer, Christoph Pacher, and Norman Finger. Tridiagonalizing Complex Symmetric Matrices in Waveguide Simulations. In *8^{th} International Conference on Computational Science (ICCS)*, LNCS, pages 945–954. Springer, 2008. DOI 10.1007/978-3-540-69384-0_99.

[6] Xian-He Sun and Lionel M. Nie. Another View on Parallel Speedup. In *Supercomputing (SC)*, pages 324–333. IEEE, 1990. DOI 10.1109/SUPERC.1990.130037.

[7] Robert C. Ward and Yihua Bai. Performance of Parallel Eigensolvers on Electronic Structure Calculations II. Technical Report UT-CS-06-572, University of Tennessee at Knoxville, 2006. http://www.cs.utk.edu/~library/TechReports/2006/ut-cs-06-572.pdf.