SAND2020-7731J

Comparing Field Data using Alpert Multi-Wavelets

Maher Salloum*[1], Kyle N. Karlson[2], Helena Jin[2], Judith A. Brown[3],

Dan S. Bolintineanu[4], Kevin N. Long[5]

July 6, 2020

∗ (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, 7011 East Ave., MS 9042, Livermore, CA 94550

3 Sandia National Laboratories, 1515 Eubank SE., MS 0828, Albuquerque, NM 87123

4 Sandia National Laboratories, 1515 Eubank SE., MS 1064, Albuquerque, NM 87123

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

{mnsallo, hjin, judbrow, dsbolin, knlong, knkarls}@sandia.gov

Abstract

In this paper we introduce a method to compare sets of full-field data using Alpert tree-wavelet transforms. The Alpert tree-wavelet methods transform the data into a spectral space allowing the comparison of all points in the fields by comparing spectral amplitudes. The methods are insensitive to translation, scale and discretization and can be applied to arbitrary geometries. This makes them especially well suited for comparison of field data sets coming from two different sources such as when comparing simulation field data to experimental field data. We have developed both global and local error metrics to quantify the error between two fields. We verify the methods on two-dimensional and three-dimensional discretizations of analytical functions. We then deploy the methods to compare full-field strain data from a simulation of elastomeric syntactic foam.

Keywords. Comparison, Wavelets, Field data, Mesh, Error Metric, Compression, Threshold, Error field

1 Introduction

Simulation is a critical part of the development of new designs because simulations can predict the performance of the designs in expected environments. These predictions can then be used to improve different aspects of the design and guarantee all requirements for the design are met. The models behind these simulations frequently require calibration and validation to ensure the accuracy and predictivity of their results. During the calibration and validation processes, quantitative comparisons are made between simulations results and experimental results of a system representative of the problem of interest.

Typically, these quantitative comparisons have been done using point-wise local data from local sensors or global data from point sensors that measure a global characteristic of a system (e.g. mass, external load, etc.). A small number of local and global quantities-of-interest are identified and measured for model calibration or validation activities. Comparing a small number of these measurements to simulation results is done by comparing the measured quantities to the simulations as a function of a common dependent variable such as time. This results in a sets of data where the model error can be easily calculated by interpolating the measured and simulated quantities of interest to common intervals on the independent variable space.

With the development and wide-spread adoption of full-field experimental techniques (e.g. digital image correlation, digital volume correlation, infrared cameras, etc.), a single experiment can provide a great deal of data for validation and calibration. [1] However, several challenges exist when making a quantitative comparison between full-field simulation results and corresponding experimental data. A point-wise comparison of results can be difficult to make due to the fact that the simulation and experimental results will most likely exist in different coordinate systems and at different points within their respective coordinate systems. This requires translating and rotating results into a common coordinate system, and interpolating the results onto a common set of points in space for direct comparison. General interpolation of three dimensional experimental data with noise is non-trivial [2], while interpolating simulation results may not always be straight forward. For example, interpolating finite element results that lie on element integration points can introduce errors unless specialized methods are used [3]. Even interpolating nodal quantities from finite element simulations when using commercial software can pose a challenge because direct access to element shape functions and mesh connectivity may not be available. An additional challenge when working with full-field data, is that large data sets are usually produced and extracting meaningful error metrics can be a challenge. While point-wise comparisons natively provide an error field and a total global error metric can be calculated from this error field, further post processing is required to determine the nature of the error such as distinguishing between high frequency, local error

or low frequency, global error.

Methods have been proposed to deal with these issues by comparing the fields through moment-based shape descriptors. The first to propose using shape descriptors, also referrred to as shape features, in such a way was Wang, et. al. [4] They present the following as the primary benefits of using these techniques for full-field data comparison: (1) shape features do not require that the data exist at the same locations in space, (2) the large full-field data sets can be represented and compared by a small number of shape features and (3) by using shape features the images are effectively filtered by the underlying polynomials used for the shape descriptor calculations. Many follow-on efforts have focused on using moment-based shape features for model validation and model calibration work flows. [5–7] These works use either Tchebichef or Zernike polynominals as an orthogonal basis for the calculation of moment shape descriptors suitable for field data represented on circular and rectangular subdomains on the entire domain of interest, respectively [8]. By projecting the fields onto a common set of basis functions, a point-wise comparison is not needed and the fields can be directly compared in the projected space. A benefit of using Zernike polynomial basis functions is that the resulting shape descriptors from the projection are invariant to rotation. However, a disadvantage is that numerical errors when integrating a discrete image to calculate the Zernike moments can lead to numerical instabilities. [9, 10] Tchebichef polynomials are discrete polynomials that avoid the numerical instabilities when being integrated on a discrete data set. [11] Other discrete polynomials, such as the Krawtchouk, Hahn, dual Hahn and Racah polynomials, have been used for image reconstruction and classification, but not applied to full-field data comparison. [12–15]

Some of the works using Zernike or Tchebichef shape descriptors cite the invariance of these shape features to scaling, rotation and translation as a benefit of their use for comparison of field data. [16–18] It should be noted that these shape feature methods were originally developed for machine vision applications where the unique identification of field values was not the goal. Instead, these original works were interested in uniquely identifying shapes within an image. [19] As a result, the scaling, rotational and translational invariance is only guaranteed for certain types of shape descriptors under specific conditions. [20–22] When applied to general full-field data for field comparison, there is no guarantee of invariance for scaling, rotational and translation transformations for moment-based shape descriptors using Zernike or Tchebichef orthogonal basis functions. Therefore, using these shape features for field data comparisons still requires the proper alignment and selection of regions within the compared data sets. Methods to guarantee invariance for Tchebichef [23,24] and Zernike [25] shape descriptors have been derived but these specific methods have not been used in full-field data comparisons. These methods rely on coordinate system transformations and normalization techniques to achieve invariance to rotation,

translation and scale.

Regardless of which orthogonal polynomial set is used or if the shape descriptors are invariant to scale, rotation or translation, a common disadvantage of using these shape features is that they cannot inherently represent fields on irregular geometry. Work has been done to extend moment based shape features to more general geometries. In [26], Zernike polynomials were mapped from circular to rectangular domains using the Schwarz-Christoffel transform. Thus, a method was developed allowing the use of Zernike shape descriptors on field-data represented on both circular and rectangular geometries.Wang *et.al.* [7] extended Zernike polynomials to a rectangular plate with circular hole located at the plate center. They used Gram-Shmidt orthogonalization to ensure the orthogonality of the adapted polynomials. A general set of shape descriptors can be derived for any geometry using the adaptive geometric moment descriptor (AGMD) technique also developed by Wang. [27]. The AGMD was recently improved using basis-updating to address its limitation in handling large full-field images [28]. However, a new set of basis functions must be derived for each geometry investigated. This complexity lead Lampeas [18] to use several sub-domains with the fields identified using Zernike shape descriptors instead of the AGMD technique.

The last shortcoming of the moment-based shape features is that these techniques are restricted to two-dimension (2D) field data. A general technique applicable to three-dimensional (3D) data, such as digital volume correlation data, is still needed due to the massive data size and large difference in the field data representation (mesh) between experiments and simulations [29, 30].

1.1 Proposed Method

We present a systematic method to compare N-dimensional data fields sampled at arbitrary locations with no geometric constraints. This is accomplished by comparing the data fields in a unified spectral space using Alpert multi-wavelet (AMW) [31] transforms of the data. Similar to the methods using moment based-shape descriptors, the differences in the spatial discretization of the fields, the existence of noise and issues arising due to missing points are all bypassed when using our method. However, by using AMW for the spectral representation of the data, numerical instabilities are avoided, data can natively be represented on irregular geometry, the proposed methods can be applied to N-dimensional data and the comparisons are insensitive to translation and scale.

While in our previous paper [32], the work focused on data compression of scientific data using AMW in order to reduce storage size, our current work focuses on extending the capabilities of AMW to verification and validation, that is by comparing two field datasets represented on irregular meshes and geometries.

We first start by a brief mathematical background on wavelets. Second, we describe the forward and

inverse wavelet transforms along with the truncation that results in a unified spectral space for field data comparison. Third, we present the comparison method based on the wavelet representation of the data. Finally, we apply our methods to two 3D problems. The first is an analytical verification problem on an irregular 3D domain. The second is a realistic test case involving an irregular 3D geometry in a solid mechanics application. For both of these problems, we also create error fields for the comparisons and introduce a new wavelet-based method for projecting the field data from two different irregular sets of locations onto a common set of spatial points.

2 Background on Wavelets

Wavelets consist of multi-resolution bases suitable to represent any function in a spectral domain [33]. One major benefit of wavelets is their ability to transform different spatial sampling of the same data into a unified space. Let $\boldsymbol{f}$ be a vector of dimensionality N representing a data field, *e.g.* a displacement field in a solid mechanics finite element simulation. Here N can be the number of mesh points in the simulation, it also can be the number of pixels in an image. $\boldsymbol{f}$ can be transformed back and forth from the physical and spectral domains using a wavelet operator $[\Psi]$. During the transforms, the spectral representation is often truncated for compression and filtering purposes [34].

Several types of wavelet bases exist in the literature. The majority of these wavelets (Daubechies, Meyer, Biorthogonal, etc.) can only be applied to data represented on regular grids, such as images. Such wavelets are known as first generation wavelets [35]. Wavelets that generalize to irregular grids such unstructured meshes are called second generation wavelets or tree-wavelets [34, 36]. In fact, the multi-wavelets developed by Alpert [31] have been found in a recent study [37] to be a suitable basis for data represented on unstructured meshes. Hence, we use Alpert multi-wavelets (AMW) in this paper to transform data represented on unstructured grids such as results from large engineering simulations and digital image correlation data sets. More information on the mathematics of wavelets are given in Appendix A.

2.1 Wavelet Transforms

Similarly to any basis transform, a wavelet transform can be cast as a matrix-vector product, where the matrix and vector contain the wavelet bases and the data, respectively. Wavelet transforms can also be performed in a matrix-free fashion. For convenience purposes, we will present our wavelet tranforms and comparison algorithms in matrix-vector products form.

In the forward wavelet transform, $\boldsymbol{f}$ 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 requires building the full wavelet operator $[\Psi]$ when $\boldsymbol{f}$ is represented on an unstructured mesh [32] which is often the case in engineering simulations. We build $[\Psi]$ according to Alpert multi-wavelets (AMW) described in [38] and outlined in our recent study [37]. The rows of the matrix $[\Psi]$ contain the different wavelet function bases for all wavelet modes. The major advantage of AMW 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 [36]. AMW are also able to ignore missing points during the forward transform and recover their values during an inverse wavelet transform. During the inverse transform, the data f is reconstructed using:

$$\boldsymbol{f} = [\Psi]^T \cdot \boldsymbol{\omega}_f \tag{2}$$

2.2 Data compression and filtering through wavelet truncation

Since the data is transformed into a spectral space using the AMW, truncation of the non-dominant AMW modes can provide data compression and filtering. After applying the operation in Eq. 1 on the data $\boldsymbol{f}$, we truncate $\boldsymbol{w}_f$ as follows. We first sort the amplitude (absolute value) of $\boldsymbol{w}_f$ in descending order then truncate the latter part of the sorted entities *i.e.* the entities with lower magnitude. We can decide where to truncate $\boldsymbol{w}_f$ using either of the following approaches:

1. Truncating by keeping a given number M of large amplitude coefficients.
2. Truncating by keeping the coefficients that are larger than a given threshold θ [32] (see Eq. 9).

The resulting truncated vector $\boldsymbol{\phi}$ is of size $M < N$. Hence, truncation induces compression. The compression ratio is defined as:

$$R = \frac{N}{M} \tag{3}$$

Along with $\boldsymbol{\phi}$, the indices $\boldsymbol{\mathcal{I}}$ of the sorted and retained large coefficients also have to be stored to be able to reconstruct an approximation of the original function f through the inverse wavelet transform:

$$\hat{\boldsymbol{f}} = [\Psi]^T_{\boldsymbol{\mathcal{I}}} \cdot \boldsymbol{\phi} \tag{4}$$

We define the reconstruction error of the compressed function $\hat{\boldsymbol{f}}$ as the normalized relative mean squared error (NRMSE) given by:

$$NRMSE_f = \frac{||\boldsymbol{f} - \hat{\boldsymbol{f}}||_2}{\sqrt{N}\,[\max(\boldsymbol{f}) - \min(\boldsymbol{f})]} \tag{5}$$

Data reconstruction using Eq. 4 and the associated NRMSE have been discussed in details in [32] where it is found from the considered examples that the error can be estimated before truncation by a suitable selection of the truncation threshold coefficient θ_0 (see Eq. 9). Filtering is accomplished by truncating the modes with higher spatial frequency. In general, these high frequency modes are the modes with low amplitude coefficients. As a result, compression and filtering are accomplished together. Some formal works on filtering and compressing with wavelets are [39–42].

2.3 Comparing Wavelet Spectra as a Global Error Metric

As stated previously, a quantitative error metric is required for calibration and validation purposes. AMW enable such a comparison through their unified spectral space and wavelet mode magnitudes. Consider two data vectors $\boldsymbol{f}$ and $\boldsymbol{g}$ that we wish to compare. $\boldsymbol{f}$ and $\boldsymbol{g}$ are field data and they are, in general, defined on different meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, and involves different number of points N_1 and N_2, respectively. We first perform a forward wavelet transform on $\boldsymbol{f}$ and $\boldsymbol{g}$ following:

$$\begin{aligned} \boldsymbol{w}_f &= [\Psi]_1 \cdot \boldsymbol{f} \\ \boldsymbol{w}_g &= [\Psi]_2 \cdot \boldsymbol{g} \end{aligned} \tag{6}$$

where the wavelet operators $[\Psi]_1$ and $[\Psi]_2$ are computed based on the meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. The resulting vectors $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ contain both positive and negative entities. We truncate the amplitude (*i.e.* absolute value) of these two vectors and sort their amplitudes into $\boldsymbol{\phi}_{f,\mathcal{I}}$ and $\boldsymbol{\phi}_{g,\mathcal{J}}$ both of size $M < N_1$ and $M < N_2$. M is chosen such that the error between the error in f and g before and after truncation is below a given tolerance (see Eq. 5). The choice of a truncation threshold that controls the resulting error is described in [32]. $\mathcal{I}$ and $\mathcal{J}$ are the sorting indices, they should be consistent with the wavelet mode functions. This condition is automatically satisfied if the meshes $\mathcal{M}_1$ and $\mathcal{M}_2$ are the same. If not, mode switching could occur with respect to the amplitude sorting. The reason why mode switching occurs is an interesting mathematical question that is beyond the scope of this paper. Although a minor error can result if the mode switching is neglected, we choose to correct it by finding

the inconsistent indices wherever $\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{J}} \neq 0$ and suitably flipping them resulting in $\boldsymbol{\mathcal{I}}'$ and $\boldsymbol{\mathcal{J}}'$. Now $\boldsymbol{\phi}_f$ and $\boldsymbol{\phi}_g$ are of the same size and exist in a unified spectral domain. We define the global error metric as the norm of the difference between $\boldsymbol{\phi}_f$ and $\boldsymbol{\phi}_g$. In general, the trends of the wavelets amplitudes span several orders of magnitudes. Thus, a convenient approach is to compute the difference ϵ as the difference between the logarithms of $\boldsymbol{\phi}_f$ and $\boldsymbol{\phi}_g$ following:

$$\epsilon = \frac{||\log(\boldsymbol{\phi}_f/\sqrt{N_f}) - \log(\boldsymbol{\phi}_g/\sqrt{N_g})||_2}{\sqrt{M}\max[\log(\boldsymbol{\phi}_f/\sqrt{N_f})]} \tag{7}$$

where $\sqrt{N_f}$ and $\sqrt{N_g}$ account for the difference in the number of points in the field meshes. Although there can be several ways to normalize the global error, we choose the maximum wavelet coefficient as the normalization factor. The procedure to compute the global error ϵ is presented in Algorithm 1.

Algorithm 1 Steps to compute the L_2 difference between two vector fields $\boldsymbol{f}$ and $\boldsymbol{g}$ (of sizes N_f and N_g, respectively) in the wavelet domain.

1: Transform $\boldsymbol{f}$ and $\boldsymbol{g}$ using wavelets into $\boldsymbol{w}_f$ and $\boldsymbol{w}_g$ (see Eq. 1)

2: Find a truncation at M wavelet coefficients where $NRMSE_{\boldsymbol{f}}$ and $NRMSE_{\boldsymbol{g}}$ are both below a given tolerance

3: Sort and truncate $|\boldsymbol{w}_f|$ and $|\boldsymbol{w}_g|$ into $\boldsymbol{\phi}_{f,\boldsymbol{\mathcal{I}}}$ and $\boldsymbol{\phi}_{g,\boldsymbol{\mathcal{J}}}$

4: Correct the consistency of the indices wherever $\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{J}} \neq 0$.

5: Compute $\epsilon = \frac{||\log(\boldsymbol{\phi}_f/\sqrt{N_f}) - \log(\boldsymbol{\phi}_g/\sqrt{N_g})||_2}{\sqrt{M}\max[\log(\boldsymbol{\phi}_f/\sqrt{N_f})]}$

Different logarithm bases can be used when ϵ is computed, as suitable to the given problem. Its choice depends on the ability to pre-estimate the trends and decay rate of the wavelet spectrum. Small and large decay rates would require smaller and larger logarithm bases, respectively. However, it could be likely that the metric ϵ is insensitive to the selected logarithm base since we truncate the spectrum using Eq. 9 as described below, such that only the large dominant wavelet coefficients are included in the metric.

In addition to the L_2 error metric ϵ, we can devise other error metrics. For example, we can compute the element-wise relative error between $\boldsymbol{\phi}_f$ and $\boldsymbol{\phi}_g$ then obtain the resulting average. Such error metric would be expressed as:

$$\rho = \frac{1}{M}\sum_{i=1}^{M} \frac{|\log(\phi_{f,i}/\sqrt{N_f}) - \log(\phi_{g,i}/\sqrt{N_g})|}{|\log(\phi_{f,i}/\sqrt{N_f})|} \tag{8}$$

In practice, we seek to develop a systematic method for error estimation that is insensitive to the choice of M. Furthermore, we seek to minimize M in order to maximize the compression of the data arrays representing the compared functions. In other words, we seek finding the minimal value of M above which the corresponding error metric does not change. Thus, the error metric ϵ seems to be a

suitable choice as we will show in Section 4.1. We also set M using the threshold developed by Salloum *et.al* [32] retaining the coefficients larger than:

$$\theta_f = \frac{\theta_0}{0.6745} \cdot [\max(\boldsymbol{f}) - \min(\boldsymbol{f})] \cdot \frac{\text{mean}\left(\left||\boldsymbol{w}_f| - \text{mean}(|\boldsymbol{w}_f|)\right|\right)}{\text{mean}(|\boldsymbol{w}_f|)} \cdot \sqrt{2\log(N_f)} \tag{9}$$

where we fix $\theta_0 = 0.01$. When comparing two functions f and g, the computed thresholds θ_f and θ_g results in M_f and M_g. We choose the optimal number of retained coefficients as $M = \max(M_f, M_g)$.

3 Error Field Construction

In addition to the global error metric, AMW can be used as an interpolation method between discretizations of a domain. Once results from two sources are interpolated onto a common set of spatial points, an error field between the two data sets can be used to understand and identify model form error. As a result, we seek to compute the spatial error field between two field vectors $\boldsymbol{f}_1$ and $\boldsymbol{g}_2$ defined on different meshes $\mathcal{M}_1$ and $\mathcal{M}_2$ of sizes N_1 and N_2. The resulting error field can be defined on two different meshes $\mathcal{M}_1$ or $\mathcal{M}_2$. At a first glance, the straightforward way to compute the error field $\boldsymbol{e}$ appears to directly compute the difference:

$$\boldsymbol{e} = [\Psi] \cdot (\boldsymbol{w}_1 - \boldsymbol{w}_2) \tag{10}$$

$$\begin{aligned} \boldsymbol{w}_1 &= [\Psi]_1 \cdot \boldsymbol{f}_1 \\ \boldsymbol{w}_2 &= [\Psi]_2 \cdot \boldsymbol{g}_2 \end{aligned} \tag{11}$$

where $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are the wavelet transforms of $\boldsymbol{f}_1$ and $\boldsymbol{g}_2$, respectively according to Eq. 11. The wavelet matrix $[\Psi]$ can be the one corresponding to either mesh $\mathcal{M}_1$ or $\mathcal{M}_2$, depending on the intended mesh representation of the error field $\boldsymbol{e}$. Eq. 10 holds if $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are obtained with matrices $[\Psi]_1$ and $[\Psi]_2$ containing wavelet function bases of consistent directions. While this consistency is guaranteed in the case of regular grid data (*e.g.* images), it might not always hold for unstructured meshes. In fact, the wavelet basis direction inconsistency arises in unstructured meshes due to the need of Gram-Shmidt orthogonalization in Alpert multi-wavelets [37] which is sensitive to the underlying mesh, *i.e.* the position of the mesh points. This results in orthogonal vectors of inconsistent directions. Thus, when forward wavelet transform is performed on the same function using two different unstructured meshes, the resulting coefficients are expected to have inconsistent signs.

3.1 Enforcing Wavelet Coefficient Signs

There are several ways to correct for the sign inconsistency ocurring in matrix operations involving eigenvectors [43,44]. Such operations are encountered when building the Alpert wavelet operator through the Gram-Shmidt orthogonalization [38]. In order to enforce a wavelet coefficient sign consistency (*resp.* a wavelet basis direction consistency), we propose two methods: a direct method that acts directly on the wavelet matrix and suitably updates the wavelet functions sign. The second method acts on the wavelet coefficients and iteratively updates their sign.

3.1.1 Direct Method

Let $[\Psi]_1$ and $[\Psi]_2$ be the wavelet matrices corresponding to the $\mathcal{M}_1$ or $\mathcal{M}_2$ meshes with N_1 and N_2 mesh points, respectively. We consider in our derivation the case where $N_1 > N_2$. The first N_2 wavelet functions in $[\Psi]_1$ and $[\Psi]_2$ are essentially the same but the sign of some of them are flipped.

In order to find the wavelet functions with sign inconsistency, we compute a scalar global measure on the wavelet functions that indicates their sign. Our initial choice is to compute the integral of the wavelet functions ψ_i over the spatial domain. However such integral is equal to zero due to the vanishing moment property of wavelets [34] where:

$$\int_{\Omega} \psi_i(\boldsymbol{x}) \boldsymbol{x}^{\mathcal{O}-1} d\Omega = 0 \tag{12}$$

where $\mathcal{O}$ is the Alpert wavelet polynomial order. Therefore we choose to compute the sign indicator as:

$$\mathcal{S}_i = \int_{\Omega} \psi_i(\boldsymbol{x}) \boldsymbol{x}^{\mathcal{O}} d\Omega \tag{13}$$

Numerically we approximate these integrals for $[\Psi]_1$ and $[\Psi]_2$ as:

$$\begin{aligned} \boldsymbol{\mathcal{S}_1} &= [\Psi]_{1,J} \cdot \prod_{i=1}^{d} \boldsymbol{x}_i^{\mathcal{O}} \\ \boldsymbol{\mathcal{S}_2} &= [\Psi]_2 \cdot \prod_{i=1}^{d} \boldsymbol{x}_i^{\mathcal{O}} \end{aligned} \tag{14}$$

where d is the number of spatial dimensions and J indicates the first N_2 rows of $[\Psi]_1$. We correct the signs of the wavelet functions in $[\Psi]_2$ following:

$$[\Psi]_{2,s} = [\Psi]_2 \otimes sign(\boldsymbol{\mathcal{S}_1} \otimes \boldsymbol{\mathcal{S}_2}) \tag{15}$$

that is, we multiply each column in $[\Psi]_2$ by the sign of the product of the two vectors $\boldsymbol{\mathcal{S}_1}$ and $\boldsymbol{\mathcal{S}_2}$. Hence, the projection of the function $\boldsymbol{f}_1$ on the mesh $\mathcal{M}_2$ is:

$$\hat{\boldsymbol{f}}_2^1 = [\Psi]_{2,s}^T \cdot \boldsymbol{w}_{1,J} \cdot \sqrt{N_2/N_1} \tag{16}$$

where the factor $\sqrt{N_2/N_1}$ accounts for the difference in mesh size. Thus results in the following error field on $\mathcal{M}_2$:

$$\boldsymbol{e} = \hat{\boldsymbol{f}}_2^1 - \boldsymbol{g}_2 \tag{17}$$

Remark: A derivation similar to Eqs. 13-17 holds for the case when $N_1 < N_2$.

3.1.2 Iterative Method

In the derivation below, the error field will be represented on $\mathcal{M}_2$. Our approach is to find $\boldsymbol{f}_2$, the representation of $\boldsymbol{f}$ on the mesh $\mathcal{M}_2$ using wavelet transforms. As such, computing the error field can be performed in a straightforward manner as $\boldsymbol{f}_2 - \boldsymbol{g}_2$ on the mesh $\mathcal{M}_2$.

Let $\mathcal{M}_a = \mathcal{M}_2 \cup \mathcal{M}_1$ be the combined mesh with $[\Psi]_a$ its corresponding wavelet operator. In order to combine the two meshes, they have to be aligned to the same range *e.g.* to the [0,a] hypercube. Let $\boldsymbol{f}_a \in \mathbb{R}^{N_1+N_2}$ be the joint vector of the unknown $\boldsymbol{f}_2$ and the given $\boldsymbol{f}_1$ following:

$$\boldsymbol{f}_a = \begin{pmatrix} \boldsymbol{f}_2 \\ \boldsymbol{f}_1 \end{pmatrix} \tag{18}$$

$[\Psi]_a$ transforms $\boldsymbol{f}_a$ into a vector $\boldsymbol{w}_{f_a} \in \mathbb{R}^{N_1+N_2}$ of corresponding amplitude vector $\boldsymbol{\phi}_{f_a}$. Since all $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ and $\boldsymbol{f}_a$ represent the same function, their truncated wavelet transforms should be the same in amplitude, up to a multiplication factor accounting for the sizes of each vector (see Eq. 7). Thus, $\boldsymbol{\phi}_{f_a}$ can be written as:

$$\boldsymbol{\phi}_{f_a} = \begin{pmatrix} \boldsymbol{\phi}_{f_1} \sqrt{(N_1+N_2)/N_1} \\ \emptyset \end{pmatrix} \tag{19}$$

where $\boldsymbol{\phi}_{f_1} \in \mathbb{R}^M$ is the truncated wavelet transform of $\boldsymbol{f}_1$ and the the rest of $\boldsymbol{\phi}_{f_a}$ contains zeros. The entities in $\boldsymbol{\phi}_{f_a}$ are the amplitude of those in $\hat{\boldsymbol{w}}_{f_a}$, the wavelet transform of $\hat{\boldsymbol{f}}_a$. $\hat{\boldsymbol{w}}_{f_a}$ and $\boldsymbol{\phi}_{f_a}$ can be

written as:

$$\begin{aligned} \hat{\boldsymbol{f}}_a &= [\Psi]_a^T \cdot \hat{\boldsymbol{w}}_{f_a} \\ \boldsymbol{\phi}_{f_a} &= \left|[\Psi]_a \cdot \hat{\boldsymbol{f}}_a\right| \\ \boldsymbol{\phi}_{f_a} &\approx \hat{\boldsymbol{w}}_{f_a} \end{aligned} \tag{20}$$

Our goal is to find $\boldsymbol{f}_2$ which is a subset of $\hat{\boldsymbol{f}}_a$. We note that the last equation in 20 signifies that $\boldsymbol{\phi}_{f_a}$ is not exactly equal to $\hat{\boldsymbol{w}}_{f_a}$ because some of the coefficients in $\hat{\boldsymbol{w}}_{f_a}$ might be negative whereas all of them are positive in $\boldsymbol{\phi}_{f_a}$. Therefore, we have to find an accurate estimation of $\hat{\boldsymbol{w}}_{f_a}$. The entities in $\boldsymbol{\phi}_f$ are all positive and their signs should be adjusted such that $\hat{\boldsymbol{f}}_a$ reflects an approximation of $\boldsymbol{f}_a$ with minimal error. Hence, we perform a series of inverse wavelet transform iterations on $\hat{\boldsymbol{w}}_{f_a}$ to minimize the error between $\boldsymbol{f}_1$ and its approximate $\hat{\boldsymbol{f}}_1 \subset \hat{\boldsymbol{f}}_a$, *i.e.* the latter part of $\hat{\boldsymbol{f}}_a$ (see Eq. 18). We loop over the M entities in $\boldsymbol{\phi}_{f_1} \subset \hat{\boldsymbol{w}}_{f_a}$ starting with the one of largest amplitude. We flip the sign of the element and compute $\hat{\boldsymbol{f}}_a$ using Eq. 20. If $||\boldsymbol{f}_1 - \hat{\boldsymbol{f}}_1||_2$ decreases, we keep the sign change; if not, we keep the original sign. These steps are summarized in the following algorithm:

Algorithm 2 Steps to compute the error field $\boldsymbol{e}$ between two functions $\boldsymbol{f}_1$ and $\boldsymbol{g}_2$ represented on two different meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. The error field is calculated on $\mathcal{M}_2$, so we must calculate $\hat{\boldsymbol{f}}_2$ where $\hat{\boldsymbol{f}}_2 \subset \hat{\boldsymbol{f}}_a$ and is the function $\boldsymbol{f}_1$ interpolated onto $\mathcal{M}_2$.

Given θ_0, $\boldsymbol{f}_1$ and $\boldsymbol{g}_2$, find $N = \max(M_1, M_2)$ where M_1, M_2 are the number of points in the grids $\mathcal{M}_1$ and $\mathcal{M}_2$

Form the joint grid $\mathcal{M}_a = \mathcal{M}_2 \cup \mathcal{M}_1$ and compute its corresponding wavelet operator $[\Psi]_a$

Form the joint vector $\boldsymbol{f}_a$ according to Eq. 18

Transform $\boldsymbol{f}_1$ into $\boldsymbol{\phi}_{f_1} \in \mathbb{R}^M$ and form $\boldsymbol{\phi}_{f_a}$ according to Eq. 19 as an initial approximation of $\hat{\boldsymbol{w}}_{f_a}$

Compute $\hat{\boldsymbol{f}}_{a,0} = [\Psi]_a^T \cdot \boldsymbol{\phi}_{f_a}$ where $\boldsymbol{\phi}_{f_a}$ is the initial approximation of $\hat{\boldsymbol{w}}_{f_{a,0}}$

Compute the initial error $\eta_0 = ||\boldsymbol{f}_1 - \hat{\boldsymbol{f}}_{1,0}||_2$ where $\hat{\boldsymbol{f}}_{1,0} \subset \hat{\boldsymbol{f}}_{a_0}$

for i=1 to M **do**

  Flip the sign of $w_{f_{i-1}} \in \hat{\boldsymbol{w}}_{f_{a,i-1}}$

  Compute $\hat{\boldsymbol{f}}_{a,i} = [\Psi]_a^T \cdot \hat{\boldsymbol{w}}_{f_{a,i-1}}$

  Compute $\eta_i = ||\boldsymbol{f}_1 - \hat{\boldsymbol{f}}_{1,i}||_2$ where $\hat{\boldsymbol{f}}_{1,i} \subset \hat{\boldsymbol{f}}_{a,i}$

  if $\eta_i > \eta_{i-1}$ **then**

    Flip the sign of $w_{f_{i-1}}$ back

  end if

end for

$\hat{\boldsymbol{f}}_2 = \hat{\boldsymbol{f}}_{2,N}$ where $\hat{\boldsymbol{f}}_{2,N} \subset \hat{\boldsymbol{f}}_{a,N}$

$\boldsymbol{e} = \hat{\boldsymbol{f}}_2 - \boldsymbol{g}_2$

Practically, both the direct and iterative methods can be used in any application according to their characteristics. The iterative method is easy to program in any language but it requires iterations which may make it relatively slow. The direct method is faster and consists of less steps in its algorithm. However it requires building and manipulating sparse matrices that are not straightforward in programming languages and avalaible matrix libraries. Thus, the choice of a method depends on the availability of computing platform. For example, if the error field estimation takes place *in situ* during a simulation where the cost of any post-processing has to be minimized then the direct method is a suitable choice. If the error field is estimated outside a simulation where the computational time is not an issue then the iterative method is more appropriate.

3.2 Optimal Wavelet Order Selection

A challenge encountered in the two field comparison method we developed is that they both require the wavelet order as input. Selecting the optimal order is common topic in machine learning that is usually

treated using cross-validation [45]. However, our methods have intrinsic indicator that allow for optimal wavelet order selection. Both methods result in the approximate projection $\hat{\boldsymbol{f}}_2$ of the field $\boldsymbol{f}_1$ on the mesh $\mathcal{M}_2$. Thus, an appropriate error inicator is the global error metric ϵ between the fields $\boldsymbol{f}_1$ and $\hat{\boldsymbol{f}}_2$ represented on the meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. These two fields are supposed to be the same thus η should approach zero. It follows that the wavelet order that results with the lowest η is the optimal one. Moreover, The iterative method we derived has another error indicator η which also can be used as a metric to assess the most adequate wavelet order used in the comparison.

Practically, the either method would be run for different wavelet orders and the one that gives the lowest η or ϵ would correspond to the most accurate estimation of the error field $\boldsymbol{e}$.

3.3 Performance Improvement

The procedures for error field estimation derived in both the direct and iterative methods contain expensive numerical operations such as wavelet transforms. These operations are essentially equivalent to the product of a vector and a sparse matrix. The cost of such numerical operations grows with the size of the meshes involved in the comparison. We propose a simple technique to reduce this cost, that is by subdividing the meshes, performing the comparison on smaller subdomains, then combine the obtained smaller error fields into the full one. Typically, large simulations are already solved across multiple processors and the discretizations are already subdivided. If this is the case, a comparison to experimental data could be done by finding the experimental data points within the simulation subdomains.

If there is no pre-existing subdivision of the geometry of interest, subdivision can be performed using a variety of methods. We explain two possible approaches here. The first is simply dividing the spatial ranges of the domain into N_x, N_y and N_z intervals in the x, y and z directions, respectively. With unstructured meshes, especially with domains containing holes such as the ones in Figures 1 and 5, this subdivision might result in empty subdomains that can be easily skipped during the computations. But, the main benefit here is that the spatial ranges of the subdomains is guaranteed to be consistent among the two mesh fields intended to be compared. The second approach is based on any mesh subdivision technique found in the machine learning community such as k-means clustering [46]. These methods result in equal subdomains but they require special attention to ensure that the subdomain borders are consistent in space between the two mesh fields intended to be compared. This condition is usually automatically met if the two meshes are uniform over the whole domain.

The benefit of subdivision is two-fold. First it substantially reduces the overall cost of the comparison by allowing parallelism on multi-core and multi-threaded systems that are ubiquitous in the current computing platforms. In fact, the subdomains formed after subdivision can be considered independent of

each others and can be compared separately. Quantifying this cost improvement is an interesting exercise but it is beyond the scope of this paper. The second benefit is that subdivision reduces the number of features (*e.g.* oscillations in the field) in each subdomain allowing lower wavelet orders for an accurate comparison. This in turn participates in further reducing the cost of the field comparison.

4 Results and Discussion

In this section, we will apply our AMW algorithms to three different sets of field data. The first set of field data is three two-dimensional analytical functions calculated on a single set of arbitrary spatial locations. We use this set of data to demonstrate the AMW transformation on an irregular grid and to evaluate the global error metrics proposed earlier. We also demonstrate the insensitivty of the global metrics to scale and translation for one of the functions on this grid. The second set of data includes two arbitrary three-dimensional functions evaluate on two different irregular grids. We use these functions an grids to verify the direct and iterative error reconstruction algorithms present in Section 3. The final set of data contains three-dimensional strain field data on a large finite element simulation of a porous foam where we demonstrate the use of AMW algorithms on a large, applied problem with over 4 million nodes. Results in the this section are produced using the SWinzip library [47].

4.1 Demonstration of AMW on two-dimensional analytical functions

Here we demonstrate the results of AMW transformations on the following two-dimensional functions:

$$\begin{aligned} f &= 48\sin(6\pi x)\sin(5\pi y)\sin(4\pi x) \\ g &= 48\sin(2\pi x)\left[\sin(2\pi x) - \sin(2\pi y)\right] - 52 \\ h &= 5\sin(2\pi x)\left[9\sin(2\pi x) - 7\sin(2\pi y)\right] - 52 \end{aligned} \tag{21}$$

The functions are evaluated on the same grid defined within an irregular geometry. Figure 1 shows the three functions displayed on the chosen irregular geometry on a unit square centered about $x = y = 0.5$ along with the first 20 dominant modes of their AMW transformations.

These functions were specifically chosen to show the spectral nature of the AMW transforms and the sensitivity of the wavelet mode amplitudes to similar and dis-similar functions. For example, the function $\boldsymbol{f}$ has higher spatial frequency than the functions g and h and is dis-similar to these functions, both of these differences are evident in the wavelet mode amplitudes. In the AMW transform for f, the

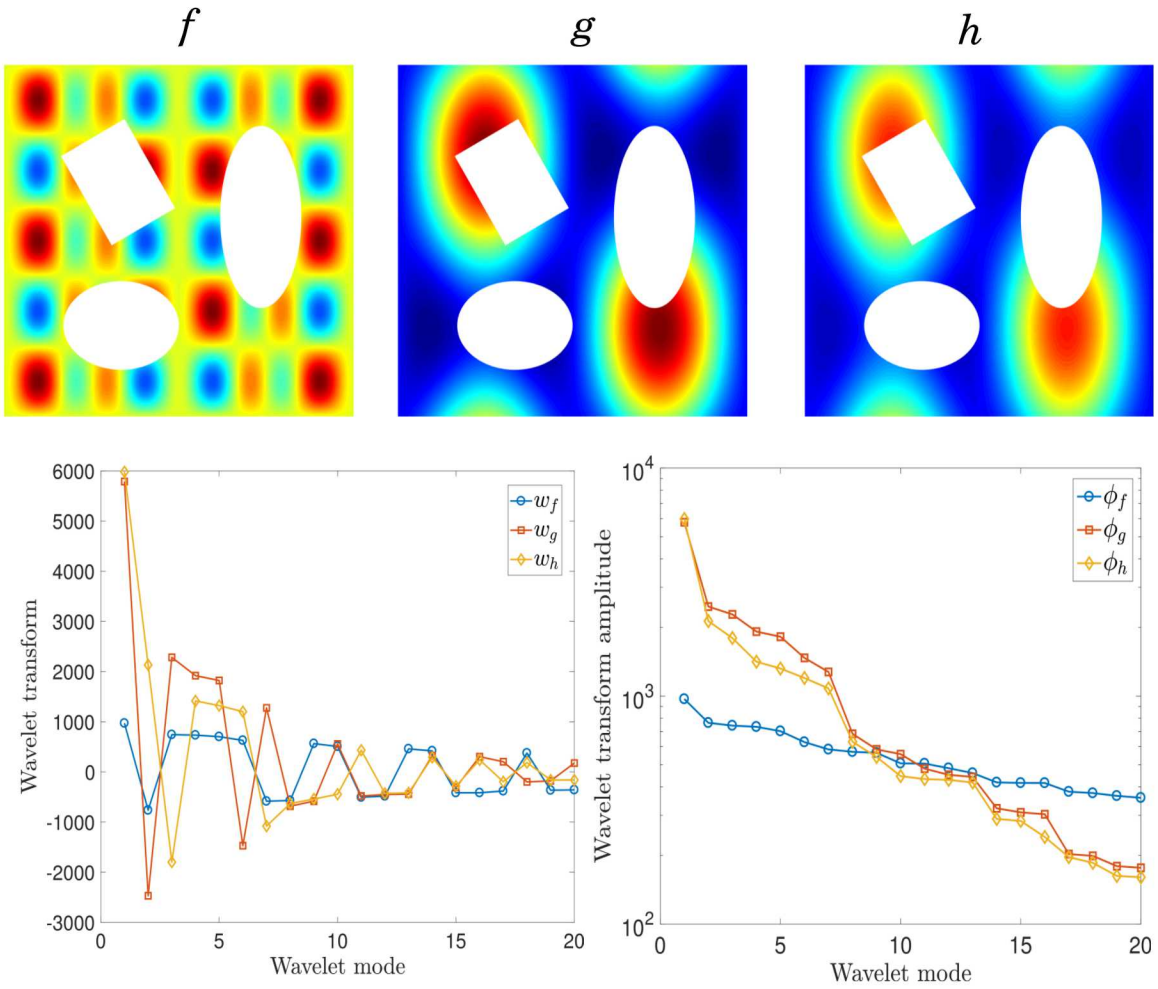

Figure 1: Plots showing (top) the functions $\boldsymbol{f}$ and $\boldsymbol{g}$ and h defined in Eq. 21 plotted on an irrugular two-dimensional geometry and the same grid, and (bottom) the first 20 modes of the wavelet transform of the considered functions performed using Alpert multiwavelets of order 5.

lower frequency amplitudes are lower than those of $\boldsymbol{g}$ and $\boldsymbol{h}$ and it exhibits a slower decay on wavelet amplitude modes. In general, the AMW for $\boldsymbol{f}$ exhibits a different spectral trend and different wavelet mode amplitudes than the functions $\boldsymbol{g}$ and $\boldsymbol{h}$. The functions $\boldsymbol{g}$ and $\boldsymbol{h}$ are similar functions that vary from each other in magnitude by approximately 30% at the maximums and 10% at their minimums. Their AMW transforms show similar trends as the wavelet mode number increases, but also shows a clear difference in wavelet mode amplitude differentiating the two data sets. Figure 2 (left) shows the normalizd relative mean square error (NRMSE) in each function when reconstructed using Eq. 4. The NRMSE decreases when more terms are kept in the functions spectra *i.e.* when less terms are truncated. This highlights the spectral content of the functions, since more wavelet modes are needed to reach the same level of error for the reconstructed image down for $\boldsymbol{f}$ when compared to the errors of $\boldsymbol{g}$ and $\boldsymbol{h}$. Other examples on the data reconstruction from the spectral form are given in [32] where it is shown how the data reconstruction quality deteriorates with larger truncations (smaller values of M).

Using the ϵ and ρ error metrics defined in Section 2.3, we compare the functions $\boldsymbol{f}$, $\boldsymbol{g}$ and $\boldsymbol{h}$ defined in Eq. 21 and represented on the irregular grid where $N_f = N_g = N_h = 33,062$. Figure 2 shows that the error between $\boldsymbol{f}$ and $\boldsymbol{g}$ is larger than the error between $\boldsymbol{g}$ and $\boldsymbol{h}$ for both metrics. For these functions, we see that that while ρ decays monotonically with the number of retained coefficients M, ϵ reached a constant value after some value of M. The quality of quickly reaching its asymptote makes the global error metric ϵ makes it an ideal choice for wavelet truncation when comparing different functions with these techniques.

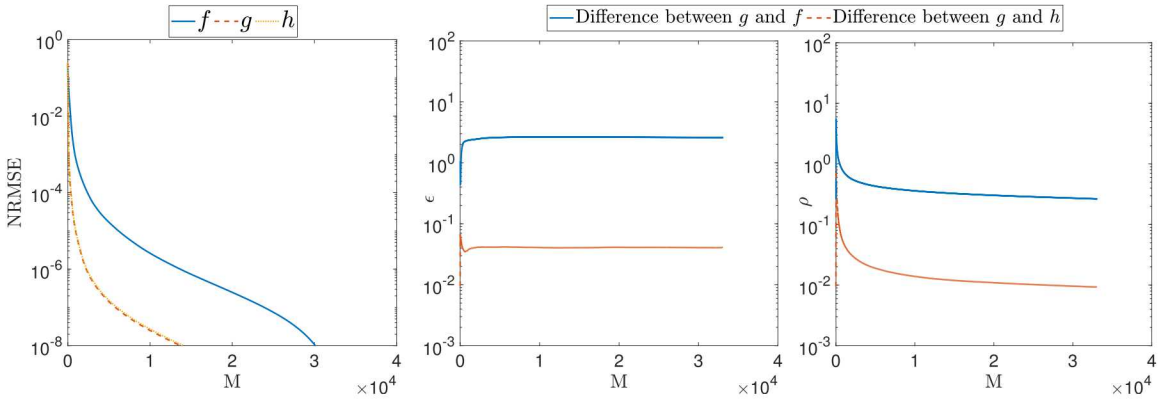

Figure 2: Plots showing (left) the normalized relative mean squared error metric (see 5) corresponding to the functions f, g and h after inverse wavelet transform, as a function of the number of retained coefficients M, and (right) the metrics ϵ and ρ comparing the functions $\boldsymbol{f}$, $\boldsymbol{g}$ and $\boldsymbol{h}$ as a function of the number of retained coefficients M. The wavelet transforms are applied with Alpert multiwavelets of order $w = 5$.

The final feature we will demonstrate is the insensitivity of the wavelet transform to scale, translation and point selection. This property of the AMW is useful for comparison between two data sets that

may not be perfectly aligned, scaled or collocated at common points. In general, this is the type of comparison that will be encountered during model validation or calibration exercises. We illustrate this property of the AMW transform by modifying the grid on which the functions were evaluated, but not changing the values at each location. Specifically, the grid is scaled by a factor of 2 and translated 1.1 units in both the x and y directions. The modified grid is also down-sampled by a factor of three so that every third point is maintained with the 1st and second points being discarded. This results in a comparison made on grids with a different number of points and different spatial sampling rates. Both the modified grid, G_2, and the original grid, G_1 are shown in Figure 3 colored according the function $\boldsymbol{g}$ evaluated on the unit square centered at 0.5. The wavelet transfrom amplitudes for $\boldsymbol{f}$ and $\boldsymbol{g}$ are also plotted in Figure 3 and show that the first 20 amplitudes for the AMW transformations on the different grids are similar, but not exact. The maximum error in mode amplitudes is 3.7% for mode 14 and the average error is 1.54%. The insensitivity to scale is due to the Gram-Schmid orthogonalization, which normalizes the mode magnitudes. The insensitivity to translation is due to the fact that the wavelet modes are calculated using polynomials in the global coordinate system using the locations of grid point relative to the other grid locations space. As a result, the actual location of the geometric centroid of the shape being analyzed does not affect the modes. The insensitivity to spatial sampling results from the multi-scale nature of the wavelet transform projecting the functions into a spectral space.

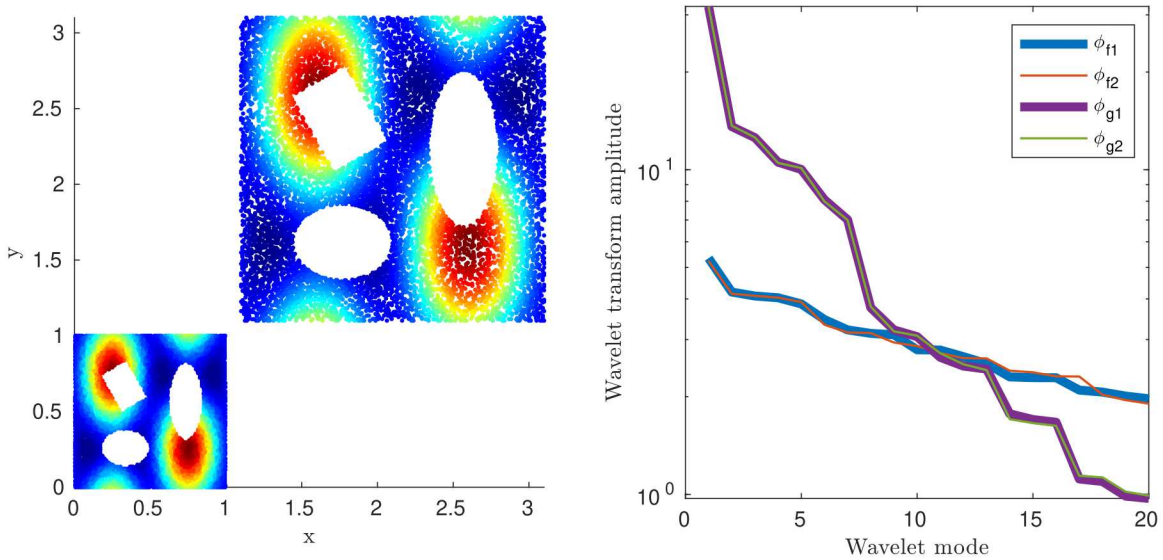

Figure 3: On the left, the function $\boldsymbol{g}$ is displayed on the two sets of points that we use to show the insensitivity of the AMW transform to scaling, translation and discretization. The plot on the right shows the resulting AMW transform amplitudes.

Note that we say the transforms are *insensitive* and not *invariant* to these translations. This is due

to the fact that the AMW is formed in a discrete rather than continuous fashion and; therefore, when performing the AMW on different collections of points, small errors are unavoidable. We believe these small errors are acceptable when comparing fields from different sources on different irregular grids, as these discretizations errors will be small relative to the actual errors in the fields. In fact, the magnitude of this discretization error can be assessed using the inverse AMW transforms. For example, the AMW transform discretization error introduced by the reduced grid size of grid G_2 is evaluated for the function $\boldsymbol{g}$ by performing the inverse AMW transform for G_1 using the ϕ_{g2} amplitudes after proper truncation and sign correction. The function $\boldsymbol{g}$ can then be reconstructed using both ϕ_{g1} and ϕ_{g2} on grid G_1. Following the notation introduced in earlier sections, the reconstructed function $\boldsymbol{g}$ on G_1 using a truncated AMW transform ϕ_{g1} is $\hat{\boldsymbol{g}}_1$ and the reconstruction on G_2 using a truncated AMW transform ϕ_{g2} is $\hat{\boldsymbol{g}}_1^2$. The error between these reconstructions is attributable to only discretization error. The relative discretization error field was calculated for the first 20 modes of the function $\boldsymbol{g}$. The discretization error from the AMW transform of $\boldsymbol{g}$ on grid G_2 is plotted on G_1 in Figure 4. The relative discretization error field is calculated using

$$e_d = \frac{||\hat{\boldsymbol{g}}_1 - \hat{\boldsymbol{g}}_1^2||_2}{[\max(\hat{\boldsymbol{g}}_1) - \min(\hat{\boldsymbol{g}}_1)]}. \tag{22}$$

For the first 20 modes, the max error is 1.16 and the average error on all points is 0.152.

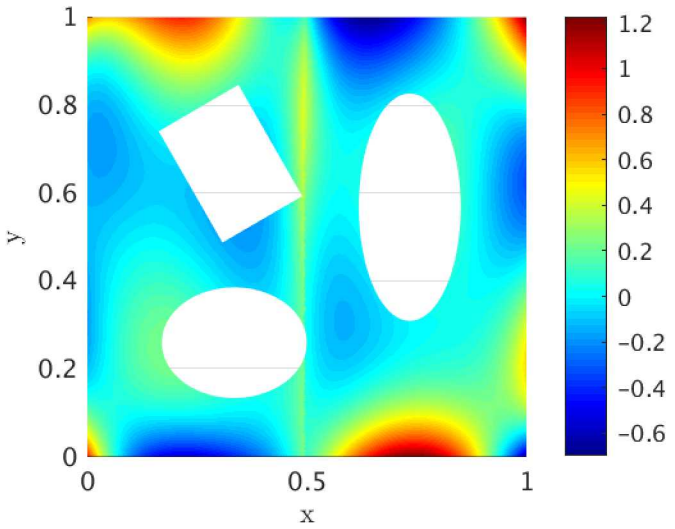

Figure 4: The relative discretization error e_d calculated using Eq. 22 on $\hat{\boldsymbol{g}}_1^2$ for the first 20 AMW transform amplitudes from $\boldsymbol{g}_2$. The reconstruction $\hat{\boldsymbol{g}}_1^2$ is the function $\boldsymbol{g}$ reconstructed on grid G_1 using the AMW transform amplitudes calculated from G_2, ϕ_{g2}.

4.2 Verification using a three-dimensional analytical function

In this section, we test our field data comparison algorithm on the two functions q_1 and q_2 given in Eq. 23. These functions are evaluated on an irregular geometry where several ellipsoidal holes are randomly drawn inside a unit cube. The functions q_1 and q_2 are discretized on this geometry using two different grids $\mathcal{M}_1$ and $\mathcal{M}_2$ defined on 68,042 and 45,176 points, respectively.

$$\begin{aligned} q_1 &= 48 \cdot [\sin(2\pi x) - \sin(2\pi y) - \sin(2.5\pi z)] \cdot \sin(2\pi x) \cdot \sin(2.5\pi z) - 52 \\ q_2 &= 38 \cdot [1.1\sin(2\pi x) - 0.9\sin(2\pi y) - 1.05\sin(2.5\pi z)] \cdot \sin(2\pi x) \cdot \sin(2.5\pi z) - 52 \end{aligned} \tag{23}$$

These two functions are plotted in Figure 5 (top) on their respective grids.

We transform these two field data vectors using Alpert wavelets and plot the first 100 largest wavelet mode amplitudes ϕ as shown in Figure 5 (bottom). The first mode amplitude is very similar for q_1 and q_2 according to Figure 5 (bottom, right), because the first mode reflects the average of the transformed functions which is similar in our case. Moreover, we notice that the trends in the remaining modes are similar but with different amplitudes. Similar to the previous example of the 2D functions g and h, this is expected because q_1 and q_2 are formed by a combination of sine and cosine functions with the same spatial frequencies (see Eq. 23) resulting in similarity in their wavelet spectrum trends. Once again, the difference in the multipliers in their equations leads to the difference in amplitude in their spectra. The wavelet coefficients w themselves are plotted in Figure 5 (bottom, left). The signs of these coefficients are inconsistent between q_1 and q_2 because they resulted from two different wavelet matrices $[\Psi]$ corresponding to the two different grids. As described in Algorithm 2, we have to adjust these signs in order to compute the error field between q_1 and q_2.

Given their analytical expressions, we compute $q_1 - q_2$ on a given grid in order to validate our wavelet error field evaluation method. Our method is capable of evaluating the error field on either $\mathcal{M}_1$ or $\mathcal{M}_2$. We plot the analytical and computed error fields on $\mathcal{M}_2$ as shown in Figure 6. These plots show very good agreement between the analytical and computed fields with both the direct and iterative methods. In order to quantify the agreement, we compute the NRMSE (see Eq. 5) between the two error fields. We also vary the number of grid points in $\mathcal{M}_1$ and $\mathcal{M}_2$ to assess the effect of the grid quality on the error field reconstruction. Table 1 shows that the normalized error resulting from the iterative method is lower than the direct method. This is due to the special feature in the iterative nethod where the combined grid is used to estimate the error field as described in Section 3.1.2, where more information is available and results in a more accurate estimation of the error field. Table 1 also shows that the error decreases with the number of grid points. In fact, the distance between grid points is larger in coarse grids inducing

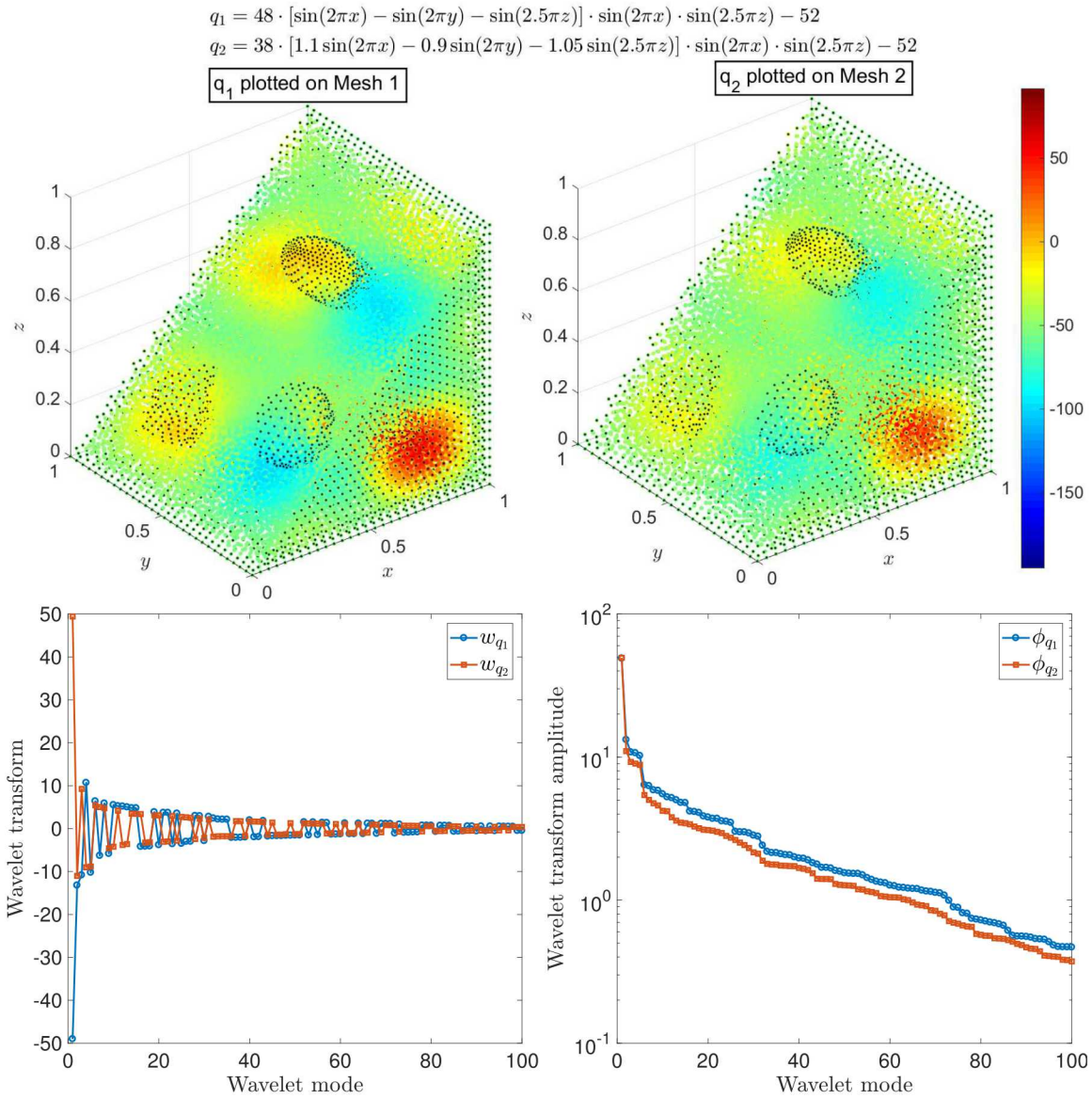

Figure 5: Plots showing (top) the functions q_1 and q_2 defined in Eq. 23 plotted on an irrugular three-dimensional geometry but on two different three dimensional grids, and (bottom) the first 100 modes of the wavelet transform of the considered functions performed using Alpert multiwavelets of order 5 and threshold factor $\theta_0 = 0.001$ (see Eq. 9). The black dots are used to visualize the holes.

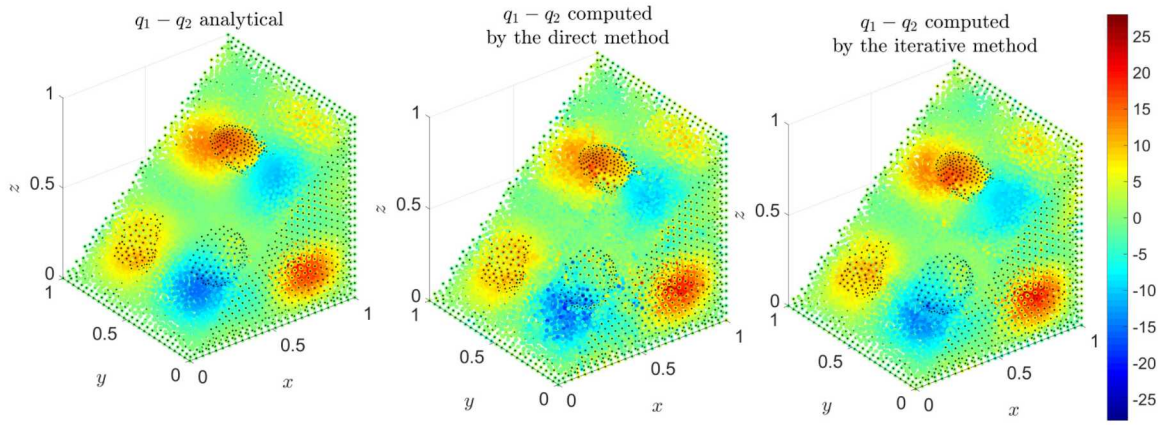

Figure 6: Plots showing (left) the analytical error field between q_1 and q_2 given in Eq. 23, (center) the error field computed using the direct method described in Section 3.1.1, and (right) the computed error field using the iterative method given in Algorithm 2 with a threshold factor $\theta_0 = 0.01$ (see Eq. 9). Shown are error fields plotted on $\mathcal{M}_2$ where the computed field is obtained by subdividing each domain into 8 subdomains.

a larger interpolation error. This is reflected in wavelets through the loss of field information with less wavelet coefficients in coarser grids.

Table 1: NRMSE between the analytical and computed error fields $q_1 - q_2$ as a function of the number of grid points in $\mathcal{M}_2$. The RMS is normalized by $\max(q_1 - q_2) - \min(q_1 - q_2)$. Results are obtained using Alpert multiwavelets of order 5 and threshold factor $\theta_0 = 0.001$ (see Eq. 9).

$N_{\mathcal{M}_2}$	11,294	22,588	33,882	45,176
$N_{\mathcal{M}_1}$	17,010	34,021	51,031	68,042
L_2 Normalized error (direct)	0.0587	0.0516	0.0446	0.0390
L_2 Normalized error (iterative)	0.0357	0.0303	0.0275	0.0234

In order to further confirm the ability of our algorithms to accurately estimate the error field between two functions, we compare q_1 with another function p_2 given in Eq. 24 and defined on the mesh $\mathcal{M}_2$ but exhibiting variations and features of higher frequencies.

$$p_2 = 48 \cdot [\sin(5\pi y) - 3\sin(3\pi x) - 2\sin(4.5\pi z)] \cdot \sin(4\pi y) \cdot \sin(5.5\pi x) \cdot \sin(3.5\pi z) - 52 \tag{24}$$

The error fields are plotted in Figure 7 which shows good agreement between the analytical error field and the one estimated by both the direct and iterative methods.

We have validated in these results the ability of AMW to accurately estimate the difference between two data fields. The major benefits of the AMW method over other methods are as follows. It can be applicable to field data represented on any mesh or point-cloud, and any irregular geometry. It incurs automatic noise filtering because it relies on the truncation of the wavelet spectrum of the data. It is also robust to scaling and translation in geometries often encountered in experimental data fields.

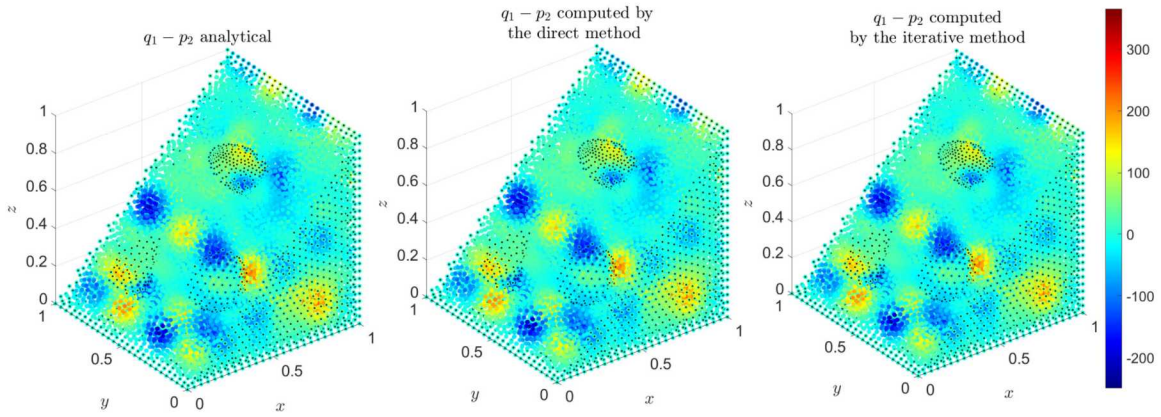

Figure 7: Plots showing (left) the analytical error field between q_1 and p_2 given in Eqs. 23 and 24, (center) the error field computed using the direct method described in Section 3.1.1, and (right) the computed error field using the iterative method given in Algorithm 2 with a threshold factor $\theta_0 = 0.01$ (see Eq. 9). Shown are error fields plotted on $\mathcal{M}_2$ where the computed field is obtained by subdividing each domain into 8 subdomains.

4.3 An Application of AMW Field Comparison to Three-dimensional Finite Element Results

We now apply our methods to a strain field obtained by simulating the compression of an elastomeric syntactic foam block that contains voids with irregular shapes and locations [48]. The computational 3D domain consists of 4,095,763 mesh points distributed among 1,584 processes. The simulation computes several the three-dimensional strain fields in the foam block at several time steps where the total simulation time is equal to 0.25 seconds. We focus in our study on the strain in the z direction s_{zz} shown in Figure 8.

We divide the mesh points into two meshes $Mesh1$ and $Mesh2$ consisting of 1,364,732 and 2,731,031 points, respectively. We seek to compare two discretizations of the strain fieldss_{zz}, s_{zz1} and s_{zz2} represented on $Mesh1$ and $Mesh2$, respectively. For the sake of validation of our method, we compute this error field for the same time step. Theoretically, this error field should be equal to zero. However, due to the differences in the meshes, small discretization errors will occur as discussed in Section 4.1. We apply the adaptive iterative method described in Sections 3.1.2 and 3.2 in parallel on each of the 1,584 computing processes. Within each process, we consider 8 subdomains in order to enhance the comparison speed as described in Section 3.3.

Figure 9 shows a slice plot of s_{zz} on $Mesh1$ and $Mesh2$ along with their error fields plotted on both meshes. We first notice that the largest errors are concentrated in the regions of large gradients. This is expected since large amplitudes require smaller thresholds to be accurately captured [32]. We also notice in Figure 9 that the errors are smaller on $Mesh1$. In fact, $Mesh1$ contains less points thus it is less

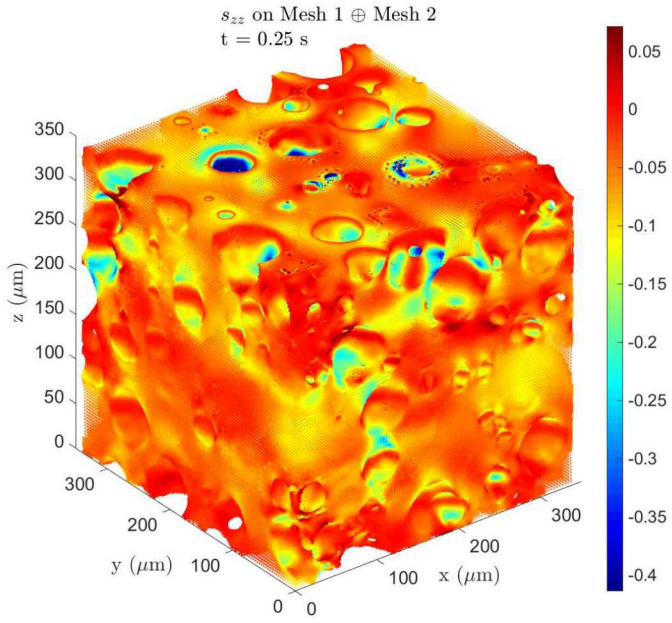

Figure 8: Plots showing the strain in the z direction s_{zz} at the last time step t=0.25s plotted at the full computational mesh $Mesh1 \oplus Mesh2$.

likely that it contains data of large gradients thus it has less regions of larger errors. For quantitative assessment of the error field, we plot its distribution as shown in Figure 10 where we can see that the mode of the normalized relative error is less than 1% and the 95^{th} percentile is less than 5% for both mesh representations.

We now compare s_{zz} fields from two different times t_1 and t_2 and represented on $Mesh1$ and $Mesh2$, that is, $s_{zz}(t_1)$ and $s_{zz}(t_2)$ are represented on $Mesh1$ and $Mesh2$, respectively. The results show very good agreement between the estimated and true error fields with 95^{th} percentile error less than 5% for all considered times.

5 Conclusions

In this paper we have introduced a method to compare full-field data of N dimensions using an AMW transform on the data and making the comparison in a spectral space. We have shown that the transforms are *truly* insensitive to translation, scale and discretization on arbitrary geometry. This makes them especially well suited for comparison of field-data sets coming from two different sources such as when comparing simulation field data to experimental field data. The primary benefits of using our methods are:

1. The AMW transform can be performed on arbitrary geometry containing N dimensional data.

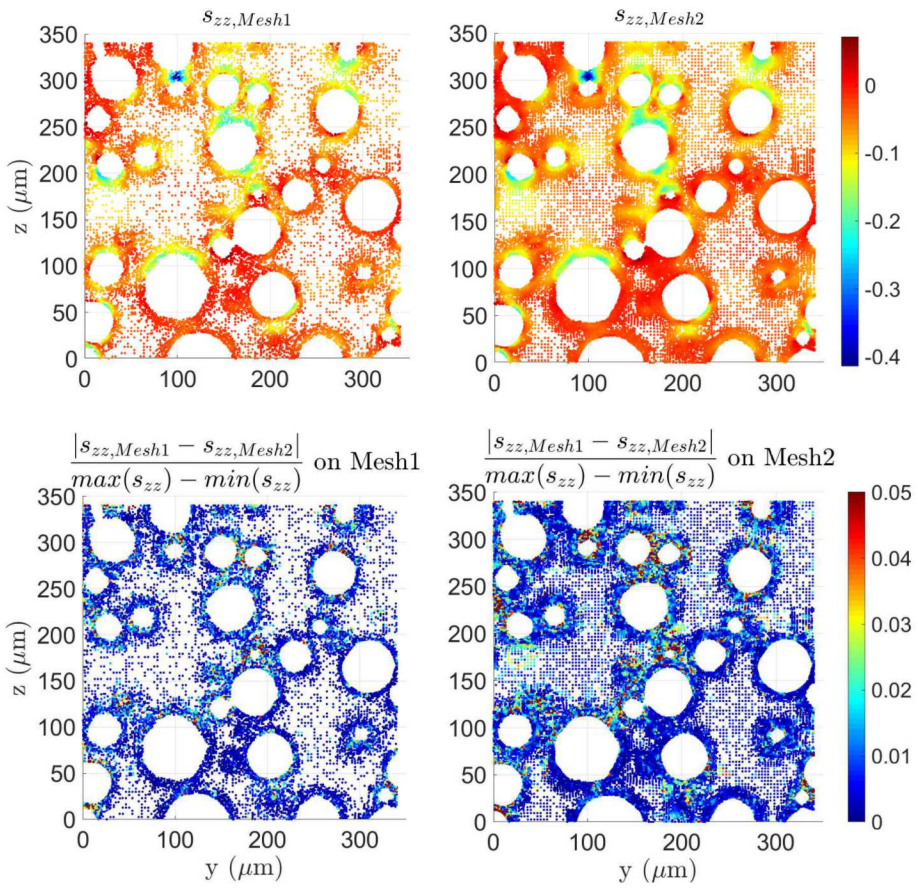

Figure 9: Plots showing (top row) a slice plot of s_{zz} for $x = 40.7\mu$m plotted on $Mesh1$ and $Mesh2$, and (bottom) the error field between these two representation plotted on both meshes. Results obtained using the adaptive iterative method ran in parallel with 8 subdomains in each computing process with a threshold multiplier $\theta_0 = 0.01$.

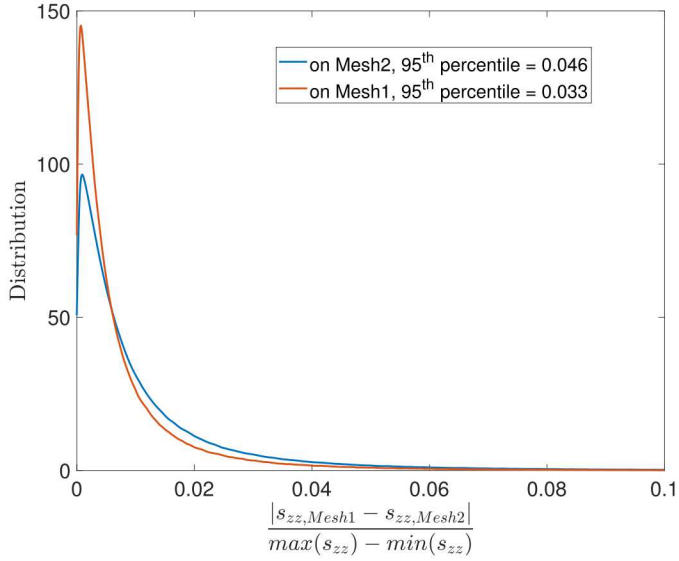

Figure 10: Plots showing the distribution of the error field computed on $Mesh1$ and $Mesh2$, as indicated. Results obtained using the adaptive iterative method ran in parallel with 8 subdomains in each computing process with a threshold multiplier $\theta_0 = 0.01$.

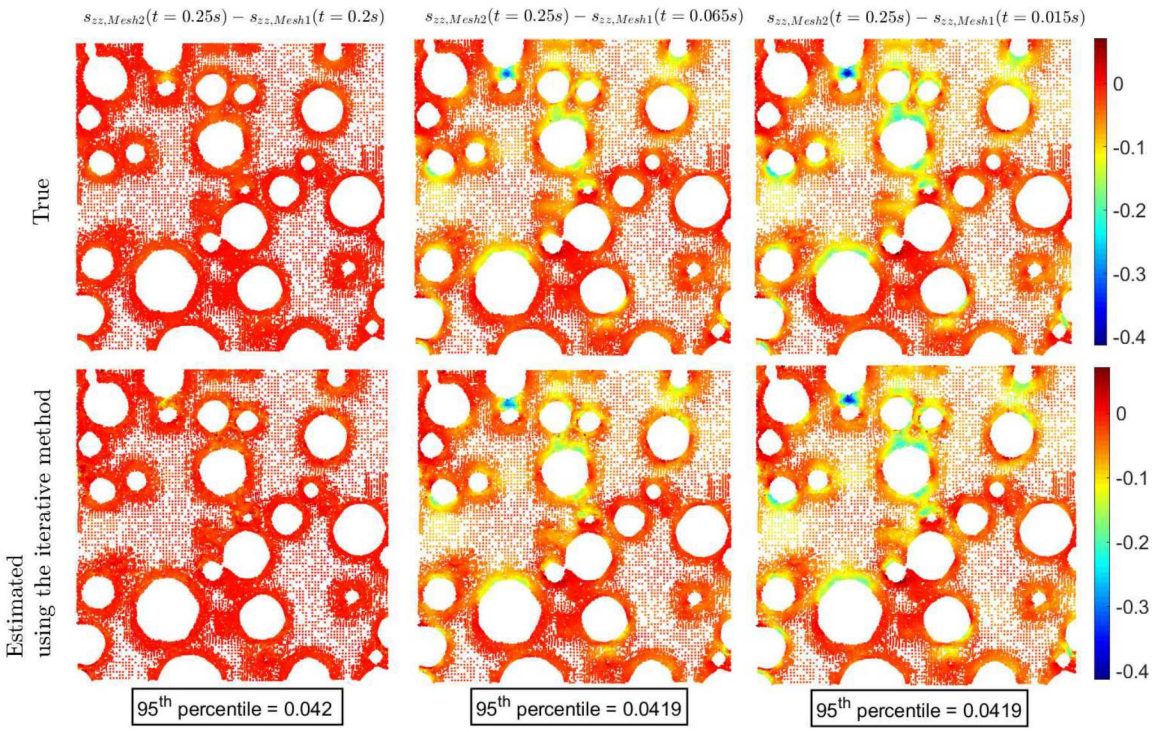

Figure 11: Plots showing true and estimated error fields in the strain s_{zz} resulting in comparison at different times and at the two $Mesh1$ and $Mesh2$ representations, as indicated. Results obtained using the adaptive iterative method ran in parallel with 8 subdomains in each computing process with a threshold multiplier $\theta_0 = 0.01$.

2. The comparison of fields in the AMW transform spectral space is insensitive to translation, scale and discretization.

3. The AMW transform and comparison algorithms can be easily parallelized to decrease the computational cost of comparing large data sets.

4. Wavelets have a long history in signal processing and filtering. This can be used to remove noise from data sets and compress the data sets to reduce storage burdens.

Potential short comings of this method include discretization errors, sensitivity to rotations and computational cost. As for the discretization error, we have shown that the inherent discretization error that is unavoidable with this method is small, bounded and quantifiable. In general, numerical error occurs for most of the methods reviewed in the Section 1 whether it be integration errors for Zernike moment methods or discretization errors for Tchebichef moments methods. Effectively accounting for the rotations requires solving an inverse problem [49] that minimizes a global field error. While the sensitivity to rotations of the AMW transform is unavoidable using the methods proposed here, the computational cost can be reduced through the use of parallel code and sparse matrix algorithms. Overall, we believe the benefits of the method warrant its consideration as a tool for general, quantifiable full-field data comparison.

6 Acknowledgments

This work was supported by the Advanced Simulation and Computing (ASC) Program of the Department of Energy.

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] Michel Grediac and Francois Hild. *Full-Field Measurements and Identification in Solid Mechanics*. John Wiley & Sons, New York, 1 edition, 2013. 2

[2] L Mitas and H Mitasova. Spatial interpolation. In P.A. Longley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind, editors, *Geographic Information Systems: Principles, Techniques, Management and Applications*, chapter 34. John Wiley & Sons, Hoboken, N.J., 2 edition, 2005. 2

[3] Alejandro Mota, WaiChing Sun, Jakob T. Ostien, James W. Foulk, and Kevin N. Long. Lie-group interpolation and variational recovery for internal variables. *Computational Mechanics*, 52(6):1281–1299, Dec 2013. 2

[4] Weizhuo Wang, John E Mottershead, and Cristinel Mares. Vibration mode shape recognition using image processing. *Journal of Sound and Vibration*, 326(3):909 – 938, 2009. 3

[5] R. B. Berke, C. M. Sebastian, R. Chona, E. A. Patterson, and J. Lambros. High temperature vibratory response of hastelloy-x: Stereo-dic measurements and image decomposition analysis. *Experimental Mechanics*, 56(2):231–243, Feb 2016. 3

[6] A.C. Santos Silva, C.M. Sebastian, J. Lambros, and E.A. Patterson. High temperature modal analysis of a non-uniformly heated rectangular plate: Experiments and simulations. *Journal of Sound and Vibration*, 443:397 – 410, 2019. 3

[7] W. Wang, J.E. Mottershead, C.M. Sebastian, and E.A. Patterson. Shape features and finite element model updating from full-field strain data. *Int. J. Solids. Struct.*, 48:1644–1657, 2011. 3, 4

[8] Y. Chen. *A Non-Model Based Expansion Methodology for Dynamic Characterization*. PhD thesis, University of Massachusetts Lowell, November 2019. 3

[9] G.A. Papakostas, Y.S. Boutalis, C.N. Papaodysseus, and D.K. Fragoulis. Numerical error analysis in zernike moments computation. *Image and Vision Computing*, 24(9):960 – 969, 2006. 3

[10] Chandan Singh and Ekta Walia. Fast and numerically stable methods for the computation of zernike moments. *Pattern Recognition*, 43(7):2497 – 2506, 2010. 3

[11] R. Mukundan, S. H. Ong, and P. A. Lee. Image analysis by tchebichef moments. *IEEE Transactions on Image Processing*, 10(9):1357–1364, Sep. 2001. 3

[12] P. . Yap, R. Paramesran, and Seng-Huat Ong. Image analysis by krawtchouk moments. *IEEE Transactions on Image Processing*, 12(11):1367–1377, Nov 2003. 3

[13] P. Yap, R. Paramesran, and S. Ong. Image analysis using hahn moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):2057–2062, Nov 2007. 3

[14] Hongqing Zhu, Huazhong Shu, Jian Zhou, Limin Luo, and J.L. Coatrieux. Image analysis by discrete orthogonal dual hahn moments. *Pattern Recognition Letters*, 28(13):1688 – 1704, 2007. 3

[15] Hongqing Zhu, Huazhong Shu, Jun Liang, Limin Luo, and Jean-Louis Coatrieux. Image analysis by discrete orthogonal racah moments. *Signal Processing*, 87(4):687 – 708, 2007.

[16] A. S. Patki and E. A. Patterson. Decomposing strain maps using fourier-zernike shape descriptors. *Experimental Mechanics*, 52(8):1137–1149, Oct 2012. 3

[17] Christopher Sebastian, Erwin Hack, and Eann Patterson. An approach to the validation of computational solid mechanics models for strain analysis. *The Journal of Strain Analysis for Engineering Design*, 48(1):36–47, 2013. 3

[18] G. Lampeas, V. Pasialis, X. Lin, and E.A. Patterson. On the validation of solid mechanics models using optical measurements and data decomposition. *Simulation Modelling Practice and Theory*, 52:92 – 107, 2015. 3, 4

[19] Richard J Prokop and Anthony P Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54(5):438 – 460, 1992. 3

[20] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962. 3

[21] A. P. Reeves, R. J. Prokop, S. E. Andrews, and F. P. Kuhl. Three-dimensional shape analysis using moments and fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):937–943, Nov 1988. 3

[22] Whoi-Yul Kim and Yong-Sung Kim. A region-based shape descriptor using zernike moments. *Signal Processing: Image Communication*, 16(1):95 – 102, 2000. 3

[23] R. Mukundan. Radial tchebichef invariants for pattern recognition. In *TENCON 2005 - 2005 IEEE Region 10 Conference*, pages 1–6, Nov 2005. 3

[24] Haiyong Wu and Senlin Yan. Computing invariants of tchebichef moments for shape based image retrieval. *Neurocomputing*, 215:110 – 117, 2016. SI: Stereo Data. 3

[25] A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, May 1990. 3

[26] J.W. Ngan, C.C. Caprani, and Y. Bai. Full-field finite element model updating using zernike moment descriptors for structures exhibiting localized mode shapes. *Mech. Sys. Sig. Proc.*, 121:373–388, 2019. 4

[27] W. Wang, J.E. Mottershead, T. Siebert, and A. Pipino. Frequency response functions of shape features from full-field vibration measurements using digital image correlation. *Mech. Sys. Sig. Proc.*, 28:333–347, 2012. 4

[28] Y-H. Chang, W. Wang, T. Siebert, J-Y Chang, and J.E. Mottershead. Basis-updating for data compression of displacement maps from dynamic dic measurements. *Mech. Sys. Sig. Proc.*, 115:405–417, 2019. 4

[29] A. Buljac, C. Jailin, A. Mendoza, J. Neggers, T. Taillandier-Thomas, A. Bouterf, B. Smaniotto, F. Hild, and S. Roux. Digital volume correlation: Review of progress and challenges. *Exp. Mech.*, 58:661–708, 2018. 4

[30] J. Neggers, O. Allix, F. Hild, and S.Roux. Big data in experimental mechanics and model order reduction: Todays challenges and tomorrows opportunities. *Arch. Computat. Methods. Eng.*, 25:143–164, 2018. 4

[31] 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. 4, 5

[32] M. Salloum, K.L. Johnson, J.E. Bishop, J.M. Aytac, D. Dagel, and B.G. van Bloemen Waanders. Adaptive wavelet compression of large additive manufacturing experimental and simulation datasets. *Computational Mechanics*, 63(3):491–510, 2019. DOI: 10.1007/s00466-018-1605-6. 4, 6, 7, 9, 17, 23

[33] D.M. Radunovic. *Wavelets from Math to Practice*. Springer, 2009. 5, 32

[34] Mohlenkamp M.J. and Pereyra M.C. *Wavelets, Their Friends, and What They Can Do for You*. European Mathematical Society, 2008. 5, 10

[35] Frazier M.W. *An Introduction to Wavelets Through Linear Algebra*. Springer, 1999. 5

[36] 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. 5, 6, 33, 34

[37] 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.*, 3(1):1–23, 2018. DOI: 10.1007/s41019-017-0042-4. 5, 6, 9, 34

[38] 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. 6, 10, 33, 34

[39] S. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38(2):617–643, March 1992. 7

[40] M. Vetterli and C. Herley. Wavelets and filter banks: theory and design. *IEEE Transactions on Signal Processing*, 40(9):2207–2232, Sep. 1992. 7

[41] Yansun Xu, J. B. Weaver, D. M. Healy, and Jian Lu. Wavelet transform domain filters: a spatially selective noise filtration technique. *IEEE Transactions on Image Processing*, 3(6):747–758, Nov 1994. 7

[42] Strang G. and Nguyen T. *Wavelets and Filter Banks*. Wesley-Cambridge Press, 1996. 7

[43] R. Bro, E. Acar, and T.G. Kolda. Resolving the sign ambiguity in the singular value decomposition. *J. Chemometrics.*, 22(2):135–140, 2008. 10

[44] M. Salloum, A. Alexanderian, O.P.Le Matre, H.N. Najm, and O.M.Knio. Simplified csp analysis of a stiff stochastic ode system. *Computer. Meth. App. Mech. Eng.*, 217:121–138, 2012. 10

[45] Hastie T., Tibshirani R., and Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2017. 14

[46] James G., Witten D., Hastie T., and Tibshirani R. *An Introduction to Statistical Learning with Applications in R*. Springer, 2013. 14

[47] Swinzip v2.0: A matlab and C++ library for scientific lossy data compression and reconstruction using compressed sensing and tree-wavelets transforms.

https://github.com/msalloum80/swinzip/tree/master/swinzip-v2.0, Sandia National Laboratories, 2020. 15

[48] J.A. Brown, J.D. Carroll, B. Huddleston, Z. Casias, and K.N. Long. A multiscale study of damage in elastomeric syntactic foams. *J. Mater. Sci.*, 53(14):10479–10498, 2018. 23

[49] Y. Heider, K. Wang, and W. Sun. SO(3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials. *Comp. Meth. App. Mech. Eng.*, 363, 2020. 27

[50] M. Jansen and P. Oonincx. *Second Generation Wavelets and Applications*. Springer, 2005. 33

[51] W. Sweldens. The lifting scheme: a construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998. 33

[52] M. Maggioni, J.C. Bremer, R.R. Coifman, and A.D. Szlam. Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs. In *Proc. SPIE 5914, Wavelets XI, 59141M*, San Diego, USA, September 2005. 33

A Appendix: Alpert Multi-Wavelets

Multi-wavelets and wavelets can be different types of bases where multiresolution functions can be represented by linear combinations. Wavelets encode all the scales in the data and their locations in space and/or time. Initial wavelets were developed for data defined on regular grids such as signal, image and video data. They are referred to as First generation wavelets (FGW) given by the approximation basis $\phi(x)$ and wavelet basis $\psi(x)$ to find the details in a function [33] following:

$$\phi_j(x) = \sum_{k\in\mathbb{Z}} a_k \phi_{j-1}(2x-k) \tag{25}$$

$$\psi_j(x) = \sum_{k\in\mathbb{Z}} b_k \phi_{j-1}(2x-k) \tag{26}$$

where x is the spatio-temporal coordinate, j and k are the detail level and location indices, respectively, and a_k and b_k are coefficients intrinsic to the FGW type. Each wavelet basis $\psi_j(x)$ models a finer resolution detail as j increases. Regular grids are dyadic such that the maximum number of detail levels $j_{\max}$ is equal to $\log_2(N)$ where N is the number of grid points in each dimension.

FGWs are not a suitable representation in our work because we analyze data represented on unstructured meshes [50]. Instead, non-traditional wavelet bases such as second generation wavelets [50, 51], diffusion wavelets [52] or Alpert multi-wavelets (AMW) [36, 38] are required; we use the latter type in in our work.

These types of wavelets have similar characteristics to FGWs. While Eqs. 25 and 26 are also used used to compute AMWs, there are two main differences between FGW and AMWs. First, $j_{\max}$ is computed by recursive splitting of the non-dyadic mesh into separate subdomains that form a multiscale hierarchical tree [50]. Thus, AMWs can accommodate non-dyadic grids such as finite intervals and irregular geometries. Second, the AMW bases are not based on constant coefficients a_k and b_k as in FGWs but they are calculated according to the mesh coordinates x_k.

The major advantage of AMWs is they do not require any special treatment of irregular boundaries (*e.g.* holes) present in the domain and they avoid them by construction [36]. The discrete Alpert wavelets $\psi_j(x_k)$ are polynomials that are easy to compute; they are represented in a square sparse matrix $[\Psi]$.

A.1 Building the Wavelets Matrix

It is unclear how to compute $[\Psi]$ using Eqs. 25 and 26 in a systematic and practical manner. Instead, we employ a discrete methodology to build the Alpert wavelet matrix. We briefly present a technique for the case of one-dimensional (1D) non-uniform mesh for a polynomial order w. The 1D mesh is constituted of N points $x_1 < \ldots < x_i < \ldots < x_N$.

First, the mesh is subdivided into P almost equally sized bins where the number of points n per bin is $2w \geq n \gtrsim w$. This subdivision operation is not required when FGW are used in a dyadic grid.

Second, the so-called initial moment matrices $[M]_{1,1\leq p\leq P} \in \mathbb{R}^{n\times w}$ are computed for each bin:

$$[M]_{1,p} = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{w-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{w-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{w-1} \end{bmatrix} \tag{27}$$

$[M]_{1,p}$ contain the wavelet functions ψ (see Eq. 26), assumed to be polynomials in AMW [38]. We also compute the matrices $[U]_{1,p} \in \mathbb{R}^{n\times n}$ for each bin by orthogonalizing the matrices $[M]$ (*e.g.* using a QR operation).

Third, the matrices $[U]_{1,p}$ are assembled in a large matrix $[V]_1 \in \mathbb{R}^{N\times N}$ as illustrated in Figure 12 (left column). The columns of the matrix $[V]$ correspond to the N mesh points and the N rows correspond

to the N wavelet coefficients.

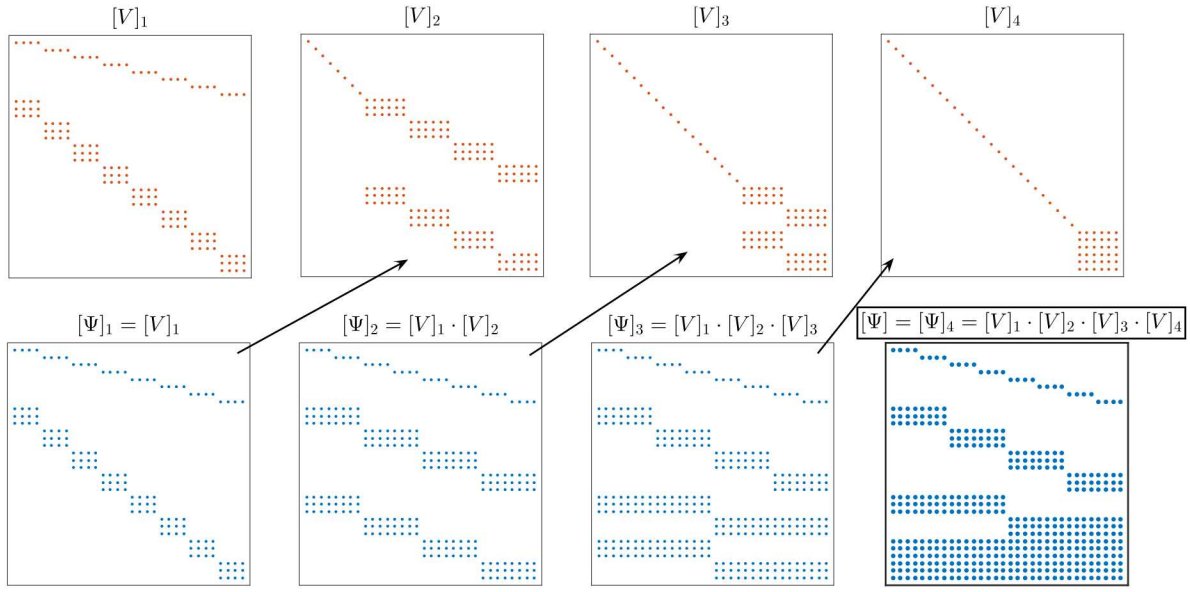

Figure 12: Representations of the sparse Alpert wavelet matrices at different detail levels j. For each level j, the wavelet matrix $[\Psi]_j$ is the cumulative product of the matrices $[V]_{i \leq j}$. Shown are matrices computed for a one-dimensional mesh containing $N = 32$ points where the wavelet order is $w = 3$ such that the mesh is divided into $P = 8$ bins and exhibits $j_{max} = 4$ detail levels. The mesh is not necessarily dyadic.

Fourth, the finer detail levels in the mesh are accounted for. The P bins form a hierarchy of the details where $j_{max} = \log_2(P) + 1$ is the maximum number of detail levels. Matrices $[V]_{2 \leq j \leq j_{max}}$ are computed similarly to $[V]_1$ and the full wavelet matrix as is obtained as:

$$[\Psi] = \prod_{j=1}^{j_{\max}} [V]_j \tag{28}$$

More details on these four steps are given in [36–38]. This procedure can be generalized to D-dimensional meshes. The procedure described above to build the wavelet matrix $[\Psi]$ guarantees that it is orthonormal such that its matrix inverse is equal to its transpose, $[\Psi]^{-1} = [\Psi]^T$.