2013-01-01

Reduced-Order Modeling Using Orthogonal and Bi-orthogonal Wavelet Transforms

Miguel Hernandez Iv
University of Texas at El Paso, miguel.hernandez4@gmail.com

REDUCED-ORDER MODELING USING ORTHOGONAL AND BI-ORTHOGONAL WAVELET TRANSFORMS

MIGUEL HERNANDEZ IV

Computational Science Program

APPROVED:

_____

Miguel Argaez, Ph.D., Chair

_____

Sergio D. Cabrera, Ph.D.

_____

Hugo Sandoval, Ph.D.

_____

Benjamin Flores, Ph.D.

Dean of the Graduate School

Dedicated with all my heart and love to

Erika, Miguel V., and Nadia.

REDUCED-ORDER MODELING USING ORTHOGONAL AND BI-ORTHOGONAL WAVELET TRANSFORMS

by

MIGUEL HERNANDEZ IV, M.S.C.P.S., M.S.E.E., B.S.E.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Computational Science Program

THE UNIVERSITY OF TEXAS AT EL PASO

December 2013

Acknowledgments

Many people have participated both directly and indirectly to this thesis. Here, I acknowledge your contributions. Beginning with Miguel Argaez, thank you for introducing me to the world of numerical optimization. That single class has influenced me more than any other. Your ability to break down a complex problem into its fundamental components and the patience you have while explaining each component are inspiring. I thank you for welcoming me into your research team.

To Leticia Velazquez, thank you for your support and for welcoming me into the Computational Science Program. I don't think either of us knew that a simple email asking if I knew how to program in a Linux environment would develop into research in reduced order models with wavelets. I thank you for your understanding, support, and probing questions.

To Reinaldo Sanchez, thank you for your insights and suggestions, all of which served to improve this body of work. You also have a gift for taking complex problems and concepts and simplifying them. You will make an excellent educator should you choose to enter academia.

To my son Miguel V: You are the reason I went back to school and started working on this Ph.D. Your birth set in motion a chain of events which enabled me to make a career change and to be in the right place at the right time. You gave me the courage to complete this degree, and I am overjoyed and proud that you will witness this accomplishment. You continue to inspire me with your love for life, energy, and sense of humor, and I can't wait to see the man you will become. I love you.

To my daughter Nadia: Although you are still a baby and will not remember this event, we are blessed to have you with us. December 2013 has a special significance to me, much more than the Ph.D I will earn because this month is your birthday and marks the first year of your life. Mommy, Daddy, and Miggy will always remember the year and circumstances under which you were born. Your smile and infectious laugh melt my heart, and although I know "It Won't Be Like This For Long," you will always be my little girl. I love you.

And now for the most difficult part, to my beautiful wife Erika: Words cannot begin to describe

how much I appreciate the love and endless support you have provided me throughout this journey. To say ”thank you” does not do you justice. And while the words escape me, I hope to show my deep gratitude with actions and pledge to be the best husband and father I can. I promise to spend more time with you and our family as we enter this new chapter in our lives and continue “Making Memories of Us.” I will forever love you.

Erika, Miguel V and Nadia, this dissertation is dedicated to you with all my heart.

Abstract

It is well known that model reduction methods borrow techniques typically found in data compression, and current state-of-the-art techniques for data compression are based on the wavelet transform. Given these facts, it is surprising that model reduction using wavelets has not received much attention and has not been adequately addressed in the literature. This research seeks to determine if wavelets can be used for model reduction and if wavelet model reduction is a viable alternative to existing model reduction methods.

In this work we propose a novel method for model reduction using wavelets. Specifically, we introduce techniques for deriving wavelet reduced-order models for solving inverse problems. Algorithms are developed for both orthogonal and bi-orthogonal wavelets, and two methods are proposed using Galerkin and Petrov-Galerkin wavelet reduced-order models. Also, we propose a computationally efficient method for wavelet model reduction using a reduced filter bank structure that has $\mathcal{O}(n)$ complexity and further reduces the time required for online computations.

To evaluate the performance of the wavelet reduced-order models, we apply our methods to the nonlinear Burgers' partial differential equation. The numerical results are then compared to model reduction based on the proper orthogonal decomposition (POD) and the full-order model.

We conclude that wavelet model reduction is an alternative to the POD and can enable the researcher to begin and execute online, real-time simulations faster while being memory efficient and simple to design. Further, wavelet model reduction does not require costly offline computations or snapshots. This gives model reduction with wavelets the additional advantage of data independence, meaning that the wavelet basis does not need to be re-computed as the full-order model changes.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Inverse, nonlinear least-squares and parameter estimation problems have received much attention in diverse fields of science and engineering. Medical imaging [22, 17], geophysics [31], heat transfer [4], quantum, wave, and electromagnetic scattering [41], and electrical impedance tomography [13] are a few examples. In all of these fields, it has been shown that reliable parameter estimation is a cornerstone for improving predictions and reducing the risk of application decisions.

Processes and applications such as those previously mentioned are described primarily using mathematical models which simulate and/or control the underlying behavior of the systems. In this framework, there is an ever increasing need for improved model accuracy which leads to models of higher complexity. However, as the size of the problem and the number of parameters to estimate increases, the computational cost involved in obtaining a solution also increases. Realistic models can have millions of equations and unknowns that need to be satisfied. Simplification of the model and the underlying parameters is thus needed in order to perform simulations within an acceptable amount of time and to cope with limited memory and storage capacity among other requirements.

To reduce the computational cost of the so-called full-order model (FOM), model reduction methods have been developed that approximate the FOM and simplify computations by reducing the dimensions of the system in question [2, 8, 14, 26]. Model reduction in this context is defined as a systematic approach for deriving computationally efficient representations of large-scale systems that result from the discretization of partial differential equations (PDEs).

One of the most popular, state of the art techniques for model reduction is the proper orthogonal decomposition (POD) which is based on the singular value decomposition (SVD) [7, 21, 27, 30, 40]. The idea behind the POD is as follows: The high dimensional FOM is solved and m measurements (commonly referred to as samples or snapshots) are taken of the n state variables

(or model parameters). These snapshots are then arranged into an $n \times m$ matrix A which is then factored using the singular value decomposition. Since the singular values are ordered in descending order $\sigma_1 \geq \cdots \geq \sigma_n$, $r < n$ column vectors are selected which correspond to the r largest singular values and are used to form a basis which is in turn used for model reduction [9].

While the POD is one of the most popular approaches to model reduction, it has several disadvantages. First: The POD requires a series of offline computations in order to form the snapshot matrix. These snapshots are computed by solving the computationally expensive FOM for selected values of the parameters and inputs. The quality of the resulting reduced-order model depends heavily on the choice of parameters and inputs over which the snapshots are computed. Thus, without a "good" snapshot matrix, the reduced-order model obtained may not provide solutions representative of the FOM. A key question, then, is how to sample and choose the parameters/inputs over which to compute the reduced basis. One approach is to use subject matter experts and their deep knowledge of the application at hand to determine representative inputs and associated snapshots. The challenge of sampling to produce the snapshots is the subject of ongoing research [3, 6, 16, 18].

Second: It can take an extended amount of time to solve the FOM, generate good snapshots, and apply the POD before a reduced basis is formed and a reduced model is obtained prior to beginning real-time simulations. The computational savings gained during the online phase in which the reduced-order model is used is thus offset by the possibly large investment in time during the offline computation phase.

Third: Since the POD relies on snapshots formed from a FOM, this method is data dependent, that is, the basis used for model reduction cannot be used to reduce a different FOM [5]. This is not amenable to "what if" experimentation where arbitrary system excitations and boundary conditions can be applied to gain a deeper understanding of the problem at hand. This type of experimentation requires us to go through the POD with each of the disadvantages listed previously for each scenario to be considered. The common theme here are the need for snapshots, and this leads to the first question this research aims to answer: Does an approach exist that does not rely on snapshots for model reduction?

In [8], model reduction using the POD is referred to as a form of data compression. In fact, model reduction using the POD borrows heavily from ideas traditionally found in the areas of image processing, data compression and transform coding where singular values are used to approximate and compress an image, i.e. the FOM. Although model reduction borrows techniques typically found in data compression, it is very surprising that model reduction using wavelets has not received much attention and has not been adequately addressed in the literature. The use of the wavelet transform in image processing is by far the most popular and practical technique used in many applications including JPEG 2000, digital cameras, high definition video, and medical imaging among others [19, 20, 24]. This observation leads to the second question this research aims to answer: Can wavelets be used for model reduction? To this end, in this work we propose methods for model reduction using wavelets.

Chapter 2

Problem Formulation

Consider a square nonlinear system of equations that represents a FOM and is expressed by

$$R(x) = 0 \tag{2.1}$$

where $R(x)$ is a mapping $R : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $x \in \mathbb{R}^n$. Equation (2.1) shall be considered to be a high fidelity model which might arise from the discretization of a set of partial differential equations. To solve this nonlinear system, Newton's method for nonlinear systems is applied and is shown below:

$$J_k \Delta x_k = -R_k \tag{2.2}$$

$$x_{k+1} = x_k + \alpha_k \Delta x_k \tag{2.3}$$

where $J_k \equiv J(x_k)$, $J_k \in \mathbb{R}^{n \times n}$ is the Jacobian, $\Delta x_k \in \mathbb{R}^n$ is the Newton step, $R_k \equiv R(x_k)$, $R_k \in \mathbb{R}^n$ is the residual, and $\alpha_k \in (0, 1]$ yields a sufficient decrease for the merit function $M_f(x) = \frac{1}{2} R(x)^T R(x)$ at iteration k.

One of the most important reasons why Newton's method is a very popular choice for solving nonlinear systems is due to the fast rate of convergence of the algorithm (Q-quadratic convergence). There are however, some disadvantages to using Newton's method. First, Newton's method is a local method which depends on the initial value. This can be overcome by using trust region or line search methods which globalize the algorithm such that the method does not depend on the initial point. In the formulation above, the parameter α_k has been included which represents the globalization of the method and addresses this disadvantage.

Second, Newton's method requires that a sequence of linear systems be solved in order to

obtain a solution as shown in Eq. (2.2). In this work, Eq. (2.2) shall be considered to be a large scale system which is too computationally expensive to solve. Recall that this is where the motivation for using model reduction comes from. In this dissertation, this disadvantage is addressed and methods for model reduction using wavelets are proposed to obtain reduced-order models which can be solved within an acceptable amount of time.

2.1 Reduced-Order Models

To reduce the computational cost of solving the high dimensional system, a reduced-order model is needed in order to perform simulations within an acceptable amount of time and to cope with limited memory and storage capacity among other requirements. We thus seek an approximate solution of the form

$$x^* = \Phi\hat{x} \tag{2.4}$$

where $\Phi = [\Phi_1\Phi_2 \cdots \Phi_r]$, $\Phi \in \mathbb{R}^{n\times r}$ is a basis, $\hat{x} \in \mathbb{R}^r$ is a vector of coordinates of the approximate solution $x^* \in \mathbb{R}^n$ in the reduced basis Φ, and $r < n$. Substituting Eq. (2.4) into Eq. (2.1) yields an overdetermined system of n equations in r unknowns.

$$R(\Phi\hat{x}) = 0 \quad . \tag{2.5}$$

To solve this system, we identify a basis Ω which results in a reduced system of r equations with r unknowns.

$$\Omega^T R(\Phi\hat{x}) = 0 \tag{2.6}$$

To solve the nonlinear reduced-order model in Eq. (2.6), Newton's method is used which results in solving a sequence of linear equations, known as the reduced Newton's equations [8] shown below.

$$\Omega^T J_k \Phi \Delta \hat{x}_k = -\Omega^T R_k \tag{2.7}$$

If the basis Ω is the same as the basis Φ such that $\Omega = \Phi$, the reduced-order model in Eq. (2.7) is obtained via a Galerkin projection; otherwise, if $\Omega \neq \Phi$, then the reduced-order model is obtained with a Petrov-Galerkin projection. Equation (2.7) can be compactly represented as

$$\hat{J} x_r = \hat{R} \tag{2.8}$$

where $\hat{J} = \Omega^T J_k \Phi$, $\hat{J} \in \mathbb{R}^{r \times r}$, $x_r = \Delta \hat{x}$, $x_r \in \mathbb{R}^r$, and $\hat{R} = -\Omega^T R_k$, $\hat{R} \in \mathbb{R}^r$ with $r < n$. In this work, the objective is to derive the basis' Ω and Φ which yield the reduced-order model, and the reduced parameter space $\Delta \hat{x}_k$ using orthogonal and bi-orthogonal wavelets.

2.2 Proper Orthogonal Decomposition

As mentioned previously, the current state-of-the-art technique for obtaining reduced-order models is the POD. Here this technique is briefly described [14, 8, 2, 9].

The idea behind the POD is as follows: The high dimensional FOM in Eq. (2.1) is solved for m measurements (commonly referred to as samples or snapshots) taken of the n state variables x (also referred to as the model parameters). These solutions are then arranged into an $n \times m$ matrix A which is then factored using the singular value decomposition

$$A = U \Sigma V^T \tag{2.9}$$

where $U^T U = I$, $U \in \mathbb{R}^{n \times n}$, $V^T V = I$, $V \in \mathbb{R}^{m \times m}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{n \times m}$ is a matrix composed of the singular values σ_i along the diagonal arranged in descending order with all other elements equal to zero. The n columns of U and the m columns of V are called the

left-singular vectors and right-singular vectors of A respectively.

To obtain a reduced-basis, U can also be expressed by $U = [U_1 U_2 \cdots U_n]$ where $U_i \in \mathbb{R}^n$. The reduced basis is formed by choosing $r < n$ columns of U to obtain $\Phi = [U_1 U_2 \cdots U_r]$ with $\Phi \in \mathbb{R}^{n \times r}$. With Φ, the approximate solution to x can now be obtained from Eq. (2.4).

We now turn our attention to the method for obtaining the left basis Ω. We see that combining $\Delta x_k = -J_k^{-1} R_k$ and $\Delta x_k \approx \Phi \Delta \hat{x}_k$ yields the overdetermined system of equations shown below.

$$\Phi \Delta \hat{x}_k \approx -J_k^{-1} R_k \tag{2.10}$$

To obtain a solution, the overdetermined system is solved as a weighted least-squares optimization problem given by

$$\min_{\Delta \hat{x}_k} \frac{1}{2} \left\| \Phi \Delta \hat{x}_k + J_k^{-1} R_k \right\|_Q^2 \tag{2.11}$$

where Q is a positive definite matrix. The solution to this optimization problem are the well-known weighted normal equations and expressed here as

$$\Phi^T Q \Phi \Delta \hat{x}_k = -\Phi^T Q J_k^{-1} R_k \tag{2.12}$$

From Eq. (2.12), we see that there can be several choices for Q, namely $Q = I$, $Q = J_k$, and $Q = J_k^T J_k$. If $Q = I$ then we have the linear least-squares problem and the solution is given by

$$\Delta \hat{x} = -L J_k^{-1} R_k. \tag{2.13}$$

Notice that by allowing $Q = I$, we are forced to solve for J_k^{-1} from the FOM which is precisely what we want to avoid. We thus conclude that no computational advantage is realized when $Q = I$ and do not consider this case.

When the Jacobian J_k is positive definite, then we obtain an optimal solution to the weighted least-squares problem with $Q = J_k$ which yields the following form for the reduced Newton

equations.

$$\Phi^T J_k \Phi \Delta \hat{x}_k = -\Phi^T R_k \tag{2.14}$$

Notice that Eq. (2.14) is equivalent to the form of Eq. (2.7) with $\Omega = \Phi$ and is known as a Galerkin projection. While in general the Jacobian for nonlinear problems may not necessarily be positive definite, the Galerkin projection defined above is optimal for the particular case when J_k is positive definite.

For those situations where Jacobian is not positive definite, a more general approach is to let $Q = J_k^T J_k$ which is always positive definite from the standard Newton method assumptions. With $Q = J_k^T J_k$, the following Petrov-Galerkin projection is obtained.

$$\left(J_k \Phi\right)^T \left(J_k \Phi\right) \Delta \hat{x}_k = -\left(J_k \Phi\right)^T R_k \tag{2.15}$$

Notice that Eq. (2.15) is equivalent to the form of Eq. (2.7) with $\Omega = J_k \Phi$.

Now that the general nonlinear problem and the method for model reduction using the POD and SVD have been introduced, the proposed method for model reduction using wavelets can now be discussed*.

*From this point forward, all references to the SVD or the POD shall be taken to mean POD model reduction using the SVD.

Chapter 3

Wavelet Reduced-Order Models

In this chapter, methods for model reduction using wavelets are proposed. We begin with a discussion of the wavelet transform and the property of energy compaction to motivate its use for model reduction. Then, the mathematics behind the orthogonal and bi-orthogonal wavelet transforms are introduced and used to form reduced-order models using both Galerkin and Petrov-Galerkin methods. Finally, the performance of wavelet reduced-order models is compared to model reduction based on the SVD.

3.1 Wavelets, Data Compression and Reduced-Order Models

The motivation for using the wavelet transform for model reduction comes from the fields of image processing, image compression, and transform coding [32]. In these fields, huge amounts of information (for example, images) are transmitted over limited-bandwidth communication lines and networks such as the Internet. For fast transmission and to reduce storage requirements, these signals must be compressed. In short, the goal of data compression is, given a signal $x \in \mathbb{R}^n$, find a lower dimensional signal $\hat{x} \in \mathbb{R}^r$ with $r < n$ to transmit or store. The most widely used transform coding techniques for data compression are based on the wavelet transform.

The key to using wavelets to find the lower dimensional signal $\hat{x}$ lies in a property known as energy compaction. It is well known that the wavelet transform tends to concentrate energy in the low frequency sub-band of the wavelet decomposition [1, 37, 42]. To quantify this property, the energy compaction ratio shall be defined as

$$\mathcal{E} = \frac{||\hat{x}||}{||x||}. \tag{3.1}$$

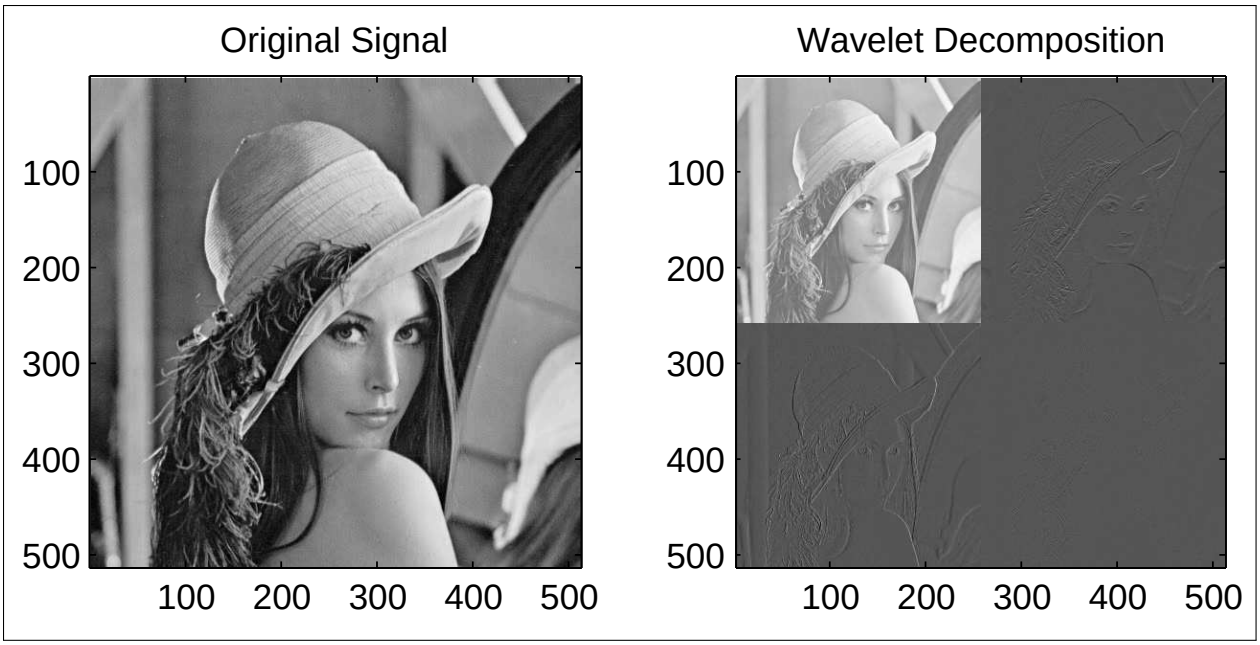


Figure 3.1: Original signal x and the corresponding wavelet decomposition. The upper left sub-band is the low frequency information and represents the low dimensional approximation to x

To illustrate the energy compaction property, consider Figure 3.1. Here, we have the original signal $x \in \mathbb{R}^{512\times512}$ and its corresponding wavelet transform decomposition also in $\mathbb{R}^{512\times512}$. Notice that the upper left quadrant of the wavelet decomposition is a low dimensional approximation $\hat{x} \in \mathbb{R}^{256\times256}$ that is one-fourth of the size of the original signal and corresponds to the low frequency sub-band wavelet coefficients. Using Eq. (3.1), we quantify the amount of energy contained in $\hat{x}$ and find that it contains 99.89% of the energy using only one-fourth of the coefficients.

Since $\hat{x}$ contains most of the energy, a data compression scheme would be to make all of the other wavelet sub-bands zero and preserve only the low frequency information. Using $\hat{x}$, we can reconstruct an approximation of x using the inverse wavelet transform as in Figure 3.2. To the human eye, the difference between the original signal and the approximation may not be readily detected. To measure how well $x^* = L^T\hat{x}$ approximates the original signal, we use the relative error measure defined as

$$e_{rel} = \frac{\|x - x^*\|}{\|x\|} \quad . \tag{3.2}$$

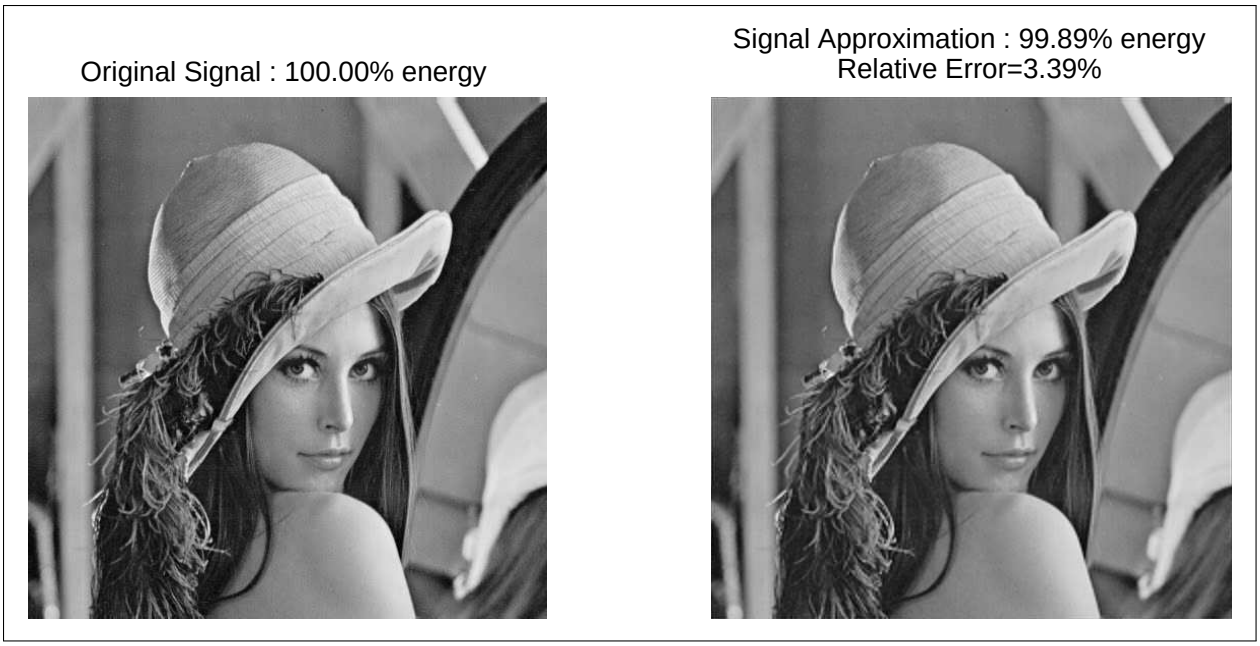


Figure 3.2: A side-by-side comparison of the original signal x and a reconstructed approximation using only the low-low frequency information in $\hat{x}$.

Using this measure, we find that the relative error between the original signal and the approximation is 3.39%. It becomes clear that by exploiting the property of energy compaction, the wavelet transform offers a way to achieve model reduction.

To illustrate the similarity between data compression with wavelets, the idea behind data compression with the SVD shall now be briefly discussed since the SVD is used for POD. Recall the definition for the SVD for a matrix (image) A given in Eq. (2.9). An alternative form is to express the SVD as a sum of rank-one outer products of the left and right singular vectors u_i and v_i, where each outer product is multiplied by the corresponding singular value σ_i.

$$A = u_1\sigma_1 v_1^T + \cdots + u_n\sigma_n v_n^T = \sum_{i=1}^{n} \sigma_i u_i {v_i}^T \tag{3.3}$$

In the context of data compression then, an approximation to the image can be obtained by keeping only $r < n$ singular values and omitting all other terms from this sum which correspond to smaller

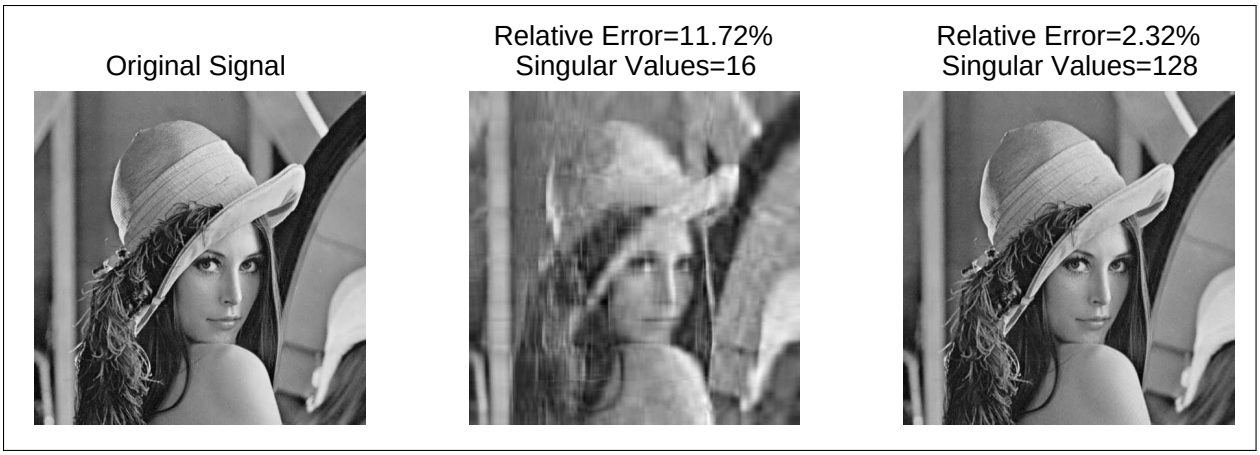


Figure 3.3: A side-by-side comparison of the original signal x and a reconstructed approximation using the singular value decomposition with 16 and 128 singular values.

singular values.

$$A^* = u_1\sigma_1 v_1^T + \cdots + u_r\sigma_r v_r^T = \sum_{i=1}^{r} \sigma_i u_i {v_i}^T \tag{3.4}$$

Data compression and image approximation using the singular value decomposition are shown in Figure 3.3 with r equal to 16 and 128 for $n = 512$. We see that there is a close relationship between the wavelet transform and the singular value decomposition with respect to data compression and model reduction.

3.2 Orthogonal Wavelet Reduced-Order Models

In this section, Galerkin and Petrov-Galerkin reduced-order models are proposed using the orthogonal wavelet transform. We begin with an introduction to the orthogonal wavelet transform and then formulate the orthogonal reduced-order wavelet transform. Finally, orthogonal wavelet reduced-order models are proposed.

3.2.1 The Orthogonal Wavelet Transform

Let x be a vector $x \in \mathbb{R}^n$ and Ψ be a matrix that represents an orthogonal wavelet transform. The wavelet matrix Ψ is composed of low-pass and high-pass sub-matrices as seen in (3.5)

$$\Psi = \begin{bmatrix} L \\ H \end{bmatrix} \tag{3.5}$$

where $\Psi \in \mathbb{R}^{n \times n}$, $L \in \mathbb{R}^{r \times n}$ and $H \in \mathbb{R}^{s \times n}$ [33]. Note that $s = n - r$. The matrix Ψ has $rank(\Psi) = n$ and the sub-matrices in (3.5) have $rank(L) = r$ and $rank(H) = s$ respectively. Since the matrix Ψ is orthogonal, it satisfies the following property:

$$\Psi^T \Psi = \Psi \Psi^T = I \quad . \tag{3.6}$$

By taking (3.6) and substituting the low-pass and high-pass sub-matrices from (3.5), the condition for orthogonality can also be represented as

$$\Psi^T \Psi = \begin{bmatrix} L^T & H^T \end{bmatrix} \begin{bmatrix} L \\ H \end{bmatrix} \tag{3.7}$$

$$I_{n \times n} = L^T L + H^T H \tag{3.8}$$

and as

$$\Psi\Psi^T = \begin{bmatrix} LL^T & LH^T \\ HL^T & HH^T \end{bmatrix} \tag{3.9}$$

$$= \begin{bmatrix} I_{r\times r} & 0_{r\times s} \\ 0_{s\times r} & I_{s\times s} \end{bmatrix} \tag{3.10}$$

where $I_{n\times n}$ denotes $I \in \mathbb{R}^{n\times n}$. We thus obtain the following relationships:

$$LL^T = I_{r\times r} \tag{3.11}$$

$$HH^T = I_{s\times s} \quad . \tag{3.12}$$

Let the forward wavelet transform of x be represented in matrix form by

$$\bar{x} = \Psi x \tag{3.13}$$

where

$$\bar{x} = \begin{bmatrix} \hat{x} \\ \tilde{x} \end{bmatrix} \tag{3.14}$$

with $\hat{x} \in \mathbb{R}^r$, $\tilde{x} \in \mathbb{R}^s$, and $\bar{x} \in \mathbb{R}^n$. Similarly, let the inverse wavelet transform be denoted by

$$x = \Psi^T \bar{x} \quad . \tag{3.15}$$

When equations (3.13), (3.5) and (3.14) are combined, it becomes evident that $\hat{x}$ and $\tilde{x}$ represent the results of filtering x with the low-pass sub-matrix L and the high-pass sub-matrix H respectively.

$$\begin{bmatrix} \hat{x} \\ \tilde{x} \end{bmatrix} = \begin{bmatrix} Lx \\ Hx \end{bmatrix} \tag{3.16}$$

We can also derive an expression for the inverse wavelet transform in terms of $\hat{x}$ and $\tilde{x}$. Beginning with Eq. (3.15) and expanding yields

$$\begin{aligned} x &= \Psi^T \bar{x} \\ &= \begin{bmatrix} L^T & H^T \end{bmatrix} \begin{bmatrix} \hat{x} \\ \tilde{x} \end{bmatrix} \\ &= L^T \hat{x} + H^T \tilde{x} \quad . \end{aligned} \tag{3.17}$$

The expression in Eq. (3.17) indicates that the inverse wavelet transform is given by the sum of the low-pass and high-pass components obtained from Ψ.

3.2.2 Reduced-Order Models Using Orthogonal Wavelets

To meet our objective, we begin by seeking an expression for the reduced model parameters using the orthogonal wavelet transform. The key property of wavelets that we wish to exploit here is that of energy compaction which states that the majority of the energy is concentrated in the low-pass wavelet vector $\hat{x}$ and very little in the high-pass vector $\tilde{x}$. We thus seek an expression for $\hat{x}$. To achieve this, we introduce the compression matrix C defined as

$$C = \begin{bmatrix} I_{r\times r} & 0_{r\times s} \end{bmatrix} \tag{3.18}$$

where $C \in \mathbb{R}^{r\times n}$. By pre-multiplying the forward wavelet transform in Eq. (3.13) by C, we obtain

$$\begin{aligned} C\bar{x} &= C\Psi x \\ \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \tilde{x} \end{bmatrix} &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} L \\ H \end{bmatrix} x \\ \hat{x} &= Lx \quad . \end{aligned} \tag{3.19}$$

Equation (3.19) shall be called the forward reduced-order wavelet transform of x. This expression provides us with a method to obtain a reduced-order parameter space given the low-pass wavelet sub-matrix L and x. We see that the length of $\hat{x}$ is less than the length of x since $r < n$, and the dimension has been reduced from n to r.

To complement the forward reduced-order wavelet transform, an expression to recover the full-order parameter space x^* from $\hat{x}$ is needed. The energy compaction property shall be used again. Using the expression for the inverse wavelet transform in Eq. (3.17) and assuming the contribution from $H^T\tilde{x}$ is small such that $H^T\tilde{x} \approx 0$ yields an approximation to the full-order parameter space.

$$x^* = L^T\hat{x} \quad . \tag{3.20}$$

Equation (3.20) shall be called the inverse reduced-order wavelet transform. We see that $x^* \in \mathbb{R}^n$ and we have obtained a FOM parameter estimate of the vector x from $\hat{x}$. This leads to the following property.

Property 1 *The inverse reduced-order wavelet transform $x^* = L^T\hat{x}$ is an optimal estimate of the full-order parameter space x.*

Proof. Consider Eq. (3.19) as a system of linear equations with x as the model parameters we wish to estimate. Since $L \in \mathbb{R}^{r\times n}$, this system of equations is underdetermined and has an infinite number of solutions. To find an optimal estimate of x, we can express our problem as the constrained optimization problem shown below:

$$\begin{aligned} \min_{x} \quad & ||x||^2 \\ \text{s.t.} \quad & Lx = \hat{x} \quad . \end{aligned} \tag{3.21}$$

The solution to this system is given by the Moore-Penrose pseudoinverse for underdetermined systems

$$x^* = L^T\hat{x} \tag{3.22}$$

where x^* denotes an optimal estimate of x. In this derivation, we used the wavelet sub-matrix property $LL^T = I_{r\times r}$. ■

To quantify the error introduced when using the reduced-order wavelet transform, the error shall be defined as

$$e = \|x - x^*\|^2 \quad , \tag{3.23}$$

which leads to the next property.

Property 2 *The error introduced between the full-order parameter space and the approximation to the full-order parameter space when using the reduced-order wavelet transform is equal to the energy (ℓ_2-norm) of the high-pass approximation coefficients.*

Proof. Using Eqs. (3.15), (3.5) and (3.22) then simplifying gives

$$\begin{aligned} e &= \left\|\Psi^T\bar{x} - L^T\hat{x}\right\|^2 \\ &= \left\|L^T\hat{x} + H^T\tilde{x} - L^T\hat{x}\right\|^2 \\ &= \left\|H^T\tilde{x}\right\|^2 \\ e &= ||\tilde{x}||^2 \end{aligned} \tag{3.24}$$

where the orthogonal wavelet sub-matrix property $HH^T = I_{s\times s}$ was used. This expression for the error shows that it is equal to the energy in the high-pass components of the wavelet transform which we disregarded in our derivation of the reduced-order wavelet transform. ■

This observation is intuitively satisfying. This expression indicates the greater the energy compaction, the smaller the quantity $||\tilde{x}||^2$. The following property naturally follows from Property 2.

Property 3 *The wavelet that concentrates most of the energy of x in the low-pass vector $\hat{x}$ will yield the smallest error.*

Proof. An expression for the error in terms of the low-pass wavelet sub-matrix is equivalently

derived below.

$$\begin{aligned} ||\tilde{x}||^2 &= \tilde{x}^T\tilde{x} \\ &= \left(x^T H^T\right)(Hx) \\ &= x^T\left(I - L^T L\right)x \\ &= \|x\|^2 - \|Lx\|^2 \\ &= \|x\|^2 - \|\hat{x}\|^2 \end{aligned} \tag{3.25}$$

This expression in Eq. (3.25) indicates that the wavelet that concentrates the majority of its energy in the low-pass vector $\hat{x}$ will yield the smallest error. Note that if energy compaction is perfect, then $\|x\| = \|\hat{x}\|$ and the error will be equal to zero. ■

We are now ready to propose and derive the methods for obtaining reduced-order models using wavelets for solving nonlinear problems. To accomplish this, the following property shall be provided.

Property 4 *The approximate solution of the form $x^* = \Phi\hat{x}$ is obtained using the wavelet transform when the basis Φ is given by the low-pass orthogonal wavelet sub-matrix L^T.*

Proof. By comparing Eq. (3.22) to (2.4), it is clear that by choosing the low-pass sub-matrix L^T taken from an orthogonal wavelet transform as the basis yields an orthogonal wavelet reduced-order model where $\Phi = L^T$. ■

The goal of using reduced-order models is to reduce the computational cost of solving a nonlinear system of equations which is in turn solved using Newton's method. Using $L^T\Delta\hat{x_k}$ as an approximation to the Newton step yields the overdetermined system of equations shown below.

$$L^T\Delta\hat{x}_k \approx -J_k^{-1}R_k \tag{3.26}$$

To obtain a solution, the overdetermined system is posed as a weighted least-squares optimization

problem similar to the problem shown in (2.11) and given by

$$\min_{\Delta\hat{x}_k} \frac{1}{2} \left\| L^T \Delta\hat{x}_k + J_k^{-1} R_k \right\|_Q^2 \quad . \tag{3.27}$$

The solution to this optimization problem is given in Eq. (2.12) with $\Phi = L^T$.

$$LQL^T \Delta\hat{x}_k = -LQJ_k^{-1} R_k \tag{3.28}$$

Using this expression, the left basis Ω shall be derived for Eq. (3.28) based on the choice of Q in the following properties.

Property 5 *The Galerkin wavelet reduced-order model is given by* $LJ_kL^T\Delta\hat{x}_k = -LR_k$.

Proof: Assuming that J_k is symmetric positive definite and substituting $Q = J_k$ into Eq. (3.28) produces the following form of the reduced Newton equations:

$$LJ_kL^T\Delta\hat{x}_k = -LR_k \tag{3.29}$$

Since Eq. (3.29) is equivalent to Eq. (2.7) with $\Omega = L$, then Eq. (3.29) shall be called the Galerkin wavelet reduced-order model. ■

Property 6 *The Petrov-Galerkin wavelet reduced-order model is given by* $(J_kL^T)^T(J_kL^T)\Delta\hat{x}_k = -(J_kL^T)^T R_k$.

Proof: Under the Newton assumptions, J_k is full rank and $J_k^T J_k$ is positive definite. Substituting $Q = J_k^T J_k$ into Eq. (2.7) produces the following form of the reduced Newton equations:

$$(J_kL^T)^T(J_kL^T)\Delta\hat{x}_k = -(J_kL^T)^T R_k \quad . \tag{3.30}$$

Since Eq. (3.30) is equivalent to Eq. (2.7) with $\Omega = J_kL^T$, then Eq. (3.30) shall be called the Petrov-Galerkin wavelet reduced-order model. ■

The derivation is complete and we now have two methods given by Galerkin and Petrov-Galerkin projections to obtain wavelet reduced-order models which reduce the Newton equations from n to r using orthogonal wavelets.

3.3 Orthogonal Wavelet Reduced-Order Model Results

In this section, the performance of the orthogonal wavelet reduced-order models is compared to model reduction based on the SVD. This technique is tested using the nonlinear Burgers' partial differential equation.

3.3.1 Experimental Setup

To evaluate the performance of the wavelet reduced-order model, the technique was applied to the nonlinear Burgers' partial differential equation [39] given below

$$\frac{\partial x}{\partial t} - \frac{\partial^2 x}{\partial l^2} - x\frac{\partial x}{\partial l} = 0 \tag{3.31}$$

for the spatial coordinate $0 \leq \ell \leq 5$ and time $0 \leq t \leq 5$ with initial condition $x(\ell, 0) = i_0(\ell)$, boundary conditions $x(0, t) = b_0(t)$ and $x(\ell_f, t) = b_{\ell_f}(t)$, number of spatial discretization points $n' = 514$ and number of time discretization points $q = 128$. Note that since the boundary conditions are known, the number of unknowns x solved for are $n = 512$. The boundary conditions are given by

$$b_0(t) = \frac{2(0.75)}{1 + e^{(-0.75^2 t)}} \tag{3.32}$$

$$b_{\ell_f}(t) = \frac{2(0.75)}{1 + e^{(-0.75^2 t - 0.75\ell_f)}} \tag{3.33}$$

and the initial condition is given by

$$i_0(\ell) = \frac{2(0.75)}{1 + e^{(-0.75\ell)}} \tag{3.34}$$

respectively.

To evaluate the performance, the technique was applied to the nonlinear Burgers' PDE with pseudocode shown in Algorithm 1. To solve the PDE, it was discretized using the finite difference method to obtain a nonlinear system of equations. Newton's Method was then used to linearize the nonlinear system and thus amounts to solving a sequence of linear systems of equations at each time step [25].

Algorithm 1 outlines the methodology used to solve the Burgers' equation. Algorithm 1 is written to accept inputs in terms of Ω and Φ and is applicable to the Galerkin and Petrov-Galerkin orthogonal and bi-orthogonal wavelet model reduction approaches. In lines 4 and 5, the residual R and Jacobian J are calculated. In line 6, R is reduced from $\mathbb{R}^n$ to $\mathbb{R}^r$ and in line 7, the reduced-order model $\hat{J}$ is obtained. The system of equations is solved in line 8 to obtain the reduced-order parameter space $\Delta\hat{x}$. The full-order parameter space Δx is recovered and increased from $\mathbb{R}^r$ to $\mathbb{R}^n$ in line 9. For the inner loop, this process is repeated until the maximum number of iterations is reached or a suitable stopping criteria is met. For the outer loop, the process repeats until the maximum number of time steps is reached.

In order to compare the performance of the wavelet reduced-order modeling techniques, a reduced-order model using the SVD was also implemented. In addition, the FOM was also executed without model reduction to obtain a set of baseline metrics for comparison. To quantify the performance, the following metrics from Table 3.1 shall be defined:

- e_{rel} : The relative error between the true FOM parameter x and the FOM parameter approximation x^* as recovered from the reduced-order model parameter $\hat{x}$.
- $\mathcal{S}$: The speedup factor measures how much faster a solution is obtained when using model reduction versus the FOM. t is the time taken to solve the PDE using the FOM and $\hat{t}$ is the time taken to solve the problem using model reduction.
- $\mathcal{R}_r$: The model parameter reduction ratio measures the reduction in dimension from x to $\hat{x}$.

Algorithm 1 Wavelet Reduced-Order Model Applied to Nonlinear PDEs

Input: Ω, Φ

for $j = 0$ **to** MaxTimeStep **do**
  Given an initial condition $x_{j,0}$
  for $k = 0$ **to** MaxIterations **or** stopping criteria met **do**
    Obtain the residual $R := R_k(x_{j,k})$.
    Obtain the Jacobian $J := J_k(x_{j,k})$.
    Reduce R to obtain $\hat{R}$.

$$\hat{R} = \Omega^T R$$

    Reduce J to obtain $\hat{J}$

$$\hat{J} = \Omega^T J \Phi$$

    Solve for the reduced-order parameter space $\Delta\hat{x}$.

$$\hat{J}\Delta\hat{x} = -\hat{R}$$

    Recover the full-order parameter space Δx.

$$\Delta x = \Phi \Delta \hat{x}$$

    Update

$$x_{j,k+1} = x_{j,k} + \Delta x$$

  end for
end for
return

Table 3.1: Performance Metrics

Relative Error	$e_{rel} = \frac{\|x - x^*\|}{\|x\|}$
Speedup Factor	$\mathcal{S} = \frac{t}{\hat{t}}$
Model Parameter Reduction Ratio	$\mathcal{R}_r = \frac{n - r}{n}$

Table 3.2: Decomposition Levels

d	$r \times r$	r	$\mathcal{R}_r$ (%)
N/A	512x512	512	0.0000
1	256x256	256	50.0000
2	128x128	128	75.0000
3	64x64	64	87.5000
4	32x32	32	93.7500
5	16x16	16	96.8750
6	8x8	8	98.4375
7	4x4	4	99.2188
8	2x2	2	99.6094

The experiments were executed on a workstation with a 2.13 GHz Intel Core 2 Duo with 4GB of 1067MHz DDR3 memory running OS X.

3.3.2 Numerical Results

Since many orthogonal wavelet transforms exist, the wavelet reduced-order modeling technique was tested using a suite of wavelets from the Daubechies* and Symlet families† at decomposition levels 1 - 8 as seen in Table 3.3. For each wavelet and decomposition level, each wavelet reduced-order modeling technique (Galerkin, Petrov-Galerkin) was applied to the system of equations. Performance metrics were obtained and then compared to the FOM and SVD.

*db1 is also known as the Haar wavelet.

†Wavelet naming conventions used are as defined in MATLAB. Wavelet coefficients were also taken from [11]

Table 3.3: Orthogonal Wavelet Options

Daubechies	db1, db2, db3, db4
Symlet	sym1, sym2, sym3, sym4, sym5
Decomposition Levels	1 - 8
Wavelet Model Reduction	Galerkin, Petrov-Galerkin

Table 3.4 contains the results obtained from the experiment where each method was compared at decomposition level 2. The columns of Table 3.4 are defined as follows:

- ROM - Refers to the technique used for model reduction.
- e_{rel} - The relative error.
- $\mathcal{S}_{off}$ - The speedup obtained during the offline computation phase. This is computed with respect to the SVD offline time.
- Wavelet Reduction - Indicates whether the Galerkin or Petrov-Galerkin method was used.
- Ψ - The wavelet used for the Galerkin and Petrov-Galerkin methods.
- $r \times r$ - The dimension of the reduced-order model $\hat{J} \in \mathbb{R}^{r \times r}$.
- r - The dimension of the reduced-order parameter space $\Delta\hat{x} \in \mathbb{R}^{r}$.
- $\mathcal{R}_r$ (%) - The model parameter reduction ratio.

Table 3.4: Orthogonal Wavelet Model Reduction

1D Burgers' Equation (n = 512, q = 32, d = 2, Orthogonal)							
ROM	e_{rel}	$\mathcal{S}_{off}$	Wavelet Reduction	Ψ	$r \times r$	r	$\mathcal{R}_r$ (%)
None	0.0000	0.0000	N/A	N/A	512x512	512	0.0000
SVD	0.0741	1.0000	N/A	N/A	128x128	128	75.0000
Wavelet	0.0322	21.4724	Galerkin	sym1	128x128	128	75.0000
Wavelet	0.0946	21.4258	Petrov-Galerkin	sym1	128x128	128	75.0000

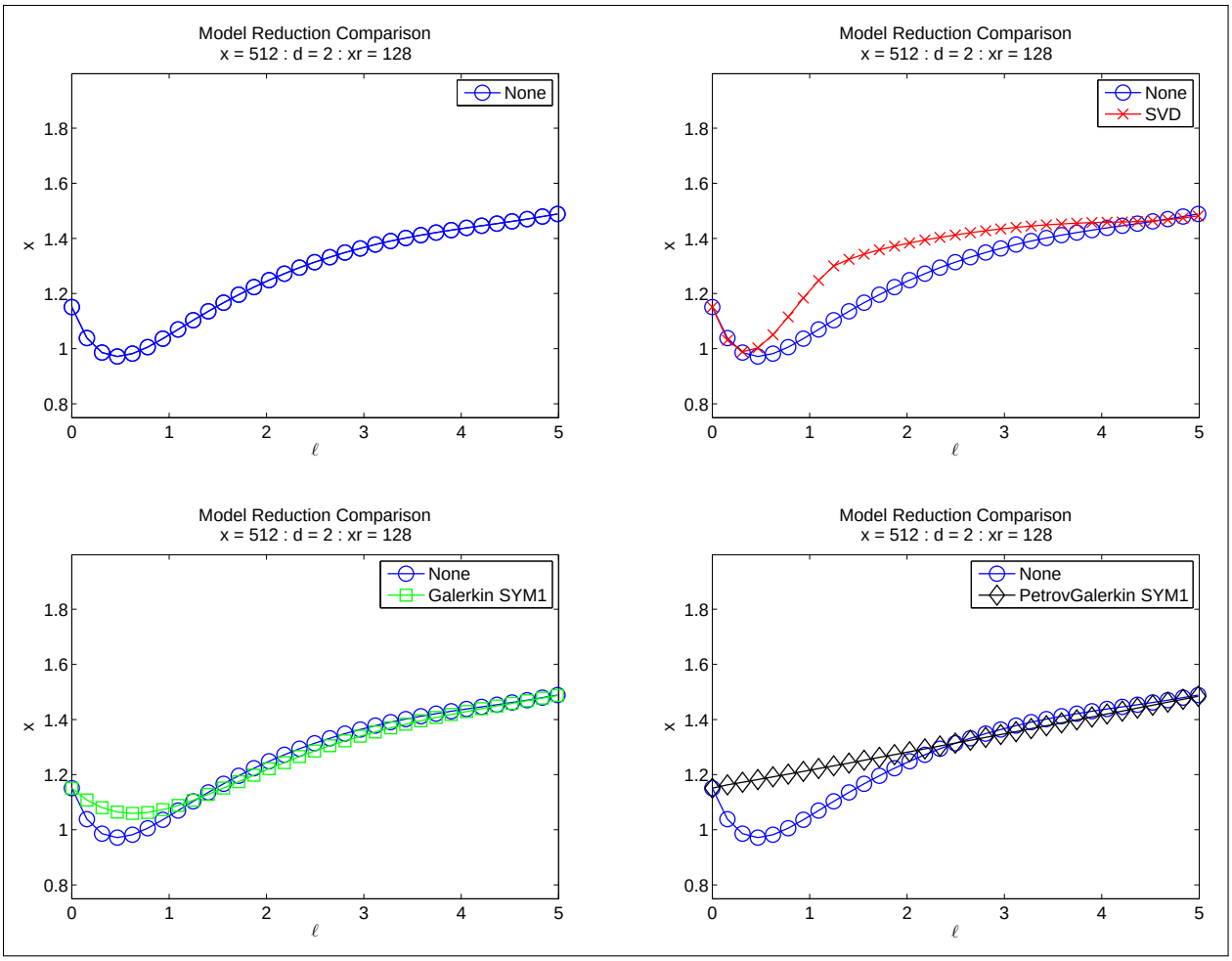


Figure 3.4: Nonlinear Orthogonal Model Reduction Comparison

From Table 3.4 we see that at decomposition level 2, J_k has been reduced from $\mathbb{R}^{512\times512}$ to $\mathbb{R}^{128\times128}$ and x has been reduced from $\mathbb{R}^{512}$ to $\mathbb{R}^{128}$ (a reduction of 75%). Of the two wavelet model reduction techniques, the Galerkin method has the smallest relative error compared to the SVD.

Figure 3.4 shows the solution of the nonlinear Burgers' equation at discrete time point 16 of 32 for the solution obtained using the Galerkin and Petrov-Galerkin orthogonal wavelet reduced-order model, the SVD reduced-order model, and the FOM at decomposition level 2. Visually, we see that the Galerkin orthogonal wavelet model reduction more closely approximates the solution obtained using the FOM when compared to the SVD.

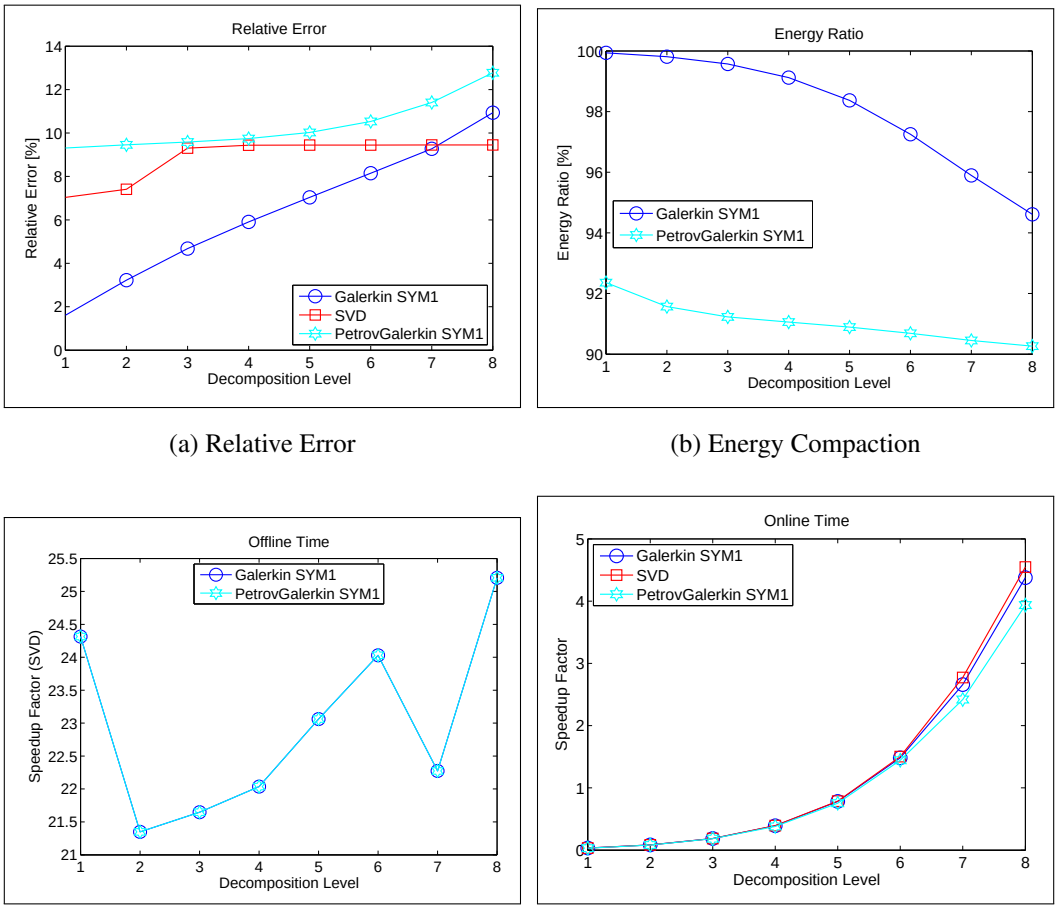
Figure 3.5 contains graphs of four performance metrics for the Galerkin and Petrov-Galerkin wavelet reduced-order models and the SVD reduced-order model. All graphs are plotted against decomposition level and Table 3.2 shows the relationship between decomposition level d, the size of $\hat{J} \in \mathbb{R}^{r\times r}$, the size of $\Delta\hat{x} \in \mathbb{R}^r$, and model parameter reduction ratio $\mathcal{R}_r$. Figure 3.5a shows that the relative error obtained using wavelet reduction is less than that obtained using SVD reduction up to decomposition level 7. It is not until decomposition level 7 that the SVD reduced-order model begins to have a better relative error value. Although the Galerkin wavelet reduced order model has a better relative error than that obtained with the SVD, we remind the reader of the disadvantages of the SVD listed in Chapter 1. That is, that that the quality of the reduced-order model depends heavily on how "good" the snapshot matrix is. It is possible that the snapshot matrix used here is not representative of the FOM. Additionally, recall that a key question is how to sample and choose the parameters and inputs over which to compute the reduced basis. Here, the samples were taken at uniform intervals with no special algorithm used. This may have contributed to the poor performance of the SVD.

Figure 3.5b shows a plot of the energy compaction ratio for the wavelet methods. This plot shows that as we increase the level of decomposition, the amount of energy contained in the reduced parameter space decreases. We also notice that as the amount of energy decreases, the error increases.

Figure 3.5d shows the speedup factor obtained with respect to the FOM during the online[‡] computation phase. Figure 3.5d indicates that in this particular case all model reduction techniques begin to outperform the FOM at decomposition levels greater than 5. In addition, there is essentially no performance gain in terms of online speedup when the wavelet model reduction is compared to the SVD since the approaches have almost equivalent speedup factors. Intuitively this makes sense since $\hat{J} = \Omega^T J_k \Phi$ and $\hat{R} = \Omega^T R_k$ involve matrix-matrix and matrix-vector calculations regardless of the method of model reduction, be it by wavelet model reduction or SVD model reduction. This inefficiency will be addressed and a computationally efficient method will

[‡]Model reduction typically occurs in two phases. The first phase is termed *offline computation* where the matrices Φ and Ω are formed. The second phase is termed *online computation* where real-time calculations are performed using the reduced-order models obtained from Φ and Ω.

be proposed in Chapter 4.



(a) Relative Error

(b) Energy Compaction

(c) Offline Computation Time

(d) Online Computation Time

Figure 3.5: Nonlinear Orthogonal Performance Metrics

Figure 3.5d shows the speedup factor obtained with respect to the FOM during the online[§] computation phase. Figure 3.5d indicates that in this particular case all model reduction techniques begin to outperform the FOM at decomposition levels greater than 5. In addition, there is essentially no performance gain in terms of online speedup when the wavelet model reduction is compared to the SVD since the approaches have almost equivalent speedup factors. Intuitively this makes sense since $\hat{J} = \Omega^T J_k \Phi$ and $\hat{R} = \Omega^T R_k$ involve matrix-matrix and matrix-vector cal-

[§]Model reduction typically occurs in two phases. The first phase is termed *offline computation* where the matrices Φ and Ω are formed. The second phase is termed *online computation* where real-time calculations are performed using the reduced-order models obtained from Φ and Ω.

culations regardless of the method of model reduction, be it by wavelet model reduction or SVD model reduction. This inefficiency will be addressed and a computationally efficient method will be proposed in Chapter 4.

Figure 3.5c plots the speedup factor with respect to SVD model reduction versus decomposition level for the offline computations required to obtain Φ and Ω. Since the offline time for the FOM is zero (no offline time required), we compute the speedup with respect to the SVD. It is in this performance metric that the advantage of wavelet reduced-order models over the SVD is seen. We see that by using wavelet reduced order models that the offline computation time completes at least 21x faster and at most 25x faster than the SVD. By using wavelet reduced-order models, simulations can go from offline computations to real-time simulations (online computations) much faster than by using SVD reduced-order models because the FOM does not need to be calculated and no snapshots and snapshot algorithms are required. The only computation required in the offline phase is to generate the low-pass wavelet sub-matrix.

Table 3.5 compares the orthogonal wavelet reduced-order models with the SVD and summarizes the results of the experiment. The "$\uparrow$" and "$\downarrow$" represent positive and negative aspects respectively while the "$-$" represents a neutral result.

In this particular problem, the Galerkin wavelet reduced-order model provided a good balance between each of the performance metrics. Up to decomposition level 7, the Galerkin approach outperformed the SVD in terms of relative error and was 22x faster than the SVD during the offline computation phase ($\mathcal{S}_{off}$). Using wavelets for model reduction is also simpler compared to the SVD because no offline simulations are needed, and no snapshots are required to obtain Φ and Ω.

We reiterate that in terms of the online computation phase, each of the model reduction approaches perform equally well with each outperforming the FOM and resulting in a 3x increase in real-time simulation performance at decomposition level 7. In Section 4, we propose a computationally efficient wavelet model reduction method to increase the online speedup factor.

Table 3.5: Model Reduction Summary

Model Reduction	$\mathcal{S}_{on}$	$\mathcal{S}_{off}$	Design Simplicity
Galerkin	-	$\uparrow$	$\uparrow$
Petrov-Galerkin	-	$\uparrow$	$\uparrow$
SVD	-	$\downarrow$	$\downarrow$

3.4 Bi-orthogonal Wavelet Reduced-Order Models

In this section, Galerkin and Petrov-Galerkin reduced-order models are proposed using the bi-orthogonal wavelet transform. We begin by introducing the bi-orthogonal wavelet transform and then formulate the bi-orthogonal reduced-order wavelet transform. Finally, bi-orthogonal wavelet reduced-order models are proposed.

3.4.1 The Bi-orthogonal Wavelet Transform

The family of wavelet transforms can be divided into two main categories: orthogonal and bi-orthogonal. In order to provide a complete treatment of wavelet model reduction, model reduction using bi-orthogonal wavelets is also covered in this research. The main differences between the orthogonal and bi-orthogonal wavelets are that orthogonal wavelets do not possess symmetry but preserve energy. Bi-orthogonal wavelets sacrifice energy preservation but have the advantage of symmetry. Table 3.6 provides a comparison of the advantages and disadvantages of orthogonal and bi-orthogonal wavelets. Because model reduction with wavelets has a strong foundation in data compression and image processing, Table 3.6 provides a comparison primarily from this perspective [12].

The orthogonal wavelet transform can be generalized to bi-orthogonal form in which two wavelet transform matrices are used instead of only one. Let x be a vector $x \in \mathbb{R}^n$ and Ψ_1 and Ψ_2 be matrices that represent a bi-orthogonal wavelet transform pair. The wavelet matrices are

Table 3.6: Orthogonal and Bi-orthogonal Wavelet Comparison

	Advantages	Disadvantages
Orthogonal	• Preserves energy between the time and wavelet domains:$\|\|x\|\| = \|\|\bar{x}\|\|$. • Only one wavelet matrix Ψ is required.	• The filter coefficients are not symmetric (except for the trivial Haar wavelet). • In terms of images, the human eye is not as tolerant of imperfections which are not symmetric. • Introduces artifacts at the boundaries of signals and the borders of images.
Bi-orthogonal	• The filter coefficients are symmetric. • In terms of image procesing, symmetry is valuable in human vision perception because the eye is more tolerant of symmetric imperfections. • Generally favored over orthogonal wavelets for image processing because symmetry and symmetric extension at the boundaries help to avoid edge artifacts and lead to better image processing.	• Does not preserve energy between the time and wavelet domains: $\|\|x\|\| \neq \|\|\bar{x}\|\|$. • Two wavelet matrices Ψ_1 and Ψ_2 are required.

composed of low-pass and high-pass sub-matrices as seen in (3.35) and (3.36)

$$\Psi_1 = \begin{bmatrix} L_1 \\ H_1 \end{bmatrix} \tag{3.35}$$

$$\Psi_2 = \begin{bmatrix} L_2 \\ H_2 \end{bmatrix} \tag{3.36}$$

where $\Psi_1 \in \mathbb{R}^{n\times n}$, $L_1 \in \mathbb{R}^{r\times n}$, $H_1 \in \mathbb{R}^{s\times n}$ and $\Psi_2 \in \mathbb{R}^{n\times n}$, $L_2 \in \mathbb{R}^{r\times n}$, $H_2 \in \mathbb{R}^{s\times n}$. In addition, $rank(\Psi_1) = n$, $rank(\Psi_2) = n$ and the corresponding sub-matrices have $rank(L_1) = r$, $rank(H_1) = s$, $rank(L_2) = r$, $rank(H_2) = s$ respectively.

The bi-orthogonal matrix pair satisfies the properties seen in Eqs. (3.37) and (3.38).

$$\Psi_1^T \Psi_2 = \Psi_1 \Psi_2^T = I \tag{3.37}$$

$$\Psi_2^T \Psi_1 = \Psi_2 \Psi_1^T = I \tag{3.38}$$

By taking Eq. (3.37) and substituting the low-pass and high-pass sub-matrices from Eqs. (3.35) and (3.36), the condition for bi-orthogonality can also be represented as [29]

$$\Psi_1^T \Psi_2 = \begin{bmatrix} L_1^T & H_1^T \end{bmatrix} \begin{bmatrix} L_2 \\ H_2 \end{bmatrix} \tag{3.39}$$

$$I_{n\times n} = L_1^T L_2 + H_1^T H_2 \tag{3.40}$$

and as

$$\Psi_1 \Psi_2^T = \begin{bmatrix} L_1 L_2^T & L_1 H_2^T \\ H_1 L_2^T & H_1 H_2^T \end{bmatrix} \tag{3.41}$$

$$I_{n\times n} = \begin{bmatrix} I_{r\times r} & 0_{r\times s} \\ 0_{s\times r} & I_{s\times s} \end{bmatrix} \quad . \tag{3.42}$$

Thus, $L_1 L_2^T = I_{r\times r}$ and $H_1 H_2^T = I_{s\times s}$.

Let the forward bi-orthogonal wavelet transform of x be represented in matrix form by

$$\bar{x} = \Psi_1 x \tag{3.43}$$

where $\bar{x}$ is defined as in Eq. (3.14). Similarly, let the inverse wavelet transform be denoted by

$$x = \Psi_2^T \bar{x} \quad . \tag{3.44}$$

When equations (3.43), (3.35), and (3.14) are combined, we see that $\hat{x}$ and $\tilde{x}$ represent the results of filtering x with the low-pass sub-matrix L_1 and the high-pass sub-matrix H_1 respectively.

$$\begin{bmatrix} \hat{x} \\ \tilde{x} \end{bmatrix} = \begin{bmatrix} L_1 x \\ H_1 x \end{bmatrix} \tag{3.45}$$

We can also derive an expression for the inverse bi-orthogonal wavelet transform in terms of $\hat{x}$ and $\tilde{x}$. Beginning with Eq. (3.44) and expanding yields

$$\begin{aligned} x &= \Psi_2^T \bar{x} \\ &= \begin{bmatrix} L_2^T & H_2^T \end{bmatrix} \begin{bmatrix} \hat{x} \\ \tilde{x} \end{bmatrix} \\ &= L_2^T \hat{x} + H_2^T \tilde{x} \quad . \end{aligned} \tag{3.46}$$

The expression in Eq. (3.46) indicates that the inverse bi-orthogonal wavelet transform is given by the sum of the low-pass and high-pass components obtained from Ψ_1 multiplied by the low-pass and high-pass sub-matrices from Ψ_2.

3.4.2 Reduced-Order Models Using Bi-orthogonal Wavelets

We begin by seeking an expression for the reduced model parameters using the bi-orthogonal wavelet transform. Again, the key property of wavelets that we wish to exploit is that of energy compaction. To achieve this, the compression matrix C from Eq. (3.18) shall be used. Pre-multiplying the forward wavelet transform in Eq. (3.43) by C, we obtain

$$\begin{aligned} C\bar{x} &= C\Psi_1 x \\ \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \tilde{x} \end{bmatrix} &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} L_1 x \\ H_1 x \end{bmatrix} \\ \hat{x} &= L_1 x \end{aligned} \tag{3.47}$$

Eq. (3.47) shall be called the forward reduced-order bi-orthogonal wavelet transform of x. This expression provides us with a method to obtain a reduced-order parameter space given the low-pass wavelet sub-matrix L_1 and x. We see that the length of $\hat{x}$ is less than the length of x since $r < n$, and the dimension has been reduced from n to r just as in the orthogonal case.

To complement the forward reduced-order bi-orthogonal wavelet transform, an expression to recover the full-order parameter space x^* from $\hat{x}$ is needed. The energy compaction property shall be used again. Using the expression for the inverse bi-orthogonal wavelet transform in Eq. (3.46) and assuming the contribution from $H_2^T\tilde{x}$ is small such that $H_2^T\tilde{x} \approx 0$ yields an approximation to the full-order parameter space.

$$x^* = L_2^T\hat{x} \tag{3.48}$$

Equation (3.48) shall be called the inverse reduced-order bi-orthogonal wavelet transform. We see that $x^* \in \mathbb{R}^n$ and we have obtained a FOM parameter estimate of the vector x from $\hat{x}$. An expression for the error introduced when using the reduced-order bi-orthogonal wavelet transform pair shall be defined according to Eq. (3.23) which leads to the following property:

Property 7 *The error introduced between the full-order parameter space and the approximation to the full-order parameter space when using the reduced-order bi-orthogonal wavelet transform is equal to the energy (ℓ_2-norm) of $H_2^T\tilde{x}$.*

Proof: Using Eqs. (3.36), (3.44) and (3.48) then simplifying gives

$$\begin{aligned} e &= \|x - x^*\|^2 \\ &= \left\|\Psi_2^T\bar{x} - L_2^T\hat{x}\right\|^2 \\ &= \left\|L_2^T\hat{x} + H_2^T\tilde{x} - L_2^T\hat{x}\right\|^2 \\ &= \left\|H_2^T\tilde{x}\right\|^2 \quad . \end{aligned} \tag{3.49}$$

Equation (3.49) shows that the error is equal to the energy in the high-pass components of the inverse bi-orthogonal wavelet transform which was disregarded in our derivation of the inverse

reduced-order bi-orthogonal wavelet transform. This expression indicates that the greater the energy compaction, the smaller the quantity $\left\|H_2^T \tilde{x}\right\|^2$ and thus the error becomes. ■

We are now ready to propose and derive the methods for obtaining reduced-order models using bi-orthogonal wavelets. To accomplish this, the following property shall be provided.

Property 8 *The approximate solution of the form $x^* = \Phi\hat{x}$ is obtained using the wavelet transform when the basis Φ is given by the low-pass bi-orthogonal wavelet sub-matrix L_2^T.*

Proof. By comparing (3.48) to Eq. (2.4) it is clear that by choosing the low-pass sub-matrix L_2^T taken from a bi-orthogonal wavelet transform as the basis which approximates the desired solution yields a bi-orthogonal wavelet reduced-order model where $\Phi = L_2^T$. ■

The goal of using reduced-order models is to reduce the computational cost of solving a nonlinear system of equations which is in turn solved using Newton's method. Using $L_2^T \Delta\hat{x_k}$ as an approximation to the Newton step yields the overdetermined system of equations shown below.

$$L_2^T \Delta\hat{x}_k \approx -J_k^{-1} R_k \tag{3.50}$$

To obtain a solution, the overdetermined system is solved as a weighted least-squares optimization problem and given by

$$\min_{\Delta\hat{x}_k} \frac{1}{2} \left\| L_2^T \Delta\hat{x}_k + J_k^{-1} R_k \right\|_Q^2 \quad . \tag{3.51}$$

The solution to this optimization problem is given in Eq. (2.12) with $\Phi = L_2^T$.

$$L_2 Q L_2^T \Delta\hat{x}_k = -L_2 Q J_k^{-1} R_k \tag{3.52}$$

Using this expression, the left basis Ω shall be derived for Eq. (3.52) based on the choice of Q in the following properties.

Property 9 *The Galerkin bi-orthogonal wavelet reduced-order model is given by $L_2 J_k L_2^T \Delta\hat{x}_k = -L_2 R_k$.*

Proof: Assuming that J_k is symmetric positive definite and substituting $Q = J_k$ into Eq. (3.52) produces the following form of the reduced Newton equations:

$$L_2 J_k L_2^T \Delta\hat{x}_k = -L_2 R_k \quad . \tag{3.53}$$

Since Eq. (3.53) is equivalent to Eq. (2.7) with $\Omega = L_2$, then Eq. (3.53) shall be called the Galerkin bi-orthogonal wavelet reduced-order model. ■

Property 10 *The Petrov-Galerkin bi-orthogonal wavelet reduced-order model is given by* $(J_k L_2^T)^T (J_k L_2^T) \Delta\hat{x}_k = -(J_k L_2^T)^T R_k$.

Proof: Assuming that J_k is full rank and substituting $Q = J_k^T J_k$ into Eq. (2.7) produces the following form of the reduced Newton equations:

$$(J_k L_2^T)^T (J_k L_2^T) \Delta\hat{x}_k = -(J_k L_2^T)^T R_k \quad . \tag{3.54}$$

Since Eq. (3.54) is equivalent to Eq. (2.7) with $\Omega = J_k L_2^T$, then Eq. (3.54) shall be called the Petrov-Galerkin bi-orthogonal wavelet reduced-order model. ■

The derivation is complete and we now have two methods given by Galerkin and Petrov-Galerkin projections to obtain wavelet reduced-order models which reduce the Newton equations from n to r using bi-orthogonal wavelets.

3.5 Bi-orthogonal Wavelet Reduced-Order Model Results

In this section, the performance of the bi-orthogonal wavelet reduced-order models is compared to model reduction based on the SVD. As in the orthogonal case, this technique is tested using the nonlinear Burgers' partial differential equation.

3.5.1 Numerical Results

The bi-orthogonal wavelet reduced-order modeling technique was tested using a suite of wavelets from the bior1.x, bior2.x, bior3.x, and bior4.x[¶] families at decomposition levels 1 - 8 as seen in Table 3.7. For each wavelet and decomposition level, each wavelet reduced-order modeling technique (Galerkin, Petrov-Galerkin) was applied to the system of equations. Performance metrics were obtained and then compared to the FOM and SVD.

Table 3.7: Bi-orthogonal Wavelet Options

bior1.x	bior1.1, bior1.3, bior1.5
bior2.x	bior2.2, bior2.4, bior2.6, bior2.8
bior3.x	bior3.1, bior3.3, bior3.5, bior3.7, bior3.9
bior4.x	bior4.4
Decomposition Levels	1 - 8
Wavelet Model Reduction	Galerkin, Petrov-Galerkin

Table 3.8 contains the results obtained from the experiment where each method was compared at decomposition level 2. We see that at decomposition level 2, J_k has been reduced from $\mathbb{R}^{512\times512}$ to $\mathbb{R}^{128\times128}$ and x has been reduced from $\mathbb{R}^{512}$ to $\mathbb{R}^{128}$ (a reduction of 75%). Of the two wavelet model reduction techniques, the Galerkin method has the smallest relative error compared to the SVD.

Figure 3.6 shows the solution of the nonlinear Burger's equation at discrete time point 16 of 32 for the solution obtained using the Galerkin and Petrov-Galerkin orthogonal wavelet reduced-order model, the SVD reduced-order model, and the FOM at decomposition level 2. Visually, we see that the Galerkin bi-orthogonal wavelet model reduction more closely approximates the solution obtained using the FOM when compared to the SVD.

Figure 3.7 contains graphs of four performance metrics for the Galerkin and Petrov-Galerkin wavelet reduced-order models and the SVD reduced-order model. All graphs are plotted against

[¶]Wavelet naming conventions used are as defined in MATLAB.

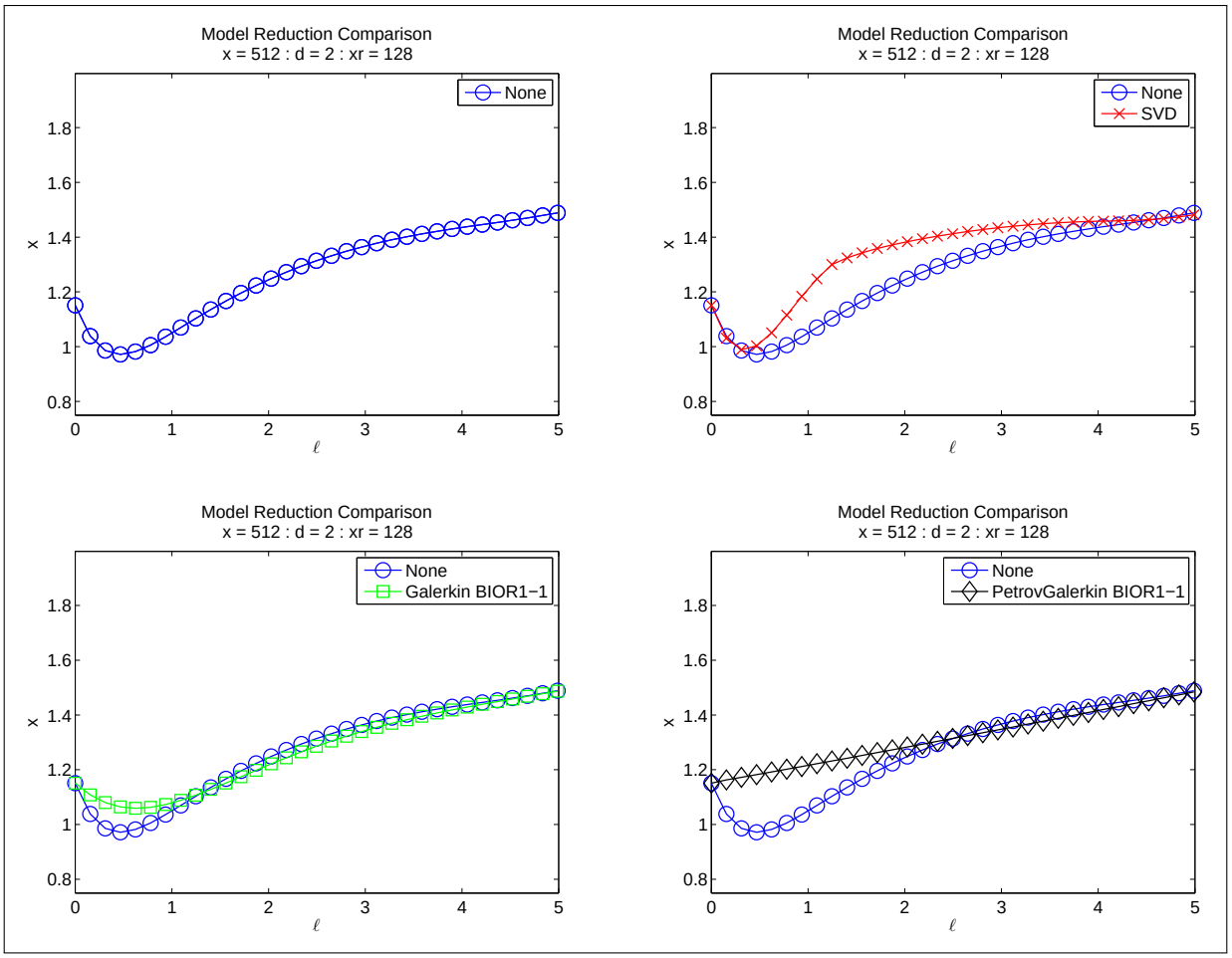


Figure 3.6: Nonlinear Bi-orthogonal Model Reduction Comparison

decomposition level and Table 3.2 shows the relationship between decomposition level d, the size of $\hat{J} \in \mathbb{R}^{r \times r}$, the size of $\Delta\hat{x} \in \mathbb{R}^{r}$, and model parameter reduction ratio $\mathcal{R}_r$. Figure 3.7a shows that the relative error obtained using wavelet reduction is less than that obtained using SVD reduction up to decomposition level 7. It is not until decomposition level 7 that the SVD reduced-order model begins to have a better relative error value. We highlight again that although the Galerkin wavelet reduced-order model has a better relative error than that obtained with the SVD, we remind the reader of the disadvantages of the SVD listed in Chapter 1. That is, that the quality of the reduced-order model depends heavily on how ”good” the snapshot matrix is and that no special algorithm was used to select the sample points for the snapshot matrix. Again, this may have contributed to the poor performance of the SVD.

Table 3.8: Bi-orthogonal Wavelet Model Reduction

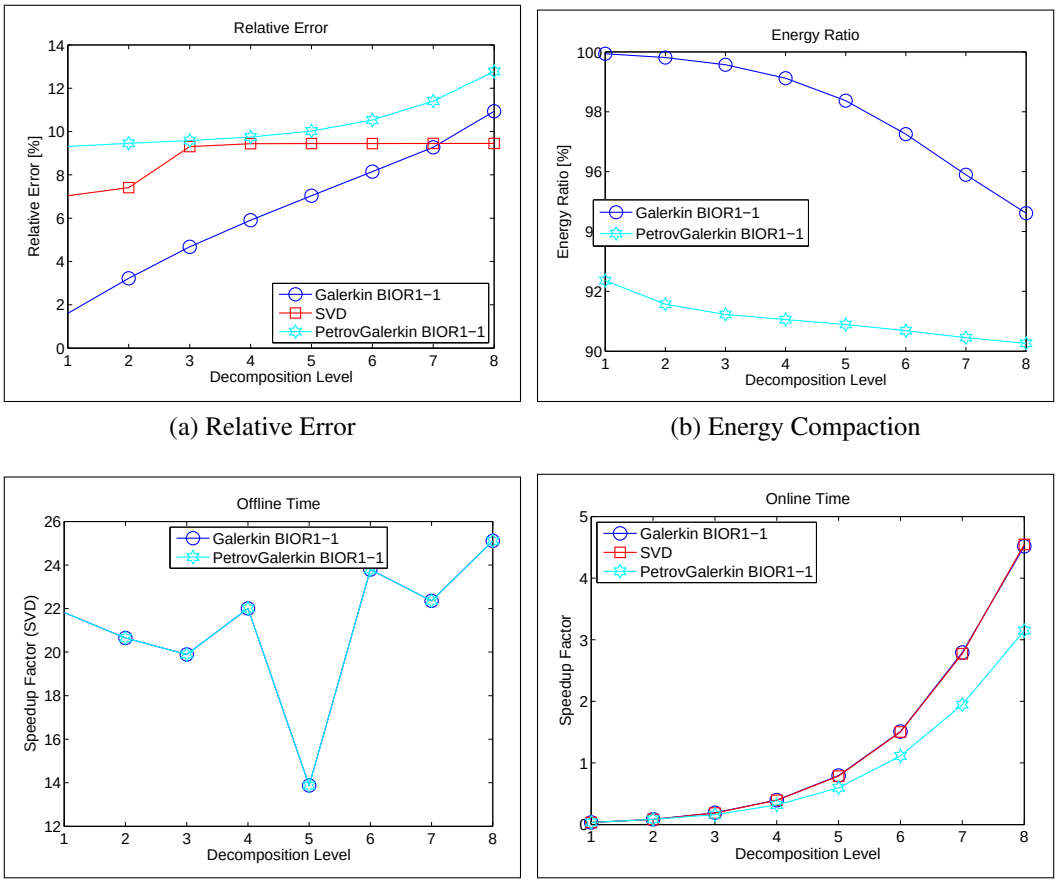
1D Burgers' Equation (n = 512, q = 32, d = 2, Bi-orthogonal)							
ROM	e_{rel}	$\mathcal{S}_{off}$	Wavelet Reduction	Ψ_1	$r \times r$	r	$\mathcal{R}_r$ (%)
None	0.0000	0.0000	N/A	N/A	512x512	512	0.0000
SVD	0.0741	1.0000	N/A	N/A	128x128	128	75.0000
Wavelet	0.0318	21.1464	Galerkin	bior1.1	128x128	128	75.0000
Wavelet	0.0979	21.2122	Petrov-Galerkin	bior1.1	128x128	128	75.0000

Figure 3.7b shows a plot of the energy compaction ratio for the wavelet methods. This plot shows that as we increase the level of decomposition, the amount of energy contained in the reduced parameter space decreases. We also notice that as the amount of energy decreases, the error increases.

Figure 3.7d shows the speedup factor obtained with respect to the FOM during the online[‖] computation phase. Figure 3.7d indicates that in this particular case all model reduction techniques begin to outperform the FOM at decomposition levels greater than 5. Again, there is essentially no performance gain in terms of online speedup when the wavelet model reduction is compared to the SVD since the approaches have almost equivalent speedup factors. This inefficiency will be addressed and a computationally efficient method will be proposed in Chapter 4.

Figure 3.7c plots the speedup factor with respect to SVD model reduction versus decomposition level for the offline computations required to obtain Φ and Ω. Since the offline time for the FOM is zero (no offline time required), we compute the speedup with respect to the SVD. As in the orthogonal case, the advantage of wavelet reduced-order models over the SVD is seen. We see that by using wavelet reduced-order models that the offline computation time completes at least 14x faster and at most 25x faster than the SVD. By using wavelet reduced-order models, simulations can go from offline computations to real-time simulations (online computations) much faster than by using SVD reduced-order models because the FOM does not need to be calculated and no snapshots and snapshot algorithms are required. The only computation required in the offline phase is to generate the low-pass wavelet sub-matrix.

[‖]Model reduction typically occurs in two phases. The first phase is termed *offline computation* where the matrices Φ and Ω are formed. The second phase is termed *online computation* where real-time calculations are performed using the reduced-order models obtained from Φ and Ω.



(a) Relative Error

(b) Energy Compaction

(c) Offline Computation Time

(d) Online Computation Time

Figure 3.7: Nonlinear Bi-orthogonal Performance Metrics

Table 3.5 shall again be used, this time to compare the bi-orthogonal wavelet reduced-order models with the SVD and summarizes the results of the experiment. In this particular problem, the Galerkin wavelet reduced-order model provided a good balance between each of the performance metrics. Up to decomposition level 7, the Galerkin approach outperformed the SVD in terms of relative error and was 23x faster than the SVD during the offline computation phase ($\mathcal{S}_{off}$). We again see that using wavelets for model reduction is also simpler compared to the SVD because no offline simulations are needed, and no snapshots are required to obtain Φ and Ω.

We reiterate that in terms of the online computation phase, each of the model reduction approaches perform equally well with each outperforming the FOM and resulting in a 3x increase in real-time simulation performance at decomposition level 7. In Chapter 4, we propose a computa-

tionally efficient wavelet model reduction method to increase the online speedup factor.

Chapter 4

Orthogonal and Bi-Orthogonal Wavelet Reduced-Order Models Using Filter Banks

In this section, the filter bank approach to the wavelet transform is briefly discussed [10]. The method discussed here applies to both the orthogonal and bi-orthogonal wavelet transforms. Using the filter bank approach for the wavelet transform, we obtain a computationally efficient alternative based on convolution as opposed to matrix multiplication.

4.1 Matrix Multiplication and Convolution

In order to understand the filter bank approach for taking the wavelet transform, the convolution operation must first be understood. In this section, both linear and circular convolution are explained and the relationship between convolution and matrix multiplication is discussed [23].

4.1.1 Linear Convolution

Let $v[k] \in [0, p-1]$ and $x[k] \in [0, n-1]$ be signals in the discrete time domain k. The convolution of v with x shall be denoted by

$$y[k] = v[k] * x[k] \tag{4.1}$$

$$= \sum_{\tau=0}^{n-1} v\left[k-\tau\right] x\left[\tau\right] \tag{4.2}$$

where $y[k] \in [0, m]$ with $m = p + n - 1$. Convolution can also be expressed as the matrix multiplication $y = Vx$ with $V \in \mathbb{R}^{m \times n}$ where each of the matrix elements are given by

$$y = Vx \tag{4.3}$$

$$= \begin{bmatrix} v_0 & & & & \\ v_1 & v_0 & & & \\ \vdots & v_1 & \ddots & & \\ v_{p-2} & \vdots & \ddots & v_0 & \\ v_{p-1} & v_{p-2} & \ddots & v_1 & v_0 \\ & v_{p-1} & \ddots & \vdots & v_1 \\ & & \ddots & v_{p-2} & \vdots \\ & & & v_{p-1} & v_{p-2} \\ & & & & v_{p-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix} \tag{4.4}$$

4.1.2 Circular Convolution

To develop the concept of circular convolution for a length n discrete sequence $x[k]$, we define a circular time-reversal and then apply a circular time-shift. Let $v[k] \in [0, p-1]$ and $x[k] \in [0, n-1]$ be signals in the discrete time domain k. The circular convolution of v with x shall be denoted by

$$y\left[k\right] = x\left[k\right] \circledast v\left[k\right] \tag{4.5}$$

$$= \sum_{\tau=0}^{n-1} x\left[\tau\right] v\left[\left\langle k - \tau \right\rangle_n\right] \tag{4.6}$$

where $y[k] \in [0, n-1]$. Circular convolution can also be expressed as the matrix multiplication $y = Vx$ where each of the matrix elements are given by

$$y = Vx \tag{4.7}$$

$$= \begin{bmatrix} v_0 & & v_{p-1} & v_{p-2} & \cdots & v_1 \\ v_1 & v_0 & & v_{p-1} & v_{p-2} & \cdots \\ \vdots & v_1 & v_0 & & v_{p-1} & v_{p-2} \\ v_{p-2} & \vdots & v_1 & v_0 & & v_{p-1} \\ v_{p-1} & v_{p-2} & \vdots & v_1 & v_0 & \\ & v_{p-1} & v_{p-2} & \vdots & v_1 & v_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix} \tag{4.8}$$

In the context of reduced-order models, the key difference between linear and circular convolution is that by convolving modulo-n, the output y is the same length as the input sequence x as opposed to linear convolution where the output sequence is of length $m > n$. It is this property that is used to reduce x from n to r as seen in later sections. From this point forward, we shall assume that all convolution operations in this research are circular.

4.2 The Wavelet Transform Using Filter Banks

In this section, the filter bank approach for taking the wavelet transform is introduced. Both the 1-dimensional and 2-dimensional wavelet transforms are discussed.

4.2.1 The 1-Dimensional Wavelet Transform

The matrix form of the orthogonal and bi-orthogonal transform can be implemented using a more computationally efficient approach using a bank of filters, known as a filter bank. In this approach, the model parameter x is treated as a signal in either the discrete time domain or in the z-transform domain. Let $x(z)$ be a signal to be filtered by a series of low-pass and high-pass wavelet filters $L_1(z)$, $H_1(z)$, $L_2(z)$, and $H_2(z)$ as shown in Figure 4.1 for 1 level of decomposition.

The filtering operation for the forward wavelet transform is the convolution of x with both low-pass $L_1(z)$ and high-pass $H_1(z)$ filters followed by decimation by a factor of 2 denoted by $(\downarrow 2)$. x_L and x_H are the low-pass and high-pass coefficients that form $\bar{x}$, the signal in the wavelet domain. The convolution operation followed by decimation can be expressed as

$$\begin{aligned}\begin{bmatrix} x_L \\ x_H \end{bmatrix} &= \begin{bmatrix} (\downarrow 2)\sum_{\tau} x[\tau]\, l_1[\langle k-\tau\rangle_n] \\ (\downarrow 2)\sum_{\tau} x[\tau]\, h_1[\langle k-\tau\rangle_n] \end{bmatrix} \\ &= \begin{bmatrix} \sum_{\tau} x[\tau]\, l_1[\langle k-2\tau\rangle_n] \\ \sum_{\tau} x[\tau]\, h_1[\langle k-2\tau\rangle_n] \end{bmatrix} \end{aligned} \tag{4.9}$$

where k denotes signals and filters in the discrete-time domain [43, 34] . The filters $L_1(z)$ and $H_1(z)$ on the left side of Figure 4.1 are known as the analysis filter banks.

The original signal x is reconstructed (or recovered) using the inverse wavelet transform. The 1D inverse wavelet transform is taken by interpolating x_L and x_H by a factor of 2, denoted by $(\uparrow 2)$, convolving the coefficients through the filters $L_2(z)$ and $H_2(z)$, and then adding the signals together as seen in Figure 4.1. The filters $L_2(z)$ and $H_2(z)$ on the right side of Figure 4.1 are known as the synthesis filter banks.

Let the forward and inverse wavelet transform of x be represented using transform notation as

$$\bar{x} = \Psi\{x\} \tag{4.10}$$

$$x = \Psi^{-1}\{\bar{x}\} \tag{4.11}$$

with

$$\bar{x} = \begin{Bmatrix} \hat{x} \\ \tilde{x} \end{Bmatrix} \tag{4.12}$$

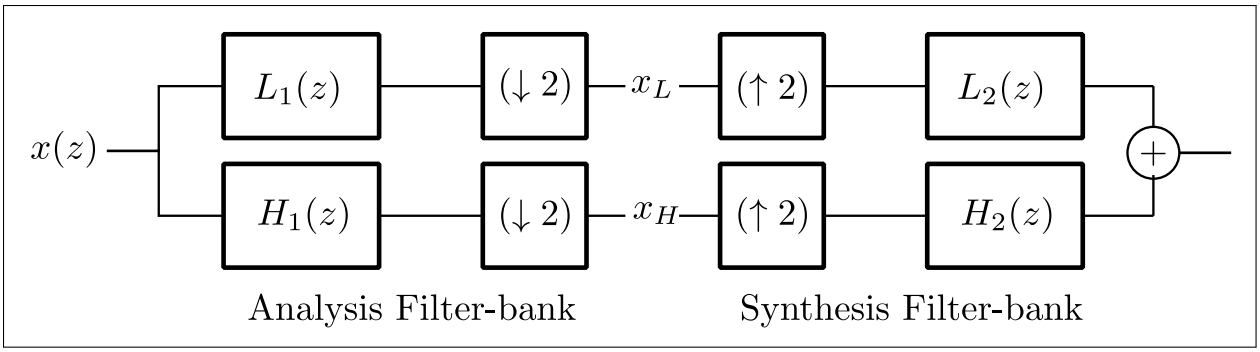


Figure 4.1: 1 Dimensional Wavelet Transform Filter Bank (1 Level of Decomposition)

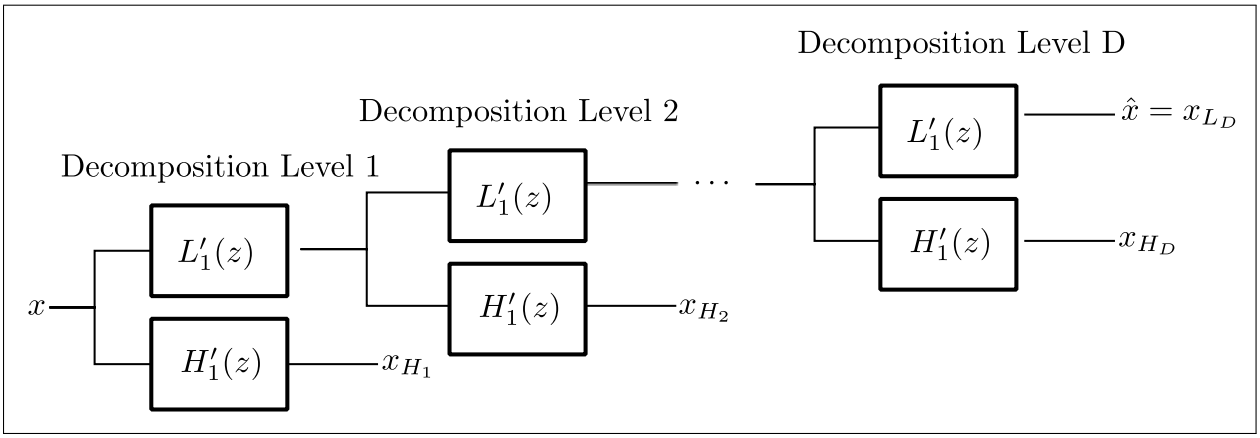


Figure 4.2: 1D Wavelet Transform Filter Bank at Decomposition Level D

and

$$\hat{x} = x_{L_D} \tag{4.13}$$

$$\tilde{x} = \left\{x_{H_D}, x_{H_{D-1}}, \cdots, x_{H_2}, x_{H_1}\right\} \tag{4.14}$$

where D is the level of decomposition for the wavelet transform.

When the filter bank is iterated D times as shown in Figure 4.2, a wavelet transform at decomposition level D is obtained. Here, $L_1'(z)$ and $H_1'(z)$ are used to simplify the filter bank notation and represent the combination of $L_1(z)$ and $H_1(z)$ with $(\downarrow 2)$ (compare to Figure 4.1). $\hat{x} \in \mathbb{R}^r$ are the approximation coefficients at decomposition level D obtained by iterating the analysis filter banks with the low frequency information x_L as the input at each iteration. The remaining co-

efficients $\tilde{x} \in \mathbb{R}^s$ are the detail coefficients. The dimension r of $\hat{x}$ can be determined according to

$$r = \frac{n}{2^D} \quad . \tag{4.15}$$

Eq. (4.15) is obtained using Figure 4.2 and tracing the input x through the filter bank. At the end of each decomposition level, the output has gone through one decimation operator $(\downarrow 2)$ and results in a reduction of length by a factor of 2 at each level. Assuming n is an integer power of 2, the maximum number of times the filter bank can be iterated is

$$D_{MAX} = \log_2(n). \tag{4.16}$$

The dimension of each of the vectors $x_{H_d} \in \mathbb{R}^{c_d}$ in $\tilde{x}$ from Eq. (4.14) can be determined by

$$c_d = \frac{n}{2^d} \quad d = 1, 2, \ldots, D. \tag{4.17}$$

4.2.2 The 2-Dimensional Wavelet Transform

The 2-dimensional (2D) forward wavelet transform is taken by filtering a matrix (image) $M \in \mathbb{R}^{n \times n}$ through a series of 1D filter banks where n is an integer power of 2. To perform the 2D forward wavelet transform, the rows of M are filtered and then the columns. When the filter bank is iterated D times, a 2D wavelet transform at decomposition level D is obtained as seen in Figure 4.3.

Denoting the 2D forward and inverse wavelet transform at decomposition level D as

$$\bar{M} = \Psi\{M\}_{2D} \tag{4.18}$$

$$M = \Psi^{-1}\{\bar{M}\}_{2D} \tag{4.19}$$

and the corresponding wavelet coefficients as $\bar{M}$, $\bar{M}$ shall be defined by taking the output coefficients at each level of decomposition in Figure 4.3 and arranging them by decomposition level into

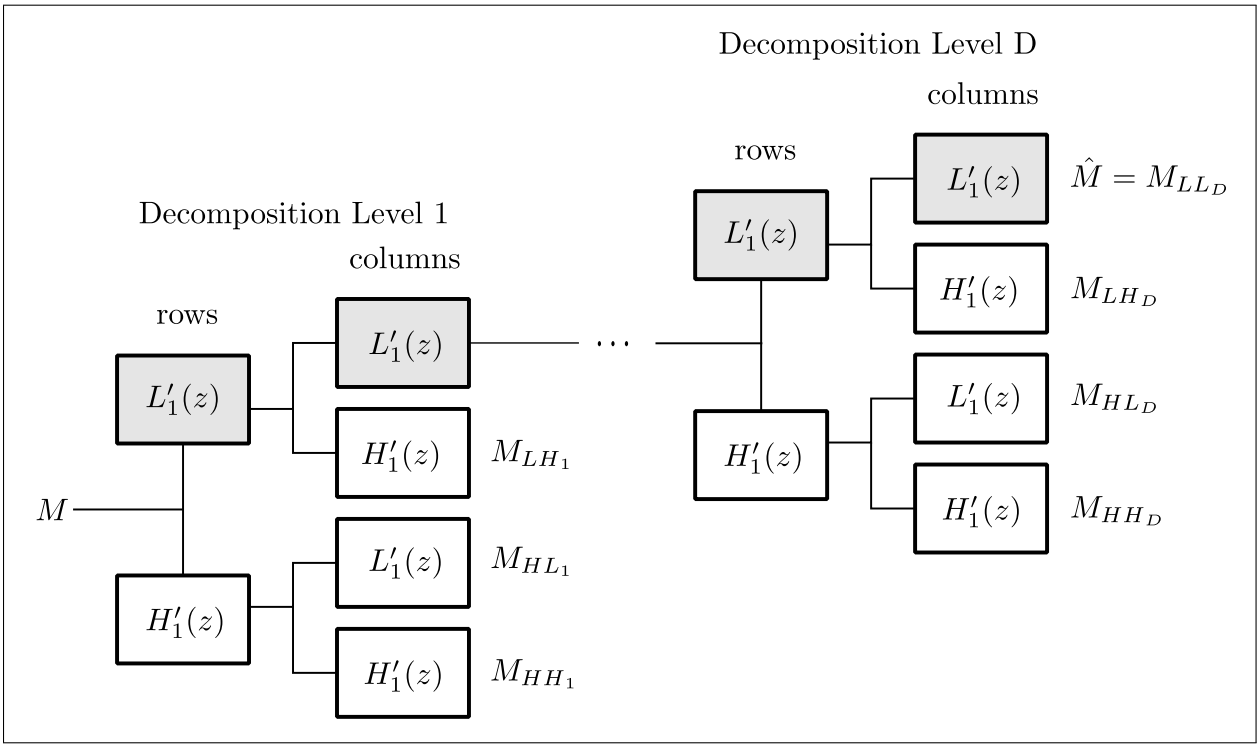


Figure 4.3: 2D Wavelet Transform Filter Bank at Decomposition Level D

a set of sub-bands as

$$\bar{M} = \left\{ \begin{array}{c} \hat{M} \\ \tilde{M} \end{array} \right\} \tag{4.20}$$

where

$$\hat{M} = M_{LL_D} \tag{4.21}$$

$$\begin{aligned} \tilde{M} = \{ & M_{LH_D}, M_{HL_D}, M_{HH_D} \\ & \cdots, M_{LH_d}, M_{HL_d}, M_{HH_d} \\ & \cdots, M_{LH_1}, M_{HL_1}, M_{HH_1} \} \quad . \end{aligned} \tag{4.22}$$

$\hat{M} \in \mathbb{R}^{(r \times r)}$ are the 2D the approximation coefficients at decomposition level D obtained by iterating the analysis filter banks with the low-low frequency information M_{LL} as the input at each

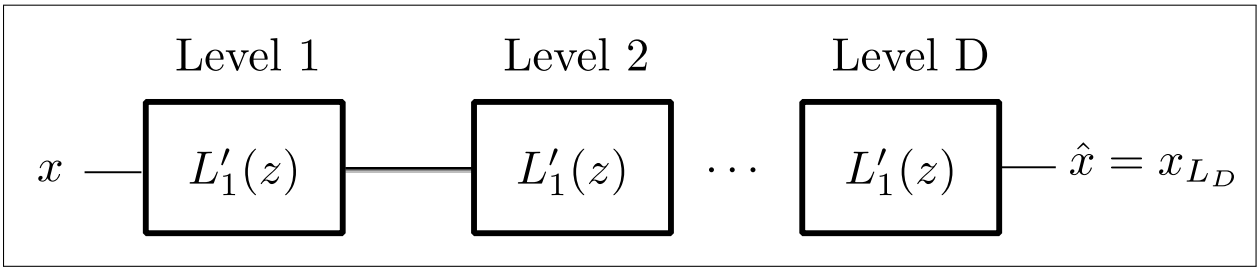


Figure 4.4: 1D Reduced Wavelet Transform

iteration. The remaining coefficients $\tilde{M}$ are the 2D detail coefficients. The dimensions $r \times r$ of $\hat{M}$ can be determined according to Eq. (4.16), and the dimensions for each of $M_{LH_d}, M_{HL_d}, M_{HH_d} \in \mathbb{R}^{q \times q}$ from Eq. (4.22) in $\tilde{M}$ can be determined by Eq. (4.17).

With expressions for the 1D and 2D wavelet transforms defined, we are now ready to introduce a method for obtaining a reduced-order model using wavelet transform filter banks.

4.3 Reduced-Order Models Using Wavelet Filter Banks

To meet our objective, we begin by seeking an expression for the reduced model parameters using the wavelet transform. Again, the cornerstone to wavelet reduced-order modeling is energy compaction where the majority of the energy is concentrated in the low-pass approximation coefficients and very little in the high-pass detail coefficients. We thus seek expressions for the reduced-order model using filter banks.

To achieve this, we begin by examining the expression for the forward reduced-order wavelet transform in matrix form in Eq. (3.47). This expression indicates that only the low-pass approximation wavelet coefficients are needed to obtain the reduced-order parameter space $\hat{x}$. The high-pass detail coefficients are never computed and are disregarded. This observation allows for a simplification in the 1D wavelet filter bank shown in Figure 4.2 where only the upper low-pass filter portion of the analysis filter bank is required as shown in Figure 4.4.

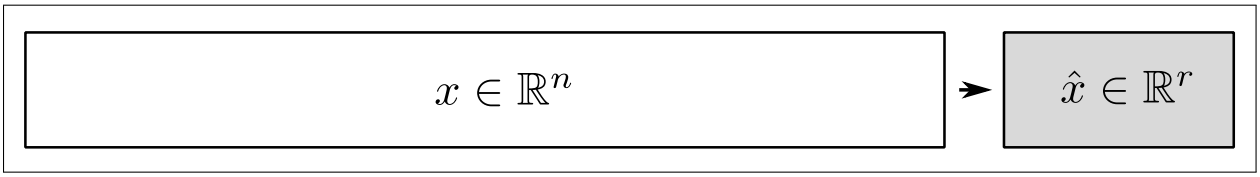


Figure 4.5: Parameter Reduction Using 1D Wavelets. Using the energy compaction property, the parameter space can be reduced from $x \in \mathbb{R}^n$ to $\hat{x} \in \mathbb{R}^r$.

We denote the 1D forward reduced-order wavelet transform using filter banks as

$$\hat{x} = \hat{\Psi}\{x\} \tag{4.23}$$

By iterating the filter bank in Figure 4.4 D times where $D \leq D_{MAX}$ and using the energy compaction property, we can reduce the parameter space of x as illustrated in Figure 4.5. Using the 1D reduced-order wavelet transform on x, the low frequency sub-band of $\bar{x}$ becomes the reduced parameter space $\hat{x} \in \mathbb{R}^r$.

To complement the forward reduced-order wavelet transform using filter banks, an expression to recover the full-order parameter space x^* from $\hat{x}$ is needed. By using similar observations for the inverse reduced-order wavelet transform in matrix form in Eq. (3.48), we note that only the low-pass synthesis filter is needed to recover an approximation of the full-order parameter space x^*. We denote the 1D inverse reduced-order wavelet transform using filter banks as

$$x^* = \hat{\Psi}^{-1}\{\hat{x}\}. \tag{4.24}$$

We now turn our attention to the reduced-order model. Recall that $\hat{J} = \Omega^T J_k \Phi$ is the general form for the reduced-order model and let $\Phi = \Omega = L^T$ as in the case of the Galerkin wavelet reduced-order model. By treating the Jacobian J_k as the 2D image M, then the reduced-order model $\hat{J} = L J_k L^T$ is computed by using only the low pass approximation wavelet coefficients and low-pass filters. Again, the high-pass detail coefficients are never computed which leads to a simplification in the 2D wavelet filter bank shown in Figure 4.3 where only the upper low-pass filter portion of the analysis filter bank is required as shown in Figure 4.6. We denote the 2D

forward reduced-order wavelet transform using filter banks as

$$\hat{M} = \hat{\Psi}\{M\}_{2D} \tag{4.25}$$

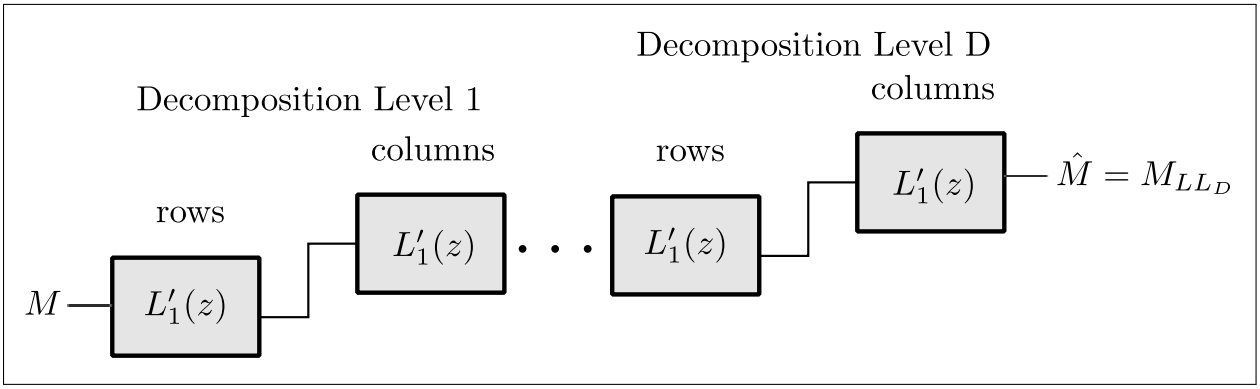


Figure 4.6: 2D Reduced Wavelet Transform

By iterating the filter bank in Figure 4.6 D times where $D \leq D_{MAX}$ and using the energy compaction property, we can reduce M as illustrated by Figure 4.7. Using the 2D reduced-order wavelet transform at decomposition level D on M, the low frequency subband of $\bar{M}$ becomes the reduced-order model $\hat{M} \in \mathbb{R}^{(r \times r)}$. To conclude this section, the governing equations for deriving wavelet reduced-order model using filter banks are given below for the Jacobian J, residual R, and the Newton step $\Delta\hat{x}$.

$$\hat{J} = \hat{\Psi}\{J\}_{2D} \tag{4.26}$$

$$\hat{R} = \hat{\Psi}\{R\} \tag{4.27}$$

$$\Delta x = \hat{\Psi}^{-1}\{\Delta\hat{x}\} \tag{4.28}$$

4.3.1 Computational Complexity

At this point, we have introduced two methods for obtaining the wavelet transform, one based on matrix multiplication and one based on convolution. Both methods are equivalent and produce the same result. However, the key difference in both methods is the computational complexity in the

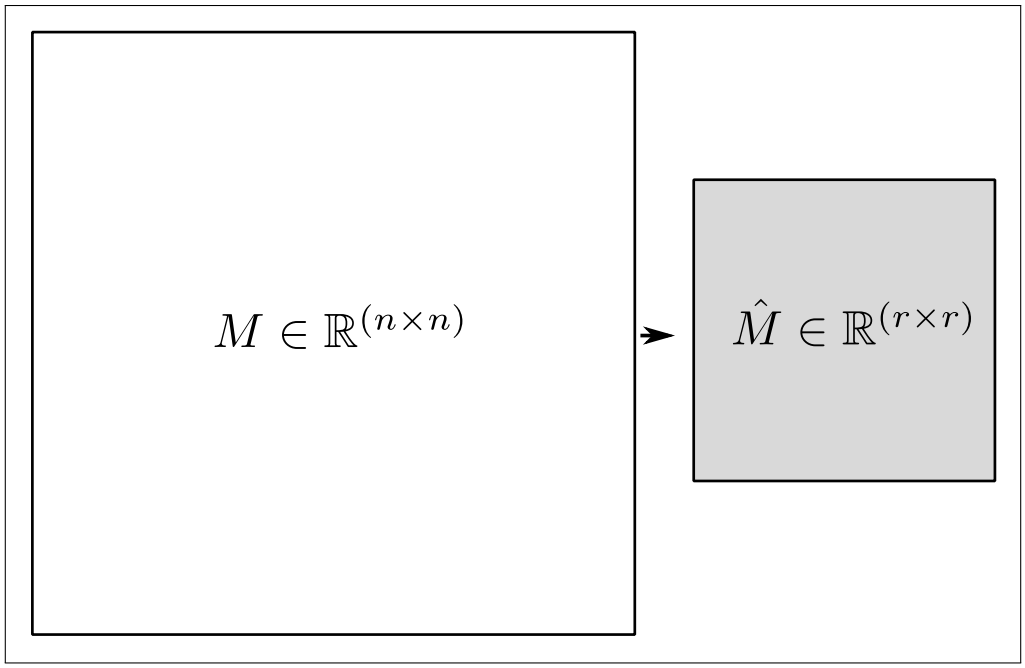


Figure 4.7: Model Reduction Using 2D Wavelets. Using the energy compaction property, the model M can be reduced from $M \in \mathbb{R}^{n \times n}$ to $\hat{M} \in \mathbb{R}^{r \times r}$

chosen approach. The matrix version of the 1D wavelet transform has $\mathcal{O}(n^2)$ complexity [15] as opposed to the convolution version which has $\mathcal{O}(n)$ complexity* [35, 38]. Here we assume that the low-pass filter $L_1(z)$ is a vector in $\mathbb{R}^p$. If we choose $L_1(z)$ with length $p << n$, then the convolution version of the wavelet transform is more computationally efficient. Using the filter bank approach for the wavelet transform is thus more computationally efficient than the matrix approach.

Now that we have an efficient algorithm for the wavelet transform, we are now ready to verify the computational savings experimentally.

*It is useful to equivalently express the computational complexity of the circular convolution operation by explicitly including the length of the wavelet filter coefficients p as $\mathcal{O}(pn)$. With this expression, it is clear that if p is not much greater than n, the size of p begins to impact the computational complexity. In the worst case, if $p = n$, then the computational complexity of the convolution operation will be $\mathcal{O}(n^2)$ [28]. In this case, the wavelet transform can be implemented with the fast Fourier transform with $\mathcal{O}(nlog_2(n))$ complexity [36].

4.4 Experimental Results

To evaluate the performance of the wavelet transform reduced-order model using filter banks, the technique was applied to the nonlinear Burgers' partial differential equation defined previously.

Algorithm 1 is modified to replace all instances of the matrix-based wavelet transform with the filter bank approach and is shown in Algorithm 2. Algorithm 2 is written to accept inputs in terms of the reduced-order wavelet transform $\hat{\Psi}$ and the desired decomposition level d. In lines 4 and 5, the residual R and Jacobian J are calculated. In line 6, R is reduced from $\mathbb{R}^n$ to $\mathbb{R}^r$ using the 1D forward reduced-order wavelet transform. In line 7, the reduced-order model $\hat{J}$ is obtained using the 2D forward reduced-order wavelet transform. The system of equations is solved in line 8 to obtain the reduced-order parameter space $\Delta\hat{x}$. Using the 1D inverse reduced-order wavelet transform, the full-order parameter space Δx is recovered and increased from $\mathbb{R}^r$ to $\mathbb{R}^n$ in line 9. For the inner loop, this process is repeated until the maximum number of iterations is reached or a suitable stopping criteria is met. For the outer loop, the process repeats until the maximum number of time steps is reached.

Figure 4.8 shows the speedup factor obtained with respect to the FOM during the online computation phase for the matrix-based wavelet reduced-order model, the SVD, and the filter bank wavelet reduced-order model. In this figure, results obtained using both orthogonal and bi-orthogonal wavelets are shown. We now see that by using the filter bank approach for obtaining the wavelet reduced-order model, the speedup between decomposition levels 1 and 6 now ranges from 3.5x to 2.75x. We thus conclude that the filter bank approach to model reduction offers a computational advantage for the lower and mid-range decomposition levels while the matrix version of model reduction begins to have the advantage when the model is very small.

By using the filter bank approach then, the wavelet reduced-order model now outperforms the SVD-based reduced-order model in terms of relative error, online computation time, offline computation time, and design simplicity.

Algorithm 2 Wavelet Filter Bank Reduced-Order Model Applied to Nonlinear PDEs

Input: $\hat{\Psi}\{\cdot\}$, d

for $j = 0$ **to** MaxTimeStep **do**
  Given an initial condition $x_{j,0}$
  for $k = 0$ **to** MaxIterations **or** stopping criteria met **do**
    Obtain the residual $R := R_k(x_{j,k})$.
    Obtain the Jacobian $J := J_k(x_{j,k})$.
    Reduce R to obtain $\hat{R}$.

$$\hat{R} = \hat{\Psi}\{R\}$$

    Reduce J to obtain $\hat{J}$

$$\hat{J} = \hat{\Psi}\{J\}_{2D}$$

    Solve for the reduced-order parameter space $\Delta\hat{x}$.

$$\hat{J}\Delta\hat{x} = -\hat{R}$$

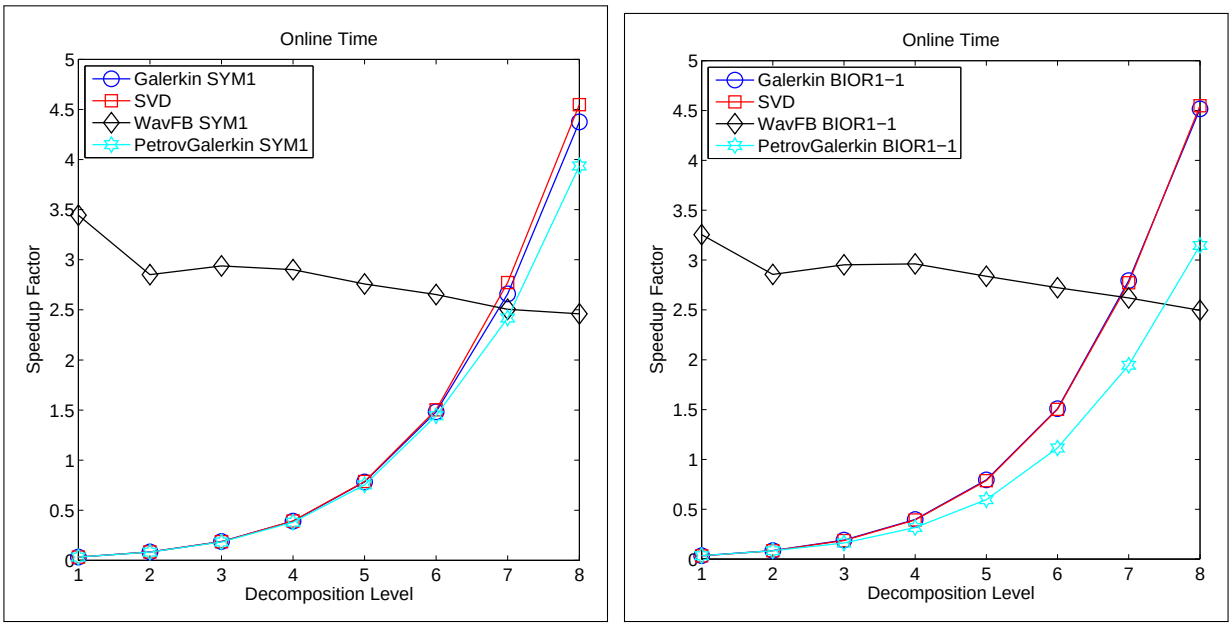
    Recover the full-order parameter space Δx.

$$\Delta x = \hat{\Psi}^{-1}\{\Delta\hat{x}\}$$

    Update

$$x_{j,k+1} = x_{j,k} + \Delta x$$

  end for
end for
return



(a) Orthogonal

(b) Bi-orthogonal

Figure 4.8: Online Speedup Using Wavelet Filter Bank Reduced-Order Models

Chapter 5

Conclusion

In this work, we proposed a novel method for model reduction using wavelets. Table 5.1 compares the filter bank approach for wavelet reduced-order models, the matrix method for wavelet reduced-order models, and the SVD approach and summarizes the overall results of the experiments. The advantage of using wavelet reduced-order models becomes clear in light of the disadvantages of the SVD. The wavelet reduced-order modeling approach does not require any offline computations or snapshots. Using the matrix version of wavelet model reduction only requires that the low-pass sub-matrices be formed before real-time simulations can occur.

Since the SVD relies on snapshots formed from a FOM, this method is data dependent, that is, the basis used for model reduction cannot be used to reduce a different FOM. This limitation does not exist with wavelet model reduction because the wavelet basis' are data independent.

There are additional benefits to using wavelet model reduction when the filter bank approach is used. When using filter banks, there is no need to store the matrices Ω and Φ. The low-pass filter coefficients can be obtained from a look-up table and the memory required to store the basis matrices is no longer needed and can be freed. The filter bank approach is thus also a memory efficient reduced-order modeling technique. In addition, the filter bank approach is more efficient and has computational complexity $\mathcal{O}(n)$ compared to $\mathcal{O}(n^2)$ for matrix methods.

We conclude that wavelet model reduction using filter banks enables the researcher to begin and execute online, real-time simulations faster while being memory efficient and simple to design.

Table 5.1: Model Reduction Summary

Model Reduction	$\mathcal{S}_{on}$	$\mathcal{S}_{off}$	Design Simplicity	Memory Efficiency	Data Independent
Wavelet ROM (filter bank)	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
Wavelet ROM	$-$	$\uparrow$	$\uparrow$	$\downarrow$	$\uparrow$
SVD	$-$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$

5.1 Future Work

Earlier in this work, we stated that model reduction using wavelets has not received much attention in the literature. As such, it is our hope that this dissertation will spark an interest in wavelet model reduction, and we provide the following ideas for future work.

First: In this research, wavelet reduced-order models were proposed using both orthogonal and bi-orthogonal wavelets. The numerical results thus far do not provide a clear advantage to orthogonal or bi-orthogonal wavelets. At this point, it is not known if orthogonal or bi-orthogonal wavelets are better for model reduction and this remains an open question. Second: The numerical experimentation presented in this work was based on the relatively simple Burgers' equation. Wavelet model reduction must be applied to more realistic, high dimensional problems. In particular, we hope to apply this method to the Army Underbody Blast problem in collaboration with researchers from Stanford and the Army High Performance Computing Research Center. Third: All wavelet model reduction algorithms were implemented in Matlab on a standalone system. An implementation of wavelet model reduction using high performance computing (HPC) techniques is necessary for wavelet model reduction to be applied to more realistic problems. Examples of HPC implementations include a version in MPI and/or a shared memory version using graphics processing units (GPUs).

References

[1] B.K Al-Abudi and L.A. George. Color image compression using wavelet transform. *International Congress for Global Science and Technology International Journal on Graphics, Vision and Image Processing*, December 2005.

[2] D. Amsallem. *Interpolation on Manifolds of CFG-Based Fluid and Finite Element-Based Structural Reduced-Order Models for On-Line Aeroelastic Predictions(Dissertation)*. Stanford University, 2010.

[3] E. Arian, M. Fahl, and E.W. Sachs. Trust-region proper orthogonal decomposition for optimal flow control. *Technical Report, Institute for Computer Applications in Science and Engineering*, 2000.

[4] B.V. Babu and K.K.N Sastry. Estimation of heat transfer parameters in a trickle-bed reactor using dierential evolution and orthogonal collocation. *Computers and Chemical Engineering*, 23, January 1999.

[5] T. Bui-Thanh, K. Wilcox, and O.Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 2008.

[6] L. Cai and R.E. White. Adaptive control of a wake flow using proper orthogonal decomposition. *Shape Optimization and Optimal Design, Lecture Notes in Pure and Applied Mathematics*, 2001.

[7] L. Cai and R.E. White. Reduction of model order based on proper orthogonal decomposition for lithium-ion battery simulations. *Journal of the Electrochemical Society*, 2009.

[8] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient nonlinear model reduction via a least-squares petrov-galerkin projection and compressive tensor approximations. *International Journal For Numerical Methods in Engineering*, 2009.

[9] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science: Special Section on Computational Science*, Vol. 78, No. 7, 2000.

[10] I. Daubechies. Ten lectures on wavelets. *SIAM: Society for Industrial and Applied Mathematics*, 1992.

[11] D. Donoho, A. Maleki, and M. Shahram et. al. Wavelab 850.

[12] D.L. Fugal. *Conceptual Wavelets In Digital Signal Processing*. Space and Signals Technologies LLC, 2009.

[13] A.D. Garnadi and D. Kurn. 2-D numerical reconstruction of electrical impedance tomography using iteratively regularized gauss-newton algorithms with a-posteriori parameter choice rule. *Sixth Asian Control Conference*, July 2006.

[14] E. Gilden. *Model and Controller Reduction of Large-Scale Structures Based on Projection Methods (Dissertation)*. The University of Texas at Austin, 2006.

[15] A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing*. Addison-Wesley, 2003.

[16] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. *Lecture Notes in Computational and Applied Mathematics*, 2005.

[17] C. Hogea, C. Davtzikos, and G. Biros. An image-driven parameter estimation problem for a reaction-diffusion glioma growth model with mass effects. *Journal of Mathematical Biology*, 56, June 2008.

[18] K. Kunisch and S. Volkwein. Control of burgers equation by reduced order approach using proper orthogonal decomposition. *Journal of Optimization Theory and Applications*, 1999.

[19] S. Lawson and J. Zhu. Image compression using wavelets and JPEG2000: a tutorial. *Journal of Electronics and Communication Engineering*, 2002.

[20] L. Lima, M.F. Brescia, N. Adami, and A. Signoroni. Wavelet-based encoding for hd applications. *IEEE International Conference on Multimedia*, 2007.

[21] A. Marquez, J.J.E Oviedo, and D. Odloak. Model reduction using proper orthogonal decomposition and predictive control of distributed reactor system. *Journal of Control Science and Engineering*, 2013.

[22] P.M. Meaney, Q. Fang, and T. Rubaek. Log transformation benefits parameter estimation in microwave tomographic imaging. *Medical Physics*, 34, June 2007.

[23] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, 2nd edition, 2001.

[24] M.Unser, A. Aldroubi, and A. Laine. Wavelets in medical imaging. *IEEE Transactions on Medical Imaging*, 2003.

[25] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[26] G. Obinata and B.D.O Anderson. *Model and Controller Reduction of Large-Scale Structures Based on Projection Methods*. Springer, 2001.

[27] M.R. Opmeer. Model order reduction by balanced proper orthogonal decomposition and by rational interpolation. *IEEE Transactions on Automatic Control*, 2012.

[28] J.G. Proakis and D.G. Manolakis. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Inc., 1996.

[29] R.M. Rao and A.S. Bopardikar. *Wavelet Transforms: Introduction to Theory and Applications*. Pearson Education, 1998.

[30] M. Rathinam and L.R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 2003.

[31] D. Russo. Determining soil hydraulic properties by parameter estimation: On the selection of a model for the hydraulic properties. *Water Resource Research*, 24, 1988.

[32] K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann, 2nd edition, 2000.

[33] G. Strang and T. Nguyen. Wavelets and filter banks. *Wellesley-Cambridge Press*, 1996.

[34] P.P Vaidyanathan. *Multirate Systems and Filter Banks*. Pearson Education, 2004.

[35] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall, 1995.

[36] M. Vrhel, C. Lee, and M. Unser. Fast continuous wavelet transform. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[37] J.S. Walker. *A Primer on Wavelets and their Scientific Applications*. Chapman and Hall CRC, 1999.

[38] R. Wang. *Introduction to Orthogonal Transforms With Applications in Data Processing and Analysis*. Cambridge University Press, 2013.

[39] G.B. Whitham. *Linear and Nonlinear Waves*. John Wiley and Sons, 1974.

[40] K. Willcox and J. Peraire. Balanced model reduction using the proper orthogonal decomposition. *Journal of the American Institute of Aeronautics and Astronautics*, 2002.

[41] H. Yamada, K. Yamamoto, Y. Yamaguchi, and M. Sengoku. Scattering parameter estimation accuracy of root-music algorithms in electromagnetic wave scattering measurements. *Electronics and Communications in Japan (Part II: Electronics)*, 79, 1996.

[42] W. Yan, Y. Chen, and S. Tai. Compressing discrete cosine transform coefficients by modified set partitioning in hierarchical trees. *Journal of Electronic Imaging*, 14(4), Dec 2005.

[43] S-H. Yang and W-J. Liao. Perfmance analysis of embedded-wavelet coders. *Journal of Optical Engineering*, 44(9), Sept 2005.

Curriculum Vitae

Miguel Hernandez IV graduated third in his class in 1998 from Parkland High School and began studies at The University of Texas at El Paso as a Presidential Scholar. In 1999, Miguel was awarded The National Action Council for Minorities in Engineering (NACME) Corporate Scholarship. In 2000 and 2002, he became a member of Eta Kappa Nu and Tau Beta Pi respectively. During the summers of 2001 - 2003, he interned at Texas Instruments in Houston, Texas. Miguel received a Bachelor of Science in Electrical Engineering with Highest Honors (Summa Cum Laude) in May 2003. In the Fall of 2003, he entered the Graduate School at The University of Texas at El Paso as a fellow of the National Consortium for Graduate Degrees for Minorities in Engineering and Science (GEM) and the National Science Foundation (NSF). During the summer of 2004, he interned one last time with Texas Instruments. In May 2005, Miguel received a Master of Science in Electrical Engineering and began employment with the U.S. Army Research Laboratory as an Electronics Engineer. During the Fall of 2009, he returned to the University of Texas at El Paso to begin graduate studies in Computational Science and received a Master of Science in Computational Science in December 2011. In December 2013, Miguel received a Doctor of Philosophy in Computational Science.