Parallel Computing 000 (2018) 1–19

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

Accelerated Cyclic Reduction: A distributed-memory fast solver for structured linear systems

Gustavo Chávez [a,*], George Turkiyyah [b], Stefano Zampini [a], Hatem Ltaief [a], David Keyes [a]

[a] Extreme Computing Research Center (ECRC), King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia
[b] Department of Computer Science, American University of Beirut (AUB), Beirut, Lebanon

ARTICLE INFO

Article history:
Received 30 December 2016
Revised 2 October 2017
Accepted 9 December 2017
Available online xxx

Keywords:
Cyclic reduction
Hierarchical matrices
Fast direct solvers
Elliptic equations

ABSTRACT

We present Accelerated Cyclic Reduction (ACR), a distributed-memory fast solver for rank-compressible block tridiagonal linear systems arising from the discretization of elliptic operators, developed here for three dimensions. Algorithmic synergies between Cyclic Reduction and hierarchical matrix arithmetic operations result in a solver that has $O(k\ N\log N\ (\log N + k^2))$ arithmetic complexity and $O(k\,N\log N)$ memory footprint, where N is the number of degrees of freedom and k is the rank of a block in the hierarchical approximation, and which exhibits substantial concurrency. We provide a baseline for performance and applicability by comparing with the multifrontal method with and without hierarchical semi-separable matrices, with algebraic multigrid and with the classic cyclic reduction method. Over a set of large-scale elliptic systems with features of nonsymmetry and indefiniteness, the robustness of the direct solvers extends beyond that of the multigrid solver, and relative to the multifrontal approach ACR has lower or comparable execution time and size of the factors, with substantially lower numerical ranks. ACR exhibits good strong and weak scaling in a distributed context and, as with any direct solver, is advantageous for problems that require the solution of multiple right-hand sides. Numerical experiments show that the rank k patterns are of $O(1)$ for the Poisson equation and of $O(n)$ for the indefinite Helmholtz equation. The solver is ideal in situations where low-accuracy solutions are sufficient, or otherwise as a preconditioner within an iterative method.

1. Introduction

Cyclic reduction, introduced in [1], is a direct solver for tridiagonal linear systems. It is effective for the solution of (block) Toeplitz and (block) tridiagonal matrices that arise from the discretization of elliptic PDEs [2,3]. For the constant-coefficient Poisson equation, since each of the blocks of the discretized system is Fourier diagonalizable, cyclic reduction can be used in combination with the fast Fourier transform (FFT) to deliver optimal complexity, as proposed in the FACR method [4]. However, in the presence of variable coefficients, the FFT-enabled version of cyclic reduction can not be used. The purpose

* Corresponding author.
E-mail addresses: gustavo.chavezchavez@kaust.edu.sa (G. Chávez), gt02@aub.edu.lb (G. Turkiyyah), stefano.zampini@kaust.edu.sa (S. Zampini), hatem.ltaief@kaust.edu.sa (H. Ltaief), david.keyes@kaust.edu.sa (D. Keyes).

https://doi.org/10.1016/j.parco.2017.12.001

of this work is to address the time and memory complexity growth in the presence of heterogeneous blocks with a variant called Accelerated Cyclic Reduction (ACR). The main observation is that elliptic operators have a hierarchical structure of off-diagonal blocks that can be approximated with low-rank matrices. Thus we approximate appropriate blocks of the initially sparse matrix with hierarchical matrices and operate on these blocks with hierarchical matrix arithmetics, instead of the usual dense operations, to obtain a direct solver of log-linear arithmetic and memory complexities. This philosophy follows recent work discussed below, but to our knowledge, this is the first demonstration of the utility of complexity-reducing hierarchical substitution in the context of cyclic reduction.

Cyclic reduction can be thought of as a direct Gaussian elimination on a permuted system that recursively computes the Schur complement of half of the unknowns until a single block remains or the system is small enough to be inverted directly. Schur complement computations have a complexity that is dominated by the cost of the inverse; by applying a red/black re-ordering of the unknowns, the linear system separates into two halves with block diagonal structure. This decoupling addresses the most expensive step of the Schur complement computation regarding operation complexity and does so in a way that launches independent subproblems. This concurrency feature, in the form of recursive bisection, can be naturally implemented in a distributed-memory parallel environment. The stability of the block cyclic reduction has been studied in [5], where the author presents error bounds for strictly and nonstrictly diagonally dominant matrices.

In order to simplify the description of the algorithm, in this work we consider structured linear systems arising from the discretizations or scalar PDEs on three-dimensional Cartesian grids. For three-dimensional problems of size $N = n^3$, where n is the number of discretization points in the linear dimension of the target domain, the synergy of cyclic reduction and hierarchical matrices leads to a parallel fast direct solver of $O(k\,N \log N\,(\log N + k^2))$ arithmetic complexity, and $O(k\,N \log N)$ memory footprint, where $k \ll N$ represents the numerical rank of compressed blocks. This is in contrast to $O(N^2)$ and $O(N^{1.5})$ respectively, if hierarchically low-rank matrices matrices are not used.

In this manuscript, we present ACR and its distributed-memory implementation, and we demonstrate its performance on a set of problems with various symmetry and spectral properties in three dimensions. These problems include the Poisson equation, the convection-diffusion equation, and the indefinite Helmholtz equation. We show that ACR is competitive in memory consumption and time-to-solution when compared to methods that rely on a global factorization and do not exploit the cyclic reduction structure.

1.1. Related work

Recent years have seen increasing interest in the use of hierarchical low-rank approximations to accelerate the direct solution of linear systems. In this section, we briefly describe some of this literature focusing primarily on efforts that target distributed-memory environments.

Arguably the most common approach for using hierarchical matrix representations in matrix factorizations is to use low-rank approximations to compress the dense frontal blocks that arise in the multifrontal variant of Gaussian elimination. The enabling property is that under proper ordering, many of the off-diagonal blocks of the Schur complement of discretized elliptic PDEs have an effective low-rank approximation [6] that improves the memory and arithmetic estimates of conventional multifrontal solvers [7]. Furthermore, there are efficient low-rank approximation methods to perform the necessary arithmetic operations and preserve the low-rank representation during the factorization and solution stages of the solver. Within this general approach, various methods that differ in the particular data-sparse format used and in the algorithms for the computation of low-rank approximations have been developed.

In Wang et al. [8] the authors investigate the use of the HSS format [9] to accelerate the parallel geometric multifrontal method, which results in a method known as the HSS-structured multifrontal solver (HSSMF). The general approach uses intra-node parallel HSS operations within a distributed-memory implementation of the multifrontal sparse factorization. This approach lowers the complexity of both arithmetic operations and memory consumption of the resulting HSS-structured multifrontal solver by leveraging the underlying numerically low-rank structure of the intermediate dense matrices appearing within the factorization process, driven by an optimal nested dissection ordering.

In a similar line of work, Ghysels et al. [10] also investigate a combination of the multifrontal method and the HSS-structured hierarchical format, extending the range of applicability of the solver to general non-symmetric matrices. Using the task-based parallelism paradigm, they introduce randomized sampling compression [11] and fast ULV HSS factorization [12]. Under the assumption of the existence of an underlying low-rank structure of the frontal matrices, randomized methods deliver almost linear complexity; this reduces the asymptotic complexity of the solver, which is mainly attributed to the frontal matrices near the root of the elimination tree. The effectiveness of these task-based algorithms in combination with a distributed-memory implementation of the multifrontal method is available in an early stage software release of the package STRUMPACK [10], which we will consider in the numerical experiments section of this article. The HSS format assumes a weak admissibility condition (see Section 2.1.1), which in practice requires the use of large numerical ranks even for approximations with modest relative accuracy. Consequently, this stresses the memory requirements and increases overall execution time.

The hierarchical interpolative factorization [13,14] is another method for finding low-rank approximations that has proved to be a fast solver for symmetric elliptic PDEs and integral equations. This decomposition relies on a "skeletonization" procedure to eliminate a redundant set of points from a symmetric matrix to further compress the dense fronts. The key step in skeletonization uses the interpolative decomposition of low-rank matrices to achieve a quasi-linear overall complexity in