Review on the SISC manuscript M129636R "Mixed-Precision Analysis of Householder QR Algorithms" by L. Yang, A. Fox, and G. Sanders

July 31, 2020

I would like to start by thanking the authors for clearly having taken the review process very seriously. I have found the revised article to be much stronger and clearer. The authors have taken into account all the major points of my first review, and in some cases have completely exceeded my expectations. I'm notably referring to the fact that the revised article now analyzes both BLAS-2 and BLAS-3 (blocked) QR algorithms, thereby covering the use of block FMA units such as GPU tensor cores. This is a valuable addition that makes the paper very useful for practitioners interested in exploiting this type of hardware to accelerate QR factorization.

After this second round of review, I have several other new comments, but I consider them to be minor and hopefully easy to address. I have no doubt that this paper will go on to be a reference on the error analysis of mixed precision QR.

Main comments

- In Lemma 2.2 and in various places in the analyses, $(1+\theta_k)/(1+\theta_j) = 1+\theta_{k+2j}$ is used when $j > k$. As noted in [2, bottom of p. 67], θ_{k+2j} can be improved to θ_{k+j} when the expressions of θ_k and θ_j are known and given by products of $(1+\delta_i)^{\pm 1}$ terms. I have not checked all the details carefully but I believe in most if not all places in this article, the improved expression could be used? Could the authors please confirm whether this is correct? This may especially be important in the analysis of section 4.2.1 where a constant of 25 could be avoided.

- In my first review I suggested using the $\widetilde{\gamma}$ notation to hide small constants and simplify the analysis. I wonder whether it would however be useful/interesting to keep track of the constant in front of m, that is, to replace $\gamma_{c_1 m+c_2}$ by $\widetilde{\gamma}_{c_1 m}$ instead of $\widetilde{\gamma}_m$. I wonder about this because the analysis can apparently sometimes lead to quite large constants, although perhaps my first comment will make this issue go away.

- Pages 6 and 7, there is a discussion of the speed benefits associated with MP Setting 2.3. I have found this discussion to be slightly confusing/questionable and not really useful, given this article focuses on error analysis and does not provide any performance experiments, as indicated on L212. I suggest removing any detailed discussion of the speed benefits of MP Setting 2.3 vs the TensorCore setting. In particular, I feel that the distinction of the 3 cases is not really meaningful and could be replaced by a simpler discussion comparing the mixed-precision bound with the uniform precisions bounds $mu^{(l)}$ and $mu^{(h)}$.

- Figure 1 is hard to decipher, mainly due to the use of a color map for the error which can vary by several orders of magnitude. Can the authors use a color map using a log-scale? Or perhaps the figure can be replaced by a simpler, less rich figure with fixed n and varying m on the x-axis, error on the y-axis, and different plots for each variant.

- Section 4: I think the paper would read better by interchanging sections 4.1 and 4.2. The bounds of section 4.2 are weaker, and adding them after section 4.1 therefore seems artificial and unnecessary. Instead, giving first the bounds for the more general MP setting 2.3 would allow to present the bFMA/TensorCore setting as a further improvement, which would neatly emphasize the accuracy benefits brought by this type of hardware.

- Equation (4.5): $\gamma_{p/4}^{(fp32)}$ should be replaced by $\gamma_p^{(fp32)}$. In an early preprint version of [1], the former was indeed given under the assumption that NVIDIA tensor cores implement a "true FMA", that is, that in (4.4) the additions in the 4×4 product AB are done in full precision. However this is not the case, the additions are performed in fp32, and so there are still p rounding errors that accumulate in a general product of inner dimension p, as indicated in the published version of [1] (cf. [1, Thm 3.1]). Fortunately I do not think this impacts the analysis of this paper beyond equation (4.5).

- The discussion on L684–691 is overall OK, although I feel that there are two points that could be better emphasized.

  - First, the sentence "However, as r grows large..." makes it seems as if r should be taken as small as possible for performance, but of course this is not the case since the factorization then also reduces to BLAS-2 operations. In practice, realistic values of r probably range between $O(10)$ and $O(100)$, depending on hardware, and not discussing double partitioning strategies. So it is entirely possible that the optimal value of r for performance is already quite large enough so that the accuracy is satisfying.
  - Second, for a fixed r, (4.11) shows that the loss of accuracy becomes less and less significand as m increases and so asymptotically, regardless of the

practical value of r, the accuracy eventually will be satisfying. I feel this should be mentioned.

- I suggest adding a Table at the end of section 4 or beginning of section 5 summarizing the bounds obtained for each of the 9 algorithms: {H,B,TS}QR and mp{H,B,TS}QR{2,3}. It would really be useful to have such a Table, for other articles to refer to, and as reference when comparing with the numerical experiments of section 5.

- Figure 2, left plot: I suggest fixing r to a constant rather than $n/4$, which I think is more realistic. Also, have the authors checked what the plots look like for fixed n rather than $n = m/4$? The bounds suggest n should not play a significant role in the relative accuracy of the MP block algorithms vs. the uniform precision ones, but it might be worth checking.

- Figure 2, right plot: the chosen m is a bit small, would it be worth giving the errors for a larger m (which would probably lead to the MP and uniform algorithms to be closer in accuracy) ?

Stylistic suggestions, typos, and other very minor things

- Everywhere: a HH → an HH.

- L44: columns → rows.

- L120–123: the behavior of the error depending on the data distribution has been explained in the recent preprint [3], which could be referenced here.

- L157: Assumption 2.3 → MP Setting 2.3.

- L175: the notation $x^{(l)}$, $y^{(l)}$ is introduced but does not seem to be used in the subsequent analysis.

- L233–236: I did not fully understand these two sentences. I think the authors are saying that computations introduce error terms of the form (2.9) while castdowns introduce error terms of the form (2.10). Perhaps consider clarifying or removing these two sentences.

- L255, L261, L382, L384, L733: zeros out → zeroes out.

- L288: remove "between".

- L294: (3.7) show → (3.7) to show ?

- L295: calculating of $\|x\|$ → remove "of".

- L316, L317, L319: I suggest rewriting $v^T xv$ either as $vv^T x$ or $(v^T x)v$.
- (3.13): $\Delta y \to \Delta x$.
- L375: constants,$\beta \to$ missing space.
- L385: $k + 1^{st} \to (k+1)^{th}$, idem at L420.
- L397, L617: TensorCore $\to$ TensorCores ?
- L399: negligble $\to$ negligible.
- L761: "the inner product mixed precision setting yields higher error bounds": specify than what (add a reference to Thm. 3.4 ?)
- L762: I suggest adding a sentence of the form "Before commenting on the significance of Theorem 4.1, we show that the same bounds hold for the BQR variant". Otherwise section 4.2.1 ends a little abruptly and without any analysis of Thm 4.1 (which is in fact to be found later, L775–L783). In fact, perhaps consider merging sections 4.2.1 and 4.2.2 since the analysis and bounds are essentially the same?
- L798: what are d_1 and d_2 here?
- Sections 4.1.2 and 4.2.3: I suggest writing consistently either $2Ln + m2^{-L}$ (as in section 4.1.2) or $m2^{-L} + 2Ln$ (as in section 4.2.3).
- L885: analysis that accurately bound $\to$ change to "analyses" or "bounds".
- Reference [5] has now appeared: cf. [1].

References

[1] Pierre Blanchard, Nicholas J. Higham, Florent Lopez, Theo Mary, and Srikara Pranesh. Mixed precision block fused multiply-add: Error analysis and application to GPU tensor cores. *SIAM J. Sci. Comput.*, 42(3):C124–C141, 2020.

[2] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. xxx+680 pp. ISBN 0-89871-521-0.

[3] Nicholas J. Higham and Theo Mary. Sharper probabilistic backward error analysis for basic linear algebra kernels with random data. MIMS EPrint 2020.4, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, January 2020. 20 pp.