Comments to Reviewers

L. Minah Yang, Alyson Fox, Geoffrey Sanders

June 1, 2020

We would like to thank both reviewers for your time and effort. First, we would like to make a few comments on the major aspects of the revisions:

- **Column partitioned Householder QR factorization algorithm (HQR) has been added to reflect on mixed precision settings of GPU tensor cores units.**
  We added the analysis of the level-3 BLAS variant of HQR since it is the standard HQR implementation in many libraries and can be effortlessly adapted to utilize the block Fused Multiply-Add operations (bFMAs) of NVIDIA TensorCore units. The WY representation of [1] is used instead of the compact, storage-efficient version of [8] since the former is discussed in both [3, 4], which we often refer to in the text.

- **We have added the mixed precision setting of NVIDIA's TensorCore bFMAs.**
  This setting is more relevant and practical since these hardware units are already in use, and result in fewer low precision errors than the inner product mixed precision setting we had introduced. While we do not discuss the speed-up advantages in depth, we do refer to the speed benchmarks for these operations that already exist and may be of interest to the readers.

- **Section 5 (Applications) has been removed from this manuscript.**
  Given the length of the first submission and the additional materials introduced as explained above, we have removed the applications section from this text. However, we plan on presenting our work on using mixed precision arithmetic in graph clustering in a future work, where we will include other graph problems.

Next, we would like to address concerns voiced by both referees:

- **Missing references to relevant and prior works.**
  Referee #1: The paper also suffers a complete lack of acknowledgment of prior work in this area and of the current state-of-the-art.
  - **lines 157-159: citation missing at the end of this sentence.**
    Citations for probabilistic rounding error analyses have been added ([5, 7]) to line A.
  - **lines 163-165: Be more specific about what rounding error analysis framework was established in the textbook [13]. As far as I know, the textbook does not establish any new error analysis framework, nor is it limited to analyses using a single precision.**
    We did not intend to imply that [4] establishes new error analysis framework. In section 3, we have included the standard rounding error analyses for HQR and its level-3 BLAS variant and have appropriately attributed and redirected readers to a main source, [4]. The textbook ([4]) does refer to mixed precision iterative refinement, but the rounding error analysis for HQR(Section 19.3) and aggregated Householder transformations (Section 19.5) both assume a uniform precision setting.

– **Section 2.2: Prior work should be discussed and cited here.**
In section 2.2, we define MP Setting 2.3 (line C), which is the mixed precision inner product that mimics TensorCore bFMAs but in a level-2 BLAS operation. Later in section 4.1 (line D), we discuss the specifics of NVIDIA TensorCore bFMAs, and reference work from [2].

In general, we tried to be clear about what is existing, standard error analysis and what are our new error analyses for mixed precision algorithms. This was mainly done by separating the two into section 3 (standard) and section 4 (mixed precision). The idea behind mixed precision analysis is not novel, and our contributions are in applying them into mixed precision QR factorization algorithms that we developed. Furthermore, we still present many details considered to be standard in order to allow readers from varying mathematical backgrounds.
Referee #2: We have added the suggested references for the following topics.

– We have referenced [2] for their work on mixed precision matrix products and LU decomposition in section 4.1 (line D).
– References to ([5, 7]) on probabilistic rounding error analyses have been added to to line A.
– Work on proving that the faithfulness of Algorithm 1 on simulating half precision arithmetic have been referenced ([6]) in line E.

- **Inconsistencies in notation and adhering to standard notation:**
  – **Referee #1:** Hyphens have been removed from "low-", "mixed-", and "high-precision".
  – **Referee #2:** We changed the notation for γ for k accumulated rounding errors in precision type q from $\gamma_q^{(k)}$ to $\gamma_k^{(q)}$ to match the standard notation.
- **P2L44-45: "QR factorization is known to provide a backward stable solution to the linear least squares problem ...'**
  We meant to motivate the need for mixed precision QR factorization algorithms since mixed precision is an active area of research. However, we have removed this sentence as it was confusing and did not serve the purpose of the paper.

We now address unique concerns from each referee separately.

Requests from Referee #1

1. **The primary problem that I see is that the authors have not convinced me that there is any novelty in what they call a "new framework" for doing mixed-precision floating point error analysis.**
   We have reestablished our contributions in this work to be developing and analyzing mixed precision Householder QR factorization algorithms. This is reflected on the change in the title, as well as in line R of the introduction, and in the abstract. We also dropped the notation of using $\gamma_{d:=\lceil mu^{(h)}/u^{(l)} \rceil}$ (which we had considered a part of the new framework in the first submission) in favor of keeping two precisions separate.

2. **line 39: What is meant by "exact products"?**
   The documentation for TensorCore bFMAs says "full precision products", which in our context is equivalent to a product in exact arithmetic. We have added a full explanation to clarify this in section 2 (line F).

3. **Pages 2-6 contain standard introductory textbook material and can be significantly shortened.**
   While this material is standard, intermediate results from these analyses are necessary for the rounding error analyses for the mixed precision variant of the algorithms in section 4. We are aware that this adds significant length to this paper and have tried to shorten it while keeping it accessible for readers who may not be familiar with rounding error analyses. Modifying the existing analyses for various mixed precision settings is precisely our major contribution, which is presented in Section 4.

Requests from Referee #2

1. **Choice of mixed precision assumptions**

   - **Relation with GPU tensor core units**
     We have added a mixed precision setting that addresses this specific hardware. This is first mentioned in line G of section 2.2, and further explored in section 4.2 (line D).
   - **Assumptions on storage precision types**
     In line H of section 4.2, we explain that intermediate matrix products should be stored in the higher precision and the low precision output is used only for the final result of the block matrix product in order to gain the highest accuracy. In the analysis for a mixed precision variant of the level-3 HQR (BQR) in section 4.2, this translates to introducing only $\mathcal{O}(n/r)$ low precision rounding errors for forming the QR factorization for an m-by-n sized matrix whose columns are partitioned in groups of r. Also, we reference the readers to [2] for a full analysis of matrix products.
   - **Distinction between Lemma 2.4 and Corollary 2.5**
     We only define the exact product variant of Corollary 2.5 in MP Setting 2.3. With adding the TensorCore bFMAs, we felt that we didn't need two different types of mixed precision inner products.

2. **Rounding error analysis framework**

   - **Add interpretation after every major result.**
     As suggested, we moved the material from Appendix A into the main body. Since we have added the bFMAs as a separate mixed precision setting, we first present the standard error analysis in section 3, present mixed precision variants of the 3 main algorithms (HQR, BQR, TSQR) and their rounding error analysis in section 4. The error analysis in section 4 heavily relies on intermediate results from section 3, and references specific equations and lemmas established in section 3.
   - **Keep different precisions in the bound.**
     In the first submission, we had converted bounds in the form of $\gamma_{k_1}^{(l)} + \gamma_{k_2}^{(h)}$ to $\gamma_{\lceil k_1 + k_2 \frac{u^h}{u^l} \rceil}^{(l)}$ by defining some constant $d \approx \lceil k_1 + k_2 \frac{u^h}{u^l} \rceil$. Since it was difficult to determine how to

relate d back to the original problem size, we have eliminated this conversion and kept track of the errors in the low and high precision separately, as suggested.

- **Use the $\tilde{\gamma}$ notation to avoid keeping track of all constants.**
  We changed to using the $\tilde{\gamma}$ notation whenever it was possible to do so, and specifically mentioned whenever we applied the assumptions under this notation to get rid of a non-leading order terms (e.g. lines I's).

3. **Conclusions from the mixed precision HQR analysis (section 3). The main results need to be highlighted, discussed in more depth**

   - **How do we feel about the low precision error term still having a dependence on n? Is there a way to drop this dependence?**
   - **Do the numerical experiments give any insight to the questions posed above?**

   The above two questions in combination with concerns about the mixed precision setting prompted us to look into the column-partitioned HQR variants. The low precision error dependence on n when using MP Setting 2.3 is an improvement than mn, but still is quite limiting. The algorithm discussed in Section 4.2. drops this dependence from n to $N := n/r$, where $1 - 1/N = 1 - r/n$ should be correlated to the speed-ups possible from TensorCores. We discuss this briefly in line Q of section 5.

4. **Conclusion from HQR vs TSQR comparison (section 4).**

   - **It may be beneficial to summarize the error bounds for HQR, TSQR, and their mixed precision variants in one place.** Now that we have added another mixed precision setting and another QR factorization algorithm, we decided to leave the discussions for each mixed precision setting separate (i.e. section 4.1 and 4.2.)
   - **Uniform precision comparison of HQR and TSQR needs to be simplified and clarified, P18L531: The $(L+1)/2^L$ factor is reversed**
     We have shorted this discussion and reversed the fraction in line O of section 3.3.3.
   - **Mixed precision comparison of HQR and TSQR needs to be emphasized, and not overshadowed by Figure 3.** This has been moved to section 4.2.3. Also, we reworded the statement that comments on the empirical results vs the error bounds in line P of the conclusion.

5. **Mislabel of forward and backward errors.**
   We fixed these inaccuracies in the various locations noted by the referee.

6. **P1L15: "standard algorithms may no longer be numerically stable when using half precision".**
   We reworded this sentence to "standard algorithms yield insufficient accuracy when using half precision" as suggested since the definition of numerical stability is relative with respect to machine precision (line K, section 1).

7. **P2L49: fp16 should be removed, bfloat should be bfloat16.**
   We made this change in line J of section 1.

8. **P5L125: Rewording is needed to clarify "k represents the number of FLOPs"**
   A rewording of this sentence eliminated that phrase in line L of section 2.1.

9. **Title suggestion:**
   We altered the title to: Rounding Error Analysis of Mixed Precision Block Householder QR Algorithms.

10. **P1L20: what does "weight" refer to in this context?**
    Weight refers to the physical weight of the hardware, which is a relevant feature in sensor formation literature (line M of section 1). We did not add an explanation in the text as it is just an example, and does not play a crucial role in the main contributions of this paper.

11. **P12L329: the middle term should be $(1+\delta_w)(x_1-\sigma-\Delta\sigma)$, rather than $(1+\delta_w)(\sigma+\Delta\sigma)$. Moreover, the last equality is only true because no cancellation can happen, since x_1 and σ have the same sign: this should be commented on.** The changes were made and the last equality being contingent on the special case that x_1 and σ have the same sign was mentioned in line N of section 3.1.2.

12. **Equations (4.6) and (4.7): isn't the $\sqrt{n}$ factor on the wrong equation?**
    The $\sqrt{n}$ factor should be on the bounds for $\|\Delta\mathbf{Q}\|_F$ and $\|\hat{\mathbf{Q}}\hat{\mathbf{R}} - \mathbf{A}\|_F$. We checked for and fixed this inaccuracy in various locations throughout sections 3 and 4.

13. **P20L599: I find it very strange that the backward error depends on the condition number of the matrix! Is it rather the forward error that is being plotted?**
    The error plotted here is $\|\hat{\mathbf{Q}}_{mpHQR2}\hat{\mathbf{R}}_{mpHQR2} - A\|F$, and is now a part of Figure 3.

14. **Section 5: given the relatively theoretical nature of this article, section 5 felt slightly out of place to me. Given that the article is quite long, perhaps the authors could consider including section 5 in another piece of work?**
    We plan to include this section in another piece of work.

References

[1] C. Bischof and C. Van Loan, *The WY Representation for Products of Householder Matrices*, SIAM Journal on Scientific and Statistical Computing, 8 (1987), pp. s2–s13, https://doi.org/10.1137/0908009.

[2] P. Blanchard, N. J. Higham, F. Lopez, T. Mary, and S. Pranesh, *Mixed Precision Block Fused Multiply-Add : Error Analysis and Application to GPU Tensor Cores*, (2019).

[3] G. H. Golub and C. F. Van Loan, *Matrix computations*, JHU press, 4 ed., 2013.

[4] N. J. Higham, *Accuracy and Stability of Numerical Methods*, 2002, https://doi.org/10.2307/2669725.

[5] N. J. Higham and T. Mary, *A New Approach to Probabilistic Rounding Error Analysis*, SIAM Journal on Scientific Computing, 41 (2019), pp. A2815–A2835, https://doi.org/10.1137/18M1226312, https://epubs.siam.org/doi/10.1137/18M1226312.

[6] N. J. Higham and S. Pranesh, *Simulating Low Precision Floating-Point Arithmetic*, SIAM Journal on Scientific Computing, 41 (2019), pp. C585–C602, https://doi.org/10.1137/19M1251308, https://epubs.siam.org/doi/10.1137/19M1251308.

[7] I. C. F. Ipsen and H. Zhou, *Probabilistic Error Analysis for Inner Products*, (2019), http://arxiv.org/abs/1906.10465, https://arxiv.org/abs/1906.10465.

[8] R. Schreiber and C. Van Loan, *A Storage-Efficient WY Representation for Products of Householder Transformations*, SIAM Journal on Scientific and Statistical Computing, 10 (1989), pp. 53–57, https://doi.org/10.1137/0910005.