SAND2018-7410J

Adaptive Wavelet Compression of Large Additive Manufacturing Experimental and Simulation Datasets

Maher Salloum*[1], Kyle L. Johnson[2], Joseph E. Bishop[3],

Jon M. Aytac[1], Daryl Dagel[4], Bart G. van Bloemen Waanders[5]

June 21, 2018

∗ (Corresponding author) Sandia National Laboratories, 7011 East Ave., MS 9158, Livermore, CA 94550, Email: mnsallo@sandia.gov, Phone: 925-294-2585

1 Sandia National Laboratories, 7011 East Ave., MS 9158, Livermore, CA 94550

2 Sandia National Laboratories, 1515 Eubank SE., MS 0346, Albuquerque, NM 87123

3 Sandia National Laboratories, 1515 Eubank SE., MS 0840, Albuquerque, NM 87123

4 Sandia National Laboratories, 1515 Eubank SE., MS 1085, Albuquerque, NM 87123

5 Sandia National Laboratories, 1515 Eubank SE., MS 1318, Albuquerque, NM 87123

{mnsallo,kyljohn,jebisho,jmaytac,djdagel,bartv}@sandia.gov

Abstract

New manufacturing technologies such as additive manufacturing require research and development to minimize the uncertainties in the produced parts. The research involves experimental measurements and large simulations, which result in huge quantities of data to store and analyze. We address this challenge by alleviating the data storage requirements using lossy data compression. We select wavelet bases as the mathematical tool for compression. Unlike images, additive manufacturing data is often represented on irregular geometries and unstructured meshes. Thus, we use Alpert tree-wavelets as bases for our data compression method. We first analyze different basis functions for the wavelets and find the one that results in maximal compression and miminal error in the reconstructed data. We then devise a new adaptive thresholding method that is data-agnostic and allows *a priori* estimation of the reconstruction error. Finally, we propose metrics to quantify the global and local errors in the reconstructed data. One of the error metrics addresses the preservation of physical constraints in reconstructed data fields, such as divergence-free stress field in structural simulations. While our compression and decompression method is general, we apply it to both experimental and computational data obtained from measurements and thermal/structural modeling of the sintering of a hollow cylinder from metal powders using a Laser Engineered Net Shape process. The results show that monomials achieve optimal compression performance when used as wavelet bases. The new thresholding method results in compression ratios that are two to seven times larger than the ones obtained with commonly used thresholds. Overall, adaptive Alpert tree-wavelets can achieve compression ratios between one and three orders of magnitude depending on the features in the data that are required to preserve. These results show that Alpert tree-wavelet compression is a viable and promising technique to reduce the size of large data structures found in both experiments and simulations.

Keywords. Wavelets, Compression, Unstructured Mesh, Additive Manufacturing, Temperature, Stress, Compression Ratio, Reconstruction, Threshold, Error Metric, Divergence

1 Introduction

Additive manufacturing (AM) is transforming our abilities to create engineered components and subassemblies. AM broadly describes the fabrication of parts by the addition of material and generally excludes material removal processes as is done in other machining techniques [1]. Material addition is frequently performed in a layer-by-layer manner. In metal AM, material is most often consolidated within each layer by feeding metal feedstock such as powder or rod into a stationary heat source such as a laser beam or by selectively melting material in a bed of powder with a moving heat source [2]. The flexibility of systematically adding layers of melted material on a substrate in a spatially and temporally non-restrictive manner has enabled the creation of new designs that previously were not possible through standard manufacturing techniques. Not only can innovative designs be considered, but the AM process is fast, efficient, and relatively inexpensive. However, most of the AM-produced parts are unfortunately plagued with spatially-varying anomalies such as porosity, texture, inclusions and residual stresses at different spatial scales. These anomalies often lead to spatially-varying material properties. Therefore one of the main challenges is to control the AM process so that parts exhibit consistent material properties at all length scales. In addition, the AM process is not always repeatable. Variations in material properties have been observed within a single build as well as variations from multiple builds using the same operational settings.

These uncertainties have motivated the development of numerical simulation capabilities to help diagnose the occurence of anomalies and to help with the prediction of material performance. For instance, the prediction of residual stress using computational mechanics or the prediction of the melt pool dynamics using the heat equation and Navier-Stokes equations are critical to the underlying performance of the part. Computational simulations are needed in three dimensions with multi-scale resolution and long time horizons. These simulations are data-intensive and require large computational resources and storage due to the complex part geometries and the coupled physical processes that take place during AM. For large parts when local temperature histories are desired for the entire build, datasets can become very large and exceed 1 TB. Though disk space is becoming cheaper and more accessible, in the coming digital twin era when build information will be stored for many parts for long periods of time, methods to reduce storage requirements will become necessary. Furthermore, I/O time represents a significant portion of solution time in the described computational simulations. Though not the main focus of this study, simulation codes that could produce and operate on compressed data would potentially greatly speed up solution times. The large storage requirements motivate our main research goal. A more efficient storage and transfer process of the final data is required to post-process solution vectors.

In our study, we investigate data compression techniques in the context of simulations relevant to additive manufacturing. We investigate heat models in which the heat source (laser) only affects a small volume of the overall computational domain at any particular time instance, which suggests that the majority of the dynamics of the computational domain do not change and can likely be significantly compressed while maintaining the representation quality. Furthermore, the calibration of these models rely on thermal images which in turn are very large with similar spatial and temporal features that can also benefit from compression.

Both experimental and computational additive manufacturing data are defined in both space and time. Most compression approaches are able to act directly on spatial data fields gathered at several snapshots in time. However, this requires accumulating in memory data fields collected at several time steps or frames, which might be prohibitive due the limited memory buffers in experimental instruments and RAM in computational clusters. Therefore, we only examine compression at specific time steps or frames in this paper. We refer to this approach as *in-situ* compression.

1.1 Review of Data Compression Methods

Several scientific data compression methods exist in the literature [3]. These can be partitioned into two categories: lossless and lossy methods. Compression ratios (R) are usually found to be low ($2 < R < 10$) in lossless methods such as *bzip2* [4] and might not address the challenges of large AM data storage and transfer. We chose lossy compression in our work because it results in higher compression ratios, up to two orders of magnitude even when exercised in-situ *i.e.* when the data is still in memory as it is produced at every simulation step or camera frame.

Lossy compression of large scientific datasets has recently been the subject of multiple research efforts. The ISABELA compressor [5, 6] is based on data sorting and B-splines. The DIRAQ compressor [7] is a faster, distributed, in-situ data indexing and query system. Similar to lossless methods, ISABELA and DIRAQ suffer from low compression ratios. In the *zfp* compressor [8], blocks of floating-point data are compressed by reducing their bitstream representation. This preserves a good accuracy in the results while producing compression ratios larger than 10. However, *zfp* is limited to data represented on regular grids. Vector Quantization (VQ) [9, 10] is a common approach to compress regular grid data. It works by encoding centroid inormation of separate data blocks. VQ was applied to unstructured grid data by compressing in time instead of space [9]. The Tucker tensor decomposition [11] is an extension of principal component analysis techniques. It gathers data from different dimensions such as time steps and variables in the form of a multi-dimensional tensor. Then, it decomposes the large tensor into a set of matrices and one small core tensor. As such, Tucker compression results in compression ratios up to three orders of magnitude but it is applicable only on regular grids. Compressed sensing [3, 12] is known to be a fast data encoding technique. It works by sampling the data on the computational cluster using any sampling technique within an alternative function space such as wavelet bases. Its performance is asymmetric *i.e.* it is fast in compression but slow in decompression. It also suffers from low data reconstruction accuracy. Unlike compressed sensing where data is sampled, wavelet compression (WC) works by directly transforming the data in a spectral space and truncating the low-magnitude modes. Wavelet compression used in this paper is thus fundamentally different than the compressed sensing method used in our previous study [3]. In fact, compression using compressed sensing works by extracting samples from a data vector using sampling matrices. Such matrices are different by concept and nature than wavelets. Decompression using compressed sensing works by solving an under-determined system using a greedy algorithm such as stagewise orthogonal matching pursuit [3]. Wavelets are introduced in the decompression stage as auxilliary bases but they are not the core of the compressed method. On the other hand, wavelet compression works by transforming a data vector using a wavelet matrix operator then truncating the resulting spectrum. Wavelet decompression works by transforming the truncated data using the inverse of the same wavelet matrix operator used during compression. No greedy algorithms are used in wavelet compression which makes wavelet decompression much faster than decompression using compressed sensing. WC has been found to have a more symmetric performance with large compression ratios and good accuracy [3]. The *jpeg* image compression technique [13] falls under wavelet compression, and is ubiquitous in photography but limited to regular grid data. However, second generation wavelets and tree-wavelets (TW) are applicable to irregular grids and geometries such as those arising from unstructured meshes. In fact, the TW developed by Alpert [14] have been found in a recent study [3] to be a suitable basis for data represented on unstructured meshes. These characteristics of WC make it the optimal choice for data compression technology for large AM data. Hence, we use Alpert tree-wavelets (ATW) in this paper to directly compress AM data represented on both regular grids (such as images obtained from experiments) and on unstructured meshes (such as simulation data).

1.2 Proposed Work

In this paper we present a compression/decompression (CoDec) method based on Alpert tree-wavelets [15, 16], suitable for scientific data represented on both regular geometries such as images and irregular geometries such as unstructured meshes. Alpert WC works in the following manner. A scientific data field is written in vector form, then transformed in the wavelet domain. The obtained spectrum is truncated by eliminating all coefficients that fall below a given threshold. The remaining large coefficients are stored along with their indices in the original vector. Different thresholding techniques exist in the literature [17]. The compression and decompression mechanisms are further described in Section 2.1. We apply this compression method to additive manufacturing data: imaging data obtained from experimental measurements and computational data obtained from a finite element simulation. Our contribution in this paper consists of:

1. Developing a wavelet compression technique that works on data represented on unstructured meshes through Alpert tree-wavelets that were originally proposed for the solution of integral operators [14, 15].

2. Selecting the wavelet bases that result in maximal compression rates with minimal loss.

3. Developing an adaptive optimal wavelet thresholding scheme applicable to any field regardless of the data distribution and scale.

4. Developing error metrics that quantify the deviation from physical constraints in the data fields due to lossy compression.

The wavelet compression technique we propose is applicable to any image data or mesh-based computational simulation data, where the mesh can be either a regular grid, unstructured mesh, or particle mesh. The technique only assumes the existence of one or more data fields represented on a mesh or grid. As a test case, we consider the compression of large additive manufacturing (AM) data obtained from experimental and computational thermal/structural modeling of the sintering of a hollow cylinder from metal powders using a Laser Engineered Net Shape (LENS) process [18]. The experiments and simulations examine the large thermal gradients during the sintering process and the large residual stresses obtained in the obtained hollow cylinder. The experimental data correspond to infrared temperature images of the cylinder during its sintering process. The computational data is obtained from the coupled thermal-solid mechanics simulation of the process. The geometry is discretized using an unstructured mesh which necessitates the use of ATW for compression. We compress the data at each camera frame and simulation time step as they are produced by the infrared device and computational cluster. This data is described in more detail in Sections 3 and 4. We do not consider compression in the time domain for this study. We study the resulting compression ratio for different image and mesh fields and their associated errors due to lossy compression. We propose error metrics that assess the local and global aspects of the loss in the data and examine their relationship to the compression rate and threshold. We also study the extent to which the physical constraints in the simulation data (*e.g.* equilibrium of the stress field, divergence-free stress) are preserved after data decompression.

2 Methods and Approach

In this section, we describe the mathematical tools and algorithms used in our compression method. First, we describe how wavelet compression and decompression work. Second, we propose a new thresholding scheme used to truncate a data spectrum. Third, we present the compression and decompression algorithms. Fourth, we describe the metrics we used to quantify the error in the reconstructed data after lossy compression.

2.1 Wavelet Compression

Wavelets consist of multi-resolution bases suitable to represent any function in a spectral domain [19]. Let $\boldsymbol{f}$ be a vector of dimensionality N representing a data field, *e.g.* temperature, color intensity, etc. Here N can be either the number of pixels in an image, the number of nodes for node-based fields, or elements for element-based fields in a finite element or volume mesh. In wavelet compression, $\boldsymbol{f}$ is transformed into a spectrum $\boldsymbol{w}_f$ using a matrix operator $[\Psi]$ consisting of functional bases following:

$$\boldsymbol{w}_f = [\Psi] \cdot \boldsymbol{f} \tag{1}$$

This transformation usually takes place in a matrix-free fashion in images [19] due to the absence of any mesh and because the data is spatially contiguous consistent with the vector $\boldsymbol{f}$. However, matrix-free transformation is not straightforward for irregular grids, although not impossible, because it requires complicated operations at the level of each irregular mesh node/element [20]. Hence, we propose in our study to build the full wavelet operator $[\Psi]$ and apply it to any data vector $\boldsymbol{f}$. This is an expensive exercise, but it is performed only once for fixed meshes and grids at the beginning of data acquisition and simulation. The cost is amortized over the number of time steps and data fields present in the dataset [3].

Building $[\Psi]$ according to ATW is described in [15] and outlined in our recent study [3]. The major advantage of ATW is that their computation does not require any special treatment of the domain boundaries or any irregularities present in the domain and they avoid them by construction [16]. The discrete Alpert wavelets can use any basis functions, *e.g.* polynomials, sines, cosines, etc.

Compression takes place in the following manner. Starting from the vector $\boldsymbol{f}$ of size N, we perform the operation in Eq. 1. We refer to the vector $\boldsymbol{w}_f$ as the wavelet coefficients vector. Among these wavelet coefficients we only store the ones of amplitude larger than a given threshold θ along with their indices. The choice of the threshold θ is discussed in Section 2.4. Let M be the number of the stored wavelet coefficients. For a given data vector of size N, the compression ratio is defined as:

$$R = \frac{N}{M} \tag{2}$$

2.2 Wavelet Decompression

Let $\mathcal{I}$ be a vector of size M containing the indices of the large stored wavelet coefficients. Wavelet decompression takes place though an inverse wavelet transform on the compressed vector $\boldsymbol{w}_{f,\mathcal{I}}$ following:

$$\hat{\boldsymbol{f}} = [\Psi_{\mathcal{I}}]^T \cdot \boldsymbol{w}_{f,\mathcal{I}} \tag{3}$$

The resulting vector $\hat{\boldsymbol{f}}$ of size N is an approximation of the original data vector $\boldsymbol{f}$ due to lossy compression. In Section 2.6 we describe error metrics to quantify this loss. Note also that when reconstructing data from the wavelet compressed version, one can be selective in choosing the set of points where data should be reconstructed. This can be done by choosing the required columns in the wavelet matrix $[\Psi_{\mathcal{I}}]$.

2.3 Wavelet Basis Functions

One of the main steps in the construction of the wavelet matrix $[\Psi]$ is the choice of the basis functions used in computing the initial moment matrices $[M] \in \mathbb{R}^{n \times w}$, where n is the number of mesh points in a given bin and w is the given wavelet order [3, 15, 16]. Each element of $[M]$ is given by:

$$M_{\alpha,\beta} = \prod_{d=1}^{D} \psi_{1 \leq \beta_d \leq w_d}(x_{d,\alpha}), \tag{4}$$

where x_d and w_d are the mesh coordinates and wavelet order in the dimension d, respectively. The function ψ can be any basis function, but it plays an important role in assessing how much the resulting wavelet matrix $[\Psi]$ can encode information in the transformed data. We compare in Section 4.2 the results of the wavelet compression for different functions ψ in terms of ability to compress data and the resulting error in the reconstructed data.

Compared to traditional wavelets, *e.g.* Haar, Daubechies, biorthogonal, etc. [21], ATW exhibit larger smoothness since they are usually constructed from polynomials [22]. At zero order, ATW coincide with Haar wavelets on regular grids. However, unlike traditional wavelets, they allow discontinuities in order to accommodate irregular boundaries.

2.4 Thresholding Method

Several thresholding methods exist in the literature [17]. These have been proposed for wavelet denoising, which is a special case of wavelet compression. In fact, compression can be thought of as the attenuation of the high frequency components of the data spectrum, which is a form of data noise reduction. We adopt in our compression method the universal Donoho *hard-thresholding* technique [21] where all wavelet coefficients of amplitude below a given threshold θ are eliminated after the Alpert wavelet transform which induces the intended compression.

There are several methods for estimating the value of the universal hard-threshold θ [23]. The threshold based on the median absolute deviation is usually used in wavelet noise filtering [21]. However, our compression method uses the *mean absolute deviation* technique to estimate θ since we found it to be more robust *i.e.* it results in most cases with reasonable compression ratios and errors. For a data field $\boldsymbol{f}$ that we intend to compress, we define θ_f as:

$$\theta_f = \frac{\theta_0}{0.6745} \cdot [\max(\boldsymbol{f}) - \min(\boldsymbol{f})] \cdot \frac{\text{mean}\left(\Big| |\boldsymbol{w}_f| - \text{mean}(|\boldsymbol{w}_f|) \Big|\right)}{\text{mean}(|\boldsymbol{w}_f|)} \cdot \sqrt{2 \log(N)} \tag{5}$$

where θ_0 is a threshold multiplier used to vary the overall θ_f according to the required error tolerance, 0.6745 is a scaling factor, $\boldsymbol{w}_f$ is the transformed field using Alpert wavelet wavelets (see Eq. 1), and the last term accounts for the data size N as suggested in the Donoho universal threshold [23].

In order to obtain a more general threshold, we have introduced two additional terms in the threshold expression. The first term is a global measure of the transformed data where θ_f is divided by the average of $|\boldsymbol{w}_f|$. The second term is a global measure of the data $\boldsymbol{f}$ itself where θ_f is multiplied by the maximum value of $\boldsymbol{f}$ minus its minimum value. The reason behind the choice of the second term is our assumption that the major features in the data that have priority to be preserved after compression exist where the data values are large in amplitude. Hence, we choose to penalize the truncated wavelet coefficients by the extreme range in the data, that is the [max(·) - min(·)] measure.

2.5 Compression Algorithm

In HPC simulations based on domain decomposition methods, the data is distributed among many cores/processes. In other words, the mesh is subdivided among the computing processes such that they act on an almost equal number of mesh points. This is the case of the AM simulation we consider in this paper. Accordingly, our wavelet compression method acts locally in each process on the mesh points and corresponding data fields contained therein, independently of the mesh owned by other processes. This allows a communication-free compression resulting in a highly scalable compression method. The compression is performed *in-situ* as the fields are computed at each time step. The data reconstruction also takes place locally and operates on the compressed data in each process. Large images can be compressed in a similar fashion, by dividing them into smaller images, e.g. according to their acquisition channels.

One shortcoming of such local compression is that discrepancies might arise at boundaries separating two adjacent processes where compression is independently performed. These discrepancies could be visually noticeable. However, they are usually of similar magnitude to the errors from lossy compression inside the domain. An adaptive thresholding method like the one we adopt in this paper minimizes these boundary anomalies and results in consistent data loss over the entire domain.

The compression algorithm proceeds in the following manner. At the beginning of the simulation, the Alpert wavelet matrix operator $[\Psi]$ is computed in parallel in each process along with other initializing simulation variables. At each time step, $[\Psi]$ is first multiplied by the vector fields present in the simulation. Second, thresholding of the resulting vectors is performed where the indices of the dominant modes are noted. Third, the truncated modes along with their indices are stored for each vector field. More details about these steps are given in Algorithm 1.

The truncated modes typically exhibit smooth variation in time. Hence, at the end of the simulation, this data can be also be compressed in the time dimension. Finally, all resulting truncated modes and indices can be further compressed using a lossless method such as *bzip2*.

Algorithm 1 The parallel Alpert wavelet compression procedure within each computing process.

Compute the Alpert wavelet matrix $[\Psi]$ according to the local mesh
for $t = 1$ until end of the simulation **do**
  for all simulation vector fields $\boldsymbol{f}_v$ **do**
    $\boldsymbol{w}_{v,t} = [\Psi] \cdot \boldsymbol{f}_{v,t}$
    Compute the threshold $\theta_{v,t}$
    Find $\boldsymbol{J}_{v,t} = \{j : |w_{j,v,t}| > \theta_{v,t}\}$
    Store $\boldsymbol{w}_{\boldsymbol{J}_{v,t}}$ and $\boldsymbol{J}_{v,t}$
  end for
end for

Wavelet decompression, described in Algorithm 2, takes place by reversing the order of the compression steps. Here the main operation is the multiplication of the truncated transposed matrix $[\Psi]$ by the compressed vector $\boldsymbol{w}_{\boldsymbol{J}}$. If all the columns of $[\Psi]$ are used in this multiplication, then the reconstructed field $\hat{\boldsymbol{f}}$ is obtained at all mesh nodes. If only a subset of this field is required to be reconstructed *i.e.* in selective decompression at specific mesh nodes, then only the columns in $\Psi_{\boldsymbol{J}}$ corresponding to the region of interest are used.

Algorithm 2 The parallel Alpert wavelet decompression procedure within each computing process.

Compute the Alpert wavelet matrix $[\Psi]$ according to the local mesh
for $t = 1$ until end of the simulation **do**
  for all simulation vector fields $\boldsymbol{f}_v$ **do**
    Read $\boldsymbol{w}_{\boldsymbol{J}_{v,t}}$ and $\boldsymbol{J}_{v,t}$
    $\hat{\boldsymbol{f}}_{v,t} = [\Psi^T_{\boldsymbol{J}_{v,t}}] \cdot \boldsymbol{w}_{\boldsymbol{J}_{v,t}}$
  end for
end for

2.6 Error Metrics

Lossy compression such as WC results in errors in the decompressed version of the data. Quantifying this error is crucial in engineering applications. Several generic error metrics exist in the literature. Additional error metrics may be developed by application specific data analysts depending on the given data and accuracy requirements. When analyzing decompressed data, one has to consider more than one error metric [3] in order to cover different aspects of the data loss [24] such as the visual, global and local aspects. For example, the visual reconstruction of a data field might be satisfactory but errors can still be present either throughout the field or localized in some regions. Moreover, some data fields are required to obey physical constraints expressed by mathematical constraints *e.g.* divergence-free stress field. It is also crucial to understand the magnitude of the deviation from such constraints due to lossy compression. In this study, we compare the visual representation of several fields in the additive manufacturing simulation and report one global and one local error metric. We also devise a method to quantify the deviation from physical constraints.

2.6.1 Global and Local Error Metrics

For error metrics intrinsic to the data, we assume that values in the data fields of large amplitude constitute the important features in the data and thus have a higher priority to be preserved. This assumption stems from the fact that the goal of engineering simulations is often to obtain extreme values such as maximum temperature, maximum tensile stress, maximum deformation, etc. Thus, we use the maximum and minimum values in the error metrics.

We assess the global data reconstruction error using the mean squared error normalized by the data size and by the difference between the maximum and minimum value in the data. We refer to this metric as the normalized relative mean squared error (NRMSE). The NRMSE between a dataset given by a field $\boldsymbol{f}$ and defined on N points, and its reconstructed version $\hat{\boldsymbol{f}}$ is computed as:

$$NRMSE = \frac{||\boldsymbol{f} - \hat{\boldsymbol{f}}||_2}{\sqrt{N}\,[\max(\boldsymbol{f}) - \min(\boldsymbol{f})]} \tag{6}$$

For the local error, the most straightforward choice of error would be the local relative error at each mesh point in a field. However, data field values $|f_i|$ often approach zero values making the relative error $|f_i - \hat{f}_i|/|f_i|$ undefined. Instead we define the local error as:

$$e_{a,f_{1\leq i\leq N}} = \frac{|f_i - \hat{f}_i|}{[\max(\boldsymbol{f}) - \min(\boldsymbol{f})]} \tag{7}$$

As discussed earlier, the rationale behind the choice of the [max(·) - min(·)] as a normalization factor is that we assume that the major feature data fields are contained in the values in the data that are large in amplitude, thus we penalize the values that are low in magnitude.

2.6.1.1 Remarks:

1. While the NRMSE error metric is a global measure of the overall error throughout the computational domain, the local error metric is evaluated at each mesh point in the domain and constitutes a vector of numbers. In order to summarize and report the local error metric, we plot its statistical distribution similar to Figure 13.

2. The way we define the threshold function and error metrics enables flexibility in the choice of the penalization factor. In cases where the maximum and minimum are not the major features in the data, the [max(·) - min(·)] normalization factor can be replaced by another factor in Eqs. 5-7. For example, in the case of solidification, it is important to preserve an accurate reconstruction of the temperature at the solid-liquid interface where temperature does not reach its minimum or maximum value. Thus, [max(·) - min(·)] range can be reduced to a smaller one around the value of interest. In solidification, this range is around the solidus to liquidus temperatures.

2.6.2 Deviation from Physical Constraints

Physical constraints such as the divergence-free velocity and stress constraints have to be satisfied in the solution of any partial differential equation. In this paper, we address the divergence-free stress constraint present in solid mechanics simulations. Let $[\sigma]$ be the local 3D stress tensor in the computational domain. The divergence-free physical constraint is expressed as:

$$\nabla \cdot [\sigma] = \mathbf{0} \tag{8}$$

Practically, finite element simulations enforce this constraint weakly, meaning that the obtained $\nabla \cdot [\sigma]$ is usually a very small number close to zero. After compression and reconstruction, errors are introduced in $[\sigma]$ and the divergence of the reconstructed stress tensor $[\hat{\sigma}]$ is significantly increased. We relate this non-zero divergence of $[\hat{\sigma}]$ to the metric e_{div} quantifying the deviation from divergence-free physical constraint, where the normalized e_{div} as:

$$e_{div} = \frac{||\nabla \cdot [\hat{\sigma}] - \nabla \cdot [\sigma]||_2 \cdot \bar{h}}{||[\sigma]||_F} \tag{9}$$

where $\bar{h}$ is the local mesh size and $||[\sigma]||_F$ is the Frobenius norm of the local stress tensor. We chose this latter as the normalization factor unlike e_a and NRMSE because the $[\max(.) - \min(.)]$ is not meaningful for the stress divergence. Eq. 9 subtracts the residual divergence in the original stress field due to the weak enforcing of the constraint. In practice, $\nabla \cdot [\sigma]$ and $||[\sigma]||_F$ are calculated during the simulation, saved as meta-data and used in the decompression stage to compute e_{div}. $\nabla \cdot [\hat{\sigma}]$ is computed after decompression from the reconstructed $[\hat{\sigma}]$. Similar expressions can be written for other free-divergence fields such the velocity field in an incompressible flow simulation.

3 Example: Additive Manufacturing Data

In metal additive manufacturing, the material deposition process involves melting metal powder feedstock with a heat source, such as a laser or electron beam. One method of metal AM is the Laser Engineered Net Shape (LENS) process where metal powder is blown into a laser beam through nozzles using a carrier gas. Upon entering the beam, the powder is melted and added to the part. The material is deposited onto a metal baseplate. Due to the highly localized heating caused by the laser, large thermal gradients arise and large residual stresses can develop in the final part [18].

3.1 Experimental Data

The experimental data used in this study consisted of infrared image (IR) data taken during the LENS build of a 304L stainless steel cylinder produced through the LENS process. The additively manufactured cylinder had a 50 mm height, a 24 mm outside diameter, and a 4 mm wall thickness. The cylinder was deposited onto a baseplate with dimensions of 152 x 152 x 6.35 mm. Because the baseplate is clamped to the movable base in the LENS process, the bottom surface of the baseplate was fixed in the simulations and serves as the only static boundary condition (see Figures 3-5). The laser diameter and speed were 4 mm and 84.6 μm/s, respectively, and the layer thickness was 0.9 mm. The images were collected using a FLIR SC6700 mid-wave IR camera (640×512 pixel, InSb FPA). The data was taken at a frequency of 21 Hz and produced approximately 20 GB of data over the course of the build. The temperature values were generated by the internal FLIR software using an emissivity of 0.3 and exported as TIFF files (see Figure 1). For a more thorough explanation of the experimental and numerical setup, see [18].

3.2 Simulation Data

A thermal-structural simulation of a 304L stainless steel cylinder produced using the LENS process was investigated. Using the finite element method, the thermal and structural simulations were performed in the Sierra multiphysics finite-element software suite in a one-way coupled manner, where the thermal results serve as input to the mechanical analysis. Consequently, the mechanical analysis had no effect on the thermal history. In the thermal simulations, a Gaussian laser heat source was scanned across a pre-defined finite element mesh, and elements were activated when their temperature reached the melting temperature of the material. Conduction, convection, and radiation boundary conditions were modeled in the thermal simulation. Once the thermal simulation was complete, the resulting temperature profile at every time step was used in the structural simulation. In a similar fashion, elements were activated upon reaching melt temperature. Residual stresses developed upon cooling to room temperature due to incompatible thermal strains. A temperature dependent viscoplastic internal state variable material model was used to model the material response. This material model is a simplified version of that developed by Brown and Bammann [25].

For a more thorough explanation of the experimental and numerical setup, see [18].

4 Compression Results

In this section, we present results from the application of Alpert wavelet compression to the AM experimental and simulation data presented in Section 3. All reported results of Alpert wavelet compression are obtained using the thresholding method expressed in Eq. 5. The compression level is set by choosing a value for θ_0.

4.1 Time and Memory Costs

Wavelet compression incurs additional costs of computational time and memory to the original simulation. In a previous study [3], we found that wavelet compression and decompression speeds are about to 50 and 20 MB/s/process, respectively, on a machine running on a 3.0 GHz CPU. These speeds do not account for the memory bandwidth and I/O rates but they scale infinitely with the number of processes since compression and decompression occur locally in each process. Decompression is slower because the stored wavelet coefficients indices are not contiguous and cause slower memory access in the wavelet matrix.

As discussed in Section 2.1, our compression technique starts by computing and storing the wavelet matrix $[\Psi]$. The time spent to compute $[\Psi]$ is not usually a major concern since it occurs only once in the whole simulation. However, $[\Psi]$ has to be stored in the system memory. Using standard fitting techniques on meshes of different sizes N and different wavelet orders w, we have derived the following correlation that estimates the memory storage cost in MB of $[\Psi]$:

$$\mathcal{S} = 5.781 \times 10^{-5} N^{1.1507} w^{1.6935} \qquad \text{(MB)} \tag{10}$$

4.2 Choice of Wavelet Basis Function

We first study the effect of the wavelet bases ψ (see Section 2.3) on the wavelet compression performance. We consider the basis functions given in Table 1 and obtained from [26]. These basis functions differ in their ability to encode field data. For example, if the data field is oscillatory, trigonometric or Bessel functions would be more suitable for the wavelet transform. However, one has also to consider the computational cost of such functions in practice. Each of the basis functions is used to build the Alpert wavelet matrix $[\Psi]$ as described in [3, 15, 16] and according to Eq. 4. The resulting compression ratio and NRMSE are calculated. We have selected three data fields from the simulation data as test cases: the total displacement U, the temperature T, and the von Mises stress σ_{VM} all recorded at the last time step.

For a given threshold θ_0, we would like to find the basis function ψ that results in the largest compression ratio R and lowest error NRMSE in the reconstructed data. Hence, we evaluate for each ψ the ratio between R and NRMSE. Table 1 shows that monomials, Tchebyshev and Gegenbauer bases maximize this ratio, most probably because the fields we chose are not oscillatory and are better represented with such polynomial bases. Based on these results, we choose ψ in the rest of this paper as the monomials bases due to their simplicity and low computational cost.

Table 1: The ratio between the compression ratio R and error measure NRMSE for different fields in the simulation. Results are generated after wavelet compression with adaptive thresholding where $\theta_0 = 0.0075$ and the indicated types of functions are used in constructing the wavelets.

	Sine	BesselI	Monomials	Tchebyshev	Gegenbauer	Jacobi	Laguerre	Legendre	Hermite
U	521	317	**42564**	**42451**	**42584**	23272	38853	4969	3421
T	297	257	**39195**	**37198**	**39357**	16936	17697	12907	7637
σ_{VM}	631	333	**1301**	**1304**	**1298**	889	1148	1156	1103

4.3 Experimental Data

We select one snapshot in time of the IR temperature measurement of the LENS AM process described in Section 3.1. The image consists of 640×512 pixels. Building the wavelet matrix $[\Psi]$ and performing the compression for such a large data vector is computationally intractable. Thus we have to subdivide the image into smaller data vectors. We choose to subdivide it into $4 \times 8 = 32$ blocks such that each one consists of $N = 10240$ pixels. We construct the matrix $[\Psi]$ with a wavelet order $w = 5$. Based on Eq. 10, the storage of the $[\Psi]$ matrix was approximately 37 MB in system memory. The image data consists of

3 color fields that we compress and reconstruct separately.

We first study the effect of the compression rate on the visual quality of the reconstructed image as shown in Figure 1. By increasing the threshold θ_0 we increase the compression ratio R. The reconstructed images are then plotted; they are visually indistinguishable from the original image for R up to approximately 124. Within this range, we do not expect significant artificats at the boundaries separating the 32 blocks thanks to the adaptive thresholding method we adopted. The visual quality significantly deteriorates for $R > 124$. However, the major qualitative features in the images, such as the hot spot where melting takes place, the overall temperature gradient and the cylinder borders, are still preserved even at large compression ratios.

Although other image compression schemes such as JPEG-2000 can result in similar compression ratios, the compressed data they produce are either encoded or quantized and are not useful for further data analysis such as parameter estimation [27, 28]. However, our current wavelet compression method results in a vector of coefficients $\boldsymbol{w}_{\mathcal{I}}$ which, even in its compressed form, can be compared numerically with a compressed version of other datasets for data analysis and further post-processing.

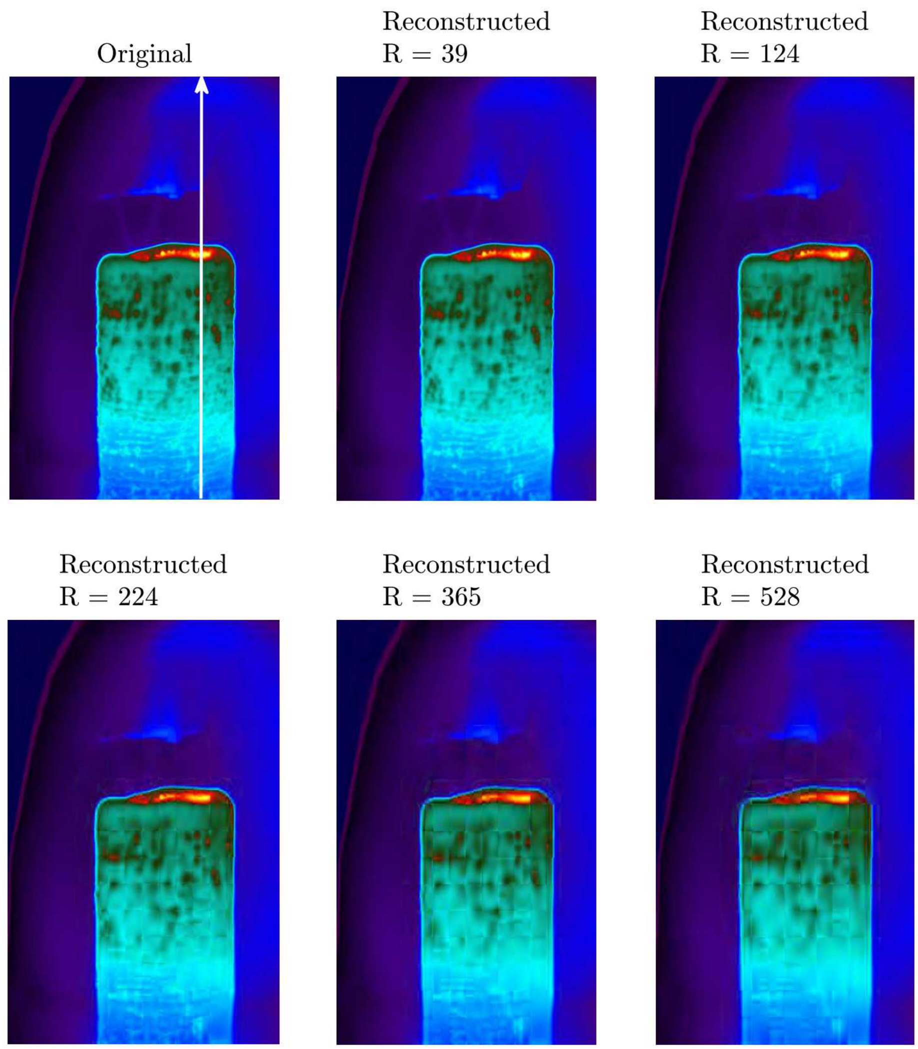

Figure 1: Color images of the original infrared experimentally measured temperature field at a given time frame (top-left) along with its reconstructed version after wavelet compression at different compression ratios R, as indicated. The upward white arrow in the original image designates the slice where the color intensity is reported in Figure 2. The images are not to scale.

Often in data visualization, scientists are only concerned about the results at one or a few slices in a given dataset. For example, in the IR image considered in this paper, one would be interested mainly in the temperature distribution at a section crossing the cylinder in order to study the temperature gradient during the AM process. Hence, we plot the intensity distribution through the white arrow shown in Figure 1 (top, left) and compare it with the one obtained from the reconstructed data. The intensity is the sum of the RGB color fields in the image, it is different than the temperature field. For a compression ratio $R = 118$, there is an excellent agreeement between the original and reconstructed intensities as shown in Figure 2. This is consistent with the observation on the full color plots shown in

Figure 1. For increasing R, the agreement is still preserved mainly for the major features in the curves such as the peak at approximately 375 pixels. Larger values of R values ($R > 1000$) result in smoothing of the intensity field as shown in Figure 2 (bottom) where the reconstructed intensity curve is smooth relative to the oscillatory original curve. This result is consistent with the well-known smoothing and denoising properties of wavelets [17].

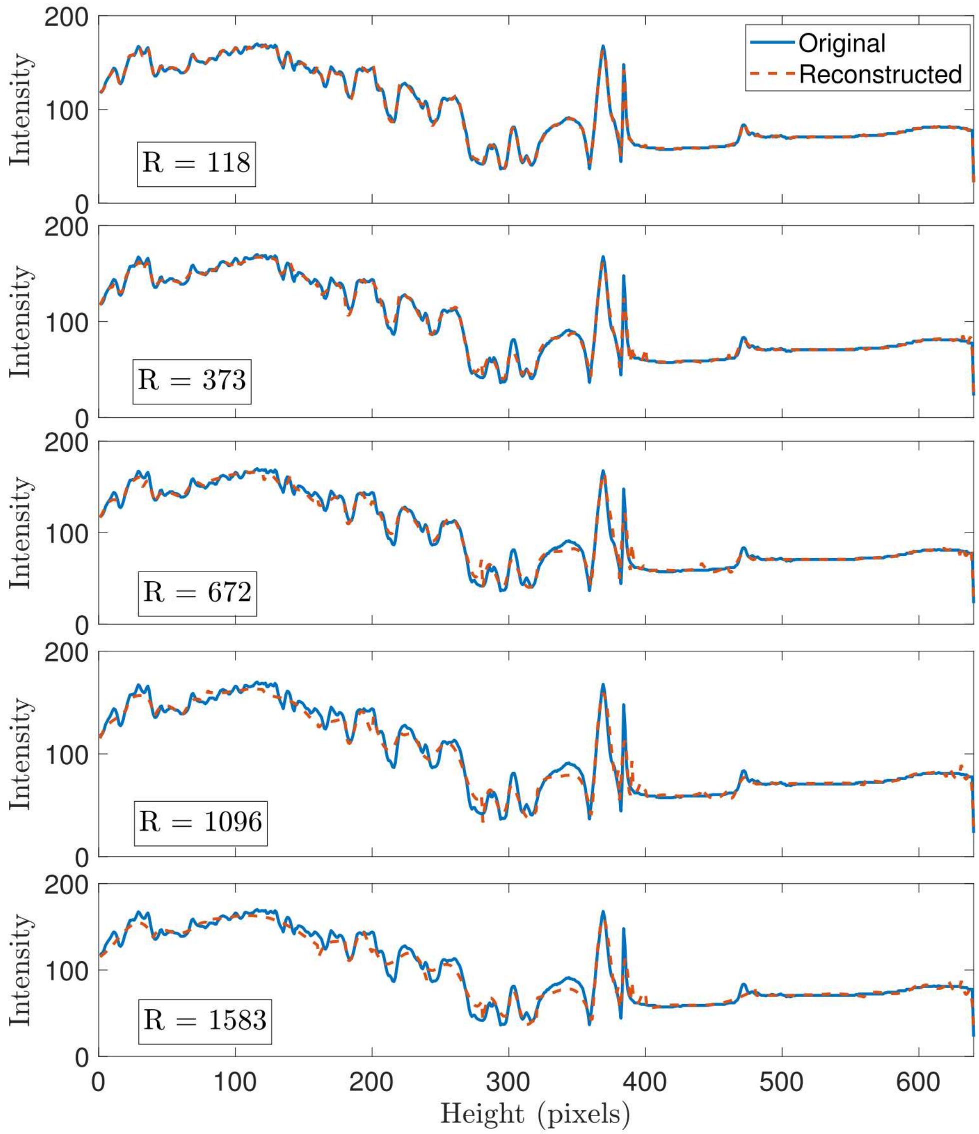

Figure 2: Plots showing the color intensity along the height of the cylinder at the location of the white arrow shown in Figure 1 (top-left). Shown are the original (solid blue curves) and reconstructed (dashed red curves) intensities after compression at different compression ratios R, as indicated.

4.4 Simulation Data

The AM simulation was performed on a parallel cluster using 32 processes. The computational mesh was unstructured and consisted of approximately 800,000 hexahedral elements which resulted in about 500,000 mesh nodes. There were 17 different fields represented on the unstructured mesh and reported for 11,000 time steps. This resulted in a large dataset of approximately 1.5TB in size. We constructed two matrices $[\Psi]$, for the element and node based fields, using a wavelet order $w = 5$. The two matrices occupied approximately 160 MB in the system memory according to Eq. 10.

4.4.1 Visual Comparison between Original and Reconstructed Data

Similarly to Section 4.3, we start by investigating the visual aspect of reconstructed data fields. Figures 3-5 show different fields at different times in the simulation. These fields exhibit different trends through the cylinder. The original and reconstructed data fields are visually indistinguishable but the achieved compression ratios differ significantly between the fields. Similar to the experimental data, we do not find significant artificats at the boundaries separating the 32 subdomains thanks to the adaptive thresholding method we adopted. A relatively low value of the threshold multiplier θ_0 guarantees accurate reconstruction throughout the subdomains. We can verify that no artifacts occur at a boundary separating two adjacent subdomains by comparing the local error metric (see Eq. 7) in the regions on the two sides near the boundary. If these errors are of the same order of magnitude then it is unlikely to find an artifact when plotting the field. On the other hand, a large difference in these local errors relative to the error away from the boundary signifies a visual artifact in the plot. As mentioned in Section 3.2, the AM process works by building the cylinder as powder metal melts and then solidifies. The temperature is largest at the laser melt pool as shown in Figure 3 (top and middle). The cylinder region above the hot spot does not exist physically but the mesh exists in the simulation. However, it is only active in the computations after being swept by the AM process. A constant temperature value is assigned in this inactive region such that its data is maximally compressed. The large temperature gradients occurring around the laser spot can be clearly seen in the middle plot of Figure 3. The compression ratio is locally low in such regions (not reported). Nevertheless, the overall compression ratio is large, on the order of 100. At the end of AM process, the heating is stopped allowing heat to diffuse throughout the cylinder leading to substantially lower gradients and a more regular temperature distribution as shown in the bottom of Figure 3. This results in the larger compression ratio of $R = 672$.

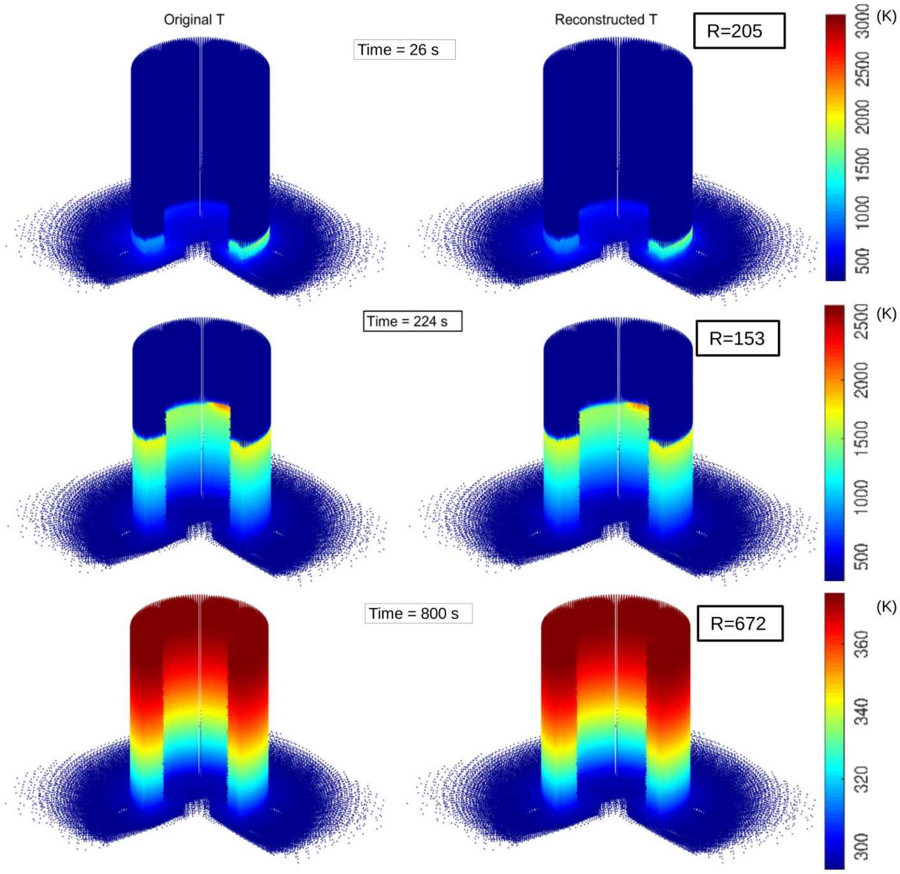

Figure 3: Plots showing the original and reconstructed temperature fields at three different times during the AM simulation, as indicated. Reconstructed results are generated after wavelet compression with adaptive thresholding where $\theta_0 = 0.0075$, which results in the indicated compression ratios R.

The solid mechanics simulation results in the displacement fields shown in Figure 4. These are smoother than the temperature fields during the sintering process. The displacement values above the laser spot have no physical meaning but they are generated and stored by necessity in the finite element simulation and the data storage format. Thus, it is not surprising that large compression ratios of approximately 100 are obtained.

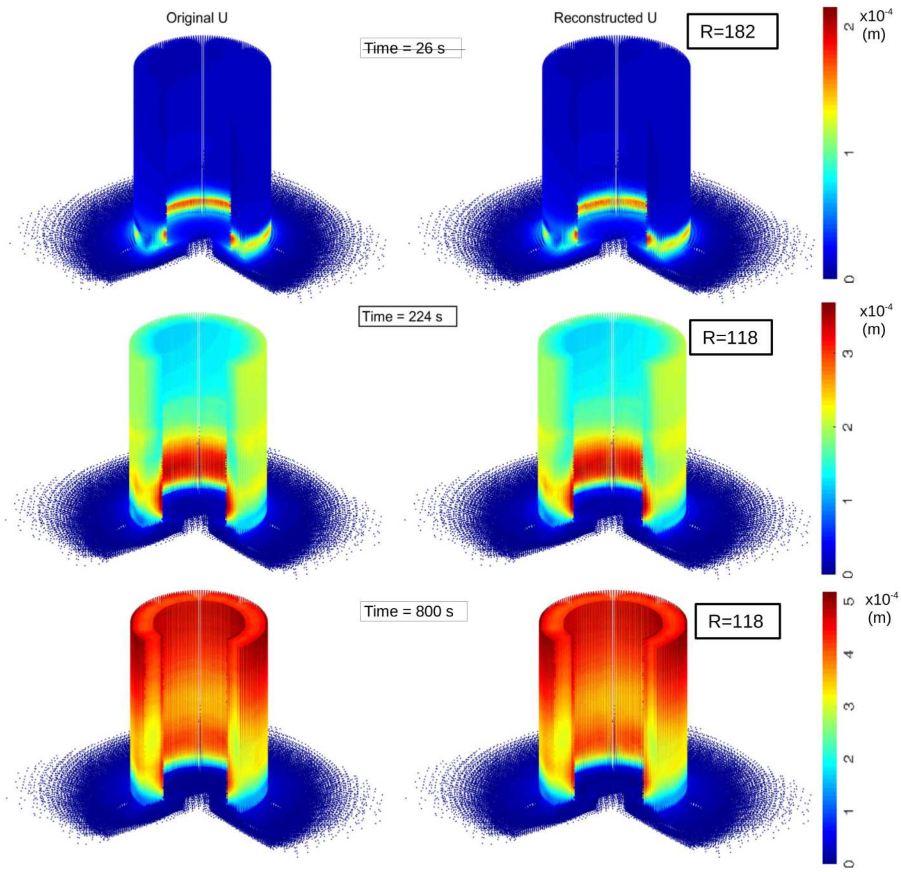

Figure 4: Plots showing the original and reconstructed displacement fields at three different times during the AM simulation, as indicated. Reconstructed results are generated after wavelet compression with adaptive thresholding where $\theta_0 = 0.0075$, which results in the indicated compression ratios R.

The evolving von Mises stress field during the AM simulation is shown in Figure 5. It exhibits large gradients at the cylinder base at all three times during the AM process. Moreover, these stresses follow a complicated, non-smooth trend due to the complex coupled thermal and structural physics involved in the finite element model. Therefore, it is expected that large compression ratios cannot be achieved, unlike the temperature and displacement fields shown in Figures 3-4.

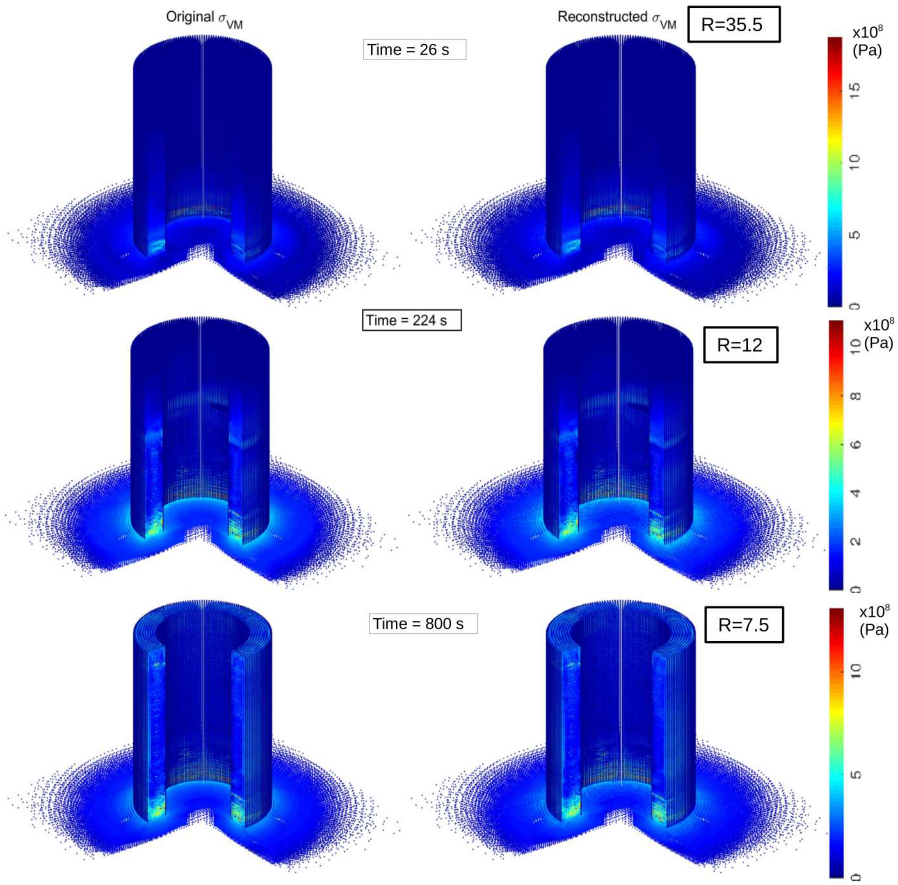

Figure 5: Plots showing the original and reconstructed von Mises stress fields at three different times during the AM simulation, as indicated. Reconstructed results are generated after wavelet compression with adaptive thresholding where $\theta_0 = 0.0075$, which results in the indicated compression ratios R.

Instead of looking at the full three-dimensional dataset, it is often beneficial and more informative to look at the variations in the fields in one-dimensional slices of the subdomain. We report results from slices along the cylinder height for the temperature and von Mises stress in Figures 6 and 7, respectively, for several time steps. There is an excellent agreeement between the original and reconstructed versions. Some noise is introduced in the reconstructed temperature fields around the region of large gradients. Avoiding this behavior requires either larger wavelet orders or lower compression ratios (lower θ_0) in order to enhance the accuracy of the reconstruction.

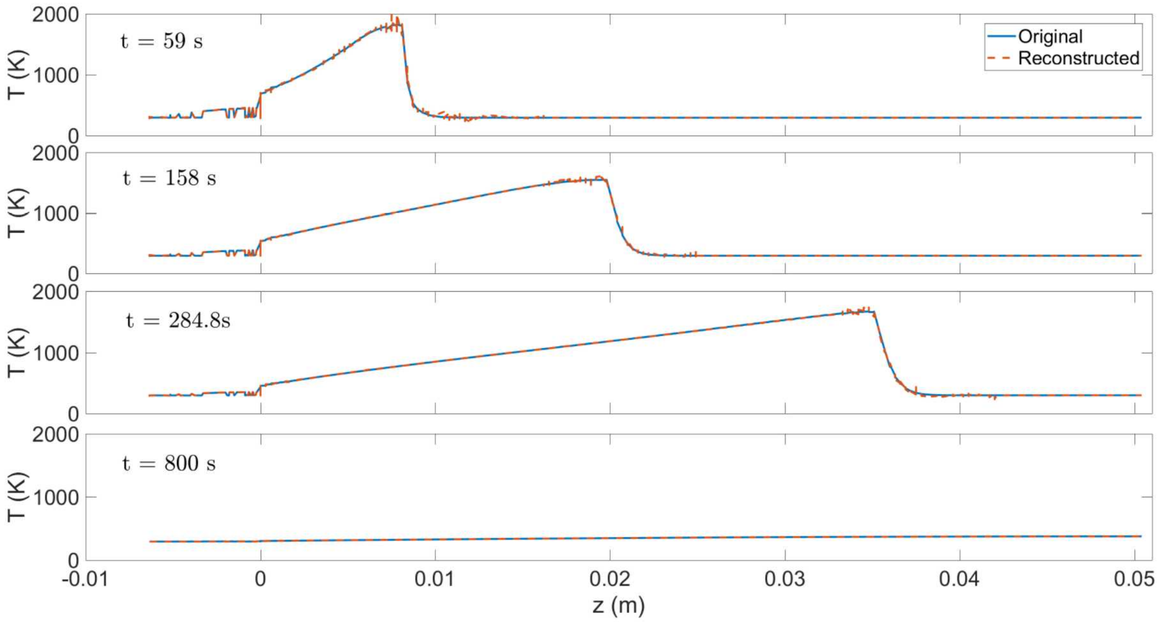

Figure 6: Plots showing the temperature variation along the z-direction at one vertical section in the cylinder at different time steps, as indicated. Shown are the original values along with the ones generated after wavelet compression with adaptive thresholding where $\theta_0 = 0.0075$.

The von Mises stress field reveals substantial oscillations that were hidden in the plots of Figure 5. These are physical oscillations in the field, yet they are accurately reconstructed at all time steps in the simulation. The oscillations cause large localized gradients throughout the cylinder and result in discrepancies in the reconstructed von Mises stress. Nevertheless, the reconstruction is still accurate relative to the maximum von Mises stress which occurs close to the cylinder base ($0 < z < 0.01m$).

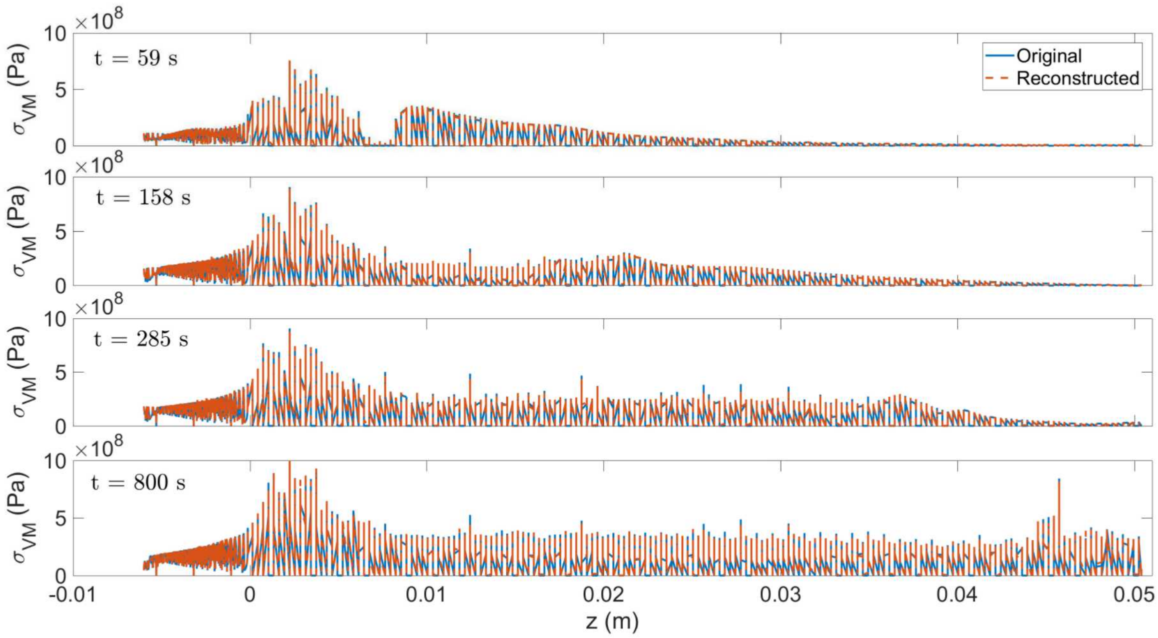

Figure 7: Plots showing the von Mises stress variation along the z-direction at one vertical section in the cylinder at different time steps, as indicated. Shown are the original values along with the ones generated after wavelet compression with adaptive thresholding where $\theta_0 = 0.0075$.

4.4.2 Global Compression Ratios and Error

We now quantify the error due to lossy wavelet compression of the AM simulation data. We first examine the NRMSE global error metric defined in Eq. 6 and report it as a function of time along with its corresponding compression ratio R using a fixed θ_0 for all data fields. The results are shown in Figures 8-10.

Figure 8 shows the evolution of R and NRMSE of the temperature and spatial displacements as a function of time. Despite the large temperature gradients observed in Figure 3, the temperature field achieves large R mainly due to the assumed constant value in the region above the laser spot. After about 400 seconds, the cylinder is complete and the temperature gradients are relaxed. This is reflected in the increased R later in the simulation. In the same time, NRMSE is decreased even though R increases. The error in u_z exhibits the opposite trend. This signifies that with the thresholding technique we use with fixed θ_0, the error does not necessarily increase with larger R as one would intuitively expect.

For the displacement fields, the compression of u_x and u_y produce very similar results, while the R of u_z is substantially lower. This can be explained by the fact that the cylinder height is growing in the z-direction where thermal expansions are dominant and the displacement gradients are large resulting in the relative decrease of R.

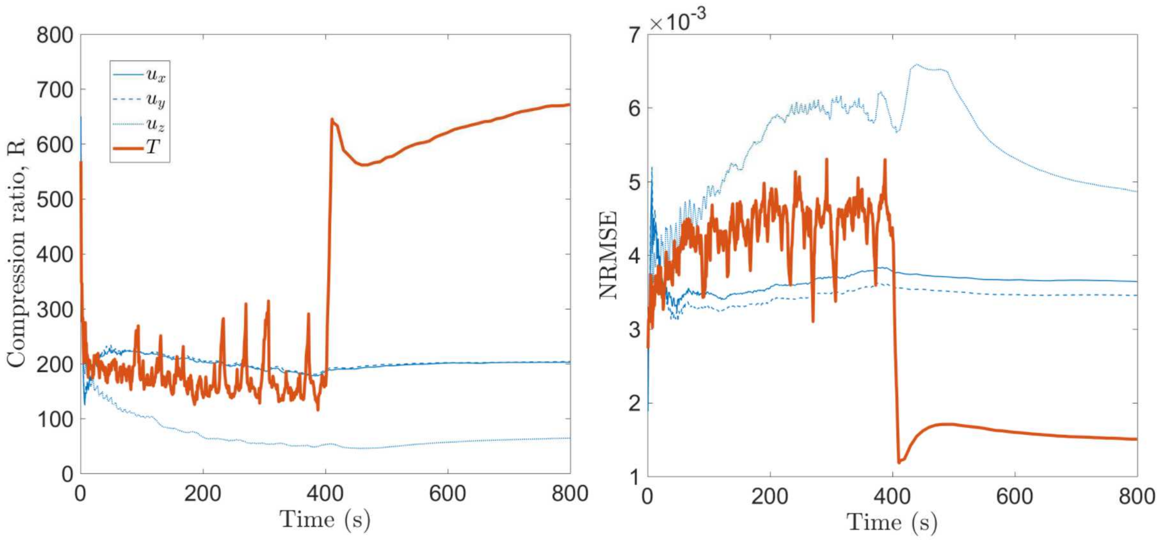

Figure 8: Plots showing the evolution of (left) the compression and (right) the NRMSE error obtained in the wavelet compression of the spatial displacements and temperature fields as indicated by the different curves. Results are generated after wavelet compression with adaptive thresholding with $\theta_0 = 0.0075$.

The strain fields are derivatives of the spatial displacements. These derivatives depend on more than one displacement field, hence they contain more information. Therefore it is not surprising that much lower compression ratios are achieved than the displacements as shown in Figure 9. All 6 strain fields have similar trends and amplitudes in R and NRMSE. The stress components ϵ_{zz}, ϵ_{yz} and ϵ_{zx} have the lowest R values most likely due to the larger gradients in u_z.

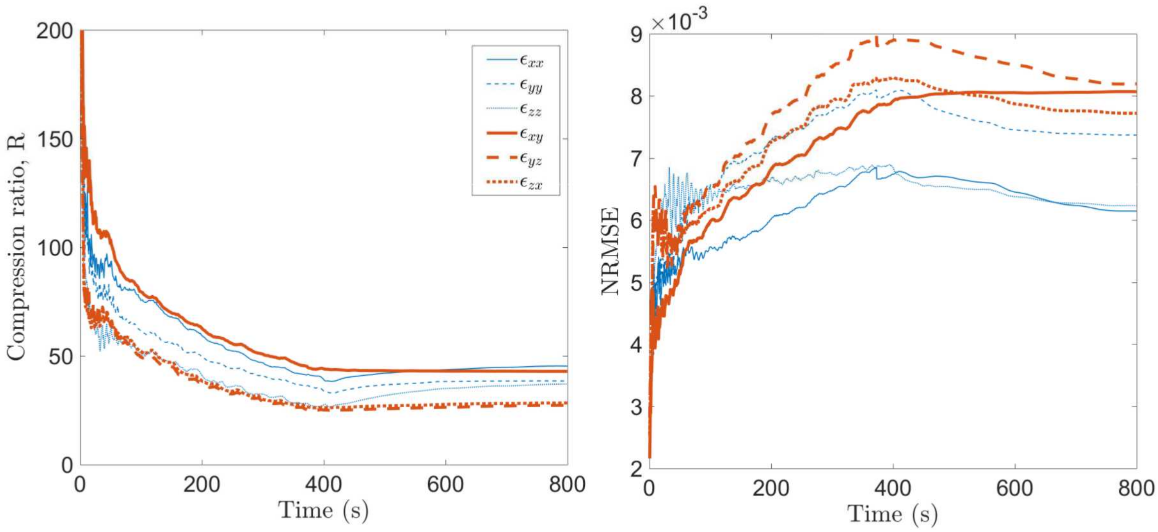

Figure 9: Plots showing the evolution of (left) the compression and (right) the NRMSE error obtained in the wavelet compression of the strain components. Results are generated after wavelet compression with adaptive thresholding with $\theta_0 = 0.0075$.

Figure 10 shows compression results for the six components of the stress tensor. While the tensile and compressive stress components admit large values of R, the shear components and the von Mises stress cannot be significantly compressed, resulting in a larger error approximately twice the error in the tensile components. The von Mises stress is a function of the shear components, which according to the results in Figure 5 exhibit large gradients and complicated trends. This lowers their compressibility similar to the case of u_z in Figure 8.

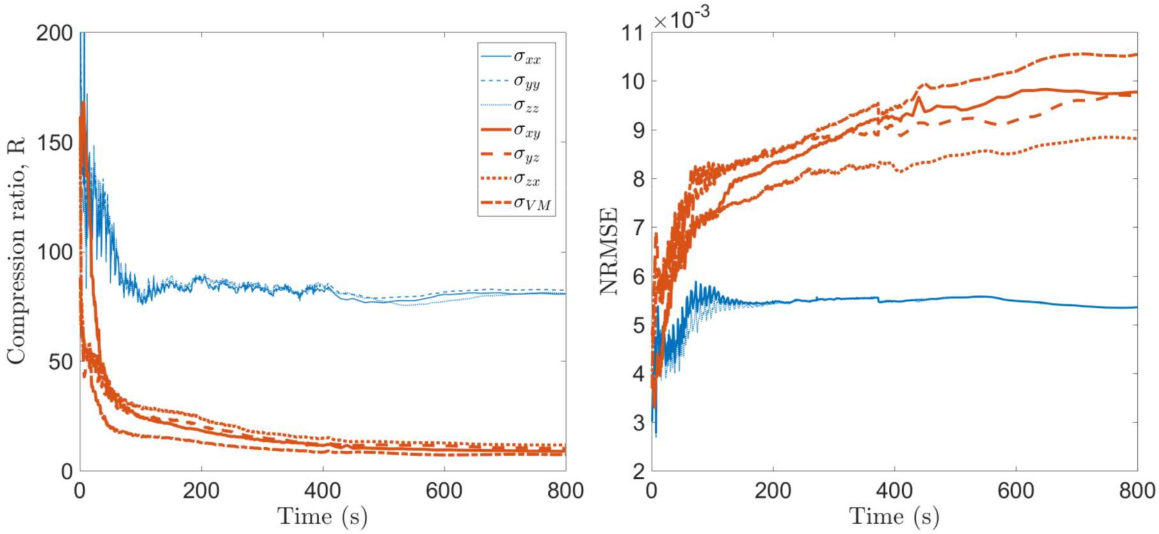

Figure 10: Plots showing the evolution of (left) the compression and (right) the NRMSE error obtained in the wavelet compression of the stress components and the von Mises stress as indicated by the different curves. Results are generated after wavelet compression with adaptive thresholding with $\theta_0 = 0.0075$.

4.4.3 Preservation of Major Features

We first look at the maximum values of quantities of interest in Figure 11. These are the maximum values of temperature, displacement and von Mises stress as a function of time. The reconstructed features are in a very good agreeement with the original ones at all times, especially at the last simulation time step. The final result in time is of crucial importance because it assesses the residual stress and deformation in the AM cylinder. Discrepancies are slighlty more pronounced during the simulation for the displacement and stress features, mainly while the AM is taking place (before 400 s) due to the large gradients in these quantities.

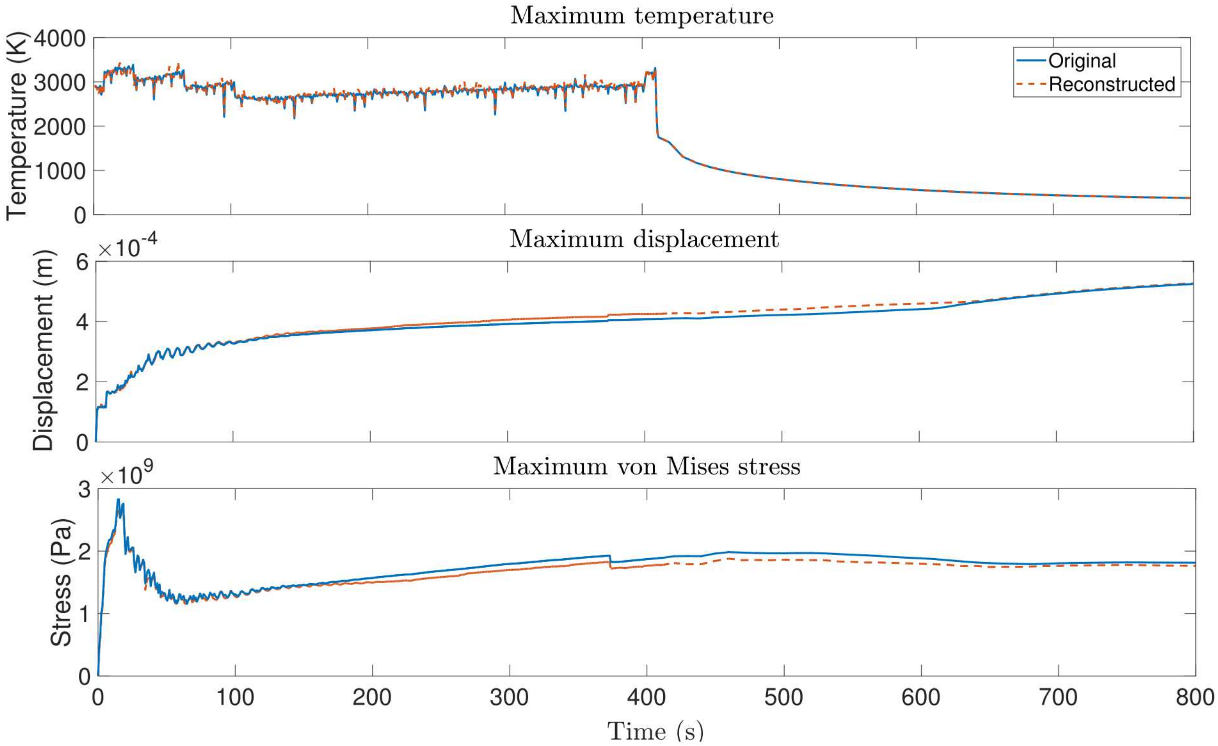

Figure 11: Plots showing the maximum value of the total displacement, temperature and von Mises stress as a function of time. Shown are the original values along with the values generated after wavelet compression with adaptive thresholding where $\theta_0 = 0.0075$.

Also of engineering interest is the assessment of the location where the maximum values reported in Figure 11 occur in the AM cylinder. We consider the locations along the cylinder height as features in the data. Figure 12 shows good agreement between their original and reconstructed versions when plotted as a function of time. The large visible discrepancy between 600 to 700s is due to the sensitivity of the maximum operator to outliers. In order to minimize these discrepancies, a more suitable way to compute the maximum is to evaluate it at the 99^{th} or 95^{th} percentile of the field values.

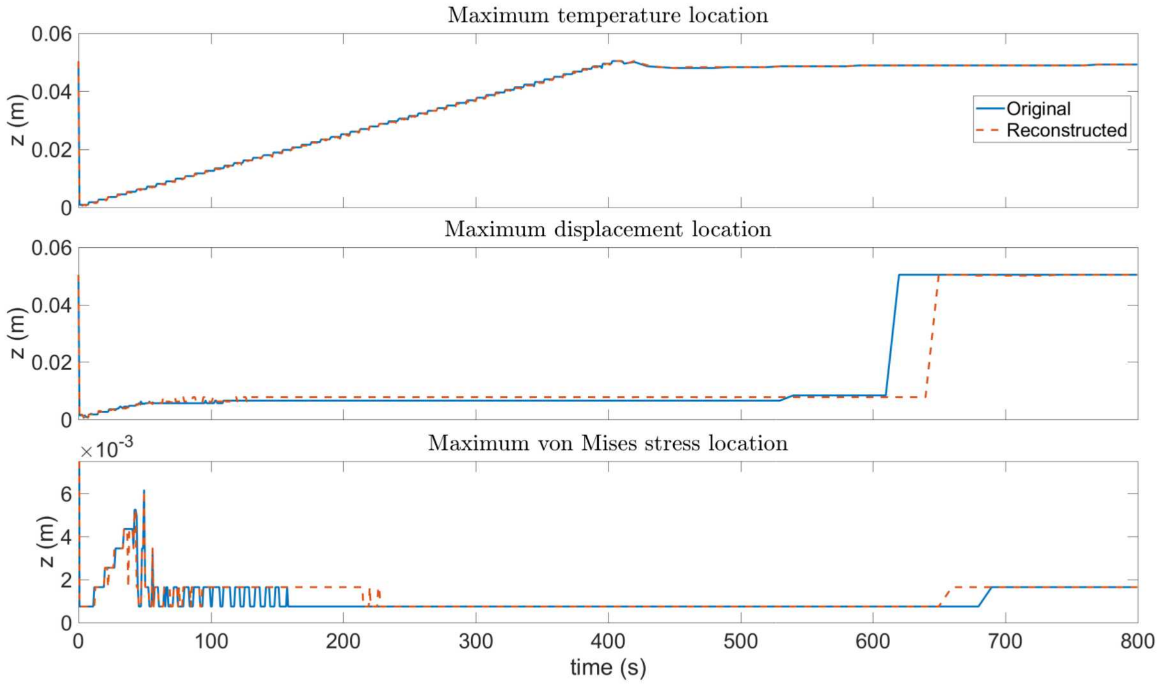

Figure 12: Plots showing the location in the z-direction of the maximum temperature, displacement and von Mises stress as a function of time. Shown are the original values along with the values generated after wavelet compression with adaptive thresholding where $\theta_0 = 0.0075$.

4.4.4 Local Error Metrics

In this section, we examine the local error metrics e_a and e_{div} defined in Eq. 7 and 9. These are defined at each mesh point in the domain. Figure 13 shows the distribution of these errors. Consistent with the global error reported in Figures 8 and 10, the local error in the von Mises stress is larger than that of the temperature and displacement. This suggests a possible relationship between e_a and NRMSE. The distribution of error e_{div} in the physical constraint is similar to a log-normal function with a mode less than 5% and a long tail signifying that the divergence-free constraint is violated in only minor regions of the subdomain. The local errors are due to the large gradients in the stress around the cylinder base. Note that we computed e_{div} by subtracting the residual divergence in the original stress field due to the weak enforcing of the constraint, according to Eq. 9.

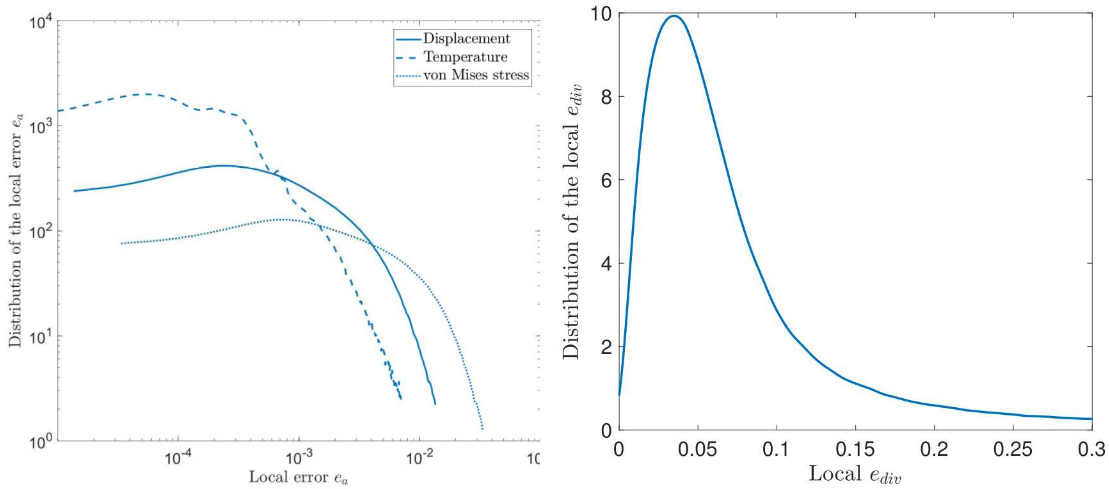

Figure 13: Plots showing the distributions of (left) the local error metric e_a for the total displacement, temperature and von Mises stress, and (right) of the physical constraint error metric e_{div}. Shown are results generated after wavelet compression of the data at the end of the simulation with adaptive thresholding where $\theta_0 = 0.0075$.

In the rest of the results, we study the relationship between the compression ratio R, the different error metrics and the threshold θ_0. Figure 14 (left) shows the expected increase of R with NRMSE. But the increase exhibits different trends for the different data fields suggesting a large dependency of the compressibility on the data. In fact, the smooth temperature and displacement fields results in a linear increase of R with NRMSE, whereas this increase is exponential for the stress fields due to their large gradients. Figure 14 (right) shows the increase of NRMSE with θ_0 and suggests a very good correlation between them. This signifies that for the considered data fields, the desired reconstruction error can be chosen *a priori* by simply setting it as the threshold θ_0. The plots shows that NRMSE is equal or smaller than θ_0 for all considered data fields in this study. The field variables we consider are representative of many other fields encountered in engineering simulations. For example, the temperature field represents fields exhibiting a localized sharp gradient (see Figure 3), the displacement field represents smooth fields all over the domain (see Figure 4) and the von Mises stress field represents fields exhibiting sharp gradients throughout the domain (Figure 5).

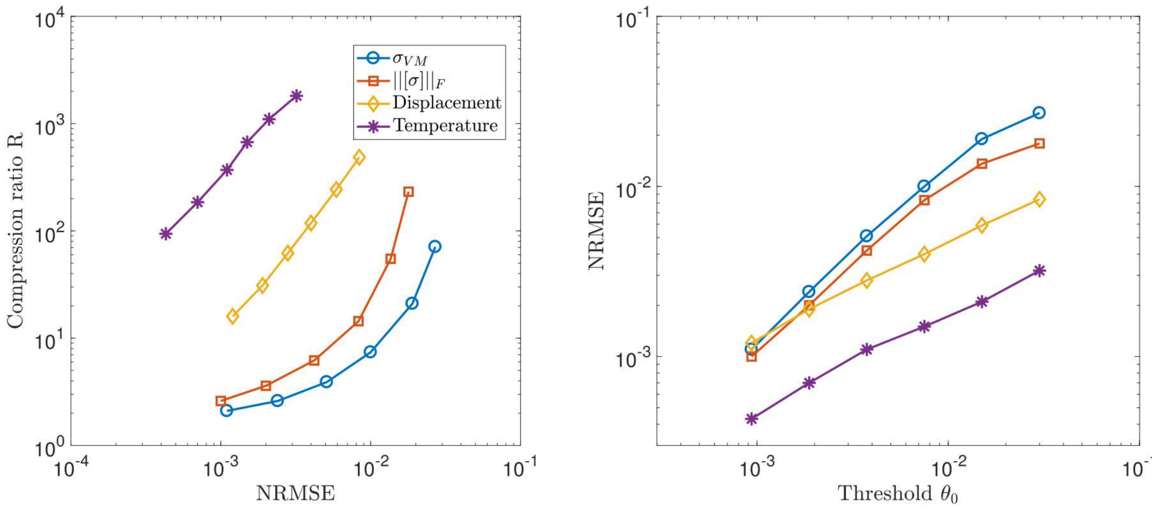

Figure 14: Plots showing (left) the variation of the compression ratio R as a function of the global error metric NRMSE, and (right) the variation of NRMSE as a function of the threshold θ_0, for different data fields as indicated. Shown are results generated after wavelet compression of the data at the end of the simulation.

Figure 15 (left) shows the increase of the maximum local error with θ_0. While the increasing trends are linear, they are significantly different than the ones of NRMSE for the different fields. This suggests that the global and local error metrics have to be assessed separately and cannot be deduced from each other. This is also the case for the error metric e_{div} that corresponds to the deviation from the physical constraint as shown in Figure 15 (right). This plot also shows that e_{div} saturates at large NRMSE *i.e.*, large R, which is not surprising since at such increased error rates, the divergence-free stress constraint is violated completely and the reconstruction is not physically meaningful. The maximum value reached by e_{div} corresponds to divergence of the largest mode of the stress field, i.e. its average value.

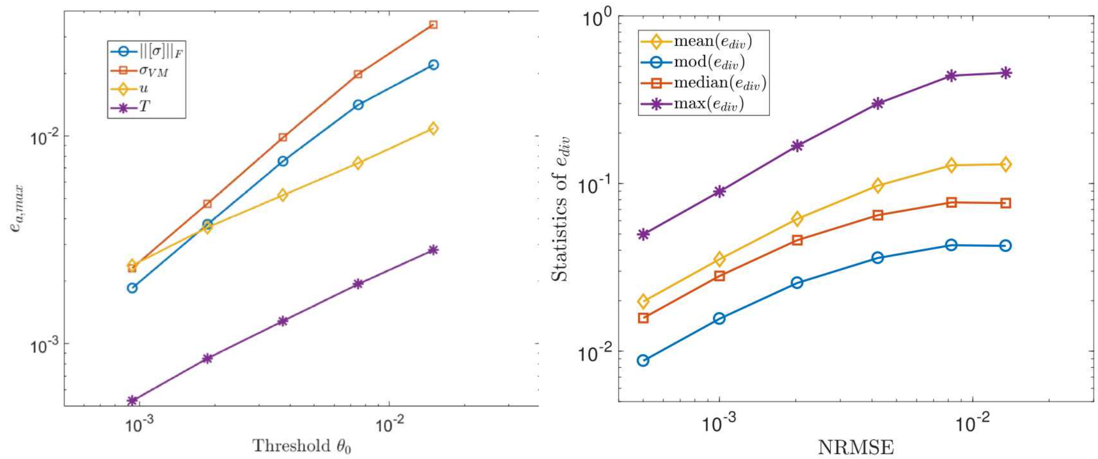

Figure 15: Plots showing the distributions of (left) the local error metric e_a for the total displacement, temperature and von Mises stress, and (right) of the physical constraint error metric e_{div}. Shown are results generated after wavelet compression of the data at the end of the simulation with adaptive thresholding where $\theta_0 = 0.0075$.

4.5 Comparison with other Thresholding methods

In this section, we present the benefits of including the additional terms in Eq. 5 to the universal Donoho threshold. This latter is given in Eqs. 11 and can be expressed in terms of the mean or the median of the wavelet coefficients amplitude as well as their associated absolute deviation.

$$\eta_{mean} = \frac{\eta_{mean,0}}{0.6745} \cdot \text{mean}\left(\Big||\boldsymbol{w}_f| - \text{mean}(|\boldsymbol{w}_f|)\Big|\right) \cdot \sqrt{2\log(N)} \tag{11a}$$

$$\eta_{median} = \frac{\eta_{median,0}}{0.6745} \cdot \text{median}\left(\Big||\boldsymbol{w}_f| - \text{median}(|\boldsymbol{w}_f|)\Big|\right) \cdot \sqrt{2\log(N)} \tag{11b}$$

For both of these cases, we vary the factors $\eta_{mean,0}$ and $\eta_{median,0}$ in order to vary the compression ratios. We present the achievable overall compression ratios for the computational dataset described in Sections 3.2 and 4.4. Figure 16 shows that the original mean and median based universal threshold result in compression ratios that are two and seven-fold, respectively, lower than the adaptive threshold developed in this paper. The universal threshold was originally developed for images involving several discontinuities and it was imperative to preserve these features. This is probably why they tend to be more conservative in terms of compression. However, in smooth computational datasets, the threshold can be relaxed by incorporating the [max(·) - min(·)] term where low values in the data get penalized without inducing substantial errors and at the same time incur more compression.

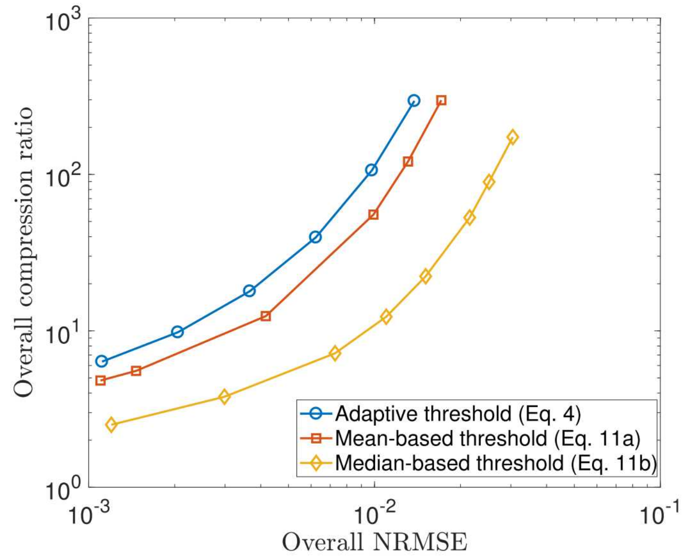

Figure 16: Plots showing the variation of the overall compression ratio with the overall compression for different thresholding strategies, as indicated. The compression ratio is varied by varying the multiplication factors θ_0, $\eta_{mean,0}$ and $\eta_{median,0}$ in Eqs. 5 and 11. Results are generated on the computational dataset described in Sections 3.2 and 4.4.

5 Conclusions

In this paper we have explored the lossy compression of large datasets obtained from both experimental measurements and simulation of a stainless steel hollow cylinder produced by additive manufacturing using Alpert wavelets. The simulation data is distributed among several processes and compression is performed locally within each process in parallel. Compression works by transforming the data vector using the Alpert multiresolution basis into a spectral space and truncating the low-amplitude modes according to a given threshold. Decompression happens through an inverse wavelet transform on the truncated vector. We have developed a general adaptive thresholding method that allows the user to set the required reconstruction error *a priori*. This threshold results in compression ratios that are two to seven times larger than the ones obtained with the original Donoho thresholding method.

We have tested several functional bases of the wavelets and found that, for a given threshold, simple monomial functions grant the largest compression ratio due to their generality in encoding functions. For both experimental and computational data, compression ratios up to two orders of magnitude can be obtained with minimal visual loss in the data fields. When one-dimensional sections are examined before and after compression, we found that larger compression ratios can be achieved without significant discrepencies in the reconstructed results. Compression can thus be controlled depending on what features are expected to be preserved in the data. Moreover, large compression ratios can lead to filtering noise encountered in experimental measurements. The existence of large localized gradients in a field variable significantly decreases the compressibility of its data while smooth variables can achieve much larger compression ratios. Wavelet compression is therefore highly dependent on the considered data. We have developed both local and global error metrics to assess the quality of reconstructed data along with an error metric to quantify the deviation from the divergence-free stress constraint due to lossy compression. Results show that while all these errors increase with compression, there is no clear relationship among them and they cannot be deduced from each other.

These results demonstrate the effectiveness of Alpert wavelets in compressing data fields including those represented on unstructured meshes. This study is the first step toward a more general compression of data. For example, these wavelets may be used to compress partial differential equations operators obtained from their finite element discretization. This can significantly accelerate the solution of these systems especially in the context of optimization and uncertainty quantification studies.

6 Acknowledgments

This work was supported by the Laboratory Directed Research and Development (LDRD) program at Sandia National Laboratories. The authors would also like to acknowledge funding provided by the Born Qualified LDRD.

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 under contract DE-NA-0003525.

7 References

[1] W.E. Frazier. Metal additive manufacturing: A review. *J. Mater. Eng. Perform.*, 23(6):1917–1928, 2014.

[2] Z. Gan, H. Liu, S. Li, X. He, , and G. Yu. Modeling of thermal behavior and mass transport in multi-layer laser additive manufacturing of ni-based alloy on cast iron. *Int. J. Heat Mass Transf.*, 11:709–722, 2017.

[3] M. Salloum, N. Fabian, D.M. Hensinger, J. Lee, E.M. Allendorf, A. Bhagatwala, M.L. Blaylock, J.H. Chen, J.A. Templeton, and I. Tezaur. Optimal compressed sensing and reconstruction of unstructured mesh datasets. *Data Sci. Eng.*, 2017. in press, DOI: 10.1007/s41019-017-0042-4.

[4] M. Burrows and D.J. Wheeler. Block-sorting lossless data compression algorithm. Technical report, Digital Systems Research Center, 1994.

[5] S. Klasky, H. Abbasi, J. Logan, M. Parashar, K. Schwan, and A. Shoshani et al. In situ data processing for extreme scale computing. In *Proceedings of SciDAC 2011*, July 2011.

[6] H. Lehmann and B. Jung. In-situ multi-resolution and temporal data compression for visual exploration of large-scale scientific simulations. In *2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV)*, Paris, France, October 2014.

[7] Sriram Lakshminarasimhan, Xiaocheng Zou, David A Boyuka Ii, Saurabh V Pendse, John Jenkins, Venkatram Vishwanath, Michael E Papka, Scott Klasky, and Nagiza F Samatova. DIRAQ: scalable in situ data-and resource-aware indexing for optimized query performance. *Cluster Computing*, 17(4):1101–1119, 2014.

[8] Peter Lindstrom. Fixed-rate compressed floating-point arrays. *Visualization and Computer Graphics, IEEE Transations on*, 20(12):2674–2683, 2014.

[9] F.F. Bernardon, S.P. Callahan, J.L.D. Comba, and C.T. Silva. Rendering of time-varying scalar fields on unstructured meshes. Technical report, Lawrence Radiation Laboratory, 2005.

[10] A. Gersho and R.M. Gray. *Vector quantization and signal compression.* Springer, 2012.

[11] Woody Austin, Grey Ballard, and Tamara G. Kolda. Parallel tensor compression for large-scale scientific data. Technical report, 2016. arXiv:1510.06689v2.

[12] X. Liu and U.R. Alim. Compressive volume rendering. *Computer Graphics Forum*, 34(3):101–110, 2015.

[13] G.K. Wallace. The JPEG still picture compression standard. *IEEE Trans. Consum. Electr.*, 38(1), 1992.

[14] Bradley K Alpert. A class of bases in L^2 for the sparse representation of integral operators. *SIAM journal on Mathematical Analysis*, 24(1):246–262, 1993.

[15] Bradley Alpert, Gregory Beylkin, Ronald Coifman, and Vladimir Rokhlin. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM Journal on Scientific Computing*, 14(1):159–184, 1993.

[16] E. Pogossova, K. Egiazarian, A. Gotchev, and J. Astola. Tree-structured legendre multi-wavelets. In *Computer Aided Systems Theory EUROCAST 2005*, volume 3643 of *Lecture Notes in Computer Science*, pages 291–300. Springer, 2005.

[17] Mohlenkamp M.J. and Pereyra M.C. *Wavelets, Their Friends, and What They Can Do for You.* European Mathematical Society, 2008.

[18] K.L. Johnson, T.M. Rodgers, O.D. Underwood, J.D. Madison, K.R. Ford, S.R. Whetten, D.J. Dagel, and J.E. Bishop. Simulation and experimental comparison of the thermo-mechanical history and 3D microstructure evolution of 304L stainless steel tubes manufactured using LENS. *Comp. Mech.*, 2017. in press, DOI: 10.1007/s00466-017-1516-y.

[19] D.M. Radunovic. *Wavelets from Math to Practice.* Springer, 2009.

[20] M. Jansen and P. Oonincx. *Second Generation Wavelets and Applications.* Springer, 2005.

[21] Starck J-L., Murtagh F., and Fadili J.M. *Sparse Image and Signal Processing.* Cambridge University Press, 2010.

[22] Alpert B.K. Wavelets and other bases for fast numerical linear algebra. In *Wavelets: A Tutorial in Theory and Applications.* Academic Press, 1992.

[23] Nason G.P. Choice of the threshold parameter in wavelet function estimation. In *Lecture Notes in Statistics.* Springer, 1995.

[24] C. Wang and K.-L. Ma. A statistical approach to volume data quality assessment. *IEEE Trans. Vis. Computer Graphics*, 14(3):590–602, 2008.

[25] Arthur A Brown and Douglas J Bammann. Validation of a model for static and dynamic recrystallization in metals. *International Journal of Plasticity*, 32(1):17–35, 2012.

[26] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth dover printing, tenth gpo printing edition, 1964.

[27] V. Yadav, A.M. Michalak, J. Ray, and Y.P. Shiga. A statistical approach for isolating fossil fuel emissions in atmospheric inverse problems. *Journal of Geophysical Research: Atmospheres*, 121(20), 2016.

[28] L. DeChant, J. Ray, S. Lefantzi, J. Ling, and S. Arunajatesan. K-ϵ turbulence model parameter estimates using an approximate self-similar jet-in-crossflow solution. In 8^{th} *AIAA Theoretical Fluid Mechanics Conference*, June 2017.