https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Geoscientific Model Development Discussions

Adaptive lossy compression of climate model data based on hierarchical tensor with Adaptive-HGFDR (v1.0)

Zhaoyuan Yu[1,2], Zhengfang Zhang[1], Dongshuang Li[3,4], Wen Luo[1,2], Yuan Liu[1], Uzair Aslam Bhatti[1], Linwang Yuan[1,2,*]

[1]Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, Nanjing, China,
[2]Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing, China,
[3]Jiangsu Key Laboratory of Crop Genetics and Physiology/Jiangsu Key Laboratory of Crop Cultivation and Physiology, Agricultural College of Yangzhou University, Yangzhou, China,
[4]Jiangsu Co Innovation Center for Modern Production Technology of Grain Crops, Yangzhou University, Yangzhou, China

Correspondence to: Linwang Yuan (email: yuanlinwang@njnu.edu.cn)

Abstract. Lossy compression has been applied to large-scale experimental model data compression due to its advantages of a high compression ratio. However, few methods consider the uneven distribution of compression errors affecting compression quality. Here we develop an adaptive lossy compression method with the stable compression error for earth system model data based on Hierarchical Geospatial Field Data Representation (HGFDR). We extended the original HGFDR by firstly dividing the original data into a series of the local block according to the exploratory experiment to maximize the local correlations of the data. After that, from the mathematical model of the HGFDR, the relationship between the compression parameter and compression error in HGFDR for each block is analyzed and calculated. Using optimal compression parameter selection rule and an adaptive compression algorithm, our method, the Adaptive-HGFDR(v1.0), achieved the data compression under the constraints that the compression error is as stable as possible through each dimension. Experiments concerning model data compression are carried out based on the Community Earth System Model (CESM) data. The results show that our method has higher compression ratio and more uniform error distributions, compared with other commonly used lossy compression methods, such as the Fixed-Rate Compressed Floating-Point Arrays method.

1 Introduction

Earth System Model Data (ESMD), which comprehensively characterize the Earth system over space-time dimensions, are presented as multidimensional arrays of high-precision floating-point numbers(Kuhn et al., 2016; Wulder et al., 2012). With the rapid development of earth system models, ESMD has shown an exponential increase in data volume and data complexity(Anon, 2011; Of and Acm, 2000). For example, the CESM (Community Earth System Model) simulation generates data on the order of terabytes per computing day (Baker et al., 2014; Kay et al., 2015; Paul et al., 2015). Therefore, compression methods, especially the lossless compression methods are applied to reduce the data volumes. However, lossless compression methods have an upper limit of compression ratios (Kumar et al., 2008; Tao et al., 2017c), which grows much slower than the

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

velocity of data volume grows (Baker et al., 2016). With the rapid exploration of ESMD, lossy compressions are recently been studied as an alternative solution for ESMD compression. How to keep the balance between the compression errors and compression ratio becomes the key issue of lossy compression of ESMD.

The existing lossy compression methods for ESMD can be classified into two main categories: file-based compression and data encoding-based compression. File-based compression considers the model data as a complete data file and then compresses data using general file compressions, such as GRIB2 and NetCDF (Bing et al., 2014; Hübbe et al., 2013). However, the compression parameters of common data files are relatively fixed, and the error cannot be controlled in a data-driven way. Some researches also use the image-based file compression method in ESMD. These image-based methods slice ESMD from different dimensions and then compress different slices as separate images. For example, there are researches use JPEG2000 or a discrete Fourier transform to compress ESMD (Taubman and Marcellin, 2002). The image-based compression method can directly inherit the advantages of existing image compression methods. However, as the compressions are applied to single image slices, the correlations between different image slices are not always well utilized during different compression processes. Therefore, the compression error control between different image slices may be non-uniform (Castruccio and Genton, 2016; Guinness and Hammerling, 2016). To summarize, the file-based compression methods can improve the compression ratio of ESMD. However, the uneven distribution of compression errors may affect the data quality and then affect the subsequent analysis of ESMD (Baker et al., 2014; Berres et al., 2017; Feng et al., 2014; Zabala and Pons, 2011).

The second type of lossy compression, the data encoding-based compression methods, mainly implement data compression by encoding the common feature of data with more compact coding mechanisms to reduce the data volumes. With different encoding strategies, the data encoding-based compression methods can be further classified into three subcategories: statistical data coding, error truncation, and feature prediction-based coding compression. The statistical data coding methods use parametrical statistical characteristics to approximate original data (Papaioannou et al., 2011; Tao et al., 2017a). For example, vector quantization (VQ) (Vector Quantization) (Guinness and Hammerling, 2016) or sparse coding (Akbudak et al., 2017) organize the data as a simple one-dimensional array (number sequence) and use a dictionary (or code table) to compose the common patterns of the data. However, the use of a dictionary or code table enlarges the complexity of the data structure, which makes the structure of the compression algorithm complicated. The construction and parsing of the dictionary also make significant time to compress and decompress when the data volume is large (Anon, 2013; Liu et al., 2014; Mummadisetty, 2015). The method based on error truncation mainly controls the precision of a floating-point expression of original data, intercepts, and eliminates redundant floating-point precision to implement the data compression. For example, FPZIP (Lindstrom and Isenburg, 2006) and APAX (Hübbe et al., 2013) compress data by intercepting the floating-point precision of data. As the distribution of floating-point precision of data is not uniform, the compression errors may also distribute unevenly. Therefore, it is difficult to control the distribution of data compression error. To make the data error distribution more evenly distributed, the feature prediction-based coding compression methods try to extract features and use functions to predict the possible structure and coding of data (Adhianto et al., 2010; Cui et al., 2007). For example, NUMARCK (Zheng et al., 2017), SSEM (Wilczyñski, 2001), SPECK (Wang and Li, 2006), and ZFP (Diffenderfer et al., 2019b) are typical methods that use

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020


Geoscientific Model Development Discussions

the feature prediction to achieve lossy compression. Although the compression ratio is higher and the error distribution is less uneven distributed compared to the pure encoding-based methods, the compression ratios of the feature prediction-based coding compression methods are highly dependent on the feature extraction and prediction model. If the distribution of original data did not fit well with the feature extraction and prediction model, the performance of the compression may low. Furthermore, to make the method flexible to different data, common feature extraction and prediction model are data-adaptive, which means the compression parameters are often cannot be modified custom by users.

By summarizing all the above two different types of lossy compression methods, we find that existing data compression methods are mostly inherited from low dimensional data compression methods (e.g. one-dimensional vector or two-dimensional images). None of these methods considers the ESMD as a unified, overall high dimensional data with the the heterogeneous correlation between different dimensions. For ESMD, there are significant correlations between different dimensions (e.g. the temporal or spatial dimensions), i.e., that values in neighboring ranges tend to be numerically close to each other. As the dimensions of ESMD are commonly high (e.g. even an ESMD with only one attribute and three spatial dimensions forms a four-dimensional data), the ignorance of the multidimensional correlation structure results low compression performance (Diffenderfer et al., 2019a; Schoellhammer et al., 2004). Without control of such high dimensional correlation structure also makes it difficult to uniformly control the error distribution of data compression in different dimensions, which leads to an uneven distribution of the compression method in different dimensions and affects the quality of data (Tao et al., 2017b).

Tensors can effectively represent a multidimensional array of numerical values. The corresponding tensor decomposition method eliminates inconsistent, uncertain, and noisy data without destroying the intrinsic data structure, making the reconstructed data in approximate tensor more accurate than the initial tensor (Li et al., 2018); this method, therefore, has been gradually introduced into data compression in recent years (Yuan et al., 2015). Among these tensor methods, the hierarchical tensor approximation, which can extract data features level by level to obtain more detailed information, achieve far higher quality than traditional tensor methods at large compression ratios (Linton and Xiao, 2001; Lyre, 2004). For example, Yuan et al. recently designed an improved hierarchical tensor method Hierarchical Geospatial Field Data Representation (HGFDR) to compress geospatial data in a hierarchical tree structure, show many more advantages in compression ratio and error distribution than traditional methods like NetCDF-based data compression. Nevertheless, the HGFDR only pay attention to global average error to assess the compression quality and is only tested with eight types of climate variables. As the average error could be quite small despite a relatively large error at one or more points, maintain the stable distribution of the local error is important for the compressed climate data for the subsequent data analysis. We still need to work out how to control the balanced distribution of local compression error of the HGFDR and research on the adaptivity and universality of HGFDR with various variables of ESMD.

In this paper, we extend the study of HGFDR by discussing the factors and constraints that affect HGFDR for ESMD compression. We study the empirical quantitative relationship between compression error and compression parameters and

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

develop an Adaptive-HGFDR(v1.0) algorithm based on an adaptive data block and rank adjustment mechanism under error constraint. Experiments on a climate simulation set of 22 variables are developed to test the general applicability of the method. The remainder of this paper is organized as follows. Section 2 introduces the basic ideas about developing Adaptive-HGFDR(v1.0). Section 3 discusses the block mechanism, the relationship between rank and error, and the derivation of empirical equations, as well as the flow of the dichotomy. Section 4 uses temperature data to verify that the method can obtain adaptive rank under error constraint. Section 5 discusses the effectiveness and computational efficiency of the method, as well as the results.

2 Basic idea

For climate model data, due to the large dataset, the partial block data is not only much smaller than the original geographic spatial tensor, but also more balanced in size. Therefore, dividing the original data into a series of uniform local blocks is beneficial to the compression effect and calculation efficiency. Generally, the size of the main block depends on the size of the file system I/O (input/output) blocks, so there are usually multiples of the smallest block of a particular I/O. For the divided data, in order to construct data compression with stable error, the key problem to be solved is how to select the best compression parameter of each block data under the condition of stable error. For HGFDR, the rank on each block of data is the main control parameter to achieve compression. The empirical relationship between compression errors and compression parameters can be used to adjust the rank value of each block of data under error constraints to achieve a balanced distribution of error on each block of data.

In order to achieve a stable distribution of errors, for each block of divided data, the compression parameters should be able to be adaptively adjusted according to the given error. To achieve this regulation process, the rank is continuously adjusted by the given target error, and the error result after each compression is compared with the target error, and an empirical relationship is used to determine whether the rank value should be increased or decreased. The iteration is continued to obtain the result that is closest to the target error and meets the target error. In order to improve the efficiency of the algorithm in continuous iteration, a fast search method is used to quickly converge the iteration to the optimal compression result with minimal algorithm complexity. Using the idea of dichotomy, before adjusting the rank each time, narrow the selection interval of half of the rank. The optimal rank of the target error is constantly approached in half, and each data block can adaptively find the rank closest to the target error by this method. Therefore, through step 1: set s an appropriate partitioning strategy, step 2: constructs an empirical relationship between compression error and compression parameters, and uses this relationship to allow each data block to determine the rank value based on the given target error, step 3: Use the fast search method to find the best rank, finally achieve a stable distribution of the overall compression error. The entire compression method flow is shown in Figure 1.

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Geoscientific Model Development Discussions

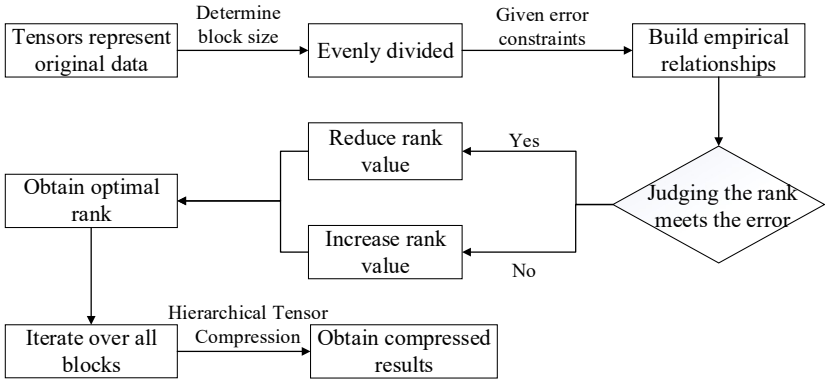


Figure 1: Flow chart of compression method in this paper.

3 Method

3.1 Block hierarchical tensor compression

EMSD is a multidimensional array with high dimensional features. it can be seen as a tensor with the spatio-temporal references and the associated attribute domain.Without loss of generality, a three-dimenisonal tensor can be definied as $Z \in \mathbb{R}^{I\times J\times K}$ (None, 1970; Suiker and Chang, 2000), where I , J , and K are values that represent the number of grids along the dimensions of longitude, latitude, and time (or height), respectively. Usually, these dimensions of EMSD are imbalance due to the different spatial and temporal resolution. For example, the data accumulation in the temporal dimension is always significantly longer than that in the spatial dimension for a spatio-temporal series with long continuous temporal observation.Thus， for the compression of EMSD, in order to reduce the affection of dimensional imbalance and make the blocks an ideal size, a blocking mechanism for original data Z is firstly formulated as:

$$Z = \{C_1, C_2, \cdots, C_m\} \tag{1}$$

where $C_i \in \mathbb{R}^{Q\times W\times E}$ represents equal data blocks divided from the original data.

Based on the divided data blocks, Yuan (Yuan et al., 2015) proposed the HGFDR based on the hierarchical tensor compression. In this method, the hierarchical tensor compression is applied to each block, then the hierarchical tensor compression of each data block is obtained by selecting the dominant feature component and filtering out the residual structure. This method utilizes the hierarchical structure of data features, greatly reducing data redundancy, and thereby achieving the efficient compression of amounts of spatiotemporal data (Yuan et al., 2015). The overall compression of the HGFDR can be formulated as:

$$\begin{cases} H(A) = (U_R \otimes U_{R-1} \otimes \cdots \otimes U_1)\tilde{B}_L \tilde{B}_{L-1} \cdots \tilde{B}_1 B_{12\cdots R} + res \\ \tilde{B}_j = B_{P_{Lj}} \otimes \cdots \otimes B_{P_L} \qquad j = \{1, 2, \ldots, L\} \end{cases} \tag{2}$$

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020


Geoscientific Model Development Discussions

Similar to the prominent components obtained by SVD (Lathauwer et al., 2000; Springer, 2011) for two-dimensional data, the matrix U_R and the sparse transfer tensor B_R are considered to be the r-th component of a third-order tensor in each dimension, respectively, where R denotes the number of multi-domain features. The residual tensor, res, in Eq. (2) denotes information not captured by the decomposition model, and $(U_R \otimes U_{R-1} \otimes \cdots \otimes U_1)\tilde{B}_L \tilde{B}_{L-1} \cdots \tilde{B}_1 B_{12\cdots R}$ in Eq. (2) is the reconstructed r-th core tensor and feature matrix (Matrices, 2006; Oseledets and Tyrtyshnikov, 2009).

3.2 Adaptive parameter selection and solution

Since the feature structure of each divided block is different (Hackbusch and Kühn, 2009), the key to control the stable distribution of compression error in HGFDR is to adaptively select the compression parameter of each local data according to the given compression error. So the key step is to construct the relationship between the compression error and compression parameter. Lars Grasedyck defines a hierarchal tensor SVD algorithm, and the approximate accuracy is determined by rank (Matrices, 2006). In HGFDR, Yuan gives the relationship between the compression error and compression parameter as $\varepsilon = \alpha Rank^{-\beta}$, since the structure of each local data is different. Under the constraint of uniform compression error distribution, the compression parameter of each block data should be the rank value closest to the given error as follows:

$$\varepsilon = \alpha Rank^{-\beta} \le \varepsilon_{Given} \tag{3}$$

ε_{Given} are the given threshold values of calculation error that depend on different application scenarios; α, β are the calculation coefficients determined by the structure and complexity of the data; In HGFDR，the relationship between the compression ratio and compression parameter are given as follows：

$$\varphi = \frac{datasize}{aRank^3 + bRank^2 + cRank + d} \tag{4}$$

As shown in Eqs. (2), (3), and (4), in the HGFDR, with rank decreases, the data compression rate of HGFDR increases, but the compression error also increases. In HGFDR, the rank value of different blocks is fixed, it results in the fluctuation of the compression error in specific dimension. Since the structure of each block is different, to achieve a uniform error distribution of compressed data under the given compression error, the key is to select the rank for each block of data separately. We can select the optimum parameter as the minimum $Rank$ that make the conpression error close to the given value.

In the following algorithm for finding the optimum parameter for data block $C_i \in \mathbb{R}^{Q \times W \times E}$, std_err is the given data error; err is the actual error obtained by each data block compression; R_Max, R_Min, and R_Mid are the transitive values for finding the optimum rank; $Round()$ is the rounding function; $Max()$ represents taking the maximum; and $Rank$ is the optimum parameter. We use a binary search to find optimal parameters, and the algorithm is implemented as follows:

(1) Input a data block $C_i \in \mathbb{R}^{Q \times W \times E}$, and set std_err, $R\_Max = Max(Q, W, E)$ and $R\_Min = 0$;

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Geoscientific Model Development Discussions

(2) $R\_Mid = Round(\frac{R\_Max + R\_Min}{2})$;

(3) $C_i \in \mathbb{R}^{Q \times W \times E}$ obtains err according to Eqs. (4) and (6)

(4) Judge $err > std\_err$. If yes, go to (5), or else to (6);

(5) Judge $R\_Mid - R\_Min = 1$. If yes, $Rank = R\_Mid$, or else $R\_Max = R\_Mid$ and return to (2);

(6) Judge $R\_Max - R\_Mid = 1$. If yes, $Rank = R\_Mid$, or else $R\_Min = R\_Mid$ and return to (2).

The complexity of the optimum parameter is $O(\log n)$. According to the above optimal component rule, the algorithm flow chart is shown in Figure 2:

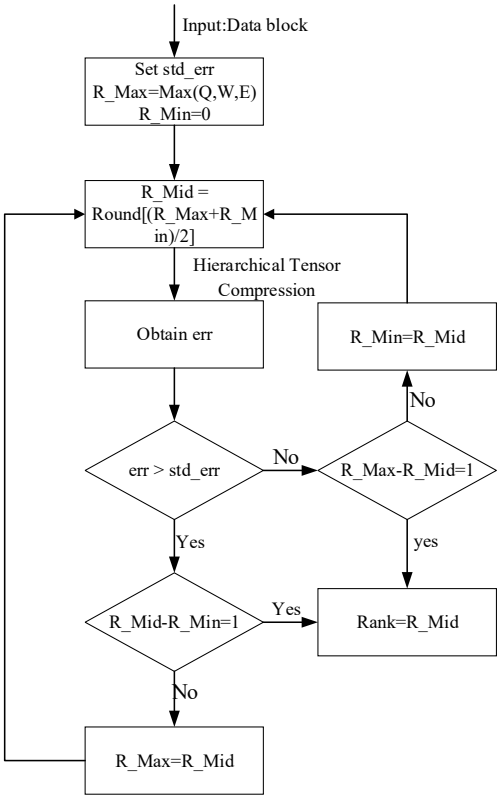


Figure 2: Data compression algorithm workflow based on dichotomy.

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Geoscientific Model Development Discussions

4 Case study

4.1 Data description and experimental configuration

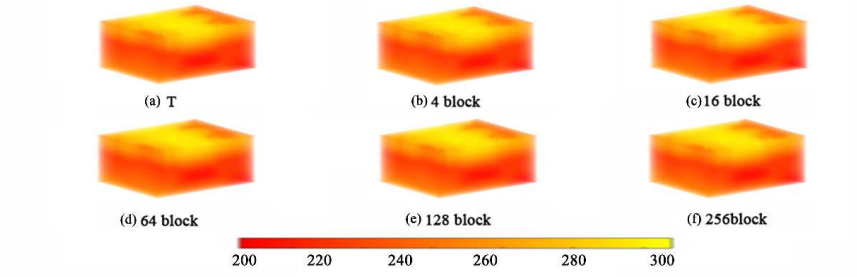
In this paper, the experimental data are dynamics simulation files obtained from Open Science Data Cloud. The data files are in the common NetCDF (network Common Data Form) format (Springer, 2011). Each file is broken down into year and month, containing all the variables for one month. The data set includes air temperature data (T) stored as a $1024 \times 512 \times 26$ (latitude × longitude × height) tensor and 22 other attributes stored as a $1024 \times 512 \times 221$ (latitude × longitude × time) tensor from 1980/01 through 1998/05. The memory occupation of the temperature data attribute was about 38.3 M, and the memory occupation of 22 other attributes were about 0.73 GB. By importing data from the NetCDF into the memory, a total of 48 GB of data was made available to test the performance of our solution. Research experiments were performed by the MATLAB R2017a environment on a Windows 10 Workstation (HP Compaq Elite 8380 MT) with Intel Corei7-3770 (3.4 GHz) processors and 8 GB of RAM.

For validation of our proposed algorithm, compression error ratio, and compression ratio are mainly used to benchmark performance. The following experiments were performed:

- Simulations with different block numbers were performed with data of constant size to find the optimized block size;
- A comparison between the compressed performance in our solution and that is commonly used compression methods;
- A comparison of compressed performances using multiple variables.

4.2 Optimal block number selection

In the proposed compression method, the block number is a vital parameter. Here, taking temperature data (T) as an example, the optimum block number is decided under the conditions of the given compression error. Due to knowledge limitations, the block numbers are randomly selected as 4, 16, 64, and 128, and the determined compression error is 10^{-4}; the proposed hierarchical tensor compression is applied. The corresponding compressed results and statistical parameters are shown in Figures 3 and 4.



https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Figure 3: Original data and compression with different block numbers: (a) original temperature data (T); (b) compression result with block number 4; (c) compression result with block number 16; (d) compression result with block number 64; (e) compression result with block number 128; (f) compression result with block number 256.

The compression results with different block strategies in Figures 3(b)–3(e) show little difference compared with the original data in Figure 3(a). This may be because the proposed method adaptively extracts the prominent feature components under the same compression error range, no matter how large the local data block; the proposed method can continually adjust the parameter to meet the same error constraints, thus providing good compression results for different block numbers.

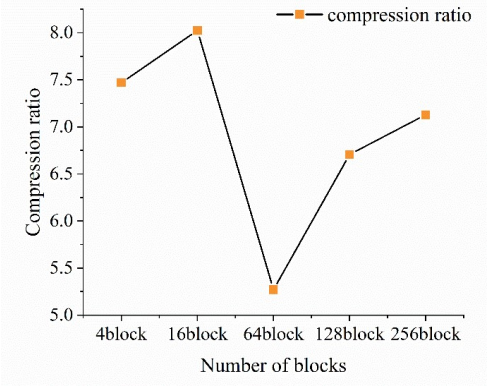


Figure 4: Compression ratio compression results with different block numbers.

Figure 4 shows that the compression ratio reaches a maximum when the block number is set to 16. Hence, the optimum block number is 16, and the corresponding block size is $256 \times 128 \times 26$. The ideal block number should achieve a high compression ratio.

4.3 Comparison with traditional methods

To verify the proposed adaptive compression method for climate model data, the method is compared with the traditional HGFDR method and the classical ZFP compression method. To compare methods, the block numbers in the proposed method and HGFDR method are both set to 16; and for convenience in comparison, the rank in HGFDR is selected as the average of the adaptive rank. Without loss of generality, the compression errors are set to 10^{-5}, 5×10^{-5}, 10^{-4}, 5×10^{-4}, 10^{-3} respectively; the compression ratios are shown in Figure 5.

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

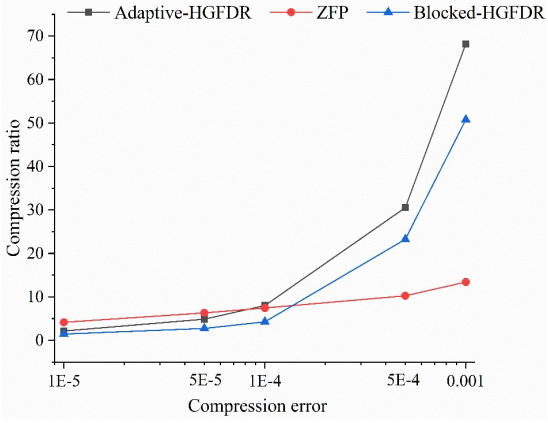


Figure 5: Compression ratio versus compression error for different approaches.

Figure 5 shows that the proposed method reaches the highest compression ratio with the corresponding compression error in most cases. This may because this method can adaptively adjust the parameter according to actual data complexity, and thus better capture data features to improve the compression ratio. To better show the stability in the compression error distribution, under the condition of the compression error of 10^{-4}, the error distribution along the longitude dimension is shown in Figure 6, and the detailed distributions of spatial error, three spatial pieces (2 layers, 8 layers, 16 layers), are randomly selected and shown in Figure 7. Figure 6 shows that the compression error in the HGFDR method and the ZFP method fluctuates dramatically, forming multiple peaks and valleys. The proposed method is more stable than traditional methods given the same whole compression error ratio. The spatial distribution of error in Figure 7 also shows that the error distribution obtained by the proposed method is more uniform than that in the HGFDR and ZFP methods.

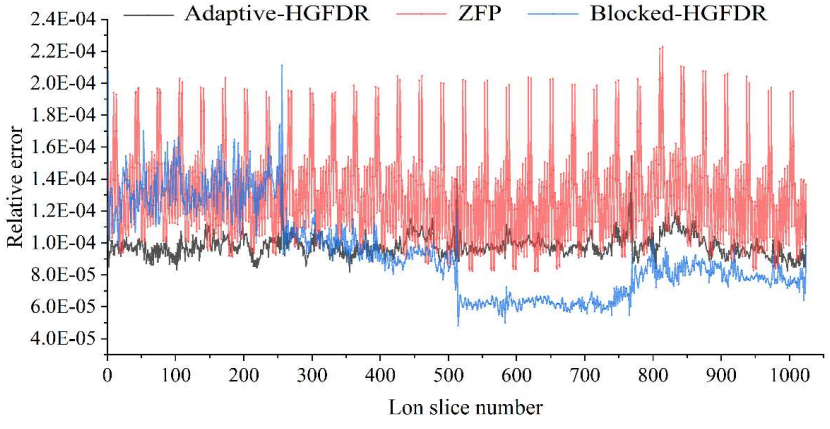


Figure 6: Compression error distribution of three compression methods on longitudanal slices.

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Geoscientific Model Development Discussions

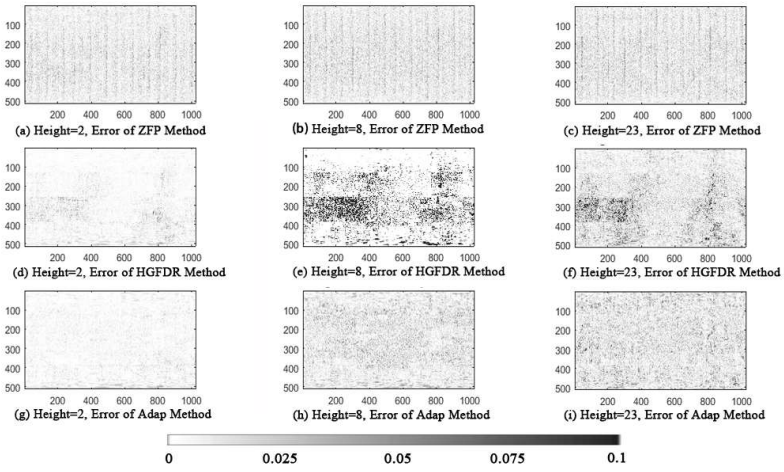


Figure 7: Spatial compression distribution of compression error for three compression methods.

4.4 Compression performance comparison for multiple variables

For a comprehensive comparison of the different methods, 22 monthly climate model data were entered as experimental data. All experimental data have a dimension of 1024×512×221. The error limit is 0.01, the block size is 256×128×26, and the block number is 144. A detailed description of the variables is shown in Table 1.

Table 1: 22 Descriptions of climate model data variables.

Variable name	Variable description	Variable name	Variable description
FLDS	Downwelling longwave flux at the surface	PCONVT	Convection top pressure
FLDSC	Clearsky downwelling longwave flux at surface	RHREFHT	Reference height relative humidity
FLNSC	Clearsky net longwave flux at surface	SOLIN	Solar insolation
FLNT	Net longwave flux at top of model	SRFRAD	Net radiative flux at surface
FLNTC	Clearsky net longwave flux at top of model	TMQ	Total (vertically integrated) precipitable water
FLUT	Upwelling longwave flux at top of model	TREFHT	Reference height temperature
FLUTC	Clearsky upwelling longwave flux at top of model	TREFMNAV	Average of TREFHT daily minimum
FSDSC	Clearsky downwelling solar flux at surface	TREFMXAV	Average of TREFHT daily maximum
FSNSC	Clearsky net solar flux at surface	TS	Surface temperature (radiative)
FSNTC	Clearsky net solar flux at top of model	TSMN	Minimum surface temperature over output period
FSNTOAC	Clearsky net solar flux at top of atmosphere	TSMX	Maximum surface temperature over output period

The proposed method, HGFDR method, and ZFP method were applied to the 22 variables. The comparison ratio, time, and standard deviation of the slice error in the X dimension were calculated and are shown in Figure 8. We can conclude that the

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

proposed method maintains the maximum compression ratio under the constraints of the same compression error for most variables shown in Figure 8(a). This may be because tensor reconstruction based on selected feature components removes data redundancy to improve the compression ratio. Moreover, the adaptive adjustment of parameters makes the proposed method yield the stationary error distribution for the multiple variables shown in Figure 8(c). In summary, the proposed method provides good adaptability for climate model data.

However, of all the methods, the proposed method is the most time consuming [Figure 8 (b)], which may be because it requires the continual adjustment of parameters to search for the optimum rank. We pay more attention to the method construction in this work, for the efficient tensor compression solution, some optimization strategies, such as the spatiotemporal indexes, the unbalanced block split, can be used for improving the efficiency.

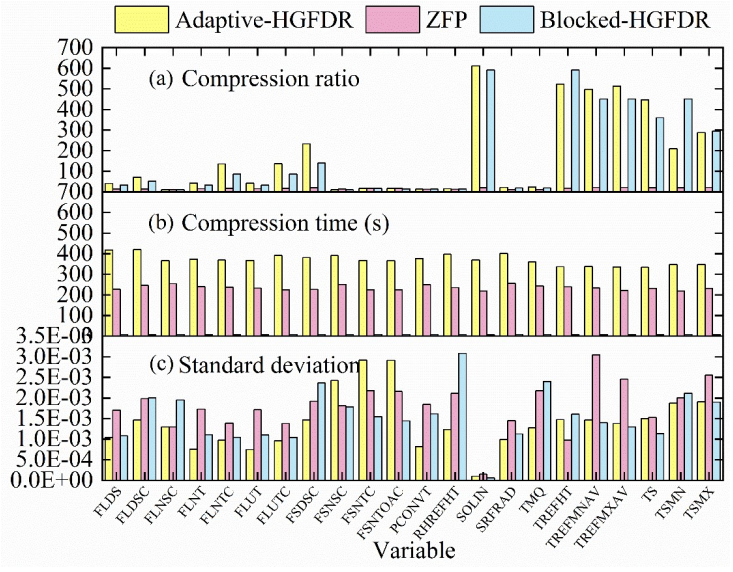


Figure 8: Comparison of the compression results of three compression methods for 22 variables: (a) Compression ratio comparison; (b) compression time comparison; (c) standard deviation comparison of slice error.

5 Conclusion

In this paper, an Adaptive-HGFDR(v1.0) algorithm is proposed for ESMD compression. Based on HGFDR, an error-compression parameter correction mechanism is established. Using the error and compression parameter empirical equations and error parameter adaptive dynamic adjustment feedback mechanism, a balanced distribution of climate model tensor compression errors is ensured. Compared with the traditional method, proposed method adaptively adjusts the local compression parameters, so that it can better take into account the local structural characteristics of the climate model data.

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Proposed algorithm not only maintain a large compression ratio under the same overall error, but it can also make the error structure more evenly distributed in all dimensions of the data. Multivariable simulation experiments show that the method has better performance and can be applied to many different types of climate model variables. Future research work should focus on conducting larger-scale compression experiments, optimizing algorithm efficiency, and evaluating the applicability of the method in large-scale climate simulation experiments.

Code and data availability. The Adaptive-HGFDR(v1.0) lossy compression algorithm proposed in this paper was conducted out in MATLAB R2017a. The exact version of Adaptive-HGFDR(v1.0) and experimental data used in this paper is archived on Zenodo(AndyWZJ, 2020). The experimental data are Large-scale Data Analysis and Visualization Symposium Data obtained from (OSDC) Open Science Data Cloud. This data set consists of files from a series of global climate dynamics simulations run on the Titan supercomputer at Oak Ridge National Laboratory in 2013 by postdoctoral researcher Abigail Gaddis, Ph.D. The simulations were performed at approximately 1/3-degree spatial resolution, or a mesh size of 1024x512 for 2D. We downloaded this simulation data in the common NetCDF (network Common Data Form) format in 2016 from https://www.opensciencedatacloud.org/.

Author contribution. Zhaoyuan Yu, Linwang Yuan and Wen Luo designed the paper's ideas and methods. Zhengfang Zhang and Yuan Liu implemented the method of the paper with code. Zhaoyuan Yu, Zhengfang Zhang and Dongshuang Li wrote the paper with considerable input from Linwang Yuan. Uzair Aslam Bhatti revised and checked the language of the paper.

Funding. This work was financially supported by the National Natural Science Foundation of China[41625004 41971404] and the National Key R&D Program of China[2017YFB0503500].

Competing interests. The authors declare that they have no conflict of interest.

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

(OGL). The Creative Commons Attribution 4.0 License and the OGL are interoperable and do not conflict with, reduce or limit each other. © Crown copyright YEAR

References

Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor-Crummey, J. and Tallent, N. R.: HPCTOOLKIT: Tools for performance analysis of optimized parallel programs, Concurr. Comput. Pract. Exp., 22(6), 685–701, doi:10.1002/cpe, 2010.

Akbudak, K., Ltaief, H., Mikhalev, A. and Keyes, D. E.: Tile Low Rank Cholesky Factorization for Climate/Weather Modeling Applications on Manycore Architectures, in International Supercomputing Conference., 2017.

AndyWZJ: AndyWZJ/Adaptive-lossy-compression- v1.0, , doi:10.5281/ZENODO.3862130, 2020.

Anon: Earth science data compression issues and activities, Remote Sens. Rev., 2011.

Anon: Data Reduction Analysis for Climate Data Sets, Int. J. Parallel Program., 2013.

Baker, A. H., Xu, H., Dennis, J. M., Levy, M. N. and Wegener, A.: A methodology for evaluating the impact of data compression on climate simulation data, ACM., 2014.

Baker, A. H., Hammerling, D. M., Mickelson, S. A., Xu, H. and Lindstrom, P.: Evaluating Lossy Data Compression on Climate Simulation Data within a Large Ensemble, Geosci. Model Dev. Discuss., 2016.

Berres, A. S., Turton, T. L., Petersen, M., Rogers, D. H. and Ahrens, J. P.: Video Compression for Ocean Simulation Image Databases, in Workshop on Visualisation in Environmental Sciences (EnvirVis)., 2017.

Bing, Li, Lin, Zhang, Zhuangzhuang, Shang, Qian and Dong: Implementation of LZMA compression algorithm on FPGA., Electron. Lett., 2014.

Castruccio, S. and Genton, M. G.: Compressing an Ensemble With Statistical Models: An Algorithm for Global 3D Spatio-Temporal Temperature, Technometrics, 58(3), 319–328, 2016.

Cui, X.-N., Kim, J.-W., Choi, J.-U. and Kim, H.-I.: Reversible Watermarking in JPEG Compression Domain, J. Korea Inst. Inf. Secur. Cryptol., 17, 121–130, 2007.

Diffenderfer, J., Fox, A. L., Hittinger, J. A., Sanders, G. and Lindstrom, P. G.: Error analysis of ZFP compression for floating-point data, SIAM J. Sci. Comput., 41(3), A1867–A1898, doi:10.1137/18M1168832, 2019a.

Diffenderfer, J., Fox, A., Hittinger, J., Sanders, G. and Lindstrom, P.: Error Analysis of ZFP Compression for Floating-Point Data, SIAM J. Sci. Comput., 41, A1867–A1898, doi:10.1137/18M1168832, 2019b.

Feng, J., Wu, Z. and Liu, G.: Fast Multidimensional Ensemble Empirical Mode Decomposition Using a Data Compression Technique, J. Clim., 27(10), 3492–3504, 2014.

Guinness, J. and Hammerling, D.: Compression and Conditional Emulation of Climate Model Output, 2016.

Hackbusch, W. and Kühn, S.: A new scheme for the tensor representation, J. Fourier Anal. Appl., 15(5), 706–722, doi:10.1007/s00041-009-9094-9, 2009.

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Hübbe, N., Wegener, A., Kunkel, J. M., Ling, Y. and Ludwig, T.: Evaluating lossy compression on climate data, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 7905 LNCS, 343–356, doi:10.1007/978-3-642-38750-0_26, 2013.

Kay, J. E., Deser, C., Phillips, A., Mai, A., Hannay, C., Strand, G., Arblaster, J. M., Bates, S. C., Danabasoglu, G., Edwards, J., Holland, M., Kushner, P., Lamarque, J. F., Lawrence, D., Lindsay, K., Middleton, A., Munoz, E., Neale, R., Oleson, K., Polvani, L. and Vertenstein, M.: The community earth system model (CESM) large ensemble project : A community resource for studying climate change in the presence of internal climate variability, Bull. Am. Meteorol. Soc., 96(8), 1333–1349, doi:10.1175/BAMS-D-13-00255.1, 2015.

Kuhn, M., Kunkel, J. and Ludwig, T.: Data compression for climate data, Supercomput. Front. Innov., 3(1), 75–94, doi:10.14529/jsfi160105, 2016.

Kumar, V. S., Nanjundiah, R., Thazhuthaveetil, M. J. and Govindarajan, R.: Impact of message compression on the scalability of an atmospheric modeling application on clusters, Parallel Comput., 34(1), 1–16, 2008.

Lathauwer, L. D. E., Moor, B. D. E. and Vandewalle, J.: On the best rank-1 and rank-(, , 21(4), 1324–1342, 2000.

Li, D. S., Yang, L., Yu, Z. Y., Hu, Y. and Yuan, L. W.: A tensor-based interpolation method for sparse spatio-temporal field data, J. Spat. Sci., 1–19, 2018.

Lindstrom, P. and Isenburg, M.: Fast and efficient compression of floating-point data, IEEE Trans. Vis. Comput. Graph., 12(5), 1245–1250, doi:10.1109/TVCG.2006.143, 2006.

Linton, O. B. and Xiao, Z.: A nonparametric regression estimator that adapts to error distribution of unknown form, Sfb Discuss. Pap., 2001.

Liu, S., Huang, X., Ni, Y., Fu, H. and Yang, G.: A high performance compression method for climate data, Proc. - 2014 IEEE Int. Symp. Parallel Distrib. Process. with Appl. ISPA 2014, 68–77, doi:10.1109/ISPA.2014.18, 2014.

Lyre, H.: Holism and structuralism in U(1) gauge theory, Stud. Hist. Philos. Sci. Part B Stud. Hist. Philos. Mod. Phys., 35(4), 643–670, 2004.

Matrices, H.: f u ¨ r Mathematik in den Naturwissenschaften Leipzig, , (March), 2006.

Mummadisetty, B. C.: Performance Analysis of Hybrid Algorithms For Lossless Compression of Climate Data, , (December), 2015.

None: On the spectra of tensor products of linear operators in Banach spaces., J. Für Die Reine Und Angew. Math., 1970(244), 1970.

Of, C. and Acm, T. H. E.: November 2000/Vol. 43, No. 11 COMMUNICATIONS OF THE ACM, Commun. ACM, 43(11), 68–77, 2000.

Oseledets, I. V and Tyrtyshnikov, E. E.: Breaking the Curse of Dimensionality, Or How to Use SVD in Many Dimensions, Siam J. Sci. Comput., 31(5), 3744–3759, 2009.

Papaioannou, T. G., Riahi, M. and Aberer, K.: Towards Online Multi-model Approximation of Time Series, 2011 IEEE 12th Int. Conf. Mob. Data Manag., 1, 33–38, 2011.

https://doi.org/10.5194/gmd-2020-124
Preprint. Discussion started: 2 June 2020

Paul, K., Mickelson, S., Dennis, J. M., Xu, H. and Brown, D.: Light-Weight Parallel Python Tools for Earth System Modeling Workflows, in IEEE BigData 2015: Workshop on Big Data in the Geosciences., 2015.

Schoellhammer, T., Greenstein, B., Osterweil, E., Wimbrow, M. and Estrin, D.: Lightweight temporal compression of microclimate datasets [wireless sensor networks], in Local Computer Networks, 2004. 29th Annual IEEE International Conference on., 2004.

Springer, U. S.: Community Earth System Model (CESM), Encycl. Parallel Comput., 351, 2011.

Suiker, A. S. J. and Chang, C. S.: Application of higher-order tensor theory for formulating enhanced continuum models, Acta Mech., 142(1–4), 223–234, 2000.

Tao, D., Sheng, D., Chen, Z. and Cappello, F.: Exploration of Pattern-Matching Techniques for Lossy Compression on Cosmology Simulation Data Sets, 2017a.

Tao, D., Di, S. and Cappello, F.: Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization, , doi:10.1109/IPDPS.2017.115, 2017b.

Tao, D., Di, S., Guo, H., Chen, Z. and Cappello, F.: Z-checker: A Framework for Assessing Lossy Compression of Scientific Data, Int. J. High Perform. Comput. Appl., (12), 2017c.

Taubman, D. and Marcellin, M.: JPEG2000: Image Compression Fundamentals, Standards and Practice, Springer Int., 11(2), 286, 2002.

Wang, N. and Li, X.: New low memory set partitioned embedded block coder, Tien Tzu Hsueh Pao/Acta Electron. Sin., 34(11), 2068–2071, 2006.

Wilczyñski, K.: SSEM: A computer model for a polymer single-screw extrusion, J. Mater. Process. Technol., 109(3), 308–313, doi:10.1016/S0924-0136(00)00821-9, 2001.

Wulder, M. A., Masek, J. G., Cohen, W. B., Loveland, T. R. and Woodcock, C. E. %J R. S. of E.: Opening the archive: How free data has enabled the science and monitoring promise of Landsat, , 122(Complete), 2–10, 2012.

Yuan, L., Yu, Z., Luo, W., Hu, Y., Feng, L. and Zhu, A. X.: A hierarchical tensor-based approach to compressing, updating and querying geospatial data, IEEE Trans. Knowl. Data Eng., 27(2), 312–325, doi:10.1109/TKDE.2014.2330829, 2015.

Zabala, A. and Pons, X.: Effects of lossy compression on remote sensing image classification of forest areas, Int. J. Appl. Earth Obs. Geoinf., 13(1), 0–51, 2011.

Zheng, Y., Hendrix, W., Son, S. W., Federrath, C., Agrawal, A., Liao, W. K. and Choudhary, A.: Parallel Implementation of Lossy Data Compression for Temporal Data Sets, 2017.