2015-01-01

Contributions to the Solution of Large Nonlinear Systems via Model-Order Reduction and Interval Constraint Solving Techniques

Leobardo Valera
University of Texas at El Paso, lvalera@miners.utep.edu

CONTRIBUTIONS TO THE SOLUTION OF LARGE NONLINEAR SYSTEMS VIA MODEL-ORDER REDUCTION AND INTERVAL CONSTRAINT SOLVING TECHNIQUES

LEOBARDO VALERA

Computational Science Program

APPROVED:

Martine Ceberio, Chair, Ph.D.

Sergio Cabrera, Ph.D.

Vladik Kreinovich, Ph.D.

Natasha Sharma, Ph.D.

Charles H. Ambler, Ph.D.
Dean of the Graduate School

Dedico este trabajo a mi querida familia, quien me ha acompañado en este sueño. Sin ellos nunca hubiese podido lograrlo. A mi esposa Ymelda y a mis niños, Dayana y Leo David

♡

CONTRIBUTIONS TO THE SOLUTION OF LARGE NONLINEAR SYSTEMS VIA MODEL-ORDER REDUCTION AND INTERVAL CONSTRAINT SOLVING TECHNIQUES

by

LEOBARDO VALERA

THESIS

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

MASTER OF SCIENCE

Computational Science Program

THE UNIVERSITY OF TEXAS AT EL PASO

December 2015

Acknowledgements

I would like to express my sincere gratitude to my mentor Dr. Ceberio for her guidance, encouragement, and infinite patience. I am thankful for the opportunity I have been given to work with her. I also want to acknowledge Dr. Vladik Kreinovich, Dr. Sergio Cabrera, and Dr. Natasha Sharma for kindly accepting being part of my thesis committee.

My friends, Manuel Maia, Horacio Florez, Reinaldo Sanchez, Dr. Dura and Luis Gutierrez for helping me with the thesis' corrections, and special thanks to Cindy, who is always willing to help.

My endless thanks goes to my dear wife Ymelda for her love and support in every moment; to my kids Dayana and Leo David. This is dedicated to you all.

I also want to thank Dr. Argaez for inviting me to obtain my PhD degree at UTEP.

Thanks to the Computational Science Program and Department of Mathematical Sciences, and all my professors: Drs. Pownuk, Moore, Romero, Yi, Argaez, Sharma, Kreinovich, and Cabrera.

This work was supported by the Department of the Army ARL Grant project number: 226351344A.

Abstract

Many engineering problems boil down to solving partial differential equations (PDEs) that describe real-life phenomena. Nevertheless, efficiently and reliably solving such problems constitutes a major challenge in computational sciences and in engineering in general.

PDE-based systems can reach sizes so large after they are discretized. The large size in these problems generate several issues, among them we can mention: large space of storing, computing time, and the most important, lost of accuracy. A popular approach to solving such problems is assume that the PDE's solution is in a subspace, and the solution is sought there. This assumption and later searching is named Model-Order Reduction (MOR). As we have mentioned before, MOR aims at reducing the size of the original large problem by projecting it onto a subspace. The quality of MOR is highly dependent on the right choice of the projection. Assuming that the projection is relevant, i.e. the behavior of the projected system reproduces that of the original system, the projected smaller system is solved much more easily and its solution is "uncompressed" into the solution of the original system. Identifying effective projections is still an open problem. Among existing popular approaches are Krylov methods, Proper Orthogonal Decomposition (POD), and Wavelets. None of these methods is perfect, each with has pros and cons.

Once the original problem is reduced, challenges still exist: nonlinear problems, solving it require going back and forth between compressed and original problems in order to solve them, which is time and space consuming. So there is a need for clever approaches to optimization that allow avoiding these pitfalls.

In any case, we can consider the following added challenges: (1) No current popular technique is global, which means that there is no evidence that the solutions we obtain are the best. (2) None of these techniques take into account additional constraints, such as contact constraints, that should be satisfied by resulting solutions. (3) Uncertainty is not taken into account and thus the solution that are obtained may not be robust enough to

possible variations in the input or the model.

The task is to address each of the above-mentioned challenges, which all tend to achieving the same goal: efficiently and reliably solving large nonlinear systems. To do so, we propose (i) to further study using wavelets for projection / compression as well as to explore the design of cost-efficient snapshot-based approaches that require minimal expert knowledge, (ii) to employ intervals and constraint-solving techniques for directly address the nonlinear systems, and (iii) to analize regularization-based approaches for optimization to ensure faster convergence and less back-and-forth computations between smaller and larger systems.

Table of Contents

Chapter 1

Introduction

Every block of stone has a statue inside it and it is the task of the sculptor to discover it.

Michelangelo

1.1 Motivation and Background

1.1.1 Motivation

Many real life problems can be represented by mathematic models. Representations vary from the simplest one, to more complicated. For example, imagine a farmer who wants to know how many cows and chickens he has, and the only information available is that they are 10 animals in total, all of them with 32 legs. In this case, the farmer only has to solve a small linear system of equations in order to obtain the answer sought. Another example, a little more complicated, could be locating a device using global position system (GPS). This device uses the distance between the object and several satellites to triangulate its position. In this case, a nonlinear system of equations, only more complex that the earlier one, has to be solved in order to find the position of the object. These two examples have in common that the number of unknowns and the number of equations are very small, so they can be easily solved.

When the size of our problems is large, then we have to face other issues, the main one being that it is impossible to solve these kind of problems "by hand" and thus using a

computer is mandatory. The use of computers does not mean that we have overcome 100% of our issues, but other kind of issues appear, such as:

1. **storage:** the RAM memory, even the storing device of our computers is limited, so all the entries of a large system of equations cannot be stored;

2. **huge number of computations:** if the system of equations is too large, a huge number of operations is needed to obtain its solution of the system and these many operations tail another problem:

3. **rounding errors:** the amount of operations along with the fact that the computers cannot represent real numbers with exact precision, leads to have in every arithmetic operation a rounding error. This error is propagated and increases a a function of the number of operation used to solve the problem;

4. **many evaluations:** we have an additional problem when we solve a nonlinear system with Newton method that is, we have to evaluate many partial derivatives in order to obtain the Jacobian, and then we have to solve a linear system several times, as many as required to obtain the desired tolerance, thus the three previous issues are multiplied by the number of iterations done to get the solution.

Other examples of phenomena in the real life that we can represent by mathematic models could be partial differential equations (PDEs). These kind of equations arise in many engineering problems describing phenomena such as the distribution of heat in a given rod or plate over time (heat equation), description of waves like vibrating string, and sound and water waves (wave equation), gas dynamics and traffic flow (Burger equation) [57, 68, 51].

Mathematicians have proven that not all ordinary differential equations can be solved analytically, moreover, only a finite number of kind of ordinary differential equations can be solved by this way. Even worst, if we are speaking about partial differential equations,

classical examples, like heat equation and wave equation only can be solved if the domain is very simple, disks or squared domains, otherwise, analytical solutions can not be obtained.

The way used to find an approximation of the solution of differential equations, either ordinary or partials, is discretizing the domain and forming a system of algebraic equations, this system of equations can be linear or nonlinear depending upon the nature of the PDE.

In order to obtain a good accuracy in the approximation of the solution sought, the domain has to be discretized in many elements and nodes, leading to a large system of equations inheriting the issues mentioned above.

1.1.2 Background

We already have mentioned that many natural phenomenal can be modeled or can be reduced to a system of equations, and the dimension of these systems is large generally, which represents issues that have to be overcome. Now, the question is: how can we overcome those issues? the answer is easy to say, "we need to reduce the system", in other words, we have to find a subspace where the solution lies, and as consequence to reduce the number of computations. The process of finding a subspace where to look for the solution is known as **Model-Order Reduction (MOR)**, and it is explained in detail in further chapters.

In this thesis we only study three different MOR's approaches :

First, we study the Krylov method, which searches the solution of linear system of equations in the subspace spanned by vectors

$$\mathcal{K}_k(A, b) = \{b, Ab, A^2b, \ldots, A^kb\} \tag{1.1.1}$$

for large enough k. We expect A^kb to be nearly parallel to v_1, where v_1 is the eigenvector corresponding to the largest eigenvalue (in absolute value). In this case (Eq. (1.1.1)) becomes linearly dependent [66]. In order to avoid that instability, the Arnoldi method is introduced. This method uses the stabilized Gram–Schmidt process to produce a sequence

of orthonormal vectors, $q_1, q_2, q_3, \ldots$, called the Arnoldi vectors, such that for every k, the vectors $q_1, \ldots, q_k$ span the Krylov subspace $\mathcal{K}_k$ [54].

The number of iteration steps can be very high with the Arnoldi method. This number is, of course, not predictable and depends on the properties of the matrix, in particular the distribution of its eigenvalues, but also on the initial vectors. High iteration counts entail a large memory requirement to store Arnoldi's vectors and a high amount of computation because of the growing cost of the reorthogonalization. Sorensen proposed a restarted version of Arnoldi method, which consists of limiting the dimension of the search space. This means that the iteration is stopped after a number of steps, the last approximation obtained is taken as a new initial point, and resumes the Arnoldi method [64, 24, 63].

Even though the Krylov method or its variant Arnoldi or restarted Arnoldi method is able to give a subspace where the solution of the problem lies, it may take a long time to give it, or the dimension of the subspace is too large that it does not represent any advantage to work with MOR using such a subspace.

Second, we study the possibility of using wavelets as MOR method due to their property known as energy compaction. The energy compaction property means that the wavelet tends to concentrate energy in the low frequency sub-band of the its decomposition, i.e, $||x|| \approx ||\tilde{x}||$, where $\tilde{x} = Lx$ is the lowpass sub-band [31, 67]. In this case we can use the subspace spanned by L^T. If x is the solution of a system of equations coming from the discretization of a PDE, this method is really good because x is suppose to be "enough smooth". Otherwise, if x has a erratic behavior, the approximation given for this method is not going to have a good accuracy. In the worst scenario if x is in the subspace spanned by H^T, where H is the highpass sub-band, it will be impossible by this method to give any approximation to the solution since $Lx = 0$.

Finally, we studied a method named Proper Orthogonal Decomposition (POD), which is based on Principal Component Analysis (PCA) [10, 41, 50]. The idea behind POD is as follows: The high dimensional Full Order Model (FOM) is solved and m measurements (commonly referred to as samples or snapshots) are taken of the n state variables. 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 $\sigma_1,\ \sigma_2, \ldots, \sigma_r,\ r < n$ column vectors are selected from the matrix U which correspond to the r largest singular values and they are used to form a basis which is in turn used for model reduction [11].

Although the POD is one of the most popular approaches to model reduction, it has several disadvantages. We can mention as its main drawback that it 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 inputs. The quality of the resulting MOR depends heavily on the choice of parameters and inputs over which the snapshots are computed. A good overview of the Wavelets Method and POD method can be found in [27, 28].

We solve different problems, using the approaches referred and we realize that the POD method gives better results in comparison with the other two, but it has the disadvantage mentioned before.

Once the basis for the model reduction is selected, challenges still exist: when we look for a faster solution of such systems of equations in a subspace of the original domain, and we are using Model-Order Reduction, we then have to handle over-determined systems. These can be solved by using the Gauss-Newton method, which in non-zero-residual cases is not as efficient as needed [18].

Additional challenges include: (1) No current popular technique is global, which means that there is no evidence that the solutions we obtain are the best. (2) None of these techniques takes into account additional constraints, such as contact constraints, that should be satisfied by the achieved solutions. (3) Uncertainty is not taken into account and therefore there is no evidence that the obtained solutions are robust to the possible variations of the input or the model.

1.2 Contribution

1. POD has proven to be one of the most efficient methods to obtain a reduced basis to apply MOR to a nonlinear problem

$$F(x, \lambda) = 0, \tag{1.2.1}$$

but it is inconvenient because it has to solve the FOM several times for different values of λ. Instead of that, we propose to use interval and constraint-solving techniques [43] to solve a modified version of Eq. (1.2.1)

$$F(x, \mathbf{I}) = 0, \tag{1.2.2}$$

where $\mathbf{I}$ is an interval containing all the parameters used to obtain the snapshots, and thus to get an enclosed picture of the all snapshots at once. Then we can apply **singular value decomposition (SVD)** to obtain the reduce basis and to proceed as before.

2. As before mentioned, once we have selected the basis to apply the MOR method, and after evaluating the approximation of the solution over the subspace spanned by such basis, the system of equations becomes an over-determined system of equations and we have to apply the Gauss-Newton method to obtain the solution, if the solution is not in the subspace but is close to it, with respect to the orthogonal projection, the Gauss-Newton method will have a q-linearly rate of convergence. We propose to use techniques inspired by Levenberg and Marquardt [26, 40] to improve the convergence rate of such systems.

3. Finally, we use Interval Constraint Solving Techniques to get over the local property of the current techniques (Newton or Gauss-Newton method) and to get all the solutions of the problem, in case it has more than one solution. We also use Interval Constraint Solving Techniques to solve nonlinear PDE with uncertainty in the initial conditions and in the boundary conditions with the intention of knowing how the dependency

of the solutions with respect to such uncertainties is, and to predict future behavior of the dynamic systems given the values at some time/location.

1.3 Outline

In Chapter 2, we briefly revisit the definitions and results from basic linear algebra necessary to understand MOR; we also explain what principal component analysis is and how it is calculated, later we define in a formal and general way the concept of MOR; we also define the concept of Interval constraint solving techniques and finally we introduce the basic operations in interval arithmetic, which opens us the door to the computation of the Interval Newton method.

In Chapter 3, we describe the statement of the problem, and also the three more common known methods will be defined and presented. The first method described is based on Krylov subspace methods. The second method uses wavelets in order to obtain a reduction of the large systems. Finally we explain in detail POD. Each of these methods have advantages and disadvantages, that will be shown when we use them to solve different problems. We will also solve numerically four problems by using the three methods mentioned. These problems are particular cases of: linear systems of equations, nonlinear systems of equations, the heat equation as an example of a linear partial differential equation and finally, and the Burger equation as example of a nonlinear partial differential equation.

Chapter 4 presents an alternative way to compute snapshots to find a basis for POD method, we show a comparative table with respect to the traditional way to find such snapshots.

It is also presented as a contribution the use of the Levenberg-Marquardt to increase the rate of convergence when Gauss-Newton method is used in conjunction with the MOR.

Last chapter presents the conclusions and the future work.

Chapter 2

Preliminaries

To know what you know and what you do not know, that is true knowledge.

Confucius

In fields like sciences and engineering, it is well know that a linear system of equations can be seen as a linear transformation from a vector space into another (possibly the same), and when we are solving a linear system of equation we are finding in the domain, a vector x, which is, pre-image of a vector b in the range of such linear transformation.

What is not completely understood is that, when the linear transformation is from a vector space over itself, the solution lies in the subspace spanned by power of $\{A^i b\}_{i=1}^{i=k}$, for some k.

In this chapter we present some of the necessary concepts that support the theory of Model-Order reduction (MOR). We can mention, the concept of vector space, subspace, basis; also we present the concept of principal component analysis and its relation with linear algebra, concept that will be very important to understand one of the methods studied in further chapters. Finally, we use all the concepts mentioned before to give a formal definition of MOR.

2.1 Basic Concepts of Vector Spaces

The following is one of the principal definitions in linear algebra.

Definition 1. *A* **vector space over a field** $\mathbb{F}$ *consists of a set* V *together with a binary operation* $+$*, called* **addition** *and an external scalar operation* $\mathbb{F} \times V \longrightarrow V$*, called* **scalar multiplication***, which satisfies the following axioms:*

1. *Addition is commutative,* $\alpha + \beta = \beta + \alpha$*, for all* $\alpha, \beta \in V$*.*
2. *Addition is associative,* $\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$*, for all* $\alpha, \beta, \gamma \in V$*.*
3. *There is a unique element* $\mathbf{0}$ *in* V*, such that* $\alpha + \mathbf{0} = \alpha$*, for all* $\alpha \in V$*.*
4. *For each element* $\alpha \in V$ *there is a unique element* $-\alpha$ *in* V *such that* $\alpha + (-\alpha) = \mathbf{0}$*.*
5. $1\alpha = \alpha$*, for all* $\alpha \in V$*.*
6. $(c_1c_2)\alpha = c_1(c_2\alpha)$*, for all* $c_1, c_2 \in \mathbb{F}, \alpha \in V$*.*
7. $(c_1 + c_2)\alpha = c_1\alpha + c_2\alpha$*, for all* $c_1, c_2 \in \mathbb{F}, \alpha \in V$*.*
8. $c(\alpha + \beta) = c\alpha + c\beta$*, for all* $c \in \mathbb{F}, \alpha, \beta \in V$*.*

In this context, the elements of V are called **vectors**. Elements of $\mathbb{F}$ are called **scalars**, and scalar multiplication is indicated by simple juxtaposition of a scalar (on the left) with a vector. For us, the most important field is $\mathbb{R}$, the field of real numbers.

Let $\mathbb{F}$ be any field and let n be any positive integer. Define $\mathbb{F}^n$ to be the set of all n-tuples of elements of $\mathbb{F}$. That is, $\mathbb{F}^n = \{(x_1, \ldots, x_n) \mid x_1, \ldots, x_n \in \mathbb{F}^n\}$. Then $\mathbb{F}^n$ is a vector space over $\mathbb{F}$ with respect to the operations

1. $(x_1, x_2, \ldots, x_n) + (y_1, y_2, \ldots, y_n) = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n)$,
2. $c(x_1, x_2, \ldots, x_n) = (cx_1, cx_2, \ldots, cx_n)$.

In the special case $\mathbb{F} = \mathbb{R}$, we speak of real n-space, $\mathbb{R}^n$. In particular, when $n = 3$, $\mathbb{R}^3$ has the geometrical interpretation of ordinary three-dimensional space. Note in particular that the additive identity is $\mathbf{0} = (0, 0, 0)$ and the additive inverse of (x_1, x_2, x_3) is $(-x_1, -x_2, -x_3)$.

Definition 2. *Let V be a vector space over the field $\mathbb{F}$. A* **subspace** *of V is a subset W of V which is itself a vector space over $\mathbb{F}$ with respect to the additive and scalar operations defined on V.*

There is a simple criterion in order to establish that a subset W of a space V is a subspace of V.

Proposition 2.1.1. *(Subspace criterion) Let W be a nonempty subset of the vector space V. Then W is a subspace of V if and only if for each pair of vectors α, β in W and each scalar c in $\mathbb{F}$, the vector $c\alpha + \beta$ is in W.*

Consider the real vector space $V = \mathbb{R}^3$, viewed as a three-dimensional space. Then the xy-plane is a subspace of V. This is most easily seen geometrically: the vector $c\alpha + \beta$ on xy-plane remains on that plane if the vectors α and β are in that plane. The same argument shows more generally that any plane through the origin constitutes a subspace of V.

A generic construction of the subspaces of a vector space is the following.

Definition 3. *Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be vectors in a vector space V over $\mathbb{F}$. Then an expression of the form*

$$c_1\alpha_1 + c_2\alpha_2 + \cdots + c_n\alpha_n,$$

where $c_1, c_2, \ldots, c_n \in \mathbb{F}$, is called a ***linear combination*** *of the vectors $\alpha_1, \alpha_2, \ldots, \alpha_n$. The set of all such linear combinations is called the* **span** *of $\alpha_1, \alpha_2, \ldots, \alpha_n$ and denoted* $\mathrm{Span}(\alpha_1, \alpha_2 \ldots, \alpha_n)$.

By using the subspace criterion, we can show that the $\mathrm{Span}(\alpha_1, \alpha_2 \ldots, \alpha_n)$ is a subspace of V.

Proposition 2.1.2. *Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be vectors in a vector space V over $\mathbb{F}$. Then* $\text{Span}(\alpha_1, \alpha_2 \ldots, \alpha_n)$ *is a subspace of V.*

Note that we can extend the notion of span to a set S (not necessarily finite) of vectors in a vector space V, with the understanding that linear combinations drawn from such a set may involve many vectors with nonzero coefficients only finitely. In this setting, Span(S) is the intersection of all subspaces of V which contain S.

If a subspace of V is such that $W = \text{Span}(S)$, for some $S \subseteq V$, we say that W is **generated or spanned by** S. We also say that this set spans W. In the case that V itself is spanned by a finite collection of vectors, we say that V is finitely generated or finite dimensional.

Let $V = \mathbb{F}^n$, where $\mathbb{F}$ is any field. Consider the family of n vectors

$$\begin{aligned} e_1 &= (1, 0, 0, \ldots, 0), \\ e_2 &= (0, 1, 0, \ldots, 0), \\ &\vdots \\ e_n &= (0, 0, 0, \ldots, 1). \end{aligned}$$

Clearly these vectors span V, since it is trivial to write any vector $(c_1, \ldots, c_n)$ as a linear combination of the $e_j's$,

$$(c_1, \ldots, c_n) = c_1 e_1 + \cdots + c_n e_n.$$

It follows that V is finite dimensional.

The following concept is important because it allows us to know when a system of equations has a solution and if it is unique.

Definition 4. *Let V a vector space over $\mathbb{F}$. A subset S of V is said to be* **linearly dependent** *if there exist distinct vectors $\alpha_1, \alpha_2, \ldots, \alpha_n$ in S and scalars $c_1, c_2, \ldots, c_n$, not all of which 0, such that*

$$c_1\alpha_1 + c_2\alpha_2 + \cdots + c_n\alpha_n = \mathbf{0}.$$

A set which is not linearly dependent is called **linearly independent**.

In $\mathbb{F}^n$, the vectors $e_1, \ldots, e_n$ are clearly linearly independent since

$$c_1e_1 + \cdots + c_ne_n = (c_1, \ldots, c_n)$$

and this can be $\mathbf{0}$ if and only if all of the coefficients are zero.

Definition 5. *A collection of vectors B in a vector space V is called a **basis** for V if the following two conditions are satisfied:*

1. *B is linearly independent,*
2. *B spans V.*

Note that the set $B = \{e_1, \ldots, e_n\}$ is obviously a basis for $\mathbb{F}^n$.

Proposition 2.1.3. *A subset B of a vector space V is a* **basis** *for V if and only if every vector in V can be written as a linear combination of the vectors in B in exactly one way.*

Definition 6. *If A is an $n \times n$ matrix over the field $\mathbb{F}$, a* **characteristic value (or eigenvalue) of A in $\mathbb{F}$** *is a scalar c in $\mathbb{F}$ such that the matrix $(A - cI)$ is singular.*

The polynomial $f = \det(xI - A)$ is monic[1] which has degree exactly n and $f(c) = 0$, such polynomial is called **characteristic polynomial**.

Let's describe some properties of eigenvalues of real symmetric matrices:

1. All eigenvalues are real.
2. Eigenvectors corresponding to distinct eigenvalues are orthogonal.
3. If $x^TAx > 0$ for each $x \neq 0$, i.e A is symmetric positive definite, all eigenvalues of A are positive.

Definition 7. *If A is an $n \times n$ matrix over F, we define the* **minimal polynomial** *for A as the smallest degree monic polynomial, p, such that $p(A) = 0$.*

[1]The nonzero coefficient of highest degree is equal to 1

Theorem 1. *If A is an $m \times n$ matrix with rank k, then it can be written in the form*

$$A = U\Sigma V^T \tag{2.1.1}$$

where U is $m \times m$ unitary matrix, V is an $n \times n$ unitary matrix, and Σ is an $m \times n$ matrix with $\sigma_{ij} = 0$ for all $i \neq j$ and $\sigma_{11} \geq \sigma_{22} \geq \cdots \geq \sigma_{kk} > \sigma_{k+1,k+1} = \cdots = \sigma_{qq} = 0$, where $q = \min\{m, n\}$. Such decomposition is called: **the singular value decomposition of A (SVD)**.

2.2 Principal Component Analysis

When information from a data sample is collected, usually we take the maximum number of variables. However, if we take too many variables from a data sample, for instance 20 variables, we must consider $\binom{20}{2} = 190$ possible correlation coefficients. If you have 40 variables that number is increased to 780. Obviously, in this case it is difficult to visualize relationships between variables. Another problem that arises is the strong correlation that often occurs between variables: if we take too many variables (which generally happens when much is not known about data, or we are only interested in exploratory tests), it is normal that they are related or they measure the same thing under different viewpoints. For example, in medical studies, blood pressure at the heart's outlet and out of the lungs are strongly related.

Therefore, it is necessary to reduce the number of variables. It is important to highlight that the concept of major information is related to the greater variability of the data or variance. The greater the variability (variance) of the data, the more information this data has.

Studying the relationships that exist between p correlated variables (which commonly measure information) transforms the original set of variables in another new set of uncorrelated variables together (that has no repetition or redundancy on the information) called a set of principal components.

2.2.1 Principal Components

Let us consider a number of variables $X = (x_1, x_2, \ldots, x_n)$ describing a group of objects or individuals and to calculate, from them, a new set of variables $(y_1, y_2, \ldots, y_n)$ uncorrelated with each other, whose variances will decrease gradually.

Each y_j (where $j = 1, \ldots, n$) is a linear combination of the original variables $x_1, x_2, \ldots, x_n$, i.e

$$\begin{aligned} y_j &= v_{1j}x_1 + v_{2j}x_2 + \cdots + v_{pj}x_n \\ &= Xv_j, \end{aligned}$$

where $v_j^{\mathrm{T}} = (v_{1j}, v_{2j}, \ldots, v_{pj})$ is a constant vector.

To keep the orthogonality of the transformation, we impose $||v_j|| = 1$.

The first component v_1 is calculated so y_1 has the greatest variance subject to the constraint that $||v_1|| = 1$. The second principal component v_2 is calculated so that the variables y_1 and y_2 are uncorrelated. Similarly are chosen $y_1, y_2, \ldots, y_p$, uncorrelated with each other.

The full principal components decomposition of X can therefore be given as

$$Y = XV,$$

where V is a $p \times p$ matrix whose columns are the eigenvectors of $X^T X$.

The principal component decomposition of X can be expressed in terms of singular value decomposition of X. Given

$$X = U\Sigma V^T,$$

then we have

$$\begin{aligned} Y &= XV \\ &= U\Sigma V^T V \\ &= U\Sigma. \end{aligned}$$

In practice, we initiate computations with p variables and we are left with a number of much smaller components that collect a large percentage of the variability. For instance, we take r variables, where r is the minimum positive integer such that:

$$\frac{\sum_{i=1}^{r} \sigma_i}{\sum_{i=1}^{p} \sigma_i} > tol.$$

where tol is an approximation of 1 by defect.

2.3 Model Order Reduction

The main idea of Model Order Reduction (MOR) is the based on the following fact:

Let $T : V \to V$ be a bijective linear transformation. Then for every $b \in V$ there exists a unique $x \in V$, such that $T(x) = b$. Every linear transformation has a matrix representation [29]. In this case let us call A the matrix representation of T. Thus, finding x such that $T(x) = b$ is equivalent to solving the linear system

$$Ax = b. \tag{2.3.1}$$

If the dimension of V is n, then Eq. (2.3.1) is $n \times n$ linear system of n equations and n unknowns.

We can assure that there exists W, a subspace of V, whose dimension is $k \ll n$ and such that $x \in W$. That is true because, in particular, the subspace generated by $\{x\}$ is a subspace of V, which contains x and whose dimension is $1 \ll n$.

Since W is a subspace of V, there exists a base $B = \{w_1, w_2, \ldots, w_k\}$ such that every element $w \in W$ can be expressed as a linear combination of the elements of B [14]. In particular, if $w = x$,

$$x = \sum_{i=1}^{k} y_i w_i. \tag{2.3.2}$$

Since every base uniquely determines a subspace of V, we can, without loss generality, speak about subspace W and its base without difference. By writing Eq. (2.3.2) in matrix form, we obtain

$$Wy = x. \tag{2.3.3}$$

After substituting Eq. (2.3.3) in Eq. (2.3.1), we have

$$(AW)y = b, \tag{2.3.4}$$

that can be solved using the normal equation [7]

$$(AW)^T(AW)y = (AW)^T b, \tag{2.3.5}$$

which is itself a $k \times k$ linear system of equations. After we identify y, we can use Eq. (2.3.3) to find x.

2.3.1 Approximation by Projection

This method truncates the solution of the original system to an appropriate basis. Let us illustrate this method by considering a basis transformation T that maps the original n-dimensional state space x into a vector that we will denote by

$$T(x) = \begin{pmatrix} T_1(x) \\ T_2(x) \end{pmatrix} = \begin{pmatrix} \hat{x} \\ \tilde{x}, \end{pmatrix}$$

where $\hat{x}$ is k-dimensional. Suppose that T has at least a right-inverse. Let us denote $S = T^{-1}$ then S can be written as

$$S = (S_1 \;\; S_2)$$

and

$$\begin{aligned} I &= \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} (S_1 \;\; S_2) \\ &= \begin{pmatrix} T_1 S_1 & T_1 S_2 \\ T_2 S_1 & T_2 S_2 \end{pmatrix} \qquad (2.3.6) \\ &= \begin{pmatrix} I_k & 0 \\ 0 & I_{n-k} \end{pmatrix}. \qquad (2.3.7) \end{aligned}$$

Since $T_1 S_1 = I_k$, we have $\Pi = S_1 T_1$ is an oblique projection along the kernel of T_1 onto the k-dimensional subspace that is spanned by the columns of the matrix S_1.

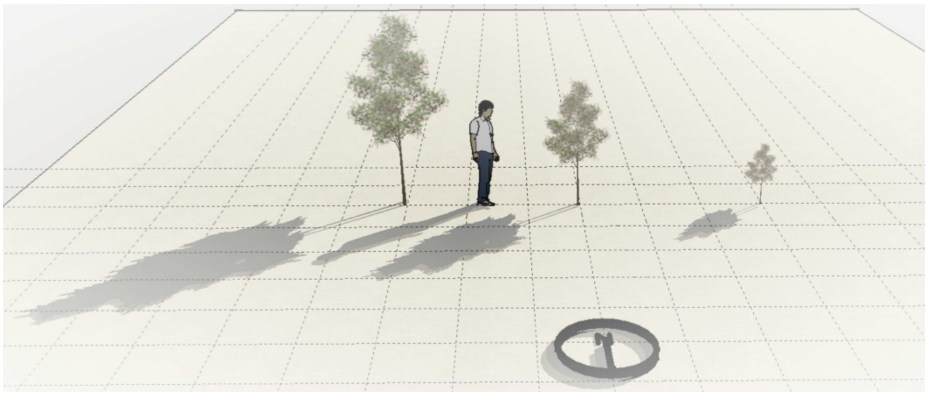


Figure 2.1: An oblique projection can be see as the shadow cast by objects on the ground when the sun is not directly vertical. Image taken from the site: http://www.schoolkitchengarden.com.au/design-your-garden/

Let

$$\begin{aligned} \frac{dx}{dt} &= f(x,u), \\ y &= g(x,u), \\ x(t=t_0) &= x_0 \end{aligned} \tag{2.3.8}$$

be the dynamical system, where u is the input of the system, y is the output, x the so-called *state variable*. If we substitute the projection into the dynamical system Eq. (3.4.1), we obtain

$$\begin{aligned} \frac{d\hat{x}}{dt} &= T_1\hat{f}(S_1\hat{x} + S_2\tilde{x}, u), \\ y &= \hat{g}(S_1\hat{x} + S_2\tilde{x}, u). \end{aligned} \tag{2.3.9}$$

The approximation occurs when we delete the terms involving $\tilde{x}$

$$\begin{aligned} \frac{d\hat{x}}{dt} &= T_1\hat{f}(S_1\hat{x}, u), \\ y &= \hat{g}(S_1\hat{x}, u). \end{aligned} \tag{2.3.10}$$

In order to obtain a good approximation to the original system, the term $S_2\tilde{x}$ must be sufficiently small.

2.4 Interval Constraint Solving Techniques

A *constraint problem (CP)* is a model that is composed of:

1. A set of real-valued variables, $\{x_1, x_2, \ldots, x_n\}$,
2. A set of interval domains, $I_1, I_2, \cdots, I_n$, such that $x_1 \in I_1, x_2 \in I_2, \cdots, x_n \in I_n$.
3. A set of numerical constraints over the given set of variables.

The problem consists of finding all values with respect to all constraints in the initial box:

$$I_1 \times I_2 \times \cdots \times I_n.$$

A solution to a CP is a complete assignment of values to the variables such that all constraints are simultaneously satisfied. A problem solver must find one or all such solutions.

The program used in this work, "RealPaver", uses CP and interval algorithms to process system of nonlinear constraints over the real numbers [23].

2.5 Interval Analysis

Let us start with the well known concept of interval[2]:

$$X = [\underline{X},\ \overline{X}] = \{x \in \mathbb{R} : \underline{X} \leq x \leq \overline{X}\}. \tag{2.5.1}$$

The concepts of intersection and union are the usual. The less familiar concept of *interval hull* of two intervals is defined as:

$$X \underline{\cup} Y = [\min\{\underline{X}, \underline{Y}\}, \max\{\overline{X}, \overline{Y}\}]. \tag{2.5.2}$$

A few other terms will be useful: The *width* of an interval X is defined and denoted by

$$w(X) = \overline{X} - \underline{X}. \tag{2.5.3}$$

[2]In this work, the term *interval* means closed interval

The *absolute value* of X, denoted $|X|$, is the maximum of the absolute values of its endpoints:

$$|X| = \max\{|\underline{X}|, |\overline{X}|\}. \tag{2.5.4}$$

Note that $|x| \leq |X|$, for every $x \in X$. The *midpoint* of X is given by

$$m(X) = \frac{\underline{X} + \overline{X}}{2}. \tag{2.5.5}$$

2.5.1 Operations of Interval Arithmetic

Since $x \in X$ means that $\underline{X} \leq x \leq \overline{X}$, and $y \in Y$ means that $\underline{Y} \leq y \leq \overline{Y}$ the followings operations are defined based on its infimum and supremum:

Addition

$$X + Y = [\underline{X} + \underline{Y}, \overline{X} + \overline{Y}], \tag{2.5.6}$$

Substraction

$$X - Y = [\underline{X} - \overline{Y}, \overline{X} - \underline{Y}], \tag{2.5.7}$$

Multiplication

$$X \cdot Y = [\min S, \max S], \tag{2.5.8}$$

where $S = \{\underline{X}\underline{Y}, \underline{X}\overline{Y}, \overline{X}\underline{Y}, \overline{X}\overline{Y}\}$,

Division (Extended version)

$$[a, b]/[c, d] = [a, b](1/[c, d]), \tag{2.5.9}$$

where

$$1/[c, d] = \{1/y : y \in [c, d]\}(c < d, \text{ any real numbers}). \tag{2.5.10}$$

If $0 \notin [c, d]$, we use ordinary interval arithmetic. If $0 \in [c, d]$, extended interval arithmetic specifies three cases:

1. If $c = 0 < d$, then $1/[c, d] = [1/d, \infty)$.

2. If $c < 0 < d$, then $1/[c, d] = (-\infty, 1/c] \cup [1/d, \infty)$.

3. If $c < d = 0$, then $1/[c, d] = (-\infty, 1/c]$.

Interval extension of a function

Definition 8. *Consider the real-valued function f, the function F is an* interval extension *of f, if for for degenerate interval arguments, F agrees with f [43]:*

$$F([x, x] = f(x). \tag{2.5.11}$$

In general, given a function f depending of n variables $x_1, x_2, \cdots, x_n$, an interval extension *of f is an interval-valued function F of n interval variables $X_1, X_2, \cdots, X_n$ such that for degenerate interval arguments*

$$F([x_1, x_1], [x_2, x_2], \cdots, [x_n, x_n]) = f(x_1, x_2, \cdots, x_n) \tag{2.5.12}$$

Chapter 3

Problem Statement and the State of the Art

If classical music is the state of the art, then the arts are in a sad state.

Frank Zappa (American Composer, Guitarist, Satirist and Song Writer, 1940-1993

Let us start this chapter with an small introduction of what to expect in it.

First, we describe the problem statement, later, we will describe three popular approaches of Model-Order Reduction: Krylov methods, Wavelets, and Proper Orthogonal Decomposition (POD). We end the chapter solving four problems of different kinds using the approaches that are studied. One of them is a parametric-family of linear system of equations, another one is a nonlinear system of equation, other will be a linear partial differential equation, and the last one is a nonlinear partial differential equation.

When solving each of the problems mentioned above with each of the methods presented, we take note about the time of computation, the dimension of the subspace and the precision of the approximation compared with the solution of the respectively problem solved with no reduction, i.e. the Full Order Model (FOM).

3.1 What are we trying to achieve?

Consider a system represented as an abstract blackbox input-output system Fig. 3.1 where

Figure 3.1: General input-output system

the blackbox system could represent:

1. A parametric linear system of equations Section 3.6.1, Eq. (3.6.1);

$$A(\lambda)x = b(\lambda) \tag{3.1.1}$$

   where $A(\lambda)$ is a $n \times n$ matrix and λ is the parameter of the system.

2. A parametric nonlinear system of equations Section 3.6.2, Eq. (3.6.3);

$$R(x; \lambda) = 0 \tag{3.1.2}$$

   where R is a nonlinear function $R : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$

3. A linear partial differential equation Section 3.6.3, for example, the Heat equation, Eq. (3.6.9);

4. A nonlinear partial differential equation Section 3.6.4, for instance, the Burger's equation, Eq. (4.1.4).

cases in the items 3 and 4 can be represented as an **initial value problem (IVP)**

$$\begin{aligned} \frac{dx}{dt} &= f(x(t), t, \lambda) \\ x(t_0) &= x_0 \end{aligned} \tag{3.1.3}$$

where $t > t_0$ denotes time, $x(t) \in \mathbb{R}^n$ denotes the state vector, and n is the dimension of the space, $f : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^n$ is in general a nonlinear function; $x_0 \in \mathbb{R}^n$ is an initial condition, and λ is the parameter of the equation, for instance, the thermal diffusivity constant in the heat equation.

After discretizing Eq. (3.1.3), it leads at each time step either to a linear system of equations Eq. (3.1.1) or to nonlinear system of equation Eq. (3.1.2):

Usually, the system Eq. (3.1.2) is solved by the Newton-Raphson method and it is called *High-Dimensional Model (HDM)* or as before *Full-Order Model (FOM)*, but it directs to some issues that we enumerate below:

Problem #1. To obtain a good accuracy, the (IVP) Eq. (3.1.3) has to be discretized taking several nodes in its domain, which is translated as a very **large-dimension system** Eq. (3.1.2) having problems related with large dimensions, such that the high amount of computational work that must be done to obtain the solution; leading to rounding errors and consequently bad approximations to the solution, besides that when the system of equations in nonlinear and the Newton method is used, we have to face with the computation and evaluation of partial derivatives.

A way to overcome the above issue is assume that the state solution $x(t)$ belongs to a subspace of $\mathbb{R}^n$ whose dimension is $k \ll n$, such procedure is named *Model-Order Reduction*. Experimentally we showed in Section 3.6 that a very good method to find a basis of such subspace is based on Principal Component Analysis named *Proper Orthogonal Decomposition*, but it tails the

Problem #2. In order to obtain the basis of the subspace where the solution of Eq. (3.1.3) lies, **the FOM has to be solved for different parameter values**, and the solution for each parameter (**snapshot**) is stored in a matrix to procedure later to calculate using **SVD** a basis W corresponding to the subspace sought. Later, when the problem needs to be solved, for a different parameter, W is used in *MOR* instead of having to solve *FOM*.

In Section 4.1 we propose an alternative way to compute the snapshots to avoid to have to solve the *FOM* several times, and to gain in time of computation.

Once the original problem is reduced Eq. (3.1.2) turn in

$$R(Wy, \lambda) = 0. \tag{3.1.4}$$

where, W is the basis of the subspace before mentioned. The system Eq. (3.1.4) is a overdetermined-nonlinear system of equations, which, usually, is solved using the Gauss-Newton method. This method avoids the use of the second order information when the Jacobian is calculated, leading to the following problem:

Problem #3. When we apply the Gauss-Newton method we lost the q-quadratic convergence, which is a characteristic of the Newton-Raphson method; in the best case, only linear convergence is obtained, to improve the rate of convergence, techniques of regularization inspired by Levenberg and Marquardt [26, 40] will be used, and they will be explained in detail in Section 4.2.1 and later we will present results that show an improvement of at least 40% in the number of iterations with respect to FOM, and in the best of the cases we were able to get an improvement of 74% Tab. 4.2.

3.2 Krylov Methods

Given a $n \times n$ matrix A, recall that the characteristic polynomial of A is $p(t) = \det(A - tI)$, where I is the $n \times n$ identity matrix and the minimal polynomial of A is the polynomial $q(t)$ such that:

1. the coefficient at the highest term is 1.
2. $q(A) = 0$, which means, $q(t)$ annihilates A.
3. No polynomial which annihilates A has a smaller degree than $q(t)$.

We can be sure that this polynomial exists because at least $q(t)$ could be $p(t)$, and $p(t)$ annihilates A [29].

Assume $\deg(q(t)) = m$, and $m < n$. We can write

$$q(t) = \sum_{i=0}^{m} \alpha_i t^i. \tag{3.2.1}$$

If A is nonsingular, then $\alpha_0 \neq 0$, and since $q(t)$ annihilates A, we have

$$\begin{aligned}
0 &= \sum_{i=0}^{m} \alpha_i A^i \\
&= \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \cdots + \alpha_m A^m \\
I &= \frac{1}{\alpha_0}\left(-\alpha_1 A - \alpha_2 A^2 - \cdots - \alpha_m A^m\right) \\
A^{-1} &= \frac{1}{\alpha_0}\left(-\alpha_1 I - \alpha_2 A - \cdots - \alpha_m A^{m-1}\right) \\
A^{-1}b &= \frac{1}{\alpha_0}\left(-\alpha_1 b - \alpha_2 Ab - \cdots - \alpha_m A^{m-1}b\right).
\end{aligned}$$

We have proved that the solution of the linear system $Ax = b$ lies in the space spanned by

$$\mathcal{K}(A, b)_m = \{b,\ Ab, A^2b, \ldots, A^{m-1}b\},$$

which is called *Krylov space*, and whose dimension is the degree of the minimal polynomial of A. Therefore, if the minimal polynomial of A has a lower degree than the dimension of subspace where the solution is searched, then is small and in that case the convergence of the method used can be fast.

Another reason to use Krylov methods is that A could be given only implicitly as a subroutine such that, given a vector v, it returns Av.

There exist several versions of Krylov methods, which we can mention: Arnoldi's method, Conjugate Gradient, among others [7, 54]. In particular in this work we use the generalized minimal residual method (GMRES) that was published by Saad and Schultz in 1986 [54].

3.3 Wavelets-based Approach (DWT)

Given a finite wave, Ψ, called "Wavelet Mother", the wavelet transform is a special kind of mathematical transform which analyze a function $f(t) \in L^2(\mathbb{R})$ expressing it in terms of

versions of dilatations and translations of the Wavelet Mother:

$$\Psi_{m,n}(t) = a_0^{-m/2}\Psi(a_0^{-m}(t - nb_0a_0^m)) \tag{3.3.1}$$

If $a_0 = 2$, $b_0 = 1$ and $\Psi(t)$ is defined by:

$$\Psi(t) = \begin{cases} 1 & \text{if} \quad 0 \leq t < \frac{1}{2} \\ -1 & \text{if} \quad \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.3.2}$$

Haar in (1910) [65] proved that Eqs. (3.3.1) and (3.3.2) provide an orthonormal basis of $L^2(\mathbb{R})$.

In the Discrete Wavelet Transform (DWT), the wavelet transform is sampled at discrete mesh points. The discrete version of Ψ, can be represented as a block matrix, W, where the half top block is called the **low-pass** and the bottom block is called the **high-pass**, see [27]:

$$W = \begin{bmatrix} L \\ H \end{bmatrix}. \tag{3.3.3}$$

The matrix W is a wavelet of level 1 and W^k is the level k of the wavelet. As an example, If $n = 8$, the discrete version, W, Eq. (3.3.3) of the wavelet, Ψ in Eq. (3.3.2) is:

$$W = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} \tag{3.3.4}$$

In general the matrix W is an orthonormal matrix, as can be easily seen in the matrix in Eq. (3.3.4) that we used as example, therefore it can prove that:

$$LL^T = I_{\frac{n}{2}\times\frac{n}{2}}, \tag{3.3.5}$$

$$HH^T = I_{\frac{n}{2}\times\frac{n}{2}}. \tag{3.3.6}$$

where n is the dimension of W. Another property of wavelets is that if $x \in \mathbb{R}^n$ is "smooth enough", i.e. $x_{i+1} \approx x_i$, then

$$||x||^2 = ||Wx||^2 \approx ||Lx||^2. \tag{3.3.7}$$

We can observe this property in Fig. 3.2.

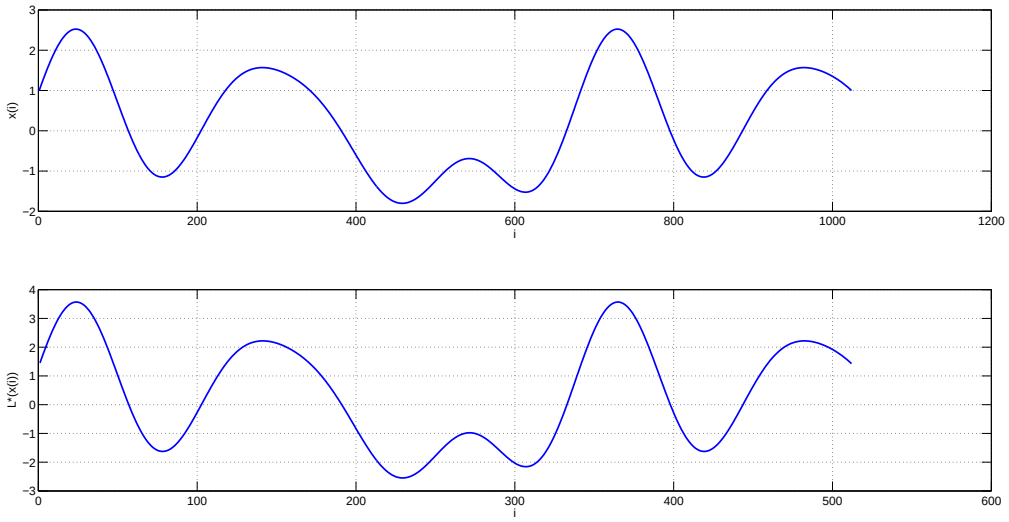


Figure 3.2: Vectors x and Lx. $||x|| = 51.4118$ and $||Lx|| = 51.4125$

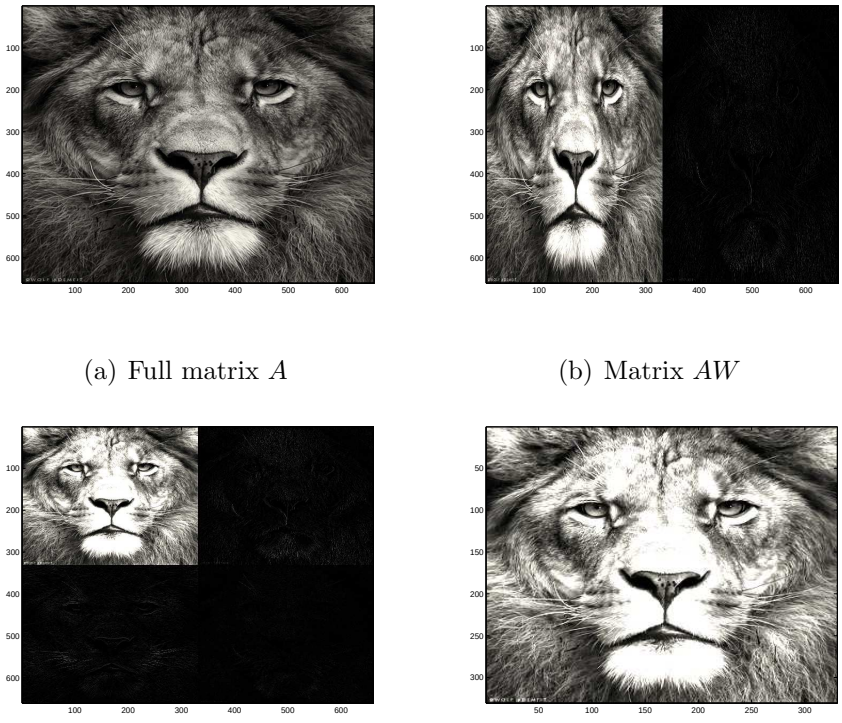
3.3.1 How to use wavelets to reduce a linear system of equations

Given a linear system of equations $Ax = b$, where x is the discretization of a continuous function and this system is the discretization of the solution of an ODE or PDE, we can

take advantage of the property Eq. (3.3.7) to reduce the system of equations by using the low-pass block

$$(LAL^T)\bar{x} = Lb. \tag{3.3.8}$$

Then, once $\bar{x}$ is found, we can obtain x using $x = L^T\bar{x}$. We illustrate Eq. (3.3.8) with the sequences of images shown in Fig. 3.3.



(a) Full matrix A

(b) Matrix AW

(c) Matrix WAW^T

(d) Reduced Matrix LAL^T

Figure 3.3: Illustration of how to use wavelets to reduce a linear system of equations

3.4 Proper Orthogonal Decomposition (POD).

Consider a parameterized static computational model described by large-scale linear system of discrete equations

$$A(\lambda)x = b. \tag{3.4.1}$$

Here we can see Eq. (3.4.1) as an input-output system, where λ is the input and the solution, $x(\lambda) \in \mathbb{R}^n$, is the output.

The idea behind this method is that, given a certain input, the solution $x(\lambda)$ of a system contains the behavior of the system [56]. Therefore, the set of outputs serves as a starting-point for POD. The outputs are called *snapshots* and these must be given or be computed first.

Assume the set of snapshots S and the solution $x(\lambda^*)$ of Eq. (3.4.1) for a particular λ^* is in the subspace spanned by S. We assume that the columns of S are highly correlated, so we can apply principal components analysis (PCA) to obtain an uncorrelated number of columns, see Section 2.2, and thus to reduce the size of linear system of equations.

Consider the SVD of S

$$S = U\Sigma V^T \tag{3.4.2}$$

and

$$T = V\Sigma^{-1}U^T. \tag{3.4.3}$$

Define

$$\begin{aligned} T_1 &= \sum_{i=1}^{k} v_i \sigma_i^{-1} u_i^T; \qquad T_2 = \sum_{i=k+1}^{n} v_i \sigma_i^{-1} u_i^T, \\ S_1 &= \sum_{i=1}^{k} u_i \sigma_i v_i^T; \qquad S_2 = \sum_{i=k+1}^{n} u_i \sigma_i v_i^T. \end{aligned} \tag{3.4.4}$$

Conditions given by Eq. (3.4.4) are a particular case of conditions given in Section 2.3.1. We conclude that we get a good approximation of Eq. (3.4.1) if $S_2\tilde{x}$ is sufficiently small ($\tilde{x} = T_2(x)$) or equivalently if $\sigma_i \approx 0$ for $k+1 \leq i \leq n$.

To obtain a basis of W we have the following algorithm (see algorithm 1):

Several problems have been solved by using this method [69]. As it has been said before, the POD is based on Principal Components Analysis. The reader who wants to read a little more about this can find a good source of information in [42].

Algorithm 1: Computing Proper Orthogonal Decomposition Base.

Input: Parameter λ's and *input-output system*

Output: Base of the subspace W.

1. Solve the full-order model to several λ's.
2. For each λ, take one or more **snapshots** which is the solution of Eq. (3.4.1) for some values of t, and store such snapshots in a matrix S.
3. Compute the SVD of S: $[W, \Sigma, V] = svd(S)$.
4. Find k such that $\sigma = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{n} \sigma_i} > 0.99$.
5. Consider only the k first columns and redefine $W = W(:, [1:k])$.

3.5 Using Model Order Reduction to Solve Nonlinear Problems

Consider a nonlinear system of equations

$$F(x) = 0, \tag{3.5.1}$$

where F is a nonlinear function from $\mathbb{R}^n$ to $\mathbb{R}^n$ which might arise from the discretization of a set of partial differential equations. The Newton Method applied to Eq. (3.5.1) results in the following iteration:

1. Set $x_0 =$ an initial guess.
2. For $n = 0, 1, 2, \ldots$ until convergence do:

$$\begin{aligned} J(x_n)\Delta x_n &= -F(x_n), \\ x_{n+1} &= x_n + \Delta x_n, \end{aligned} \tag{3.5.2}$$

where $J(x_n) = F'(x_n)$ is the Jacobian. For large problems we can use MOR techniques studied in the previous sections to reduce the dimension of Eq. (3.5.2).

3.6 Examples

In this section, four examples are solved using each approach studied in the previous sections. The first example is a set of linear system of equations, the second is a nonlinear system of equation, the third is the heat equation as example of a linear partial differential of equation and the last one is the Burgers' equation as an example of nonlinear partial differential equation.

We compare all the results in terms of computation time and relative error compared with the FOM solution.

3.6.1 Linear System of Equations

Consider a family of 400 matrices $A_{400\times400}(\lambda)$, where $\lambda \in [-1, 1]$. The matrix $A(-1)$ is represented by Fig. 3.4(a) and the matrix $A(1)$ is represented by Fig. 3.4(c). For each $\lambda \in [-1, 1]$ the matrix $A(\lambda)$ is selected as a convex linear combination of $A(-1)$ and $A(1)$. For each λ, let us define the linear system of equations:

$$A(\lambda)x = b(\lambda), \tag{3.6.1}$$

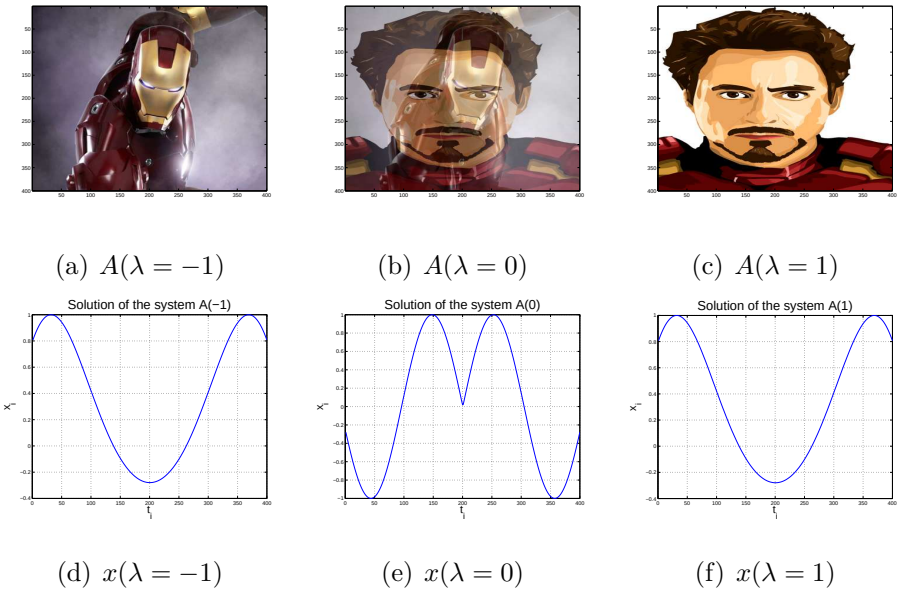
where $b(\lambda)$ is selected such that, $x(\lambda)$ satisfies:

$$x(\lambda) = \sin\left(6\sqrt{t^2 + \lambda^2}\right) \tag{3.6.2}$$

and t is the discretization of the interval $[-1, 1]$ in 400 nodes. More explicit:

$$\begin{aligned} i &= 1, 2, \cdots, 399 \\ t_1 &= -1 \\ t_{i+1} &= t_i + \frac{2}{399} \\ x_i(\lambda) &= \sin\left(6\sqrt{t_i^2 + \lambda^2}\right) \end{aligned}$$

The image representation of the matrix $A(-1)$ Fig. 3.4(a), $A(0)$ Fig. 3.4(b), and $A(1)$ Fig. 3.4(c) with their respective solutions, $x(-1)$ Fig. 3.4(d), Fig. 3.4(e), and Fig. 3.4(f) are presented in Fig. 3.4.



(a) $A(\lambda = -1)$ (b) $A(\lambda = 0)$ (c) $A(\lambda = 1)$

(d) $x(\lambda = -1)$ (e) $x(\lambda = 0)$ (f) $x(\lambda = 1)$

Figure 3.4: Three members of the family of the linear systems are shown.

When we plot in the same graph the solution corresponding to each parameter λ in the system of the Eq. (3.6.1), we obtain Fig. 3.5:

The problem was solved using MatLab, R2012b(8.0.0.783) in a MacBook Air, processor 1.7 GHz Intel Core i7, 8 GB 1600 MHz DDR3 and we get the following results:

1. **Full Order Model (FOM):** got the solution in 2.83 secs. with a relative error $\varepsilon = 4.1236 \times 10^{-09}$.

2. **Krylov Method:** a modification of this method named `gmres` (Generalized Minimum Residual Method) is used to obtain the solution of these non-symmetric systems of linear equations. The approximation of the solution is obtained in 2.40 secs with a relative error of 0.42. In Fig. 3.6(b) the approximation of the solution is presented. Observe that for this example Krylov method is not a good choice.

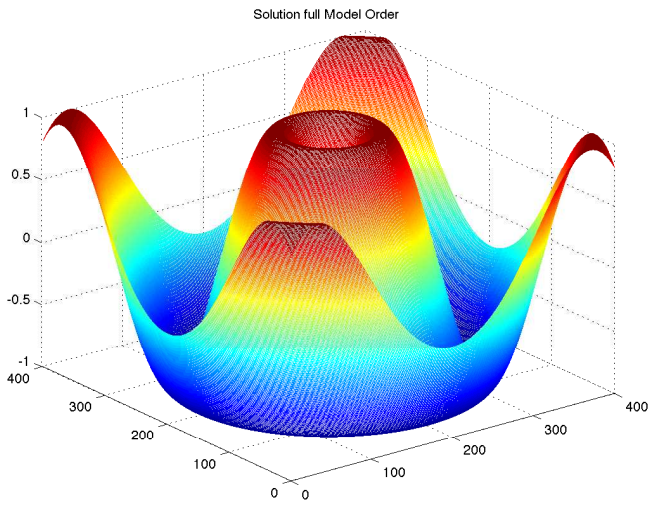


Figure 3.5: Solution of the $A(\lambda)x = b(\lambda)$, for all $\lambda \in [-1, 1]$ is depicted.

3. **Wavelet Method:** the second level of the low pass of the wavelet named `db1` (Daubechies Wavelet) is used as the reduced basis in this method. With this, the solution is obtained in 2.49 secs with a relative error of 0.0257. The graph of the solution using this method is given in Fig. 3.6(c).

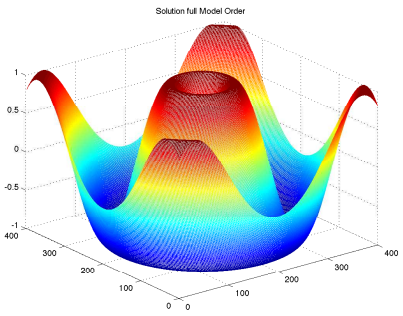
4. **Proper Orthogonal Decomposition (POD) Method:** the singular value decomposition, **SVD** is used to obtain a factorization of the full order model **FOM**,

$$FOM = USV^T.$$

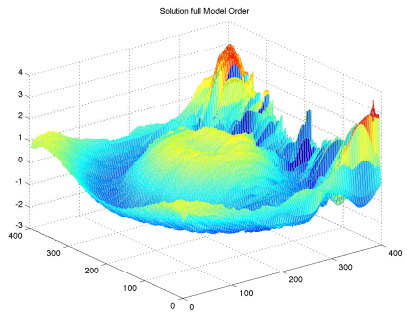
solution. Later, the first five columns of U, which represent the 99.85% of the total variance, are taken as the reduced basis. With this method the solution can be obtained in 0.37 secs and the relative error is 0.0021. This solution is shown in Fig. 3.6(d).

The graphs of the corresponding approximations are presented in Fig. 3.6.

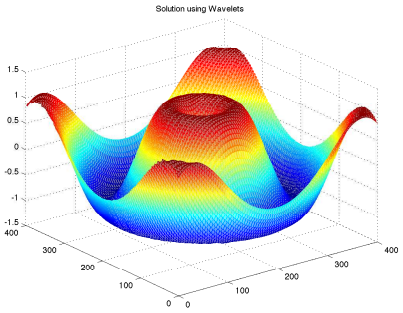
Summarizing:



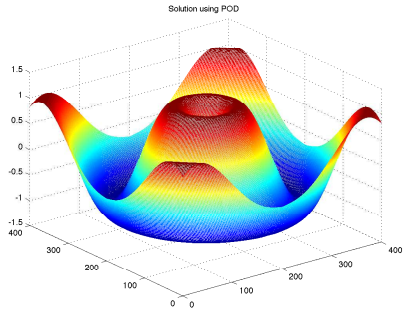
(a) FOM Solution



(b) ROM Solution Using Krylov Method



(c) ROM Solution Using Wavelets



(d) ROM Solution Using POD

Figure 3.6: Solution of the family of the linear system of equations using different methods.

	FOM	ROM Krylov	ROM Wavelet	ROM POD
Time	1.83 *secs.*	240 *secs.*	2.49 *secs*	0.37 *secs.*
Rel. Error	4.1236×10^{-9}	0.42	0.0257	0.0021
Dimension	400	100	100	5

This section concludes mentioning that the best result with respect to time and relative error was ROM based in POD.

3.6.2 Nonlinear System of Equations

The nonlinear system of equations proposed by Yamamura and Kawata [71], defined by:

$$\begin{aligned} &F : \mathbb{R}^n \to \mathbb{R}^n,\ x \mapsto (F_i(x))_{1\leq i\leq n},\ \text{where} \\ &F_i = 2.5x_i^3 - 10.5x_i^2 + 11.8x_i - i + \sum_{i=1}^{i=n} x_i = 0, \quad 0 \leq i \leq n. \end{aligned} \tag{3.6.3}$$

is solved. The Jacobian of F is a dense matrix, J, defined by

$$J_{ij} = \frac{\partial F_i(x)}{\partial x_j}.$$

A frequently used method to solve a system of this kind is the Newton Method, which is illustrated in Section 3.5.

We employ the techniques of MOR studied before to solve Eq. (3.6.3) and to determinate which of them gives the best result.

1. **(FOM):** In first place, each linear system of each iteration is solved using **FOM**. The initial point x_0 is taken in the box $\underbrace{[-10^8, 10^8] \times [-10^8, 10^8] \times \cdots \times [-10^8, 10^8]}_{n \text{ times}}$ and the stopping criterion is $\varepsilon = 10^{-5}$. The method converges in 459.74 secs, and takes 814 iterations; $||F(x)|| < 2.81 \times 10^{-10}$ in the solution.

2. **Krylov Method:** In this part the linear system of equations in Eq. (3.5.2) is solved using a modification of Krylov method named `minres`, which is based on Lanczos tridiagonalization. The implementation used in this thesis was developed by **Systems Optimization Laboratory, Stanford University Dept. of Management Science and Engineering (MS&E)** [45]. The matrix A must be symmetric but it may be definite or indefinite. The Jacobian $J(x)$ of the Yamamura problem is symmetric for all x. Under the same previous conditions the solution is obtained in 108 iterations in 50.64 secs with $||F(x)|| < 10^{-6}$ in the solution, and the relative error compared with the solution obtained with **FOM** is $\varepsilon_r = 0.0117$. `minres` is used

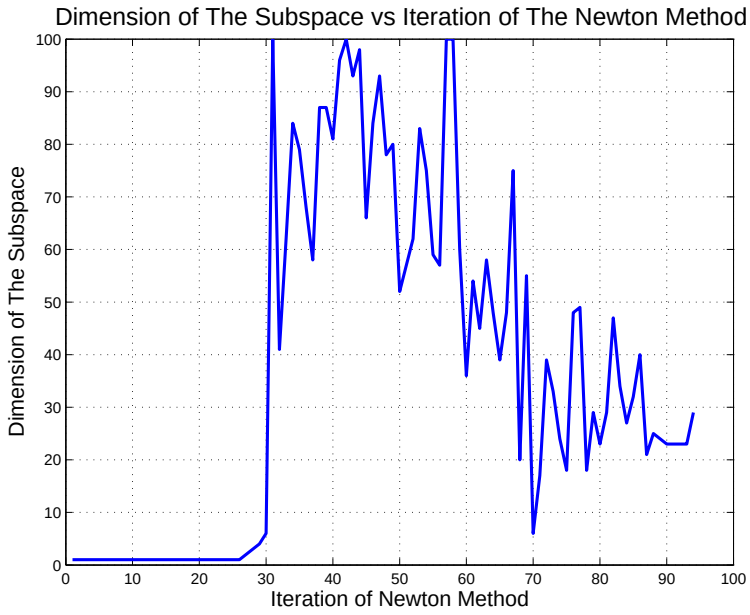


Figure 3.7: Dimension of the subspace where the solution of the linear system Eq. (3.5.2) lies in every iteration of the Newton Method.

to solve Eq. (3.5.2), to find a subspace where the Δx is located. Fig. 3.7 shows the dimension for every iteration, we can observe that the maximum dimension of such subspaces is 100 and the minimum is less than 5.

3. **Wavelet Method:** Unlike the previous method `minres`, which finds a subspace for the Newton step on each iteration of Newton Method, when Wavelet method and POD method are used, it is assumed that the solution of the problem is either in a subspace W or close to it (in terms of an orthogonal projection) and that the solution can be described as a linear combination of the elements of a basis of such subspace, U. In such a case, the nonlinear system of equations can be solved:

$$Up = x \tag{3.6.4}$$

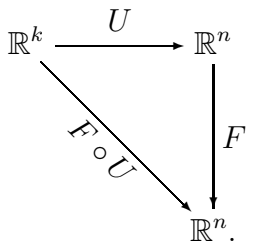
$$F(x) = 0 \tag{3.6.5}$$

Solving the system (3.6.4), (3.6.5) is equivalent to solving the overdetermined non-

linear system

$$\tilde{F}(p) = F(Up) = 0. \tag{3.6.6}$$

The composition $F(Up)$ is illustrated in the following commutative diagram:



the chain rule is used to find the Jacobian of $\tilde{F}$,

$$\tilde{J}(p) = J(Up)U. \tag{3.6.7}$$

With this modification the Newton Method becomes the **Reduced Newton Method**:

(a) Choose a initial point $p \in \mathbb{R}^k$

(b) for $i = 1$ until convergence

(c) Solve

$$J(Up)\Delta p = -F(Up). \tag{3.6.8}$$

(d) $p = p + \Delta p$

(e) end for

In this part, the second level low pass of the discrete wavelets `db1` is considered as a basis of the subspace where the solution of the Yamamura problem is supposed to be. The Reduced Newton Method is applied like previous methods. The stopping criterium used is $\varepsilon = 10^{-5}$ and the initial point is a uniform random vector p_0 in the box $[-10^8, 10^8]^{250}$. When the Reduced Newton Method is applied, the approximation of the solution is obtained in 71 secs taking 155 iterations, $||F(x)|| = 35$ and the relative error compared with the solution obtained with **FOM** was $\varepsilon_r = 0.021$.

4. **Proper Orthogonal Decomposition (POD):** In order to compute the snapshots and to find the reduced basis, a small modification of the Yamamura problem was done:

$$F : \mathbb{R}^n \to \mathbb{R}^n,\ X \mapsto (F_i(X;\lambda))_{1\leq i\leq n}$$

$$F_i = \lambda x_i^3 - 10.5x_i^2 + 11.8x_i - i + \sum_{i=1}^{i=n} x_i = 0, \quad 0 \leq i \leq n.$$

The following problem:

$$F(X,\lambda) = 0,\ \text{for } \lambda = 2.001, 2.002, \ldots, 2.999, 3.000.$$

is solved. For each λ_i a matrix of the snapshots where every column is the solution X_i, is defined

$$\text{Sna} = [X_1\ X_2\ \ldots\ X_{999}\ X_{1000}].$$

Later the singular value decomposition of Sna is computed

$$\text{Sna} = USV^T,$$

and finally, the first k columns of U are used as basis, such that

$$\frac{\sum_{i=1}^{k}\sigma_i}{\sum_{i=1}^{1000}\sigma_i} > 0.997.$$

In this experiment, $k = 100$.

POD Method obtains the solution of the problem in 46 secs with 110 iterations, $||F(X)|| < 1.12 \times 10^{-5}$ and the relative error compared with the solution obtained with the **FOM** method $\varepsilon = 4.25 \times 10^{-10}$.

Summarizing:

	FOM	ROM Krylov	ROM Wavelet	ROM POD
Time	459.74 *sec*	50.64 *sec*	71.27 *sec*	46 *sec*
Iterations	814	108	155	110
Rel. Error	–	0.0117	0.021	4.25×10^{-10}
Dimension	1000	[5, 100]	250	100

3.6.3 Linear Partial Differential equation

In this section Heat equation is solved using the four studied approach. The heat equation is a linear partial differential equation which models the flow of heat in a rod that is insulated everywhere except at the two ends, and it is defined by

$$\frac{\partial U}{\partial t} = \lambda \frac{\partial^2 U}{\partial x^2}, \tag{3.6.9}$$

where:

- The variable $x \in [0, L]$, where L is the length of the rod,
- $t \geq 0$ is the time variable
- $U(0, x) = f(x), \quad$ where $x \in [0, L]$, (Initial Condition),
- $U(t, 0) = 0 = U(t, L), \quad$ where $t > 0$, (Boundary condition).

In our experiment, $L = 1$, $f(x) = \sin(\pi x) + \sin(2\pi x)$ and $t \in [0, 0.01]$.

1. **Full Order Model:** The following equations:

$$\frac{\partial U}{\partial t} = \frac{U(t_i, x_j) - U(t_{i-1}, x_j)}{\Delta t}, \tag{3.6.10}$$

$$\frac{\partial^2 U}{\partial x^2} = \frac{U(t_i, x_{j-1}) - 2U(t_i, x_j) + U(t_i, x_{j+1})}{h^2}, \tag{3.6.11}$$

are used to discretize the heat equation in each node of space-time domain, where $h = x_{i+1} - x_i$ and $\Delta t = t_{i+1} - t_i$, and we can simplify $U(t_i, x_j) = U_{i,j}$. Substituting 3.6.10 and 3.6.11 in 3.6.9 with $\lambda = 1$, the **sparse and symmetric linear system** defined by

$$\frac{U_{i,j} - U_{i-1,j}}{\Delta t} = \lambda \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{h^2}.$$

is obtained and by collecting similar terms, the system is written as

$$-\lambda \Delta t U_{i,j-1} + (2\lambda \Delta t + h^2) U_{i,j} - \lambda \Delta t U_{i,j+1} = h^2 U_{i-1,j}. \tag{3.6.12}$$

The linear system of equations Eq. (3.6.12) is solved for each $1 \leq i \leq 400$ using **FOM** and the approximation of the solution is given in 0.572 secs.

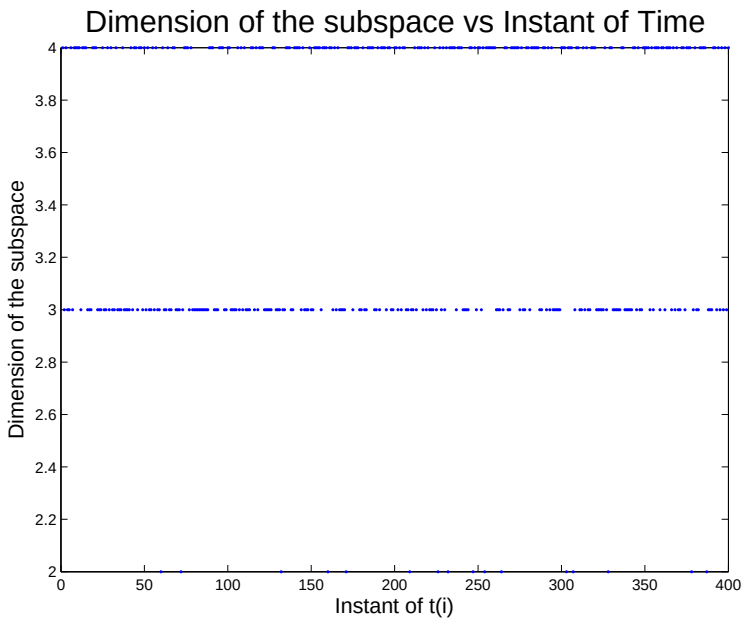


Figure 3.8: Dimension of the subspace where the solution of the linear system Eq. (3.6.12) lies in every instant of t.

2. **Krylov Method:** Due to the fact the matrix in the heat equation discretization is symmetric and sparse, `minres` is used to solve the linear systems. The program `minres` was able to present an approximation of the solution in 0.2 secs. For each instant t the method `minres` was able to find a subspace where the solution of the linear system was and the dimension of such subspace always was less than 4. In Fig. 3.8 the dimension of such subspaces for each instant of time can be observed. The relative error compared with the solution obtained with **FOM** is $\varepsilon_r = 0.021$.

3. **Wavelet Method:** The first level of a `db1`, which has dimension 200, is used as reduced basis. This method did not obtain a good approximation of the solution of the heat equation. The time needed to get the solution was 1.97 secs, with a relative error compared with the **FOM** solution of $\varepsilon_r = 0.68$.

4. **Proper Orthogonal Decomposition** The parameter λ in Eq. (3.6.9) is used as in-

put. The equation is solved for each $\lambda \in [0.5, 1.5]$ with $\Delta\lambda = 0.01$. For each parameter 50 snapshots are taken and the matrix Snap is defined with all the snapshots collected. The singular value decomposition of the matrix of the snapshots, Snap $= USV^T$ is computed and only two columns of U are enough to obtain a good approximation.

$$\frac{\sum_{i=1}^{2} \sigma_i}{\sum_{i=1}^{400} \sigma_i} \approx 0.9999.$$

Using as reduced basis the two first columns of U the approximation of solution is obtained in 0.059 secs and the relative error $\varepsilon_r = 0.059$.

Summarizing:

	FOM	ROM krylov	ROM Wavelet	ROM POD
Time	0.572 *secs*	0.2 *secs*	1.97 *secs*	0.059 *secs*
Rel. Error	–	0.021	0.68	0.059
Dimension	400	[2, 4]	200	2

The graphs of the solution by using different methods are shown in Fig. 3.9.

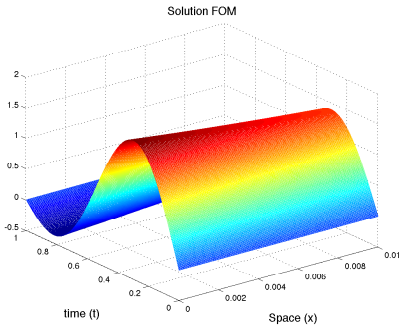
3.6.4 Nonlinear Partial Differential Equation

In this section the FOM, Krylov, Wavelets and POD method are used to solve the Burgers equation as an example of nonlinear partial differential equation.

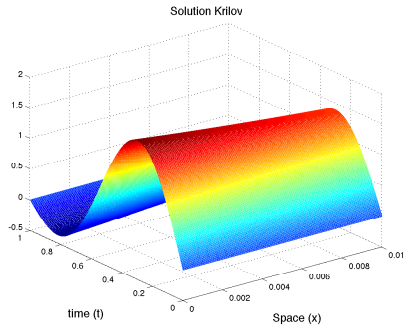
Burgers equation is a fundamental partial differential equation from fluid mechanics. It occurs in various areas of applied mathematics, such as modeling of gas dynamics and traffic flow, and it is defined by

$$\frac{\partial U(t,x)}{\partial t} + \frac{\partial f(U(t,x))}{\partial x} = g(x), \tag{3.6.13}$$

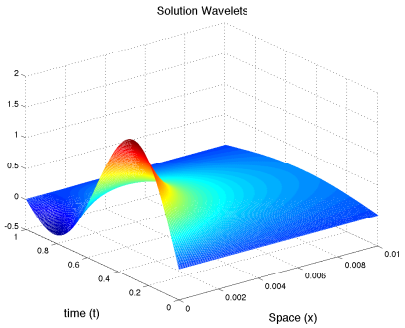
where U is the unknown conserved quantity (mass, density, heat etc.), $f(U) = 0.5U^2$ and in our example, $g(x) = 0.02\exp(0.02x)$. The initial and boundary conditions used with the above PDE are: $U(0,x) \equiv 1$; $U(t,0) = u(t)$, for all $x \in [0; 100]$, and $t > 0$.



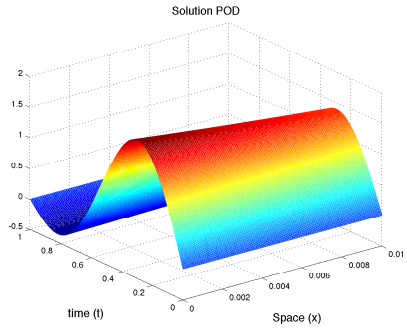
(a) FOM Solution



(b) ROM Solution Using Krylov Method



(c) ROM Solution Using Wavelets



(d) ROM Solution Using POD

Figure 3.9: Solution of the Heat Equation.

Discretizing $U(t,x)$ with respect to the variable x yields

$$\underline{U}(t) = [U_1(t), U_2(t), \cdots, U_n(t)] \tag{3.6.14}$$

where $U_i(t) = U(t, x_i)$, and $x_i = x_{i-1} + \Delta x$. Applying Godunov's scheme, also known as finite-volumen method, to approximate $\partial/\partial x$ [51, 68], the dynamical system takes the following form:

$$\frac{d\underline{U}}{dt} = F(\underline{U}) + G + Bu^2. \tag{3.6.15}$$

where:

$$F(\underline{U}) = \begin{bmatrix} -0.5U_1^2 \\ 0.5(U_1^2 - U_2^2) \\ \vdots \\ 0.5(U_{n-1}^2 - U_n^2) \end{bmatrix}, \quad G = 0.02 \begin{bmatrix} \exp(0.02x_1) \\ \exp(0.02x_2) \\ \vdots \\ \exp(0.02x_n) \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{2\Delta x} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.6.16}$$

Now, discretizing Eq. (3.6.15) with respect to the variable t:

$$\frac{\underline{U}(t_i) - \underline{U}(t_{i-1})}{\Delta t} = F(\underline{U}) + G + Bu^2, \tag{3.6.17}$$

where $\underline{U}(t_0) = U^0$, is the initial condition, fixing the value of t_i, Eq. (3.6.17) is equivalent to solve the non-linear system

$$R(\underline{U}) = \begin{pmatrix} (-0.5U_1(t_i)^2 + 0.02e^{0.02} + \frac{16}{2})\Delta t + U_1(t_i) - U_1(t_{i-1}) \\ (0.5(U_1(t_i)^2 - U_2(t_i)^2) + 0.02e^{0.02\cdot 2})\Delta t + U_2(t_i) - U_2(t_{i-1}) \\ \vdots \\ (0.5(U_{i-1}(t_i^2 - U_i(t_i^2) + 0.02e^{0.02\cdot i})\Delta t + U_i(t_i) - U_i(t_{i-1}) \\ \vdots \\ (0.5(U_{99}(t_i^2 - U_{100}(t_i^2) + 0.02e^{0.02\cdot 100})\Delta t + U_{100}(t_i) - U_{100}(t_{i-1}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.6.18}$$

and Eq. (3.6.18) can be solved using techniques to solve nonlinear system of equations.

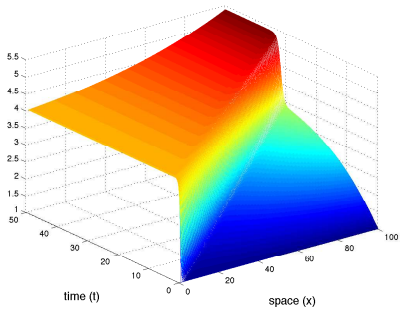
The Jacobian of R is a bidiagonal matrix with linear terms,

$$J = \begin{pmatrix} -U_1(t_i-)1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ U_1(t_i) & -U_2(t_i) - 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & U_2(t_i) & -U_3(t_i) - 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & U_3(t_i) & -U_4(t_i) - 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & U_{99}(t_i) & -U_{100}(t_i) - 1 \end{pmatrix}. \tag{3.6.19}$$

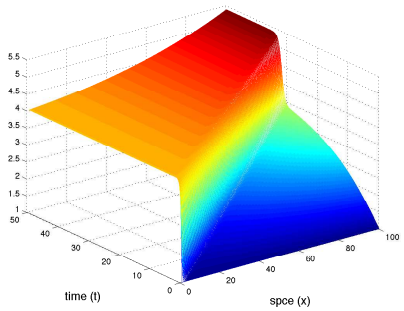
After we discretize and solve the problem using Newton Method for each time step and each approach, the results are presented in Tab. 3.1 and the graphs of the solution are presented in Fig. 3.10.

	FOM	ROM krylov	ROM Wavelet	ROM POD
Time	0.41 *sec.*	2.76 *sec.*	0.77 *sec.*	0.67 *sec.*
Rel. Error	–	3.85E-11	0.0076	9.013E-4
Dimension	100	7	50	32

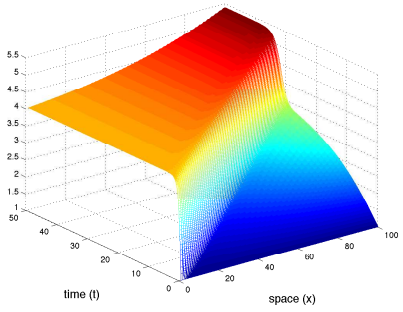
Table 3.1: Comparison of different methods solving the Burgers Equation



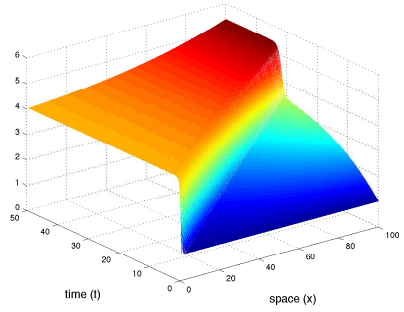
(a) FOM Solution



(b) ROM Solution Using Krylov Method



(c) ROM Solution Using Wavelets



(d) ROM Solution Using POD

Figure 3.10: Solution of the Burger Equation.

Chapter 4

Contributions

The idea is to try to give all the information to help others to judge the value of your contribution; not just the information that leads to judgment in one particular direction or another.

Richard P. Feynman. (New York, May 11 1918 - California, February 15 1988)

In this chapter we will describe the problem statement and the contributions we have achieved in order to solve it, but before to do that let us summarize what we have done so far.

First, we studied three different existing methods that already exist to reduce the dimension of problems involving linear system of equations. One of them is based on the fact that the solution of every system of equations lies in the subspace spanned by the characteristic polynomial (Krylov method); the second method is able to find the solution of a system of equations in a subspace spanned by the low-pass filter of a particular wavelet (Wavelet method), and the last one uses the fact that every linear system of equations, which is member of a family of system of equations preserves some information of the rest of the other members (POD).

Second, we used all three studied methods to solve problems besides the linear system of equations, for instance, when we need to solve a nonlinear system of equations, a way to

do it is applying the Newton method, and when we have to compute the Newton step, it is posted as a linear system of equations, then we can used any method before mentioned.

Finally we also used these different studied methods to solve either linear or nonlinear partial differential equations, since when we discretize the domain we either obtain a linear system of equations or a nonlinear system of equations and we proceed as before.

Now, let us proceed to described the problem statement and what we have done to solve it.

4.1 Alternative way to obtain the snapshots in the POD method

Given a parametric system of equations

$$R(x, \lambda) = 0, \quad \lambda \in \mathbf{I} \tag{4.1.1}$$

where R can be either linear or nonlinear function from $\mathbb{R}^n$ to $\mathbb{R}^n$ that might arise from the discretization of a set of partial differential equations and $\mathbf{I}$ is a fixed interval. The idea behind POD is to solve Eq. (4.1.1) for a sequence of values $\lambda_i \in \mathbf{I}$, and to get the solution in each case, i.e.

$$\begin{aligned} R(x, \lambda_1) &= 0, \\ R(x, \lambda_2) &= 0, \\ \vdots &\quad \vdots \\ R(x, \lambda_n) &= 0, \end{aligned} \tag{4.1.2}$$

where $\lambda_i \in \mathbf{I}$, for $i = 1, 2, \ldots, n$. The main idea of this method is based on the high correlation between solutions for such values λ_i, so PCA techniques can be applied to obtain a smaller number of columns uncorrelated with the greatest of accumulated variance Section 2.2.

In this work it is proposed to use constraint interval methods to be able to handle uncertainty and to encapsulate the solutions given for the parameters,

$$R(x, \mathbf{I}) = 0. \tag{4.1.3}$$

This modification of POD method is named **Interval Proper Orthogonal Decomposition (IPOD)** method.

4.1.1 Comparing POD and IPOD in solving the Burgers Equation.

Consider the Burgers equation

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial f(U(x,t))}{\partial x} = g(x), \tag{4.1.4}$$

where U is the unknown conserved quantity (mass, density, heat etc.), $f(U) = 0.5U^2$ and in this example, $g(x) = 0.02\exp(0.02x)$. The initial and boundary conditions used with the above PDE are: $U(x;0) \equiv 1$; $U(0;t) = 4$, for all $x \in [0;100]$, and $t > 0$.

The procedure to obtain the snapshots and the reduced basis in the POD method will be described in detail. Later, it will be compared with **IPOD method**.

POD Method algorithm:

1. Initialize a empty matrix where to collect the snapshots: Snap, $\lambda_1 = 3.5$.

2. For $i = 2, \ldots, 100$,

3. $\lambda_i = \lambda_{i-1} + 0.01$. Solve:

$$\begin{cases} \dfrac{\partial U(x,t)}{\partial t} + \dfrac{\partial f(U(x,t))}{\partial x} = g(x), \\ g(x) = 0.02\exp(0.02x) \\ U(x;0) \equiv 1, \text{ for all } x \in [0;100], \\ U(0;t) = \lambda_i, \text{ for } t > 0. \end{cases}$$

4. Collect snapshots: from $t_1, t_2, \ldots, t_n$, select a subsequence[1] $t_{i1}, t_{i2}, \ldots, t_{ip}$. Add new columns to the snapshot matrix Snap = [Snap $U(x, t_{i1})$ $U(x, t_{i2})$ $\ldots U(x, t_{ip})$]..

5. End For i.

6. Apply the principal component analysis, (SVD).

$$\text{Snap} = W\Sigma V^T.$$

7. Select from U the principal components with the greatest accumulated variance:

   (a) $\sigma = 0$.

   (b) for $k = 1 : n$ compute:

$$\sigma = \sigma + \frac{\sigma_k}{\sum_{j=1}^{n} \sigma_j}$$

   (c) If $\sigma > Tol$, $0 < Tol < 1$, break.

8. Select the first k columns of W and redefine it. $W = W(:, [1, 2, \ldots, k])$. The new W will be the reduced basis to apply the POD method.

The POD algorithm is modified in the following way:

IPOD Method algorithm:

1. Initialize a empty matrix where to collect the snapshots: Snap = [], $\lambda_1 = 3.5$.

2. Solve:

$$\begin{cases} \dfrac{\partial U(x,t)}{\partial t} + \dfrac{\partial f(U(x,t))}{\partial x} = g(x), \\ g(x) = 0.02\exp(0.02x) \\ U(x;0) \equiv 1, \text{ for all } x \in [0;100], \\ U(0;t) = \mathbf{I}, \text{ for } t > 0. \end{cases} \tag{4.1.5}$$

   The solution of Eq. (4.1.5) is an interval solution, i.e, for any $1 \leq x_0 \leq 100, 0 \leq t_0 \leq 50$, the value $U(x_0, t_0)$ is an interval. The infimum of such interval is defined $U_l(x_0, t_0)$

[1] In the experiment done in this work, $p = 5$ and the subsequence was a uniform random selection.

and $U_r(x_0, t_0)$ is the supremum. In that case, for all $1 \leq x \leq 100$, $0 \leq t \leq 50$, $U(x,t) \in [U_l(x,t), U_r(x,t)]$, see Figure Fig. 4.1.

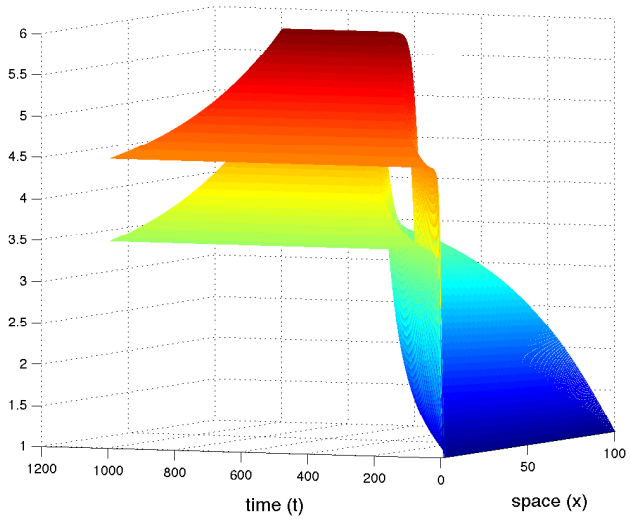


Figure 4.1: Infimum and supremum of the solution of Eq. (4.1.5) are shown.

3. Define: $\lambda = 0$.

4. For $i = 1, 2, \cdots, 100$, Compute:

$$U(x,t) = (1-\lambda)U_l(x,t) + \lambda U_r(x,t)$$

5. Collect snapshots: from $t_1, t_2, \cdots, t_n$, select a subsequence $t_{i1}, t_{i2}, \ldots, t_{ip}$. Add new columns to the snapshot matrix Snap = [Snap $U(x,t_{i1})$ $U(x,t_{i2})$ $\ldots U(x,t_{ip})$].

6. Update λ. $\lambda = \lambda + 0.001$.

7. End For i.

8. Apply the principal component analysis, (SVD).

$$\text{Snap} = W\Sigma V^T.$$

9. Select from U the principal components with the greatest accumulated variance:

   (a) $\sigma = 0$.

   (b) for $k = 1 : n$ compute:

   $$\sigma = \sigma + \frac{\sigma_k}{\sum_{j=1}^{n} \sigma_j}$$

   (c) If $\sigma > Tol$, $0 < Tol < 1$, break.

10. Select the first k columns of W and redefine it. $W = W(:, [1, 2, \ldots, k])$. The new W will be the reduced basis to apply the POD method.

By Applying both previous procedures to solve Eq. (4.1.4) we obtained the results given in the table Tab. 4.1:

Method	Tag 1	Tag 2	Tag 3	Tag 4
POD	300 secs	37	0.75 secs	$4.85E-4$
IPOD	94.45	36	0.75 secs	$5.76E-4$

Tag 1: Time computing the Reduced basis

Tag 2: Dimension of the Subspace

Tag 3: Time solving Eq. (4.1.4) using the obtained basis

Tag 4: $||u_{fom} - u_{rom}||/||u_{fom}||$

Table 4.1: Comparing POD and IPOD methods in solving a particular example of Burgers Equation

4.2 Using Regularization to Improve the Rate of Convergence in a MOR Problem

It has been shown that a way to solve systems of differential equations is by discretizing them, which often leads to a large system of (possibly nonlinear) equations. Even when the

systems are linear, direct computations only allows to solve them in n^3 operations, which may still impose practical limits for large values of n.

When the solution of such systems is searched in subspaces of the original domain and MOR is used, over-determined systems have to be handled. These can be solved using the Gauss-Newton method, which in non-zero-residual cases is not as efficient as needed. Techniques of regularization inspired by Levenberg and Marquardt are proposed to be used to improve the convergence rate of such systems.

4.2.1 Levenberg-Marquardt Method for Nonlinear Equations

Given the function

$$f(x) = \frac{1}{2} R(x)^T R(x) = \frac{1}{2} \sum_{i=1}^{m} r_i(x)^2 \tag{4.2.1}$$

where

$$R : \mathbb{R}^n \to \mathbb{R}^m$$

is a nonlinear function with $m > n$ and $r_i(x)$ is the i-th component of $R(x)$. The nonlinear least-squares problem is given by

$$\min_{x \in \mathbb{R}^n} f(x) \tag{4.2.2}$$

The gradient of $f(x)$ is:

$$\nabla f(x) = J(x)^T R(x)$$

where $J(x)$ is the jacobian of R, and the Hessian is given by

$$\begin{aligned} \nabla^2 f(x) &= \sum_{i=1}^{m} (\nabla r_i(x) \nabla r_i(x)^T + r_i(x) \nabla^2 r_i(x)) && (4.2.3) \\ &= J(x)^T J(x) + S(x), && (4.2.4) \end{aligned}$$

where

$$S(x) = \sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x)$$

denotes the second order information of the Hessian $\nabla^2 f(x)$.

If the Newton Method is applied to solve Eq. (4.2.2) the linear system

$$(J(x)^T J(x) + S(x))\Delta x = -J(x)^T R(x) \tag{4.2.5}$$

has to be solved to calculate the Newton step Δx and update $x_k = x_{k-1} + \Delta x$.

The Gauss-Newton method is a secant approximation of Newton Method. Instead of solve Eq. (4.2.5), the system

$$(J(x)^T J(x))\Delta x = -J(x)^T R(x) \tag{4.2.6}$$

is solved.

In [18] is proved that if there exist a x_* with $J(x_*)^T R(x_*) = 0$ and $J(x)$ is Lipschitz continuous, the sequence generated by the Gauss-Newton method

$$x_{k+1} = x_k - (J(x_k)^T J(x_k))^{-1} J(x_k)^T R(x_k)$$

is well defined and converges to x_* with a **linear convergence**.

In [18] is proved as well that if $R(x_*) = 0$, then the sequence generated by the The Gauss-Newton method is well defined and converges q-quadratically to x_*.

Gauss-Newton method has an oscillatory behavior during iterations and is not as robust as steepest descent. Levenberg and Marquardt [26] introduced a factor λ to switch between the steepest descent direction and the Gauss-Newton direction. The updated equation is

$$(J(x)^T J(x) + \lambda I)\Delta x = -J(x)^T R(x). \tag{4.2.7}$$

When $\lambda \to 0$, LM method is reduced to Gauss-Newton. When $\lambda \to \infty$, LM approaches to steepest descent method.

4.2.2 Use of Levenberg-Marquardt method in solving MOR problems

Consider the system of nonlinear equations

$$F(x) = 0, \tag{4.2.8}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable.

In order to apply any MOR method it is needed to find a basis W of a subspace and express the solution as a linear combination of the elements of such basis, i.e,

$$Wy = x, \tag{4.2.9}$$

Replacing Eq. (4.2.9) in Eq. (4.2.8) we obtain a least-squares problem

$$F(Wy) = 0. \tag{4.2.10}$$

There are two cases to study, either Eq. (4.2.9) has a solution, which means that (Eq. (4.2.10)) is a zero-residual problem and Gauss-Newton converges q-quadratic to the solution, or otherwise Gauss-Newton has a linear convergence to the "best solution", in this case, the Levenberg-Marquardt method Eq. (4.2.7) is applied to improve the rate of convergence.

Example 1. *Solve the Yamamura problem Eq.* (3.6.3)*, using MOR Wavelet method and compare with Levenberg-Marquardt method.*

The problem is solved for different dimension n with the following criteria:

1. Initial condition: For FOM we used $x^0 \in \mathbb{R}^n$, with coordinates $x_i^0 \in [-10^8, 10^8]$ a uniform random number and for both of MOR methods $p^0 \in \mathbb{R}^{n/8}$, with $p_i^0 \in [-10^8, 10^8]$, a uniform random number.

2. Dimension of the subspace: $n/8$, i.e. the third level low-pass block of the `db1` wavelets.

3. Stopping criteria: In the case of FOM, $||F(x)|| < 10^{-6}$, and in the case of ROM methods, we use two stopping criteria: $||F(Wp)|| < 10^{-6}$ or $||x_{k+1} - x_k|| < 10^{-5}$.

The results obtained are presented in Tab. 4.2.

	Itera			Time		
n	fom	rom	reg	fom	rom	reg
200	126	74	74	1.16	0.64	0.63
300	156	1000	106	3.18	19.92	2.03
400	243	108	65	10.20	3.86	2.50
500	125	1000	131	8.67	64.49	7.90
600	124	106	77	13.12	10.08	7.51
700	242	163	90	38.24	23.15	12.92
800	299	98	108	73.96	22.91	24.47
900	125	155	89	48.71	57.05	29.64
1000	156	179	151	88.22	98.04	82.14

Table 4.2: Comparison of FOM, MOR Wavelets and Regularized MOR Wavelets

4.2.3 Interval Newton Method

Let's consider the equation

$$f(x) = 0. \tag{4.2.11}$$

Let $f'(X)$ be an extension of $f'(x)$, and let $[a, b]$ be an interval in which a solution of the equation is sought. Consider the algorithm

$$X^{(k+1)} = X^{(k)} \cap N(X^{(k)}) \quad (k = 0, 1, 2, \dots), \tag{4.2.12}$$

where $N(X) = m(X) - f(m(X))/F'(X)$ and $X^{(0)} = [a, b]$.

In [43] it is proven that if $0 \notin F'(x)$, then the interval sequence $\{X^{(k)}\}$ is nested and converges to the solution. If $0 \in F'(X)$ the intersection $X^{(0)} \cap N(X^{(0)})$ is the union of two disjoint intervals, say X_1 and X_2. It can momentarily put aside X_1, for instance, and work with X_2. Later, in order to obtain convergence in that interval, X_1 can be used to find a different solution.

Example 2. *Use the interval Newton method with extended arithmetic to find all zeros of the polynomial* $f(x) = x^3 - 5x - 1$. *Compare with traditional Newton Method.*

The solutions are presented in the following table:

	Interval Newton Method		Newton Method	
root	Value	Time	Value	Time
x_1	-2.1284	0.0070	-2.1284	0.000724
x_2	-0.2016	0.0075		
x_3	2.3301	0.0068		

It can be said that this is the **global version of the Newton method**.

In the following example RealPaver [23] is used to obtain all the solutions of a nonlinear system of equations and later they are compared with the solution obtained with the Newton Method.

Example 3. *Consider the Yamamura problem Eq.* (3.6.3) *for* $n = 4$ *and an initial box:* $[-10^8\ ,\ 10^8] \times [-10^8\ ,\ 10^8] \times [-10^8\ ,\ 10^8] \times [-10^8\ ,\ 10^8]$.

With the Newton method only one solution is obtained and using RealPaver three solutions were obtained, one of which is Newton Method's solution. In Fig. 4.2 we can see the solutions obtained.

4.3 Constraint Solving Techniques to Solve Dynamical System

Constraint solving techniques can be used to take uncertainty into account. this fact will be illustrated with several examples.

Logistic equation:

Let's begin solving the logistic equation:

$$\frac{dy}{dt} = \theta y(M - y), \tag{4.3.1}$$

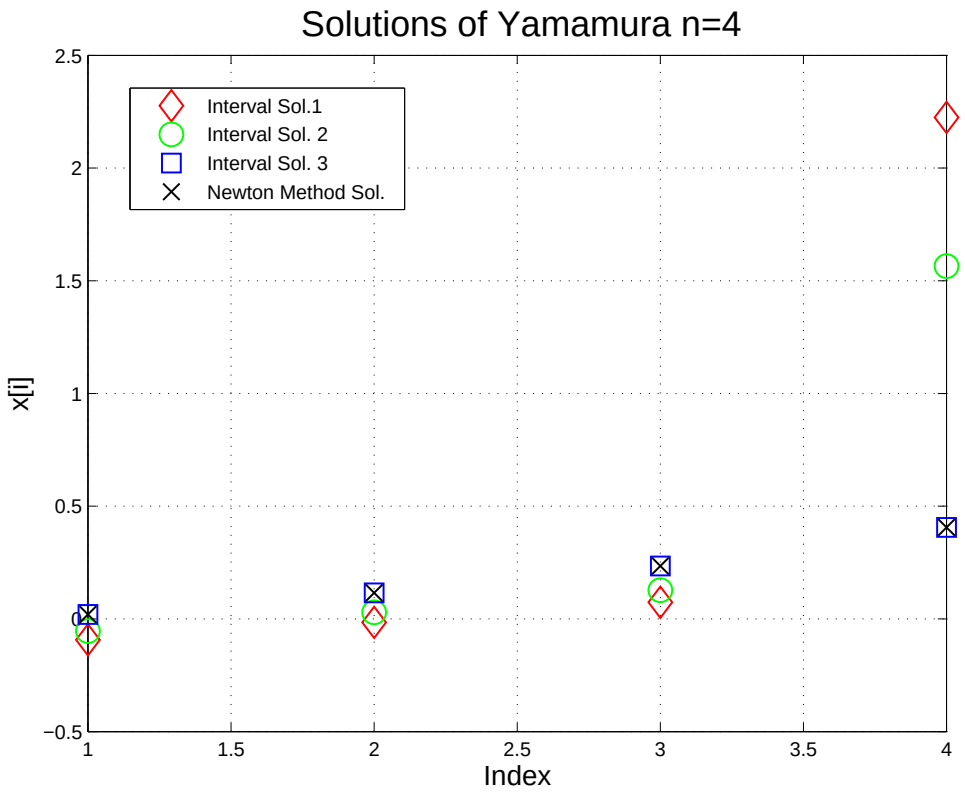


Figure 4.2: Solutions of Yamamura given by Newton Method and CSP model are depicted.

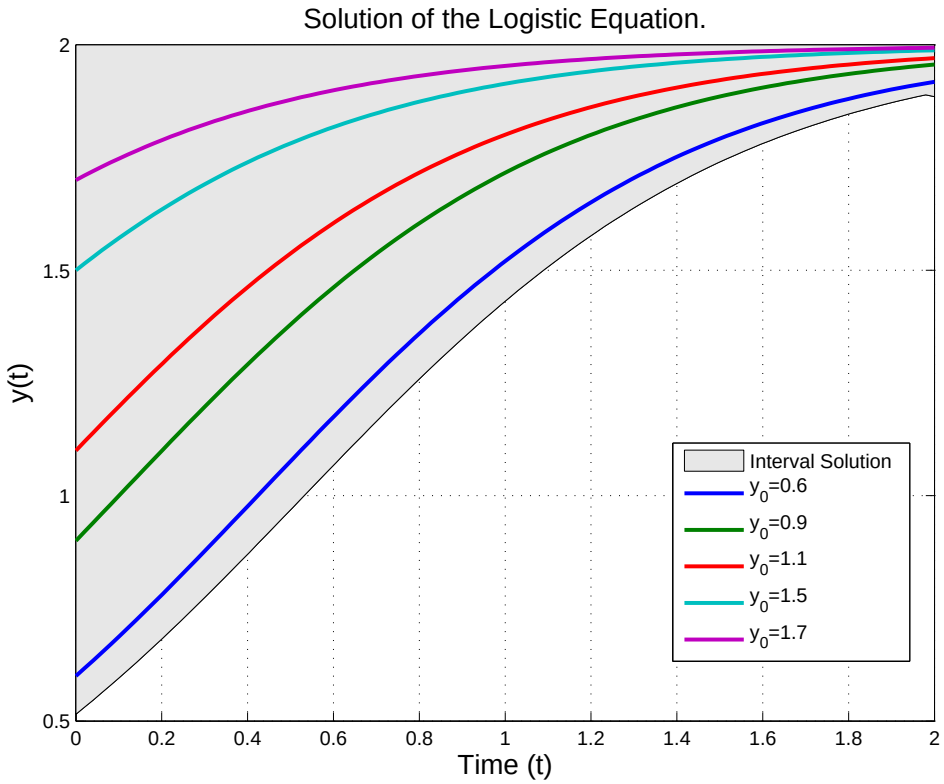


Figure 4.3: Interval solution of the logistic equation with uncertainty in the initial condition

where θ is the rate of maximum population growth and M is the so-called carrying capacity (i.e., the maximum sustainable population). If the initial condition is: $y_0 = [0.5, 2]$, all the solutions corresponding to the all initial conditions in the given interval can be enclosed. The result is shown in Fig. 4.3.

Burgers equation:

The previous example was a linear ODE. Let's retake the Burgers equation Eq. (4.1.4) as an example of nonlinear PDE. This problem has been already solved with uncertainty in the boundary condition because that was used as a new strategy to compute the snapshots in the IPOD method. Let's see more closely how the interval solution enclose all the solution with boundary condition in the given interval. Four snapshots are presented in the Figure Fig. 4.4 corresponding to four time steps.

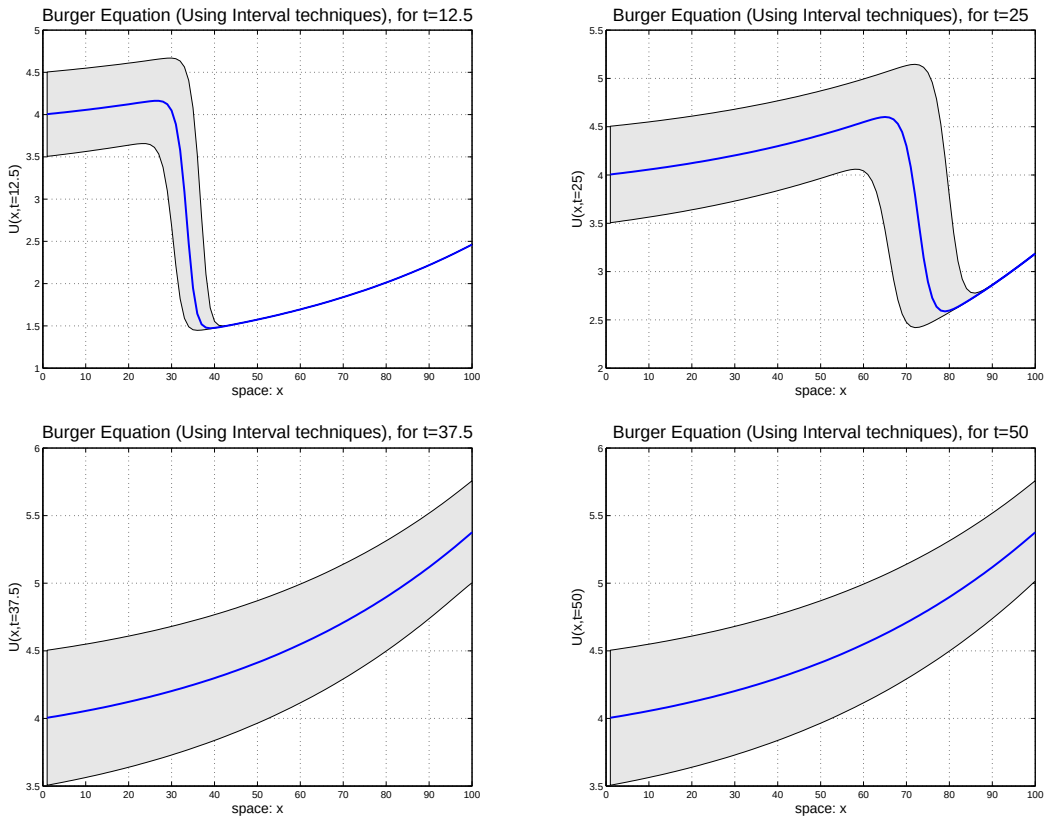


Figure 4.4: Interval solution of the Burgers equation with uncertainty in the Boundary condition: $U(0,t) = u(t) = [0.5, 1.5]$.

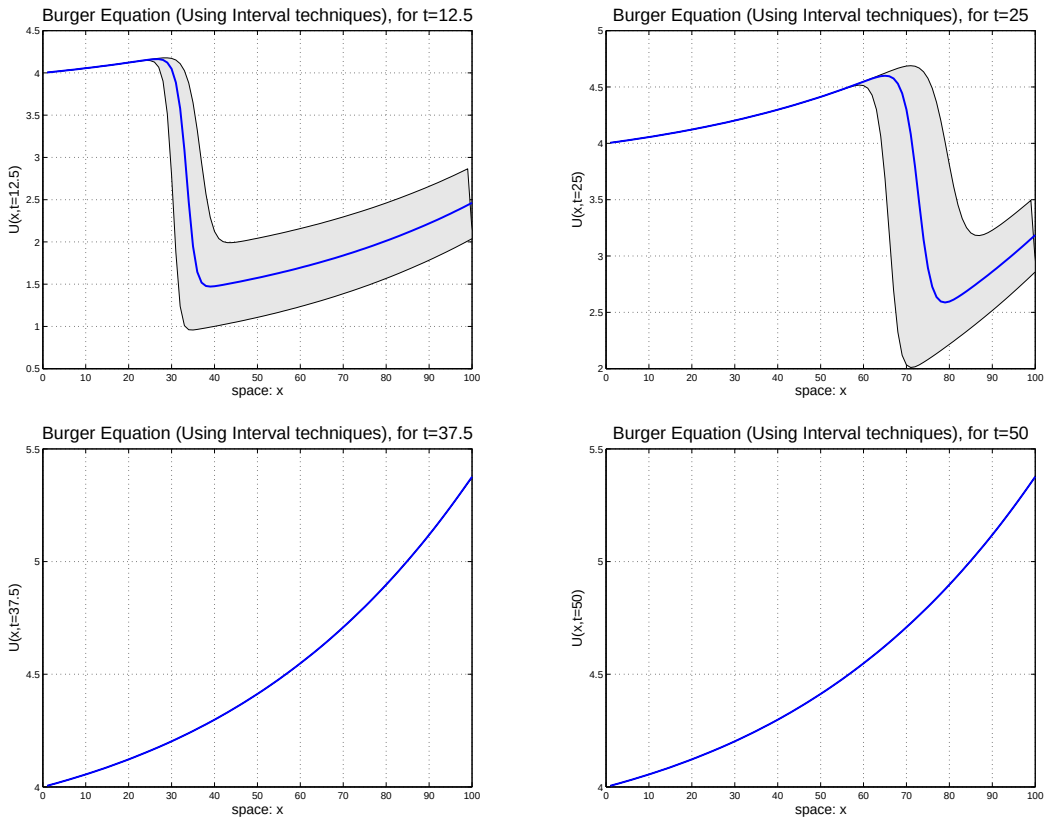


Figure 4.5: Interval solution of the Burgers equation with uncertainty in the Initial condition: $U(x,0) = u(x) = [0.5, 1.5]$

In Fig. 4.5 is presented the solution of Burgers equation, but now with uncertainty in the initial condition.

Logistic equation. Predicting behavior

Let's retake the logistic equation, but this time with the intention of to predict what would be the behavior of the solution if it has uncertainty at some piece of data. In this example the values $\theta = 1$ and $M = 2$ are fixed. The domain of

$$\frac{dy}{dt} = y(2 - y), \tag{4.3.2}$$

$t \in [0, 2]$, is discretized in 100 variables. Let us assume that the value of the variable $y_{20} = y(t_{20})$, where y is the solution of Eq. (4.3.2), then if we considerate $\bar{y}_{20} = y_{20} + [-0.1, 0.1]$

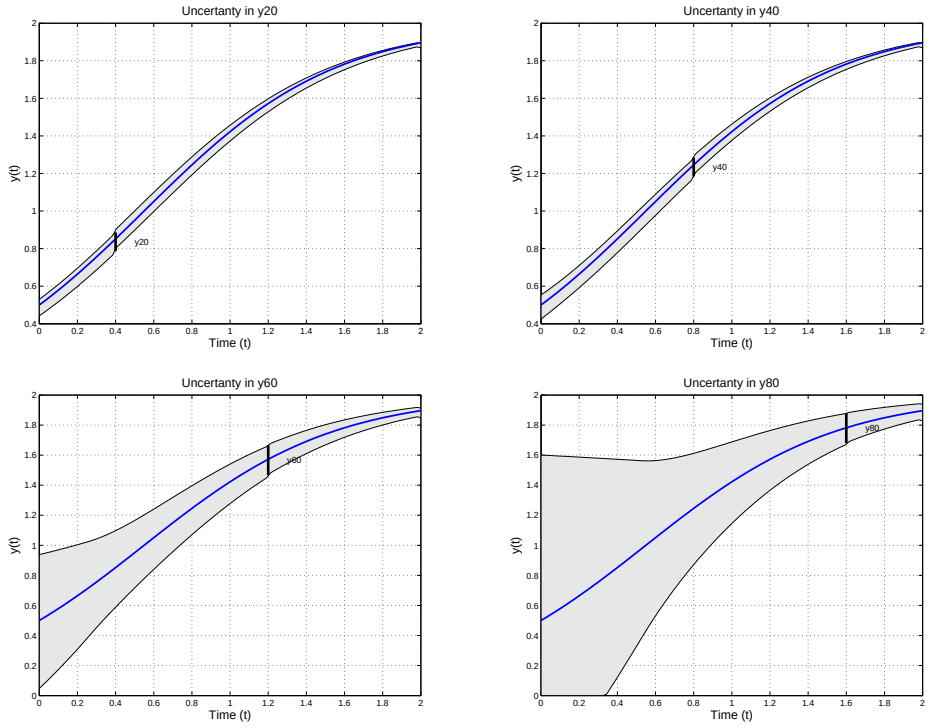


Figure 4.6: Predicting the initial condition in the logistic equation knowing uncertainty in some value of the variables.

the solution gives as initial condition $y_0 = [0.39, 0.5816]$. Three additional examples are presented in Fig. 4.6.

- $\bar{y}_{40} = y_{40} + [-0.1, 0.1]$, the initial condition is: $y_0 = [0.316, 0.6616]$
- $\bar{y}_{60} = y_{60} + [-0.1, 0.1]$, the initial condition is: $y_0 = [0.0471, 0.9381]$
- $\bar{y}_{80} = y_{80} + [-0.1, 0.1]$, the initial condition is: $y_0 = [0, 1.6004]$

Now, if uncertainty is in two pieces of data, for example, $\bar{y}_{40} = y_{40} + [-0.05, 0.05]$ and $\bar{y}_{60} = y_{60} + [-0.05, 0.05]$, the solution not only gives an interval where the initial condition is, but also information about theta is given. In this case, $y_0 = [0, 1.2841]$ and $\theta = [1, 1]$. See Fig. 4.7.

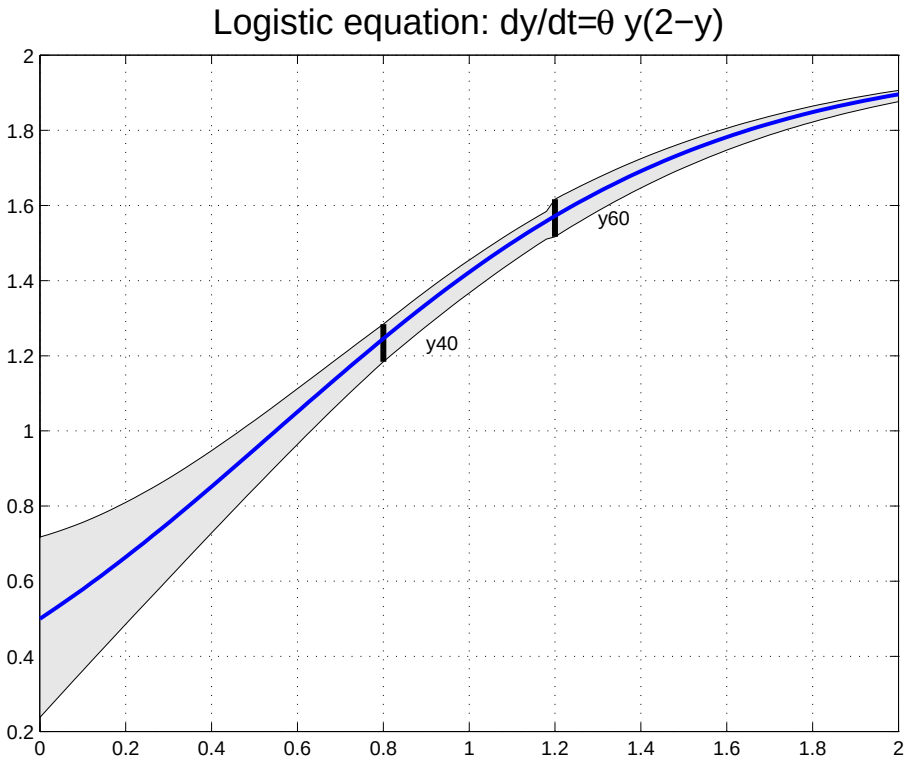


Figure 4.7: Predicting the initial condition in the logistic equation knowing uncertainty in some value of the variables

Chapter 5

Conclusions and Future Work

People do not like to think. If one thinks, one must reach conclusions. Conclusions are not always pleasant.

Helen Adams Keller. June 27, 1880 - June 1, 1968

5.1 Conclusions

1. We described three of the most well-known approach of Model-Order Reduction: Krylov method, Wavelet method, and POD method, and they were tested solving four problems of different kinds: a parametric-linear system of equations, a nonlinear system of equations, the heat equation as an example of a linear PDE and Burger's equations as an example of a nonlinear PDE. After experimenting and taking notes, we discovered that POD is the best method to reduce the dimension of the subspace where the given solution lies. However, it has the inconvenient of spending much time computing the snapshots needed to create the reduced basis that is used in future calculations.

   Because of that, we used interval analysis techniques to find a faster way to find the snapshots and reducing the time needed to compute them. We were able to reduce significantly the time of computation of the snapshots, Tab. 4.1.

2. When we are solving a nonlinear problem, and we are forced to work with a basis that is far away from the solution of the problem in terms of the orthogonal projection, we have to deal with an overdetermined nonlinear system of equation that could be solved using either Newton method or Gauss-Newton method, this lead us to a predicament: if we use Newton method, we have spend time and storage for computing the Hessian due to the second order information terms, on the other hand if we use Gauss-Newton method we save time and store spacing since we no longer need to compute the second order information, however we lose the q-quadratic rate of convergence. To alleviate this issue, we implemented techniques of regularization proposed by Levenberg and Marquardt [40, 26] when we solved the nonlinear system of equations proposed by Yamamura, we were able to reduce the number of iterations and the computation time, but a loss in precision is reported caused by the chosen basis (see Tab. 4.2).

3. Newton Method has q-quadratic order of convergence when the initial condition is close to the solution of the problem, which is the reason why classical Newton method is known as a local method. Another well known fact is that Newton method finds only one solution of the problem we are solving. In this work, we implemented a global version of Newton's method capable of finding all the solutions of a nonlinear equation, see examples 2 and 3.

4. Finally, interval analysis and constraint solving was used analyze the behavior of the solution of a dynamical system when uncertainty exist in one or more of its parameters, and it is used to predict the location of certain parameter when uncertainty in some data is known.

5.2 Future work

In the future, our efforts will be focused on the following aspects:

Numerically:

1. The model-order reduction method using wavelets can be improved, finding a basis based on a customized wavelet for each problem.

2. The use of interval constraint techniques to handle high nonlinear problems.

3. The use of interval constraint techniques to handle outliers in measured data (common with sensors).

4. A MOR method that preserves structure must be designed.

5. Find out how to handle a MOR with constraints.

Applications:

1. We can use interval constraint techniques and model-order reduction in order to given measurements obtained by sensors can be used to understand the initial conditions of a given phenomenon

2. And this knowledge in turn can be used to predict behaviors at future times or other spatial locations.

3. To get faster information about initial conditions that **are not desirable**.

References

[1] H. Aling, R. Kosut, A. Emami-Naeini, and J. Ebert, *Nonlinear model reduction with application to rapid thermal processing*, in Proceedings of the 35th IEEE Conference on Decision and Control, 1996, vol. 4, Dec. 1996, pp. 4305–4310 vol.4.

[2] Z. Bai, *Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems*, Applied Numerical Mathematics, 43 (2002), pp. 9–44.

[3] I. Balk, *Arnoldy based passive model order reduction algorithm*, in Electrical Performance of Electronic Packaging, 2000, IEEE Conference on., 2000, pp. 251–254.

[4] U. Baur, P. Benner, and L. Feng, *Model Order Reduction for Linear and Nonlinear Systems: A System-Theoretic Perspective*, Archives of Computational Methods in Engineering, 21 (2014), pp. 331–358.

[5] G. Berkooz, P. Holmes, and J. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annual Rev. Fluid Mech, (1993), pp. 539–575.

[6] W. Beyene and J. Schutt-Aine, *Krylov subspace-based model-order reduction techniques for circuit simulations*, in IEEE 39th Midwest symposium on Circuits and Systems, 1996, vol. 1, Aug. 1996, pp. 331–334 vol.1.

[7] A. Björck, *Numerical Methods for Least Squares Problems*, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, Jan. 1996.

[8] P. Brown, *A Local Convergence Theory for Combined Inexact-Newton/Finite-Difference Projection Methods*, SIAM Journal on Numerical Analysis, 24 (1987), pp. 407–434.

[9] P. Brown, A. Hindmarsh, and L. Petzold, *Using Krylov Methods in the Solution of Large-Scale Differential-Algebraic Systems*, SIAM Journal on Scientific Computing, 15 (1994), pp. 1467–1488.

[10] L. Cai and R. White, *Reduction of Model Order Based on Proper Orthogonal Decomposition for Lithium-Ion Battery Simulations*, Journal of the Electrochemical Society, 156 (2009), pp. A154–A161.

[11] A. Chatterjee, *An introduction to the Proper Orthogonal Decomposition*, Current Science, 78 (2000), pp. 808–817.

[12] J. Chen and S. Kang, *An algorithm for automatic model-order reduction of nonlinear MEMS devices*, in Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on, vol. 2, 2000, pp. 445–448 vol.2.

[13] J. Chung and J. Nagy, *An Efficient Iterative Approach for Large-Scale Separable Nonlinear Inverse Problems*, SIAM Journal on Scientific Computing, 31 (2010), pp. 4654–4674.

[14] M. Cotlar and R. Cignoli, *Introduction to Functional Analysis*, North Holland Publishing Company and Elsevier Publishing Company, first edition ed., Apr. 1974.

[15] L. Daniel and J. Phillips, *Model order reduction for strictly passive and causal distributed systems*, in Design Automation Conference, 2002. Proceedings. 39th, 2002, pp. 46–51.

[16] C. de Villemagne and R. Skelton, *Model reductions using a projection formulation*, in 26th IEEE Conference on Decision and Control, 1987, vol. 26, Dec. 1987, pp. 461–466.

[17] J. Demmel and P. Koev, *Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials*, Linear Algebra and its Applications, 417 (2006), pp. 382–396.

[18] J. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, Jan. 1996.

[19] I. M. Elfadel and D. Ling, *Zeros and passivity of arnoldi-reduced-order models for interconnect networks*, in Design Automation Conference, 1997. Proceedings of the 34th, Inc, 1997, pp. 28–33.

[20] R. Freund and P. Feldmann, *Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm*, in , 1996 IEEE/ACM International Conference on Computer-Aided Design, 1996. ICCAD-96. Digest of Technical Papers, Nov. 1996, pp. 280–287.

[21] L. Gabbay, J. Mehner, and S. Senturia, *Computer-aided generation of nonlinear reduced-order dynamic macromodels. I. Non-stress-stiffened case*, Journal of Microelectromechanical Systems, 9 (2000), pp. 262–269.

[22] E. Gad, A. Dounavis, M. Nakhla, and R. Achar, *Passive model order reduction of multiport distributed interconnects*, in Design Automation Conference, 2000. Proceedings 2000, 2000.

[23] L. Granvilliers and F. Benhamou, *RealPaver: An Interval Solver Using Constraint Satisfaction Techniques*, ACM Trans. on Mathematical Software, 32 (2006), pp. 138–156.

[24] E. Grimme, D. Sorensen, and P. Dooren, *Model Reduction of State Space Systems Via an Implicitly Restarted Lanczos Method*, Numerical Algorithms, 12 (1996), pp. 1–31.

[25] Q. Gu, M. Barbato, and J. Conte, *Handling of Constraints in Finite-Element Response Sensitivity Analysis*, Journal of Engineering Mechanics, 135 (2009), pp. 1427–1438.

[26] F. Gunawan, *Levenberg Marquardt Iterative Regularization for the Pulse-Type Impact-Force Reconstruction*, Journal of Sound and Vibration, 331 (2012), pp. 5424–5434.

[27] M. Hernandez, *Reduced-Order Modeling Using Orthogonal and Bi-Orthogonal Wavelet Transforms*, PhD thesis, The University of Texas at El Paso, 2013.

[28] M. Hernandez, L. Velazquez, and M. Argaez, *A Comparison of Wavelet-Based Schemes for Parameter Estimation*, in High Performance Computing Modernization Program Users Group Conference (HPCMP-UGC), 2010 DoD, June 2010, pp. 71–76.

[29] K. Hoffman and R. Kunze, *Linear Algebra*, Prentice-Hall, 1971.

[30] I. Ipsen and C. Meyer, *The idea behind krylov methods*, American Mathematical Monthly, 105 (1997), pp. 889–899.

[31] G. Kharate and V. Patil, *Color Image Compression Based On Wavelet Packet Best Tree*, arXiv:1004.3276 [cs], (2010). arXiv: 1004.3276.

[32] M. Kowalski and J. Jin, *Karhunen-Loeve based model order reduction of nonlinear systems*, in IEEE Antennas and Propagation Society International Symposium, 2002, vol. 2, 2002, pp. 552–555 vol.2.

[33] G. Lassaux, *High-fidelity reduced-order aerodynamic models : application to active control of engine inlets*, thesis, Massachusetts Institute of Technology, 2002.

[34] J. Li, *Model reduction of large linear systems via low rank system gramians*, thesis, Massachusetts Institute of Technology, 2000.

[35] P. Li and L. Pileggi, *NORM: compact model order reduction of weakly nonlinear systems*, in Design Automation Conference, 2003. Proceedings, June 2003, pp. 472–477.

[36] Y. Lin and M. Stadtherr, *Validated solutions of initial value problems for parametric odes*, Appl. Numer. Math., 57 (2007), pp. 1145–1162.

[37] Y. Liu, *Model order-reduction of RC(L) interconnect including variational analysis*, Center for Electronic Design Automation, Carnegie Mellon University, 1999.

[38] M. Ma and R. Khazaka, *Model Order Reduction with Load Constraints*, in 2006 IEEE Electrical Performance of Electronic Packaging, Oct. 2006, pp. 193–196.

[39] X. Ma and J. De Abreu-Garcia, *On the computation of reduced order models of nonlinear systems using balancing technique*, in , Proceedings of the 27th IEEE Conference on Decision and Control, 1988, Dec. 1988, pp. 1165–1166 vol.2.

[40] D. Marquardt, *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*, Journal of the Society for Industrial and Applied Mathematics, 11 (1963), pp. 431–441.

[41] A. Marquez, J. Espinosa, and D. Odloak, *Model Reduction Using Proper Orthogonal Decomposition and Predictive Control of Distributed Reactor System, Model Reduction Using Proper Orthogonal Decomposition and Predictive Control of Distributed Reactor System*, Journal of Control Science and Engineering, Journal of Control Science and Engineering, 2013, 2013 (2013), p. e763165.

[42] B. Moore, *Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction*, Automatic Control, IEEE Transactions on, 26 (1981), pp. 17 – 32.

[43] R. Moore, *Introduction to Interval Analysis*, Prentice-Hall, 2009.

[44] N. Nedialkov, K. Jackson, and G. Corliss, *Validated solutions of initial value problems for ordinary differential equations*, Applied Mathematics and Computation, 105 (1999), pp. 21–68.

[45] C. Paige and M. Saunders, *Solution of Sparse Indefinite Systems of Linear Equations*, Siam Journal on Numerical Analysis, (1973).

[46] C. Paige and M. Saunders, *Solution of Sparse Indefinite Systems of Linear Equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629.

[47] J. Phillips, *Model reduction of time-varying linear systems using approximate multipoint Krylov-subspace projectors*, in 1998 IEEE/ACM International Conference on Computer-Aided Design, 1998. ICCAD 98. Digest of Technical Papers, Nov. 1998, pp. 96–102.

[48] J. Phillips, *Projection-based approaches for model reduction of weakly nonlinear, time varying systems*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, (2003), pp. 171–187.

[49] M. Rathinam and L. Petzold, *An iterative method for simulation of large scale modular systems using reduced order models*, in Proceedings of the 39th IEEE Conference on Decision and Control, 2000, vol. 5, 2000, pp. 4630–4635 vol.5.

[50] M. Rathinam and L. Petzold, *A New Look at Proper Orthogonal Decomposition*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 1893–1925.

[51] M. Rewienski, *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*, thesis, Massachusetts Institute of Technology, 2003.

[52] J. Roychowdhury, *Reduced-order modelling of time-varying systems*, in Design Automation Conference, 1999. Proceedings of the ASP-DAC '99. Asia and South Pacific, Jan. 1999, pp. 53–56 vol.1.

[53] C. Ryan, L. Youdong, L. D.S., and M. A.S., *Interval methods for nonlinear equation solving applications.*

[54] Y. Saad, *Numerical Methods for Large Eigenvalue Problems, Revised Edition*, SIAM - Society for Industrial & Applied Mathematics, Philadelphia, revised edition ed., May 2011.

[55] L. Sanjay, E. Jerrold, and G. Sonja, *Empirical model reduction of controlled nonlinear systems*, in Proceedings of the IFAC World Congress, 1999, pp. 473–478.

[56] W. Schilders, H. van der Vorst, and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, vol. 13 of Mathematics in Industry, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[57] M. Sharan and D. Pradhan, *A Numerical Solution of Burgers Equation Based on Multigrid Method*, International Journal of Advancements in Electronics and Electrical Engineering – IJAEEE, 2 (2013).

[58] B. Sheehan, *ENOR: model order reduction of RLC circuits using nodal equations for efficient factorization*, in Design Automation Conference, 1999. Proceedings. 36th, 1999, pp. 17–21.

[59] B. Sheehan, *Projective convolution: RLC model-order reduction using the impulse response*, in Design, Automation and Test in Europe Conference and Exhibition 1999. Proceedings, mar 1999, pp. 669–673.

[60] S. Sidi-Ali-Cherif and K. Grigoriadis, *Efficient model reduction of large scale systems using Krylov-subspace iterative methods*, International Journal of Engineering Science, 41 (2003), pp. 507–520.

[61] L. Silveira, M. Kamon, I. Elfadel, and J. White, *A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits*, Computer Methods in Applied Mechanics and Engineering, 169 (1999), pp. 377–389.

[62] L. Silveira, M. Kamon, and J. White, *Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-d interconnect structures*, 1995.

[63] D. Sorensen, *Implicit Application of Polynomial Filters in a k-Step Arnoldi Method*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 357–385.

[64] D. C. Sorensen, *Implicitly Restarted Arnoldi/Lanczos Methods for Large Scale Eigenvalue Calculations*, in Parallel Numerical Algorithms, D. E. Keyes, A. Sameh, and V. Venkatakrishnan, eds., no. 4 in ICASE/LaRC Interdisciplinary Series in Science and Engineering, Springer Netherlands, 1996, pp. 23–25–165.

[65] J. Starck, F. Murtagh, and J. Fadili, *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*, Cambridge University Press, May 2010.

[66] L. Trefethen and D. I. Bau, *Numerical Linear Algebra*, SIAM, June 1997.

[67] J. Walker, *A Primer on Wavelets and Their Scientific Applications, Second Edition*, Chapman and Hall/CRC, Boca Raton, second edition ed., Jan. 2008.

[68] J. White, *A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices*, in International Conference Computer Aided Design, 2013, pp. 252–257.

[69] K. Willcox and J. Peraire, *Balanced Model Reduction via the Proper Orthogonal Decomposition*, AIAA Journal, 40 (2002), pp. 2323–2330.

[70] K. Willcox, J. Peraire, and J. White, *An Arnoldi approach for generation of reduced-order models for turbomachinery*, Computers & Fluids, 31 (2002), pp. 369–389.

[71] K. Yamamura, H. Kawata, and A. Tokue, *Interval Solution of Nonlinear Equations Using Linear Programming*, BIT Numerical Mathematics, 38 (1998), pp. 186–199.

[72] Y. Zhang, M. Henson, and Y. Kevrekidis, *Nonlinear model reduction for dynamic analysis of cell population models*, Chemical Engineering Science, 58 (2003), pp. 429–445.

[73] R. Zimmermann, A. Vendl, and S. Görtz, *Reduced-Order Modeling of Steady Flows Subject to Aerodynamic Constraints*, AIAA Journal, 52 (2014), pp. 255–266.

Curriculum Vitae

Leobardo Valera was born in Zaraza, Venezuela on January 19, 1970. He has always interested in math and sciences in general. He also loves reading science fiction books, Isaac Asimov is his favorite author. In August 1988, He finished high school and was accepted in the "Universidad Central de Venezuela (UCV)" in 1989, where He obtained his bachelor's degree in Mathematics. Years later, he obtained his first master degree from UCV.

Having worked as lecturer in the "Universidad Metropolitana" for three years, he was awarded as the "young researcher in exact sciences" in 2002 due to the application of his thesis.

He was a lecturer in The "Universidad Metropolitana" for 14 years, until he decided to accept Dr. Argaez's invitation to move at El Paso, Texas, and to start his PhD studies at The University of Texas at El Paso, UTEP, in the Computational Science Program. He is currently pursuing his research as PhD student under Dr. Martine Ceberio.

The University of Texas at El Paso
Program in Computational Science
500 West University Ave. CCSB 2.0902 El Paso, Texas 79968-0514
lvalera@miners.utep.edu
leobardovalera@gmail.com