A New Exact State Reconstruction Strategy for Conjugate Gradient Methods with Arbitrary Preconditioners

Viktoria Mayer
University of Vienna
UniVie Doctoral School Computer Science DoCS
Faculty of Computer Science
Vienna, Austria
viktoria.mayer@univie.ac.at

Wilfried N. Gansterer
University of Vienna
Faculty of Computer Science
Vienna, Austria
wilfried.gansterer@univie.ac.at

***Index Terms*—communication-hiding algorithms, pipelined conjugate gradient algorithms, extreme-scale parallel computing, algorithmic fault tolerance, exact state reconstruction**

The likelihood of unanticipated node failures in large-scale parallel computers increases with growing numbers of nodes. Furthermore, global reduction operations become major bottlenecks due to their limited parallel scalability. The Preconditioned Conjugate Gradient (PCG) method faces these challenges.

Resilience against node failures becomes increasingly important on today's large-scale parallel computers. The widely used Checkpoint-Restart strategy [1] is a generic approach that saves the entire state of the algorithm periodically. In the event of a node failure and the corresponding loss of data, the last saved state of a failed node can be retrieved and the algorithm can be continued. Alternatives to checkpoint-restart exploit algorithm-specific properties to reduce the overhead for fault tolerance and to achieve scalable resilience.

The PCG algorithm is a Krylov subspace method for solving large sparse linear systems with a symmetric and positive-definite system matrix. It has been shown how PCG can be made resilient against node failures with very small performance overhead compared to the existing non-resilient algorithm using algorithm-specific properties: Only a part of the state has to be stored redundantly and the inherent data redundancy of the matrix-vector product in each iteration can be exploited. In the case of node failures, *exact state reconstruction* (*ESR*) strategies [2], [3], [4] can be applied, i.e., the redundantly stored data are retrieved from surviving nodes, and the rest of the lost state is reconstructed exactly (except for effects of floating-point arithmetic) as it was before the node failures by exploiting algorithm-specific properties. Resilience to unanticipated node failures that does not impact the convergence of the solver using ESR strategies was so far studied only for a few PCG variants [4], [5], and only for a few very simple preconditioners.

On large-scale parallel computers, where global communication is a major bottleneck, accelerating convergence using effective preconditioners and the resulting reduction of iterations and global collectives is critical for performance. However, existing ESR strategies assume that the preconditioner matrix is explicitly available, which is not the case for several important types, e.g. Multigrid preconditioners. Even if the preconditioner matrix is explicitly available, existing ESR strategies tend to be more costly for most preconditioners as they cause extra communication cost during recovery.

A complementary approach to reducing the impact of global communication are communication-hiding and -avoiding PCG variants. *Communication-hiding methods* overlap global communication with local communication and computation [6], [7]. *Communication-avoiding s-step methods* reduce the number of global synchronizations by a factor of $\mathcal{O}(s)$ by rearranging PCG so that s iterations can be computed without communication [8], [9], [10]. When designing algorithms for large-scale parallel computers, both resilience against node failures and the reduction of communication bottlenecks must be aimed for.

We develop and discuss extensions for standard PCG as well as for communication-hiding PCG methods to make them resilient against node failures while preserving their scalability. By exploiting algorithm-specific properties of PCG, the overhead of continuously storing redundant information during the failure-free phase can be made very small. If nodes fail, efficient recovery is based on adapting an exact state reconstruction (ESR) strategy. In contrast to existing ESR strategies, our approach can be used efficiently with arbitrary preconditioners, and it requires at most a single additional preconditioner application during recovery. While we do not consider communication-avoiding s-step methods in this work, our approach is in principle applicable to various s-step methods [8], [10] as well. A detailed description of extensions for these s-step methods remains future work.

***Assumptions**:* We consider the parallel execution of the algorithms on multiple nodes with distributed memory where the runtime environment provides functionality comparable to

the Message Passing Interface (MPI) [11] for communication between nodes.

In the event of node failures, the data of the failed nodes are lost. Replacement nodes, which can be spare nodes or surviving nodes take the place of the failed nodes, recover these lost data, and then the solver can be continued.

As in [3], [4], we define the state of a solver as the (not necessarily minimal) data that define the future behaviour of the solver. We focus on recovering the lost data of the solver's state after multiple simultaneous node failures and assume that there is a runtime environment which provides fault-tolerance features, such as the MPI extension User Level Failure Mitigation (ULFM) [12], [13]. Such extensions support the detection of node failures, notification of the surviving nodes, provisioning of replacement nodes, and preventing global and local communication from being blocked indefinitely [14].

Related work: On large-scale parallel computers, different types of failures can occur. Resilience of the PCG method against soft errors, i.e. spontaneous changes of the state of the solver, caused for example by bit flips, are investigated in [15], [16], [17], [18], [19]. We focus on node failures, e.g. a node crashes or looses its connection to the network. A detailed overview of different Checkpoint-Restart strategies can be found in [1].

Algorithm-based approaches for resilience may achieve better scalability on large-scale parallel computers. Interpolation-Restart (IR) strategies for Krylov subspace methods interpolate the lost parts of the solution vector after one or several node failures. The algorithm is then restarted with this interpolation as the start vector [20], [21], [22]. An algorithm-based approach to make PCG resilient against single node failures using ESR is presented in [2] and extended to multiple simultaneous or overlapping node failures in [4]. Contrary to IR methods, where convergence can be severely impacted due to restarting the solver after node failures, ESR recovery hardly affects the convergence behaviour of the solver [3], [4].

Communication-hiding PCG algorithms such as Pipelined PCG (PPCG) [6] and the numerically stable p(l)-PCG method [7] hide global communication by overlapping it with local communication and computation. Communication-avoiding s-step methods reduce the number of global synchronizations by a factor of $\mathcal{O}(s)$, where s is a user-defined step size, e.g. [8], [9], [10]. In [5], a resilient version of PPCG [6] is presented which uses ESR ideas. To the best of our knowledge, ESR ideas have not yet been investigated for p(l)-PCG or for s-step methods.

Contributions: We focus on developing Conjugate Gradient methods suitable for large-scale parallel computers in terms of scalability and cost-efficient resilience. So far, no or at most very simple (Block) Jacobi preconditioners were considered in the context of existing ESR strategies for achieving resilience [2], [3], [4], [5]. All existing ESR strategies require the preconditioner matrix explicitly for solving small linear systems during recovery, which contain submatrices of the system matrix and of the preconditioner matrix. However, important classes of preconditioners are only available implicitly. In these cases, a solver is used to compute the preconditioned vector instead of operating with an explicit preconditioner matrix.

Several publications on (non-resilient) communication-hiding or -avoiding algorithms consider these more complex preconditioners that cannot be used with existing ESR approaches: [7] uses a SOR-preconditioned Chebyshev iteration preconditioner, [23] uses a Symmetric Gauss-Seidel preconditioner, and [24] investigates a geometric multigrid preconditioner using a preconditioned Chebyshev iteration smoother. When using these preconditioners, the parts of the state lost due to node failures cannot or not efficiently be reconstructed using existing ESR strategies as submatrices of the preconditioner cannot be formed. To the best of our knowledge, we are the first to investigate and improve the applicability of ESR with arbitrary preconditioners.

We develop a new exact state reconstruction strategy that can be used with *any* preconditioner. Moreover, unlike existing ESR strategies, our novel approach has much lower recovery overhead as it requires at most a single preconditioner application during recovery while still reconstructing the state exactly (apart from effects of floating-point arithmetic).

To avoid preconditioner applications and the solution of small systems with submatrices of the preconditioner during recovery, we introduce additional local vector operations during the failure-free phase of the solvers without increasing the communication overhead compared to existing ESR methods. The additional computational cost of our new approach is considered negligible on large-scale parallel computers as the dominant cost factors are preconditioner applications, matrix-vector products and global communication.

Since we store more vectors redundantly during the failure-free phase, recovery of the solver state using our new approach is less expensive than with existing ESR strategies. As we store all preconditioned vectors, no preconditioner applications are required during recovery except for a potential recomputation of the last preconditioner application that could not be stored redundantly before the node failures.

Recovery in existing ESR strategies requires solving small linear systems whose size depends on how many nodes failed. In contrast, our new approach does not require solving these small linear systems and is thus independent of the number of simultaneously failing nodes (except for the inexpensive retrieval of vector copies redundantly stored during the failure-free phase).

In addition to standard PCG, we consider the scalable state-of-the-art communication-hiding methods PPCG [6] and p(l)-PCG [7]. For the latter, no ESR strategy exists yet. We improve resilience of these Conjugate Gradient methods and illustrate very low resilience overhead compared to the existing non-resilient methods: for a synthetic test case clearly below 10%, for real-world test cases below 3% in the failure-free case and below 4% for three simultaneous node failures. Based on theoretical and experimental analyses, we also confirm that the scalability is preserved.

References

[1] T. Herault and Y. Robert, *Fault-Tolerance Techniques for High-Performance Computing*, 1st ed. Springer International Publishing, 2015.

[2] Z. Chen, "Algorithm-based recovery for iterative methods without checkpointing," in *Proceedings of the 20th International Symposium on High Performance Distributed Computing*, ser. HPDC '11. ACM, 2011, pp. 73–84.

[3] C. Pachajoa, M. Levonyak, and W. N. Gansterer, "Extending and evaluating fault-tolerant preconditioned conjugate gradient methods," in *2018 IEEE/ACM 8th Workshop on Fault Tolerance for HPC at eXtreme Scale (FTXS)*. IEEE, 2018, pp. 49–58.

[4] C. Pachajoa, M. Levonyak, W. N. Gansterer, and J. L. Träff, "How to make the preconditioned conjugate gradient method resilient against multiple node failures," in *Proceedings of the 48th International Conference on Parallel Processing*, ser. ICPP '19. ACM, 2019, pp. 67:1–67:10.

[5] M. Levonyak, C. Pacher, and W. N. Gansterer, "Scalable resilience against node failures for communication-hiding preconditioned conjugate gradient and conjugate residual methods," in *Proceedings of the 2020 SIAM Conference on Parallel Processing for Scientific Computing (PP)*. SIAM, 2020, pp. 81–92.

[6] P. Ghysels and W. Vanroose, "Hiding global synchronization latency in the preconditioned conjugate gradient algorithm," *Parallel Computing*, vol. 40, no. 7, pp. 224–238, 2014, 7th Workshop on Parallel Matrix Algorithms and Applications.

[7] S. Cools, J. Cornelis, and W. Vanroose, "Numerically stable recurrence relations for the communication hiding pipelined conjugate gradient method," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 11, p. 2507–2522, 2019.

[8] A. T. Chronopoulos and C. W. Gear, "On the efficient implementation of preconditioned s-step conjugate gradient methods on multiprocessors with memory hierarchy," *Parallel Computing*, vol. 11, no. 1, pp. 37–53, 1989.

[9] S. Toledo, "Quantitative performance modeling of scientific computations and creating locality in numerical algorithms," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1995.

[10] M. F. Hoemmen, "Communication-avoiding krylov subspace methods," Ph.D. dissertation, EECS Department, University of California, Berkeley, 2010.

[11] Message Passing Interface Forum, "MPI: A message-passing interface standard," https://www.mpi-forum.org/docs/, 2021.

[12] ——, "User level failure mitigation," https://fault-tolerance.org/, 2021.

[13] W. Bland, A. Bouteiller, T. Herault, G. Bosilca, and J. Dongarra, "Post-failure recovery of MPI communication capability: Design and rationale," *International Journal of High Performance Computing Applications*, vol. 27, no. 3, pp. 244–254, 2013.

[14] N. Losada, P. González, M. J. Martín, G. Bosilca, A. Bouteiller, and K. Teranishi, "Fault tolerance of MPI applications in exascale systems: The ULFM solution," *Future Generation Computer Systems*, vol. 106, no. 3, pp. 467–481, 2020.

[15] M. Shantharam, S. Srinivasmurthy, and P. Raghavan, "Fault tolerant preconditioned conjugate gradient for sparse linear system solution," in *Proceedings of the 26th ACM International Conference on Supercomputing*, ser. ICS '12. ACM, 2012, pp. 69–78.

[16] W. Zhang and H. He, "Fault tolerance for conjugate gradient solver based on FT-MPI," *Studies in Informatics and Control*, vol. 22, no. 1, pp. 51–60, 2013.

[17] A. Schöll, C. Braun, M. A. Kochte, and H.-J. Wunderlich, "Low-overhead fault-tolerance for the preconditioned conjugate gradient solver," in *2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS)*. IEEE, 2015, pp. 60–65.

[18] E. Agullo, S. Cools, E. F. Yetkin, L. Giraud, N. Schenkels, and W. Vanroose, "On soft errors in the conjugate gradient method: Sensitivity and robust numerical detection," *SIAM Journal on Scientific Computing*, vol. 42, no. 6, pp. C335–C358, 2020.

[19] G. Meurant, "Detection and correction of silent errors in the conjugate gradient algorithm," *Numerical Algorithms*, vol. 92, 2022.

[20] J. Langou, G. Bosilca, and J. Dongarra, "Recovery patterns for iterative methods in a parallel unstable environment," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 102–116, 2007.

[21] E. Agullo, L. Giraud, A. Guermouche, J. Roman, and M. Zounon, "Towards resilient parallel linear Krylov solvers: recover-restart strategies," INRIA, Research Report RR-8324, 2013.

[22] ——, "Numerical recovery strategies for parallel resilient krylov linear solvers," *Numerical Linear Algebra with Applications*, vol. 23, no. 5, pp. 888–905, 2016.

[23] E. C. Carson, T. Gergelits, and I. Yamazaki, "Mixed precision s-step lanczos and conjugate gradient algorithms," *Numerical Linear Algebra with Applications*, vol. 29, no. 3, p. e2425, 2022.

[24] Y. Idomura, T. Ina, S. Yamashita, N. Onodera, S. Yamada, and T. Imamura, "Communication avoiding multigrid preconditioned conjugate gradient method for extreme scale multiphase CFD simulations," in *2018 IEEE/ACM 9th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (scalA)*. IEEE, 2018, pp. 17–24.