SIAM J. MATRIX ANAL. APPL.
Vol. 42, No. 1, pp. 275–298

ON THE COMPRESSIBILITY OF TENSORS*

TIANYI SHI† AND ALEX TOWNSEND‡

Abstract. Tensors are often compressed by expressing them in data-sparse tensor formats, where storage costs in such formats are less than those in the original structure. In this paper, we develop three methodologies that bound the compressibility of a tensor: (1) Algebraic structure, (2) smoothness, and (3) displacement structure. For each methodology, we derive bounds on storage costs in various low rank tensor formats that partially explain the abundance of compressible tensors in applied mathematics. For example, using displacement structure, we show that the solution tensor $\mathcal{X} \in \mathbb{C}^{n\times n\times n}$ of a discretized Poisson equation $-\nabla^2 u = 1$ on $[-1,1]^3$ with zero Dirichlet conditions can be approximated to a relative accuracy of $0 < \epsilon < 1$ in the Frobenius norm by a tensor in tensor-train format with $\mathcal{O}(n(\log n)^2(\log(1/\epsilon))^2)$ degrees of freedom. The constructive bound also allows us to design a spectral algorithm that solves this equation with $\mathcal{O}(n(\log n)^3(\log(1/\epsilon))^3)$ complexity.

Key words. numerical low rank, tensor-train, multilinear, canonical polyadic, displacement

AMS subject classifications. 15A69, 65F99

DOI. 10.1137/20M1316639

1. Introduction.

A wide variety of applications, such as approximation theory [34], continuum mechanics [12], differential equations [33, 37], and data analysis [40], lead to problems involving data or solutions that can be represented by tensors [36]. A general d-order tensor $\mathcal{X} \in \mathbb{C}^{n_1\times\cdots\times n_d}$ has $\prod_{k=1}^{d} n_k$ entries, which prevents it from being stored explicitly except for modest d. It is often essential to represent or approximate tensors using sparse data formats, such as low rank tensor decompositions [18, 36]. However, the need for data-sparse formats does not imply that such approximations are always mathematically possible. In this paper, we derive bounds on numerical storage costs (see section 2) for certain families of tensors, and in doing so, we partially justify the use of low rank tensor decompositions. Analogous theoretical results have already been derived that explicitly bound the numerical rank of matrices [4, 42, 51, 56].

The situation for tensors is more complicated than for matrices, and this is reflected in several distinct low rank tensor decompositions [17, 36, 47]. Here, we consider three such decompositions: (a) Tensor-train (TT) decomposition (see subsection 2.1), (b) orthogonal Tucker decomposition (see subsection 2.2), and (c) canonical polyadic (CP) decomposition (see subsection 2.3). These three tensor decompositions supply three different definitions of tensor rank, and therefore each one requires separate attention.

For a given tensor $\mathcal{X} \in \mathbb{C}^{n_1\times\cdots\times n_d}$, we are interested in developing a variety of tools for theoretically explaining whether there exists a low rank tensor $\tilde{\mathcal{X}} \in \mathbb{C}^{n_1\times\cdots\times n_d}$, in one or more of the tensor formats, such that

$$\|\mathcal{X} - \tilde{\mathcal{X}}\|_F \le \epsilon\|\mathcal{X}\|_F, \qquad \|\mathcal{X}\|_F^2 = \sum_{i_1=1}^{n_1}\cdots\sum_{i_d=1}^{n_d} |\mathcal{X}_{i_1,\ldots,i_d}|^2, \tag{1.1}$$

*Received by the editors February 3, 2020; accepted for publication (in revised form) December 9, 2020; published electronically March 2, 2021.
https://doi.org/10.1137/20M1316639
Funding: The work of the authors was supported by National Science Foundation grant 1818757.
†Center for Applied Mathematics, Cornell University, Ithaca, NY 14853 USA (ts777@cornell.edu).
‡Department of Mathematics, Cornell University, Ithaca, NY 14853 USA (townsend@cornell.edu).

TABLE 1.1

Summary of the bounds of storage costs in tensor-train format of $n \times n \times n$ tensors explored in this article. The numbers s_1, s_2, and s_3 are given in their corresponding sections. Here, FD stands for finite difference.

Tensor	TT Storage bound	Method	Ref.
Sampled $e^{iM\pi xyz}$	$\mathcal{O}(M)$	Smoothness	subsection 3.3.1
Sampled sum of Gaussian bumps	Implicit	Smoothness	subsection 3.3.2
3D Hilbert tensor	$n(s_1^2 + 2s_1)$	Displacement	subsection 4.5.1
Poisson FD soln	$n(s_2^2 + 2s_2)$	Displacement	subsection 4.5.2
Poisson spectral soln	$n(s_3^2 + 2s_3)$	Displacement	subsection 4.5.2

where $0 \le \epsilon < 1$ is an accuracy tolerance. If $\mathcal{X}$ can be well approximated by $\tilde{\mathcal{X}}$, then dramatic storage and computational benefits can be achieved by replacing $\mathcal{X}$ with $\tilde{\mathcal{X}}$ [18, 20]. We say that a tensor is compressible if it can be approximated by a low rank tensor, in the sense of (1.1) that can be represented in a relatively small number of degrees of freedom. In order to compare different low rank tensor formats, we examine the number of degrees of freedom required to store an approximate tensor.

In this paper, we explore three methodologies for bounding the compressibility of a tensor:

- **Algebraic structures:** If a tensor is constructed by sampling a multivariable function that can be expressed as a sum of products of single-variable functions, then that tensor is often compressible. Occasionally, one may have to perform algebraic manipulations on a function to explicitly reveal its desired structure, for example, by using trigonometric identities (see subsection 3.1).
- **Smoothness:** If a tensor can be constructed by sampling a smooth function on a tensor-product grid, then that tensor is often compressible. This observation can be made rigorous by using the fact that smooth functions can be well approximated by polynomials in a compact domain [57, Thm. 8.2], [20, Lem. 14.5].
- **Displacement structure:** If a tensor $\mathcal{X}$ satisfies a multidimensional Sylvester equation of the form

$$\mathcal{X} \times_1 A^{(1)} + \cdots + \mathcal{X} \times_d A^{(d)} = \mathcal{G}, \qquad A^{(k)} \in \mathbb{C}^{n_k \times n_k}, \quad \mathcal{G} \in \mathbb{C}^{n_1 \times \cdots \times n_d}, \tag{1.2}$$

  where "$\times_k$" denotes the k-mode matrix product of a tensor (see (1.3)), then this—under additional assumptions—can ensure that the tensor $\mathcal{X}$ is well approximated by a low rank tensor. Multidimensional Sylvester equations, such as (1.2), appear when discretizing certain partial differential equations (PDEs) with finite differences [39] and are satisfied by several classes of structured tensors [19]. For example, we show that the solution tensor $\mathcal{X} \in \mathbb{C}^{n \times n \times n}$ to $-\nabla^2 u = 1$ on $[-1, 1]^3$ with zero Dirichlet conditions can be represented up to a relative accuracy of $0 < \epsilon < 1$ in the Frobenius norm with just $\mathcal{O}(n(\log n)^2(\log(1/\epsilon))^2)$ degrees of freedom in tensor-train format, despite the solution having weak corner singularities.

The first two methodologies are considered in [20]. The third methodology is used in various aspects of numerical linear algebra. For example, one can explicitly bound the singular values of matrices with displacement structure [4, 56], which can make matrix-vector multiplication [21] and the solution of linear systems [16, 44] more computationally efficient. The third methodology is also related to the technique of bounding singular values using exponential sums [9, 29]. However, we are not aware of any existing literature that compress tensors with displacement structure.

In this article, we formally provide bounds on the compressibility of such tensors and illustrate the methodologies with worked-out examples. Table 1.1 summarizes our bounds on the storage cost of several special tensors.

After some experience, one can successfully identify which methodology is likely to result in the best theoretical bounds on the compressibility of a tensor. We emphasize that these three methodologies provide upper bounds using numerical tensor ranks, but we do not provide a complete characterization on the compressibility of tensors. Another approach that partially explains the abundance of tensors with small storage is artificial coordinate alignment [58], though we do not know how to use this approach to derive explicit bounds on tensor ranks.

1.1. Tensor notation. We follow as closely as possible the notation for tensors found in [36], which we briefly review now for the reader's convenience.

The *k*-mode product. The k-fold (or k-mode) product of a tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ with a matrix $A \in \mathbb{C}^{n_k \times n_k}$ is denoted by $\mathcal{X} \times_k A$ and defined elementwise as

$$(\mathcal{X} \times_k A)_{i_1,\ldots,i_{k-1},j,i_{k+1},\ldots,i_d} = \sum_{i_k=1}^{n_k} \mathcal{X}_{i_1,\ldots,i_d} A_{j,i_k}. \tag{1.3}$$

It corresponds to each mode-k fiber of X being multiplied by the matrix A.

Double bracket. In the tensor literature, the double bracket denotes a mapping from the parametric space to the space of tensors. Specifically, it can be considered as a weighted sum of rank-1 tensors, i.e.,

$$[\![\mathcal{G}; A^{(1)}, \ldots, A^{(d)}]\!] = \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} \mathcal{G}_{i_1,\ldots,i_d} A^{(1)}_{i_1} \circ \cdots \circ A^{(d)}_{i_d}, \qquad A^{(k)} \in \mathbb{C}^{n_k \times r_k}, \tag{1.4}$$

where $\mathcal{G} \in \mathbb{C}^{r_1 \times \cdots \times r_d}$ is often referred to as the core tensor, and $v_1 \circ \cdots \circ v_d$ is the d-way outer-product of vectors [36].

Flattening by reshaping. One can always reorganize the entries of a tensor into a matrix, and this idea is fundamental to the tensor-train decomposition [47]. Conventionally, one reorganizes the entries so that the mode-1 fibers are stacked. This is equivalent to $X_k = \text{reshape}(\mathcal{X}, \prod_{s=1}^{k} n_s, \prod_{s=k+1}^{d} n_s)$.[1] We call X_k the kth unfolding of $\mathcal{X}$.

Flattening via matricization. Another way to flatten a tensor is called mode-n matricization (or nth matricization), which arranges the mode-n fibers into columns of a matrix [35]. We denote the mode-n matricization of a tensor $\mathcal{X}$ by $X_{(n)}$. It is easy to see that the first unfolding and the mode-1 matricization of a tensor are identical, i.e., $X_{(1)} = X_1$. In this paper, for a tensor $\mathcal{X}$, matricizations are constructed so that there exists another tensor $\mathcal{Y}^j$ satisfying [10],

$$Y^j_{(1)} = X_{(j)},\ \ldots,\ Y^j_{(d-j+1)} = X_{(d)},\ Y^j_{(d-j+2)} = X_{(1)},\ \ldots,\ Y^j_{(d)} = X_{(j-1)}. \tag{1.5}$$

1.2. Summary of paper. In the next section, we review three tensor decompositions, and in subsection 3.1 we study the ranks of tensors that are constructed by sampling multivariate functions that have some algebraic structure. In subsection 3.2,

[1] In MATLAB, the reshape command reorganizes the entries of a tensor. For example, if $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$, then $\text{reshape}(\mathcal{X}, \prod_{s=1}^{k} n_s, \prod_{s=k+1}^{d} n_s)$ returns a matrix of size $(\prod_{s=1}^{k} n_s) \times (\prod_{s=k+1}^{d} n_s)$ formed by stacking entries according to their multi-index.

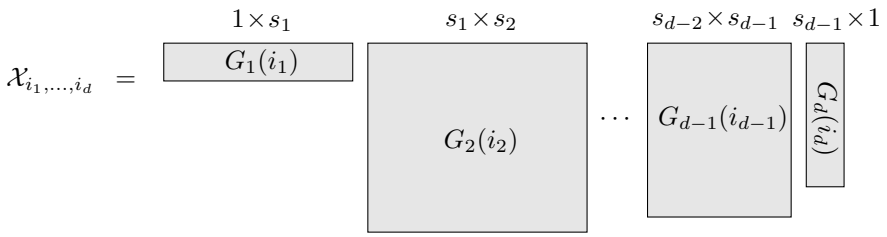

FIG. 2.1. *The tensor-train decomposition of rank at most* $\boldsymbol{s} = (s_0, \ldots, s_d)$. *Each entry of a tensor is represented by the product of* d *matrices, where the* k*th matrix in the "train" is selected based on the value of* i_k.

we consider the storage cost of tensors constructed by sampling smooth multivariate functions. Finally, in section 4 we consider tensors that satisfy a multidimensional Sylvester equation, including a fast tensor Sylvester equation solver that exploits the compressibility of these tensors in subsection 4.6.

2. Three tensor decompositions. In this section, we review three tensor decompositions: (a) Tensor-train decomposition, (b) orthogonal Tucker decomposition, and (c) CP decomposition. For each decomposition and a given $0 \le \epsilon < 1$, we say a tensor $\mathcal{X}$ has p_ϵ numerical storage cost if there exists a tensor $\tilde{\mathcal{X}}$ that can be represented with p_ϵ degrees of freedom and $||\mathcal{X} - \tilde{\mathcal{X}}||_F \le \epsilon ||\mathcal{X}||_F$. In this article, we are most interested in tensors that are not precisely low rank, so we focus on the situation when $0 < \epsilon < 1$.

2.1. Tensor-train decomposition. The tensor-train decomposition is a generalization of the singular value decomposition (SVD) that can be computed by a sequence of matrix SVDs [47, 49]. A tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ has a tensor-train rank of at most $\boldsymbol{s} = (s_0, \ldots, s_d)$ if there exists matrix-valued functions $G_k : \{1, \ldots, n_k\} \mapsto \mathbb{C}^{s_{k-1} \times s_k}$ for $1 \le k \le d$ such that

$$\mathcal{X}_{i_1,\ldots,i_d} = G_1(i_1) G_2(i_2) \cdots G_d(i_d), \qquad 1 \le i_k \le n_k. \tag{2.1}$$

This decomposition writes each entry of $\mathcal{X}$ as a product of matrices, where the kth matrix in the "train" of length d is determined by i_k. Since the product of the matrices must always be a scalar, we have $s_0 = s_d = 1$. Each G_k can be represented by an $s_{k-1} \times n_k \times s_k$ tensor, so a tensor-train decomposition of rank at most $\boldsymbol{s}$ requires $p^{\mathrm{TT}} \le \sum_{k=1}^{d} s_{k-1} s_k n_k$ degrees of freedom to store the format in memory. Figure 2.1 illustrates a tensor-train decomposition of rank at most $\boldsymbol{s} = (s_0, \ldots, s_d)$.

Normally a tensor-train decomposition is constructed by separating out one dimension at a time and compressing each dimension in turn [47]. For simplicity, the decomposition considered in this paper is performed in the order of dimension 1, dimension 2, and so on. In this way, the entries of the tensor-train rank are bounded from above by the ranks of matrices formed by flattening [47]. That is, for $1 \le k \le d-1$ we have

$$s_k \le \operatorname{rank}(X_k), \qquad X_k = \operatorname{reshape}\left(\mathcal{X}, \prod_{s=1}^{k} n_s, \prod_{s=k+1}^{d} n_s\right), \tag{2.2}$$

where $\operatorname{rank}(X_k)$ is the rank of the matrix X_k. Therefore, if the ranks of all the matrices X_k for $1 \le k \le d-1$ are small, then the tensor X can be exactly represented in a data-sparse format as a tensor-train decomposition.

2.2. Orthogonal Tucker decomposition. The orthogonal Tucker decomposition is a factorization of a tensor into a set of matrices and a core tensor, where the matrices have orthonormal columns [10, 25, 36]. A tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ has a multilinear rank[2] of at most $\boldsymbol{t} = (t_1, \ldots, t_d)$ if there are matrices $A_1, \ldots, A_d$ with orthonormal columns and a core tensor $\mathcal{G} \in \mathbb{C}^{t_1 \times \cdots \times t_d}$ such that

$$\mathcal{X} = [\![\mathcal{G}; A^{(1)}, \ldots, A^{(d)}]\!], \qquad A^{(k)} \in \mathbb{C}^{n_k \times t_k}. \tag{2.3}$$

Such a decomposition contains $p^{\mathrm{ML}} \leq \sum_{k=1}^{d} n_k t_k + \prod_{k=1}^{d} t_k$ degrees of freedom, and can be computed by the so-called higher-order SVD [10].

2.3. Canonical polyadic decomposition. The CP decomposition expresses a tensor as a sum of rank-1 tensors. A tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ is of rank at most r if there are matrices $A^{(1)}, \ldots, A^{(d)}$ and a diagonal tensor D such that

$$\mathcal{X} = [\![\mathcal{D}; A^{(1)}, \ldots, A^{(d)}]\!], \qquad A^{(k)} \in \mathbb{C}^{n_k \times r}, \quad \mathcal{D} \in \mathbb{C}^{r \times \cdots \times r}, \tag{2.4}$$

where the only nonzero entries of $\mathcal{D}$ are $\mathcal{D}_{i,\ldots,i}$ for $1 \leq i \leq r$. If $\mathcal{D}$ is omitted in this bracket notation, then by convention, all the nonzero entries of $\mathcal{D}$ are 1. This tensor decomposition can be stored using $p^{\mathrm{CP}} \leq r + r \sum_{k=1}^{d} n_k$ degrees of freedom, but the decomposition is NP-hard to compute for worst case examples [23]. The CP decomposition in (2.4) is similar to the orthogonal Tucker decomposition with two important differences: (1) The matrices $A^{(1)}, \ldots, A^{(d)}$ in (2.4) do not need to have orthogonal columns, and (2) the core tensor $\mathcal{D}$ must be diagonal. This means that (2.4) is equivalent to expressing a tensor as a sum of rank-1 tensors.

Since we are aiming for upper bounds on the rank of a tensor to bound compressibility of a tensor in CP format, we can take any decomposition of the form in (2.4) with a potentially large r and see if its factor matrices $A^{(1)}, \ldots, A^{(d)}$ are themselves low rank. For example, we find that [38, Lem. 1][3]

$$\operatorname{rank}(\mathcal{X}) \leq \min_{1 \leq j \leq d} \frac{1}{r_j} \prod_{i=1}^{d} r_i, \tag{2.5}$$

where $r_i = \operatorname{rank}(A^{(i)})$ for $1 \leq i \leq d$. The bound in (2.5) is useful because it allows one to derive upper bounds on the rank of a tensor via bounds on the rank of factor matrices.

3. Tensors derived by sampling smooth functions. One often finds that tensors derived from sampling smooth functions are compressible, and we make this observation precise. Tensors derived from sampling multivariate functions have been considered in [26, 30], and analogous results for matrices are available in the literature [51, 55].

3.1. Tensors constructed via sampling algebraically structured functions. In practice, one often encounters tensors that are sampled from multivariate functions. For example, one can take a continuous function of three variables,

[2]The closely related Tucker rank of a tensor is also associated to the Tucker decomposition, except the matrices $A^{(k)}$ in (2.3) are not constrained to have orthonormal columns. Since multilinear ranks are more commonly used in applications, we do not consider Tucker ranks in this paper.

[3]Lemma 1 of [38] shows that the dimension of the vector space that spans the slices in the νth index is equal to the rank of $\mathcal{X}$. The inequality in (2.5) follows from the additional assumption that the slices are themselves low rank tensors.

$f(x, y, z)$, and sample f on a tensor grid to obtain a tensor

$$\mathcal{X}_{ijk} = f(x_i, y_j, z_k), \qquad 1 \le i, j, k \le n,$$

where $\{x_1, \ldots, x_n\}$, $\{y_1, \ldots, y_n\}$, and $\{z_1, \ldots, z_n\}$ are sets of points.

3.1.1. Polynomials and algebraic structure. One common scenario where it is easy to spot compressible tensors is when the tensor is sampled from a polynomial. To be specific, if a tensor $\mathcal{X}$ is derived by sampling a multivariate polynomial $p(x_1, \ldots, x_d)$ of degree at most $N_j - 1$ in the variable x_j from a tensor-product grid, then one finds that $\mathcal{X}$ is highly compressible.

LEMMA 3.1. *Let $p(x_1, \ldots, x_d)$ be a polynomial of degree at most $N_j - 1$ in the variable x_j for $1 \le j \le d$, and let $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ be the tensor constructed by sampling p, i.e.,*

$$\mathcal{X}_{i_1,\ldots,i_d} = p(x_{i_1}^{(1)}, \ldots, x_{i_d}^{(d)}), \qquad 1 \le i_j \le n_j, \qquad 1 \le j \le d,$$

where $x^{(1)}, \ldots, x^{(d)}$ are sets of $n_1, \ldots, n_d$ nodes, respectively. Then,
- $p^{\mathrm{TT}}(\mathcal{X}) \le \sum_{k=1}^{d} t_{k-1} t_k n_k$, *where* $t_k = \min\{\prod_{j=1}^{k} N_j, \prod_{j=k+1}^{d} N_j\}$;
- $p^{\mathrm{ML}}(\mathcal{X}) \le \sum_{k=1}^{d} n_k N_k + \prod_{k=1}^{d} N_k$; *and*
- $p^{\mathrm{CP}}(\mathcal{X}) \le r + r \sum_{k=1}^{d} n_k$, *where* $r = \min_{1 \le k \le d} \frac{1}{N_k} \prod_{j=1}^{d} N_j$.

Here, the tensor-train decomposition is constructed in the order $x_1, \ldots, x_d$.

Proof. According to the degree assumptions on p, we can write p as

$$p(x_1, \ldots, x_d) = \sum_{q_1=0}^{N_1-1} \cdots \sum_{q_k=0}^{N_k-1} a_{q_1,\ldots,q_k}(x_{k+1}, \ldots, x_d) x_1^{q_1} \cdots x_k^{q_k}, \qquad 1 \le k \le d,$$

where $a_{q_1,\ldots,q_k}(x_{k+1}, \ldots, x_d)$ is a polynomial in the variables $x_{k+1}, \ldots, x_d$ and for $k + 1 \le j \le d$, x_j has degree at most $N_j - 1$. After sampling, this means that $\mathrm{rank}(X_k) \le \min\{\prod_{j=1}^{k} N_j, \prod_{j=k+1}^{d} N_j\}$, and the bound on $p^{\mathrm{TT}}(\mathcal{X})$ follows.

Another way to write p is

$$p(x_1, \ldots, x_d) = \sum_{j=0}^{N_k-1} b_j(x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d) x_k^j, \qquad 1 \le k \le d,$$

where b_j is a polynomial in $x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_d$ of degree at most $N_j - 1$ in x_j. After sampling, this shows that $\mathrm{rank}(X_{(k)}) \le N_k$, and the bound on $p^{\mathrm{ML}}(\mathcal{X})$ follows.

Finally, separating out x_d, we can also write p as

$$p(x_1, \ldots, x_d) = \sum_{q_1=0}^{N_1-1} \cdots \sum_{q_{d-1}=0}^{N_{d-1}-1} c_{q_1,\ldots,q_{d-1}}(x_d) x_1^{q_1} \cdots x_{d-1}^{q_{d-1}}, \tag{3.1}$$

where each term in (3.1) is a rank-1 tensor after sampling. We can do this to each variable, and thus $\mathrm{rank}(\mathcal{X}) \le \min_{1 \le k \le d} \frac{1}{N_k} \prod_{j=1}^{d} N_j$. The bound on $p^{\mathrm{CP}}(\mathcal{X})$ follows. ∎

A special case of Lemma 3.1 is when the polynomial p has maximal degree of at most $N - 1$ so that $N_1 = \cdots = N_d = N$.[4] We find that

[4] We say that a polynomial $p_N(x_1, \ldots, x_d)$ has maximal degree $\le N$ if p_N is a polynomial of degree at most N in all the variables x_i.

- $p^{\mathrm{TT}}(\mathcal{X}) \leq \sum_{k=1}^{d} N^{2t_k-1} n_k$, where $t_k = \min\{k, d-k\}$;
- $p^{\mathrm{ML}}(\mathcal{X}) \leq N \sum_{k=1}^{d} n_k + N^d$; and
- $p^{\mathrm{CP}}(\mathcal{X}) \leq N^{d-1} \sum_{k=1}^{d} n_k + N^{d-1}$.

The important observation is that tensors constructed by sampling polynomials on a grid are highly compressible. Specifically, the storage costs of these tensors, when stored in tensor-train, Tucker, or CP format, do not grow exponentially with the dimension d but linearly. In addition, in scenarios where the tensors are constructed via oversampling the polynomial, which means grid sizes n_j's are much larger than polynomial degrees N_j's, we can use much smaller degrees of freedom to represent this oversampled tensor. If one is familiar with tensor ranks, then one may notice that terms related to the N_j's are upper bounds of tensor-train, multilinear, and CP ranks. These turn out to be pretty tight bounds in practice.

There is generally not a prevalent format, in the sense that its storage cost is smaller than those of the other two. Depending on the grid sizes and polynomial degrees, all formats can have the smallest storage cost. Therefore, it is case specific to choose the optimal format for a given tensor.

3.1.2. Other special cases of algebraic structure. Similar to multivariate polynomials, it is easy to spot—after some experience—the mathematical tensor ranks of tensors constructed by sampling functions that have explicit algebraic structure since each variable in the function can be thought as a fiber of the tensor. The easiest ones to spot are those tensors derived from functions that are the sums of products of single-variable functions, such as

$$f(x,y,z) = 1 + \tan(x)y + y^2z^3, \quad g(x,y,z,w) = \cos(x)\sin(y) + e^{10z}e^{100w}.$$

If $\mathcal{F}$ and $\mathcal{G}$ are tensors constructed by sampling f and g on $n \times n \times n$ and $n \times n \times n \times n$ tensor-product grids, respectively, then the storage costs in different formats are bounded by

$$\begin{aligned} &p^{\mathrm{TT}}(\mathcal{F}) \leq 8n, \qquad p^{\mathrm{ML}}(\mathcal{F}) \leq 7n + 12, \qquad p^{\mathrm{CP}}(\mathcal{F}) \leq 9n + 3, \\ &p^{\mathrm{TT}}(\mathcal{G}) \leq 12n, \qquad p^{\mathrm{ML}}(\mathcal{G}) \leq 8n + 16, \qquad p^{\mathrm{CP}}(\mathcal{G}) \leq 8n + 2, \end{aligned}$$

where the tensor-train decompositions are performed in the order x, y, z, w. Other examples are functions that can be expressed with exponentials and powers, and similar examples have also been considered [31, 46].

Some special functions require reorganizations to reveal their algebraic structures. If the function is expressed with trigonometric functions, then the sampled tensor can often be low rank due to trigonometric identities. For example, consider the function $f(x,y,z) = \cos(x+y+z)$ that is a special case of the examples in [6, 49]. Since it can be written as

$$f(x,y,z) = (\cos(x)\cos(y) - \sin(x)\sin(y))\cos(z) - (\sin(x)\cos(y) + \cos(x)\sin(y))\sin(z),$$

any tensor $\mathcal{F}$ constructed by sampling f on an $n \times n \times n$ tensor-product grid satisfies

$$p^{\mathrm{TT}}(\mathcal{F}) \leq 8n, \quad p^{\mathrm{ML}}(\mathcal{F}) \leq 6n + 8, \quad p^{\mathrm{CP}}(\mathcal{F}) \leq 12n + 4.$$

In addition, [43] provides further insight on reorganizing special functions of the sum of variables to reveal its algebraic structure. These examples can often be combined to build more complicated functions that result in compressible tensors. This is an

ad hoc process and requires human ingenuity to express the sampled function in a revealing form. Again, tensors constructed by sampling such algebraically structured functions on a sufficiently large tensor-product grid can be represented using a small number of degrees of freedom.

3.2. Tensors derived by sampling smooth functions. Although most functions do not have the algebraic structure specified in subsection 3.1, tensors that are constructed by sampling smooth functions are often well approximated by compressible tensors. In light of Lemma 3.1, our idea for understanding the compressibility of a tensor derived from sampling a function is first to approximate that function by a multivariate polynomial, which is already a routine procedure for computing with low rank approximations to multivariate functions [22].

Without loss of generality, suppose that $\mathcal{X}$ is formed by sampling a smooth function f on a tensor-product grid in $[-1,1]^d$, i.e.,

$$\mathcal{X}_{i_1,\ldots,i_d} = f(x_{i_1}^{(1)},\ldots,x_{i_d}^{(d)}), \qquad 1 \le i_k \le n_k, \qquad 1 \le k \le d, \tag{3.2}$$

where $x^{(1)},\ldots,x^{(d)}$ are sets of $n_1,\ldots,n_d$ nodes in $[-1,1]$. Our idea is to find a multivariate polynomial p of degree $\le N_j - 1$ in the variable x_j that approximates f in $[-1,1]^d$ and then set $\mathcal{Y}_{i_1,\ldots,i_d} = p(x_{i_1}^{(1)},\ldots,x_{i_d}^{(d)})$. By Lemma 3.1, $\mathcal{Y}$ can be represented with a small number of degrees of freedom, and $\mathcal{Y}$ is an approximation to $\mathcal{X}$. In particular, we have

$$\|\mathcal{X}-\mathcal{Y}\|_F \le \left(\prod_{i=1}^d n_i\right)^{\frac{1}{2}} \|\mathcal{X}-\mathcal{Y}\|_{\max} \le \left(\prod_{i=1}^d n_i\right)^{\frac{1}{2}} \|f-p_N\|_\infty, \tag{3.3}$$

where $\|\cdot\|_\infty$ denotes the supremum norm on $[-1,1]^d$, and $\|\cdot\|_{\max}$ is the absolute maximum entry norm. Therefore, if p is a good approximation to f, then $\mathcal{Y}$ is a good approximation to $\mathcal{X}$ too. Although the error bound is good for small d, this approximation still suffers from the curse of dimensionality for large d.

One can now propose any linear or nonlinear approximation scheme to find a polynomial approximation p of f on $[-1,1]^d$. Clearly, excellent bounds on $\|\mathcal{X}-\mathcal{Y}\|_F$ are obtained by finding a p so that

$$\|f-p\|_\infty \approx \inf_{q\in\mathcal{P}_{N_1,\ldots,N_d}} \|f-q\|_\infty,$$

where $\mathcal{P}_{N_1,\ldots,N_d}$ is the space of d-variate polynomials of maximal degree $\le N_i - 1$ in x_i for $1 \le i \le d$. This best multivariable polynomial problem is often, but not always, tricky to solve directly. In those cases, near-optimal polynomial approximations are used instead. One common choice is to use p as the multivariate Chebyshev projection of f. That is,

$$p^{\text{cheb}}_{N_1,\ldots,N_d}(x_1,\ldots,x_d) = {\sum_{i_1=0}^{N_1-1}}' \cdots {\sum_{i_d=0}^{N_d-1}}' c_{i_1,\ldots,i_d} T_{i_1}(x_1)\cdots T_{i_d}(x_d),$$

$$c_{i_1,\ldots,i_d} = \left(\frac{2}{\pi}\right)^d \int_{-1}^1 \cdots \int_{-1}^1 \frac{f(x_1,\ldots,x_d)T_{i_1}(x_1)\cdots T_{i_d}(x_d)}{\sqrt{1-x_1^2}\cdots\sqrt{1-x_d^2}} dx_1\cdots dx_d,$$

where the primes indicate that the first term in the sum is halved, and $T_k(x)$ is the Chebyshev polynomial of degree k. Importantly, $p^{\text{cheb}}_{N_1,\ldots,N_d}$ is a near-best polynomial approximation to f [57], and the error $\|f - p^{\text{cheb}}_{N_1,\ldots,N_d}\|_\infty$ can be bounded. Thus, this choice of p leads to bounds on the compressibility of $\mathcal{X}$.

3.3. Worked-out examples. Here, we give two examples that illustrate how to understand the compressibility of tensors constructed by sampling smooth functions. We consider two functions: (1) a Fourier-like function, where we use best polynomial approximation, and (2) a sum of Gaussian bumps, where we use Chebyshev approximation.

3.3.1. Fourier-like function. Consider a tensor $\mathcal{X} \in \mathbb{C}^{n\times n\times n}$ constructed by sampling the following Fourier-like function on a tensor-product grid [55]:

$$f(x,y,z) = e^{iM\pi xyz}, \qquad x,y,z \in [-1,1],$$

where $M \geq 1$ is a real parameter. While representing $\mathcal{X}$ exactly requires n^3 degrees of freedom, it can be approximated by tensors that require fewer degrees of freedom (in the tensor-train and Tucker formats). To see this, let p_{k-1}^{best} and q_{k-1}^{best} be the best minimax polynomial approximations of degree $\leq k-1$ to $\cos(M\pi t)$ and $\sin(M\pi t)$ on $[-1,1]$, respectively, and define $h_{k-1} = p_{k-1}^{\text{best}} + iq_{k-1}^{\text{best}}$. Note that $h_{k-1}(xyz)$ has maximal degree at most $k-1$ so that

$$\begin{aligned}\inf_{w_{k-1}\in\mathcal{P}_{k-1}} \sup_{x,y,z\in[-1,1]} \left|e^{iM\pi xyz} - w_{k-1}(x,y,z)\right| &\leq \sup_{x,y,z\in[-1,1]} \left|e^{iM\pi xyz} - h_{k-1}(xyz)\right| \\ &= \sup_{t\in[-1,1]} \left|e^{iM\pi t} - h_{k-1}(t)\right|,\end{aligned}$$

where $\mathcal{P}_{k-1}$ is the space of trivariate polynomials of maximal degree $\leq k-1$, and the equality follows since $t = xyz \in [-1,1]$ if $x,y,z \in [-1,1]$. Furthermore, we have $e^{iM\pi t} = \cos(M\pi t) + i\sin(M\pi t)$, and so

$$\sup_{t\in[-1,1]} \left|e^{iM\pi t} - h_{k-1}(t)\right| \leq \sup_{t\in[-1,1]} \left|\cos(M\pi t) - p_{k-1}^{\text{best}}(t)\right| + \sup_{t\in[-1,1]} \left|\sin(M\pi t) - q_{k-1}^{\text{best}}(t)\right|.$$

By the equioscillation theorem [50, Thm. 7.4], $p_{k-1}^{\text{best}} = 0$ for $k-1 \leq 2\lfloor M\rfloor - 1$ since $\cos(M\pi t)$ equioscillates $2\lfloor M\rfloor + 1$ times in $[-1,1]$. Similarly, $\sin(M\pi t)$ equioscillates $2\lfloor M\rfloor$ times in $[-1,1]$, and hence $q_{k-1}^{\text{best}} = 0$ for $k-1 \leq 2\lfloor M\rfloor - 2$. However, for $k > 2\lfloor M\rfloor$, $\sup_{t\in[-1,1]}\left|e^{iM\pi t} - h_{k-1}(t)\right|$ decays supergeometrically to zero as $k\to\infty$. This also indicates that the error between the tensors sampled from $e^{iM\pi xyz}$ and $h_{k-1}(x,y,z)$ rapidly goes to 0 as $k\to\infty$. Hence, the numerical maximal degree, N_ϵ, of $e^{iM\pi xyz}$ satisfies $N_\epsilon/2M \to c$ for some constant $c \geq 1$ as $M\to\infty$. Lemma 3.1 shows that an approximant to $\mathcal{X}$ only requires $\mathcal{O}(M)$ degrees of freedom. In particular, if s_1 is the second element of the tensor-train rank of an approximant tensor to the one sampled by $e^{iM\pi xyz}$, then $s_1/(2M) \to 1$ as $M\to\infty$.

Figure 3.1 (left) plots the ratio of the second element of the tensor-train rank, s_1, of a tensor sampled from the Fourier-like function and $2M$. We observe that $s_1/(2M) \to 1$ as $M\to\infty$.

3.3.2. A sum of Gaussian bumps. Consider a tensor $\mathcal{X} \in \mathbb{C}^{n\times n\times n}$ constructed by sampling a sum of M Gaussian bumps, centered at arbitrary locations $(x_1,y_1,z_1),\ldots,(x_M,y_M,z_M)$ in $[-1,1]^3$, i.e.,

$$f(x,y,z) = \sum_{j=1}^{M} e^{-\gamma((x-x_j)^2+(y-y_j)^2+(z-z_j)^2)}, \qquad \gamma > 0. \tag{3.4}$$

Each Gaussian bump is a separable function of three variables so, mathematically, the tensor ranks of $\mathcal{X}$ depend linearly on M. However, since the sum is a smooth function,

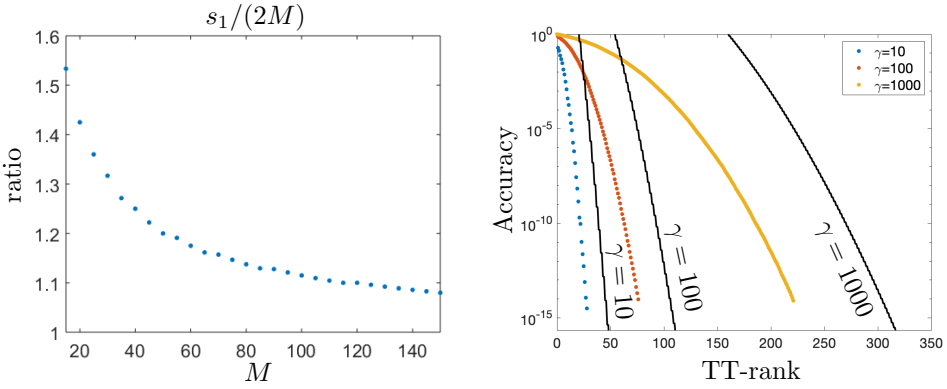

FIG. 3.1. *Left: The ratio of the second element of the tensor-train rank, s_1, of the tensors of size $n\times n\times n$ with $n = 600$ constructed by sampling the Fourier-like function e^{iMxyz} with $15 \le M \le 150$. The accuracy used to calculate the tensor-train ranks is 10^{-10}. Right: The second element, s_1, of the tensor-train rank (blue, red, and yellow dots) calculated with the TT-SVD and the theoretical bounds (black lines) of $\mathcal{X} \in \mathbb{C}^{400\times 400\times 400}$ constructed by sampling $\sum_{j=1}^{300} e^{-\gamma((x-x_j)^2+(y-y^j)^2+(z-z_j)^2)}$ on an equispaced tensor-product grid for $\gamma = 10, 100, 1000$, where (x_j, y_j, z_j) are arbitrary centers in $[-1, 1]^3$.*

the ranks are related to the polynomial degree required to approximate $f(x, y, z)$ in $[-1, 1]^3$ to an accuracy of $0 < \epsilon < 1$. Hence, the tensor ranks of X depend on γ and have very mild growth in M in the storage costs.

Exponential sums have been used to approximate f in [8, Chap. 6], but here we instead approximate it with a Chebyshev series. Due to the symmetry in x, y, and z as well as separability of each term in (3.4), we find that the Chebyshev approximation to $f(x, y, z)$ can be bounded by

$$\sup_{x,y,z\in[-1,1]} \left| f(x,y,z) - \sum_{j=1}^{M} p_\ell^j(x) q_\ell^j(y) r_\ell^j(z) \right| \le 3M \sup_{x\in[-1,1]} \left| e^{-\gamma x^2} - h_\ell(x) \right|,$$

where p_ℓ^j, q_ℓ^j, r_ℓ^j, and h_ℓ are Chebyshev approximations of degree $\le \ell$ to $e^{-\gamma(x-x_j)^2}$, $e^{-\gamma(y-y_j)^2}$, $e^{-\gamma(z-z_j)^2}$, and $e^{-\gamma x^2}$, respectively. An explicit Chebyshev expansion for $e^{-\gamma x^2}$ is known and given by [41, p. 32]

$$e^{-\gamma x^2} = {\sum_{j=0}^{\infty}}' (-1)^j e^{-\gamma/2} I_j(\gamma/2) T_{2j}(x),$$

where the prime on the summation indicates that the first term is halved, and $I_j(z)$ is the modified Bessel function of the first kind with parameter j [45, (10.25.2)]. This means that one can show that [14, Lem. 5][5]

$$h_\ell(x) = {\sum_{j=0}^{\ell}}' (-1)^j e^{-\gamma/2} I_j(\gamma/2) T_{2j}(x), \quad \sup_{x\in[-1,1]} \left| e^{-\gamma x^2} - h_\ell(x) \right| \le 2e^{-\gamma/4} I_{\lfloor \ell/2 \rfloor + 1}(\gamma/4).$$

By Lemma 3.1 and (3.3), we can understand the compressibility of $\mathcal{X}$. In particular, we can find an approximant tensor whose tensor-train ranks are bounded by the smallest integer ℓ such that $6Mn^{3/2}e^{-\gamma/4} I_{\lfloor \ell/2 \rfloor + 1}(\gamma/4) \le \epsilon$. We find it straightforward

[5] Unfortunately, there is a typo in [14, Lem. 5], and $I_{\ell+1}(\gamma/4)$ should be replaced by $I_{\lfloor \ell/2 \rfloor + 1}(\gamma/4)$.

to visualize compressibility via elements of the tensor ranks and their bounds, due to the way storage costs are calculated. Figure 3.1 (right) shows the second element of the tensor-train rank, s_1, of the approximant tensor, along with the bound that we derived. The bounds are relatively tight when ϵ is small.

4. Tensors with displacement structure. We say that $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ has an $(A^{(1)}, \ldots, A^{(d)})$-displacement structure of $\mathcal{G} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ if $\mathcal{X}$ satisfies the multi-dimensional Sylvester equation

$$\mathcal{X} \times_1 A^{(1)} + \cdots + \mathcal{X} \times_d A^{(d)} = \mathcal{G}, \qquad A^{(k)} \in \mathbb{C}^{n_k \times n_k}, \tag{4.1}$$

where "$\times_k$" is the k-mode matrix product of a tensor. In this section, we show that when $A^{(1)}, \ldots, A^{(d)}$ are normal matrices with "separated" spectra, and $\mathcal{G}$ is a low rank tensor, then $\mathcal{X}$ is compressible. Several classes of structured tensors (e.g., the Hilbert tensor) and the solution tensors of certain discretized PDEs (e.g., the discretized solution to Poisson's equation) have a displacement structure, which leads to an understanding of their compressibility.

4.1. Zolotarev numbers. The bounds that we derive on compressibility of tensors involve so-called Zolotarev numbers [1, 15, 59]. A Zolotarev number is a positive number between 0 and 1 defined via an infimum problem involving rational functions [59]. Namely,

$$Z_k(E, F) := \inf_{r \in \mathcal{R}_{k,k}} \frac{\sup_{z \in E} |r(z)|}{\inf_{z \in F} |r(z)|}, \qquad k \geq 0, \tag{4.2}$$

where E and F are disjoint complex sets, and $\mathcal{R}_{k,k}$ is the set of irreducible rational functions of the form $p(x)/q(x)$ with polynomials p and q of degree at most k. If E and F are well separated, then one finds that $Z_k(E, F)$ decays rapidly with k. This is because one can construct a low degree rational function that is small on E and large on F. If E and F are close to each other, then typically $Z_k(E, F)$ decreases much more slowly with k.

Zolotarev numbers can be used to bound the singular values of matrices with displacement structure [4, Thm. 2.1]. In particular, if $X \in \mathbb{C}^{m \times n}$ with $m \geq n$ satisfies the displacement structure

$$AX - XB = MN^*, \qquad M \in \mathbb{C}^{m \times \nu}, \quad N \in \mathbb{C}^{n \times \nu}, \tag{4.3}$$

where $A \in \mathbb{C}^{m \times m}$ and $B \in \mathbb{C}^{n \times n}$ are normal matrices with spectra $\Lambda(A) \subseteq E$ and $\Lambda(B) \subseteq F$, then the singular values of X satisfy [4, Thm. 2.1]

$$\sigma_{j+\nu k}(X) \leq Z_k(E, F)\sigma_j(X), \quad 1 \leq j + \nu k \leq n. \tag{4.4}$$

Roughly speaking, if $\Lambda(A)$ and $\Lambda(B)$ are well separated and ν is small, then the singular values $\sigma_j(X)$ decrease rapidly to 0.

When working with tensors, we translate the inequalities in (4.4) into Frobenius norm error bounds so that matrix results can then be utilized.

LEMMA 4.1. *If $X \in \mathbb{C}^{m \times n}$ is a matrix satisfying* (4.3)*, and $X_{\nu k}$ is the best rank νk approximation to X, then*

$$\|X - X_{\nu k}\|_F \leq Z_k(E, F)\|X\|_F,$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm.

Proof. To simplify notation let $Z_k = Z_k(E,F)$, $r = \nu k$, $\sigma_j = \sigma_j(X)$ for $1 \le j \le n$, and $\sigma_j = 0$ for $j > n$. If $k = 0$, then $r = 0$ and $Z_k = 1$, so $X_r = 0$, and the statement follows automatically. Now consider $k > 0$; note that for any $s \ge 1$ we have

$$\sum_{j=sr+1}^{(s+1)r} \sigma_j^2 = \sum_{j=1}^{r} \sigma_{j+sr}^2 \le Z_k^2 \sum_{j=1}^{r} \sigma_{j+(s-1)r}^2 \le \cdots \le Z_k^{2s} \sum_{j=1}^{r} \sigma_j^2,$$

where the inequalities come from the repeated application of the bound in (4.4). Therefore, we can bound $\|X - X_r\|_F^2$ by partitioning the singular values into groups of r. That is,

$$\|X - X_r\|_F^2 = \sum_{j=r+1}^{n} \sigma_i^2 \le \sum_{s=1}^{\infty} \sum_{j=sr+1}^{(s+1)r} \sigma_i^2 \le \sum_{s=1}^{\infty} Z_k^{2s} \sum_{j=1}^{r} \sigma_j^2 = \frac{Z_k^2}{1 - Z_k^2} \sum_{j=1}^{r} \sigma_j^2,$$

where the last equality is obtained by summing up the geometric series. Since $\|X\|_F^2 = \sum_{j=1}^{n} \sigma_j^2$, we find that

$$\left(1 + \frac{Z_k^2}{1 - Z_k^2}\right) \|X - X_r\|_F^2 \le \frac{Z_k^2}{1 - Z_k^2} \|X\|_F^2.$$

The result follows by rearranging. ∎

For $0 < \epsilon < 1$, the numerical rank of X measured in the Frobenius norm is the smallest integer, r_ϵ, such that

$$\inf_{\operatorname{rank}(\tilde{X}) \le r_\epsilon} \|X - \tilde{X}\|_F \le \epsilon \|X\|_F .$$

We denote this integer by $\operatorname{rank}_\epsilon(X)$. From Lemma 4.1, we find that for matrices that satisfy (4.3), we have

$$\operatorname{rank}_\epsilon(X) \le \nu k, \tag{4.5}$$

where k is the smallest integer so that $Z_k(E,F) \le \epsilon$. Therefore, Zolotarev numbers are very useful when trying to bound the numerical rank of matrices with displacement structure. For example, for an $n \times n$ Pick matrix P_n constructed with real numbers from an interval $[a,b]$ with $0 < a < b < \infty$, one can find that $\operatorname{rank}_\epsilon(P_n) \le 2\lceil \log(4b/a)\log(4/\epsilon)/\pi^2 \rceil$ [4].

4.2. The compressibility of tensors with displacement structure in the tensor-train format. Zolotarev numbers can also be used to understand the compressibility of tensors satisfying (4.1). From the bounds in (2.2), one finds that the numerical ranks of each unfolding provides an upper bound on all entries of the tensor-train ranks of approximant tensors. More precisely, if $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ is a tensor and $0 < \epsilon < 1$, then there exists a tensor $\tilde{\mathcal{X}}$ such that [47, Thm. 2.2]

$$||\mathcal{X} - \tilde{\mathcal{X}}||_F \le \epsilon ||\mathcal{X}||_F, \quad \operatorname{rank}^{\mathrm{TT}}(\tilde{\mathcal{X}}) = (1, \operatorname{rank}_\delta(X_1), \ldots, \operatorname{rank}_\delta(X_{d-1}), 1), \tag{4.6}$$

where $\delta = \epsilon/\sqrt{d-1}$, and X_k is the kth unfolding of $\mathcal{X}$. In order to easily relate tensor-train ranks with multilinear ranks in the next subsection, we choose to use $\delta = \epsilon/\sqrt{d}$.

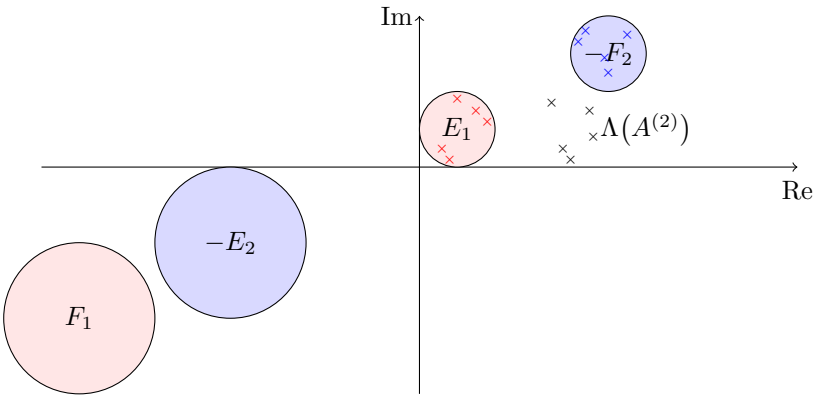

FIG. 4.1. *Minkowski sum separated matrices $A^{(1)}, A^{(2)}$, and $A^{(3)}$, where the colored crosses denote the spectra of $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$, respectively. Here, $\Lambda(A^{(1)}) \subseteq E_1$, $\Lambda(A^{(3)}) \subseteq -F_2$, $\Lambda(A^{(1)}) + \Lambda(A^{(2)}) \subseteq E_2$, and $\Lambda(A^{(2)}) + \Lambda(A^{(3)}) \subseteq -F_1$. By definition, we must have that E_1 is disjoint from F_1 (red regions) and that E_2 is disjoint from F_2 (blue regions).*

If $\mathcal{X}$ satisfies (4.1), then by rearranging (4.1) one can show that each unfolding matrix, X_j, has a displacement structure. This is precisely $B_j X_j - X_j C_j^T = G_j$, where G_j is the jth unfolding of $\mathcal{G}$ and

$$B_j = I \otimes \cdots \otimes I \otimes A^{(1)} + \cdots + A^{(j)} \otimes I \otimes \cdots \otimes I,$$
$$C_j = -(I \otimes \cdots \otimes I \otimes A^{(j+1)} + \cdots + A^{(d)} \otimes I \otimes \cdots \otimes I).$$

From properties of the Kronecker product [52, Thm. 2.5], we know that B_j and C_j are normal matrices with $\Lambda(B_j) = \Lambda(A^{(1)}) + \cdots + \Lambda(A^{(j)}) \subseteq E_j$ and $\Lambda(C_j) = -(\Lambda(A^{(j+1)}) + \cdots + \Lambda(A^{(d)})) \subseteq F_j$.[6] From Lemma 4.1 we see that for any integer k_j such that $Z_{k_j}(E_j, F_j) \leq \delta$, we have

$$\operatorname{rank}_\delta(X_j) \leq k_j \nu_j, \qquad \nu_j = \operatorname{rank}(G_j), \qquad 1 \leq j \leq d-1.$$

Therefore, a necessary condition for bounding the numerical tensor-train ranks of $\mathcal{X}$ using this approach is that the spectra of $A^{(1)}, \ldots, A^{(d)}$ are *Minkowski sum separated*.

DEFINITION 4.2. *We say that normal matrices $A^{(1)}, \ldots, A^{(d)}$ are Minkowski sum separated if there are disjoint sets E_j and F_j so that*

$$\Lambda(A^{(1)}) + \cdots + \Lambda(A^{(j)}) \subseteq E_j, \quad -(\Lambda(A^{(j+1)}) + \cdots + \Lambda(A^{(d)})) \subseteq F_j, \quad 1 \leq j \leq d-1,$$

where the set additions are Minkowski sums, and $\Lambda(A^{(j)})$ denotes the spectrum of $A^{(j)}$.

Figure 4.1 illustrates three Minkowski sum separated matrices $A^{(1)}, A^{(2)}$, and $A^{(3)}$, along with possible choices for the sets E_j and F_j for $j = 1, 2$. We summarize our findings as a theorem.

THEOREM 4.3. *Suppose $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ satisfies (4.1), where $A^{(1)}, \ldots, A^{(d)}$ are Minkowski sum separated with disjoint sets E_j and F_j for $1 \leq j \leq d-1$. Then, for a*

[6] By $\Lambda(A) + \Lambda(B)$ we mean the Minkowski sum, formed by adding each element in $\Lambda(A)$ to each element in $\Lambda(B)$, i.e., $\Lambda(A) + \Lambda(B) = \{a + b \mid a \in \Lambda(A),\ b \in \Lambda(B)\}$.

fixed $0 < \epsilon < 1$*, we have*

$$p_\epsilon^{\mathrm{TT}}(\mathcal{X}) \le \sum_{j=1}^{d} (k_{d-1}\nu_{d-1})(k_d\nu_d)n_d, \quad \nu_j = \operatorname{rank}(G_j), \quad 1 \le j \le d-1,$$

where G_j *is the* j*th unfolding of* $\mathcal{G}$*, and* k_j *is an integer so that* $Z_{k_j}(E_j, F_j) \le \epsilon/\sqrt{d}$*.*

For special choices of E_j and F_j, explicit bounds on $Z_{k_j}(E_j, F_j)$ are known, and therefore the bounds in Theorem 4.3 are also explicit. Here we mention two special cases.

Intervals. If $\Lambda(A^{(j)}) \subseteq [a, b]$ for $0 < a < b < \infty$, then one can take $E_j = [ja, jb]$ and $F_j = [-(d-j)b, -(d-j)a]$ in Theorem 4.3. From [4, Cor. 4.2], we find that

$$Z_{k_j}(E_j, F_j) \le 4\left[\exp\left(\frac{\pi^2}{2\log(16\gamma_j)}\right)\right]^{-2k}, \quad \gamma_j = \frac{(da + j(b-a))(db - j(b-a))}{abd^2}.$$

In particular, the following bound holds:

$$p_\epsilon^{\mathrm{TT}}(\mathcal{X}) \le \sum_{j=1}^{d} (k_{d-1}\nu_{d-1})(k_d\nu_d)n_d, \qquad k_j = \left\lceil \frac{\log(16\gamma_j)\log(4\sqrt{d}/\epsilon)}{\pi^2} \right\rceil, \tag{4.7}$$

where $\nu_j = \operatorname{rank}(G_j)$.

Disks. If $\Lambda(A^{(j)}) \subseteq \{z \in \mathbb{C} : |z - z_0| \le \eta\}$ for $0 < \eta < z_0$ and $z_0, \eta \in \mathbb{R}$, then one finds that $\Lambda(A^{(1)}) + \cdots + \Lambda(A^{(j)}) \subseteq \{z \in \mathbb{C} : |z - jz_0| \le j\eta\}$ and $-(\Lambda(A^{(j+1)}) + \cdots + \Lambda(A^{(d)})) \subseteq \{z \in \mathbb{C} : |z + (d-j)z_0| \le (d-j)\eta\}$. From [53, p. 123], we find that

$$Z_{k_j}(E_j, F_j) = \rho_j^{-k_j}, \qquad \rho_j = \frac{2j(d-j)\eta^2}{d^2 z_0^2 - ((d-j)^2 + j^2)\eta^2 - \sqrt{\xi_j}},$$

where $\xi_j = \left(d^2 z_0^2 - ((d-j)^2 + j^2)\eta^2\right)^2 - 4j^2(d-j)^2\eta^4$. In particular,

$$p_\epsilon^{\mathrm{TT}}(\mathcal{X}) \le \sum_{j=1}^{d} (k_{d-1}\nu_{d-1})(k_d\nu_d)n_d, \qquad k_j = \left\lceil \log(\sqrt{d}/\epsilon)/\log(\rho_j) \right\rceil, \tag{4.8}$$

where $\nu_j = \operatorname{rank}(G_j)$.

In subsection 4.5, we use (4.7) to bound the numerical storage cost in tensor-train format of the Hilbert tensor and the solution tensor of a discretized Poisson equation.

4.3. The compressibility of tensors with displacement structure in the Tucker format. The matrix SVD can be used to calculate the numerical multilinear rank [10, 36]. Indeed, if $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ is a tensor and $0 < \epsilon < 1$, then there exists a tensor $\tilde{\mathcal{X}}$ such that [10]

$$||\mathcal{X} - \tilde{\mathcal{X}}||_F \le \epsilon ||\mathcal{X}||_F, \quad \operatorname{rank}^{\mathrm{ML}}(\tilde{\mathcal{X}}) = (\operatorname{rank}_\delta(X_{(1)}), \ldots, \operatorname{rank}_\delta(X_{(d)})),$$

where $\delta = \epsilon/\sqrt{d}$, and $X_{(j)}$ is the jth matricization of $\mathcal{X}$.

Since the first unfolding of $\mathcal{X}$ coincides with the first matricization of $\mathcal{X}$, the bound on the second element of the tensor-train rank is also a bound on the first element of the multilinear rank of $\mathcal{X}$. One can use a similar idea to bound all entries of the multilinear ranks by considering the various matricizations. However, one finds that the spectra of $A^{(1)}, \ldots, A^{(d)}$ need to be separated in a slightly different sense.

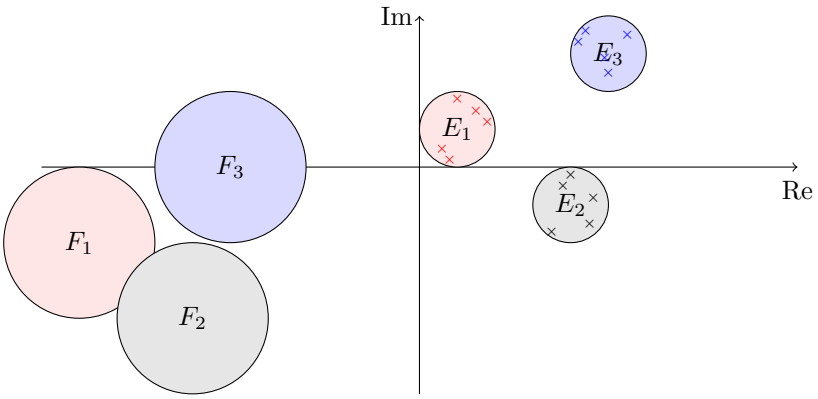

FIG. 4.2. *Minkowski singly separated matrices $A^{(1)}, A^{(2)}$, and $A^{(3)}$, where the colored crosses denote the spectrum of $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$, respectively. Here, $\Lambda(A^{(1)}) \subseteq E_1$, $\Lambda(A^{(2)}) \subseteq E_2$, $\Lambda(A^{(3)}) \subseteq E_3$, $-(\Lambda(A^{(1)})+\Lambda(A^{(2)})) \subseteq F_3$, $-(\Lambda(A^{(1)})+\Lambda(A^{(3)})) \subseteq F_2$, and $-(\Lambda(A^{(2)})+\Lambda(A^{(3)})) \subseteq F_1$. By definition, we have that E_1 is disjoint from F_1 (red regions), E_2 is disjoint from F_2 (gray regions), and E_3 is disjoint from F_3 (blue regions).*

DEFINITION 4.4. *We say that normal matrices $A_1, \ldots, A_d$ are Minkowski singly separated if there are disjoint sets E_j and F_j so that*

$$\Lambda(A_j) \subseteq E_j, \quad -\left(\sum_{k=1,k\neq j}^{d} \Lambda(A_k)\right) \subseteq F_j, \quad 1 \leq j \leq d,$$

where the set additions are Minkowski sums, and $\Lambda(A_j)$ denotes the spectrum of A_j.

Figure 4.2 illustrates the spectra of Minkowski singly separated matrices $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ along with their enclosed sets and Minkowski sums of the sets. Under this separation condition, we have the following theorem.

THEOREM 4.5. *Suppose $\mathcal{X} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ satisfies* (4.1)*, where $A^{(1)}, \ldots, A^{(d)}$ are Minkowski singly separated with disjoint sets E_j and F_j for $1 \leq j \leq d$. Then, for a fixed $0 < \epsilon < 1$, we have*

$$p_\epsilon^{\mathrm{ML}}(\mathcal{X}) \leq \sum_{j=1}^{d} n_j k_j \mu_j + \prod_{j=1}^{d} k_j \mu_j, \qquad \mathrm{rank}^{\mathrm{ML}}(\mathcal{G}) = (\mu_1, \ldots, \mu_d),$$

where k_j is an integer so that $Z_{k_j}(E_j, F_j) \leq \epsilon/\sqrt{d}$.

Proof. One can bound all the entries of the multilinear rank vector of $\mathcal{X}$ by the second entry of the tensor-train rank vector of the tensors $\mathcal{Y}^1, \ldots, \mathcal{Y}^d$ (see (1.5)). Due to the way $\mathcal{Y}^j$ is constructed, it can be shown that $\mathcal{Y}^j$ satisfies

$$\mathcal{Y}^j \times_1 A^{(j)} + \cdots + \mathcal{Y}^j \times_{d-j+1} A^{(d)} + \mathcal{Y}^j \times_{d-j+2} A^{(1)} + \cdots + \mathcal{Y}^j \times_d A^{(j-1)} = \mathcal{H}^j,$$

where $H^j_{(1)} = G_{(j)}, \ldots, H^j_{(d-j+1)} = G_{(d)}, H^j_{(d-j+2)} = G_{(1)}, \ldots, H^j_{(d)} = G_{(j-1)}$, and $\mathcal{H}^j$ is constructed from $\mathcal{G}$ in the same way that $\mathcal{Y}^j$ is constructed from $\mathcal{X}$. The result follows from Theorem 4.3 as the jth element of the multilinear rank of $\mathcal{X}$ is bounded above by the bound of the second entry of the tensor-train rank of $\mathcal{Y}^j$. ∎

As before, explicit bounds on the compressibility in Tucker format can be obtained from Theorem 4.5 by special choices of E_j and F_j, such as when they are intervals or disks.

Intervals. If $\Lambda(A^{(j)}) \subseteq [a,b]$ for $0 < a < b < \infty$, then one can take $E_j = [a,b]$ and $F_j = [-(d-1)b, -(d-1)a]$. Therefore, we find that [4, Cor. 4.2]

$$p_\epsilon^{\mathrm{ML}}(\mathcal{X}) \le k \sum_{j=1}^{d} n_j \mu_j + k^d \prod_{j=1}^{d} \mu_j, \qquad k = \left\lceil \frac{\log(16\gamma)\log(4\sqrt{d}/\epsilon)}{\pi^2} \right\rceil,$$

where $\gamma = (da + (b-a))(db - (b-a))/(abd^2)$ and $\mathrm{rank}^{\mathrm{ML}}(\mathcal{G}) = (\mu_1, \ldots, \mu_d)$.

Disks. If $\Lambda(A^{(j)}) \subseteq \{z \in \mathbb{C} \,:\, |z - z_0| \le \eta\}$ for $0 < \eta < z_0$ and $z_0, \eta \in \mathbb{R}$, then one can take $E_j = \{z \in \mathbb{C} \,:\, |z - z_0| \le \eta\}$ and $F_j = \{z \in \mathbb{C} \,:\, |z + (d-1)z_0| \le (d-1)\eta\}$. From [53, p. 123], we find that

$$p_\epsilon^{\mathrm{ML}}(\mathcal{X}) \le k \sum_{j=1}^{d} n_j \mu_j + k^d \prod_{j=1}^{d} \mu_j, \qquad k = \left\lceil \log(\sqrt{d}/\epsilon)/\log(\rho) \right\rceil,$$

where $\rho = (2(d-1)\eta^2)/(d^2 z_0^2 - ((d-1)^2+1)\eta^2 - \sqrt{\xi})$, $\xi = (d^2 z_0^2 - ((d-1)^2+1)\eta^2)^2 - 4(d-1)^2\eta^4$, and $\mathrm{rank}^{\mathrm{ML}}(\mathcal{G}) = (\mu_1, \ldots, \mu_d)$.

4.4. The compressibility of tensors with displacement structure in CP format. While deriving the bounds in this paper, we also considered including bounds on the compressibility of tensors with displacement structure in CP format. We were unable to come up with nontrivial bounds unless we introduced several additional and arbitrary assumptions in the statements of our theorem.

4.5. Worked-out examples. Here, we give two examples that illustrate how to use the displacement structure of a tensor to understand its compressibility. Since the bounds in tensor-train format and Tucker format are related through ranks, we only show results for the tensor-train format. As in the previous examples, we use the second element of the tensor-train rank and its bound to visualize the compressibility. We consider two tensors: (1) the 3D Hilbert tensor, and (2) the solution tensor of a Poisson equation.

4.5.1. The 3D Hilbert tensor. Consider the Hilbert tensor $\mathcal{H} \in \mathbb{C}^{n\times n\times n}$ defined by

$$\mathcal{H}_{ijk} = \frac{1}{i+j+k-2}, \qquad 1 \le i,j,k \le n.$$

This tensor is analogous to the notoriously ill-conditioned Hilbert matrix [11, 24]. It is easy to verify that the tensor possesses the following displacement structure:

$$\mathcal{H} \times_1 D + \mathcal{H} \times_2 D + \mathcal{H} \times_3 D = \mathcal{S},$$

where $\mathcal{S}$ is the tensor of all ones, and D is a diagonal matrix with $D_{ii} = i - \frac{2}{3}$. Thus, $\mathrm{rank}(\mathcal{S}) = 1$, and the ranks of the unfoldings of $\mathcal{S}$ are all 1.

Since the spectrum of D is contained in $[\frac{1}{3}, \frac{3n-2}{3}]$, (4.7) tells us that for any $0 < \epsilon < 1$ we have

$$p_\epsilon^{\mathrm{TT}}(\mathcal{H}) \le n(s_1^2 + 2s_1), \qquad s_1 = \left\lceil \frac{1}{\pi^2} \log\left(\frac{16n(2n-1)}{3n-2}\right) \log\left(\frac{4\sqrt{3}}{\epsilon}\right) \right\rceil. \tag{4.9}$$

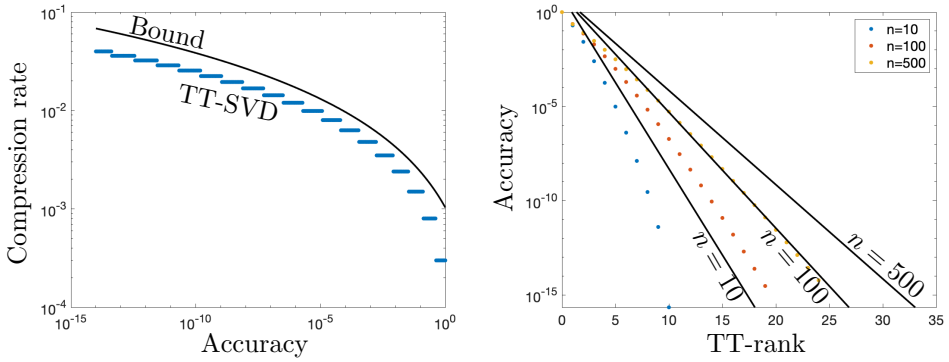

FIG. 4.3. *The compressibility of the 3D Hilbert tensor in tensor-train format. Left: The ratio of the storage costs for representing a* $100 \times 100 \times 100$ *Hilbert tensor in a tensor-train format calculated using the TT-SVD algorithm and* 100^3 *(blue dots), along with our theoretical bound on the compression rate (black line). Right: The second element of the tensor-train rank,* s_1*, (dots) and the theoretical bound (black lines) for* $n = 10, 100,$ *and* 500*.*

That is, $s_1 = \mathcal{O}(\log n \log(1/\epsilon))$ and means that the $n \times n \times n$ Hilbert tensor can be stored, up to an accuracy of ϵ in the Frobenius norm, in just $\mathcal{O}(n(\log n)^2(\log(1/\epsilon))^2)$ degrees of freedom. Figure 4.3 (left) shows the compressibility of $\mathcal{H}$ with $n = 100$ by computing the ratio of the storage costs using tensor-train format and explicit storage. Our theoretical results bound the savings well. Figure 4.3 (right) shows the compressibility of $\mathcal{H}$ by plotting s_1 and its bound in (4.9) for different values of n. The actual tensor-train ranks of $\mathcal{H}$ are computed with the TT-SVD algorithm [47].

4.5.2. Tensor solution of a discretized Poisson equation. Tensor decompositions can be incorporated into efficient solvers of PDEs [2, 7, 28, 32, 48, 54]. Displacement structure arises for the solution tensor when one discretizes a Laplace operator or any Laplace-like operator. Here, consider the 3D Poisson equation on $[-1, 1]^3$ with zero Dirichlet conditions, i.e.,

$$-(u_{xx} + u_{yy} + u_{zz}) = f \text{ on } \Omega = [-1, 1]^3, \qquad u|_{\partial\Omega} = 0. \tag{4.10}$$

If one writes down a second-order finite difference discretization of (4.10) on an $n \times n \times n$ equispaced grid, then one obtains the multidimensional Sylvester equation

$$\mathcal{X} \times_1 K + \mathcal{X} \times_2 K + \mathcal{X} \times_3 K = \mathcal{F}, \qquad K = -\frac{1}{h^2}\begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix},$$

where $h = 2/n$ and $\mathcal{F}_{ijk} = f(ih-1, jh-1, kh-1)$ for $1 \le i, j, k \le n-1$. The solution tensor $\mathcal{X}$ is unknown, and for large n one assumes that $\mathcal{X}_{ijk} \approx u(ih-1, jh-1, kh-1)$ for $1 \le i, j, k \le n-1$ is a reasonably good approximation. The eigenvalues of K are given by $4/h^2 \sin^2(\pi k/(2n))$ for $1 \le k \le n$ with $h = 2/n$ [39, (2.23)]. Since $(2/\pi)x \le \sin x \le 1$ for $x \in [0, \pi/2]$ and $h = 2/n$, the eigenvalues of K are contained in the interval $[1, n^2]$.

We are interested in understanding the compressibility of $\mathcal{X}$ in tensor-train format

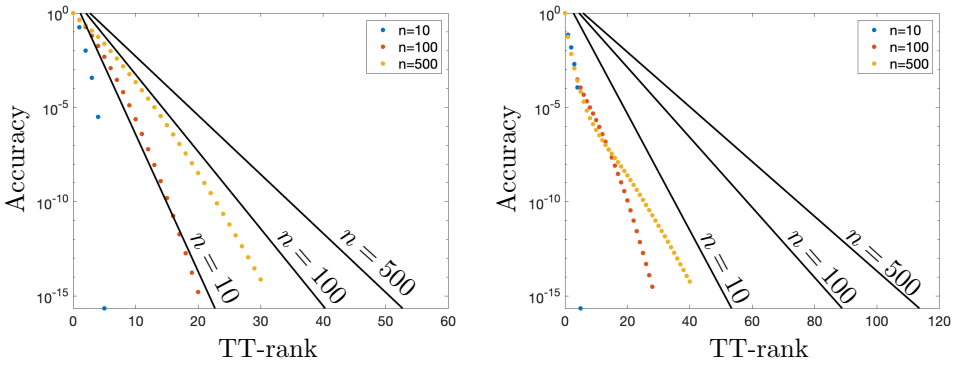

FIG. 4.4. *Left: The second element of the tensor-train rank, s_1 (blue, red, and yellow dots) of the finite difference solution to $-(u_{xx}+u_{yy}+u_{zz})=1$ on $[-1,1]^3$ with zero Dirichlet conditions, and the theoretical bound in* (4.11) *(black lines). Right: The second element of the tensor-train rank, s_1 (blue, red, and yellow dots) of the ultraspherical spectral solution to $-(u_{xx}+u_{yy}+u_{zz})=1$ on $[-1,1]^3$ with zero Dirichlet conditions, and the theoretical bound in* (4.13) *(black lines).*

when $f=1$. Since $\Lambda(K)\subseteq[1,n^2]$ and $\operatorname{rank}^{\mathrm{TT}}(\mathcal{F})=(1,1,1,1)$, (4.7) gives

$$p_\epsilon^{\mathrm{TT}}(\mathcal{X})\le n(s_1^2+2s_1),\quad s_1=\left\lceil\frac{1}{\pi^2}\log\left(\frac{16(n^2+2)(2n^2+1)}{9n^2}\right)\log\left(\frac{4\sqrt{3}}{\epsilon}\right)\right\rceil. \tag{4.11}$$

Figure 4.4 (left) shows the second element of the tensor-train rank, s_1, and the bound of the approximate solution tensor to the Poisson equation via finite difference discretization.

One wonders if there is also a fast Poisson solver for spectral discretizations. This turns out to be feasible with a carefully constructed ultraspherical spectral discretization. The Poisson equation can be discretized to a tensor equation as in [13],

$$\mathcal{X}\times_1 A^{-1}+\mathcal{X}\times_2 A^{-1}+\mathcal{X}\times_3 A^{-1}=\mathcal{G}, \tag{4.12}$$

where

$$u(x,y,z)=(1-x^2)(1-y^2)(1-z^2)\sum_{p=0}^{n}\sum_{q=0}^{n}\sum_{r=0}^{n}\mathcal{X}_{pqr}\tilde{C}_p^{(3/2)}(x)\tilde{C}_q^{(3/2)}(y)\tilde{C}_r^{(3/2)}(z),$$

$$f(x,y,z)=\sum_{p=0}^{n}\sum_{q=0}^{n}\sum_{r=0}^{n}\mathcal{F}_{pqr}\tilde{C}_p^{(3/2)}(x)\tilde{C}_q^{(3/2)}(y)\tilde{C}_r^{(3/2)}(z),$$

$\tilde{C}_k^{(3/2)}$ is the degree k orthonormalized ultraspherical polynomial with parameter $\frac{3}{2}$ [45, Table 18.3.1], $\mathcal{G}=\mathcal{F}\times_1 M^{-1}\times_2 M^{-1}\times_3 M^{-1}$, $A=D^{-1}M$, D is a diagonal matrix, both M and A are symmetric pentadiagonal matrices, and the spectrum of A satisfies $\Lambda(A)\in[-1,\ -1/(30n^4)]$. If $f=1$, (4.7) gives

$$p_\epsilon^{\mathrm{TT}}(\mathcal{X})\le n(s_1^2+2s_1),\ s_1=\left\lceil\frac{1}{\pi^2}\log\left(\frac{16(30n^4+2)(60n^4+1)}{270n^4}\right)\log\left(\frac{4\sqrt{3}}{\epsilon}\right)\right\rceil. \tag{4.13}$$

Figure 4.4 (right) shows the second element of the tensor-train rank, s_1, and the bound of the approximate solution tensor to the Poisson equation via ultraspherical

spectral discretization. This spectral discretization indicates that the $n \times n \times n$ tensor discretization of the solution can be approximated with only $\mathcal{O}(dn(\log n)^2(\log(1/\epsilon))^2)$ degrees of freedom. This is a significant reduction in the cost of storing the solution, with a relatively straightforward decomposition. Comparatively, one can achieve $\mathcal{O}(d \log n \log(1/\epsilon))$ with quantics tensor formats [27, 31], but their structures are more complicated and not as simple to use as the tensor-train format.

Some special functions can be well approximated by exponential sums of the form

$$S_k(x) = \sum_{j=1}^{k} \alpha_j e^{-t_j x}, \qquad \alpha_j, t_j \in \mathbb{R},$$

and these approximants can be used to represent the solution to PDEs with Laplace-like operators [16, 29]. In [29], the author uses exponential sums to show that the solution tensor to several 3D elliptic PDEs can be approximated with $\mathcal{O}(dn(\log n)^2(\log(1/\epsilon))^2)$ degrees of freedom. In this scenario, the Laplacian inverse operator can be approximated with a low CP rank tensor. In general, both the exponential sum approximation and Zolotarev numbers can be used to bound the kth singular value of matrices with displacement structure and capture the geometric decay, but the Zolotarev bound tends to be cleaner and does not involve an algebraic factor related to k [56, (34),(35)].

4.6. Solving for tensors in compressed formats. Since the proofs of Theorems 4.3 and 4.5 are constructive, we can use their implicit algorithms to solve 3D tensor Sylvester equations of the form

$$\mathcal{X} \times_1 A^{(1)} + \mathcal{X} \times_2 A^{(2)} + \mathcal{X} \times_3 A^{(3)} = \mathcal{F}, \tag{4.14}$$

where $A^{(1)} \in \mathbb{C}^{n_1 \times n_1}$, $A^{(2)} \in \mathbb{C}^{n_2 \times n_2}$, $A^{(3)} \in \mathbb{C}^{n_3 \times n_3}$, and $\mathcal{F} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$. We can compute approximate solutions to (4.14) efficiently in tensor-train or Tucker format when $\mathcal{F}$ is a low rank tensor and the spectra of $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ are well separated. In particular, if $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ are Minkowski sum separated and the unfoldings F_1 and F_2 of $\mathcal{F}$ have low rank decompositions $F_1 = W_1 Z_1^*$ and $F_2 = W_2 Z_2^*$ with rank r_1 and r_2, respectively, then we can solve for $\mathcal{X}$ in tensor-train format.

The tensor-train factors of $\mathcal{X}$ obtained by the TT-SVD algorithm are orthogonal matrices for the column and row spaces of unfoldings of X. For example, the first tensor-train factor U_1 of $\mathcal{X}$ can be found as a matrix with orthonormal columns spanning the column space of the first unfolding X_1. Since $\mathcal{X}$ satisfies (4.14), we find that X_1 satisfies the Sylvester equation

$$A^{(1)} X_1 + X_1 (I \otimes A^{(2)} + A^{(3)} \otimes I)^T = W_1 Z_1^*. \tag{4.15}$$

We can use the factored alternating direction implicit (fADI) method to solve (4.15) for a matrix V_1 such that $X_1 = V_1 D_1 Y_1^*$ [5]. One can then use the QR decomposition of V_1, i.e., $V_1 = U_1 R_1$, to calculate the first tensor-train core U_1.

The second and third tensor-train factors can be computed by finding matrices with orthonormal columns for the column and row spaces associated to C_2, where $C_2 = \text{reshape}(R_1 D_1 Y_1^*, r_1 n_2, n_3)$. It can be shown that C_2 satisfies the Sylvester equation

$$(I \otimes (U_1^* A^{(1)} U_1) + A^{(2)} \otimes I) C_2 + C_2 (A^{(3)})^T = (I \otimes U_1^*) W_2 Z_2^*.$$

One can again use fADI to solve for a low rank decomposition of C_2, i.e., $C_2 = V_2 D_2 Y_2^*$. This low rank decomposition can be compressed by performing a QR factorization of V_2 and Y_2 and then doing an SVD to obtain $C_2 \approx U_2 \Sigma T_2^*$, where U_2 and T_2 are matrices with r_2 orthonormal columns and Σ is a diagonal matrix. In this way, the second tensor-train factor is $U_2 = \text{reshape}(U_2, [r_1, n_2, r_2])$ and the third factor $U_3 = \Sigma T_2^*$. Although the fADI method requires the solution of shifted linear systems with $I \otimes (U_1^* A^{(1)} U_1) + A^{(2)} \otimes I$, the Kronecker product structure allows one to reshape these linear systems into Sylvester equations, which can themselves be solved with the alternating direction implicit (ADI) method [5]. Specifically, to solve the linear system

$$(I \otimes (U_1^* A^{(1)} U_1) + A^{(2)} \otimes I - \alpha I) y = b,$$

where α is a constant denoting the shift, we can rewrite it as a matrix equation

$$\left(U_1^* A^{(1)} U_1 - \frac{\alpha}{2} I\right) Y + Y \left(A^{(2)} - \frac{\alpha}{2} I\right) = B, \tag{4.16}$$

where $Y = \text{reshape}(y, [r_1, n_2])$, and $B = \text{reshape}(b, [r_1, n_2])$. This Sylvester matrix equation is solvable as U_1 is orthogonal, and $A^{(1)}$ and $A^{(2)}$ have distinct spectra. We can use the ADI method to solve the matrix equation efficiently. This means that one can completely avoid solving a huge linear system. As a result, if $n_1 = n_2 = n_3 = n$, $0 < \epsilon < 1$ is desired accuracy, and solving shifted linear systems of $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ takes T complexity, then solving (4.16) takes $\mathcal{O}(T \log n \log(1/\epsilon))$, and thus the Sylvester matrix equation solver has a complexity of $\mathcal{O}((s_1 + s_2) T \log n \log(1/\epsilon) + s_2 T (\log n)^2 (\log(1/\epsilon))^2)$. When s_1, s_2, and T are small, we have an efficient solver. In summary, the ADI-based tensor Sylvester equation solver is described in Algorithm 4.1.

Algorithm 4.1 A 3D Sylvester equation (4.14) solver that returns the solution in tensor-train form.

1: Use fADI to solve for the column space Z_1 of X_1 that satisfies $A^{(1)} X_1 + X_1 (I \otimes A^{(2)} + A^{(3)} \otimes I)^T = F_1 = M_1 N_1^*$.
2: Perform a QR decomposition, $Z_1 = U_1 R_1$, and let $U_1 = U_1(:, 1 : s_1)$ if $R_1(s_1 + 1, s_1 + 1)$ is small enough.
3: Use fADI to solve for $C_2 = Z_2 D_2 Y_2^*$, where C_2 satisfies $(I \otimes (U_1^* A^{(1)} U_1) + A^{(2)} \otimes I) C_2 + C_2 (A^{(3)})^T = (I \otimes U_1^*) F_2 = (I \otimes U_1^*) M_2 N_2^*$.
4: Find a low rank decomposition of $C_2 \approx U_2 \Sigma T_2^*$ using Z_2, D_2, and Y_2, and denote the rank by s_2.
5: Let $U_2 = \text{reshape}(U_2, [s_1, n_2, s_2])$.
6: Let $U_3 = \Sigma T_2^*$.
7: The solution $\mathcal{X}$ is in the tensor-train form with cores U_1, U_2, and U_3.

Similarly, if all matricizations of $\mathcal{F}$ are low rank, and $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ are Minkowski singly separated, then we can obtain the solution in orthogonal Tucker format via the higher-order singular value decomposition (HOSVD) method [10]. Each factor matrix of $\mathcal{X}$ is a matrix with orthonormal columns that span the column space of the matricization of $\mathcal{X}$, which satisfies the Sylvester equation

$$A^{(j)} X_{(j)} + X_{(j)} (I \otimes A^{(i)} + A^{(k)} \otimes I)^T = F_{(j)},$$

where

$$i = \begin{cases} 1, & j = 3, \\ j + 1, & j = 1, 2, \end{cases} \qquad k = \begin{cases} 3, & j = 1, \\ j - 1, & j = 2, 3. \end{cases}$$

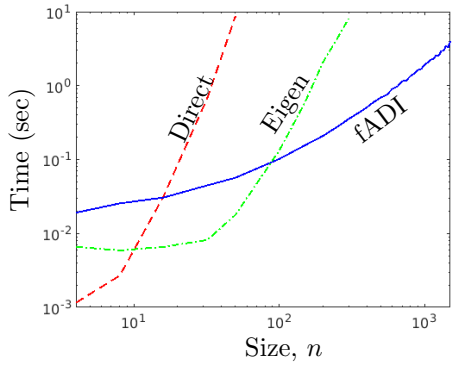

FIG. 4.5. *The execution time of a direct solver (dashed line), eigensolver (dash-dot line), and our fADI solver (solid line) of the spectrally discretized Poisson equation* $-(u_{xx} + u_{yy} + u_{zz}) = 1$ *on* $[-1, 1]^3$ *with zero Dirichlet conditions with discretization size* n *with* $4 \le n \le 1500$.

If solving shifted linear systems with $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ is fast, then we can use fADI to solve for the orthogonal column space of $X_{(j)}$ and use a direct method, such as a 3D Bartels–Stewart algorithm, to solve for the core tensor [3].

4.6.1. Poisson equation solver. Consider the example of Poisson equation in subsection 4.5.2 with ultraspherical discretization (4.12). Since A is a penta-diagonal matrix, we can solve shifted linear systems with A^{-1} in $\mathcal{O}(n)$ time using the Thomas algorithm. In addition, (4.13) indicates that s_1 and s_2 are of order $\mathcal{O}(\log n \log(1/\epsilon))$. Therefore, we can obtain a fast Poisson equation solver that computes the solution in tensor-train or orthogonal Tucker format. In tensor-train format, the complexity is $\mathcal{O}(n(\log n)^3(\log(1/\epsilon))^3)$, where $0 < \epsilon < 1$ is the accuracy.

Figure 4.5 shows the running time of different discretized Poisson solvers. The dashed line represents the direct solver that converts (4.13) into a huge linear system via the Kronecker product. The dash-dot line represents an eigendecomposition solver, which computes the eigendecomposition of A to diagonalize the equation and solves each element of $\mathcal{X}$ directly by scaling. The solid line represents our fADI-based tensor-train solver. We can see that as n gets large, our algorithm is the winner.[7]

Similarly, we can bound the solution of (4.12) in Tucker format with

$$\mathcal{O}((\log n)^3(\log(1/\epsilon))^3 + n(\log n)^2(\log(1/\epsilon))^2)$$

degrees of freedom and obtain the solution efficiently using fADI and ADI with the Tucker solver described in the previous section. If one is interested in the solution in CP format, then the exponential sum method in [29] (also see subsection 4.5.2) can solve (4.12) via low rank truncation.

Acknowledgments. We have had discussions with David Bindel, Dan Fortunato, Leslie Greengard, and Madeleine Udell about the results in this paper, and we appreciate their thoughts and comments. We are grateful to Nicolas Boulle and Heather Wilber for carefully reading an earlier draft of this manuscript.

[7]The fADI solver is implemented in C++, while the direct and the eigensolvers are implemented in MATLAB. However, both backslash linear system solver and eigendecomposition are carried out in LAPACK, so our comparison of the three solvers is still fair. All timings are performed in MATLAB R2019a on the super computer of Cornell's math department.

REFERENCES

[1] N. I. Akhiezer, *Elements of the Theory of Elliptic Functions*, translated from the second Russian edition by H. H. McFaden, Transl. Math. Monogr. 79, American Mathematical Society, 1990.

[2] J. Ballani and L. Grasedyck, *A projection method to solve linear systems in tensor format*, Numer. Linear Algebra Appl., 20 (2013), pp. 27–43.

[3] R. H. Bartels and G. W. Stewart, *Solution of the matrix equation $AX + XB = C$[F4]*, Comm. ACM, 15 (1972), pp. 820–826.

[4] B. Beckermann and A. Townsend, *On the singular values of matrices with displacement structure*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1227–1248, https://doi.org/10.1137/16M1096426.

[5] P. Benner, R.-C. Li, and N. Truhar, *On the ADI method for Sylvester equations*, J. Comput. Appl. Math., 233 (2009), pp. 1035–1045.

[6] G. Beylkin and M. J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 10246–10251.

[7] G. Beylkin and M. J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159, https://doi.org/10.1137/040604959.

[8] D. Braess, *Nonlinear Approximation Theory*, Springer Ser. Comput. Math. 7, Springer-Verlag, 1986.

[9] D. Braess and W. Hackbusch, *On the efficient computation of high-dimensional integrals and the approximation by exponential sums*, in Multiscale, Nonlinear and Adaptive Approximation, Springer, 2009, pp. 39–74.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278, https://doi.org/10.1137/S0895479896305696.

[11] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[12] J. Eshelby, *Energy relations and the energy-momentum tensor in continuum mechanics*, in Fundamental Contributions to the Continuum Theory of Evolving Phase Interfaces in Solids, Springer, 1999, pp. 82–119.

[13] D. Fortunato and A. Townsend, *Fast Poisson solvers for spectral methods*, IMA J. Numer. Anal., 40 (2020), pp. 1994–2018.

[14] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson, *GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration*, in Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc., 2018, pp. 7576–7586.

[15] A. A. Gončar, *Zolotarev problems connected with rational functions*, Sb. Math., 7 (1969), pp. 623–635.

[16] L. Grasedyck, *Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure*, Computing, 72 (2004), pp. 247–265.

[17] L. Grasedyck, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2029–2054, https://doi.org/10.1137/090764189.

[18] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitt., 36 (2013), pp. 53–78.

[19] V. S. Grigorascu and P. A. Regalia, *Tensor displacement structures and polyspectral matching*, in Fast Reliable Algorithms for Matrices with Structure, T. Kailath and A. H. Sayed, eds., SIAM, 1999, pp. 245–276, https://doi.org/10.1137/1.9781611971354.ch9.

[20] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, Springer Ser. Comput. Math. 42, Springer-Verlag, 2012.

[21] W. Hackbusch, B. N. Khoromskij, and E. E. Tyrtyshnikov, *Hierarchical Kronecker tensor-product approximations*, J. Numer. Math., 13 (2005), pp. 119–156.

[22] B. Hashemi and L. N. Trefethen, *Chebfun in three dimensions*, SIAM J. Sci. Comput., 39 (2017), pp. C341–C363, https://doi.org/10.1137/16M1083803.

[23] J. Håstad, *Tensor rank is NP-complete*, J. Algorithms, 11 (1990), pp. 644–654.

[24] D. Hilbert, *Ein Beitrag zur Theorie des Legendre'schen Polynoms*, Acta Math., 18 (1894), pp. 155–159.

[25] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, Stud. Appl. Math., 6 (1927), pp. 164–189.

[26] I. Ibragimov and S. Rjasanow, *Three way decomposition for the Boltzmann equation*, J. Comput. Math., 27 (2009), pp. 184–195.

[27] V. Kazeev and C. Schwab, *Quantized tensor-structured finite elements for second-order elliptic PDEs in two dimensions*, Numer. Math., 138 (2018), pp. 133–190.
[28] B. Khoromskij and S. Repin, *Rank structured approximation method for quasi-periodic elliptic problems*, Comput. Methods Appl. Math., 17 (2017), pp. 457–477.
[29] B. N. Khoromskij, *Tensor-structured preconditioners and approximate inverse of elliptic operators in* $\mathbb{R}^d$, Constr. Approx., 30 (2009), 599.
[30] B. N. Khoromskij, *Fast and accurate tensor approximation of a multivariate convolution with linear scaling in dimension*, J. Comput. Appl. Math., 234 (2010), pp. 3122–3139.
[31] B. N. Khoromskij, *O (dlog N)-quantics approximation of N-d tensors in high-dimensional numerical modeling*, Constr. Approx., 34 (2011), pp. 257–280.
[32] B. N. Khoromskij, *Tensor Numerical Methods in Scientific Computing*, Radon Ser. Comput. Appl. Math. 19, De Gruyter, 2018.
[33] B. N. Khoromskij, V. Khoromskaia, and H.-J. Flad, *Numerical solution of the Hartree–Fock equation in multilevel tensor-structured format*, SIAM J. Sci. Comput., 33 (2011), pp. 45–65, https://doi.org/10.1137/090777372.
[34] B. N. Khoromskij and C. Schwab, *Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs*, SIAM J. Sci. Comput., 33 (2011), pp. 364–385, https://doi.org/10.1137/100785715.
[35] T. G. Kolda, *Multilinear Operators for Higher-Order Decompositions*, Tech. report SAND2006-2081, Sandia National Laboratories, 2006.
[36] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500, https://doi.org/10.1137/07070111X.
[37] D. Kressner and C. Tobler, *Preconditioned low-rank methods for high-dimensional elliptic PDE eigenvalue problems*, Comput. Methods Appl. Math., 11 (2011), pp. 363–381.
[38] J. B. Kruskal, *Rank decomposition and uniqueness for* 3*-way and N-way arrays*, in Multiway Data Analysis, North-Holland, 1988, pp. 7–18.
[39] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, 2007, https://doi.org/10.1137/1.9780898717839.
[40] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, *A survey of multilinear subspace learning for tensor data*, Pattern Recognition, 44 (2011), pp. 1540–1551.
[41] Y. L. Luke, *The Special Functions and Their Approximations, Vol.* II, Mathematics in Science and Engineering 53, Academic Press, 1969.
[42] S. Massei, D. Palitta, and L. Robol, *Solving rank-structured Sylvester and Lyapunov equations*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1564–1590, https://doi.org/10.1137/17M1157155.
[43] M. J. Mohlenkamp and L. Monzón, *Trigonometric identities and sums of separable functions*, Math. Intell., 27 (2005), pp. 65–69.
[44] V. Olshevsky, I. Oseledets, and E. Tyrtyshnikov, *Superfast inversion of two-level Toeplitz matrices using Newton iteration and tensor-displacement structure*, in Recent Advances in Matrix and Operator Theory, Springer, 2007, pp. 229–240.
[45] F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, eds., *NIST Handbook of Mathematical Functions*, with 1 CD-ROM (Windows, Macintosh and UNIX), U.S. Department of Commerce, National Institute of Standards and Technology; Cambridge University Press, 2010.
[46] I. Oseledets, *Constructive representation of functions in low-rank tensor formats*, Constr. Approx., 37 (2013), pp. 1–18.
[47] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317, https://doi.org/10.1137/090752286.
[48] I. V. Oseledets and S. V. Dolgov, *Solution of linear systems and matrix inversion in the TT-format*, SIAM J. Sci. Comput., 34 (2012), pp. A2718–A2739, https://doi.org/10.1137/110833142.
[49] 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.
[50] M. J. D. Powell, *Approximation Theory and Methods*, Cambridge University Press, 1981.
[51] J. B. Reade, *Eigenvalues of positive definite kernels*, SIAM J. Math. Anal., 14 (1983), pp. 152–157, https://doi.org/10.1137/0514012.
[52] K. Schacke, *On the Kronecker Product*, Master's thesis, University of Waterloo, 2004.
[53] G. Starke, *Near-circularity for the rational Zolotarev problem in the complex plane*, J. Approx. Theory, 70 (1992), pp. 115–130.
[54] Y. Sun and M. Kumar, *Numerical solution of high dimensional stationary Fokker–Planck*

equations via tensor decomposition and Chebyshev spectral differentiation, Comput. Math. Appl., 67 (2014), pp. 1960–1977.
[55] A. Townsend, *Computing with Functions in Two Dimensions*, Ph.D. thesis, University of Oxford, 2014.
[56] A. Townsend and H. Wilber, *On the singular values of matrices with high displacement rank*, Linear Algebra Appl., 548 (2017), pp. 19–41.
[57] L. N. Trefethen, *Approximation Theory and Approximation Practice*, extended ed., SIAM, 2019, https://doi.org/10.1137/1.9781611975949.
[58] L. N. Trefethen, *Cubature, approximation, and isotropy in the hypercube*, SIAM Rev., 59 (2017), pp. 469–491, https://doi.org/10.1137/16M1066312.
[59] E. Zolotarev, *Application of elliptic functions to questions of functions deviating least and most from zero*, Zap. Imp. Akad. Nauk. St. Petersburg, 30 (1877), pp. 1–59.