This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2019-5666C

Functional Sparse Tucker Tensor for Scientific Data Compression

PRASHANT RAI, HEMANTH KOLLA, and LEWIS CANNADA, Sandia National Laboratories
ALEX GORODETSKY, University of Michigan

We propose a strategy to compress and store large volumes of scientific data represented on unstructured grids. Approaches utilizing tensor decompositions for data compression have already been proposed (e.g. TuckerMPI [2]). Here, data on a structured grid is stored as a tensor which is then subjected to appropriate decomposition in suitable tensor formats. Such decompositions are based on generalization of singular value decomposition like procedures to tensors and capture essential features in the data with storage cost lower by orders of magnitude. However, tensor based data compression is limited by the fact that one can only consider scientific data represented on structured grids. In case of data on unstructured meshes, we propose to consider data as realizations of a function that is based on functional view of the tensor thus avoiding such limitations. The key is to efficiently estimate the parameters of the function whose complexity is small compared to the cardinality of the dataset (otherwise there is no compression). Here, we introduce the set of functional sparse Tucker tensors and propose a method to construct approximation in this set such that the resulting compact functional tensor can be rapidly evaluated to recover the original data.

Additional Key Words and Phrases: data compression, functional tensors, tucker decomposition, sparse approximations

ACM Reference Format:
Prashant Rai, Hemanth Kolla, Lewis Cannada, and Alex Gorodetsky. 2019. Functional Sparse Tucker Tensor for Scientific Data Compression. In *Proceedings of KDD Workshop on Tensor Methods for Emerging Data Science Challenges (KDD).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Functional tensors are based on interpretation of high dimensional functions as tensors and their decomposition in several tensor formats as particular approximations. Consequently, functional tensors have been studied and applied for sampling based approximation of high dimensional functions in cases where the number of available function evaluations is small. Several functional tensor formats have been studied for various applications e.g. [8, 11, 14, 15]. These approaches rely on linearity between the parameters of the low-rank format and the output of the function. Utilizing this multilinear parameterization, they convert the low-rank function approximation to one of low-rank tensor decomposition for the coefficients of a tensor-product basis.

The novelty of the present paper, in contrast, is aimed at detecting low rank structure in the large volumes of data in order to obtain a low complexity functional tensor representation for a small loss in accuracy. As opposed to high dimensional function approximation using tensors in earlier works, high dimensionality does not come from the number of inputs to the function but from the number of data points required to be processed in order to obtain a functional tensor form. This functional tensor, stored as a surrogate at a fraction of cost of the original dataset, can be rapidly evaluated to recover accurate approximations of the data. We note that the compressed functional form can act as a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Association for Computing Machinery.
Manuscript submitted to ACM

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

preview of the full dataset, which may reside on long-term storage and need not replace the original dataset. In this paper, we consider the functional sparse Tucker representation of the data.

Several compression methods largely focus on compressing local structure with very little loss in precision. Examples of such methods include multivariate volume block data reduction by taking advantage of local multiway structure [13], compression of data in local blocks [9, 17]. Tensor based methods, in contrast, aim at detecting global structure in the data. It does not process the data in blocks but rather considers the data in its entirety. In this work, in order to take advantage of tensor based compression, we first interpolate the unstructured data on a structured grid followed by its Tucker decomposition [20]. Singular vectors with truncated rank for each mode thus obtained are represented as functions on a suitable basis using least squares with sparsity constraints thus resulting in a functional sparse Tucker representation of the dataset (see section 2 below).

The manuscript is organized as follows. We introduce and formalize the notion of functional sparse Tucker tensors in section 2. In order to construct approximations in this set, we review least squares with sparse regularization in section 3. We then present our construction algorithm in section 4 and illustrate it on simulation datasets in section 5 with a short conclusion in section 6.

2 FUNCTIONAL SPARSE TUCKER TENSORS

The key idea in this work is to represent the dataset as realizations of a multivariate function

$$u(y_1,\ldots,y_d) = \sum_{i_1=1}^{n_1}\cdots\sum_{i_d=1}^{n_d} \beta_{i_1,\ldots,i_d}\phi_{i_1}^{(1)}(y_1)\cdots\phi_{i_d}^{(d)}(y_d),$$

where $\phi_{i_k}^{(k)}, 1 \le k \le d$ are basis functions (e.g. polynomials, wavelets...). The number of expansion coefficients $\beta_{i_1,\ldots,i_d}$ are $\prod_{k=1}^{d} n_k$ thus manifesting the curse of dimensionality if n_k or d or both are large. In such cases, we instead represent the data as realizations of a Tucker low rank approximation $\tilde{u}$ of u where

$$\tilde{u}(y_1,\ldots,y_d) = \sum_{j_1=1}^{r_1}\cdots\sum_{j_d=1}^{r_d} \alpha_{j_1,\ldots,j_d} w_{j_1}^{(1)}(y_1)\cdots w_{j_d}^{(d)}(y_d). \tag{1}$$

Storage of $\tilde{u}$ in (1) require $\prod_{k=1}^{d} r_k$ coefficients and $\sum_{k=1}^{d} n_k r_k$ expansion coefficients of $w_{j_k}^{(k)}(y_k), 1 \le j_k \le r_k, 1 \le k \le d$ such that

$$w_{j_k}^{(k)}(y_k) = \sum_{i_k=1}^{n_k} w_{i_k,j_k}^{k}\phi_{i_k}^{(k)}(y_k). \tag{2}$$

In additional, in order to gain advantage from sparsity based regularization, we also constraint the number of non zero coefficients in (2). In the following, we formalize the notion of functional sparse Tucker tensors.

We introduce approximation spaces $\mathcal{S}_{n_k}^{k}$ with orthonormal basis $\{\phi_j^{(k)}\}_{j=1}^{n_k}$, such that

$$\mathcal{S}_{n_k}^{k} = \left\{ v^{(k)}(y_k) = \sum_{j=1}^{n_k} v_j^k \phi_j^{(k)}(y_k); v_j^k \in \mathbb{R} \right\} = \left\{ v^{(k)}(y_k) = \boldsymbol{\phi}^{(k)}(y_k)^T \mathbf{v}^{(k)}; \mathbf{v}^{(k)} \in \mathbb{R}^{n_k} \right\},$$

where $\mathbf{v}^{(k)}$ denotes the vector of coefficients of $v^{(k)}$ and where $\boldsymbol{\phi}^{(k)} = (\phi_1^{(k)}, \ldots, \phi_{n_k}^{(k)})^T$ denotes the vector of basis functions. An approximation space $\mathcal{S}_{\boldsymbol{n}}$ is then obtained by tensorization of approximation spaces $\mathcal{S}_{n_k}^k$:

$$\mathcal{S}_{\boldsymbol{n}} = \mathcal{S}_{n_1}^1 \otimes \ldots \otimes \mathcal{S}_{n_d}^d = \left\{ v = \sum_{i \in I_{\boldsymbol{n}}} v_i \phi_i \ ; \ v_i \in \mathbb{R} \right\},$$

where $I_{\boldsymbol{n}} = \times_{k=1}^d \{1 \ldots n_k\}$ and $\phi_i(y) = (\phi_{i_1}^{(1)} \otimes \ldots \otimes \phi_{i_r}^{(d)})(y_1, \ldots, y_d) = \phi_{i_1}^{(1)}(y_1) \ldots \phi_{i_r}^{(d)}(y_r)$. An element $v = \sum_i v_i \phi_i \in \mathcal{S}_{\boldsymbol{n}}$ can be identified with the algebraic tensor $\mathbf{v} \in \mathbb{R}^{n_1} \otimes \ldots \otimes \mathbb{R}^{n_d}$ such that $(\mathbf{v})_i = v_i$. Denoting $\boldsymbol{\phi}(y) = \boldsymbol{\phi}^{(1)}(y_1) \otimes \ldots \otimes \boldsymbol{\phi}^{(d)}(y_r) \in \mathbb{R}^{n_1} \otimes \ldots \otimes \mathbb{R}^{n_d}$, we have the identification $\mathcal{S}_{\boldsymbol{n}} \simeq \mathbb{R}^{n_1} \otimes \ldots \otimes \mathbb{R}^{n_d}$ with

$$\mathcal{S}_{\boldsymbol{n}} = \{ v(y) = \langle \boldsymbol{\phi}(y), \mathbf{v} \rangle ; \mathbf{v} \in \mathbb{R}^{n_1} \otimes \ldots \otimes \mathbb{R}^{n_d} \},$$

where $\langle \cdot, \cdot \rangle$ denotes the canonical inner product in $\mathbb{R}^{n_1} \otimes \ldots \otimes \mathbb{R}^{n_d}$.

Here, we suppose that the approximation space $\mathcal{S}_{\boldsymbol{n}}$ is sufficiently rich to allow accurate representations of a large class of functions (e.g. by choosing polynomial spaces with high degree, wavelets with high resolution...). We now introduce the set of functional sparse tensors.

Let $\mathcal{R}_1$ denote the set of (elementary) rank-one tensors in $\mathcal{S}_{\boldsymbol{n}} = \mathcal{S}_{n_1}^1 \otimes \ldots \otimes \mathcal{S}_{n_d}^d$, defined by

$$\mathcal{R}_1 = \left\{ w(y) = \left(\otimes_{k=1}^d w^{(k)} \right)(y) = \prod_{k=1}^d w^{(k)}(y_k) \ ; \ w^{(k)} \in \mathcal{S}_{n_k}^k \right\},$$

or equivalently by

$$\mathcal{R}_1 = \left\{ w(y) = \langle \boldsymbol{\phi}(y), \mathbf{w}^{(1)} \otimes \ldots \otimes \mathbf{w}^{(d)} \rangle ; \mathbf{w}^{(k)} \in \mathbb{R}^{n_k} \right\},$$

where $\boldsymbol{\phi}(y) = \boldsymbol{\phi}^{(1)}(y_1) \otimes \ldots \otimes \boldsymbol{\phi}^{(d)}(y_d)$, with $\boldsymbol{\phi}^{(k)} = (\phi_1^{(k)}, \ldots, \phi_{n_k}^{(k)})^T$ the vector of basis functions of $\mathcal{S}_{n_k}^k$, and where $\mathbf{w}^{(k)} = (w_1^k, \ldots, w_{n_k}^k)^T$ is the set of coefficients of $w^{(k)}$ in the basis of $\mathcal{S}_{n_k}^k$, that means $w^{(k)}(y_k) = \sum_{i=1}^{n_k} w_i^k \phi_i^{(k)}(y_k)$. Correspondingly, we define $\boldsymbol{m}$-sparse rank-one subset defined as

$$\mathcal{R}_1^{\boldsymbol{m}\text{-sparse}} = \left\{ w(y) = \langle \boldsymbol{\phi}(y), \mathbf{w}^{(1)} \otimes \ldots \otimes \mathbf{w}^{(d)} \rangle ; \mathbf{w}^{(k)} \in \mathbb{R}^{n_k}, \|\mathbf{w}^{(k)}\|_0 \leq m_k \right\}$$

with effective dimension $\sum_{k=1}^d m_k \ll \sum_{k=1}^d n_k$ (here we only count the values of the non-zero coefficients and not the integers indicating their locations). However performing least-squares approximation in this set may not be computationally tractable. We thus introduce a convex relaxation of the ℓ_0-"norm" to define the subset $\mathcal{R}_1^{\boldsymbol{\gamma}}$ of $\mathcal{R}_1$ defined as

$$\mathcal{R}_1^{\boldsymbol{\gamma}} = \left\{ w(y) = \langle \boldsymbol{\phi}(y), \mathbf{w}^{(1)} \otimes \ldots \otimes \mathbf{w}^{(d)} \rangle ; \mathbf{w}^{(k)} \in \mathbb{R}^{n_k}, \|\mathbf{w}^{(k)}\|_1 \leq \gamma_k \right\},$$

where the set of parameters $(\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(d)})$ is now searched in a convex subset of $\mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_d}$.
Finally, we introduce the set of functional Tucker tensors with multilinear Tucker rank $\mathbf{r} = (r_1, \ldots, r_d)$

$$\mathcal{T}_{\mathbf{r}} = \left\{ v = \sum_{j_1=1}^{r_1} \cdots \sum_{j_d=1}^{r_d} \alpha_{j_1,\ldots,j_d} w_{j_i,\ldots,j_d} \ ; w_{j_1,\ldots,j_d} \in \mathcal{R}_1 \right\}$$

and the corresponding sparse subset

$$\mathcal{T}_{\mathbf{r}}^{\boldsymbol{\gamma}} = \left\{ v = \sum_{j_1=1}^{r_1} \cdots \sum_{j_d=1}^{r_d} \alpha_{j_1,\ldots,j_d} w_{j_i,\ldots,j_d} \ ; w_{j_1,\ldots,j_d} \in \mathcal{R}_1^{\gamma} \right\}.$$

In the following, we propose algorithms for the construction of approximations in tensor subsets $\mathcal{T}_{\mathbf{r}}^{\boldsymbol{\gamma}}$ which requires sparse approximation of functions $w_{j_k}^k(y_k)$. For this purpose, we use least squares with sparse regularization as described in the next section.

3 LEAST SQUARES WITH SPARSE REGULARIZATION

A sparse function is one that can be represented using few non zero terms when expanded on a suitable basis. In general, a successful reconstruction of sparse solution vector depends on sufficient sparsity of the coefficient vector and additional properties (incoherence) depending on the samples and of the chosen basis (see [4, 10]). More precisely, an approximation $\sum_{i=1}^{P} u_i \phi_i(y)$ of a function $u(y)$ is considered as sparse on a particular basis $\{\phi_i(y)\}_{i=1}^{P}$ if it admits a good approximation with only a few non zero coefficients. Under certain conditions, a sparse approximation can be computed accurately using only $Q \ll P$ samples of $u(y)$ via sparse regularization. Given the random samples $\mathbf{z} \in \mathbb{R}^Q$ of the function $u(y)$ at sample points $\{y^q\}_{q=1}^{Q}$, a best m-sparse (or m-term) approximation of u can be ideally obtained by solving the constrained optimization problem

$$\min_{\mathbf{v} \in \mathbb{R}^P} \|\mathbf{z} - \mathbf{\Phi}\mathbf{v}\|_2^2 \quad \text{subject to} \quad \|\mathbf{v}\|_0 \leq m, \tag{3}$$

where $\|\mathbf{v}\|_0 = \#\{i \in \{1, \ldots, P\} \ : \ v_i \neq 0\}$ is the so called ℓ_0-"norm" of $\mathbf{v}$ which gives the number of non zero components of $\mathbf{v}$ and and $\mathbf{\Phi} \in \mathbb{R}^{Q \times P}$ the matrix with components $(\mathbf{\Phi})_{q,i} = \phi_i(y^q)$. Problem (3) is a combinatorial optimization problem which is NP hard to solve. Under certain assumptions, problem (3) can be reasonably well approximated by the following constrained optimization problem which introduces a convex relaxation of the ℓ_0-"norm":

$$\min_{\mathbf{v} \in \mathbb{R}^P} \|\mathbf{z} - \mathbf{\Phi}\mathbf{v}\|_2^2 \quad \text{subject to} \quad \|\mathbf{v}\|_1 \leq \delta, \tag{4}$$

where $\|\mathbf{v}\|_1 = \sum_{i=1}^{P} |v_i|$ is the ℓ_1-norm of $\mathbf{v}$. Since the ℓ_2 and ℓ_1-norms are convex, we can equivalently consider the following convex optimization problem, known as Lasso [19] or basis pursuit [7]:

$$\min_{\mathbf{v} \in \mathbb{R}^P} \|\mathbf{z} - \mathbf{\Phi}\mathbf{v}\|_2^2 + \lambda \|\mathbf{v}\|_1, \tag{5}$$

where $\lambda > 0$ corresponds to a Lagrange multiplier whose value is related to δ. Problem (5) appears as a regularized least-squares problem. The ℓ_1-norm is a sparsity inducing regularization function in the sense that the solution $\mathbf{v}$ of (5) may contain components which are exactly zero. Several optimization algorithms have been proposed for solving (5) (see [1]). In this paper, we use the Lasso modified least angle regression algorithm (see LARS presented in [12]) and fast leave-one-out cross validation error estimate [5] for optimal sparse solution (corresponding to regularization parameter λ) which relies on the use of the Sherman-Morrison-Woodbury formula (see [3] for its implementation within Lasso modified LARS algorithm). In this work, we have used Lasso modified LARS implementation of SPAMS software [18] for ℓ_1-regularization.

4 FUNCTIONAL SPARSE TUCKER USING TUCKERMPI

4.1 Interpolation on structured grid

Representation of the dataset in functional sparse Tucker format defined in section 2 requires estimation of the core tensor $\boldsymbol{\alpha}$ and univariate functions $w_j^{(k)}(y_k)$. If the dataset is available on a structured grid, it can be stored as a tensor

$\mathcal{U}$ which can then be decomposed in Tucker format

$$\mathcal{U} \approx \tilde{\mathcal{U}} = \boldsymbol{\alpha} \times_1 W^{(1)} \times_2 W^{(2)} \cdots \times_d W^{(d)},$$

where $\times_k$ is mode k product of $\mathcal{U}$ with a factor matrix $W^{(k)} \in \mathbb{R}^{n_k \times r_k}$. Here, the compression precision is given by

$$\epsilon = \frac{\|\mathcal{U} - \tilde{\mathcal{U}}\|_F}{\|\mathcal{U}\|_F},$$

where $\|\cdot\|_F$ is the Frobenius norm. Since the dataset considered is unstructured, we propose to interpolate the data on a structured grid. Let us denote the grid size in mode $k, 1 \leq k \leq d$ as I_k and, for the sake of simplicity, consider that the gird points are equispaced. A structured grid of size $I_1 \times I_2 \cdots \times I_d$ can thus be obtained. Now, we consider only a small subset of the original dataset for linear interpolation on this grid and the interpolated data is stored as a tensor which is then decomposed in Tucker format. We use TuckerMPI [2], a parallel C++/MPI software package for compressing distributed data, for this purpose. Note that TuckerMPI is a parallel implementation of the sequentially-truncated HOSVD (ST-HOSVD) [21]. We thus obtain factor matrices $W^{(k)}, 1 \leq k \leq d$, the columns of which are realizations of univariate functions $w_{j_k}^{(k)}(y_k), 1 \leq j \leq r_k$.

4.2 Sparse approximation of singular vectors

We now wish to obtain a functional representation $w_{j_k}^{(k)}(y_k), 1 \leq j_k \leq r_k$, of the singular vectors $W_{:,j_k}^{(k)}$ such that $W_{i_k,j_k}^{(k)}$ are evaluations of $w_{j_k}^{(k)}(y_k^{i_k})$ at grid locations $\{y_k^{i_k}\}_{i_k=1}^{I_k}$ along mode k. For this purpose, we use least squares with sparse regularization in section 3 to obtain coefficients on suitable basis functions. It is well known that singular vectors are more oscillatory (see for e.g. Figure 3(a)) for higher rank as they capture high frequency phenomenon in the dataset. Thus, choice of basis functions for representation of $w_{j_k}^{(k)}(y_k)$ corresponding to small j_k may not be appropriate for the ones with higher j_k. Therefore, in this work, we propose to construct approximation in two spaces $\mathbb{P}_p$, where $\mathbb{P}_p$ is the space of Legendre polynomials of degree p and $\mathbb{W}_{s,p}$, where $\mathbb{W}_{s,p}$ is the space of multi-resolution wavelets with resolution s and degree p. We can then choose the approximation that gives smaller approximation error. We present the overall compression scheme in Algorithm 1 below.

Algorithm 1 Compression of unstructured data in functional sparse Tucker format

Input: Original dataset, interpolation grid $I_k, 1 \leq k \leq d$, compression precision ϵ

Output: Function sparse Tucker tensor core $\boldsymbol{\alpha}$ and coefficients of $w_{j_k}^{(k)}(y_k), 1 \leq k \leq d, 1 \leq j_k \leq r_k$.

```
Interpolate the data on structured grid of size I_1 × ··· × I_d
Use TuckerMPI to get core tensor α and factor matrices W^(k) for given compression precision
for k = 1, ..., d do
    for j_k = 1, ..., r_k do
        Approximate w_{j_k}^(k) using components of W_{:,j_k}^(k) in P_p and W_{s,p} and estimate error (See section 3)
        Store coefficients of w_{j_k}^(k) corresponding to smaller approximation error
    end for
end for
```

Table 1. Specification of interpolation grid size and Tucker ranks obtained for different precision using TuckerMPI for two test cases

Dataset	Interpolation grid size	Precision (ϵ)	Size of core tensor
SP3D	$500 \times 500 \times 300$	1.0×10^{-2}	$25 \times 24 \times 8$
		1.0×10^{-4}	$57 \times 50 \times 17$
SP4D	$500 \times 500 \times 500 \times 300$	1.0×10^{-2}	$30 \times 38 \times 5 \times 11$

5 APPLICATION EXAMPLE

5.1 Specification of Datasets

We apply the method on a data set pertaining to a direct numerical simulation (DNS) of turbulent combustion. A "statistically planar" (SP) premixed flame [16] stabilized in homegeneous isotropic turbulence is simulated using the massively parallel DNS code S3D [6]. A premixed mixture of methane and air establish a flame that remains statistically planar and stationary in an oncoming turbulent flow. The combustion chemistry is described using a chemical mechanism containing six chemical species. Accordingly, at each point in the spatial grid and time the solution vector contains eleven dependent variables describing the full thermo-chemical state of the flame. The data set is mapped onto a 3-dimensional structured grid comprising 500 grid points in each satial dimension, and a total of 400 time snapshots are considered. For the illustration of the method, to follow, we consider two variants of this fundamentally 4-dimensional data set: a 3-D data set comprising only two spatial dimensions and time, and the full 4-D data set comprising all three spatial dmensions and time.

5.2 Illustrations

We apply our compression strategy on the dataset of the previous section for two cases. In the first case, henceforth referred to as SP3D, we consider that the data belongs to a three dimensional space, consisting of two spatial axis and one time axis. The total number of data points in this set is 7.5×10^7 with total storage cost of 0.6 gigabytes for double precision. The second case considers a 4 order tensor, SP4D, which also considers the third spatial axis, in addition to the ones in SP3D. The total storage cost of data in this case is 300 gigabytes with 3.75×10^{10} data points. In the following, we illustrate results of SP3D case, and mention that a similar illustrations can be obtained for SP4D.

In case of SP3D, we interpolate the data on a structured grid of size $500 \times 500 \times 300$ using only 10% of the data in the original set and decompose the resulting tensor in Tucker format using TuckerMPI. Figure 1 shows the decay in the absolute value of the components of the core tensor $\boldsymbol{\alpha}$ versus rank (multilinear Tucker rank on horizontal axis is converted to canonical rank) of SP3D for Tucker decomposition precision of 1.0×10^{-3}. We clearly see that there is a fast decay in the singular values thus indicating strong scope for compressibility of this dataset. Table 1 summarizes the interpolation parameters and multilinear Tucker ranks i.e. size of the core tensor thus obtained for different decomposition precisions for the two datasets.

We now consider functional approximations of singular vectors along the first mode. Figure 2(a) shows first singular vector $W^{(1)}_{:,1}$ and its corresponding functional approximations in $\mathbb{P}_{20}$ and $\mathbb{P}_{40}$. For better illustration, the corresponding approximation errors are plotted in Figure 2(b). We find that a sufficiently rich approximation space is necessary for accurate representation of singular vectors as point wise error for $p = 40$ is much smaller than with $p = 20$.

Figure 3(a) and (b) show similar plots for the last singular vector in the first mode i.e. $W^{(1)}_{:,57}$ in approximation spaces $\mathbb{P}_{40}$ and $\mathbb{W}_{3,5}$. We clearly see that, in this case, a multi-resolution wavelet basis is essential to get an accurate functional

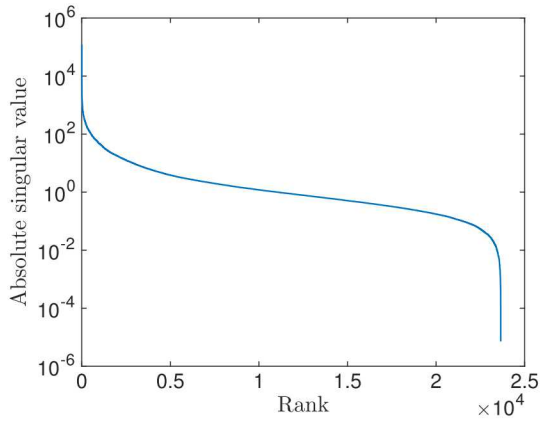
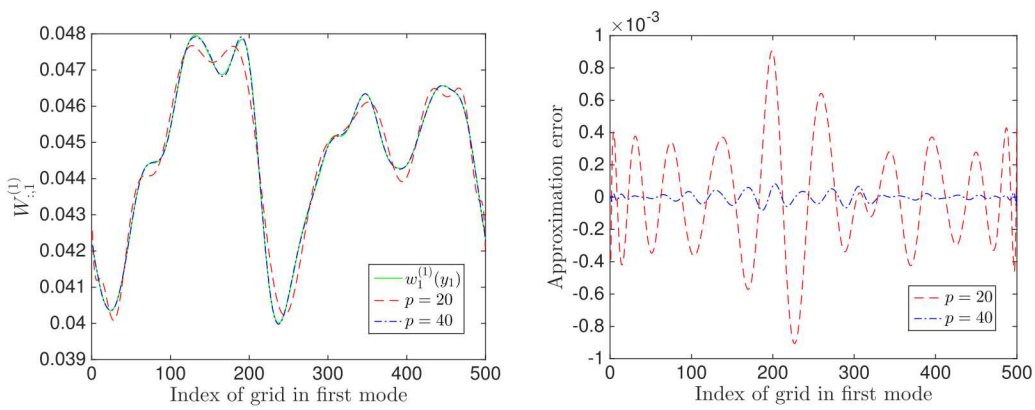


Fig. 1. Decay of singular values i.e. absolute value of components of core tensor $\boldsymbol{\alpha}$ v/s rank of SP3D with TuckerMPI precision of 1.0×10^{-3}. The rank on horizontal axis is converted to canonical rank by sorting the singular values in descending order.



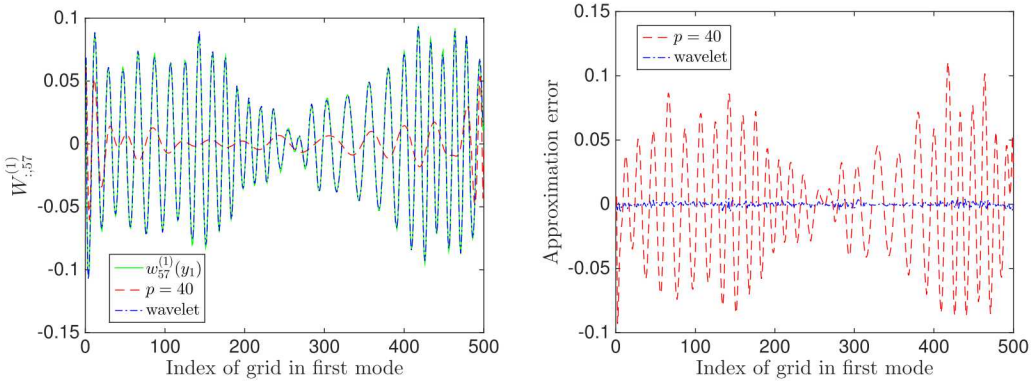
(a) (b)

Fig. 2. (a) Approximation of $w_1^{(1)}(y_1)$ using least squares with ℓ_1 regularization from data points as components of $W_{:,1}^{(1)}$ in the approximation space of Legendre polynomials of degree $p = 20$ and $p = 40$. (b) Point wise approximation error v/s grid index of the two approximations in (a)

Table 2. Compression results using functional sparse Tucker tensor

Dataset	Precision	Compression ratio	Storage cost
SP3D	1.01×10^{-2}	3879	155 KB
	1.9×10^{-3}	936	640KB
SP4D	1.1×10^{-2}	4.45×10^5	673 KB

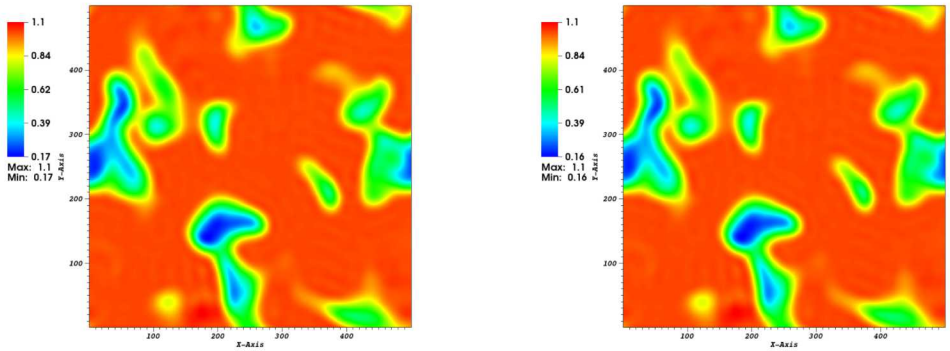
representation, although point wise approximation error is high as compared to the first singular vector. Table 2 shows compression error, compression ratio and storage cost for functional sparse Tucker tensor for both test cases.



(a) (b)

Fig. 3. Approximation of $w^{(1)}_{57}(y_1)$ using least squares with ℓ_1 regularization from data points as components of $W^{(1)}_{:,57}$ in the approximation space of Legendre polynomials of degree $p = 20$ and wavelets with resolution level 5 and degree 3. (b) Point wise approximation error v/s grid index of the two approximations in (a)

Finally, Figure 4 shows visualization of 2D slice of the original dataset obtained from reconstruction of data from TuckerMPI and functional Tucker tensor.



(a) (b)

Fig. 4. Visualization of a 2D slice of SP4D dataset using reconstruction of tensor obtained from TuckerMPI and functional sparse Tucker tensor.

6 CONCLUSION

We presented a novel technique to compress large volume of data using functional sparse Tucker decomposition. The key idea is to find a sufficiently accurate representation of data in the set of functional Tucker tensors with complexity smaller by orders of magnitude as compared to the size of dataset. In order to achieve this objective, we defined the set

of sparse functional Tucker tensors and used existing parallel implementation of Tucker decomposition to construct approximation in this set. The singular vectors are approximated as functions represented on suitable basis using least squares with sparse regularization. The entire compression scheme was tested on datasets obtained from high fidelity combustion modeling simulations. For small loss of accuracy, the proposed strategy results in compression ratio of up to 936 and 4.45×10^5 for a third order and fourth order dataset respectively.

7 ACKNOWLEDGMENT

We would like to acknowledge Tamara Kolda for discussions related to this work. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration.

REFERENCES

[1] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. 2012. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning* 4, 1 (2012), 1–106.

[2] Grey Ballard, Alicia Klinvex, and Tamara G Kolda. 2019. TuckerMPI: A Parallel C++/MPI Software Package for Large-scale Data Compression via the Tucker Tensor Decomposition. *arXiv preprint arXiv:1901.06043* (2019).

[3] G. Blatman and B. Sudret. 2011. Adaptive sparse polynomial chaos expansion based least angle regression. *J. Comput. Phys.* 230 (2011), 2345–2367.

[4] E. Candes, J. Romberg, and T. Tao. 2006. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on information theory* 52(12) (2006), 5406–5425.

[5] Gavin C Cawley and Nicola LC Talbot. 2004. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural networks* 17, 10 (2004), 1467–1475.

[6] J H Chen, A Choudhary, B de Supinski, M DeVries, E R Hawkes, S Klasky, W K Liao, K L Ma, J Mellor-Crummey, N Podhorszki, R Sankaran, S Shende, and C S Yoo. 2009. Terascale direct numerical simulations of turbulent combustion using S3D. *Computational Science & Discovery* 2, 1 (jan 2009), 015001. https://doi.org/10.1088/1749-4699/2/1/015001

[7] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. 2001. Atomic decomposition by basis pursuit. *SIAM review* 43, 1 (2001), 129–159.

[8] Mathilde Chevreuil, Régis Lebrun, Anthony Nouy, and Prashant Rai. 2015. A least-squares method for sparse low rank approximation of multivariate functions. *SIAM/ASA J. Uncertain. Quantif.* 3, 1 (2015), 897–921.

[9] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *2016 ieee international parallel and distributed processing symposium (ipdps)*. IEEE, 730–739.

[10] David L Donoho et al. 2006. Compressed sensing. *IEEE Transactions on information theory* 52, 4 (2006), 1289–1306.

[11] Alireza Doostan, AbdoulAhad Validi, and Gianluca Iaccarino. 2013. Non-intrusive low-rank separated approximation of high-dimensional stochastic models. *Comput. Methods Appl. Mech. Eng.* 263 (2013), 42–55.

[12] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. 2004. Least angle regression. *The Annals of statistics* 32, 2 (2004), 407–499.

[13] Nathaniel Fout, Kwan-Liu Ma, and James Ahrens. 2005. Time-varying, multivariate volume data reduction. In *Proceedings of the 2005 ACM symposium on Applied computing*. ACM, 1224–1230.

[14] Alex Gorodetsky, Sertac Karaman, and Youssef Marzouk. 2019. A continuous analogue of the tensor-train decomposition. *Computer Methods in Applied Mechanics and Engineering* 347 (2019), 59–84.

[15] Alex A Gorodetsky and John D Jakeman. 2018. Gradient-based optimization for regression in the functional tensor-train format. *J. Comput. Phys.* 374 (2018), 1219–1238.

[16] H. Kolla, X.-Y. Zhao, N. Swaminathan, and J.H. CHen. 2016. Velocity and Reactive Scalar Dissipation Spectra in Turbulent Premixed Flames. *Combustion Science and Technology* 188, 9 (2016), 1424–1439.

[17] Peter Lindstrom. 2014. Fixed-rate compressed floating-point arrays. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2674–2683.

[18] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2010. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* 11, Jan (2010), 19–60.

[19] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58, 1 (1996), 267–288.

[20] Ledyard R Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 3 (1966), 279–311.

[21] Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen. 2012. A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing* 34, 2 (2012), A1027–A1052.