LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-SR-821153

Quarter 4 Report: Report on Final Findings and Opportunities for Future Work in the Use of Mixed Precision in Iterative Solvers

E. Carson, T. Gergelits

April 6, 2021

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

B639388 Subcontract Award LLNL-Charles University: Mixed Precision Numerical Linear Algebra

Quarter 4 Report: Report on Final Findings and Opportunities for Future Work in the Use of Mixed Precision in Iterative Solvers

Erin Carson and Tomáš Gergelits

March 30, 2021

1 Summary of activities

The fourth quarter of the project was spent developing an error analysis of the s-step Lanczos and CG algorithms. Our theoretical bounds and numerical experiments show that the numerical behavior of the algorithm can be significantly improved by using extra precision in a small part of the computation related to the computation and application of the Gram matrix. We have published a technical report which includes all steps of the analysis [8]; a shortened version for journal submission is in preparation. We plan to submit this paper in the following weeks.

Activities related to this also include a collaboration with Ichitaro Yamazaki on gathering performance results for these new mixed precision s-step Krylov subspace methods using single/double precision on GPUs. Namely, we would like to obtain performance results that show that the performance overhead of using double the working precision in these select computations is minimal.

Other activities include attending biweekly xSDK meetings and presenting a pitch talk on this work to the group on February 25, 2021.

In the remainder of the document, we summarize our findings on the potential for mixed precision in classical Krylov subspace methods and s-step Krylov subspace methods, as well as key opportunities for future work.

2 Findings on mixed precision in classical CG/Lanczos

In our Q3 report, we detailed our analysis of the maximum attainable accuracy in the classical CG algorithm which uses different (or potentially the same) precisions for the SpMVs, the inner products, and the remaining vector operations. The analysis says that lower precision can be used in computing the inner products without affecting the maximum attainable accuracy, under the assumption that the method still converges; indeed, if one uses too low precision for the inner products, the algorithm may not converge at all. Our analysis does not say anything about when this will occur. Further, even if the algorithm does still converge, the convergence can in some cases be delayed, which may negate any performance benefit of the lower precision inner products. However, for other certain linear systems, the convergence rate is not affected.

One could also combine this with the work on inexact Krylov subspace methods [19, 23], which says that the SpMV precision can be decreased at a rate inversely proportional to the updated residual norm without affecting accuracy. Again, the piece of the puzzle missing is a rigorous theoretical analysis which says how this will affect the convergence rate of CG.

We thus think that there is little potential for low/mixed precision computation in (unpreconditioned) classical CG in general, although this may make sense in certain linear systems. The key things that must be determined are 1) the particular matrix structure, size, and machine parameters such that the use of low precision has significant performance benefit, and 2) the properties of the problem for which convergence rate is not significantly affected by the use of low precision, and exactly how low this precision can be. The latter is a difficult mathematical problem.

One thing that has not yet been investigated, but which may prove to be fruitful, is a mixed precision version of preconditioned CG in which the preconditioner may be computed and applied in lower precision. We believe that this case could be theoretically investigated using the theory of Greenbaum [14].

Yet another area that could be investigated is, instead of using lower precision, to use higher than the working precision in select computations. This could aid in maintaining closer orthogonality of the Krylov basis vectors, which would in turn improve the convergence rate. If the convergence rate is improved, this may enable faster time-to-solution despite any overhead of using extended precision. This tradeoff should be investigated. This similar to the approach used in s-step Krylov subspace algorithms, detailed below, although in the s-step case we expect the potential benefits to be relatively higher than for classical CG/Lanczos.

3 Findings on mixed precision in s-step CG/Lanczos

In the last quarter, we proved that if, in s-step Lanczos, the Gram matrix is computed and applied in double the working precision, the numerical behavior, namely, orthogonality among Lanczos basis vectors, is improved by a factor related to the condition numbers of the s-step bases. For reference, we show the resulting mixed precision s-step Lanczos algorithm in Algorithm 1.

Algorithm 1 Mixed precision s-step Lanczos

Input: n-by-n real symmetric matrix A and length-n starting vector v_1 such that $\|v_1\|_2 = 1$, stored in precision ε

$u_1 = Av_1$ (precision ε)
for $k = 0, 1, \ldots$ until convergence **do**
  Compute $\mathcal{Y}_k$ with change of basis matrix $\mathcal{B}_k$ (precision ε).
  Compute and store $G_k = \mathcal{Y}_k^T \mathcal{Y}_k$ in precision ε^2.
  $v'_{k,1} = e_1$
  if $k = 0$ **then**
    $u'_{0,1} = \mathcal{B}_0 e_1$
  else
    $u'_{k,1} = e_{s+2}$
  end if
  for $j = 1, 2, \ldots, s$ **do**
    Compute $g = G_k u'_{k,j}$ in precision ε^2, store in precision ε.
    $\alpha_{sk+j} = v'^T_{k,j} g$ (precision ε)
    $w'_{k,j} = u'_{k,j} - \alpha_{sk+j} v'_{k,j}$ (precision ε)
    Compute $c = G_k w'_{k,j}$ in precision ε^2, store in precision ε.
    $\beta_{sk+j+1} = (w'^T_{k,j} c)^{1/2}$ (precision ε)
    $v'_{k,j+1} = w'_{k,j} / \beta_{sk+j+1}$ (precision ε)
    $v_{sk+j+1} = \mathcal{Y}_k v'_{k,j+1}$ (precision ε)
    $u'_{k,j+1} = \mathcal{B}_k v'_{k,j+1} - \beta_{sk+j+1} v'_{k,j}$ (precision ε)
    $u_{sk+j+1} = \mathcal{Y}_k u'_{k,j+1}$ (precision ε)
  end for
end for

To summarize the analysis from Q3, the resulting bound on the loss of orthogonality has the same structure as the uniform precision s-step Lanczos appearing in [7], but with the notable exception that the bound now contains only a factor of $\bar{\Gamma}_k$ rather than $\bar{\Gamma}_k^2$, where $\bar{\Gamma}_k = \max_{i \in \{0,\ldots,k\}} \|\hat{\mathcal{Y}}_i^+\|_2 \| \, |\hat{\mathcal{Y}}_i| \, \|_2$. As Γ_k can potentially grow very quickly with s, this is a significant improvement, and indicates that, among other things, the Lanczos basis vectors will maintain significantly better orthogonality and normality due to the selective use of higher precision.

3.1 New results: Extension of bounds on eigenvalue accuracy and convergence

In Q4, we have expanded upon this analysis, and worked on extending the subsequent results of Paige [17] to this case, which allows for results on the accuracy and convergence of eigenvalues. In the uniform precision s-step Lanczos, these results are applicable as long as

$$\bar{\Gamma}_k < \big(24\varepsilon(n + 11s + 15)\big)^{-1/2} = O(1/\sqrt{n\varepsilon}).$$

Due to the use of extended precision in the mixed precision case, the constraints are now relaxed, requiring that

$$\bar{\Gamma}_k < \big(2\varepsilon(6s + 11)\big)^{-1} = O(1/\varepsilon),$$

under the assumption that $n\varepsilon\Gamma_k < 1$ for all k. This is significant improvement. For example, if the working precision is double, then in the uniform precision case, we can only expect predictable behavior as long as the s-step bases have condition number bounded by $\approx 10^8$, whereas this becomes 10^{16} in the mixed precision case.

We do not reproduce all theorems here (they will appear in the submitted journal version of the paper), but merely summarize the main points. The analyses rely on the definition of a quantity ε_2, which is $O(\varepsilon n)$ in the classical Lanczos case, $O(\varepsilon n \bar{\Gamma}_k^2)$ in the uniform precision s-step Lanczos case, and $O(\varepsilon \bar{\Gamma}_k^2)$ in the mixed precision s-step Lanczos case.

The main result is that, assuming no breakdown occurs and the size of $\bar{\Gamma}_k$ satisfies the respective constraints on $\bar{\Gamma}_k$, these results say the same thing for the mixed precision s-step Lanczos case as in the uniform

precision s-step Lanczos and classical Lanczos cases: *until an eigenvalue has stabilized, the mixed precision s-step Lanczos algorithm behaves very much like the error-free Lanczos process, or the Lanczos algorithm with reorthogonalization.*

The primary difference among these three Lanczos variants is how tight the constraints are by which we consider an eigenvalue to be "stabilized". The larger the value of ε_2, the looser the constraint on stabilization becomes, and thus the sooner an eigenvalue is considered to be "stabilized". Thus, somewhat counterintuitively, for the uniform precision s-step Lanczos process where ε_2 is expected to be largest (as it contains a factor $\bar{\Gamma}_k^2$), we expect "stabilization" to happer sooner than in the other methods (but again, to within a larger interval around the true eigenvalues of A), *and thus we expect faster deviation from the exact Lanczos process.* In the classical Lanczos method, the smaller value of ε_2 means that we are more discriminating in what we consider to be a stabilized eigenvalue, and thus stabilization will take longer, which means we follow the exact Lanczos process for more iterations. For the mixed precision s-step Lanczos case, we expect the value of ε_2 to fall somewhere in the middle of the other two variants.

In the classical Lanczos case, the results in [17] say that we have at least one eigenvalue of A with high accuracy by iteration $m = n$. In both uniform and mixed precision s-step Lanczos algorithms, it is still true that we will find at least one eigenvalue with some degree of accuracy by iteration $m = n$ as long as the respective constraints on $\bar{\Gamma}_k$ hold, but here the limit on accuracy is determined by the size of $\bar{\Gamma}^2_{\lceil n/s \rceil}$ in the uniform precision case and $\bar{\Gamma}_{\lceil n/s \rceil}$ in the mixed precision case. Thus we can expect in general, eigenvalue estimates will be about a factor $\bar{\Gamma}_{\lceil n/s \rceil}$ more accurate in the mixed precision case versus the uniform precision case.

3.2 New results: A mixed precision s-step CG

The CG method for solving linear systems is based on an underlying Lanczos process. Similarly, the s-step CG algorithm is based on an underlying s-step Lanczos algorithm. We therefore expect that the improved Ritz value accuracy obtained by the use of the mixed precision approach in the s-step Lanczos algorithm will lead to improvements in convergence behavior in a corresponding mixed precision s-step CG algorithm. We note that we do not expect that the use of extended precision in the Gram matrix computation will improve the maximum attainable accuracy in s-step CG, as this is primarily dependent on the precision used for SpMVs; see, e.g., [6] for bounds on the maximum attainable accuracy for s-step CG.

We present a few numerical experiments in MATLAB (R2020a) that support this conjecture. We compare classical CG (the 2-term recurrence variant) in double precision to s-step CG in double precision and to mixed precision s-step CG in double/quad. The mixed precision variant follows the same principle as the mixed precision s-step Lanczos variant. Namely, that the Gram matrix is computed and applied in double the working precision, and everything else is done in the working precision. For double precision, we use the built-in MATLAB datatype and for quadruple precision we use the Advanpix MATLAB Toolbox with 34 decimal digits. We use right-hand sides generated to have equal components in the eigenbasis of A and unit 2-norm, which represents a difficult case for CG (all components must be found). In each experiment, we measure the relative error measured in the A-norm, where the true solution is computed using MATLAB backslash in quadruple precision via Advanpix.

We test 3 small symmetric positive definite matrices from the SuiteSparse collection [13], `lund_b`, `bcsstk02`, and `nos4`. For each matrix, we run the test using two different s values, $s = \{6, 10\}$, and two different polynomial bases, monomial and Chebyshev. No preconditioning is used in these experiments.

There are a few interesting things to observe. First, in all cases, the mixed precision s-step CG variant exhibits convergence behavior much closer to that of the classical method than the uniform precision s-step variant. This is true regardless of the s value used, and regardless of the polynomial basis used. Another thing to note is that, as expected, the attainable accuracy of the uniform and mixed precision s-step variants is about the same. This is largely due to errors in matrix-vector products, and thus the extra precision does not improve this. Investigating whether using extra precision here is beneficial from a theoretical and performance standpoint remains future work, although we note that heuristic techniques such as residual replacement may help in some practical cases [6].

Another interesting point is that the Chebyshev basis can decrease the attainable accuracy versus the monomial basis, as is seen, for example, in Figure 1 for $s = 6$. In Figure 2, for $s = 10$ and Chebyshev basis, the uniform precision variant does not converge at all, whereas it does when a monomial basis is used. This behavior needs further investigation; it could be the case that a Newton basis would better capture spectral properties and would result in better behavior.

A final point is that it would appear that in some cases, such as in Figure 3, use of the Chebyshev basis does a good enough job of reducing the s-step basis condition numbers, and there is likely not a need for the use of extra precision in this case. We argue that because the very selective use of extra precision is likely not a huge performance overhead, these two techniques should be considered orthogonal, and should be used in combination for the best behavior in practice as they improve numerical behavior through different means; using a more well-conditioned polynomial basis constructed using Chebyshev or Newton polynomials will reduce

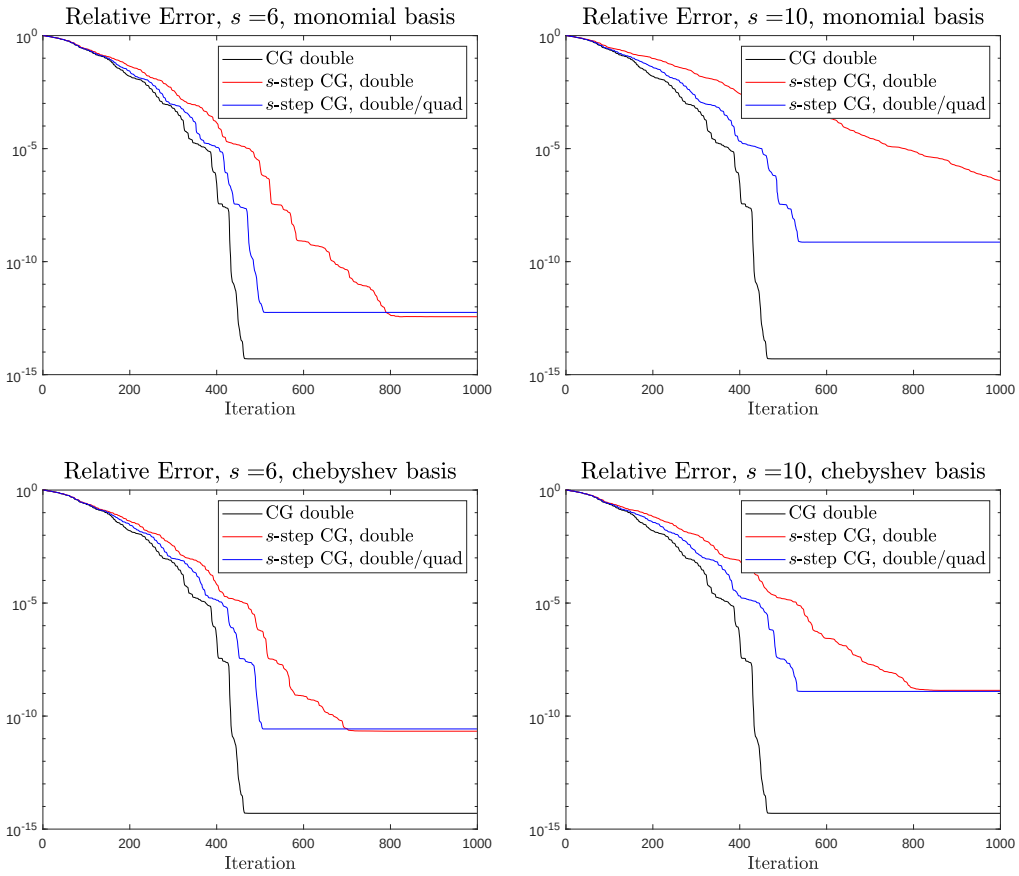

Figure 1: `lund_b` from SuiteSparse

the value of $\bar{\Gamma}_k$, and the use of mixed precision will reduce the loss of orthogonality bound dependence from $\bar{\Gamma}_k^2$ to $\bar{\Gamma}_k$.

3.3 Ongoing work: Performance implications

It is clear from analysis and experimental results that the convergence behavior of short-term recurrence s-step Krylov subspace methods can be improved by the selective use of extended precision as described. It remains to justify that the performance overhead of the use of extra precision does not negate any potential advantage of the s-step approach.

We can reason about the expected performance in the parallel setting. For the computation of the Gram matrix in double the working precision, the MPI Allreduce that happens every s iterations is doubled in size (we send twice as much data), but depending on the architecture and the value of s used, this likely still fits in one message (it is still of size $O(s^2)$ words, and s is expected to be very small, around 10). So as long as the computation is not bandwidth-bound, we expect that this will not have significant performance impact. The Gram matrix itself is small ($O(s^2)$) and thus fits locally in cache of each processor, and so application of the Gram matrix to length-$O(s)$ vectors in double the working precision is expected to have negligible overhead. We are currently collaborating with Ichitaro Yamazaki to confirm this experimentally using single/double precision on GPUs.

4 Potentially opportunities for future exploration

We have identified a number of interesting questions that can be addressed in the future, which we itemize below. We do not give details here, but are happy to elaborate on ideas these further upon request. We will decide on one or more of these ideas for potential exploration in the following year of the project.

- Extension of Greenbaum's theory [14] to mixed precision CG, to mixed precision preconditioned CG, and to mixed precision s-step CG (relevant works include, e.g., [14, 7, 8])

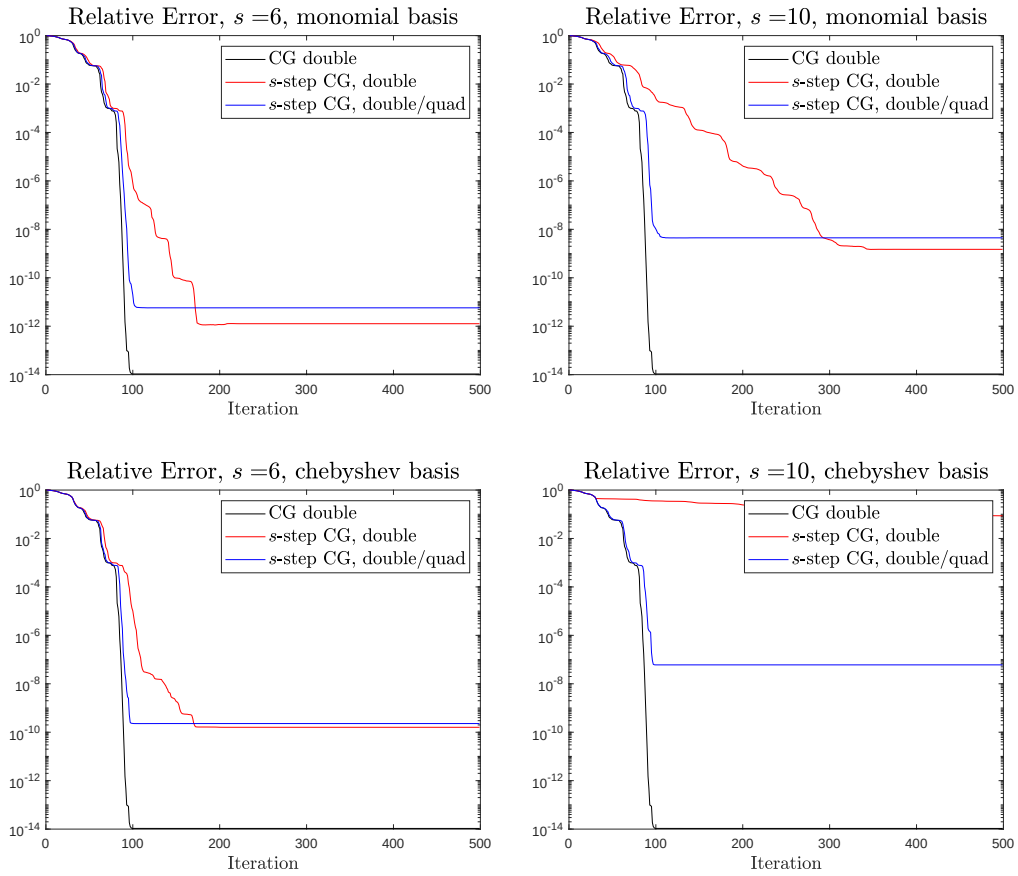

Figure 2: `bcsstk02` from SuiteSparse

- Investigation of mixed precision randomized preconditioners for least squares problems (relevant works include, e.g., [1, 15, 16, 11])
- The use of incomplete LU factorizations within mixed precision GMRES-based iterative refinement (relevant works include, e.g., [9, 10])
- Mixed precision hierarchical matrices and their use as preconditioners (relevant works include, e.g., [5, 2, 4, 3, 24])
- Mixed precision low-sync Gram-Schmidt algorithms and the backward stability of resulting GMRES variants (relevant works include, e.g., [22, 21, 25, 12])
- Three-precision iterative refinement with recycling (relevant works include, e.g., [9, 10, 18, 20])

References

[1] Haim Avron, Petar Maymounkov, and Sivan Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing*, 32(3):1217–1236, 2010.

[2] Mario Bebendorf. Efficient inversion of the Galerkin matrix of general second-order elliptic operators with nonsmooth coefficients. *Mathematics of Computation*, 74(251):1179–1199, 2005.

[3] Mario Bebendorf and Wolfgang Hackbusch. Stabilized rounded addition of hierarchical matrices. *Numerical Linear Algebra with Applications*, 14(5):407–423, 2007.

[4] Liefeng Bo, Kevin Lai, Xiaofeng Ren, and Dieter Fox. Object recognition with hierarchical kernel descriptors. In *CVPR 2011*, pages 1729–1736. IEEE, 2011.

[5] Steffen Börm, Lars Grasedyck, and Wolfgang Hackbusch. Introduction to hierarchical matrices with applications. *Engineering analysis with boundary elements*, 27(5):405–422, 2003.

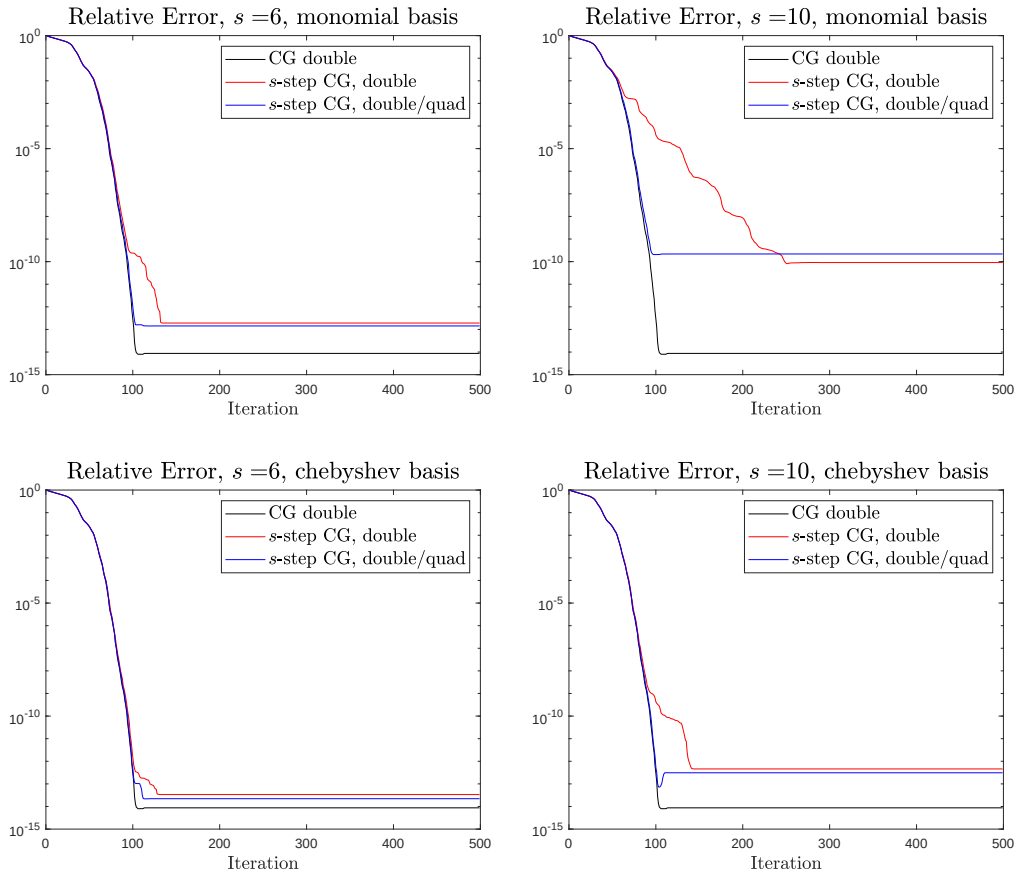

Figure 3: `nos4` from SuiteSparse

[6] E. Carson and J. Demmel. A residual replacement strategy for improving the maximum attainable accuracy of s-step Krylov subspace methods. *SIAM J. Matrix Anal. Appl.*, 35(1):22–43, 2014.

[7] E. Carson and J. W. Demmel. Accuracy of the s-step Lanczos method for the symmetric eigenproblem in finite precision. *SIAM J. Matrix Anal. Appl.*, 36(2):793–819, 2015.

[8] Erin Carson and Tomáš Gergelits. Mixed precision s-step Lanczos and conjugate gradient algorithms. *arXiv preprint arXiv:2103.09210*, 2021.

[9] Erin Carson and Nicholas J Higham. A new analysis of iterative refinement and its application to accurate solution of ill-conditioned sparse linear systems. *SIAM Journal on Scientific Computing*, 39(6):A2834–A2856, 2017.

[10] Erin Carson and Nicholas J Higham. Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM Journal on Scientific Computing*, 40(2):A817–A847, 2018.

[11] Erin Carson, Nicholas J Higham, and Srikara Pranesh. Three-precision GMRES-based iterative refinement for least squares problems. *SIAM Journal on Scientific Computing*, 42(6):A4063–A4083, 2020.

[12] Erin Carson, Kathryn Lund, Miroslav Rozložník, and Stephen Thomas. An overview of block Gram–Schmidt methods and their stability properties. *arXiv preprint arXiv:2010.12058*, 2020.

[13] Timothy A Davis and Yifan Hu. The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1–25, 2011.

[14] A. Greenbaum. Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences. *Lin. Alg. Appl.*, 113:7–63, 1989.

[15] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[16] Per-Gunnar Martinsson and Joel Tropp. Randomized numerical linear algebra: Foundations & algorithms. *arXiv preprint arXiv:2002.01387*, 2020.

[17] C. C. Paige. Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. *Lin. Alg. Appl.*, 34:235–258, 1980.

[18] Michael L Parks, Eric De Sturler, Greg Mackey, Duane D Johnson, and Spandan Maiti. Recycling Krylov subspaces for sequences of linear systems. *SIAM Journal on Scientific Computing*, 28(5):1651–1674, 2006.

[19] Valeria Simoncini and Daniel B Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM Journal on Scientific Computing*, 25(2):454–477, 2003.

[20] Kirk M Soodhalter, Eric de Sturler, and Misha E Kilmer. A survey of subspace recycling iterative methods. *GAMM-Mitteilungen*, 43(4):e202000016, 2020.

[21] Kasia Swirydowicz, Julien Langou, Shreyas Ananthan, Ulrike Yang, and Stephen Thomas. Low synchronization GMRES algorithms. *arXiv preprint arXiv:1809.05805*, 2018.

[22] Katarzyna Świrydowicz, Julien Langou, Shreyas Ananthan, Ulrike Yang, and Stephen Thomas. Low synchronization Gram–Schmidt and generalized minimal residual algorithms. *Numerical Linear Algebra with Applications*, 28(2):e2343, 2021.

[23] Jasper Van Den Eshof and Gerard LG Sleijpen. Inexact Krylov subspace methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):125–153, 2004.

[24] Jianlin Xia, Shivkumar Chandrasekaran, Ming Gu, and Xiaoye S Li. Fast algorithms for hierarchically semiseparable matrices. *Numerical Linear Algebra with Applications*, 17(6):953–976, 2010.

[25] Ichitaro Yamazaki, Stephen Thomas, Mark Hoemmen, Erik G Boman, Katarzyna Świrydowicz, and James J Elliott. Low-synchronization orthogonalization schemes for s-step and pipelined Krylov solvers in trilinos. In *Proceedings of the 2020 SIAM Conference on Parallel Processing for Scientific Computing*, pages 118–128. SIAM, 2020.