Error analysis of the Gram low-rank approximation (and why it is not as unstable as one may think)

Théo Mary

ERROR ANALYSIS OF THE GRAM LOW-RANK APPROXIMATION (AND WHY IT IS NOT AS UNSTABLE AS ONE MAY THINK)*

THEO MARY†

Abstract. Given $A \in \mathbb{R}^{m\times n}$ and its singular value decomposition (SVD) $U\Sigma V^T$, the eigenvalue decomposition (EVD) of the Gram matrix $G = A^TA$ is $V\Sigma^2V^T$. When $m \gg n$, it is computationally attractive to compute the truncated SVD of A from the truncated EVD of G. This idea has in particular been used to efficiently compress low-rank tensors. In finite precision arithmetic, however, there is a good reason to fear instability from this approach, since the Gram matrix G has condition number $\kappa(A)^2$. We carry out a rounding error analysis of this approach that uses eigenvector perturbation theory. We first explain that a naive application of standard results from this theory leads to an error bound proportional to $\kappa(\bar{A})^2u$, where u is the machine precision and $\kappa(\bar{A})$ is the generalized condition number of the matrix truncated to the target rank. Importantly, this bound is pessimistic and we prove that we can significantly improve it with a more careful analysis. Specifically, we obtain two improvements: first, we show that the error bound is at most proportional to $\kappa(\bar{A})u$, instead of $\kappa(\bar{A})^2u$. Second, we show that regardless of how large $\kappa(\bar{A})$ is, the error cannot exceed $\sqrt{u}$. Hence our final bound is of order $\min(\kappa(\bar{A})u, \sqrt{u})$. Moreover, we also propose the use of iterative refinement to further improve the accuracy in some cases. We illustrate the unusual and attractive behavior of this algorithm with numerical experiments that showcase its effectiveness, despite its partial instability. We believe that our results explain the success that this approach has encountered in large scale tensor computations.

Key words. low-rank approximation, Gram matrix, finite precision arithmetic, rounding error analysis, singular value decomposition, eigenvalue decomposition, iterative refinement, mixed precision

AMS subject classifications. 65F55, 65G50, 65F15, 65Y20

1. Introduction.

Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ and let $U\Sigma V^T$ be its singular value decomposition (SVD), where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal and $\Sigma \in \mathbb{R}^{m\times n}$ is diagonal. A low-rank approximation (LRA) of A can be computed via the truncated SVD

$$\bar{A} = \bar{U}\bar{\Sigma}\bar{V}^T, \quad \bar{U} \in \mathbb{R}^{m\times k}, \bar{\Sigma} \in \mathbb{R}^{k\times k}, \bar{V} \in \mathbb{R}^{n\times k} \tag{1.1}$$

where $\bar{U}$ and $\bar{V}$ correspond to the first k right and left singular vectors, and $\bar{\Sigma}$ to their associated singular values. For a given threshold $\varepsilon > 0$, choosing k as the smallest integer such that $\|\Sigma - \bar{\Sigma}\| \leq \varepsilon\|A\|$ provides a low-rank approximation such that $\|A - \bar{U}\bar{\Sigma}\bar{V}^T\| \leq \varepsilon\|A\|$.

This article is concerned with a Gram LRA approach that consists in computing the Gram matrix $G = A^TA$, computing its eigenvalue decomposition (EVD) $G = W\Lambda W^T$, and using it to recover the LRA $(A\bar{W})\bar{W}^T$ where $\bar{W}$ are the truncated eigenvectors of G, which in exact arithmetic are equal to the truncated right singular vectors $\bar{V}$ of A. This approach is described in Algorithm 1.1 where the truncation rank k is determined based on the eigenvalues in Λ.

Note that if necessary the truncated left singular vectors $\bar{U}$ can be recovered with the observation that

$$A\bar{W}\bar{\Lambda}^{-1/2} = U\Sigma V^T\bar{V}\bar{\Sigma}^{-1} = U\Sigma I_{m,k}\bar{\Sigma}^{-1} = UI_{m,k} = \bar{U},$$

where $I_{m,k}$ is the $m \times k$ identity matrix.

*Version of April 20, 2024.

†Sorbonne Université, CNRS, LIP6, Paris, France (theo.mary@lip6.fr)

Algorithm 1.1 Gram low-rank approximation.

Input: $A \in \mathbb{R}^{m\times n}$, a truncation threshold ε.
Output: $\bar{X}\bar{Y}^T$, a rank-k approximation of A satisfying $\|A - \bar{X}\bar{Y}^T\| \leq \varepsilon\|A\|$.

1: Compute the Gram matrix $G = A^T A \in \mathbb{R}^{n\times n}$.
2: Compute the EVD $G = W\Lambda W^T$.
3: Truncate W and Λ into $\bar{W} \in \mathbb{R}^{n\times k}$ and $\bar{\Lambda} \in \mathbb{R}^{k\times k}$, where k is the smallest integer such that $\|\Lambda - \bar{\Lambda}\| \leq \varepsilon^2\|A\|^2$.
4: Compute $\bar{X} = A\bar{W}$ and set $\bar{Y} = \bar{W}$.

This Gram LRA approach is computationally attractive when $m \gg n$ because its bottleneck lies in the matrix–matrix product $A^T A$, which requires $O(mn^2)$ flops, whereas the EVD of G only requires $O(n^3)$ flops. Gram LRA has been shown to be particularly useful for the rounding (or recompression) of tensors. It was initially proposed for the compression of the tensor train (TT) format [14], and even though a stable rounding approach was later proposed as an alternative [13], Gram LRA remains highly efficient; the recent work of Al Daas, Ballard, and Manning [1] presents a high performance parallel implementation where Gram LRA is a key component. Gram LRA is also a central tool for the rounding of hierarchical Tucker tensors [10]. Indeed, in both of these contexts, we repeatedly compute truncated SVDs of various matricizations $A^{(i)} \in \mathbb{R}^{m_i\times n_i}$ of third-order tensors $\mathcal{A} \in \mathbb{R}^{r_1\times r_2\times r_3}$, with $m_i = \prod_{j\neq i} r_j$ and $n_i = r_i$. We thus indeed have $m \gg n$ (in particular if all the r_i are equal, $m = n^2$).

Unfortunately, in finite precision arithmetic, Gram LRA no longer provides an exact truncated SVD and hence an optimal LRA, and can in fact be unstable. This instability does not come as a surprise due to the computation of the Gram matrix (whose condition number is $\kappa(A)^2$). However, to the best of our knowledge, a thorough rounding error analysis of this approach in inexact arithmetic cannot be found in the literature. The discussion has mainly been limited to experimental observations in the context of the previously mentioned works on tensors [1, 14], which mention in particular the rule of thumb that Gram LRA provides satisfactory restults as long as the truncation threshold ε does not exceed the square root of the machine precision, $\sqrt{u}$. Gram SVD (untruncated) is also briefly discussed in the books of Demmel [4, p. 241] and Trefethen and Bau [17, p. 234]; both references mention a loss of accuracy of order $\sqrt{u}$, but neither provides a thorough analysis or precise error bounds, presumably because this approach was at the time discarded for being too unstable. However, since this approach has experienced a surge in popularity in the context of tensor computing, we believe that a thorough analysis has become of wide interest. The goal of this article is to provide such an analysis in order to determine precisely the attainable accuracy of Gram LRA and to fully characterize its numerical behavior.

Our analysis is based on some basic fundamental results in eigenvector perturbation theory. However, a naive application of these results leads to overly pessimistic error bounds depending on $\kappa(\bar{A})^2$, where $\kappa(\bar{A}) = \sigma_1/\sigma_k$ denotes the generalized spectral condition number of $\bar{A}$. While such a dependence may seem at first sight inevitable due to the computation of the Gram matrix, we actually show that with a careful inspection of the algorithm, we can obtain an error bound that only grows as $\kappa(\bar{A})$. Even better, we show that regardless of how large $\kappa(\bar{A})$ is, the error can never exceed $\sqrt{u}$, the square root of the machine precision, thereby proving the rule of thumb mentioned above. For example, in double precision arithmetic ($u \approx 10^{-16}$), at least eight

significant digits are thus always guaranteed. We believe that this explains why this Gram LRA approach, which may seem at first sight quite unstable, has encountered success in practical applications which use truncation.

One weakness of Gram LRA is that it cannot reliably exploit low precision arithmetics, which provide significant performance benefits on modern hardware. Indeed, the relative error of order $\sqrt{u}$ means that single precision ($u \approx 10^{-8}$) and especially half precision ($u \approx 10^{-4}$) arithmetics are unlikely to be able to deliver satisfactory results except for very crude truncation thresholds ε. Based on the conclusions of our error analysis, we propose a mixed precision iterative refinement approach that can improve the accuracy of the approximation in some cases. This improvement is achieved at the price of some extra operations that must be applied in a higher precision u^2, but that only amount to $O(mn)$ flops; the overhead cost is therefore negligible. Moreover, we also show that the last step of the algorithm, the multiplication of A and $\bar{W}$, is much less sensitive to rounding errors and can therefore be performed in lower precision; thus, at least part of Gram LRA can benefit from the speed of lower precisions.

The rest of this article is structured as follows. In Section 2, with begin with technical preliminaries on eigenvector perturbation theory and give a first pessimistic error bound for Algorithm 1.1 resulting from the naive application of this theory to the entire truncated decomposition. In Section 3, we carry out a refined blockwise error analysis whose main conclusion, summarized in Theorem 3.1, is that the algorithm is unexpectedly not so unstable. Moreover, in Section 4, we propose a mixed precision iterative refinement approach that can in some cases improve the accuracy. In Section 5, we explain that the last step can be performed in lower precision without affecting the accuracy. We provide numerical experiments in Section 6, that illustrate the unusual and attractive behavior of Algorithm 1.1. We provide our concluding remarks in Section 7.

Throughout this article, the unsubscripted norm $\|\cdot\|$ denotes the Frobenius norm.

2. Technical preliminaries and a naive bound.

2.1. Assumptions. For our analysis we will model the inexactness by assuming that the computed EVD of G satisfies

$$W\Lambda W^T = G + \Delta G, \quad \|\Delta G\| \leq c_{m,n} u \|A\|^2, \tag{2.1}$$

where $c_{m,n}$ is a constant that only depends on the dimensions and u is a precision parameter. The error term ΔG can account both for the error incurred in the matrix–matrix product $G = A^T A$ and in the eigendecomposition of G.

We postpone taking into account the error incurred in the matrix product $A\bar{W}$ to Section 5, in which we will show that this product can make use of lower precision arithmetic.

We assume the eigenvectors W to be orthogonal. In finite precision arithmetic W is stored inexactly and thus incurs a moderate loss of orthogonality of order u. Denoting as $\widehat{W}$ these inexactly orthogonal eigenvectors, (2.1) therefore becomes $\widehat{W}\Lambda\widehat{W}^T = G + \Delta G$. However we can define an exactly orthogonal W such that $W = \widehat{W}Z$ and, for some suitable $Z \in \mathbb{R}^{n\times n}$ arising for example from the polar decomposition or QR factorization of $\widehat{W}$, we have $\|\widehat{W} - W\| \leq c_{m,n}u$ [2]. Therefore we can safely assume that (2.1) holds with exactly orthogonal eigenvectors for some constant $c_{m,n}$.

Finally, we assume the perturbation ΔG to be symmetric. This assumption is relatively weak, because the error coming from the matrix product $A^T A$ can certainly

be enforced to be symmetric, either by computing G via a symmetric rank-n update (that is, only g_{ij} is computed and g_{ji} is implicitly considered equal to g_{ij}) or simply by computing G via a general matrix product with deterministic arithmetic (so that the errors in computing g_{ij} and g_{ji} are equal). As for the error coming from the EVD of G, it can also be assumed to be symmetric [2]. We nevertheless mention that our analysis extends to unsymmetric perturbations ΔG by making a few adjustments: first, instead of computing the EVD of G, Algorithm 1.1 must compute its SVD $G = T\Lambda W^T$, with no other changes to the rest of the steps. Second, in the analysis below, the symmetric matrix perturbation results (the Davis–Kahan theorem [3]) must be replaced by their unsymmetric counterpart (the Wedin theorem [18]).

2.2. Eigenvector perturbation theory. Since $V^T\bar{V} = I_{m,k}$ and so $\bar{A} = A\bar{V}\bar{V}^T$, it is natural to bound the error as

$$\begin{aligned}\|A - A\bar{W}\bar{W}^T\| &\leq \|A - \bar{A}\| + \|A(\bar{V}\bar{V}^T - \bar{W}\bar{W}^T)\| \\ &\leq \|A - \bar{A}\| + \|A\|\|\bar{V}\bar{V}^T - \bar{W}\bar{W}^T\|.\end{aligned}$$

We must therefore bound $\|\bar{V}\bar{V}^T - \bar{W}\bar{W}^T\|$, which measures the distance between the subspaces spanned by the first k right singular vectors of A and the first k computed eigenvectors of G. To do so the following classical result of eigenvector perturbation theory will be helpful.

THEOREM 2.1. *Let $A \in \mathbb{R}^{m\times n}$ and let $U\Sigma V^T$ be its SVD, with $\sigma_1 \geq \ldots \geq \sigma_n > 0$. Let $G = A^TA$ and let $W\Lambda W^T = G + \Delta G$ be its computed EVD, where ΔG is a symmetric perturbation and $\lambda_1 \geq \ldots \geq \lambda_n > 0$. Let V_i and W_i be the block-columns of V and W composed of columns s through r. Let*

$$\delta = \min(\sigma_{r-1}^2 - \sigma_r^2, \sigma_s^2 - \sigma_{s+1}^2),$$

where we define $\sigma_0^2 = \infty$ and $\sigma_{n+1}^2 = -\infty$, and assume $\delta > 0$. Then

$$\|V_iV_i^T - W_iW_i^T\| \leq \sqrt{2}\,\frac{\|\Delta G\|}{\delta}. \tag{2.2}$$

Proof. This result is a consequence of [19, Thm. 2], which itself follows from the Davis–Kahan theorem [3], [15, Thm. V.3.6], applied to the EVD of $G+\Delta G = W\Lambda W^T$ and the EVD of $G = V\Sigma^2V^T$. Bound (2.2) is usually stated as $\|\sin\Theta(V_i, W_i)\| \leq \|\Delta G\|/\delta$ where $\Theta(V_i, W_i)$ is the canonical angle between the subspaces spanned by V_i and W_i. To conclude we use $\|V_iV_i^T - W_iW_i^T\| = \sqrt{2}\|\sin\Theta(V_i, W_i)\|$ [15, Eq. (II.4.11)]. ∎

Theorem 2.1 shows that the subspaces spanned by $V_iV_i^T$ and $W_iW_i^T$ will be close under two conditions. First, the perturbation ΔG must be small. Second, the gap δ between the eigenvalues σ_i^2 of G included in the block-column V_i and those not included in it must not be too small, that is, the eigenvalues in this block-column must be well separated from the rest.

2.3. A naive bound. To bound the error $\|A - A\bar{W}\bar{W}^T\|$, we may think of directly applying Theorem 2.1 to the entire vectors $\bar{V}$ and $\bar{W}$, that is, with $r = 1$ and $s = k$. Together with the bound on $\|\Delta G\|$ in (2.1), this yields the bound

$$\|\bar{V}\bar{V}^T - \bar{W}\bar{W}^T\| \leq \sqrt{2}c_{m,n}u\,\frac{\|A\|^2}{\sigma_k^2 - \sigma_{k+1}^2},$$

which is thus of order at least $\kappa(\bar{A})^2 u$. This readily yields the bound

$$\|A - A\bar{W}\bar{W}^T\| \leq \|A - A\bar{V}\bar{V}^T\| + \|A(\bar{V}\bar{V}^T - \bar{W}\bar{W}^T)\| \tag{2.3}$$

$$\leq \left(\varepsilon + \sqrt{2}c_{m,n}u\,\frac{\|A\|^2}{\sigma_k^2 - \sigma_{k+1}^2}\right)\|A\|. \tag{2.4}$$

This bound suggests the error may be very large when the truncated matrix is ill-conditioned, that is, when σ_k is small. In the worst possible case, σ_k may be of order $\varepsilon\|A\|$, yielding a relative error of order at least u/ε^2; thus, to recover the target accuracy ε would require setting the machine precision to $u = \varepsilon^3$, *triple* the target accuracy.

3. A refined blockwise error analysis.

3.1. Analysis. Fortunately, bound (2.4) is overly pessimistic and we may significantly refine it through a more careful inspection. The key idea is to apply Theorem 2.1 not directly on the entire vectors $\bar{V}$, but to individual blocks V_i that are suitably chosen. Let us for now consider an arbitrary block partitioning into q blocks of the form

$$\bar{U} = [U_1 \dots U_q],\ \bar{\Sigma} = \mathrm{diag}(\Sigma_1, \dots, \Sigma_q),\ \bar{V} = [V_1 \dots V_q],\ \bar{W} = [W_1 \dots W_q].$$

We will later specify how to choose the blocks.

We can now write $\bar{A}$ as

$$\bar{A} = \bar{U}\bar{\Sigma}\bar{V}^T = \sum_{i=1}^{q} U_i \Sigma_i V_i^T.$$

Defining $E_i = V_i V_i^T - W_i W_i^T$, we have

$$\begin{aligned}
\bar{A}\bar{W}\bar{W}^T &= \sum_{i=1}^{q} U_i\Sigma_i V_i^T \sum_{j=1}^{q} W_j W_j^T \\
&= \sum_{i=1}^{q} U_i\Sigma_i V_i^T V_i V_i^T \sum_{j=1}^{q} W_j W_j^T \\
&= \sum_{i=1}^{q} U_i\Sigma_i V_i^T (W_i W_i^T + E_i) \sum_{j=1}^{q} W_j W_j^T \\
&= \sum_{i=1}^{q} U_i\Sigma_i V_i^T (W_i W_i^T + E_i\bar{W}\bar{W}^T) \\
&= \sum_{i=1}^{q} U_i\Sigma_i V_i^T (V_i V_i^T - E_i + E_i\bar{W}\bar{W}^T) \\
&= \bar{A} + \sum_{i=1}^{q} U_i\Sigma_i V_i^T E_i (I - \bar{W}\bar{W}^T),
\end{aligned}$$

where we have used $V_i^T V_i = I$, $W_i^T W_i = I$, and $W_i^T W_j = 0$ for $i \neq j$. We therefore obtain the bound

$$\|\bar{A} - \bar{A}\bar{W}\bar{W}^T\| \leq (\sqrt{k} + 1)\sum_{i=1}^{q} \|\Sigma_i\|\|E_i\|$$

since the Frobenius norm is unitarily invariant and so $\|U_i\Sigma_i V_i^T\| = \|\Sigma_i\|$ and $\|\bar{W}\bar{W}^T\| = \sqrt{k}$. Together with the triangle inequality

$$\|A - A\bar{W}\bar{W}^T\| \le \|(A - \bar{A})(I - \bar{W}\bar{W}^T)\| + \|\bar{A} - \bar{A}\bar{W}\bar{W}^T\|$$

we finally obtain

$$\|A - A\bar{W}\bar{W}^T\| \le (\sqrt{k}+1)\left(\varepsilon\|A\| + \sum_{i=1}^{q}\|\Sigma_i\|\|E_i\|\right). \tag{3.1}$$

This is a significantly better bound than the bound of order $\kappa(\bar{A})^2 u$ used in (2.4), because the $\|E_i\|$ terms (which we will soon be bounding using Theorem 2.1) are deamplified by the $\|\Sigma_i\|$ terms. We are thus able to exploit the structure of the singular values of A to improve the bound.

In fact, it is now clear how the block partitioning should be defined. We want the quantities $\|\Sigma_i\|\|E_i\|$ to be as small as possible. Since $\|\Sigma_i\|$ is at least equal to the largest singular value in the block, to minimize $\|\Sigma_i\|$ we should use blocks that are as small as possible. On the other hand, to guarantee that $\|E_i\|$ stays small, we want to maintain a sufficiently large gap between the singular values of different blocks. Hence, the goal is to build blocks that only regroup singular values that are tightly clustered. The key to achieve this is to use gap parameters δ_i that are smaller for blocks corresponding to smaller singular values.

Specifically, we define the blocks as follows: starting with $r = 1$, we define for $i = 1$

$$U_i = [u_r \dots u_s],\ \Sigma_i = \operatorname{diag}(\sigma_r, \dots, \sigma_s), \tag{3.2a}$$
$$V_i = [v_r \dots v_s],\ W_i = [w_r \dots w_s], \tag{3.2b}$$
$$s = \min \mathcal{S} := \left\{ r \le s \le k : \sigma_s^2 - \sigma_{s+1}^2 \ge \delta_i := \frac{\sigma_r^2}{2k} \right\}, \tag{3.2c}$$

and we recursively define the blocks for the next i by updating $r \leftarrow s+1$.

Let us for now focus on the blocks 1 through $q-1$ (we will handle the last block later). With this definition, Theorem 2.1 yields for $i = 1\colon q-1$

$$\|E_i\| \le \sqrt{2}c_{m,n}u\frac{\|A\|^2}{\delta_i} = 2^{3/2}kc_{m,n}u\frac{\|A\|^2}{\|\Sigma_i\|^2}. \tag{3.3}$$

Therefore

$$\|\Sigma_i\|\|E_i\| \le c_{m,n,k}u\frac{\|A\|^2}{\|\Sigma_i\|} \le c_{m,n,k}u\kappa(\bar{A})\|A\| \tag{3.4}$$

shows that we have reduced the error bound from $\kappa(\bar{A})^2 u$ to $\kappa(\bar{A})u$.

We can further refine this bound by observing that

$$\|E_i\| = \|V_iV_i^T - W_iW_i^T\| \le \sqrt{2k_i}, \tag{3.5}$$

where k_i is the size of block i. This means that $\|E_i\|$ cannot exceed a constant factor, and hence bound (3.3) is pessimistic when $u\|A\|^2/\|\Sigma_i\|^2 \gg 1$. Taking this into account, we can improve (3.4) to

$$\|\Sigma_i\|\|E_i\| \le c_{m,n,k}\min\left(u\frac{\|A\|^2}{\|\Sigma_i\|}, \|\Sigma_i\|\right). \tag{3.6}$$

Since the first term in the minimum decreases when $\|\Sigma_i\|$ increases, whereas the second term has the opposite behavior, the worst case value of this bound is achieved when both terms are equal, that is, when

$$u\frac{\|A\|^2}{\|\Sigma_i\|} = \|\Sigma_i\| \quad \Leftrightarrow \quad \|\Sigma_i\| = \sqrt{u}\|A\|.$$

In this case the minimum in (3.6) becomes equal to $\sqrt{u}\|A\|$ and hence, combining (3.6) with (3.4) we obtain, for $i = 1\colon q-1$,

$$\|\Sigma_i\|\|E_i\| \le c_{m,n,k} \min\left(\kappa(\bar{A})u, \sqrt{u}\right)\|A\|. \tag{3.7}$$

All that remains is to handle the case of the last block V_q, which is different because depending on the gap between σ_k and σ_{k+1} the set $\mathcal{S}$ in (3.2c) may be empty. In fact the gap between $\sigma_k > \varepsilon$ and $\sigma_{k+1} \le \varepsilon$ may be arbitrarily small, so that in general we cannot expect a better bound than (3.5), which unlike (3.3), also holds for $i = q$. This is no reason for concern, however, because the construction of the block partitionong (3.2) guarantees that if the gap condition is not met for the last block, that is, if $\sigma_k^2 - \sigma_{k+1}^2 < \delta_q$, then $\|\Sigma_q\|$ must be small. Indeed, let $V_q = [v_r \dots v_k]$ and thus $\delta_q = \sigma_r^2/(2k)$. Then

$$\varepsilon^2\|A\|^2 \ge \sigma_{k+1}^2 \ge \sigma_k^2 - \delta_q \ge \sigma_{k-1}^2 - 2\delta_q \ge \dots \ge \sigma_r^2 - (k-r+1)\delta_q \ge \sigma_r^2 - \frac{k-r+1}{2k}\sigma_r^2 \ge \frac{\sigma_r^2}{2},$$

which shows that

$$\|\Sigma_q\| \le \sqrt{k}\sigma_r \le \sqrt{2k}\varepsilon\|A\|$$

and so

$$\|\Sigma_q\|\|E_q\| \le \sqrt{2k}\sqrt{2k_q}\varepsilon\|A\| \le 2k\varepsilon\|A\|. \tag{3.8}$$

We are now ready to conclude. Going back to (3.1) and bounding each $\|\Sigma_i\|\|E_i\|$ using (3.6) for $i = 1\colon q-1$ and (3.8) for $i = q$, we obtain

$$\begin{aligned}\|A - A\bar{W}\bar{W}^T\| &\le c_{m,n,k}\left(\varepsilon\|A\| + \sum_{i=1}^{q} \min\left(u\frac{\|A\|^2}{\|\Sigma_i\|}, \|\Sigma_i\|\right)\right)\\ &\le c_{m,n,k}\left(\varepsilon + \min\left(\kappa(\bar{A})u, \sqrt{u}\right)\right)\|A\|.\end{aligned}$$

3.2. Summary and discussion. We summarize the conclusions of our analysis in the following theorem.

THEOREM 3.1. *Let $A \in \mathbb{R}^{m\times n}$ and let $\bar{A}$ be its rank-k truncated SVD. Consider a block partitioning of the first k singular values of A, $\bar{\Sigma} = \operatorname{diag}(\Sigma_1, \dots, \Sigma_q)$, as defined in (3.2) which produces blocks Σ_i regrouping singular values that are tightly clustered. Let the approximate truncated SVD $A\bar{W}\bar{W}^T$ be computed with Algorithm 1.1 with precision parameters ε and u controlling the truncation error and the rounding errors, respectively. Then, under the assumptions described in Subsection 2.1, we have*

$$\|A - A\bar{W}\bar{W}^T\| \le c_{m,n,k}\left(\varepsilon\|A\| + \sum_{i=1}^{q} \min\left(u\frac{\|A\|^2}{\|\Sigma_i\|}, \|\Sigma_i\|\right)\right) \tag{3.9}$$

$$\le c_{m,n,k}\left(\varepsilon + \min\left(\kappa(\bar{A})u, \sqrt{u}\right)\right)\|A\| \tag{3.10}$$

where $c_{m,n,k}$ is a constant depending only on the dimensions m, n, and k, and $\kappa(\bar{A}) = \sigma_1/\sigma_k$ is the generalized condition number of $\bar{A}$.

The main conclusion of Theorem 3.1 is that compared with the relative error of order ε introduced by truncation, the use of a finite precision u introduces an additional relative error of order $\min(\kappa(\bar{A})u, \sqrt{u})$. This represents a significant improvement compared with a bound of order $\kappa(\bar{A})^2 u$ obtained in Subsection 2.3 via a straightforward but naive application of eigenvector perturbation theory.

Bound (3.9) is sharp up to constants, as we will show via experiments in the next section. However, note that this is not the case of (3.10), which can be a large overestimate of (3.9) if there is a large gap between the large singular values (for which the first term $u\|A\|^2/\|\Sigma_i\|$ in the minimum is the smaller one) and the small singular values (for which the second term $\|\Sigma_i\|$ is the smaller one). To illustrate this remark, consider an example with two blocks such that $\|\Sigma_1\| \approx \|A\|$ and $\|\Sigma_2\| \approx \varepsilon\|A\|$. Then $\kappa(\bar{A})u \approx u/\varepsilon$ is large and (3.10) thus gives a bound of order $(\varepsilon + \min(u/\varepsilon, \sqrt{u}))\|A\|$. Yet, in (3.9) the minimum is of order $u\|A\|$ for Σ_1 and of order $\epsilon\|A\|$ for Σ_2, hence (3.9) yields a much better bound of order $(\varepsilon + u)\|A\|$.

From a practical point of view, bound (3.10) can be used as follows. Given a prescribed truncation threshold ε, the goal is to decide which precision u is needed to leave the accuracy of the truncated SVD unaffected by the use of finite precision arithmetic. Thus, we should select u sufficiently small so that $\min(\kappa(\bar{A})u, \sqrt{u}) \ll \varepsilon$. In general $\kappa(\bar{A})$ may be hard to estimate, but we know at least that $\kappa(\bar{A}) \leq 1/\varepsilon$. In fact, without specific knowledge on the singular values of A it is not unreasonable to assume $\bar{A}$ will have singular values barely larger than ε, which means that to be safe one could assume $\kappa(\bar{A}) \approx 1/\varepsilon$ in any case. Then $\kappa(\bar{A})u \ll \varepsilon$ translates to $\sqrt{u} \ll \varepsilon$. So we can conclude that without specific knowledge $\sqrt{u} \ll \varepsilon$ is about as good a condition as we might get. This means that in general we should use a precision u that is *double* the target accuracy ε. For example, with double precision arithmetic ($u \approx 10^{-16}$), at least eight significant digits of accuracy are guaranteed, so that Algorithm 1.1 can handle truncation thresholds as small as $\varepsilon = 10^{-8}$. Since truncation thresholds in applications involving low-rank tensors are typically larger, we believe that this explains the success that this algorithm has encountered for these applications.

4. Mixed precision iterative refinement. The analysis of the previous section reveals one weakness of Gram LRA: it cannot reliably exploit low precision arithmetics, which provide significant performance benefits on modern hardware. Indeed, the relative error of order $\sqrt{u}$ means that single precision ($u \approx 10^{-8}$) and especially half precision ($u \approx 10^{-4}$) arithmetics are unlikely to be able to deliver satisfactory results except for very crude truncation thresholds ε.

In this section, we explore the possibility of using iterative refinement to improve the accuracy of the approximation. Indeed, as shown by the bound (3.10), an error of order (at most) $\sqrt{u}$ may be caused by only a small number of eigenvalues corresponding to a group Σ_i such that $\|\Sigma_i\| \approx \sqrt{u}$. This suggests the idea of selectively refining those eigenvalues causing a large error.

Here we will focus on iterative refinement for individual eigenpairs as originally proposed by Dongarra, Moler, and Wilkinson [7, 5, 6]. We mention the recent work of Ogita and Aishima [11, 12] that seeks to refine the entire eigendecomposition, which could also be of interest in our context when most of the eigenvalues are close to $\sqrt{u}$.

Given an approximate eigenpair (w, λ) of $A^T A$ such that $\|w\|_\infty = 1 = w_s$, iterative refinement can be seen as Newton's method applied to the function

$$F(x) = F\left(\begin{bmatrix} w \\ \lambda \end{bmatrix}\right) = \begin{bmatrix} (A^T A - \lambda I)w \\ e_s^T w - 1 \end{bmatrix}$$

whose Jacobian is

$$J(x) = \begin{bmatrix} A^T A - \lambda I & -w \\ e_s^T & 0 \end{bmatrix}.$$

Newton's method then consists in iteratively repeating

$$x \leftarrow x - J(x)^{-1} F(x).$$

Note that $J(x)$ is composed of the Gram matrix $G = A^T A$. Since computing G in higher precision would defeat the purpose of running the algorithm in lower precision, we naturally use the G computed in precision u for $J(x)$. In contrast, using higher precision for evaluating $F(x)$ is paramount and can fortunately be done inexpensively since we only require the action of $A^T A$ on the vector w. Specifically, evaluating $F(x)$ only requires $O(mn)$ flops per iteration. $J(x)$ can be naively inverted for $O(n^3)$ flops, which is already negligible when $m \gg n$, and this cost can be further reduced to $O(n^2)$ by exploiting the approximate eigendecomposition $W \Lambda W^T$ of G [9, sect. 11].

Algorithm 4.1 Gram low-rank approximation, with iterative refinement.

Input: $A \in \mathbb{R}^{m \times n}$, a truncation threshold ε, a tolerance τ, a number of IR steps n_{IR}.
Output: $\bar{X}\bar{Y}^T$, a rank-k approximation of A.

Compute the Gram matrix $G = A^T A \in \mathbb{R}^{n \times n}$ in precision u.
Compute the EVD $G = W \Lambda W^T$ in precision u.
Truncate W and Λ into $\bar{W} \in \mathbb{R}^{n \times k}$ and $\bar{\Lambda} \in \mathbb{R}^{k \times k}$, where k is the smallest integer such that $\|\Lambda - \bar{\Lambda}\| \leq \varepsilon^2 \|A\|^2$.
for $i = 1$ **to** k **do**
  if $\lambda_i \leq \tau$ **then**
    Let $x = \begin{bmatrix} w_i \\ \lambda_i \end{bmatrix}$.
    for $j = 1$ **to** n_{IR} **do**
      Compute $f = F(x)$ in precision u^2.
      Solve $J(x)d = f$ for d in precision u.
      Compute $x = x + d$ in precision u.
    end for
  end if
end for
Compute $\bar{X} = A\bar{W}$ and set $\bar{Y} = \bar{W}$.

We summarize the proposed procedure in Algorithm 4.1. The Gram matrix and its initial eigendecomposition are computed in precision u just like in Algorithm 1.1. Then, we selectively refine eigenpairs, using precision u^2 for evaluating $F(x)$ and precision u for the rest of the operations. The amount of refinement is controlled by two parameters: the number of steps n_{IR}, and the number of refined eigenpairs. The latter is determined by a tolerance τ: we refine any eigenpair of G such that $\lambda_i \leq \tau$.

The error analysis of Tisseur [16] provides conditions for this form of iterative refinement to converge. [16, Corollary 3.4] states that if (w, λ) is a simple eigenpair, if $J(x)$ is not too ill conditioned, if the initial approximation to (w, λ) is sufficiently good, and if the solution d is computed with a stable linear solver, then iterative refinement produces a refined eigenpair with relative accuracy of order u.

In our context, iterative refinement is therefore only applicable to eigenvalues λ of G such that $\lambda u \ll 1$, because if λ is too close or smaller than u, it will be lost

to numerical noise when computing G in precision u. Eigenvalues λ candidate to be refined thus correspond to singular values of A smaller than $\sqrt{u}$. Singular values of order $\sqrt{u}$ are precisely those that maximize the bound (3.9). Therefore, the potential accuracy improvement achievable by iterative refinement completely depends on the singular values of A: if the singular values for which the error bound (3.9) is maximal are large than $\sqrt{u}$, then the accuracy will be improved, but if they are smaller than $\sqrt{u}$, iterative refinement will have no effect.

5. Computing $A\bar{W}$ in lower precision. So far we have ignored the impact of the computation of the final product $\bar{X} = A\bar{W}$ on the accuracy of the approximation $\|A - A\bar{W}\bar{W}^T\| = \|A - \bar{X}\bar{Y}^T\|$. We now show that the accuracy is unsurprisingly much less sensitive to this operation than the ones working on G. As a result, the product $A\bar{W}$ can be computed in lower precision arithmetic without affecting the error bound in Theorem 3.1.

Indeed, denote u_X the precision used for computing $A\bar{W}$. The computed $\bar{X}$ satisfies [8]

$$\bar{X} = A\bar{W} + \Delta X, \quad \|\Delta X\| \leq k u_X \|A\|\|\bar{W}\| = k^{3/2} u_X \|A\|.$$

Hence,

$$\|A - \bar{X}\bar{Y}^T\| = \|A - A\bar{W}\bar{W}^T - \Delta X \bar{W}^T\| \leq \|A - A\bar{W}\bar{W}^T\| + \|\Delta X\|.$$

The computation of $A\bar{W}$ thus only adds an error of order $u_X\|A\|$ to the error $\|A - A\bar{W}\bar{W}^T\|$ bounded in Theorem 3.1. Therefore, in view of (3.9), we can set

$$u_X = \varepsilon + \sum_{i=1}^{q} \min\left(u\frac{\|A\|}{\|\Sigma_i\|}, \frac{\|\Sigma_i\|}{\|A\|}\right)$$

to obtain, up to constants, the same bound on $\|A - \bar{X}\bar{Y}^T\|$ than (3.9). In particular, when the upper bound (3.10) is sharp, this means that we can set u_X as low as $\min(\kappa(\bar{A})u, \sqrt{u})$.

The observation that $A\bar{W}$ can be computed in lower precision can be important, because this product requires $O(mnk)$ flops. Even though this is less than the $O(mn^2)$ flops required for computing $G = A^TA$, this is not completely negligible either, especially in contexts where the rank k is not much smaller than n. An important situation where this arises is when A is the sum of two low-rank matrices (or tensors) that we wish to recompress. In this context, if A is given as the sum of rank-k_1 and k_2 matrices, then $n = k_1 + k_2$ and its rank k after recompression will often (though not always) be larger than $\max(k_1, k_2)$, so that $k \geq n/2$.

6. Experiments. In this section, we perform some numerical experiments that illustrate the conclusions of our analysis. We begin in Subsection 6.1 by evaluating the sharpness of bounds (3.9) and (3.10) and comparing the accuracy of Gram LRA for various matrices with different singular value distributions and condition numbers. Then, we investigate the impact of using iterative refinement in Subsection 6.2 and that of performing $A\bar{W}$ in lower precision in Subsection 6.3.

All our experiments are performed with MATLAB, version R2022b.

6.1. Sharpness of the bounds. To illustrate the sharpness of the bound (3.9) in Theorem 3.1, we use a matrix A generated as follows. We first generate random orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ with $m = 100$ and $n = 50$. We then define $A = U\Sigma V^T$ with

$$\Sigma = \begin{bmatrix} \Sigma_1 & & \\ & \Sigma_2 & \\ & & \Sigma_3 \end{bmatrix} = \begin{bmatrix} I_{k/2} & & \\ & \kappa^{-1} I_{k/2} & \\ & & \varepsilon I_{n-k} \end{bmatrix},$$

where $\varepsilon = 10^{-16}$ and $k = 20$. The rank-k approximation $\bar{A}$ of A is therefore equal to $U\bar{\Sigma}V^T$, where $\bar{\Sigma}$ is equal to Σ with Σ_3 replaced by zero, and $\kappa(\bar{A}) = \kappa$.

We then apply Algorithm 1.1 with a precision $u = 10^{-8}$. For this matrix, bound (3.9) gives an error of order

$$\varepsilon\|A\| + \min\left(u\frac{\|A\|^2}{\|\Sigma_1\|}, \|\Sigma_1\|\right) + \min\left(u\frac{\|A\|^2}{\|\Sigma_2\|}, \|\Sigma_2\|\right).$$

We plot in Figure 6.1 the value of each of these quantities as a function of κ, as well as the actual error $\|A - A\bar{W}\bar{W}^T\|$. For this matrix:

- $u\|A\|^2/\|\Sigma_1\|$ is a constant of order $u = 10^{-8}$ independent of κ;
- $\|\Sigma_1\|$ is a constant of order 1 independent of κ;
- $u\|A\|^2/\|\Sigma_2\| \approx \kappa u$ increases linearly with κ.
- $\|\Sigma_2\| \approx 1/\kappa$ decreases linearly with κ;

Therefore, bound (3.9) follows $u\|A\|^2/\|\Sigma_2\| \approx \kappa u$ for $\kappa \in [1, 10^4]$, $\|\Sigma_2\| \approx 1/\kappa$ for $\kappa \in [10^4, 10^8]$, and $u\|A\|^2/\|\Sigma_1\| \approx u$ for $\kappa \geq 10^8$.

Figure 6.1 not only confirms that bound (3.9) is sharp, since the error closely follows it, but also illustrates the unusual behavior of the error that prevents it from exceeding $\sqrt{u}$ and can even be much smaller, even when $\kappa(\bar{A})$ is large. In particular, the figure shows that the error, just like its bound (3.9), does not monotonically increase with $\kappa(\bar{A})$: in this case, the largest possible error is attained for $\kappa(\bar{A}) = 10^4$.

Next, we perform another experiment to illustrate the difference between bounds (3.9) and (3.10). We generate two matrices A, as previously described, but with different singular values:

$$\begin{aligned} &\text{Mode 2:} \quad \sigma_1 = \ldots = \sigma_{k-1} = 1, \quad \sigma_k = \kappa^{-1} \\ &\text{Mode 3:} \quad [\sigma_1, \ldots, \sigma_k] = \texttt{logspace}(1, \kappa^{-1}, k), \end{aligned}$$

where the MATLAB command `logspace` generates a vector with k logarithmically spaced values decreasing from 1 to κ^{-1}. Figure 6.2 compares bounds (3.9) and (3.10) for these two matrices with the actual error. For both matrices bound (3.9) is sharp. However, bound (3.10) is only sharp for the mode 3 matrix; for the mode 2 one, the error when κ is large is much smaller than both $\sqrt{u}$ and κu. The explanation lies in the discussion following Theorem 3.1: for the mode 2 matrix, there is a large gap between the large singular values of order 1 and the small singular value of order κ^{-1}; for the mode 3 one, there are singular values close to $\sqrt{u}$ and so bound (3.10) is attained. This experiment illustrates that, even though (3.10) is more easily interpretable than (3.9), it is not as sharp and should be used with care.

Finally, we report in Figure 6.3 various results for the same mode 2 and mode 3 matrices for various values of the precision u and the condition number $\kappa(\bar{A})$. The results confirm once more the behavior of the algorithm expected from our analysis, and in particular the fact that its accuracy is always at least $\sqrt{u}$.

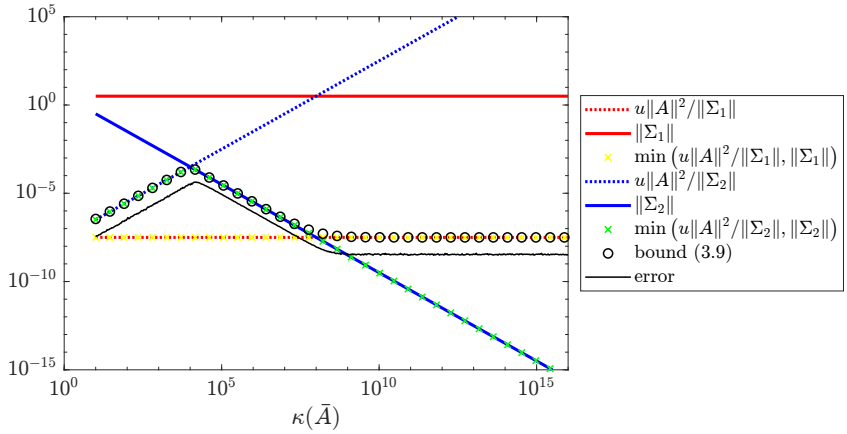


Fig. 6.1: Experimental illustration of bound (3.9).

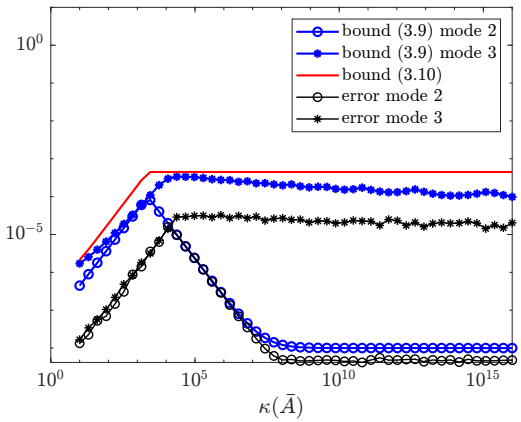
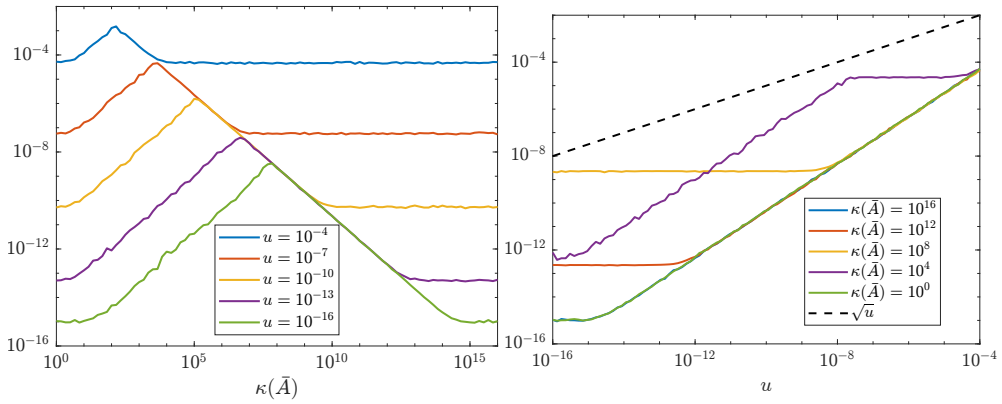


Fig. 6.2: Comparison between bounds (3.9) and (3.10) and the actual error for the mode 2 and mode 3 matrices.

6.2. Results with iterative refinement. We now turn to the use of iterative refinement proposed in Algorithm 4.1. In the following experiments, we compute the Gram matrix G and its initial eigendecomposition in precision $u = 10^{-8}$ and evaluate the residual $F(x)$ in the refinement loop in precision $u^2 = 10^{-16}$.

Figure 6.4 illustrates the accuracy improvement that can be achieved by refining selected eigenpairs. We consider the same mode 2 and mode 3 matrices as before. For the mode 2 matrix, a single eigenpair corresponding to $\lambda_k = \kappa^{-2}$ is responsible for the instability of the algorithm. Therefore, by refining this eigenpair, we can recover stability with an error of order u. This is however only possible when the eigenpair is not too ill-conditioned with respect to the precision u, that is, when $\kappa^{-2}u \ll 1$. Thus, for $u = 10^{-8}$, we can recover the full accuracy as long as $\kappa(\bar{A}) \ll 10^4$. As $\kappa(\bar{A})$ approaches this limit, more steps (larger n_{IR}) are needed to ensure the eigenpair is successfully refined. For larger $\kappa(\bar{A})$, iterative refinement does not converge any



(a) $\kappa(\bar{A})$ varies (mode 2 matrix)

(b) u varies (mode 2 matrix)

(c) $\kappa(\bar{A})$ varies (mode 3 matrix)

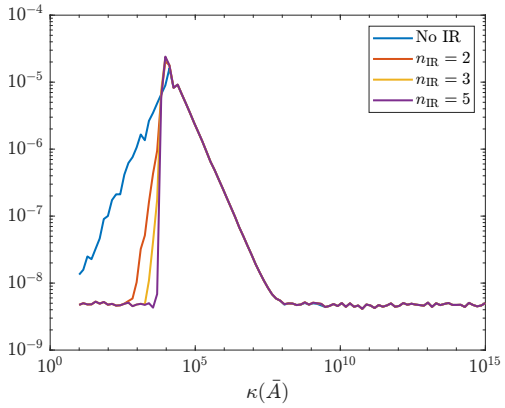
(d) u varies (mode 3 matrix)

Fig. 6.3: Accuracy for various matrices and values of $\kappa(\bar{A})$ and u.

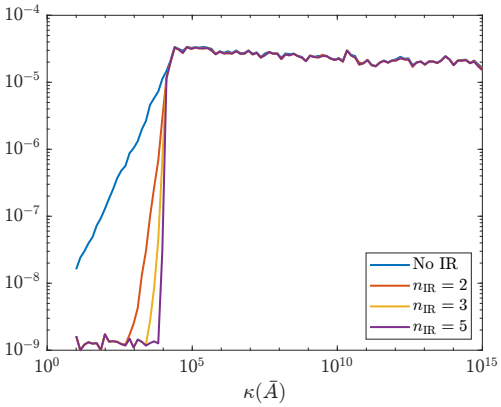
longer and is thus not helpful. For the mode 2 matrix, the tolerance τ does not play any role since there is only one (very small) eigenpair that needs to be refined.

The same observations hold for the mode 3 matrix, with the difference that more than one eigenpair must be refined. Setting $\tau = 0.9$ refines almost all k eigenpairs, and yields an accuracy of order u. Smaller values of τ mean less pairs are refined, and the achievable accuracy is then determined by the smallest eigenvalue λ_i that is not refined: thus, we achieve an accuracy of order $u/\sqrt{\tau}$. In any case, even when all k eigenpairs must be refined, the cost remains in $O(mnk)$ flops, which may be acceptable if $k \ll n$.

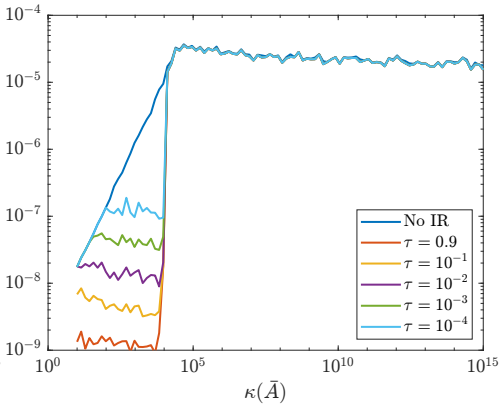
6.3. Results with $A\bar{W}$ in lower precision. We conclude this section by experimentally investigating the effect of the precision u_X used to compute the product $\bar{X} = A\bar{W}$. Figure 6.5 confirms the discussion in Section 5: computing $\bar{X}$ inexactly adds an error of order $u_X\|A\|$. Therefore, depending on the singular values of the matrix, it is possible to set u_X to a much lower precision than u (as low as $\sqrt{u}$) without affecting the overall accuracy.



(a) Mode 2 matrix ($\tau = 0.9$, n_{IR} varies)

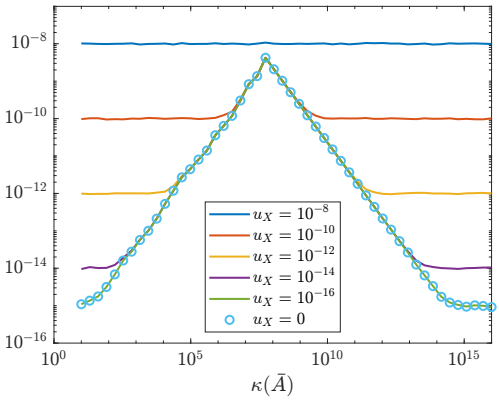


(b) Mode 3 matrix ($\tau = 0.9$, n_{IR} varies)

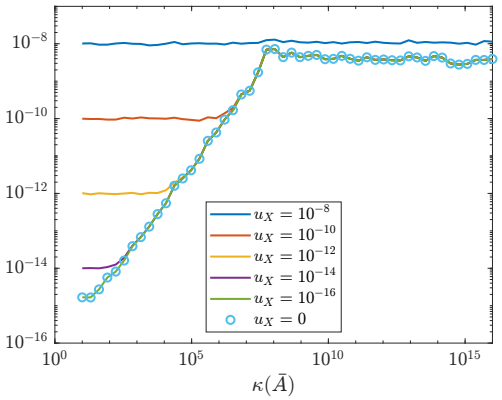


(c) Mode 3 matrix ($n_{\mathrm{IR}} = 5$, τ varies)

Fig. 6.4: Effect of iterative refinement on the accuracy ($u = 10^{-8}$).



(a) Mode 2



(b) Mode 3

Fig. 6.5: Effect of the precision u_X of the product $\bar{X} = A\bar{W}$ on the accuracy ($u = 10^{-16}$).

7. Conclusion. We have analyzed a Gram low-rank approximation (Gram LRA) approach, which has generated interest in the context of low-rank tensor computations. Despite the computation of the Gram matrix, we have shown that Gram LRA is in fact much less unstable than one may think. We have obtained in Theorem 3.1 a blockwise error bound that takes into account the structure of the singular values of the matrix, which leads to a refined bound. In particular, the bound cannot exceed the square root of the machine precision, $\sqrt{u}$, so that if the truncation threshold ε is sufficiently larger than $\sqrt{u}$, the effect of finite precision arithmetic will go unnoticed. We believe that this explains the success of this approach when using double precision arithmetic, since ε is typically larger than $\sqrt{u} \approx 10^{-8}$ in applications.

We have also proposed two new ideas to improve the algorithm in a finite precision setting. The first is to use mixed precision iterative refinement to refine the small eigenvalues of the Gram matrix, which can lead to a significant accuracy improvement in some cases. The second idea is to accelerate the final multiplication between the original matrix and the computed eigenvectors by performing it in lower precision, which can be done without affecting the accuracy because this operation is much less sensitive to rounding errors than the operations involving the Gram matrix.

We have performed extensive numerical experiments that confirm that our error bounds are sharp and correctly describe the unusual numerical behavior of this algorithm.

Acknowledgments. We thank Oguz Kaya and Matthieu Robeyns for helpful discussions on the use of Gram LRA in tensor rounding methods.

Funding. This work was partially supported by the InterFLOP (ANR-20-CE46-0009), MixHPC (ANR-23-CE46-0005-01) and NumPEx ExaMA (ANR-22-EXNU-0002) projects of the French National Agency for Research (ANR).

REFERENCES

[1] H. Al Daas, G. Ballard, and L. Manning, *Parallel tensor train rounding using Gram SVD*, in 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2022, pp. 930–940, https://doi.org/10.1109/IPDPS53621.2022.00095.

[2] S. Chandrasekaran and I. C. Ipsen, *Backward errors for eigenvalue and singular value decompositions*, Numer. Math., 68 (1994), pp. 215–223.

[3] C. Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation III*, SIAM J. Numer. Anal., 7 (1970), pp. 1–46, https://doi.org/10.1137/0707001.

[4] J. W. Demmel, *Applied numerical linear algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[5] J. J. Dongarra, *Improving the accuracy of computed matrix eigenvalues*, Preprint ANL-80-84, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, USA, Aug. 1980, https://doi.org/10.2172/5047973.

[6] J. J. Dongarra, *Algorithm 589 SICEDR: A FORTRAN subroutine for improving the accuracy of computed matrix eigenvalues*, ACM Trans. Math. Software, 8 (1982), pp. 371–375, https://doi.org/10.1145/356012.356016.

[7] J. J. Dongarra, C. B. Moler, and J. H. Wilkinson, *Improving the accuracy of computed eigenvalues and eigenvectors*, SIAM J. Numer. Anal., 20 (1983), pp. 23–45, https://doi.org/10.1137/0720002.

[8] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second ed., 2002, https://doi.org/10.1137/1.9780898718027.

[9] N. J. Higham and T. Mary, *Mixed precision algorithms in numerical linear algebra*, Acta Numerica, 31 (2022), pp. 347–414, https://doi.org/10.1017/s0962492922000022.

[10] D. Kressner and C. Tobler, *Algorithm 941: Htucker—A MATLAB toolbox for tensors in hierarchical Tucker format*, ACM Trans. Math. Software, 40 (2014), pp. 1–22, https://doi.org/10.1145/2538688.

[11] T. Ogita and K. Aishima, *Iterative refinement for symmetric eigenvalue decomposition*, Japan J. Indust. Appl. Math., 35 (2018), p. 1007–1035, https://doi.org/10.1007/s13160-018-0310-3.

[12] T. Ogita and K. Aishima, *Iterative refinement for symmetric eigenvalue decomposition II: Clustered eigenvalues*, Japan J. Indust. Appl. Math., 36 (2019), pp. 435–459, https://doi.org/10.1007/s13160-019-00348-4.

[13] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317, https://doi.org/10.1137/090752286.

[14] I. V. Oseledets and E. E. Tyrtyshnikov, *Breaking the curse of dimensionality, or how to use SVD in many dimensions*, SIAM J. Sci. Comput., 31 (2009), pp. 3744–3759, https://doi.org/10.1137/090748330.

[15] G. W. Stewart and J.-g. Sun, *Matrix perturbation theory*, Elsevier, Amsterdam, The Netherlands, 1990.

[16] F. Tisseur, *Newton's method in floating point arithmetic and iterative refinement of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1038–1057, https://doi.org/10.1137/S0895479899359837.

[17] L. N. Trefethen and D. Bau, *Numerical linear algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2022.

[18] P.-Å. Wedin, *Perturbation bounds in connection with singular value decomposition*, BIT Numerical Mathematics, 12 (1972), pp. 99–111, https://doi.org/10.1007/BF01932678.

[19] Y. Yu, T. Wang, and R. J. Samworth, *A useful variant of the Davis–Kahan theorem for statisticians*, Biometrika, 102 (2015), pp. 315–323, https://doi.org/10.1093/biomet/asv008.