2018-01-01

Decision Making For Dynamic Systems Under Uncertainty: Predictions And Parameter Recomputations

Leobardo Valera
University of Texas at El Paso, lvalera@gmail.com

DECISION MAKING FOR DYNAMIC SYSTEMS UNDER UNCERTAINTY:
PREDICTIONS AND PARAMETER RECOMPUTATIONS

LEOBARDO VALERA

Doctoral Program in Computational Science

APPROVED:

Martine Ceberio, Chair, Ph.D.

Vladik Kreinovich, Ph.D.

Sergio Cabrera, Ph.D.

Natasha Sharma, Ph.D.

Mark Stadtherr, Ph.D.

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

I wanted a perfect end–
ing. Now I have learned, the hard
way, that some poems do not rhyme, and
some stories do not have a clear begin–
ning, middle, and end. Life is about not
knowing, having to change, taking the
moment and making the best of it,
without knowing what is going
to happen next. Deli–
cious Ambiguity.

♡

Gilda Radner

DECISION MAKING FOR DYNAMIC SYSTEMS UNDER UNCERTAINTY: PREDICTIONS AND PARAMETER RECOMPUTATIONS

by

LEOBARDO VALERA

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Doctoral Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

May 2018

Acknowledgements

If I have seen further than others, it is by standing upon the shoulders of giants.

Isaac Newton

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, Dr. Natasha Sharma, and Dr. Mark Statdherr for kindly accepting being part of my thesis committee.

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

My endless thanks goes to my kids Dayana and Leo David. This is dedicated to you both.

I also want to thank Dr. Argaez for inviting me to obtain a Ph.D. 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, Cabrera, and Hossain.

This work was supported by the Department of the Army ARL Grant project.

Abstract

In this thesis, we are interested in making decision over a model of a dynamic system. We want to know, on one hand, how the corresponding dynamic phenomenon unfolds under different input parameters (simulations). These simulations might help researchers to design devices with a better performance than the actual ones. On the other hand, we are also interested in predicting the behavior of the dynamic system based on knowledge of the phenomenon in order to prevent undesired outcomes. Finally, this thesis is concerned with the identification of parameters of dynamic systems that ensure a specific performance or behavior.

Understanding the behavior of such models is challenging. The numerical resolution of those model leads to systems of equations, sometimes nonlinear, that can involve millions of variables making it prohibitive in CPU time to run repeatedly for many different configurations. These types of models rarely take into account the uncertainty associated with the parameters. Changes in the definition of a model, for instance, could have dramatic effects on the outcome of simulations. Therefore, neither reduced models nor initial conclusions could be 100% relied upon.

Reduced-Order Modeling (ROM) provides a concrete way to handle such complex simulations using a realistic amount of resources. Interval Constraint Solving Techniques can also be employed to handle uncertainty on the parameters.

There are many techniques of Reduced-Order Modeling (ROM). A highly used technique is the Proper Orthogonal Decomposition (POD), which is based on the Principal Com-

ponent Analysis (PCA). In this thesis, we use interval arithmetic and Interval Constraint Solving Techniques to modify the POD to be able to handle uncertainty associated with the the parameters of the system.

In this thesis, we propose an increased use of such techniques for dynamic phenomena at the time they unfold to identify their key features to be able to predict their future behavior. This is specifically important in applications where a reliable understanding of a developing situation could allow for preventative or palliative measures before a situation aggravates.

When solving or simulating the reality of physical phenomena, the Finite Element Method (FEM) often constitutes a method of choice. In this thesis, we propose a novel technique to handle uncertainty in FEM using interval computations and Interval Constraint Solving Techniques. We then demonstrate the performance of our work on two problems: a static convection-diffusion problem and a transitory nonlinear heat equation as a first step towards our goal of fuel simulation.

There exists situations where simulations or experiments have to be executed while neither high performance computers nor internet connection is available. For example, soldiers that need to take an immediate decision in an unexpected situation or engineers on a field work. In this thesis, we used the Reduced-Order Modeling techniques to design an application for mobile devices.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Introduction

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

Michelangelo

Newton's law of cooling states that

> The rate of heat loss of a body is directly proportional to the difference in the temperatures between the body and its surroundings

If this law is presented this way, not many people might understand what it means, and if it is presented as

$$\frac{dT}{dt} = k(T - T_A)$$

where T represents the temperature of a body, and T_A is the temperature of the room where the body is, then even fewer people might follow it.

The reality is that everybody "knows" this law, knowing neither its name nor its representation as a differential equation or dynamic systems. For instance, when someone has a glass of hot water and he or she requires that the liquid get cold faster, he or she puts the glass in the refrigerator. If the temperature surrounding the glass changes, then the liquid cools faster. What it has been done is **change a parameter** (the temperature of

the surrounding) to reduce the cooling time. The trajectory of the **dynamic system** has been changed as consequence of changing a parameter.

Let us continue with this example, but now, we assume that a forensic doctor working along the police on a case. An important information that is required to solve any case is the approximate time of death. The doctor **estimates** the time of death t_0 using

- the temperature of the body when the person was alive (around 37^oC);
- the temperature of the room where the body was found; and
- the temperature of the body (he or she took an observation).

This is an example of parameters estimation. The doctor knows that his or her estimation is not reliable because the temperature of a person is not exactly 37^oC, the temperature of the room changes depending on the heater or the AC, and the temperature of the body can be affected by the measuring instrument. In other words, the estimation of t_0 is not reliable because the data is full of uncertainty. In order to obtain reliable results, the doctor needs to take into account the uncertainty in his or her calculations.

In this thesis, we work with complex dynamic systems that are modeled using differential equations and are usually solved using numerical methods. A classical way to solve these equations consists in discretizing them and using real Newton-Like methods, but occasionally, such methods are unable to find the roots of a function even when the functions have real zeros. For example, take the function

$$f(x) = \operatorname{sign} x \sqrt{|x|}.$$

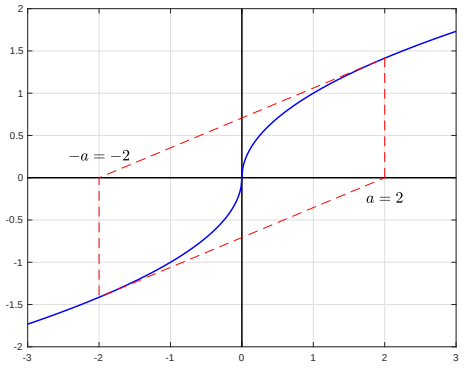
If $x \neq 0$, $f'(x)$ is defined

$$f'(x) = \frac{1}{2\sqrt{|x|}}.$$

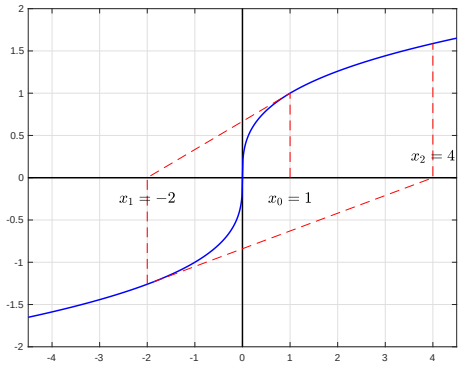
Therefore, the Newton iteration is

$$x_{n+1} = -x_n.$$

It does not matter what the initial point is, the Newton method will not converge, see Figure 1(a).



(a) Infinite loop



(b) Iterations move away from the solution

Figure 1: Examples of two cases where the Newton Method in which does not converge

Now, consider the function $f(x) = \sqrt[3]{x}$. The derivative of this function is $f'(x) = \frac{1}{3\sqrt[3]{x^2}}$. In this case, the Newton iteration becomes

$$x_{n+1} = -2x_n,$$

which means that the iterations move away more and more from the solution, see Figure 1(b). In both cases, if we use intervals techniques such as the well-known bisection method [24], then the solution is reached in few iterations.

Not finding a solution of system of equations is bad, but it is worse when Newton method returns a wrong solution. In his article: *False Numerical Convergence in Some Generalized Newton Methods* [92], Dr. Stephen M. Robinson proves that these methods can produce false convergence.

In this thesis, we aim to conduct fast a reliable simulations of dynamic systems. Such dynamic systems can be represented as a system of differential equations that can involve parameters and millions de variables, making it prohibitive in CPU time to run repeatedly for many configurations. Reduced-Order Model provides a way to handle such complex simulations using a realistic amount of time. The most used approach for doing so being Proper Orthogonal Decomposition (POD). The key idea of POD is to reduce a large number of correlated vectors (snapshots) to a much smaller number of uncorrelated ones while retaining as much as possible of the variation of the original ones. Although this method is highly used, it barely takes into account the uncertainty on the data.

In this thesis, we show how intervals and constraint solving techniques can be used to compute all the snapshots at once. With this new process, we have two achievement: on one hand, we can handle uncertainty in one or more parameters, and on the other hand, we reduce the snapshots computational cost.

The ability of searching solutions of a system of equations on a subspace allows us to run simulations much more lightweight while maintaining quality of the outputs. However, going beyond merely solving Full-Order discretized Models and looking at other problems than simulations, we could simply be making an observation (or observations) of an unfolding phenomenon and it would be valuable to be able to understand it as it unfolds and predict its future behavior.

We show how we can translate FOM data into ROM data and combine Interval Constraint Solving Techniques (ICST) with Reduced-Order Modeling techniques to properly account for both challenges: size and uncertainty.

Using the model of a dynamic system, we aim to find certain parameter values that guarantee a specific outcome of the modeled dynamic phenomenon. Assuming this phenomenon is modeled as a parametric differential equation (ODE / PDE), the parameters that lead to a certain outcome can be found discretizing the ODE/PDE equation leading to a parametric system of equations, and Reduced-Order Modeling and interval constraint solving techiques to determine such values of the phenomenons parameters.

In this work, we used Reduced-Order Modeling and interval constraint solving techniques to identify reliable intervals in which the desired parameters values lie to guarantee a specific outcome of the modeled dynamic phenomenon.

Ordinary differential equations and partial differential equations can be used to model complex physical simulations, such as combustions JP-8 fuels. Real fuels are chemically complex and often contain thousands of hydrocarbon compounds. Simulations of these kind of models are complex due to the difficulty of capturing uncertainty.

In this thesis, we use a transitory nonlinear heat equation as mock-up problem that mimics the fundamental physics of combustions, and we introduce the Interval Finite Element Method to determine the envelope where the solution lies.

Organization of the document: This manuscript consist of four parts:

Part I: contains the technical pre-requisites. It is splitted in three chapters: Chapter 1 is dedicated to reduced-order modeling. We start with the definition of ROM, and we give a small example of it. We present the Krylov, wavelets, and the proper orthogonal decomposition method (POD).

Chapter 2 contains the basic concepts of interval arithmetic and interval constraint solving techniques.

Chapter 3 is the last chapter of this part. Here is presented the Theorem of Green, which is necessary to write a partial differential equation in the weak form. In this part, we show two examples of nonlinear partial differential equations.

Part II: refers to the contributions. It starts with the Chapter 4 (problem statement) and follows with the contributions. In Chapter 5, the first contribution is presented, which is to use interval constraint solving techniques to improve the well-known method POD.

We can use Interval constraint solving techniques and reduced-order modeling to estimate parameters of a dynamical systems when we have observations of the dynamic phenomenon. We present the techniques of predicting behavior of dynamic systems and parameter estimation in (Chapter 6), and we show how to recompute parameters of a model when the dynamic system has been perturbed in (chapter 7). We show how to handle uncertainty in the finite element method in the (Chapter 8), and we will use these procedure to handle uncertainty in combustions problem models.

The last chapter of this part (Chapter 9) is not really a contribution, but it is an application. We take advantage of our ability of handling uncertainty and reduced-order model to design an application that can run on mobile devices such as tablets or cell phones.

Part III: This part consists of only the Chapter 10. In this chapter, the reader can find the conclusion and the future work.

Part IV (Appendix): This part also consists of two chapters: In (Chapter 11), we can find the basic concepts of linear a metric spaces. In (Chapter 12), it is illustrated with a example how we can use interval constraint solving techniques to compute the spectrum of a non-defective matrix.

In this thesis, there will be few statements that will require formal proofs, but for those that require one, the end proof mark will be a black square symbol (■).

Typed in LaTeX

Part I

Technical Pre-Requisites

UTEP 2018

Chapter 1

Reduced-Order Modeling

All problems become smaller when you confront them instead of dodging them.

William F. Halsey

When solving a large-dimensional system of equations, the main idea of Reduced-order modeling is finding a subspace spanned by a basis Φ whose dimension is much smaller than the dimension of the original space. As a result, finding the solution on the original space is mathematically equivalent to find the coordinates of the solution in such subspace, but computationally the process is faster because the solution is sought in a smaller space.

1.1 Background

Let us consider a system of equations:

$$F(x) = 0 \tag{1.1.1}$$

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

Assuming that Equation (1.1.1) has solution, there exists a x^* such that $F(x^*) = 0$, and we acknowledge that:

$$\exists\ (n \times k) \text{ matrix } \Phi, \text{ with } k \ll n,$$

whose columns are the vectors of a basis of W, such that x^* can be expressed as:

$$x^* = \Phi p^* + z$$

where $p \in \mathbb{R}^k$.

Assuming that the choice of the subspace where x^* lies is accurate, we can infer that $z \approx 0$. As a result, we are now solving:

$$F(\Phi p) = 0 \quad \text{over } \mathbb{R}^k.$$

In Figure 1.1, we illustrate this process on an example where we originally aim to solve a 3D problem but are able to focus on a 2D subspace. The solution $x \in \mathbb{R}^n$ corresponds to the non-perpendicular vector (red) not contained in the plane, which is the subspace spanned by Φ. The lower-dimension unknown $p \in W \subset \mathbb{R}^k$ corresponds to the orthogonal projection (green) of x onto the plane W. The orthogonal vector (blue) corresponds to the orthogonal component, z, that is approximately equal to zero when Φ is a "good" basis.

Let us now illustrate the workings of Reduced-Order Modeling through the following example:

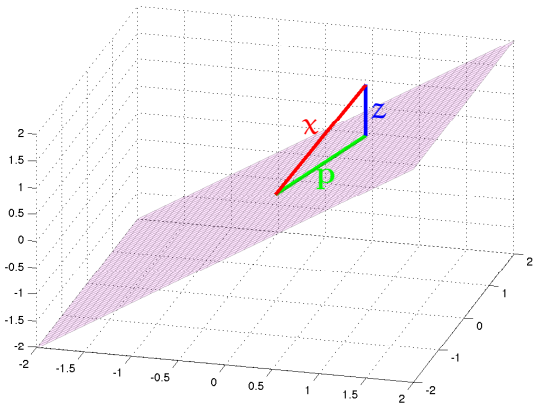


Figure 1.1: Graphical Representation of Reduced-Order Modeling.

Example 1. *Consider the following nonlinear system of equations:*

$$\begin{cases} (x_1^2 + x_1 - 2)(x_2^2 + 1) &= 0 \\ (x_2^2 - 5x_2 + 6)(x_1^2 + 1) &= 0 \\ (x_3^2 - 2x_3 - 3)(x_4^2 + 1) &= 0 \\ (x_4^2 - 4)(x_3^2 + 1) &= 0 \end{cases} \tag{1.1.2}$$

We can reduce this 4D system of equations by searching for a solution in the subspace W spanned by $\Phi = \{(2, 4, -2, 0)^T; (0, 0, 0, -4)^T\}$.

$$\begin{cases} (16y_1^2 + 1)(4y_1^2 + 2y_1 - 2) &= 0 \\ (4y_1^2 + 1)(16y_1^2 + 20y_1 + 6) &= 0 \\ (16y_2^2 + 1)(4y_1^2 + 4y_1 - 3) &= 0 \\ (4y_1^2 + 1)(16y_2^2 - 4) &= 0 \end{cases} \tag{1.1.3}$$

This new reduced system (size 2) has two solutions $Y_1 = (0.5, 0.5)^T$ *and* $Y_2 = (0.5, -0.5)^T$. *We can obtain the solutions of (Equation* (1.1.2)*) by plugging* Y_1 *and*

Y_2 *into* $\Phi \cdot Y = X$. *Note that the original nonlinear system (Equation* (1.1.2)*) has 16 solutions, but the only two contained in* W *are obtained by solving the Reduced Model. To overcome the issue of not obtain all the solutions, we consider as future work to take an union of subspace instead of only one subspace to be able to find all the solutions of a system of equations.*

There are many ways to find an appropriate basis Φ. Among others, we can mention reduction based on Krylov methods, in particular, Arnoldi methods (see [10]), wavelet-based methods (see [37]), model reduction based on spectral projection (see [15]), and Proper Orthogonal Decomposition method (see [18, 69, 61]), which is based on principal component analysis.

In next sections, we will present a brief introduction to the methods mentioned above.

1.2 Krylov Methods

Given an $(n \times n)$ matrix A, recall that the characteristic polynomial of A is $p(t) = \det(A - tI)$, where I is the $(n \times n)$ identity matrix [19, 103], 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; and
3. No polynomial which annihilates A has a smaller degree than $q(t)$.

We are sure that such polynomial exists because at least $q(t) = p(t)$, and the polynomial

$p(t)$ annihilates A [53, 67, 78].

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

$$q(t) = \sum_{i=0}^{m} \alpha_i t^i. \tag{1.2.1}$$

If A is nonsingular, then $\alpha_0 \neq 0$ [53], and since $q(t)$ annihilates A, we have

$$\begin{aligned} 0 &= \sum_{i=0}^{m} \alpha_i A^i \\ &= \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \cdots + \alpha_m A^m; \end{aligned}$$

we can solve for I :

$$I = \frac{1}{\alpha_0}\left(-\alpha_1 A - \alpha_2 A^2 - \cdots - \alpha_m A^m\right).$$

We left-multiply both sides by A^{-1} :

$$A^{-1} = \frac{1}{\alpha_0}\left(-\alpha_1 I - \alpha_2 A - \cdots - \alpha_m A^{m-1}\right).$$

Finally, we right-multiply both sides by b :

$$A^{-1}b = \frac{1}{\alpha_0}\left(-\alpha_1 b - \alpha_2 Ab - \cdots - \alpha_m A^{m-1}b\right).$$

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*, whose dimension is the degree of the minimal polynomial of A. Therefore, if the degree of the minimal polynomial is less than the dimension of the space, then the convergence of the numerical method is faster.

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

There exist several versions of Krylov methods, which we can mention: Arnoldis method, Conjugate Gradient, among others [21, 95]. In particular, in this disseration, we use the

generalized minimal residual method (GMRES) that was published by Saad and Schultz in 1986 [95].

1.3 Wavelets-based Approach (DWT)

Given a finite wave, Ψ, called Wavelet Mother, the wavelet transform is a special kind of mathematical transform which analyzes a function $f(t) \in L^2(\mathbb{R})$ expresses it in terms of versions of dilatations and translations of the Wavelet Mother:

$$\Psi_{m,n}(t) = a_0^{-m/2}\Psi(a_0^{-m}(t - nb_0a_0^m)) \tag{1.3.1}$$

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

$$\Psi(t) = \begin{cases} 1 & \text{if} \quad 0 \leq t < \frac{1}{2} \\ -1 & \text{if} \quad \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{1.3.2}$$

Haar in (1910) [102] proved that Equation (1.3.1) provides 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 [51]:

$$W = \begin{bmatrix} L \\ H \end{bmatrix}. \tag{1.3.3}$$

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, Equation (1.3.3) of the wavelet, Ψ in Equation (1.3.2) is:

$$W = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} \tag{1.3.4}$$

In general, the matrix W is an orthonormal matrix. The matrix in Equation (1.3.4) that we used as example, therefore it can prove that:

$$LL^T = I_{\frac{n}{2}\times\frac{n}{2}}, \tag{1.3.5}$$

$$HH^T = I_{\frac{n}{2}\times\frac{n}{2}}. \tag{1.3.6}$$

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

$$||x||^2 = ||Wx||^2 \approx ||Lx||^2. \tag{1.3.7}$$

We can observe this property in Figure 1.2.

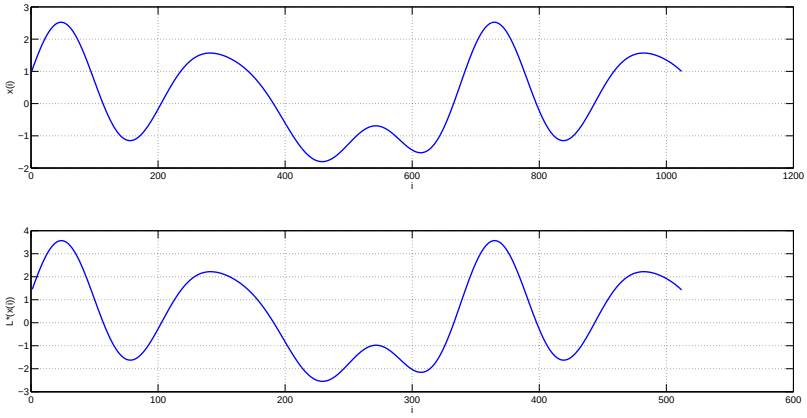


Figure 1.2: Comparison of the norm of a vector ($||x||$) and the norm of the action of a low-pass filter on it ($||L(x)||$).

1.3.1 How to use Wavelets to Reduce a System of Equations

Given a linear system of equations

$$Ax = b, \tag{1.3.8}$$

we can take advantage of the property Equation (1.3.7) to reduce the size of system of equations by using the low-pass block of a wavelets.

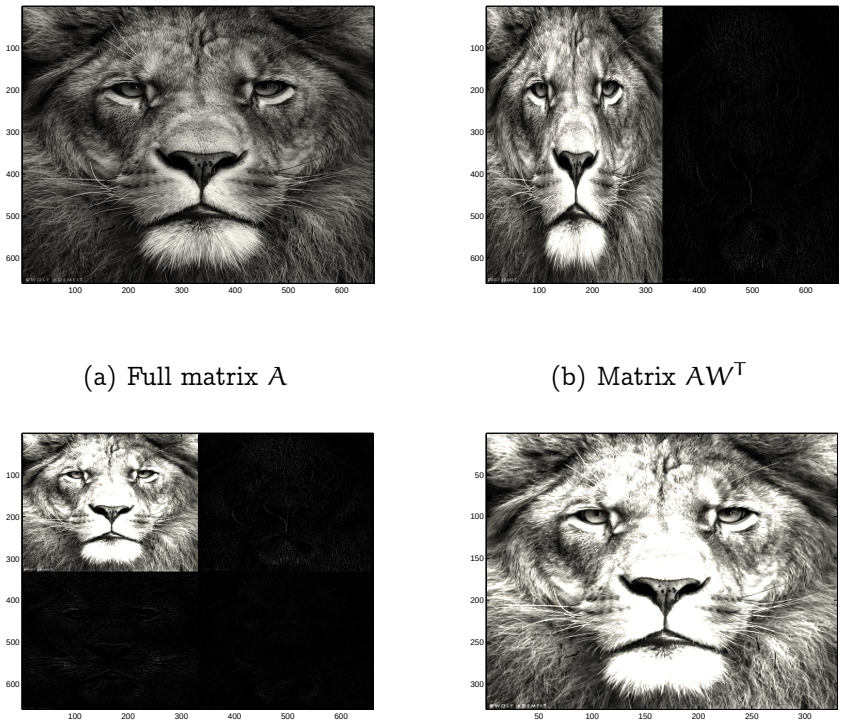
Consider the basis $\Phi = L^T$. The system Equation (1.3.8) is reduced

$$(\Phi^T A \Phi)\bar{x} = \Phi^T b. \tag{1.3.9}$$

Then, once $\bar{x}$ is found, we can obtain x using $x = \Phi\bar{x}$.

We can see how the wavelet and the low-pass block act over the matrix A of Equation (1.3.9) in Figure 1.3.

Assume that the matrix A is represented by Figure 1.3(a). We can see in Figure 1.3(b)



(a) Full matrix A

(b) Matrix AW^{T}

(c) Matrix WAW^{T}

(d) Reduced Matrix $\Phi^{\mathsf{T}}A\Phi$

Figure 1.3: Illustration of how wavelets can be used to reduce a linear system of equations.

the visualization of the multiplication AW^{T}. We observe that AW^{T} consists of two blocks, and one of them is black, which means its elements are close to zero. We can see in the third image Figure 1.3(c) the product matrix WAW^{T}. This matrix consist of 4 blocks and only the elements corresponding to $\Phi^{\mathsf{T}}A\Phi$ are significantly greater than zero Figure 1.3(d).

We can conclude this section observing that wavelets can be used to reduce the associate matrix of a system of equations to a smaller matrix, but this reduction is just 50% of the original system. If we want to reduce the system even more, we need to apply the same procedure to the matrix represented by the inmage in Figure 1.3(d), but in this case, we

will lose more accuracy. We will see in Section 1.6 that this method will not be useful when we require a good approximation of our solution.

1.4 Proper Orthogonal Decomposition (POD).

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

$$A(\lambda)x = b. \tag{1.4.1}$$

Here, we can see Equation (1.4.1) as an input-output system, where λ is the input and the solution, $x(\lambda) \in \mathbb{R}^n$, is the output of it.

The idea behind this method is that, given a certain input, the solution $x(\lambda)$ of a system contains the behavior of the system [98]. 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.

Consider the set of snapshots S. The solution $x(\lambda^*)$ of Equation (1.4.1) for a particular λ^* is a particular snapshot, and therefore, is in the subspace spanned by S. If the columns of S are highly correlated, then we can apply principal components analysis (PCA) to obtain an uncorrelated number of columns, and thus to reduce the size of linear system of equations.

Consider the singular value decomposition (SVD) of S

$$S = U\Sigma V^T \tag{1.4.2}$$

We can split Equation (1.4.2) into two summands

$$S = U\Sigma V^T \tag{1.4.3}$$

$$= \sum_{i=1}^{k} u_i \sigma_i v_i^T + \sum_{i=k+1}^{n} u_i \sigma_i v_i^T. \tag{1.4.4}$$

If there exists k, such that $\sigma_i \approx 0$ for all $i > k$, then the matrix of snapshots can be approximated by

$$S \approx \sum_{i=1}^{k} u_i \sigma_i v_i^T, \tag{1.4.5}$$

and we can consider the matrix Φ consisting of the first k columns of U, i.e.,

$$\Phi_{*,i} = U_{*,i}, \forall i \in \{1, 2, \ldots, k\} \tag{1.4.6}$$

The reader who wants to find more information about Reduced-Order Modeling can find in [117, 79] a good source of information.

1.5 Using Reduced-Order Modeling for Nonlinear Systems

Consider a nonlinear system of equations

$$F(x) = 0, \tag{1.5.1}$$

where F is a nonlinear function $F : \mathbb{R}^n \to \mathbb{R}^n$. This nonlinear system of equations could be the result of the discretization of a ordinary or partial differential equation.

Algorithm 1 Newton Method Algorithm

Require: A function $F : \mathbb{R}^n \to \mathbb{R}^n$, the Jacobian Matrix (J) of F, an initial point x, the tolerance ε, and the maximum number of iterations $Maxiter$

1: **for** $i = 1$ **to** $i = Maxiter$ **do**
2: Solve: $J(x)\Delta x = -F(X)$
3: Update x: $x = x + \Delta x$
4: **if** $||F(x)|| \leq \varepsilon$ or $i = Maxiter$ **then**
5: break
6: **end if**
7: **end for**
8: **return** x

A widely used method to solve Equation (1.5.1) is the Newton Method, whose algorithm is shown in **Algorithm 1**.

If a solution, x, of Equation (1.5.1) lies on a k-dimensional subspace W, whose basis is given by the matrix Φ, then

$$x = \Phi p \tag{1.5.2}$$

where $p \in \mathbb{R}^k$.

The system Equation (1.5.1) can be reduced as follows

$$F(\Phi p) = 0, \tag{1.5.3}$$

the Jacobian matrix of $F(\Phi p)$ is $J(\Phi p)\Phi$. In **Algorithm 2**, we show how the Newton method is modified.

Algorithm 2 Reduced Newton Method Algorithm

Require: A function $F : \mathbb{R}^n \to \mathbb{R}^n$, the Jacobian Matrix (J) of F, a basis Φ, an initial point p, the tolerance ε, and the maximum number of iterations $Maxiter$

1: **for** $i = 1$ **to** $i = Maxiter$ **do**
2: Solve: $J(\Phi)\Delta p = -F(\Phi p)$
3: Update p: $p = p + \Delta p$
4: **if** $\|F(\Phi p)\| \leq \varepsilon$ or $i = Maxiter$ **then**
5: break
6: **end if**
7: **end for**
8: **return** $x = \Phi p$

1.6 Examples

In this section, four examples are solved using each approach studied in the previous section. The first example is a set of linear system of equations. The second is a nonlinear system of equations. 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 compared all results with the FOM solution in terms of computation time and relative error.

1.6.1 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 Figure 1.4(a) and the matrix $A(1)$ is represented by Figure 1.4(e). 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:

$$A(\lambda)x = b(\lambda), \tag{1.6.1}$$

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

$$x(\lambda) = \sin\left(6\sqrt{t^2+\lambda^2}\right) \tag{1.6.2}$$

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

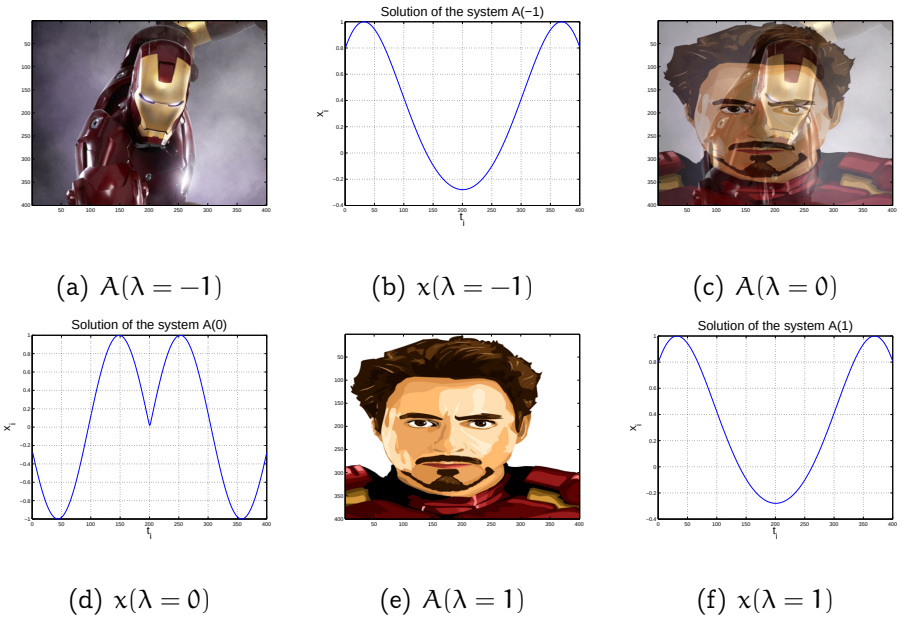
$$x_i(\lambda) = \sin\left(6\sqrt{t_i^2+\lambda^2}\right) \tag{1.6.3}$$

Where $t_i = -1 + hi$ with $i \in \{0,1,\ldots,399\}$. In Figure 1.4, we present three members of the parametric family $A(\lambda)$ for $\lambda \in \{-1,0,1\}$ Figures 1.4(a), 1.4(c) and 1.4(e), and their respective solutions Figures 1.4(b), 1.4(d) and 1.4(f).

In Figure 1.5, we plot in a same graph the solutions corresponding to $\lambda \in [-1,1]$.

The problem was solved using MatLab, R2017a(9.2.0.556344) on a MacBook Pro, processor 2.9 GHz Intel Core i7, 16 GB 1600 MHz DDR3 and we obtain the following results:

1. **Full Order Model (FOM):** The solution (Figure 1.6(a)) of Equation (1.6.1) is obtained in 2.83 s. with a relative error $\varepsilon = 4.1236 \times 10^{-09}$.



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

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

Figure 1.4: Three members of the family of the linear systems are shown.

2. **Krylov Method:** We use `gmres` (Generalized Minimum Residual Method), which is used when the associate matrix of the system is non-symmetric. The approximation of the solution is obtained in 2.40 s. with a relative error of 0.42. In Figure 1.6(b), we show the approximation of the solution. Observe that for this example Krylov method is not a good choice.

3. **Wavelet Method:** In this method, we use the second level of the low pass of the wavelet named `db1` (Daubechies Wavelet) as the reduced basis, and the solution is obtained in 2.49 secs with a relative error of 0.0257.

4. **Proper Orthogonal Decomposition (POD) Method:** We compute the singular value decomposition of the set of snapshots, which consists of the solution of the FOM. Then, we select the first five columns of the matrix U as the reduced basis. These five columns represent the 99.85% of the total variance. Using this method, we can find the solution (Figure 1.6(d)) in 0.37 s. with a relative error is 0.0021.

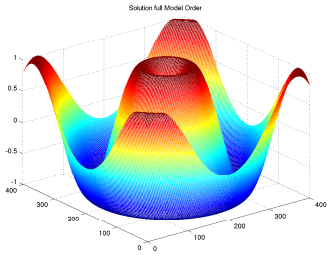


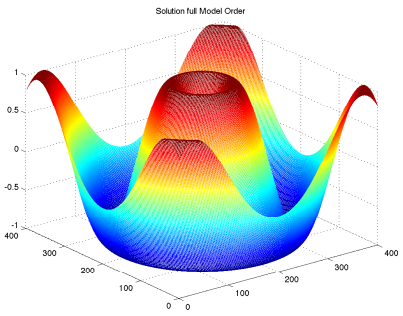
Figure 1.5: Solution of the $A(\lambda)x = b(\lambda)$, for all $\lambda \in [-1, 1]$ is depicted.

In the previous example, we observed that the Krylov method is not a option to reduce our problem. The best option is the POD method for two reasons: on one hand the dimension of the subspace is the smallest, and on the other hand, it finds the solution faster than the other approaches.

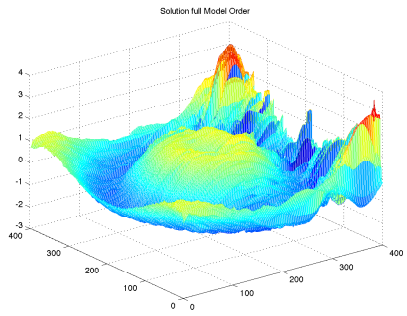
Table 1.1: Comparison of FOM, Krylov, Wavelets, and POD methods when solving a parametric linear system of equations

	FOM	ROM Krylov	ROM Wavelet	ROM POD
Time	1.83 s.	240 s.	2.49 s.	0.37 s.
Relative Error	4.1236×10^{-9}	0.42	0.0257	0.0021
Dimension of the subspace	400	100	100	5

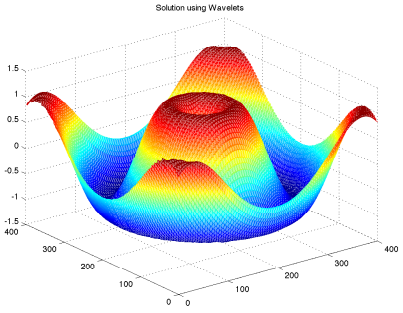
We summarize the results of this example in Table 1.1.



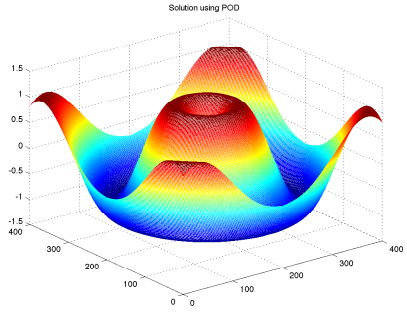
(a) FOM Solution



(b) ROM Solution Using Krylov Method



(c) ROM Solution Using Wavelets



(d) ROM Solution Using POD

Figure 1.6: Solution of a parametric linear system of equations using full-order model, Krylov, Wavelet, and POD method.

1.6.2 Nonlinear System of Equations

In this section, we aim to solve the nonlinear system of equations proposed by Yamamura and Kawata [118, 29], defined by:

$$F : \mathbb{R}^n \to \mathbb{R}^n,\ x \mapsto (F_i(x))_{1\le i\le n},\ \text{where}$$
$$F_i = 2.5x_i^3 - 10.5x_i^2 + 11.8x_i - i + \sum_{i=1}^{i=n} x_i = 0, \quad 0 \le i \le n. \tag{1.6.4}$$

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 1.5.

We use the different ROM techniques studied before to solve Equation (1.6.4) and to determinate which of them is the best with respect to runtime and accurate of the approximation.

1. **Full Order Model (FOM):** The initial x_0 is selected in the n cartesian product $[-10^8, 10^8] \times \left[-10^8, 10^8\right]^n$, and the stopping criterion is $\varepsilon = 10^{-5}$. The method converges in 459.74 s. in 814 iterations with $\|F(x)\| < 2.81 \times 10^{-10}$.

2. **Krylov Method:** When Newton method is applied to solve a nonlinear system the equation, we have to solve the linear system of equations

$$J(x)\Delta x = -F(x) \tag{1.6.5}$$

as many times as the number of iteration Newton method takes to converge. We solved the linear system Equation (1.6.5) using `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)** [84]. 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 this conditions, the solution is obtained in 108 iterations in 50.64 s with $\|F(x)\| < 10^{-6}$ in the solution, and the relative error compared

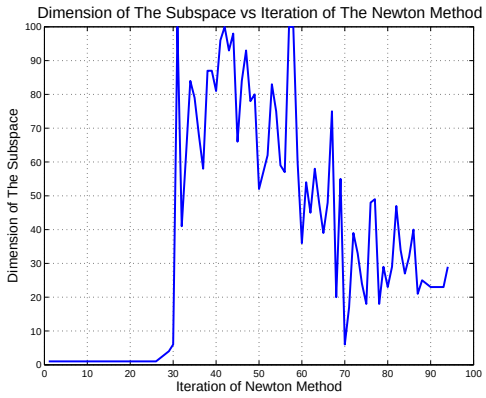


Figure 1.7: Dimension of the subspace `minres` in each iteration of Newton Method when solving Yamamura problem.

with the solution obtained with **FOM** is $\varepsilon_r = 0.0117$. The method `minres` is used to solve Equation (1.6.5) finding a subspace where Δx lies. In Figure 1.7, we can observe the dimension of the subsapce for each 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, with wavelet method, we assume that the solution is either in a fixed subspace or close to it with respect to the orthogonal projection. Therefore, we search for Δx on the same subspace.

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

4. **Proper Orthogonal Decomposition (POD):** In order to compute the snapshots and to find the reduced basis, we m:

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

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

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

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

Later the singular value decomposition of Sna is computed

$$Sna = USV^T,$$

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:

Table 1.2: Comparison of FOM, Krylov, Wavelets, and POD methods when solving a parametric nonlinear system of equations (Yamamura problem).

	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	$\{5, 6, \ldots, 100\}$	250	100

1.6.3 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

$$\frac{\partial U}{\partial t} = \lambda \frac{\partial^2 U}{\partial x^2}, \tag{1.6.6}$$

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:

$$\frac{\partial U}{\partial t} = \frac{U(t_i, x_j) - U(t_{i-1}, x_j)}{\Delta t}, \tag{1.6.7}$$

$$\frac{\partial^2 U}{\partial x^2} = \frac{U(t_i, x_{j-1}) - 2U(t_i, x_j) + U(t_i, x_{j+1})}{h^2}, \tag{1.6.8}$$

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 1.6.7 and 1.6.8 in 1.6.6 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

$$-\lambda \Delta t U_{i,j-1} + (2\lambda \Delta t + h^2) U_{i,j} - \lambda \Delta t U_{i,j+1} = h^2 U_{i-1,j}. \tag{1.6.9}$$

The linear system of equations Equation (1.6.9) is solved for each $1 \leq i \leq 400$ using **FOM** and the approximation of the solution is given in 0.572 secs.

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 Figure 1.8 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 $\varepsilon_r = 0.68$.

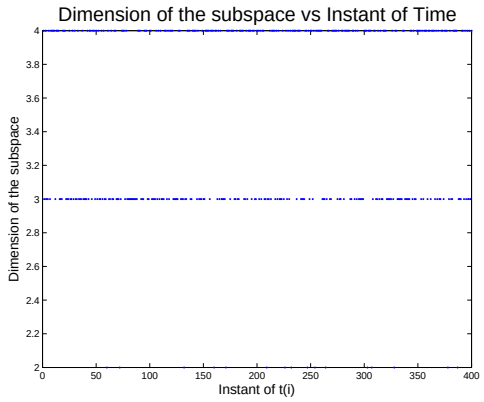


Figure 1.8: Dimension of the subspace in each t-stepsize in the Heat equation.

4. **Proper Orthogonal Decomposition** The parameter λ in Equation (1.6.6) is used as input. 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, $\text{Snap} = USV^T$ 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:

The graphs of the solution by using different methods are shown in Figure 1.9.

Table 1.3: Comparison of FOM, Krylov, Wavelets, and POD methods when solving a linear partial differential equation (heat equation).

	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, 3, 4}	200	2

1.6.4 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

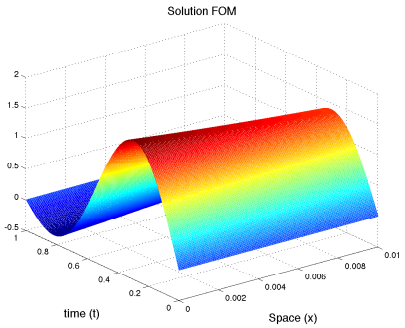
$$\frac{\partial U(t,x)}{\partial t} + \frac{\partial f(U(t,x))}{\partial x} = g(x), \tag{1.6.10}$$

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 $t > 0$.

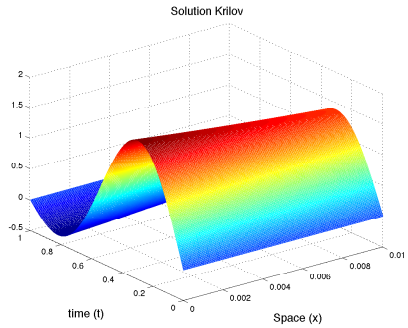
Discretizing $U(t,x)$ with respect to the variable x yields

$$\underline{U}(t) = [U_1(t), U_2(t), \cdots, U_n(t)] \tag{1.6.11}$$

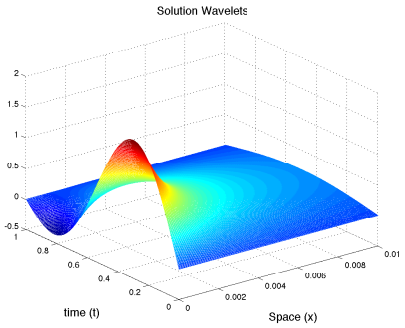
where $U_i(t) = U(t,x_i)$, and $x_i = x_{i-1} + \Delta x$. Applying Godunov's scheme, also known as finite-volumen method, to approximate $\partial/\partial x$ [90], the dynamical system takes the



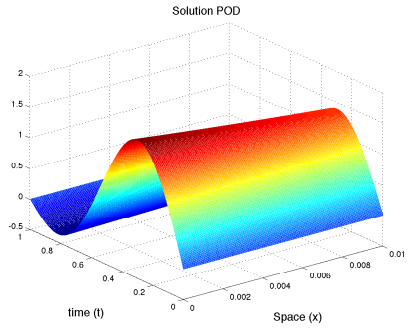
(a) FOM Solution



(b) ROM Solution Using Krylov Method



(c) ROM Solution Using Wavelets



(d) ROM Solution Using POD

Figure 1.9: Solution of the Heat Equation.

following form:

$$\frac{d\underline{U}}{dt} = F(\underline{U}) + G + Bu^2. \tag{1.6.12}$$

where:

$$F(\underline{U}) = \begin{bmatrix} -0.5U_1^2 \\ 0.5(U_1^2 - U_2^2) \\ \vdots \\ 0.5(U_{n-1}^2 - U_n^2) \end{bmatrix}, \quad G = 0.02 \begin{bmatrix} \exp(0.02x_1) \\ \exp(0.02x_2) \\ \vdots \\ \exp(0.02x_n) \end{bmatrix}, \quad B = \begin{bmatrix} \frac{1}{2\Delta x} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{1.6.13}$$

Now, discretizing Equation (1.6.12) with respect to the variable t:

$$\frac{\underline{U}(t_i) - \underline{U}(t_{i-1})}{\Delta t} = F(\underline{U}) + G + Bu^2, \tag{1.6.14}$$

where $\underline{U}(t_0) = U^0$, is the initial condition, fixing the value of t_i, Equation (1.6.14) is equivalent to solve the non-linear system

$$R(\underline{U}) = \begin{pmatrix} (-0.5U_1(t_i)^2 + 0.02e^{0.02} + \frac{16}{2})\Delta t + U_1(t_i) - U_1(t_{i-1}) \\ (0.5(U_1(t_i)^2 - U_2(t_i)^2) + 0.02e^{0.02\cdot 2})\Delta t + U_2(t_i) - U_2(t_{i-1}) \\ \vdots \\ (0.5(U_{i-1}(t_i^2 - U_i(t_i^2) + 0.02e^{0.02\cdot i})\Delta t + U_i(t_i) - U_i(t_{i-1}) \\ \vdots \\ (0.5(U_{99}(t_i^2 - U_{100}(t_i^2) + 0.02e^{0.02\cdot 100})\Delta t + U_{100}(t_i) - U_{100}(t_{i-1}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{1.6.15}$$

and Equation (1.6.15) can be solved using techniques to solve nonlinear system of equations.

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

$$J = \begin{pmatrix} -U_1(t_i -)1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ U_1(t_i) & -U_2(t_i) - 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & U_2(t_i) & -U_3(t_i) - 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & U_3(t_i) & -U_4(t_i) - 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & U_{99}(t_i) & -U_{100}(t_i) - 1 \end{pmatrix}. \tag{1.6.16}$$

After we discretize and solve the problem using Newton Method for each time step and each approach, the results are presented in Section 1.6.4 and the graphs of the solution are presented in Figure 1.10.

Table 1.4: Comparison of FOM, Krylov, Wavelets, and POD methods when solving a nonlinear partial differential equation (Burgers equation).

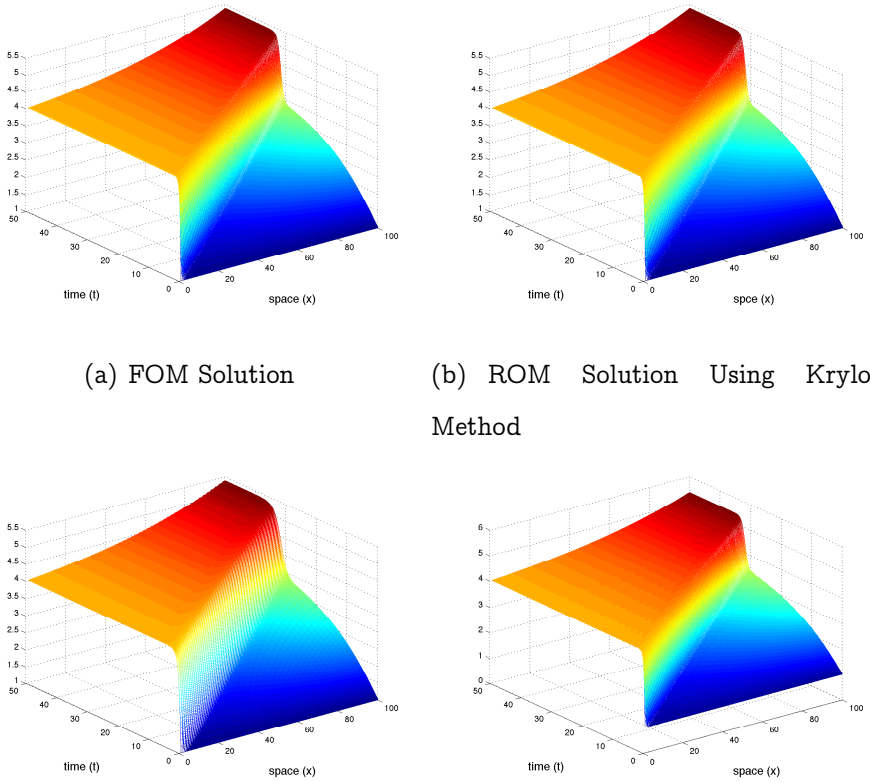
	FOM	ROM krylov	ROM Wavelet	ROM POD
Time	0.41 *sec.*	2.76 *sec.*	0.77 *sec.*	0.67 *sec.*
Rel. Error	–	3.85E-11	0.0076	9.013E-4
Dimension	100	7	50	32

1.7 Conclusion

We have solved four classical problems: a parametric linear system of equations, a parametric nonlinear system of equations, a linear partial differential of equations, and a nonlinear partial differential of equations using four strategies: full order model, rom using Krylov method, wavelets, and POD. We realized that, except in the case Burgers equation, the best method to solve a problem using ROM is POD in terms of dimension and relative error.

In the case of Burger equation, Krylov method is which gives better results. It can be explained because in each iteration, the Jacobian matrix is a sparse bidiagonal matrix, and the for that reason the convergence is fast. When other methods are used in this case the property of sparsity is lost, and as consequence, the dimension of the subspace is larger and, therefore the time of convergence is large as well.

Finally, As we expected, we observed that ROM based on wavelets was which had the worst performance in terms of reducing the dimension of the subspace and runtime due to it get rid of lower frequency data which can be important in the approximation to the



(a) FOM Solution

(b) ROM Solution Using Krylov Method

(c) ROM Solution Using Wavelets

(d) ROM Solution Using POD

Figure 1.10: Solution of the Burger Equation.

solution.

the reader who wants to have more information about how to obtain a reduced basis on a nonlinear problems can see [98, 90, 26, 117].

Chapter 2

Interval Arithmetic and Interval Constraint Solving Techniques

---

Problems are hidden opportunities, and constraints can actually boost creativity.

Martin Villeneuve

In this chapter, we present interval arithmetic to extend real function evaluation to interval evaluation to be able to obtain with certainty an envelope of the exact answer to mathematical problems. Algorithms based on interval computations can provide bounds on a sequence of operations due to the way how the operations are executed.

2.1 Motivation

This section shows how interval computations can be used to conduct reliable computation.

2.1.1 A little bit of history

Two very well-known real numbers are $\sqrt{2}$ and π. Both numbers are irrationals, which means that none of them can be expressed as the ratio of two integers.

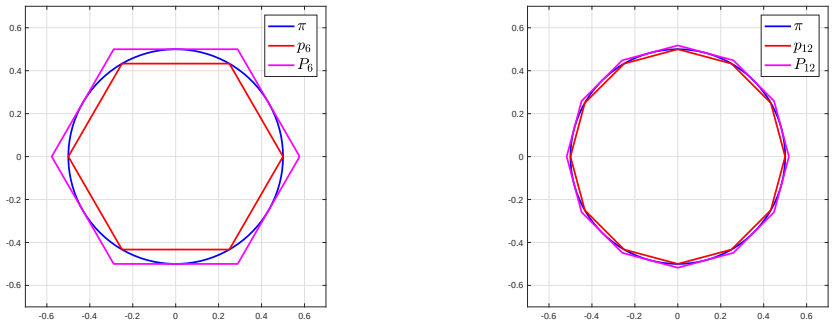
Figure 2.1: Archimedes of Syracuse.

Although both numbers are irrationals, there exists a significantly difference between them. On one hand, we can build a segment whose length is $\sqrt{2}$; we only have to take the hypotenuse of an isosceles triangle whose side is one. Geometrically speaking, $\sqrt{2}$ can be built using straightedge and compass, and algebraically speaking, $\sqrt{2}$ is a root of a polynomial with integer coefficients. All numbers that are root of integer-coefficient polynomial are called **algebraic numbers** [9, 47, 64].

On the other hand, the number π is not an algebraic number. These kind of numbers are called **transcendent number**. The question is: how was it possible to have a reliable approximation to π? The answer of that question was given by Archimedes of Syracuse (287 - 212 BC), Figure 2.1 with his **exhaustion method**.

Let us recall how π is defined: The number π is the ratio of the circumference of a circle and its diameter. In particular, if the diameter of the circle is one, then the length of circumference is π.

The Archimedes' exhaustion method consists in approximate the circumference of a circle from above and below by circumscribing and inscribing regular polygons [62, 16], see: Figure 2.2.



(a) Approximation of π using two hexagon. (b) Approximation of π using two 12-sided polygon.

Figure 2.2: Bounding π using Archimedes' exhaustion method.

Consider p_k and P_k the regular perimeter of k sides that are inscribed and circumscribed in a circle of diameter one. Archimedes proved that

$$P_{2n} = \frac{2p_n P_n}{p_n + P_n}, \qquad p_{2n} = \sqrt{p_n P_{2n}} \tag{2.1.1}$$

Archimedes found an approximation of π using p_{96} and P_{96} in Equation (2.1.1) and rational approximations for radicals. He concluded that

$$3\frac{10}{71} < \pi < 3\frac{1}{7} \tag{2.1.2}$$

Archimedes used no trigonometry. The last approximation using this method is attributed to Grienberger in 1630 using Snell's refinement [36].

We can use similar ideas of Archimedes' exhaustion method to compute interval enclosures for the exact values of integrals.

2.2 Riemann Integral

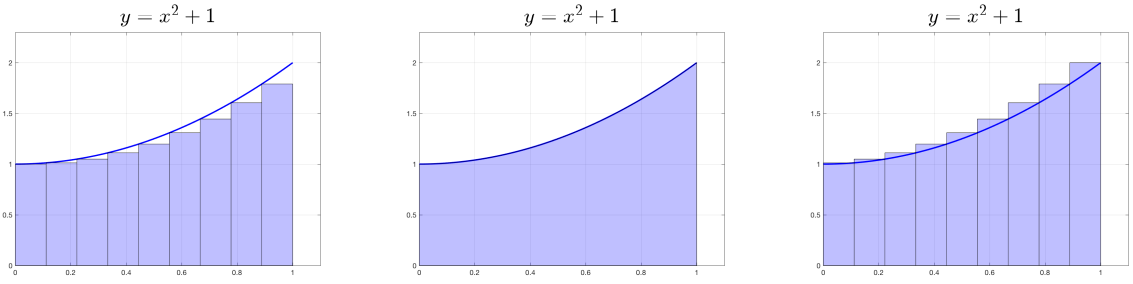
Let f be a real-valued function defined on the interval $[a, b]$. The lower (see: Figure 2.3(a)) and upper (see: (Figure 2.3(c))) sum of f with respect to a partition $P : a = x_0 < x_1 < \cdots < x_n = b$ is defined as follows:

$$\begin{aligned} L(f, P) &= \sum_{i=0}^{n-1} \inf_{t \in [x_i, x_{i+1}]} f(t)(x_{i+1} - x_i) \\ U(f, P) &= \sum_{i=0}^{n-1} \sup_{t \in [x_i, x_{i+1}]} f(t)(x_{i+1} - x_i) \end{aligned} \tag{2.2.1}$$

If f is continuous, then

$$L(f, P) < \int_a^b f(x)dx < U(f, P) \tag{2.2.2}$$

The Riemann integral (see: Figure 2.3(b)) is the limit of the Riemann sums of a function as the partitions get finer [107, 5].



(a) Lower sum. (b) Riemman integral. (c) Upper sum.

Figure 2.3: Lower and Upper approximation to the Riemann integral.

We consider in Figure 2.3 an increasing function as illustrative purposes.

2.3 Computations with Intervals

Uncertainty in a given quantity can be expressed as an interval of possible values for this quantity. Measurement instruments often come with a known maximum error. As a result, when measuring a quantity x, we often instead get $\tilde{x}$, which in fact corresponds to $\tilde{x} \pm \Delta_x = [\tilde{x} - \Delta_x, \tilde{x} + \Delta_x]$, in which the true observed quantity lies. In what follows, we start by showing how we can handle uncertainty with intervals.

In a nutshell, interval analysis, as opposed to real analysis, consists in handling every quantity as a closed interval, as opposed to a real number. The idea of interval arithmetic is as follows:

$$\forall X, Y \text{ intervals}, \quad X \bowtie Y = \{x \bowtie y,\ x \in X,\ y \in Y\},$$
$$\text{where } \bowtie \in \{+, -, \times, \div\}$$

More specifically, since Y might contain 0, when $\bowtie$ is $\div$, $X \bowtie Y$ would not be an interval unless we add an extra operator, $\square$, the hull, to ensure that all interval computations result in an interval. As a result, we in fact have:

$$\forall X, Y \text{ intervals}, \quad X \bowtie Y = \square\{x \bowtie y,\ x \in X,\ y \in Y\},$$
$$\text{where } \bowtie \in \{+, -, \times, \div\}$$

From there, all functions over the reals can be extended to intervals. For any function $f : \mathbb{R}^n \to \mathbb{R}$ over the reals, a valid interval extension $\tilde{F} : \mathbb{I}^n \to \mathbb{I}$ satisfies:

$$\forall X \in \mathbb{R}^n,\ \{f(x),\ x \in X\} \subset \tilde{F}(X).$$

This definition allows for many interval functions to qualify as extensions of any given real function f but the goal is to identify interval functions $\tilde{F}$ that best enclose the range of the original real function f. As a result, when solving nonlinear systems of equations using interval computations, we seek to use extensions that are tight, so as to avoid overestimations as much as possible.

Indeed, in general, different interval extensions of the same real function f yield different enclosures [56]. For instance, Volker Stahl demonstrated in [100] that the Horner extension [28, 54] reduces the overestimation on polynomial evaluation; see Figure 2.4.

Example 2. *Consider two different symbolic expressions, called* f *and* g*:*

$$f(x) = 2x^5 + x^3 - 3x^2 \tag{2.3.1}$$

$$g(x) = x^2(-3 + x(1 + 2x^2)) \tag{2.3.2}$$

These two expressions in fact represent the same real function: $\forall x \in \mathbb{R}$, $f(x) = g(x)$. *However, when extending these symbolic expressions to being evaluated on intervals, the resulting enclosures are very different; see Figure 2.4.*

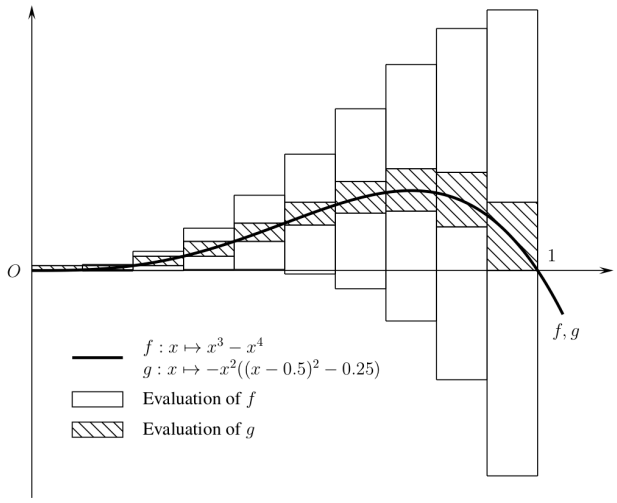


Figure 2.4: Comparison of the natural and Horner expressions of the same real polynomial f.

So, although techniques based on interval computations yield reliable results, they are subject to overestimation. These are usually caused by one of the following problems (see [80] for more details):

1. Dependency problem: when variables appear more than once in an equation, each occurrence is handled as a separate variable, hence possibly resulting in overestimation, which compounds over repeated computations. For example, consider the function:

$$f(x) = x^2 - x \tag{2.3.3}$$

if we evaluate Equation (2.3.3) in $\mathbf{x} = [0, 1]$, we obtain

$$\begin{aligned} f([0,1]) &= [0,1]^2 - [0,1] \\ &= [0,1] - [0,1] \\ &= [-1,1], \end{aligned}$$

which is an overestimation of the real range.

On the other hand, if we re-write Equation (2.3.3) as follows

$$f(x) = \left(x - \frac{1}{2}\right)^2 - \frac{1}{4}, \tag{2.3.4}$$

and we evaluate in the same interval. We obtain

$$\begin{aligned} f([0,1]) &= \left([0,1] - \frac{1}{2}\right)^2 - \frac{1}{4}; \\ &= \left[-\frac{1}{2}, \frac{1}{2}\right]^2 - \frac{1}{4}; \\ &= \left[0, \frac{1}{4}\right] - \frac{1}{4}; \\ &= \left[-\frac{1}{4}, 0\right], \end{aligned}$$

which is the exact range.

2. Wrapping effect: as intervals are cartesian boxes, they are not suited for the approximation of other shapes (other than right shapes). As a result, uncertainty is generated by the cartesian approximation of such shapes and only grows as the computations repeat. This effect is illustrated in Figure 2.5.

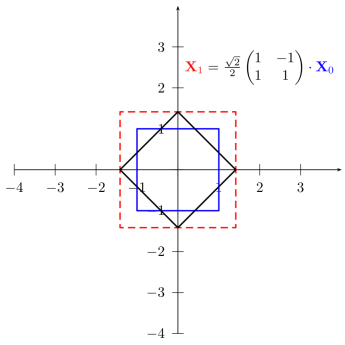


Figure 2.5: Illustration of the wrapping effect of interval computations.

Note that, in practice, we conduct interval computations with floating-point-bounded intervals. What this changes is that each time an interval computation is carried out, if the bounds of the resulting interval are not floating points, they are outward rounded (to the closest outward floating point), to guarantee that the range of the interval computation still be enclosed. For example the rational number $x = 0.1$ has no binary representation, so the interval $[0.1, 0.1]$ is approximated as:

$$[0.1, 0.1] \approx [0.09999999999999999, 0.1000000000000001]. \tag{2.3.5}$$

In this work, we use interval computations provided in RealPaver [14, 43, 42, 113] and the natural extensions that this library provides.

2.4 How to Solve Nonlinear Equations with Intervals?

When solving equations with intervals, we use constraint solving techniques [56, 96], namely contractors on each equation as follows. In short, given an equation $f_i(x_1, \ldots, x_n) = 0$ with k occurrences of x_j in the symbolic expression of f_i, a contractor on f_i for x_j is defined by:

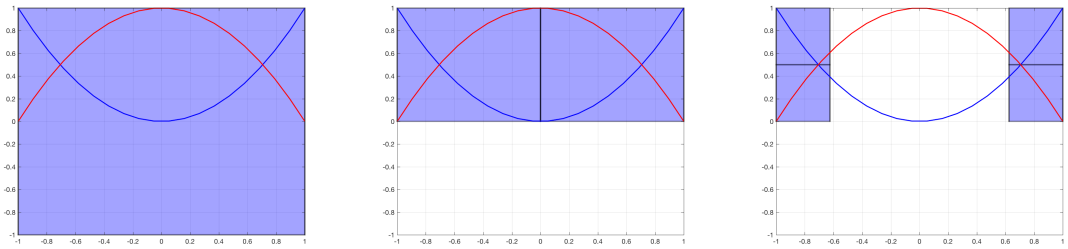
$$c_{i,j,l} : \mathbb{IR} \to \mathbb{IR} \text{ where: } l \in \{1, \ldots, k\}$$
$$X \mapsto c_{i,j,l}(X) = X \cap F_i, j, l(X_1, \ldots, X_n)$$

where F_i, j, l is such that $F_i, j, l(x_1, \ldots, x_n) = x_k$ is equivalent to $f_i(x_1, \ldots, x_n) = 0$. An example of this is as follows.

Example 3.

$$\begin{cases} c_1 : & x^2 + y & = & 1 \\ c_2 : & -x^2 + y & = & 0 \end{cases} \tag{2.4.1}$$

where $x \in X = [-1, 1]$ *and* $y \in Y = [-1, 1]$.



(a) Initial Box (b) Two Outer Boxes (c) Four Outer Boxes

Figure 2.6: Outer approximations of the solutions of the nonlinear system of equations.

Let us explain, step by step, how the domains of each variable, x and y, are narrowed (see [45, 46] for details):

1. First, we start with the initial box, $[-1,1] \times [-1,1]$, containing the solution(s) of this problem, if any; see Figure 2.6(a).

2. Constraint c_1 is considered and first used to define a contractor for the domain of variable y:

$$c_{1,y,1} : \; Y \mapsto Y \cap (1 - X^2)$$

   As a result, we obtain:

$$\begin{aligned} Y &= [-1,1] \cap (1 - [-1,1]^2) \\ &= [-1,1] \cap (1 - [0,1]) \\ &= [-1,1] \cap [0,1] \\ &= [0,1] \end{aligned}$$

3. A similar step is repeated on the contractor for variable x using Constraint c_1: $c_{1,x,1}$, but this step yields not contraction for the domain of x, leaving it as $X = [0,1]$.

4. Moving on to Constraint c_2, and processing the domains of x and y using their respective contractors, no more contraction is obtained. The new domain after seeking a first fixed-point of contraction on the original domains yields a new domain: $[-1,1] \times [0,1]$.

5. If the accuracy of this new domain is deemed insufficient (which it is, in this case), then the domain is split along one of its dimension (in Figure 2.6(b), it is split w.r.t. the domain of x), and the above steps 2–4 are repeated until the whole domain has been discarded (no solution) or a desired level of accuracy has been reached.

6. Repeating this process further, we reach configuration Figure 2.6(c) in Figure 2.6.

Unlike real-valued traditional non-interval Newton methods, Interval Constraint Solving Techniques do not involve or depend on the choice of the method's starting point. Instead,

they start with the whole domain or search space and keep discarding parts of it that are guaranteed not to contain a solution, leaving as solutions and quasi solutions, those that may contain solutions. This approach is complete: it will not miss solutions, and in the event that no solution is returned, it is safely guaranteed that it is because there was no solution to the problem at hand in the first place.

Let us illustrate the advantages of working with intervals when there is uncertainty in one parameter of our model in the following example.

Example 4. *Say we want to know the behavior of a phenomenon represented by the Gaussian model*

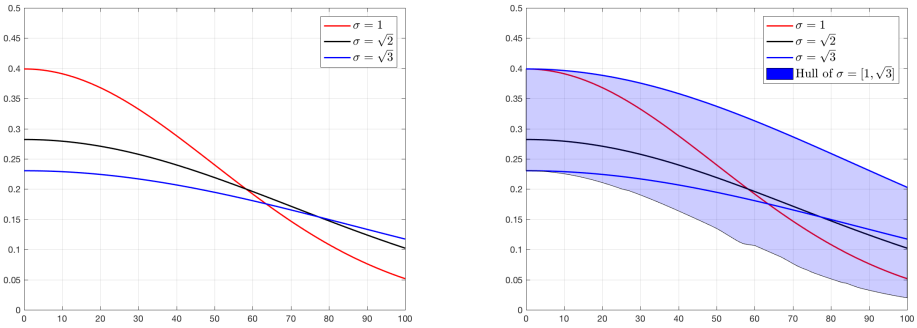
$$y' + \frac{x}{\sigma^2} y = 0, \ \textit{where } y(0) = \frac{1}{\sigma\sqrt{2\pi}} \tag{2.4.2}$$

for all $\sigma \in [1, \sqrt{3}]$. *The analytical solution of Equation* (2.4.2) *is*

$$\forall \sigma, f_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{2.4.3}$$

it and can be used to illustrate the advantages of working with intervals when there is uncertainty in one parameter of the model.

It is easy to believe that considering the functions $f_1(x)$ and $f_{\sqrt{3}}$ as boundary of the parametric family $\{f_\sigma\}_{\sigma\in[1\sqrt{3}]}$ would be enough, but, in fact, it is not true. Information corresponding to f_{sqrt2} is lost; see Figure 2.7(a). On the other hand, if we discretize and solve the resulting system of equations (in this case linear) using ICST, then we obtain an envelope to the family of functions; see Figure 2.7(b).



(a) Plots of f_1, $f_{\sqrt{3}}$, and $f_{\sqrt{2}}$ (b) Interval envelope of the $\{f_\sigma\}_\sigma$

Figure 2.7: Interval vs. Naive approach for $\sigma = [1, \sqrt{3}]$

2.5 Conclusion

In this chapter, we illustrated how intervals can be used to obtain reliable approximations of transcendent numbers and exact integrals. After introducing interval arithmetic, we were able to show the importance of intervals to obtain a reliable envelope of all the solutions of a parametric differential equation.

Chapter 3

Finite Element Method

I believe every human has a finite number of heartbeats. I don't intend to waste any of mine.

Neil Armstrong

Many natural phenomena can be modeled using parametric partial differential equations (PDE) over a non-regular boundary domain. A widely-used method to solve such PDEs is Finite Elements Method. In this disseration, we present a modification of FEM that allows us to handle uncertainty on the parameters of such equations.

In this chapter, we start by recalling the principle of the Finite Element Method to solve differential equations before to modify it in order to be able to handle uncertainty in the involved parameters of the model.

3.1 Basic Concepts

FEM allows finding numerical approximations to solutions of PDEs over a continuous domain. Such PDEs are in an integral (or weak) form, which characterizes the behavior of the physical model on a large number of disjoint subdomains named "finite elements" [20, 25, 101, 119]. To write a PDE in a **weak or variational** form, we use Green's Theorem:

$$\int_{\Omega} (\Delta u)v + \int_{\Omega} \nabla u \cdot \nabla v = \int_{\Gamma} (\partial_n u)v, \tag{3.1.1}$$

which is a particular case of the Divergence Theorem:

$$\int_{\Omega} (\text{div } \mathbf{p})v + \int_{\Omega} \mathbf{p} \cdot \nabla v = \int_{\Gamma} (\mathbf{p} \cdot \mathbf{n})v, \tag{3.1.2}$$

where div p is the divergence of the vector field $\mathbf{p}$, that is, if $\mathbf{p} = (p_1, p_2)$:

$$\text{div } p = \frac{\partial p_1}{\partial x_1} + \frac{\partial p_2}{\partial x_2}.$$

If we take $\mathbf{p} = \nabla u$, we obtain Green's Theorem.

The following is an example of how to express a partial differential equation as the weak form using the Green Theorem.

Example 5. *Consider the boundary value problem:*

$$\begin{cases} -\Delta u + cu = f, \ \textit{in } \Omega, \\ \\ u = g_0, \ \textit{on } \Gamma_D, \\ \\ \partial_n u = g_1 \ \textit{on } \Gamma_N. \end{cases} \tag{3.1.3}$$

where $\partial_n u = \nabla u \cdot \mathbf{n}$*, with* $\mathbf{n}$ *the unit normal vector on points of* Γ *always pointing outwards. Without specifying spaces where* u *and* v *are, the weak formulation of (3.1.3) can be written as follows:*

$$\begin{cases} \textit{Find } u \textit{ such that} \\ \\ u = g_0, \ \textit{on } \Gamma_D, \\ \\ \int_\Omega \nabla u \cdot \nabla v + c \int_\Omega uv = \int_\Omega fv + \int_{\Gamma_N} g_1 v \end{cases} \tag{3.1.4}$$

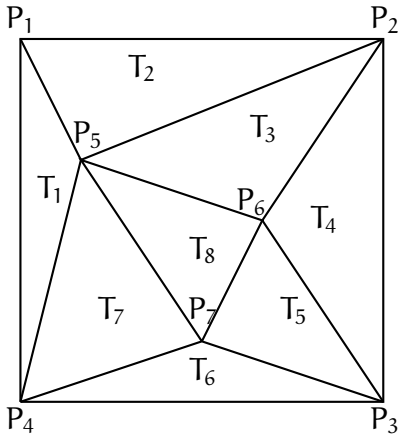
for all v*, such that* $v = 0$ *on* Γ_D*.*

Triangularization: let us assume that Ω is a polygonal domain, and let us make a partition of Ω called a triangularization, by subdividing Ω into a set T_h of non-overlapping triangles K_i, i.e.,

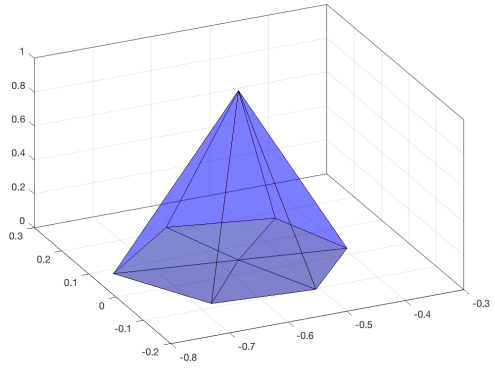
$$\Omega = \bigcup_{K \in T_h} K$$

such that no vertex of one triangle lies in the interior of the edge of another triangle [3, 34, 30, 38].

A triangularization or mesh is defined by point coordinates (x_i, y_i), $1 \leq i \leq n_p$, where n_p denotes the total number of points on the mesh. It is necessary to know how these points are connected to form the elements, so we need a connectivity matrix $(conn_{i,j})_i, j$ to describe the relation between local points and global points. For a linear triangular grid, $conn_{i,j}$, $j = 1, 2, 3$, $i = 1, 2, \ldots, n_e$, denotes the global label of the jth point of the ith element, where n_e is the total number of elements. For example, in Fig 3.1(a), we see that $conn_{1,1} = 1$, $conn_{1,2} = 4$, $conn_{1,3} = 5$ meaning that the element T_1 is defined by the points: P_1, P_4, P_5 written counterclockwise.



(a) Triangularization of $\Omega = [0,1] \times [0,1]$ on eight elements



(b) A piecewise-linear basis function

Figure 3.1: Example of a triangularization of Ω and an element of the basis function

To enforce boundary conditions, we need to know what nodes are on the boundary of the domain, and what type of boundary conditions exist there. We define a two-column matrix gbc as:

$$gbc_{i,1} = \begin{cases} 0 & \text{if} \quad \text{it is an interior point} \\ 1 & \text{if} \quad \text{it is a Dirichlet point} \\ 2 & \text{if} \quad \text{it is a Newmann point} \end{cases}$$

Moreover, when $gbc_{i,1} \neq 0$ for any i, we use $gbc_{i,2}$ to specify the corresponding boundary value.

Once the discretization is done and the boundary conditions are set, the next step consists in defining the finite element space V_h containing all functions v continuous on Ω, such that each v is affine on each triangle K, and $v = 0$ on $\partial\Omega$.

Assuming that T_h contains N interior vertices x_i, $1 \leq i \leq N$ then we can define a basis

of functions $v_j \in V_h$ for each interior point satisfying:

$$v_j(x_i) = \delta_{ij} \tag{3.1.5}$$

Note that the support of ϕ_j consists of the triangles with x_j as a common vertex, see Figure 3.1(b).

The discrete version of the weak formulation, specifying the numbering of Dirichlet nodes in the discrete Dirichlet conditon unfolds as follows:

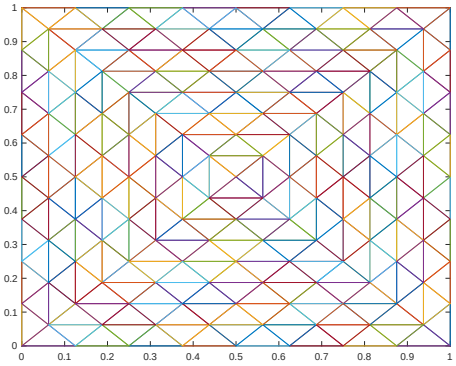
$$\begin{cases} \text{Find } u_h \in V_h \text{ such that} \\ \\ u_h(p_j) = g_0(p_j), \text{ on } \Gamma_D, \\ \\ \int_\Omega \nabla u_h \cdot \nabla v_h + c \int_\Omega u_h v_h = \int_\Omega f v_h + \int_{\Gamma_N} g_1 v_h \end{cases} \tag{3.1.6}$$

Example 6. *The first example to test out implementation is based on the boundary condition in a 2-D poisson problem [44, 1] on* $\Omega = [0,1] \times [0,1]$*:*

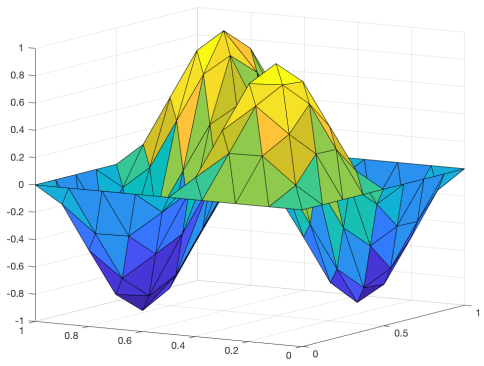
$$-\Delta u = 4\pi^2 \sin(2\pi x) \sin(2\pi y) \tag{3.1.7}$$

$$u(x,y) = 0;\ \forall (x,y) \in \partial\Omega \tag{3.1.8}$$

In this example, the domain Ω is discretized on $ne = 256$ elements. The number of points, and therefore the number of equations and variables is $np = 145$. The first



(a) Triangularization of a compact domain



(b) Solution of the Poisson equation using triangular finite elements

Figure 3.2: Solution of a poisson problem in a triangular finite domain Ω

column of the vector gbc consists of 0's or 1's due to the fact that boundary conditions are only Dirichlet Figure 3.2(a).

After solving the integral each integral equation, and solve the system of equations we have a numerical approximation to the solution Figure 3.2(b). In Chapter 8, we will retake this example but handling uncertainty on the boundary condition.

Next example is integrated on the same domain and with the same discretization. This time is a equation that not only depends on the space but also depends on time.

Example 7. *Consider the transient nonlinear heat conduction in a given body:*

$$\begin{aligned} \rho C_p \frac{\partial u}{\partial t} &= \nabla \cdot (k \nabla u) + Q_u \ \textit{on}\ \Omega \times [0, T] \\ u &= g \ \textit{on}\ \partial\Omega \times [0, T] \\ u(x, 0) &= T_0 = 0, \ \forall x \in \Omega \end{aligned} \tag{3.1.9}$$

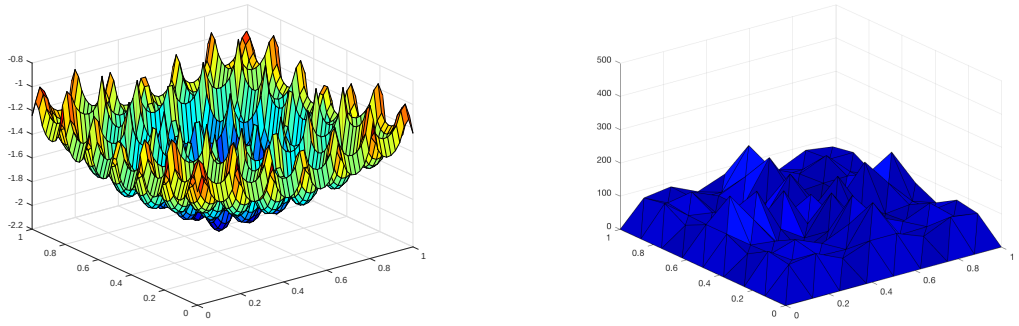
In (3.1.9), $u = u(x,t)$ *is the temperature,* $C_p = 50.0$ *is the heat capacity to constant preassure and* $k = k(u) = 2\left(1 + \frac{5u}{1000}\right)$ *is the thermal conductivity, and* $\rho = 1$ *is the density of the material. The function* $Q_u = Q_0 Q(x,y)$ *represents heat sources, where*

$$f(\tilde{x},\tilde{y}) = \left(\left|\sin(\tilde{x})\sin(\tilde{y})\exp\left(\left|100 - \frac{\sqrt{\tilde{x}^2+\tilde{y}^2}}{\pi}\right|\right)\right| + 1\right)^{0.1}$$

$$Q(x,y) = -10^{-4}\frac{f(\tilde{x},\tilde{y}) - f_{min}}{f_{max} - f_{min}}$$

$$\tilde{x} = 20x - 10;\ \tilde{y} = 20y - 10$$

The function $f(\tilde{x},\tilde{y})$ *is known as the Cross-in-Tray function, see Figure. 3.3(a).*



(a) The heat generation function

(b) Solution of the transient nonlinear heat conduction equation using triangular finite elements

Figure 3.3: Solution of transient nonlinear heat conduction and the source of heat

In Problem (3.1.9), the derivative in time is approximated by finite differences, i.e., $\frac{\partial u}{\partial t} = \frac{u^i - u^{i-1}}{\Delta t}$ with $\Delta t = 0.005$. The graph of the solution in $t = 0.025$ is given in Figure 3.3(b).

3.2 Conclusion

In this chapter, we gave an introduction of to the Finite Elements Method. We presented the discretization of a domain, and present the elements of the functional basis. We closed the chapter with two examples, one of them a partial differential equations, which depends only of the space domain, and another example, which depends on space and time. These examples are considered with Dirichlet condition without uncertainty. This allows us to observe the form of the solution to later observe the changes when the same examples are considered with uncertainty on the boundary conditions in Chapter 8.

Part II

Contributions

UTEP 2018

Chapter 4

Problem Statement

We can not solve our problems with the same level of thinking that created them.

Albert Einstein

A dynamical system is a system, which describes the dependence of a state vector in an instant of time of both: time and the state vector corresponding to the previous instant of time. [77, 104, 108]. Dynamical systems are usually represented as differential equations, and these differential equations depend on one or more parameters [52, 87, 112].

We will work with differential equations whose domain could be an interval of time, a compact domain of space, or the cartesian product of a compact domain and a closed interval of time.

We work in thesis with the following kind of parametric differential equations:

- Nonlinear system of ordinary differential of equations [58]:

$$\dot{X} = f(X, \Lambda), \quad X(t_0) = v_0, \tag{4.0.1}$$

where $t \in [t_0, t_m]$, the function $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$, $X \in \mathbb{R}^n$ is the state vectors, and $\Lambda \in \mathbb{R}^p$ is a p-dimensional vector of time-invariant parameters. For example:

$$\begin{cases} x_1' &= \alpha x_1 - \beta x_1 x_2 \\ x_2' &= -\gamma x_2 + \delta \nu x_1 x_2 \end{cases}, \quad x_1(0) = 10, x_2(0) = 10. \tag{4.0.2}$$

here, $X = (x_1(t), x_2(t))^T$, and $\Lambda = (\alpha, \beta, \gamma, \delta)^T$

- Parametric partial differential equations [13]: It is an equation of the form:

$$f\left(x_1, \ldots, x_n, u, \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_n}, \ldots, \Lambda\right) = 0 \tag{4.0.3}$$

the function u is defined on a compact Ω, and it satisfies either: Dirichlet, Neumann, or Robin boundary conditions.

After discretizing the domain, Equations (4.0.1) and (4.0.3) turn into a large, sometimes nonlinear, system of equations

$$F(X; \Lambda) = 0. \tag{4.0.4}$$

where $F : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$. We saw in Chapter 1 that in order to deal with large dimension issues we have to search for the solution in a smaller subspace than the original search space. The proper orthogonal decomposition method (POD) is a well-known method to locate a subspace containing the solution, but it does not take into account the uncertainty on the parameters to find the snapshots. Chapter 5 shows how to use interval arithmetic and interval constraint solving techniques to obtain a basis of the subspace taking into account possible uncertainty of the parameters Λ. Much work has been done to find an enclosure $[\underline{X}, \overline{X}]$ of the solution X when we have an enclosure of the parameters Λ.

Assume we have access to observations of the dynamic systems, i.e., for some subset $I_{Obs} \subset \{1, 2, \ldots, n\}$, we have $x_i = \tilde{x}_i$, $\forall i \in I_{Obs}$. The observation could be inaccurate,

so x_i may be represented as an interval. Chapter 6, we use interval constraint solving techniques along reduced-order modeling to estimate parameters which leads to a given behavior of a dynamic system. With

$$\begin{cases} F_\Phi(p, \lambda) = 0 \\ \sum_{j=1}^{m} \Phi_{i,j} p_j = \tilde{x}_i, \forall i \in I_{Obs}. \end{cases} \tag{4.0.5}$$

Equation (4.0.5) can be used to model a natural phenomenon that has been perturbated and requires recomputation of some parameters to guarantee a specific outcome of the modeled dynamic system Chapter 7.

Finally, capturing the uncertainty in combustion simulations is of great interest not only for fundamental research but also for vehicle designers to evaluate the behavior of novel fuels to engine performance. In this point, to run simulations is essential to design engines with a good performance. To run such simulations, researchers represent the model they want to simulate as partial differential equations over a domain, and then, they use Finite Element Method (FEM) to solve the PDE. In Chapter 8, we propose a novel technique to handle uncertainty in FEM using interval computations and Interval Constraint Solving Techniques to capture uncertainty in JP-8 simulations.

Chapter 5

Interval POD (I-POD)

I became insane, with long intervals of horrible sanity.
Edgar Allan Poe

Once again, the problem that we are solving is as follows: Given a parametric system of equations (also known as the Full Order Model):

$$F(x, \lambda) = 0, \quad \lambda \in \mathbf{I} \tag{5.0.1}$$

where F can be either linear or nonlinear function $F : \mathbb{R}^n \to \mathbb{R}^n$, that might emerge from the discretization of a set of partial differential equations and $\mathbf{I}$ is a fixed interval. The idea behind POD is to solve (Equation (5.0.1)) for a sequence of values $\lambda_i \in \mathbf{I}$, i.e.,

$$F(x, \lambda_i) = 0 \tag{5.0.2}$$

where $\lambda_i \in \mathbf{I}$, for $i = 1, 2, \ldots, n$. The main idea of this method is based on the assumption that there exists a 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.

This chapter proposes an interval version of POD. The original idea behind this new Interval POD is that we aim to reduce the amount of work in solving the Full Order Model for many different values of the input parameters (λ). Instead we suggest and experimented solving the Full Order Model once on the entire interval containing all desirable values of λ.

This slight change in concept (many processes solving for many different values of λ vs. one process solving for an entire interval instead) has consequences in our ability to solve the Full Order Model. Now that an interval is part of the problem we are bound to use interval-computation-based solving techniques and we found interval constraint solving techniques to be very practical to do so.

More specifically, we are now solving:

$$F(x, \mathbf{I}) = 0, \tag{5.0.3}$$

which is a nonlinear system of equations with explicit uncertainty in the shape of an interval. We called this variation of POD the **Interval Proper Orthogonal Decomposition (I-POD)** method.

5.1 Numerical Results

This section describes and reports on preliminary experiments of our I-POD method on four well-known problems: the Gauss model, Burgers' equation, the Transport equation, and the Bratu equation. In each of these experiment, we aim to assess the ability of I-

POD to generate snapshots that yield a reduced basis of quality enough that the solution of the reduced-order model yields a very small error (w.r.t. FOM solution) in comparison to what a similar process using POD achieves.

5.1.1 Gauss Model

We start with a simple σ-parameter Gauss equation model, which is an ordinary differential equation with initial condition. Find $y_\sigma(x)$ such that

$$y' + \frac{x}{\sigma^2} y = 0, \forall x \in [0, 2] \text{ and } y(0) = \frac{1}{\sigma\sqrt{2\pi}} \tag{5.1.1}$$

To solve Equation (5.1.1) using finite differences method, we divide the domain $[0, 2]$ in N uniforms points; we denote as y_i the approximation of the solution in an arbitrary point x_i, and we approximate the derivative substituting

$$y_i' = \frac{y_i - y_{i-1}}{h} \tag{5.1.2}$$

where $h = x_i - x_{i-1}$. In this scheme, $x_i = i \cdot h$ for $i = 0, 1, \ldots, N$. Substituting the approximations of y_i', y_i, and x_i in Equation (5.1.1), we can construct the system of equation

$$(\sigma^2 + h^2 i) y_i - \sigma^2 y_{i-1} = 0, \ \forall i = 1, 2, \ldots, N. \tag{5.1.3}$$

In this case, y_0 is the initial condition.

We want to find the snapshots solving for y in the linear system of equations

$$(\sigma_j^2 + h^2 i) y_i - \sigma_j^2 y_{i-1} = 0, \ \forall i = 1, 2, \ldots, N. \tag{5.1.4}$$

where σ_j is uniformly distributed in the interval $[1, \sqrt{3}]$. We can store the solution of

Equation (5.1.4) as a column in the matrix of snapshots $\text{Snap}([1,\dots,N],j) = y$, see Figure 5.1(a).

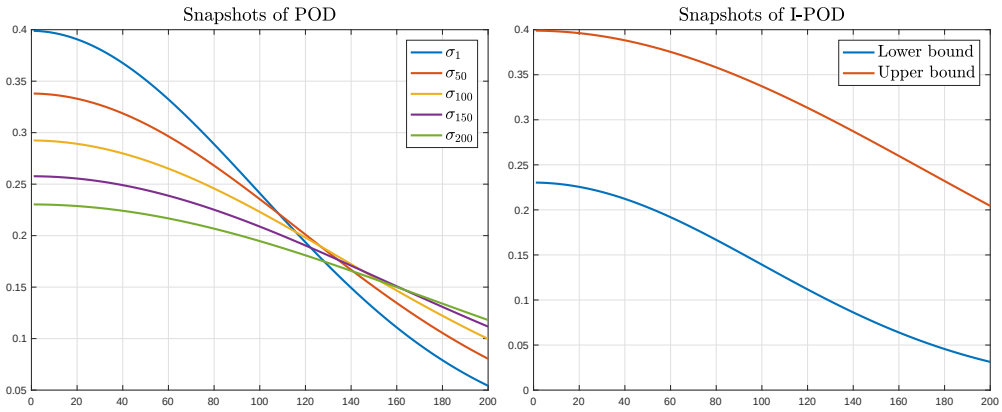
After computing the singular values decomposition of Snap

$$\text{Snap} = W\Sigma V^T \tag{5.1.5}$$

To define the basis Φ_{POD}, for a fixed $\varepsilon < 1$, we select the first k columns of W such

$$\frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{N} \sigma_i} \geq \varepsilon. \tag{5.1.6}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ are the k first singular values of Snap.



(a) Snapshots of Gauss model using POD (b) Snapshots of Gauss model using I-POD

Figure 5.1: Snapshots of Gauss model using POD and I-POD method

5.1.2 Burgers Equation

Consider the Burgers' equation:

$$\frac{\partial U(x,t)}{\partial t} + \frac{\partial f(U(x,t))}{\partial x} = g(x), \tag{5.1.7}$$

where U is the unknown conserved quantity (mass, density, heat etc.), $f(U) = 0.5U^2$ and in this example, $g(x) = 0.02\exp(0.02x)$ [63, 99, 71, 68]. 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 $t > 0$.

Below, in Section 5.1.2 and **Algorithm 4**, we describe the procedure to obtain the snapshots and the reduced basis in the POD method for the Burgers equation. We will then compared it with **I-POD**.

The POD algorithm is modified in the following way:

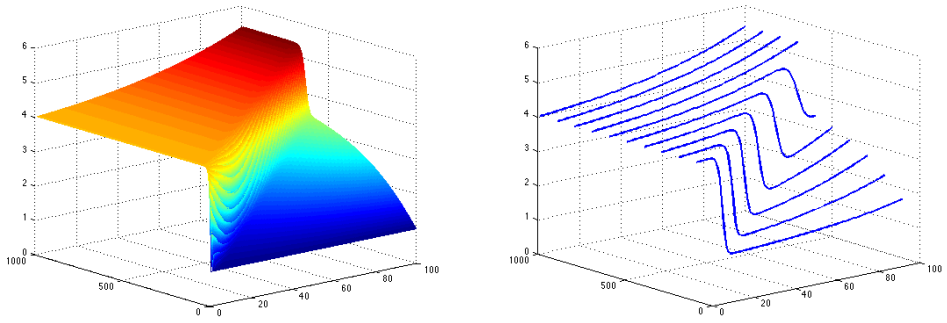


Figure 5.2: Solution of Burger equation for $\lambda = 4$, and some snapshots corresponding to this parameter.

The matrices returned by Section 5.1.2 and **Algorithm 4** will be the reduced basis to apply ROM strategies. We applied previous procedures, FOM, POD, and I-POD, to solve (Equation (5.1.7)) and we obtained the results reported in the Table 5.1.

Algorithm 3 Computing a Proper Orthogonal Decomposition Basis.

Require: An empty matrix where we will collect the snapshots: Snap $= []$, and an initial $\lambda_1 = 3.5$.

1: **for** $i = 2$ **to** $i = 100$ **do**

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

3: Collect snapshots: From $t_1, t_2, \ldots, t_n$, select a subsequence $t_{i1}, t_{i2}, \ldots, t_{ip}$.

4: Add new columns to the snapshot matrix: Snap $=$ [Snap $U(x, t_{i1})$ $U(x, t_{i2})$ $\ldots U(x, t_{ip})$].

5: Update λ: $\lambda_i = \lambda_{i-1} + 0.01$

6: **end for**

7: Compute the singular value decomposition of the matrix Snap: Snap $= W\Sigma V^T$

8: Select from W the principal components with the greatest accumulated variance:

9: $\sigma = 0$ $(\sigma_i = \Sigma_{ii})$

10: **for** $k = 1$ **to** $k = n$ **do**

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

12: **if** $\sigma > Tol$, $0 < Tol < 1$ **then**

13: break

14: **end if**

15: **end for**

16: Select the first k columns of W, and define Φ: $\Phi = W(:, [1, 2, \ldots, k])$

17: **return** Φ.

Algorithm 4 Computing a Proper Orthogonal Decomposition Basis Using I-POD.

Require: An empty matrix where we will collect the snapshots: Snap $= []$.

1: Solve:

$$\begin{cases} \dfrac{\partial U(x,t)}{\partial t} + \dfrac{\partial f(U(x,t))}{\partial x} = g(x), \\ g(x) = 0.02\exp(0.02x) \\ U(x;0) \equiv 1, \text{ for all } x \in [0;100], \\ U(0;t) = \lambda_i, \text{ for } t > 0. \end{cases} \tag{5.1.8}$$

The solution of Equation (5.1.8) 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 $U_l(x_0, t_0)$ 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 5.3

2: Collect snapshots: From $t_1, t_2, \ldots, t_n$, select two subsequences $t_{il1}, t_{il2}, \ldots, t_{ilp}$ and $t_{ir1}, t_{ir2}, \ldots, t_{irp}$.

3: Take snapshots from U_l and add new columns to the snapshot matrix: Snap $=$ [Snap $U_l(x, t_{il1})$ $U_l(x, t_{il2})$ $\ldots, U_l(x, t_{ilp})$].

4: Take snapshots from U_r and add new columns to the snapshot matrix: Snap $=$ [Snap $U_r(x, t_{ir1})$ $U_l(x, t_{ir2})$ $\ldots, U_l(x, t_{irp})$].

5: Compute the singular value decomposition of the matrix Snap: Snap $= W\Sigma V^T$

6: Select from W the principal components with the greatest accumulated variance:

7: $\sigma = 0$ $(\sigma_i = \Sigma_{ii})$

8: **for** $k = 1$ **to** $k = n$ **do**

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

10: **if** $\sigma > Tol$, $0 < Tol < 1$ **then**

11: break

12: **end if**

13: **end for**

14: Select the first k columns of W, and define Φ: $\Phi = W(:, [1, 2, \ldots, k])$

15: **return** Φ.

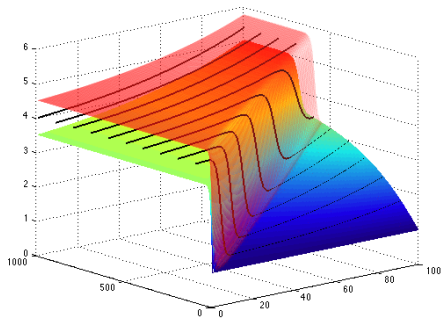


Figure 5.3: Infimum and supremum of the solution of Burger equation and some snapshots corresponding to $\lambda = 4$ are shown.

Table 5.1: Comparing POD and I-POD methods in solving a particular example of Burgers' Equation

Method	FOM	POD	I-POD
Dimension	100	37	36
Time to solution	1.5 s	0.75 s	0.75 s
Relative error: $\|u_{fom} - u_{rom}\|/\|u_{fom}\|$	-	$4.85E-4$	$5.76E-4$

We observe that there is no significant difference between the traditional method (using POD) and the method we propose (using I-POD) w.r.t. (1) the dimension of the subspace, (2) the time it takes to solve the problem once we have identified the reduced basis, and (3) the relative error compared with the FOM solution. The major two advantages of our proposed method are:

- the computational time it requires to obtain the snapshots: Our approach requires 68.52% less time than the original one and the quality of the snapshots our method generates is comparable to that generated by POD as observed in the relative error;

and

- the ability to handle uncertainty: the interval that contains λ, handled at once by I-POD, is similar to uncertainty and is handled without problems. Further experiments will aim to demonstrate that I-POD produces meaningful results also when there are other sources of uncertainty (beyond the interval for λ).

5.1.3 Transport Equation

The transport equation is a partial differential equation that models the concentration of a contaminant in the position x in at time t in a fluid that is flowing with velocity v in a thin straight tube whose ross section, denoted by A is constant [121, 27]. Such concentration will be denoted by $U(x,t)$. if the function $U(x,t)$ and its partial derivatives of order one are continuous functions of x and t, and the fluid velocity v and the cross section of the tube, A, are constants, then the Transport Equation is reduced to:

$$\begin{aligned} &\frac{\partial U}{\partial t} + v\frac{\partial U}{\partial x} = 0 \\ &(x,t) \in \Omega \end{aligned} \tag{5.1.10}$$

Where Ω is a convex domain. In particular, we solve (Equation (5.1.10)) with $U(x,t)$ subject to the following boundary and initial conditions:

$$U(0,t) = u(t) = -\sin(2\pi t) + \sin(\pi t) \tag{5.1.11}$$

$$U(x,0) = u(x) = \sin(2\pi x) + \sin(\pi x) \tag{5.1.12}$$

for all $t \in [0,1]$, and $x \in [0,1]$.

Using $\nu \in [0.5, 1.5]$ as the input parameter, we can proceed, similarly to how we did in the Burger equation case, and compute, first, a basis using POD method, and later, using I-POD.

Comparative values are presented in Table 5.2:

Table 5.2: Comparing POD and I-POD methods in solving a particular example of Transport Equation

Method	FOM	POD	I-POD
Dimension	100	12	76
Time to solution	0.15 s	0.022 s	0.042 s
Relative error: $\|u_{fom} - u_{rom}\|/\|u_{fom}\|$	-	$7.97E-5$	$1.81E-5$

In this experiment, we observed that even when the dimension of the subspace is larger using the I-POD (76) than when using POD (12), the time needed to compute the reduced basis in I-POD is significantly less than the computing time needed using POD. Once, both basis are known, solving the reduced problem from POD or I-POD take the same time.

In Figure 5.4, we can observe the plot of the solution of the transport equation for time-steps 20, 40, 60, 80. In green and red are respectively the infimum and the supremum of the interval containing the solution.

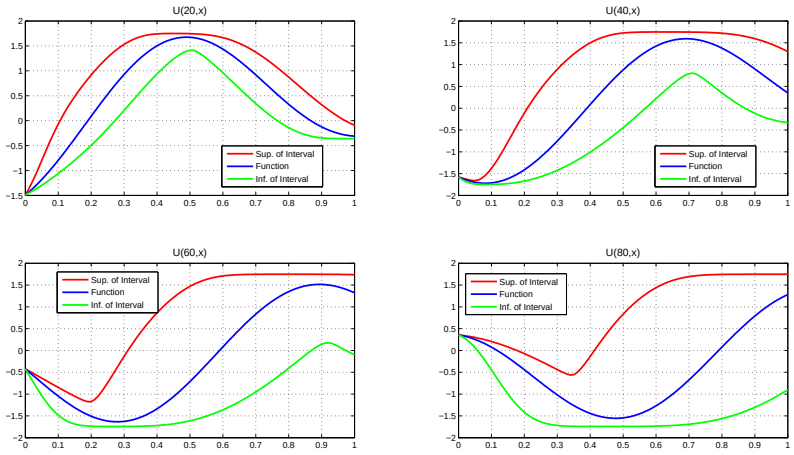


Figure 5.4: Solution of Transport Equation for time-steps 20, 40, 60, 80, enclosed in the Interval solution.

5.1.4 Lotka-Volterra

Consider a particular case of the Lotka-Volterra problem, which involves a model of a predator-prey system.

$$\begin{cases} y_1' = \theta_1 y_1(1 - y_2), & y_1(0) = 1.2 \quad \theta_1 = 3, \\ y_2' = \theta_2 y_2(y_1 - 1), & y_2(0) = 1.1 \quad \theta_2 = 1 \end{cases} \tag{5.1.13}$$

y_1 and y_2 respectively represent the amount of preys and predators. In this particular example, the growth rate of the first species reflects the effect the second species has on the population of the first species (θ_1). Similarly, the growth rate of the second species reflects the effect the first species has on the population of the second species ($\theta 2$). The system was integrated from $t_0 = 0$ to $t_m = 10$. Numerical experiments were carried out with a constant step size $h = 0.1$. Ranges for the parameters $\theta_1 \in [2.95, 3.05]$ and $\theta_2 = [0.95, 1.05]$ were used as input. Comparative values are presented in Section 5.1.4:

In Figure 5.5, we observe the interval enclosure of (Equation (7.2.1)) when $\theta_1 =$

Table 5.3: Comparing POD and I-POD methods in solving a particular example of Lotka-Volterra

Method	FOM	POD	I-POD
Dimension	200	3	3
Time to solution	0.0129 s	0.0746 s	0.0484 s
Relative error: $\|\|u_{fom} - u_{rom}\|\|/\|\|u_{fom}\|\|$	-	$4.56E-4$	$1.20E-3$

$[2.95, 3.05]$ and $\theta_2 = [0.95, 1.05]$.

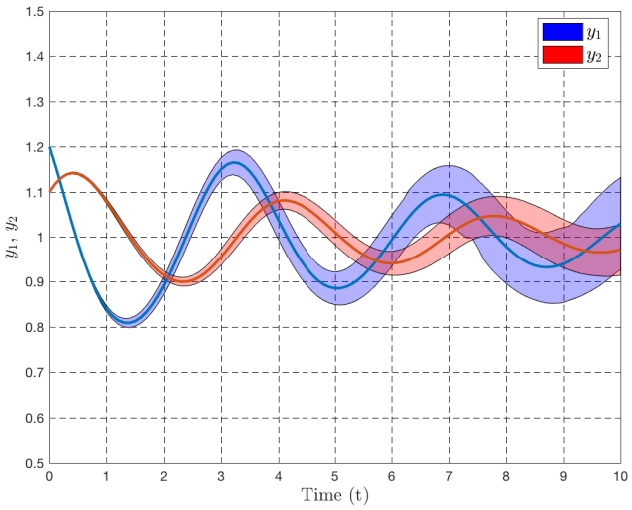


Figure 5.5: The interval enclosure of Lotaka-Volterra model for $\theta_1 = [2.95, 3.05]$ and $\theta_2 = [0.95, 1.05]$

The examples of the Burger's equation (Equation (5.1.7)) and the Transport equation (Equation (5.1.10)) were partial differential equations with uncertainty in one parameter. The number of unknowns was the same as the number of points in the discretization of the domain. Problem (Equation (7.2.1)) is more challenging because it is a system of

two nonlinear partial differential equations, which means that the number of unknowns is twice as large as the number of nodes in the discretization. Also in this experiment, we show that we are able to handle uncertainty in two parameters. The results reported in Table (Section 5.1.4) show that we can significantly reduce the size of the search space since we were able to go from dimension $n = 200$ to a reduced dimension $k = 3$, which constitutes a 98.5% contraction. Having uncertainty in two parameters did not yield a large loss of quality, since, given the reduction of the subspace, just one order of magnitude is lost.

5.1.5 The FitzHugh-Nagumo Model

The following nonlinear model is based on the classical FitzHugh-Nagumo oscillator[82, 57, 70, 85]. Let:

$$f(v) = v(v - \alpha)(1 - v)$$

and let (v_{eq}, w_{eq}) be the equilibrium point of the nonlinear system. This system has been modified so that the equilibrium point coincides with the initial condition, i.e., $(v(0), w(0)) = (v_{eq}, w_{eq})$

$$\begin{cases} \dfrac{dv}{dt} &= f(v + v_{eq}) - f(v_{eq}) - w \\ \\ \dfrac{dw}{dt} &= \varepsilon(v - \gamma w) \end{cases} \tag{5.1.14}$$

We will illustrate the behavior of the FitzHugh-Nagumo model using the following values for the parameters: $\alpha = 0.139$, $\varepsilon = 0.008$, $\gamma = 2.54$, $v_0 = v_{eq} = 0.15$, $w_0 = w_{eq} = -0.028$, the domain $t = [0, 10]$ was discretized using $\Delta_t = 0.1$.

Let us consider the initial condition $v_0 = [0.1, 0.2]$ as the input parameter. Observe

in Table 5.4 the comparative values, and in Figure 5.6 an enclosure of the solution when $v_0 = [0.1, 0.2]$.

Table 5.4: Comparing POD and I-POD methods in solving a particular example of The FitzHugh-Nagumo Model

Method	FOM	POD	I-POD
Dimension	200	3	2
Time to solution	0.022 s	0.079 s	0.188 s
Relative error: $\|u_{fom} - u_{rom}\|/\|u_{fom}\|$	-	$6.28E-05$	0.0110

Function f in the definition of FHN (Equation (5.1.14)) is highly nonlinear. As a consequence, the nonlinear system of equations obtained when discretizing the domain is highly sensitive to overestimation when we use interval computations. In Table 5.4 we report our experimental results, which show that the performance of IPOD is affected by the highly nonlinear nature of the function we simulate. We observe that the time to solution of the ROM, even in a smaller space than POD, is larger. Similarly, the obtained accuracy (relative error to FOM) of the obtained solution is not nearly as good as that of POD. This is definitely an area of improvement, but overall, we observe that this problem does not naturally lend itself to POD (or IPOD for that matter) since even POD's time to solution is larger that that of FOM.

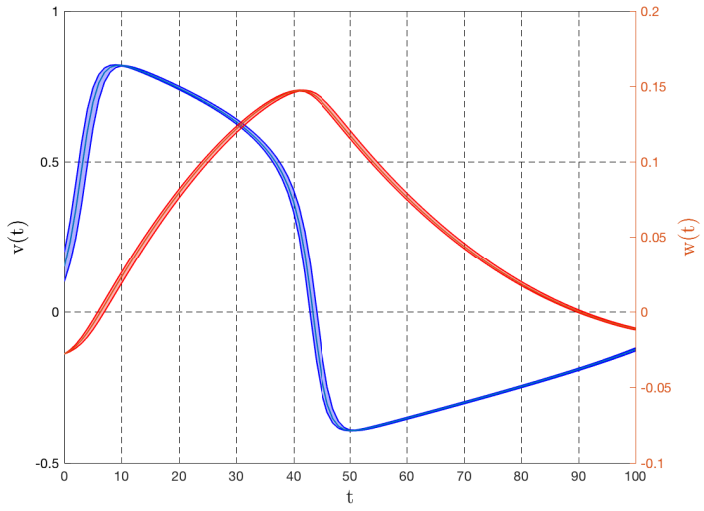


Figure 5.6: The interval enclosure of FHN model for $v_0 = [0.1, 0.2]$

5.1.6 The Bratu problem

In this section, we study the Bratu problem [105, 7, 114], which is an the elliptic PDE:

$$\begin{aligned} \Delta u + re^u &= 0 \qquad \text{on } \Omega : \{(x, y) \in 0 \leq x \leq 1, 0 \leq y \leq 1\} \\ \text{with } u &= 0 \qquad \text{on } \partial\Omega \end{aligned} \tag{5.1.15}$$

Studying Bratu's problem in more details is interesting because the existence and uniqueness of (Section 5.1.6) depends on the parameter r. There actually exists a critical value r_*, called the **Frank-Kamdndtskii** value, such that for $r > r_*$, there does not exist any solution to the Bratu's problem, and two solutions exist for $0 < r < r_*$ [39]. Once r_* has been determined, the finite difference only gives the lower branch of the solution [83]. Experimentally, it has been proven that for (Section 5.1.6) the Frank-Kamdndtskii value is around $r_* \approx 6.78$ [37, 8]. Zero-Dirichlet solutions are used to model combustion reactions.

We consider the discretization of the domain Ω, $u(x_i, y_j) = u_{i,j}$ with $1 \leq i, j \leq 30$

(Equation (5.1.16)).

$$\begin{aligned} \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{(\Delta y)^2} + \frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{(\Delta x)^2} + re^{u_{ij}} &= 0 \\ u_{0,j} = u_{31,j} = u_{i,0} = u_{i,31} &= 0 \end{aligned} \tag{5.1.16}$$

After substituting $r = [6.79, \infty]$ in (Equation (5.1.16)) and using ICST to solve it. We can prove that (Equation (5.1.16)) has no solution for $r > 6.79$. If $r \to 0$ then $u_{ij} \to 0$ for all $1 \leq i, j \leq 30$, we will focus in the solutions for $1 \leq r \leq 6.78$.

Now, let us consider the parameter $r = 1$, and apply ICST. In order to use ICST to solve the problem, we need to set the parameter as an interval $r = [1, 1]$. In that case, we obtain two solutions for Equation (5.1.16); see Figure 5.7.

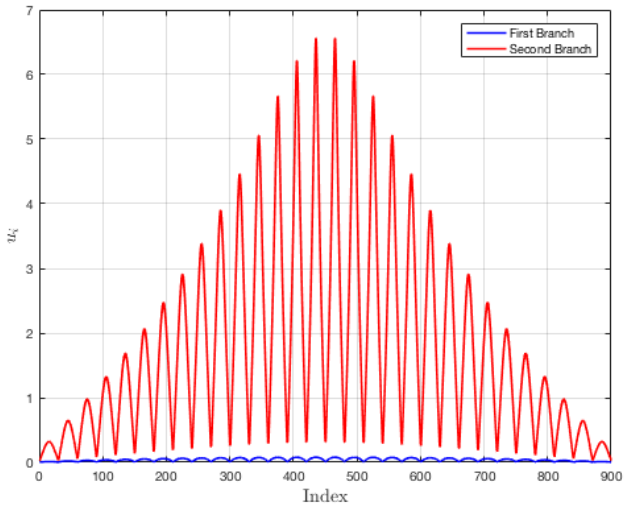


Figure 5.7: Two solutions of Bratu equation for $r = [1, 1]$

In order to perform a comparison between POD and I-POD, we first need to take 100 uniformly distributed values of parameter $r \in [1, 6.78]$ to compute the snapshots. For each r, the solution is column-wise sorted and stored in the snapshot matrix. Later, we use ICST to solve the same problem but now $r = [1, 6.78]$. Results are shown in Table 5.5.

Table 5.5: POD and I-POD methods in solving a particular example of The Bratu Problem

Method	FOM	POD	I-POD
Time to compute the reduced basis	-	2.56 s	7,200 s
Dimension	900	4	4
Time to solution	66.4 ms	11.2 ms	14.4 ms
Relative error: $\|u_{fom} - u_{rom}\|/\|u_{fom}\|$	-	$9.94E-04$	0.0045

Figure 5.8 shows a solution's enclosure of (Section 5.1.6) for $r = [1, 6.78]$.

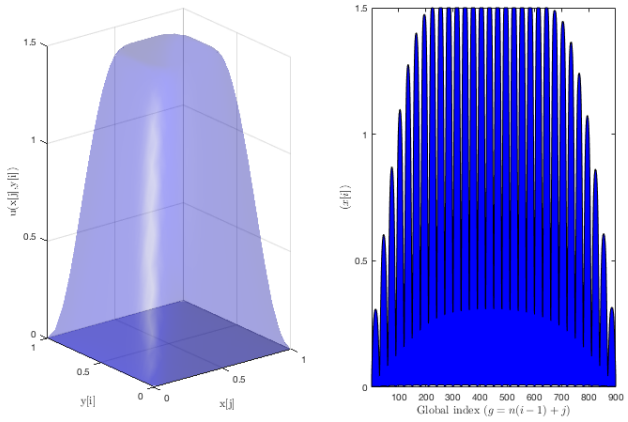


Figure 5.8: The interval enclosure of Bratu equation for $r = [1, 6.78]$

5.2 Conclusion

We proposed and described a novel (ROM) approach that improves the well-known Proper Orthogonal Decomposition method by: freeing traditional ROM techniques from

snapshot identification, providing reliable solutions, and allowing to handle uncertainty.

Our new approach is based on the use of Interval analysis and Interval Constraint Solving Techniques. We called this new method the Interval Proper Orthogonal Decomposition (I-POD). We tested I-POD on five nonlinear partial differential equations problems: Burgers' equation, the Transport equation, Lotka-Volterra problem, the FitzHugh-Nagumo Model, and the Bratu's Problem. We observed and reported promising performance of I-POD, when compared to POD.

Chapter 6

Predictions of Large Dynamic Systems Behavior

Never make predictions, especially about the future.

Hans Vestberg

The ability to conduct fast and reliable simulations of dynamic systems is of special interest to many fields of operations. Such simulations can be very complex and, to be thorough, involve millions of variables, making it prohibitive in CPU time to run repeatedly for many different configurations. Reduced-Order Modeling (ROM) provides a concrete way to handle such complex simulations using a realistic amount of resources. However, uncertainty is hardly taken into account. Changes in the definition of a model, for instance, could have dramatic effects on the outcome of simulations. Therefore, neither reduced models nor initial conclusions could be 100% relied upon.

In this chapter, we show how using information acquired about an unfolding dynamic phenomenon is possible, despite the possibly large size and embedded uncertainty of

its Full-Order discretized Model. We show how we can translate FOM data into ROM data and combine Interval Constraint Solving Techniques (ICST) with Reduced-Order Modeling techniques to properly account for both challenges: size and uncertainty.

6.1 Understanding Unfolding Events from Observations

In design situations, one has control of the input parameters and other conditions (e.g., initial conditions, boundary conditions) and simulating a dynamic phenomenon under said parameters and conditions provides knowledge or information to better understand the dynamic phenomenon under study. Such amount of control is not always available. It is the case of many phenomena; their occurrence are beyond the observers' control. However, if the general type of phenomenon is known, it could be crucial to be able to draw predictions of future behavoir of an unfolding event; e.g., predicting how epidemics could unfold in the future, which policies should be set to stop them.

The idea is that, based on the knowledge of a dynamic phenomenon and on observations of said phenomenon, we aim to predict its behavior further than the time of observations (to be able to take palliative action if necessary) as well as to identify potential ranges of its key features (e.g., input parameters, initial conditions).

Note that we assume that we know the kind of dynamic phenomenon we are observing (e.g., it is a pendulum, a predator-prey situation, or an disease propagation situation).

1. As a result, we can assume that we have access to the system of equations that results from discretizing the generic description of this problem:

$$F : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n, \quad \text{solve: } F(x, \lambda) = 0, \tag{6.1.1}$$

where $x \in \mathbb{R}^n$ will provide us information about the quantity of interest of the simulation, including the initial conditions and $\lambda \in \mathbb{R}^k$ stands for the input parameters.

2. In addition to the above, which is underdetermined, we also have access to observations of the unfolding phenomenon, i.e., observed values $\tilde{x_i}$ of some of the x_i, for some $i \in \{1, \ldots, n\}$:

$$\begin{aligned} Obs &= \{\tilde{x_i}, \text{ for some } i \in \{1, \ldots, n\}\} \\ &= \{\tilde{x_i}, \forall i \in I_{Obs} \subset \{1, \ldots, n\}\} \text{ with } I_{Obs} \neq \emptyset \end{aligned}$$

Observations add the following constraints (equations in this case) to the above system of equations $F(x, \lambda) = 0$:

$$\forall i \in I_{Obs}, \; x_i = \tilde{x_i} \tag{6.1.2}$$

Note that observations will be inaccurate (noise caused by measurement tools): we represent these as intervals. As a result, Obs is a set of interval values.

As stated earlier, solving $F(x, \lambda) = 0$ is tedious as the number of unknowns may exceed what can be solved/computed in a practical amount of time. We address this problem by using a reduced-order model, as described in Chapter 5. So we do not solve exactly the above – full-order – problem but instead, its reduced form:

$$F_\Phi : \mathbb{R}^m \times \mathbb{R}^k \to \mathbb{R}^n, \quad \text{solve: } F_\Phi(p.\lambda) = F(\Phi \cdot p, \lambda) = 0,$$

where Φ is the basis of a subspace where the solution of the original problem $F(x, \lambda) = 0$ is expected to lie. If we add to these the constraints coming from observations, such constraints express restrictions on variables (x_i, $\forall i \in I_{Obs}$) that are not part of the rest of the problem $F_\Phi(p.\lambda) = 0$. However, by definition of the reduced basis Φ, there is a linear relationship between the original variables $x \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$ with $m \ll n$:

$$x = \Phi \cdot p + z, \quad \text{with: } z \approx 0$$

As a result, we replace the constraints defined in Equation (6.1.2) by:

$$\forall i \in I_{Obs},\ \tilde{x_i} = \sum_{j=1}^{m} \Phi_{i,j} p_j$$

The reduced constraint system we solve is therefore as follows:

$$\begin{cases} F_\Phi(p, \lambda) = 0 \\ \forall i \in I_{Obs},\ \sum_{j=1}^{m} \Phi_{i,j} p_j = \tilde{x_i} \end{cases} \tag{6.1.3}$$

Note that the system is reduced because it has a lot fewer variables than the original full-order one, but it still has the same number of constraints.

This reduced problem is solved using interval constraint solving techniques, as explained in Chapter 5. What we expect to obtain from solving this problem is an envelope of the expected behavior of the dynamic system under observation. Most importantly, as interval constraint solving techniques will return quasi solutions in addition to validated ones, we will use the envelope as a way to rule out situations / behaviors, namely, everything outside the envelope.

6.2 Numerical Experiments

In this section, we illustrate our work with two examples: the Gaussian-model problem, already studied in Section 5.1.1, and the Lotka-Volterra problem (modeling the dynamics of predator / prey populations). For each of these problems, we consider that we have observations at several points in time. The values of these observations are intervals.

6.2.1 Gaussian Model

Let us consider the Gaussian Model [115, 97, 11]

$$y' + \frac{x}{\sigma^2} y = 0, \text{ where } y(0) = \frac{1}{\sigma\sqrt{2\pi}}, \tag{6.2.1}$$

we have already used this model to show the advantages of using ICST to handle uncertainty in dynamic systems Equation (2.4.2) and to illustrate how we can find a reduced basis using I-POD method Section 5.1.1.

In this chapter, we suppose we have a number of observations on the phenomenon and use them to identify the parameter that leads to such behavior. Consider the discretization of the original problem

$$(\sigma^2 + ih^2) y_i - \sigma^2 y_{i-1} = 0; \; i = 1, 2, \ldots, 100 \tag{6.2.2}$$

with $y(0) = 0$, and observed data given as intervals of the phenomenon in different instants of time:

Variable	Value
y_{10}	[0.24 , 0.32]
y_{20}	[0.23 , 0.31]
y_{30}	[0.22 , 0.30]
y_{40}	[0.21 , 0.28]

This problem has $(100 - m) + 1 = 101 - m$ unknowns, where m is the number of observations: the y_i's are we have not observed $(100 - m)$ plus the parameter σ $(+1)$. We solve this problem using the Interval Reduced-Order Model (Chapter 5). We have already identified a matrix ϕ using POD. The values of y_i's can be obtained as: $y_i =$

$\sum_{j=1}^{k} \phi(i,j)p_j$. The value of k in this case is two. The reduced problem we solve is the following:

$$\begin{cases} (\sigma^2 + h^2)\left(\sum_{j=1}^{2} \phi(1,j)p_j\right) - \sigma^2 y_0 &= 0; \\ (\sigma^2 + h^2)\left(\sum_{j=1}^{2} \phi(i,j)p_j\right) - \sigma^2\left(\sum_{j=1}^{2} \phi(i-1,j)p_j\right) &= 0;\ i = 2,3,\ldots,100 \\ \sum_{j=1}^{2} \phi(k,j)p_j &= y_k;\ k = 10,20,30,40 \end{cases} \tag{6.2.3}$$

where the unknowns are now p_1, p_2, and σ. The dimension was reduced from $101 - m$ to 4.

To solve this problem (Equation (6.2.3)) using interval constraint solving techniques, we need to start with an initial box that contains the feasible values of vector p. The tighter the initial box, the faster the solver will return the solution; so it is a good idea to invest some time in finding a good initial box.

Let us start by rearranging Equation (6.2.2) as follows:

$$\frac{y_1}{y_{i-1}} = \frac{\sigma^2}{\sigma^2 + ih^2} \tag{6.2.4}$$

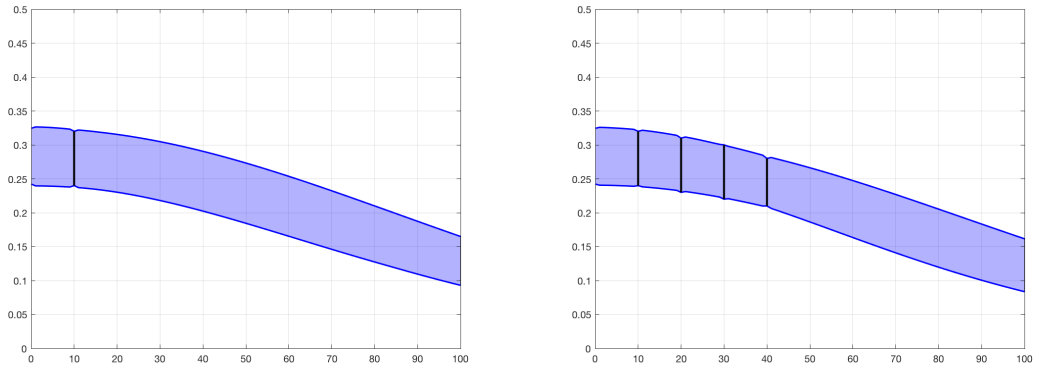
since the right hand side of Equation (6.2.4) and y_0 are positive, it follows that the sequence $\{y_i\}_{i=1}^{100}$ is positive and non increasing. Considering $\sigma \in [1, 100]$, we therefore have $y_0 = \frac{1}{\sigma\sqrt{2\pi}} = [0.003989422804014, 0.398942280401433]$. SInce $0 < y_i < y_0$, it follows

$$\forall i \geq 1, y_i \in [0, 0.398942280401433]$$

Knowing the interval where each $y_i$lies, we use the linear constraints system $\phi p = Y$, where Y is an interval vector whose i-component is y_i and obtain the following starting domains for variables p_i's:

$$P = \begin{bmatrix} -0.9635940410489494, 1.018651306585007 \\ -3.5921593513132990, 1.397845978701148 \\ -0.3901435316130543, 0.4776937802351998 \end{bmatrix} \tag{6.2.5}$$

We can use Equation (6.2.5) as an initial box along with the initial domain for $\sigma = [1, 100]$ to solve Equation (6.2.3) using ICST. The envelope for Equation (6.2.2) is shown in Figure 6.1(a).



(a) Interval solution for two observations

(b) Interval solution for four observations

Figure 6.1: Interval solutions of the Gaussian model for two and four observations respectively

We repeat the experiment this time for four observations, and we see that there is not much improvement; see Figure 6.1(b). This is because the most contraction for σ was

obtained as a result of the first observation. The estimation of σ for one observation is

$$\sigma = [1.229577791480822, 1.646728008711858],$$

and the estimation for four observations is

$$\sigma = [1.229577791480823, 1.646728008711857],$$

The contraction obtained at the first "try" is due to the fact that there was not much overestimation involved in the solving process. However, we can still observe a slight overestimation as the plot bulges around each of the observations. The fact that there is not much improvement despite the (even slight) overestimation points us that efforts must be put towards a better contraction of the domain of the parameter (here σ). However, since we are looking at predicting behavior and that we get actionable results, at this point, seeking more contraction on the parameter's domain could compromise the solving time.

6.2.2 Lotka-Volterra

Consider the Lotka-Volterra problem[89, 73, 116, 48], which describes a predator-prey system. We use the following system of differential equations to model it

$$\begin{cases} v' &= \alpha v - \beta v w, \quad v(0) = v_0 = 10; \\ w' &= -\gamma w + \delta v w, \quad w(0) = w_0 = 10, \end{cases} \tag{6.2.6}$$

where v and w represent the number of preys and predators respectively. Parameter α represents the reproduction rate of the preys, β corresponds to the mortality rate of prey

per predator, γ is the mortality rate of predators, and δ the reproduction rate of predator per prey.

Note

It is clear that if there is no interaction between both species ($\beta = \delta = 0$), then the population of the prey increases exponentially while the population of the predator tends exponentially to extinction. A well-known example of a species with no predator is the European rabbit (Oryctolagus cuniculus); see Figure 6.2(a) that was introduced in Australia in 1859 spreading quickly and causing erosion of gullys in the south the country; see Figure 6.2(b) [65, 66, 17].

(a) European rabbit (Oryctolagus cuniculus) (b) Erosion of gullys in South Australia

Figure 6.2: Consequences for the environment caused by a specie with no predator

The system is integrated from $t_0 = 0$ to $t_m = 50$ with a constant step size $h = 0.5$. So our original full-order model is of size 200 (100 for the preys and 100 for the predators).We define this problem on a smaller model, obtained using POD that allows us to identify a (reduced) basis, which we call Φ, that spans a subspace containing the solution. In this case, the reduced model is of size 3 (withunknowns p_1, $p2$, p_3) and is described as follows.

For $i = 1$, we have

$$\left(\sum_{j=1}^{3} \phi(i,j)p_j\right) - v_0$$
$$-h\left(\alpha\left(\sum_{j=1}^{3} \phi(i,j)p_j\right) - \beta\left(\sum_{j=1}^{3} \phi(i,j)p_j\right)\left(\sum_{j=1}^{3} \phi(n+i,j)p_j\right)\right) = 0$$

$$\left(\sum_{j=1}^{3} \phi(n+i,j)p_j\right) - w_0$$
$$-h\left(-\gamma\left(\sum_{j=1}^{3} \phi(n+i,j)p_j\right) + \delta\left(\sum_{j=1}^{3} \phi(i,j)p_j\right)\left(\sum_{j=1}^{3} \phi(n+i,j)p_j\right)\right) = 0$$

where $v_0 = v(0)$, and $w_0 = w(0)$. For $i \in \{2, 3, \ldots, n\}$, with $n = 100$, we have the following equations:

$$\left(\sum_{j=1}^{3} \phi(i,j)p_j\right) - \left(\sum_{j=1}^{3} \phi(i-1,j)p_j\right)$$
$$-h\left(\alpha\left(\sum_{j=1}^{3} \phi(i,j)p_j\right) - \beta\left(\sum_{j=1}^{3} \phi(i,j)p_j\right)\left(\sum_{j=1}^{3} \phi(n+i,j)p_j\right)\right) = 0$$

$$\left(\sum_{j=1}^{3} \phi(n+i,j)p_j\right) - \left(\sum_{j=1}^{3} \phi(n+i-1,j)p_j\right)$$
$$-h\left(-\gamma\left(\sum_{j=1}^{3} \phi(n+i,j)p_j\right) + \delta\left(\sum_{j=1}^{3} \phi(i,j)p_j\right)\left(\sum_{j=1}^{3} \phi(n+i,j)p_j\right)\right) = 0$$

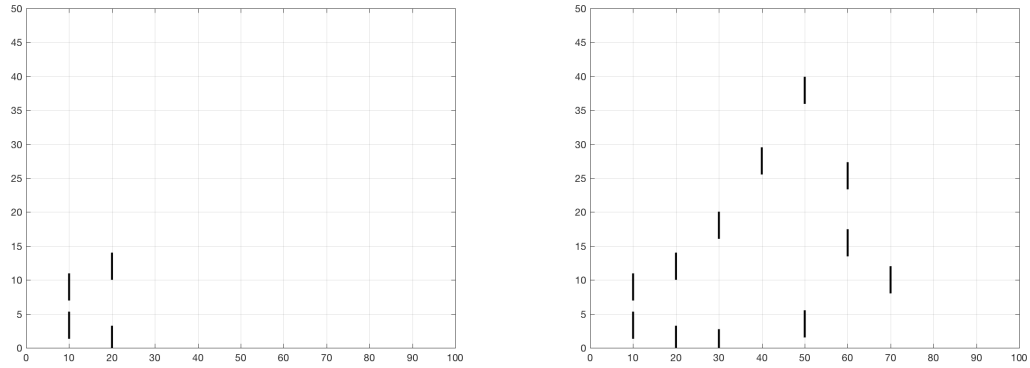
The above system of the equations is the reduced representation (ROM) of the Lotka-Volterra Equation (6.2.6) model. When we use it to understand the its behavior using observations, our unknowns are:

- The coordinates of the solution with respect to the basis of the subspace, i.e., p_1, p_2, and p_3;
- The parameters defining the model α, β, γ, and δ; and
- The initial size of the population of both species v_0 and w_0.

Let us now consider a set of observations. We started with observations at two different times on each of the numbers of preys and predators (hence 4 observations):

$$\begin{aligned} v_{10} &= \sum_{j=1}^{k} \phi(10,j)p_j &&= [6.96, 10.96] \\ v_{20} &= \sum_{j=1}^{k} \phi(20,j)p_j &&= [10.01, 14.02] \\ w_{10} &= \sum_{j=1}^{k} \phi(110,j)p_j &&= [1.32, 5.32] \\ w_{20} &= \sum_{j=1}^{k} \phi(120,j)p_j &&= [0.00, 3.26]. \end{aligned} \tag{6.2.7}$$

We then proceeded to utilize 7 different observations at seven different times (hence in our case, 14 observations). The observed values we picked were computed from the simulation of a Lotka-Volterra system (to ensure that we were picking observations that would yield solutions). The observations are plotted Figure 6.3.

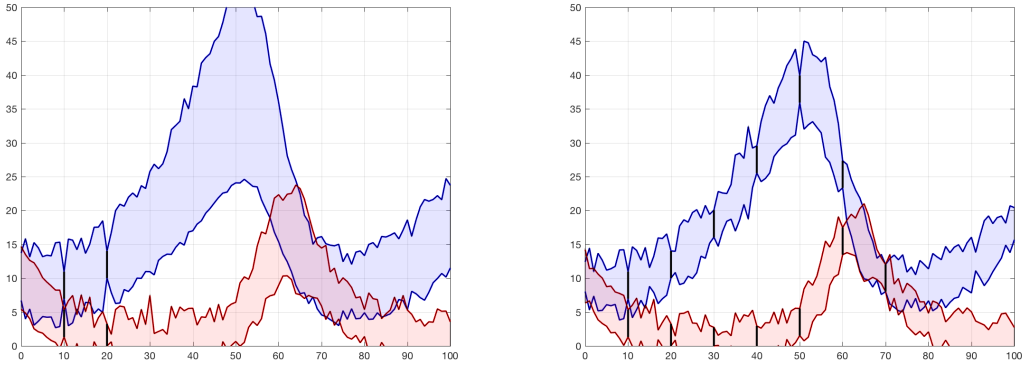


(a) Observations at $t = 10$ and $t = 20$ (b) Seven observations at $t = 10, 20, \ldots, 70$

Figure 6.3: Observed interval data for Lotka-Volterra Problem

The prediction of the Lotka-Volterra behavior was obtained by solving the above-

described reduced system (200 by 9) along with the constraints described in Equation (6.2.7). The results we obtain when solving this prediction problem, for different numbers of observations, are plotted in Figure 6.4



(a) Envelope for four observations

(b) Envelope for 14 obsrvations

Figure 6.4: Envelopes of the bahavior of a Lotka-Volterra system

As we can see from Figure 6.4, we were able to obtain an envelope of the expected behavior of the Lotka-Volterra system, beyond the times of observations. The ragged aspect of the plots is a result of the fact that we solved for p and then recovered v and w using ϕ. Moreover, from the plots, we observe that the envelope suffers from overestimation. We are still obtaining quite a good indication of the expected behavior of the two species over time.

6.2.3 Lotka-Volterra: Three Species

The Lotka-Volterra model is not restricted to only two species, but it can model the interaction of many different species. In this section, we will describe how we can estimate parameters and study the behavior of the trajectory of three different species [86]. The

lowest-level prey is preyed by the middle-level, and the middle is preyed by the top-level[49, 55, 76]. For instance, the ecosystem of mice, snakes and, owls, see Figure 6.5.

(a) Lowest level (Mouse) (b) Middle Level (Snake) (c) Top Level (Owl)

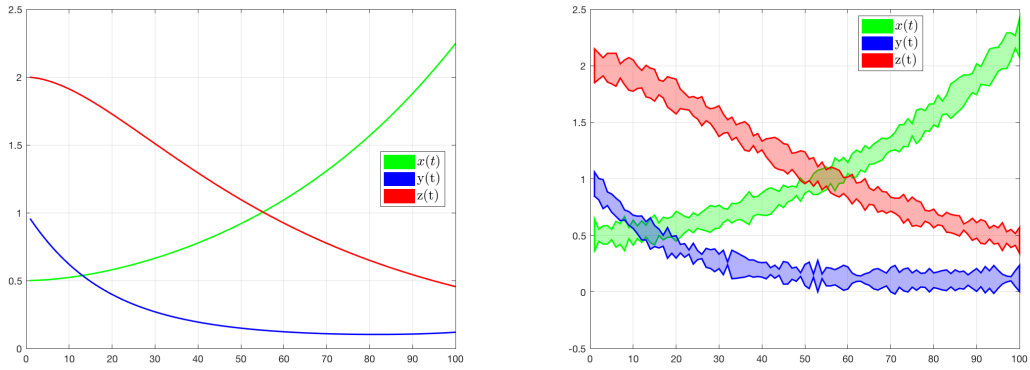
Figure 6.5: Example of a predator-prey system with three species

This ecosystem is modeled with the system of differential equations

$$\begin{cases} \dfrac{dx}{dt} &= ax - bxy \\ \dfrac{dy}{dt} &= -cy + dxy - eyz \\ \dfrac{dz}{dt} &= -fz + gyz \end{cases} \tag{6.2.8}$$

Let us discretize the domain $t \in [0, 4]$, with $h = 0.04$. Let us use finite difference to approximate the derivative. First, we solve Equation (6.2.8) with no uncertainty on the parameters. Second, we fix the initial conditions to see if we are able to handle uncertainty in all the parameters.

We fix the initial conditions, and we solve Equation (6.2.8) for $a = b = c = d = e = f = g = 1.0$ Figure 6.6(b).



(a) Solution with no uncertainty on the parameters

(b) Solution with uncertainty on the parameters

Figure 6.6: Solution of Food-Chain model with uncertainty and with no uncertainty on the parameters.

6.3 Conclusion

In this chapter, we proved that we were able to handle uncertainty in dynamical systems parameter-dependent. We also proved that in some cases we are able predict the behavior of such dynamic systems and estimate the parameter leading to such behavior if we have available some observations.

A particular case that has to be studied in detail is the Food-Chain model Equation (6.2.8). In this case, we were able to handle uncertainty in all the parameters, but we were not able to reduce the overestimation when we tried to estimate the parameters given some observations. We left as future work to find out what are the necessary conditions the parameters have to satisfy to be able to reduce the overestimation.

Chapter 7

Parameter Identification for Dynamic Systems Control

What I'm doing is not really based on a definite identification or a definition of what it is. It's intended to be open to interpretation.

Reggie Watts

In this chapter, we aim to address situations in which an unfolding dynamic phenomenon, for which we know all input parameters and other properties, is perturbated and requires recomputation of some parameters [122] so as to ensure that some properties be satisfied (e.g., the below helicopter example where the landing zone is guaranteed even after perturbation, see Figure 7.1). In general, if we priori restrict ourselves to a lower-dimensional space, we only get an approximation solution.

In the event of a perturbation, observations are essential to understanding the perturbation but observations are inherently inaccurate. As a result, if we are to solve such problems, we need to handle and quantify uncertainty to assess the quality of our solu-

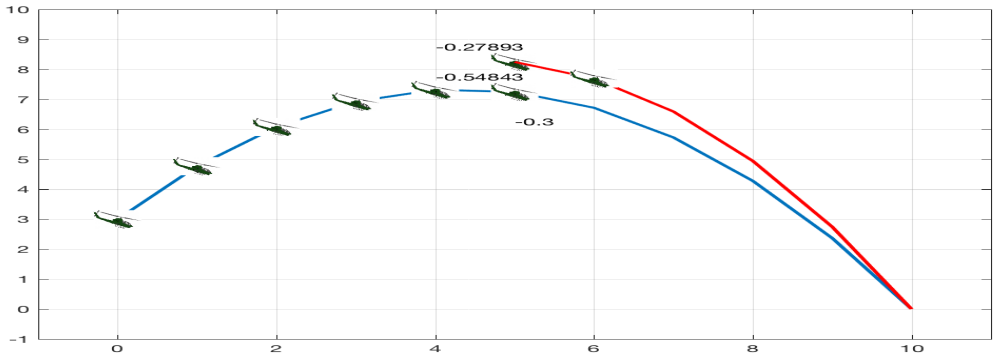


Figure 7.1: Parameters of the flight are reliably recomputed to reach the landing zone after perturbation

tions. We use interval computations to handle uncertainty.

7.1 Problem Statement and Proposed Approach

Let us recall the problem we want to solve. Using the model of a dynamic system, we aim to find certain parameter values that guarantee a specific outcome of the modeled dynamic phenomenon.

Assuming this phenomenon is modeled as a parametric differential equation (ODE / PDE), the parameters that lead to a certain outcome can be found as follows:

1. Discretize the ODE/PDE equation leading to a parametric system of equations $F(X, P) = 0$, where $X = (x_1, x_2, \ldots, x_n)$ is the approximation of the solution and P are the parameters of the ODE/PDE equation, and $F : \mathbb{R}^n \to \mathbb{R}^n$.

2. Let $[i_1, i_2, \ldots, i_m]$ a subset of $[1, 2, \ldots, n]$ where n is the dimension. Fix the values

of $x_{i_j} = [\underline{x_{i_j}}, \overline{x_{i_j}}]$, with $j = 1, 2, \ldots, m$ representing the expected outcomes. Solve for P using ICST the following system:

$$\begin{aligned} \Phi(i_1,:)Y = x_{i_1} &= [\underline{x_{i_1}}, \overline{x_{i_1}}] \\ \Phi(i_2,:)Y = x_{i_2} &= [\underline{x_{i_2}}, \overline{x_{i_2}}] \\ \ldots &\quad \ldots \\ \Phi(i_m,:)Y = x_{i_m} &= [\underline{x_{i_m}}, \overline{x_{i_m}}] \\ F(\Phi Y, P) &= 0 \end{aligned} \tag{7.1.1}$$

Where $\Phi(i_j,:)$ is the i_j-row of Φ. The solutions P correspond to the sought parameters.

7.2 Numerical Experiments

In this section, we report on preliminary experiments of our approach on the Lotka-Volterra model when the rate of reproduction of the prey coincides with the rate of interaction with the predator. Similarly, the rate of death of the predator when there is not interaction with the prey coincides with the of interaction with the prey.

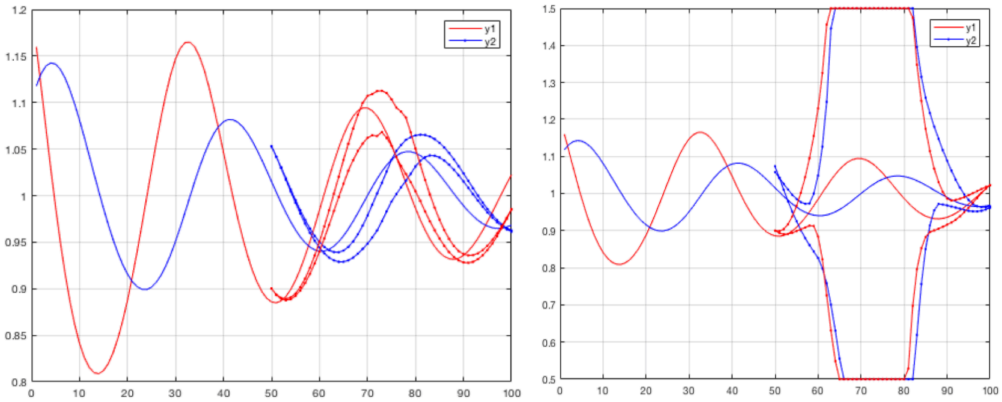
7.2.1 Lotka-Volterra Model

Consider the Lotka-Volterra problem when $\alpha = \beta = \theta_1$ and the parameters $\gamma = \delta = \theta_2$. We use the following equations to describe this problem:

$$\begin{cases} v' = \theta_1 v(1 - w), & v(0) = v_0 = 1.2 \\ w' = \theta_2 w(v - 1), & w(0) = w_0 = 1.1 \end{cases} \tag{7.2.1}$$

where v and w respectively represent the number of preys and predators represented in thousands. The system was integrated from time $t_0 = 0$ to $t_m = 10$ with a constant step

size $h = 0.1$. We used $\theta_1 = 3$ and $\theta_2 = 1$. Let us assume that at $t = 5$, a perturbation occurs, which changes the number of predators and preys. Since, it is not possible to know the new real number of animals of each species, the new number of both species is handled with uncertainty.



(a) Trajectory 1 (b) Trajectory 2

Figure 7.2: Perturbation of a dynamical system on the midtime before an experiment ends

In Figure 7.2, we can see two kind of perturbations. On one hand, the perturbation and the final goal are consistent, which means Equation (7.1.1) is satisfied for some parameters, and as consequence, we have a little of overestimation of the trajectories for both species Figure 7.2(a). On the other hand, in Figure 7.2(b), are not consistent and there is no way we can know the behavior of both population before the end point. We are confident that the population of both species will reach the final condition, please observe the final values of the trajectories en Figure 7.2(b).

7.3 Conclusion

We aimed to design a technique that allows the identification of parameters of a given (known and observed) dynamic system that has been perturbated, in such a way that some final conditions still hold. We used Reduced-Order Modeling and interval constraint solving techiques to determine such values of the phenomenon's parameters.

We were able to identify reliable intervals in which the desired parameters' values lie. We improved the runtime of this method by using ROM.

Chapter 8

Interval Finite Element Method

You see the world as fixed and finite, and it is not. It is liquid and ever moving, and one act can change everything.

A.C. Gaughen

Real fuels are chemically complex and often contain thousands of hydrocarbon compounds. The refining industry requires these fuels to meet general chemical/physical property specifications, such as boiling range, heat of combustion, and freezing. This results in an intractable real fuel composition making it difficult to ensure reproducible results. Due to this variability, combustion simulations can only be performed based on a "surrogate" that limits the number of compounds in the blend to enforce fuel composition tractability. Capturing the uncertainty in combustion simulations is of great interest not only for fundamental research but also for designers to evaluate the behavior of novel fuels to engine performance.

When solving or simulating the reality of physical phenomena, the Finite Element Method (FEM) often constitutes a method of choice. However, uncertainty is hardly handled and

when it was, only the linear case was mentioned, or in the case of stochastic approaches, the number of simulations can be prohibitive or the results not 100% guaranteed.

In this chapter, we propose a novel technique to handle uncertainty in FEM using interval computations and Interval Constraint Solving Techniques. We then demonstrate the performance of our work on two problems: a static convection-diffusion problem and a transitory nonlinear heat equation as a first step towards our goal of fuel simulation.

8.1 Handling Uncertainty in FEM

8.1.1 Our Approach

In this section, we present our algorithm for handing uncertainty when a PDE is being solved using FEM.

The general procedure is as follows:

- Discretize the domain into finite elements;
- Write the PDE in its weak form;
- Compute the contribution of each element to the respective constraint;
- Implement the boundary conditions;
- Set the parameters with uncertainty; and
- Solve the system of equations using ICST.

8.1.2 Example

Let us consider the equation:

$$\begin{aligned} -\Delta u &= -4 \\ u(x,y) &= [0.8(x^2+y^2), 1.2(x^2+y^2)] \text{ on } \partial\Omega \end{aligned} \tag{8.1.1}$$

The domain $\Omega = [0,1] \times [0,1]$.

Step 1. Discretize the domain in finite elements, as show in Figure 3.1(a).

Step 2. Write (8.1.1) in its weak form:

$$\int_\Omega \nabla u \nabla v = -4 \int_\Omega v \tag{8.1.2}$$

Step 3. Compute the contribution of each element to the respective constraint:

Element 1: $P_1 = (0,1)$, $P_4 = (0,0)$, $P_5 = (1/6, 4/6)$; $Area(T_1) = 0.0833$

$$\nabla v_1|_{T_1} = \begin{pmatrix} -4 \\ 1 \end{pmatrix}, \quad \nabla v_4|_{T_1} = \begin{pmatrix} -2 \\ -1 \end{pmatrix}, \quad \nabla v_5|_{T_1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{8.1.3}$$

The constraints after the integrals are calculated in the first element are as follow:

$$
\begin{aligned}
c1: &\quad 1.4167x_1 + 0.5833x_4 - 0.333x_5 &&= -0.1111,\\
c2: &\quad 0 &&= 0,\\
c3: &\quad 0 &&= 0,\\
c4: &\quad 0.5833x_1 + 0.4167x_4 - 0.1667x_5 &&= -0.1111\\
c5: &\quad -0.3333x_1 - 0.1667x_4 + 0.0833x_5 &&= -0.1111,\\
c6: &\quad 0 &&= 0,\\
c7: &\quad 0 &&= 0;
\end{aligned}
\tag{8.1.4}
$$

Element 2: $P_1 = (0,1)$, $P_2 = (1,0)$, $P_5 = (1/6, 4/6)$; $Area(T_1) = 0.1667$

$$
\nabla v_1|_{T_2} = \begin{pmatrix} -1 \\ 2.5 \end{pmatrix}, \quad \nabla v_2|_{T_2} = \begin{pmatrix} 0 \\ -3 \end{pmatrix}, \quad \nabla v_5|_{T_2} = \begin{pmatrix} 0.3333 \\ 0.1667 \end{pmatrix} \tag{8.1.5}
$$

The constraints are updated as follows:

$$
\begin{aligned}
c1: &\quad 2.6250x_1 - 1.25x_2 + 0.5833x_4 - 0.3191x_5 &&= -0.3333,\\
c2: &\quad -1.25x_1 + 1.5x_2 - 0.0833x_5 &&= -0.2222,\\
c3: &\quad 0 &&= 0,\\
c4: &\quad 0.5833x_1 + 0.4167x_4 - 0.1667x_5 &&= -0.1111,\\
c5: &\quad -0.3194x_1 - 0.0833x_2 - 0.1667x_4 + 0.1064x_5 &&= -0.3333,\\
c6: &\quad 0 &&= 0,\\
c7: &\quad 0 &&= 0;
\end{aligned}
$$

We repeat this procedure until **Element 8** and the resulting final system is:

$$
\begin{aligned}
c1: &\quad 2.6250x_1 + 0.0417x_2 + 0.5833x_4 - 3.2500x_5 && = -0.3333,\\
c2: &\quad 0.0417x_1 + 1.2045x_2 + 0.2083x_3 - 0.1136x_5 - 1.3409x_6 && = -0.6481,\\
c3: &\quad 0.2083x_2 + 2.1050x_3 + 0.6367x_4 - 0.6500x_6 - 2.3000x_7 && = -0.4296,\\
c4: &\quad 0.5833x_1 + 0.6367x_3 + 1.5681x_4 - 1.2373x_5 - 1.5507x_7 && = -0.4704,\\
c5: &\quad -3.2500x_1 - 0.1136x_2 - 1.2373x_4 + 5.9368x_5 - 1.1148x_6 - 0.2211x_7 && = -0.9037,\\
c6: &\quad -1.3409x_2 - 0.6500x_3 - 1.1148x_5 + 4.6994x_6 - 1.5938x_7 && = -0.6407,\\
c7: &\quad -2.3000x_3 - 1.5507x_4 - 0.2211x_5 - 1.5938x_6 + 5.6656x_7 && = -0.5741;
\end{aligned}
$$

Step 4. and Step 5. Implement the boundary conditions and set the parameters with uncertainty: $x_1 = [0.9, 1.1]$, $x_2 = [1.8, 2.2]$, $x_3 = [0.9, 1.1]$, $x_4 = [0, 0]$

$$
\begin{aligned}
c1: &\quad 2.6250[\mathbf{0.9}, \mathbf{1.1}] + 0.0417[\mathbf{1.8}, \mathbf{2.2}] - 3.2500x_5 && = -0.3333,\\
c2: &\quad 0.0417[\mathbf{0.9}, \mathbf{1.1}] + 1.2045[\mathbf{1.8}, \mathbf{2.2}] + 0.2083[\mathbf{0.9}, \mathbf{1.1}] - 0.1136x_5 - 1.3409x_6 && = -0.6481,\\
c3: &\quad 0.2083[\mathbf{1.8}, \mathbf{2.2}] + 2.1050[\mathbf{0.9}, \mathbf{1.1}] - 0.6500x_6 - 2.3000x_7 && = -0.4296,\\
c4: &\quad 0.5833[\mathbf{0.9}, \mathbf{1.1}] + 0.6367[\mathbf{0.9}, \mathbf{1.1}] - 1.2373x_5 - 1.5507x_7 && = -0.4704,\\
c5: &\quad -3.2500[\mathbf{0.9}, \mathbf{1.1}] - 0.1136[\mathbf{1.8}, \mathbf{2.2}] + 5.9368x_5 - 1.1148x_6 - 0.2211x_7 && = -0.9037,\\
c6: &\quad -1.3409[\mathbf{1.8}, \mathbf{2.2}] - 0.6500[\mathbf{0.9}, \mathbf{1.1}] - 1.1148x_5 + 4.6994x_6 - 1.5938x_7 && = -0.6407,\\
c7: &\quad -2.3000[\mathbf{0.9}, \mathbf{1.1}] - 0.2211x_5 - 1.5938x_6 + 5.6656x_7 && = -0.5741;
\end{aligned}
$$

Step 6. Solve the system using ICST :

$$
\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} [0.8999999999999999, 1.100000000000001] \\ [1.799999999999999, 2.200000000000001] \\ [0.8999999999999999, 1.100000000000001] \\ [0.0, 0.0] \\ [0.5700639986955393, 0.6877342144440121] \\ [0.8692719389895366, 0.9738156052408669] \\ [0.5370470020083591, 0.6398964257062801] \end{pmatrix}
$$

8.2 Experimental Results

In this section, we apply our approach to a stationary and a transitory problems, expressed as partial differential equations, to measure our ability to handle uncertainty in a time independent case as well as in a time dependent case. The system of equations coming from the discretization is solved using Realpaver [14, 43, 42, 113].

8.2.1 Handling Uncertainty in a $2-\mathrm{D}$ convection-diffusion problem

The first example to test out implementation is based on the boundary condition in a 2-D convection-diffusion problem [74, 12, 11].

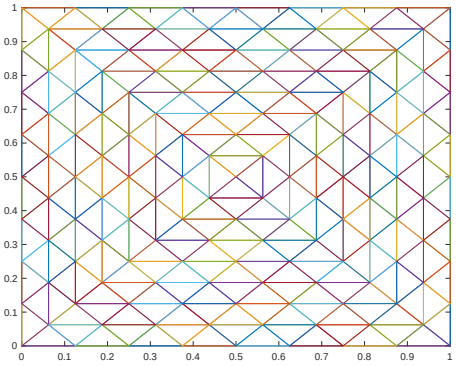
Example 8. *Consider the 2-D convection-diffusion problem on* $\Omega = [0,1] \times [0,1]$*:*

$$-\Delta u + v \cdot \nabla u = -2y^2 - 2x^2 + 2xy^2 + 2yx^2 \tag{8.2.1}$$

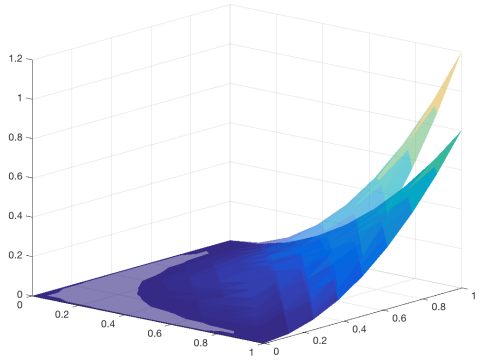
$$u(x,y) = [0.8x^2y^2, 1.2x^2y^2] \ \textit{on}\ \partial\Omega \tag{8.2.2}$$

We solved (8.2.3)-(8.2.4) discretizing the physical domain into finite triangular elements, and using ICST, we solved the resulting system of equations, see Figure. 8.1(a) and Figure. 8.1(b). Since this is a stationary model, we only need to solve the system of equation once, so the uncertainty on the boundary conditions (8.2.4) leads to an interval solution represented here as two surfaces (lower and upper), see (8.1(b)).

Let us retake the example studied in example 6, but this time, we consider uncertainty on the boundary condition:



(a) Triangular finite elements



(b) Solution of a convection-diffusion problem using triangular finite elements

Figure 8.1: Solution of a convection-diffusion problem in a triangular finite domain Ω

Example 9. *A 2-D poisson problem on* $\Omega = [0,1] \times [0,1]$*:*

$$-\Delta u = 4\pi^2 \sin(2\pi x)\sin(2\pi y) \tag{8.2.3}$$

$$u(x,y) = [-0.2, 0.2];\ \forall (x,y) \in \partial\Omega \tag{8.2.4}$$

We discretize the domain and set the equations as we did in Chapter 3, but we solve the equations using Interval Constraint Solving Techniques.

In Figure 8.2, we can see clearer than in the previous example how the solution is given as en envelope for all the solutions corresponding to the boundary condition in the interval $[-0.2, 0.2]$. The blue part corresponds with the lower bound of the interval solution and the red one with the upper bound.

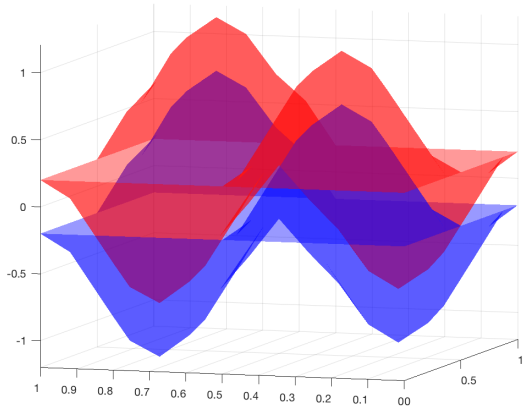


Figure 8.2: Interval Solution of the Poisson equation with uncertainty on the boundary condition.

Finally, we see again the transitory studied in example 7, and the uncertainty is present in the initial condition. We present it in what follows:

8.2.2 Transitory nonlinear heat transfer equation

We study the transient nonlinear heat conduction [41, 31, 91] in a given body:

Example 10. *Consider the transitory partial differential equation:*

$$\begin{aligned} \rho C_p \frac{\partial u}{\partial t} &= \nabla \cdot (k \nabla u) + Q_u \ \textit{on}\ \Omega \times [0, T] \\ u &= g \ \textit{on}\ \partial\Omega \times [0, T] \\ u(x, 0) &= T_0, \ \forall x \in \Omega \end{aligned} \tag{8.2.5}$$

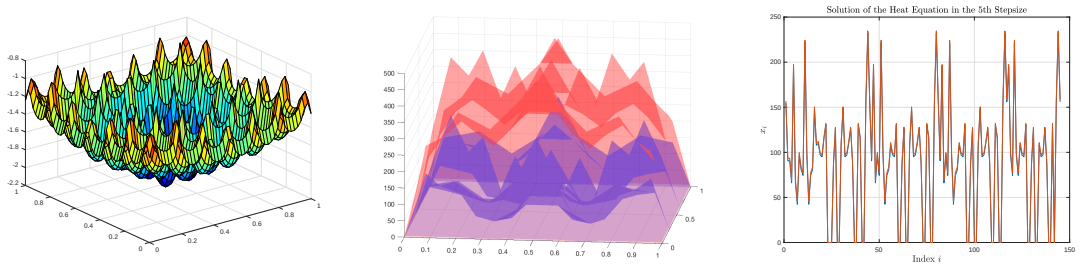
In (8.2.5), $u = u(x,t)$ *is the temperature,* $C_p = 50.0$ *is the heat capacity to constant preassure and* $k = k(u) = 2\left(1 + \frac{5u}{1000}\right)$ *is the thermal conductivity, and* $\rho = 1$ *is the density of the material. The function* $Q_u = Q_0 Q(x,y)$ *represents heat sources, where*

$$f(\tilde{x},\tilde{y}) = \left(\left|\sin(\tilde{x})\sin(\tilde{y})\exp\left(\left|100 - \frac{\sqrt{\tilde{x}^2+\tilde{y}^2}}{\pi}\right|\right)\right| + 1\right)^{0.1}$$

$$Q(x,y) = -10^{-4}\frac{f(\tilde{x},\tilde{y}) - f_{min}}{f_{max} - f_{min}}$$

$$\tilde{x} = 20x - 10;\ \tilde{y} = 20y - 10$$

The function $f(\tilde{x},\tilde{y})$ is known as the Cross-in-Tray function [81], see Figure. 8.3(a).



(a) The heat generation function

(b) Interval solution of transitory heat equation using triangular finite elements

(c) Solution expressed in row-wise order

Figure 8.3: Solution of transient nonlinear heat conduction using triangular finite elements

In Problem (8.2.5), the derivative in time is approximated by finite differences, i.e., $\frac{\partial u}{\partial t} = \frac{u^i - u^{i-1}}{\Delta t}$. Uncertainty is present in both the boundary condition $u = g = [0, 0.01]$ on $\partial\Omega \times [0,T]$ and the initial condition $T_0 = [0, 0.01]\ \forall\ x \in \Omega$. Using ICST (8.2.5) for $t_i = t_{i-1} + \Delta_t$, with $\Delta_t = 0.005$ and $i = 1, 2, \cdots, 10$.

8.3 Conclusion

In this chapter, we were able to present a interval version of the Finite Element. This new approach allows us to handle uncertainty in problems that normally are solved using FEM.

We illustrated our approach with three examples. Two of them depended on the space domain and the last one, the transient problem, depended on both, space and time.

The fact that Problem (8.2.5) is a transitory model makes it very challenging. The system of equations coming from the triangularization (nonlinear in this case) must be solved several times, once for each stepsize. The solution at each instant of time carries on (or propagates) with the uncertainty of the previous realization, and it is passed on to the next one. Uncertainty is overestimated until it is not possible to handle it anymore. This behavior of uncertainty can be observed in Figure. 8.3(c), where the solution is plotted column-wise.

The solution at the 5th stepsize of Problem (8.2.5) is plotted on the top part of Figure. 8.3(c): in particular, we can barely observe any difference between the lower bound and the upper of the interval solution. However, when reaching even only the 9th instant of time, the difference between the lower and the upper bound is already significant. The uncertainty propagates and an uncertain situation usually leads to an even more uncertain future: this is what we observe. In real applications, where observations of a phenomenon are available in real time, observations allow to disambiguate a situation where uncertainty has exploded via a few propagation steps. The interval solution for $t = 0.05$ is presented in Figure. 8.3(b).

Let us observe how Figure 8.3(b) shows two surfaces: the first one, the upper surface, represents the upper bound of the interval solution, and underneath it, the second one, the lower surface, represents the lower bound. These two surfaces constitute an envelope of all possible solutions to the problem at hand. Despite the uncertainty of having solutions lie within such a thick envelope, it is important to stress that we can guarantee that all solutions were identified, which means that we can also certify that no solution lies outside of the envelope. This is very important when trying to avoid a scenario: if it is outside the envelope, it is indeed guaranteed to be avoided.

Chapter 9

Applications: Mobile Apps

There's no more important consumer product today than a cell phone.

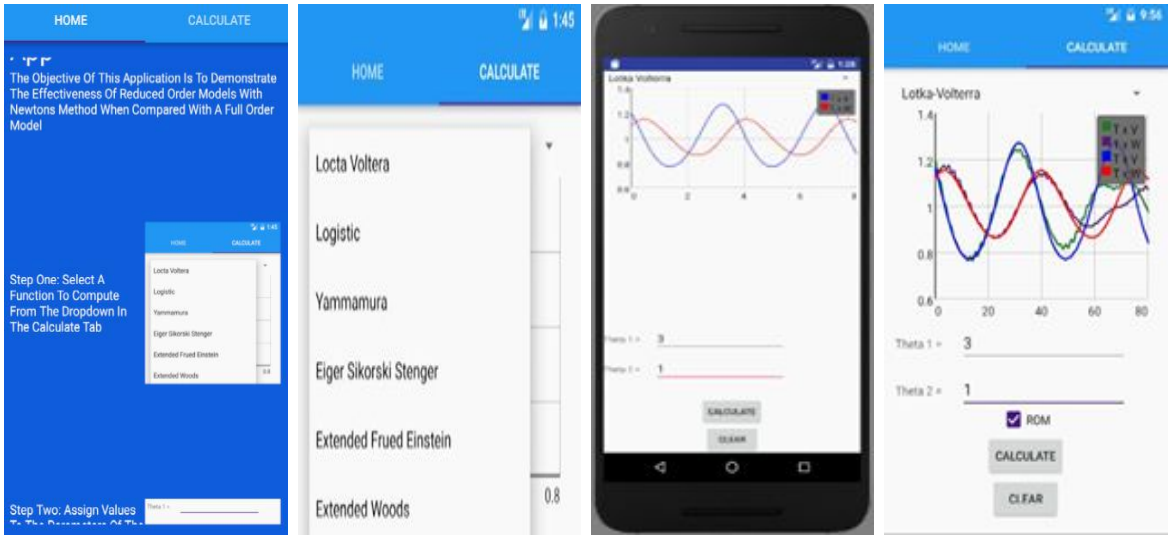
Mary Dillon

There exists situations where simulations or experiments have to be executed while neither high performance computers nor internet connection is available. For example, soldiers that need to take an immediate decision in an unexpected situation or engineers on a field work.

In this thesis, we used the Reduced-Order Model techniques to design an application for mobile devices. This application was designed in two stages. The first stage was developed to measure our ability to handle large dimensions on a nonlinear system of equations, and for the second stage, we use ROM and Interval Constraints Solving Techniques to estimate parameters and unfold future behavior of dynamical systems.

9.1 Full Order Model Vs Reduced Order Model

The beta version of the application(Figure 9.1) was designed to solve default problems that were previously loaded on it; see Figure 9.1(b). The users have to choose the problem they want to solve and only to insert the parameter associated with the model; see Figure 9.1(c). In Figure 9.1(d), we can see the comparison of FOM versus ROM method.



(a) Initial Page (b) Preloaded Problems (c) FOM solution of a preloaded problem (d) FOM vs Rom solution of a preloaded problem

Figure 9.1: FOM vs ROM on a mobile device

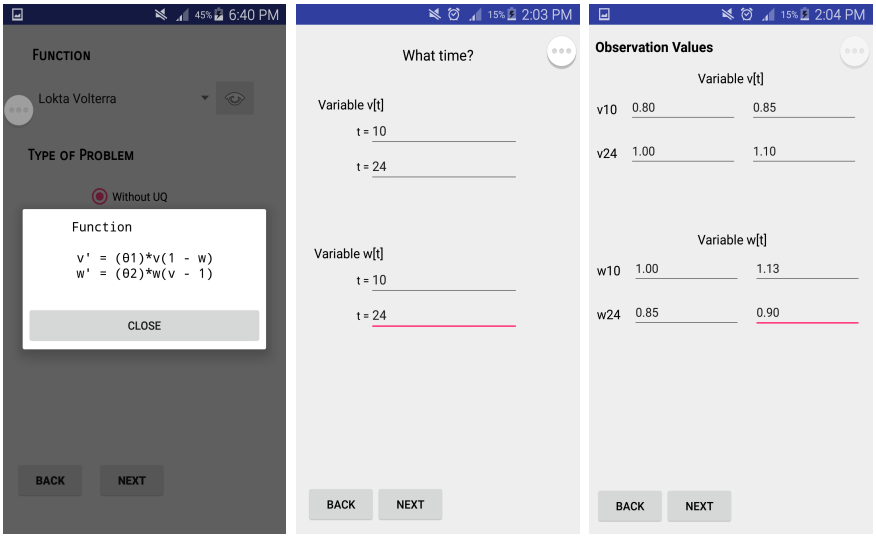
9.2 Interval Constraint Solving Techniques on a Mobile Device

The second version of our application was more complete, but still needs to be improved. We can run simulations or estimate parameters given some observations on some particular instant of times. To handle uncertainty on our application, we needed to develop our own interval library on Java to run our programs on Android, which is the operating system of the mobiles we were working on. As future work we are planning to export our library to IOS to be able to run our application on IPhones.

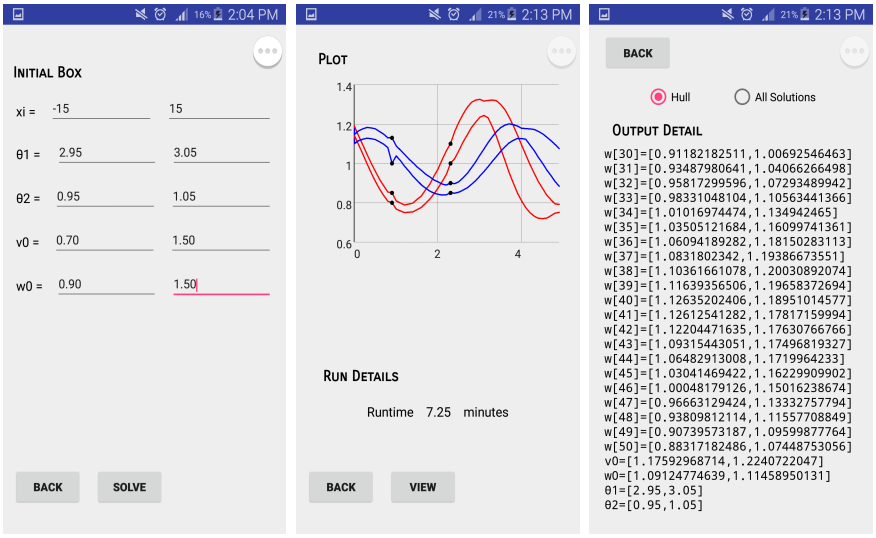
Given a model, input parameters, and observations with uncertainty (given as intervals) at several points in time we are able to run ROM on the app to estimate the parameters that lead to such observations, and we were able to unfold the future behavior of the phenomenon on our tablet as illustrated in Figure 9.2.

Let us describe in detail how to use our application to estimate parameter given some observations with uncertainty:

1. we selected the problem we want to solve from a collection of preloaded models (Figure 9.2(a));
2. select the instant of time where to set the observations. In this version, we have up to three observations. (Figure 9.2(b)).
3. We set the values of the observations corresponding to t set in the previou step (Figure 9.2(c)), and we click `NEXT` to obtain the results.
4. set the initial box for the variables of our problem. In this version of our application,



(a) Selecting the problem (b) Setting the instants of time (c) Setting the observations



(d) Setting the initial box (e) Plot of the interval solution (f) Output of the interval solution

Figure 9.2: Prediction and parameter estimation on a mobile device.

the initial box has to be cube, i.e., $x_i \in [a, b]$ for all $i \in \{1, 2, \ldots, n\}$, where $[a, b]$ is a constant interval (Figure 9.2(d));

5. We have the plot of the solution (Figure 9.2(f)).

6. Finally, we can show the result of the problem. This solution can be presented as a union of intervals or a hull of them (Figure 9.2(f)).

With this, we finish this short chapter about the application we developed to mobile devices.

Part III

Conclusions and Future Work

UTEP 2018

Chapter 10

Conclusion and Future Work

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

Helen Keller

In this thesis, we aimed to study the behavior of dynamic systems. The most of the dynamical systems are solved numerically using finite differences and to obtain an acceptable accuracy of the approximation of the solution, we have to do a finer discretization of the domain. This discretization leads to a large and usually nonlinear system of equations.

To reduce the runtime computing the solution, the Proper Orthogonal Decomposition (POD) method is frequently used to reduced the space where the solution is sought (ROM). This method is based on principal component analysis, and it used the concept of inputs and snapshots. The inputs used to compute the snapshots that will be used to compute the basis of the reduced subspace are assumed that they are free of uncertainty, which is real life is not true.

We proposed and described a novel (ROM) approach that improves the well-known Proper Orthogonal Decomposition method by:

1. freeing traditional ROM techniques from snapshot identification,
2. providing reliable solutions, and
3. allowing to handle uncertainty. Our new approach is based on the use of Interval analysis and Interval Constraint Solving Techniques. We called this new method the Interval Proper Orthogonal Decomposition (I-POD). We tested I-POD on five nonlinear partial differential equations problems: Burgers' equation, the Transport equation, Lotka-Volterra problem, the FitzHugh-Nagumo Model, and the Bratu's Problem. We observed and reported promising performance of I-POD, when compared to POD.

Our ability of solving large dynamical systems while handling uncertainty using Reduced-Order Modeling allows us to address the problem of predicting the behavior of an unfolding dynamic event under observation. This problem rose challenges, namely handling uncertainty and the size of the original problem. We demonstrated that using Reduced-Order Modeling techniques along with Interval Constraint Solving techniques, we were able to tackle this problem of prediction. Our experimental results showed that our model worked properly.

As future work, we plan to take observation outliers into account to make our approach more realistic. In addition, at this point, when we go from the full-order model to the reduced model, we only reduce the number of un- knowns but we do not reduce the

number of constraints. In truly large systems (even starting at hundreds of thousands of constraints), this can certainly be a problem. Prior attempts to deterministically reduce the number of constraints resulted in systems that were really hard to solve (limiting the computational time advantage of reducing the order). We now plan to devise strategies not to use all constraints at each step of computation so as to lighten the solving process, but we need to study the tradeoff with quality. Finally, we plan to apply the approach proposed in this article to the problem of fuel combustion performance where the mix that goes into the fuel is usually uncertain.

We also aimed to design a technique that allows us to identify parameters of a given (known and observed) dynamic system that has been perturbed, in such a way that some final conditions still hold. We used Reduced-Order Modeling and interval constraint solving techniques to determine such values of the phenomenon's parameters. We were able to identify reliable intervals in which the desired parameters' values lie.

Starting with the need to handle uncertainty in a complex fuel problem, we proposed a Finite Element Method able to handle uncertainty, that integrated interval computations and constraint solving techniques. We tested it on preliminary problems to gauge its feasibility. We obtained promising results of our new approach for the static convection-diffusion and the transitory nonlinear heat equation problems. In both cases, we were able to identify a reliable solution: an interval solution that was actionable (not overly uncertain).

Future work includes taking into account the recomputation time and not assuming that new behavior can be "plugged" directly from where perturbation happened. As a result,

more uncertainty needs to be taken into account, which includes time uncertainty.

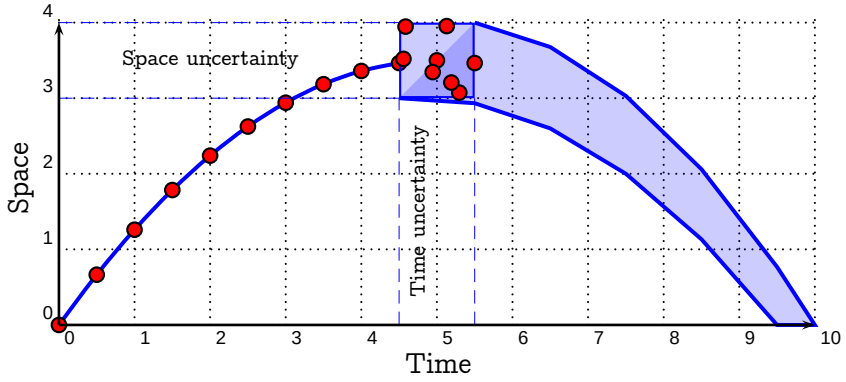


Figure 10.1: Space and Time uncertainty on a trajectory simulation

Additionally, we plan to consider perturbations as fuzzy numbers. This will require us to consider our dynamic system with uncertainty differently (e.g., with fuzzy derivatives). But most importantly, this is expected to help us make more informed decisions: if we can label our input uncertainty with fuzzy values, how does this inform us about labels on uncertain solutions to focus on the best ones? [33].

Future work includes working on PDEs with uncertainty in parameters other than the initial and boundary conditions. Immediate work includes enhancing the JP-8 surrogate model with uncertainty and identifying uncertain parameters to solve it with.

Our ability of handling large system of equations and uncertainty allows us to developed an application that can run on mobile devices. This application is very useful when decisions have to be taken on field and we have neither access to a fast computer nor internet conection. The results obtained with these kind of mobile devices could be not as accurate as we want, but it could be enough to make a decision in critical situations. We are planing, as future work, to export our library to IOS to be able to run our application

on IPhones.

Part IV

Appendices

UTEP 2018

Chapter 11

Background in Vector and Metric Spaces

Celestial mechanics is the origen of dynamical systems, linear algebra, topology, variational calculus, and symplectic geometry.

Vladimir Arnold

Many concepts used in this thesis have their support in linear algebra and in mathematical analysis, for example: subspaces, inner product, density of subsets, compactness, etc. In this chapter, which is part of the appendices, we introduce the notion of space vector over a field. We lead the reader through the basic concepts such as subspace, basis, and dimension. The key concept of inner product is defined, and therefore, the concept of a metric space is introduced.

11.1 Basic Concepts of Vector Spaces

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

Definition 1. *A* **vector space over a field F** *consists of a set* V*, a closed binary operation*
$+ : V \times V \to V$*, satisfying:*

1. $u + v = v + u$*, for all* $u, v \in V$*,*
2. $u + (v + w) = (u + v) + w$*, for all* $u, v, w \in V$*,*
3. *for all* $v \in V$*, there is an element* $\mathbf{0}$ *in* V*, such that* $v + \mathbf{0} = v$*,*
4. *for each element* $v \in V$ *there is a unique element* $-v$ *in* V *such that* $v + (-v) = \mathbf{0}$*,*

and a scalar multiplication, $* : \mathbf{F} \times V \to V$*, such that*

1. *for all* $v \in V$ *and* $1 \in \mathbb{F}$*,* $1 * v = v$*,*
2. *for all* $\alpha, \beta \in \mathbb{F}$ *and* $v \in V$*,* $(\alpha\beta) * v = \alpha * (\beta * v)$*,*
3. *for all* $\alpha, \beta \in \mathbb{F}$ *and* $v \in V$*,* $(\alpha + \beta) * v = \alpha * v + \beta * v$*,*
4. *for all* $\alpha \in \mathbb{F}, v, w \in V$*,* $\alpha * (v + w) = \alpha * v + \alpha * w$*.*

In short, a **vector space over a field F** is a 4-tuple $(V, +, F, *)$, where $(V, +)$ is an Abelian group, and the action of the field F over V, $\alpha * v$, is denoted by αv. Each element of V is called **vector**, and the elements of F are called **scalars** [93, 67, 53].

The set $\mathbb{R}^n$, the set of continuous functions, and the set of derivable functions are examples of vector spaces.

Let us introduce another example of a vector space in the following:

Proposition 1. *Consider a closed polygon, Ω, in the plane $\mathbb{R}^2$. Let us denote its boundary Γ. The space of the functions*

$$L^2(\Omega) = \left\{ f : \Omega \to \mathbb{R} \;\middle|\; \int_\Omega |f|^2 < \infty \right\} \tag{11.1.1}$$

is a vector space [2, 94, 23, 106].

Proof. The addition of two elements in $L^2(\Omega)$ is defined according to the following rule

$$(f + g)(x) = f(x) + g(x)$$

and the scalar multiplication is defined as

$$(\alpha f)(x) = \alpha f(x)$$

The above-defined operations satisfy all the axioms that define a vector space. We only prove the closure of the addition. Given $f, g \in L^2(\Omega)$,

$$\begin{aligned} \int_\Omega |f + g|^2 &= \int_\Omega |f + g| \cdot |f + g| \\ &\leq \int_\Omega (|f| + |g|) \cdot (|f| + |g|) \\ &= \int_\Omega |f|^2 + |g|^2 + 2|f||g| \end{aligned}$$

since

$$0 \leq (|f| - |g|)^2 = |f|^2 + |g|^2 - 2|f||g|,$$

it follows that

$$2|f||g| \leq |f|^2 + |g|^2$$

$$\int_\Omega |f|^2 + |g|^2 + 2|f||g| \leq \int_\Omega 2(|f|^2 + |g|^2) < \infty \tag{11.1.2}$$

with (11.1.2), we conclude our proof. ■

Let us continue introducing the concept of subspace.

Definition 2. *Given two spaces W and V over a field $\mathbf{F}$, the space W is said to be* **subspace** *of V if $W \subseteq V$.*

Note

From now on, we will write the terms "subspace" and "space", omitting the expression "over a field $\mathbf{F}$".

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

Proposition 2. *(Subspace criterion) Let W be a nonempty subset of the vector space V. The subset W is a subspace of V if for each pair of vectors v, w in W and each scalar α in $\mathbf{F}$, the vector $\alpha v + w \in W$.*

The reader can find a general proof in [53].

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

Definition 3. *Let $v_1, v_2, \ldots, v_n$ be vectors in a vector space V, the expresion of the form*

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n,$$

where $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbf{F}$, *is called a* ***linear combination*** *of the vectors* $v_1, v_2, \ldots, v_n$. *The set of all such linear combinations is called the* **span** *of* $v_1, v_2, \ldots, v_n$, *and it is denoted by* $\text{Span}\{v_1, v_2 \ldots, v_n\}$ *[50].*

We can extend the notion of span to a set S of vectors in a vector space V. In this case, $\text{Span}(S)$ 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, and we also say that this set spans W.

Let us see when we can get rid of a vector from a subset of a space vector and still to preserve its Span.

Theorem 1. *Given a subset* S *and an element* v *of a vector space* V. *A necessary and sufficient condition to* $\text{Span}\{S \cup \{v\}\} = \text{Span}\{S\}$ *is* $v \in \text{Span}\{S\}$.

Proof. Let us prove the sufficient condition first. Assume $v \in \text{Span}\{S\}$, then we need to proof that $\text{Span}\{S \cup \{v\}\} = \text{Span}\{S\}$. Any linear combination of elements of § is also a linear combination of $\{S \cup \{v\}\}$, so we have: $\text{Span}\{S\} \subset \text{Span}\{S \cup \{v\}\}$.

To prove that $\text{Span}\{S \cup \{v\}\} \subset \text{Span}\{S\}$, consider $w \in \text{Span}\{S \cup \{v\}\}$:

$$w = \alpha_0 v + \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_k v_k \text{ with } \{v_1, v_2, \ldots v_k\} \subset S \tag{11.1.3}$$

Since $v \in \text{Span}\{S\}$,

$$v = \beta_1 w_1 + \beta_2 w_2 + \cdots + \beta_m w_m \text{ with } \{w_1, w_2, \ldots w_m\} \subset S \tag{11.1.4}$$

substituting the value of v of Equation (11.1.4) in Equation (11.1.3), it follows that $w \in S$.

Conversely, let us assume

$$\text{Span}\{S \cup \{v\}\} = \text{Span}\{S\} \tag{11.1.5}$$

The vector $v \in \text{Span}\{S \cup \{v\}\}$ by Equation (11.1.5) it follows $v \in \text{Span}\{S\}$. ■

Given a spanned subspace $W = \text{Span}\{S\}$, we can assure that we can remove vectors that are a linear combination of others in S, and W remains the same based on Theorem 1. We can wonder if there exist a $B \subseteq S$, such that none of vector of B is linear combination of others in B, and $\text{Span}\{B\} = \text{Span}\{S\}$. This question opens the way for three very important definitions:

Definition 4. *Given a set* B *of a vector space* V*,*

1. *The set* B *is* **linearly independent** *if none vector of* B *is linear combination of other elements of* B*,*
2. *The set* B *is a* **basis** *of* V *if* B *is linearly independent and* $\text{Span}\{S\} = V$*,*
3. *The dimension of* V *is the cardinality of* B*, where* B *is a basis of* V*.*

The term “a basis” in Definition 4 Item 2 means that the basis of a space is not unique. A space vector could have a basis with infinity elements. It is clear that the vector space $V = (\mathbb{R}^n, +, \mathbb{R}, *)$ with the usual operations is a finite-dimensional space vector.

Note

Here and subsequently, $\mathbb{R}^n$ refers to "the vector space $\mathbb{R}^n$ over the field of $\mathbb{R}$ with the usual addition and the scalar multiplication".

An example of a space vector whose dimension is infinity is the following:

Example 11. *Consider the closed interval* $[0, 1]$ *and the set*

$$C([0,1], \mathbb{R}) = \{f \mid f : [0,1] \to \mathbb{R}, f \text{ is continous}\} \tag{11.1.6}$$

The set $C([0,1],\mathbb{R})$ is a vector space over the field $\mathbb{R}$ under the addition and scalar multiplication defined by

$$(f+g)(x) = f(x)+g(x) \tag{11.1.7}$$

$$(\alpha f)(x) = \alpha f(x), \tag{11.1.8}$$

and it can be proven that the set of functions

$$B = \{\sin(2\pi nx), \cos(2\pi mx), \text{ with } , n \in \mathbb{N} \text{ and } m \in \mathbb{N}\cup\{0\}\}$$

is a basis with infinite number of elements of $C([0,1],\mathbb{R})$ [22, 59, 40].

There are some vector spaces with an additional structure that are very useful in computer sciences programs since they allow us define the concept of "closeness" between elements, and therefore the concept of "convergence".

Definition 5. *A vector space with an operation between its elements* $\langle\cdot,\cdot\rangle : V\times V \to \mathbb{R}$ *satisfying*

1. *for all* $v,w \in V$, $\langle v,w\rangle = \langle w,v\rangle$ *(***symmetry***),*
2. *for all* $u,v,w \in V$ *and* $\alpha \in \mathbb{R}$, $\langle \alpha u+v,w\rangle = \alpha\langle u,w\rangle + \langle v,w\rangle$ *(***bilinear***),*
3. *for all* $u \in V$, $\langle u,u\rangle \geq 0$, *and* $\langle u,u\rangle = 0$ *if and only if* $u = 0$ *(***positive-definite***),*

is called a **vector space with inner product**.

Example 12. *The space* $\mathbb{R}^n$ *with the product defined*

$$\langle v,w\rangle = \sum_{i=1}^{n} v_i w_i,$$

and $L^2(\Omega)$ *Equation* (11.1.1) *with the product defined*

$$\langle f, g\rangle = \int_\Omega f \cdot g$$

are vector spaces with inner product.

The structure of inner product allows us the generalize the definition of orthogonal vectors. We say that two vectors are orthogonal if their inner product is equal to zero. According with this definition, the vector zero is orthogonal to all vector of the space. Moreover, if a vector, e, is orthogonal to all vectors of a space, then $e = 0$.

Definition 6. *Let V a space vector and* B *a basis. The basis* $B = \{v_1, v_2, \ldots, v_n\}$ *is orthonormal if*

$$\langle v_i, v_j\rangle = \delta_{ij} \tag{11.1.9}$$

where δ_{ij} *is the* **Kronecker delta**.

In any vector space with inner product, we can define a norm $\|v\| = \sqrt{\langle v, v\rangle}$. Therefore, given two elements $v, w \in V$

$$d(v, w) = \|v - w\|.$$

We will formalize the concept of metric space in the following section, and with this, we can conclude the first part of this chapter, which is to endow $\mathbb{R}^n$ and $L^2(\Omega)$ with a structure of a metric space.

11.2 Basic Concepts of Metric Spaces

In this section, we will formalize the concept of metric spaces.

Definition 7. *Let V be a set. A metric on V is a function*

$$d : V \times V \to [0, \infty) \tag{11.2.1}$$

such that, for all $u, v, w \in V$

1. $d(u, v) = d(v, u)$ *(***symmetry***),*
2. $d(u, v) = 0$ *if and only if* $u = v$*,*
3. $d(u, w) \leq d(u, v) + d(v, w)$ **(triangle inequality)**.

The spaces $\mathbb{R}^n$ and $L^2(\Omega)$ with the metric inherited by their inner product are examples of metric spaces. Let us present two more examples of metric spaces.

Example 13. *In $V = \{a, b, c\}$, consider the function*

$$\begin{aligned} &d : V \times V \to [0, \infty] \\ &d(v, w) = \begin{cases} 0 & \text{if } v = w \\ \\ 1 & \text{otherwise} \end{cases} . \end{aligned} \tag{11.2.2}$$

The function d in Equation (11.2.2) is a metric known as **discrete metric**. The pair (V, d) is a metric space, which does not satisfy vector space axioms, i.e., not all metric spaces are vector spaces.

Let us denote by $\mathbb{IR}$ the set of all closed intervals in $\mathbb{R}$. The elements in $\mathbb{IR}$ are denoted in bold print, and the real numbers $\underline{x}$ and $\overline{x}$ are the endpoints of the interval, i.e., $\mathbf{x} = [\underline{x}, \overline{x}]$. In $\mathbb{IR}$, let us consider the function

$$\begin{aligned} &d : \mathbb{IR} \times \mathbb{IR} \to [0, \infty) \\ &d(\mathbf{x}, \mathbf{y}) = \max\{|\underline{x} - \underline{y}|, |\overline{x} - \overline{y}|\} \end{aligned} \tag{11.2.3}$$

We prove that the function d in Equation (11.2.3) satisface the definition of metric in the next next theorem:

Theorem 2. *The pair* $(\mathbb{IR}, d)$ *where* d *is defined in Equation* (11.2.3) *is a metric space.*

Proof. We give the proof only for the triangle inequality axiom; the other axioms follow from the properties of absolute value in the definition of d.

Consider $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{IR}$, and

$$\mathrm{d}(\mathbf{x}, \mathbf{z}) = \max\{|\underline{x} - \underline{z}|, |\overline{x} - \overline{z}|\}.$$

There is no loss of generality in assuming

$$\max\{|\underline{x} - \underline{z}|, |\overline{x} - \overline{z}|\} = |\overline{x} - \overline{z}|,$$

applying the triangle inequality of the absolute value and the definition of d

$$\begin{aligned} |\overline{x} - \overline{z}| &\leq |\overline{x} - \overline{y}| + |\overline{y} - \overline{z}| \\ &\leq \max\{|\underline{x} - \underline{y}|, |\overline{x} - \overline{y}|\} + \max\{|\underline{y} - \underline{z}|, |\overline{y} - \overline{z}|\} \\ &= \mathrm{d}(x, y) + \mathrm{d}(y, z). \end{aligned}$$

This finishes the proof. ■

The metric d in Theorem 2 is known as **The Moore metric** [80]. We will study the metric space $(\mathbb{IR}, \mathrm{d})$ in detail in Chapter 2.

Note

In what follows, we will refer to "a metric space" only writing the set V instead of writing the pair (V, d).

Definition 8. *In a space metric* V*, given a* $v_0 \in V$ *and* $r > 0$*, the* **open ball centered in** v_0 **and radius** r *is the set*

$$B(v_0, r) = \{x \in V \mid d(v, v_0) < r\} \tag{11.2.4}$$

and the **closed ball** *with the same center and radius is*

$$\overline{B}(v_0, r) = \{x \in V \mid d(v, v_0) \leq r\}. \tag{11.2.5}$$

Let us introduce the concept of **open set**.

Definition 9. *Let* V *a metric space. A subset* $U \subseteq V$ *is* **open in** V *if for each* $v \in U$*, there exist* $r > 0$ *such that* $B(v, r) \subseteq V$*.*

Given a subset S of a space V. Let us denote by

$$S^c = \{x \in V \mid x \notin S\}. \tag{11.2.6}$$

A subset $W \in V$ is **closed** if W^c is open.

The following is a generalization of the well-known concept of continuous function:

Definition 10. *Let* V *and* W *be metric spaces. A function* $f : V \to W$ *is continuous if for all* $v \in V$ *and* $\varepsilon > 0$*, there exist* $\delta > 0$ *such that if* $v_1 \in B(v, \delta)$*, then* $f(v_1) \in B(f(x), \varepsilon)$*.*

This theorem provides a criterion for determining the continuity of a function based on open subsets.

Theorem 3. *Let* V *and* W *metric spaces. If* $f : V \to W$ *is continuous, then for all open* $W_1 \subset W$*,* $f^{-1}(W_1) \subseteq V$ *is open.*

Proof. Choose $x \in f^{-1}(W_1)$, the element $f(x) \in W_1$, which is open. By definition of open set, there exists $\varepsilon > 0$ such that

$$B(f(x), \varepsilon) \subset W_1 \tag{11.2.7}$$

since f is continuous, there exists $\delta > 0$, such that

$$\forall x' \in B(x, \delta) \rightarrow f(x') \in B(f(x), \varepsilon) \tag{11.2.8}$$

which means that $B(x, \delta) \in f^{-1}(W_1)$. This is our desired conclusion, The best general reference here is [35, 60]. ■

Corollary 1. *Let V and W metric spaces. If $f : V \rightarrow W$ is continuous, then the preimage of any closed set in W is closed.*

Proof.

Consider W_1 a closed set in W. To prove that $f^{-1}(W_1)$ is closed, it is sufficient to show that $(f^{-1}(W_1))^c$ is open. Since

$$(f^{-1}(W_1))^c = f^{-1}(W_1^c),$$

the function f is continuous, and W_1^c is open, it follows that $(f^{-1}(W_1))^c$, which proves the corollary. ■

In a metric space V, the union of an arbitrary family of open sets in V is open, and in consequence, the intersection of any family of closed sets is closed in V.

Example 14. *Consider the function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, where $F(x) = [f_1(x), f_2(x), \ldots, f_n(x)]^T$. Suppose $f_1, f_2, \ldots, f_n$ are continuous functions. The set of*

solutions

$$\begin{aligned} S &= \{x \in \mathbb{R}^n : f_i(x) = 0,\ i = 1, 2, \ldots, n\} \\ &= \bigcap_{i=1}^{n} f_i^{-1}(\{0\}),\ \forall i = 1, 2, \ldots, n \end{aligned} \tag{11.2.9}$$

is closed.

The proof above gives more, if $f_i^{-1}(\{0\})$ is bounded for any $i = 1, 2, \ldots, n$, then S is bounded. There exists an M, such that

$$S \subseteq [-M, M] \times [-M, M] \times \cdots \times [-M, M] \tag{11.2.10}$$

Equation (11.2.10) gives us a reduced domain where to start to search the set of solutions of a system of equations.

Let us define the concept of *dense set*, which will play a key role hereafter:

Definition 11. *In a metric space V. A set S is dense if for each $v \in V$ and for all $\varepsilon > 0$*

$$S \cap B(v, \varepsilon) \neq \emptyset.$$

We can rephrase Definition 11 as:

The set S is dense in V if each $v \in V$ can be **approximated** by an element of $s \in S$.

Example 15. *The set of rational numbers, $\mathbb{Q}$, is dense in $\mathbb{R}$. Moreover, $\mathbb{Q}^n$ is dense in $\mathbb{R}^n$.*

Definition 12. *Given a function $f : [a, b] \to \mathbb{R}$. Suppose the interval $[a, b]$ is partitioned as $a = x_0 < x_1 < \ldots < x_n = b$, and the values $f(x_1), f(x_2,), \ldots, f(x_n)$ are knowns. The function f is continuous-linear piecewise if*

$$f(x) = \frac{x - x_i}{x_{i-1} - x_i} f(x_{i-1}) + \frac{x - x_{i-1}}{x_i - x_{i-1}} f(x_i), \ \forall x \in [x_{i-1}, x_i], i = 1, 2, \ldots, n \qquad (11.2.11)$$

Theorem 4. *The set of continuous-linear piecewise function is dense in* $C([0,1], \mathbb{R})$.

Proof.

Let us recall how the metric in $L^2(\Omega)$ is defined. We begin with the definition of the inner product

$$\langle f, g \rangle = \int_\Omega f \cdot g$$

replacing $g = f$, we can assert that

$$||f||^2 = \int_\Omega f^2$$

the distance follows

$$||f - g||^2 = \int_\Omega (f - g)^2$$

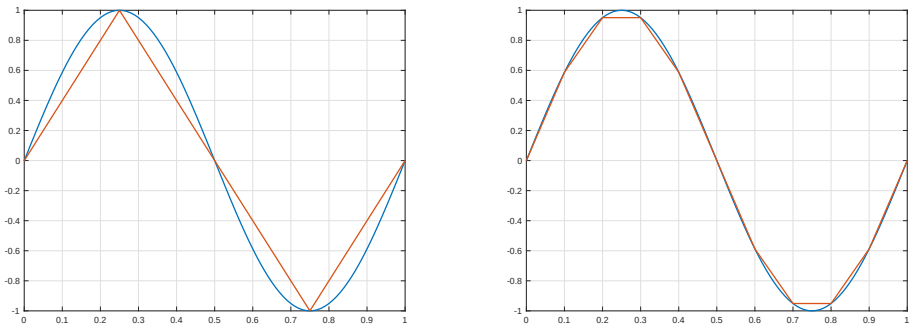
Since $C([0,1], \mathbb{R}) \subset L^2([0,1])$, the space $C([0,1], \mathbb{R})$ is a metric space with the metric inherited by $L^2([0,1])$.

To prove the theorem, we need to show that given any function $f \in C([0,1], \mathbb{R})$ and $\varepsilon > 0$, there exists a continuous-linear piecewise function, g, such that $d(f, g) < \varepsilon$.

Consider $f \in C([0,1], \mathbb{R})$. Set a uniform partition of $[0,1]$

$$0 = x_0 < x_1 < \ldots < x_n = 1,$$

define the continuous-linear piecewise function, g, as Equation (11.2.11) with $h(x_i) = f(x_i)$, see Figure 11.1.



(a) Partition of $[0,1]$ in 4 subintervals (b) Partition of $[0,1]$ in 10 subintervals

Figure 11.1: Continuous-linear piecewise approximation of a continuous function $f : \mathbb{R} \to \mathbb{R}$

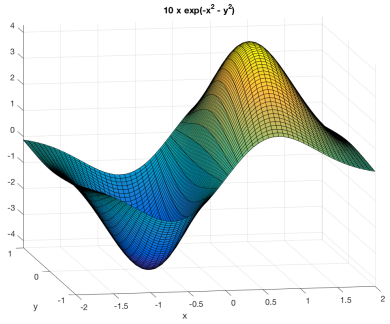
We can take n enough large such that $\int_\Omega (f-g)^2 < \varepsilon$. This is precisely the assertion of the theorem.

■

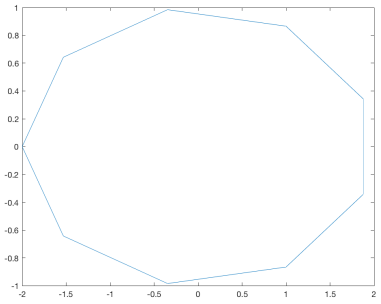
The same proof works for $f \in C(\Omega, \mathbb{R})$, where $\Omega \subset \mathbb{R}^2$ is closed and bounded, see Figure 11.2.

Let us study a generalization of the concept of derivative of functions defined on $\mathbb{R}^n$.

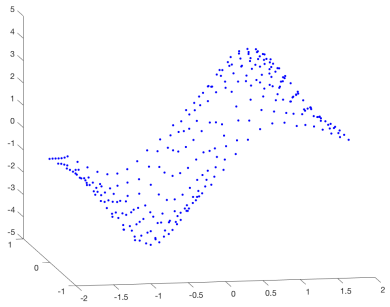
Definition 13. *Suppose that* $F : \mathbb{R}^n \to \mathbb{R}^m$. *The function* $F = (f_1, f_2, \ldots, f_m)$, *where* $f_i : \mathbb{R}^n \to \mathbb{R}$. *If for all* i *and* j, $\partial f_i / \partial x_j$ *is continuous near* a, *then* F *is differentiable in* a *and the* **Jacobian matrix** *[72, 75, 120] is given by*



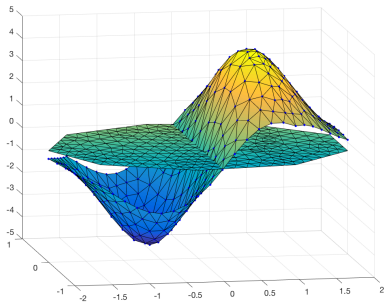
(a) $f = 10x\exp(-x^2 - y^2)$



(b) Domain Ω



(c) Uniform sampling of values of f



(d) Continuous-linear piecewise approximation of f along the partition of the domain Ω

Figure 11.2: Continuous-linear piecewise approximation of a continuous function $f : \mathbb{R}^2 \to \mathbb{R}$

$$J = \begin{pmatrix} \frac{\partial f_1(a)}{\partial x_1} & \frac{\partial f_1(a)}{\partial x_2} & \cdots & \frac{\partial f_1(a)}{\partial x_n} \\ \frac{\partial f_2(a)}{\partial x_1} & \frac{\partial f_2(a)}{\partial x_2} & \cdots & \frac{\partial f_2(a)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(a)}{\partial x_1} & \frac{\partial f_m(a)}{\partial x_1} & \cdots & \frac{\partial f_m(a)}{\partial x_n} \end{pmatrix} \tag{11.2.12}$$

The following theorem will provide a generalization of the well-known **Chain Rule**.

Theorem 5. *Let* $F : \mathbb{R}^k \to \mathbb{R}^n$ *and* $G : \mathbb{R}^n \to \mathbb{R}^m$*. If* F *is differentiable in* a*, and* G *is differentiable in* $F(a)$*. Then the function* $G \circ F$ *is differentiable in* a *[4, 32, 88], and its Jacobian matrix is given by*

$$J_{G\circ F}(a) = \begin{pmatrix} \frac{\partial g_1(f(a))}{\partial x_1} & \frac{\partial g_1(f(a))}{\partial x_2} & \cdots & \frac{\partial g_1(f(a))}{\partial x_n} \\ \frac{\partial g_2(f(a))}{\partial x_1} & \frac{\partial g_2(f(a))}{\partial x_2} & \cdots & \frac{\partial g_2(f(a))}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m(f(a))}{\partial x_1} & \frac{\partial g_m(f(a))}{\partial x_1} & \cdots & \frac{\partial g_m(f(a))}{\partial x_n} \end{pmatrix} \begin{pmatrix} \frac{\partial f_1(a)}{\partial x_1} & \frac{\partial f_1(a)}{\partial x_2} & \cdots & \frac{\partial f_n(a)}{\partial x_k} \\ \frac{\partial f_2(a)}{\partial x_1} & \frac{\partial f_2(a)}{\partial x_2} & \cdots & \frac{\partial f_2(a)}{\partial x_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(a)}{\partial x_1} & \frac{\partial f_n(a)}{\partial x_1} & \cdots & \frac{\partial f_n(a)}{\partial x_k} \end{pmatrix}, \tag{11.2.13}$$

The demonstration can be found [6].

In particular, if F is a linear function, then it can be expressed as $F(x) = \Phi x$, where Φ is a $n \times k$ matrix. The jacobian of F is the constant matrix $J_F(a) = \Phi$, and In this case, the chain rule states

$$J_{G\circ F}(a) = J_G(\Phi a)\Phi \tag{11.2.14}$$

In this chapter, we have presented the notion of linear space. We defined inner product space vectors, which allow us to define orthonormal basis. In the second part of the

chapter, we study the basic properties of metric spaces. We proved that $C([0,1],\mathbb{R})$ and $L^2(\Omega)$ are metric spaces, moreover we proved that each element in $C([0,1]$ can be approximated by continuous-linear piecewise functions.

Chapter 12

Reliable Computation of the Spectrum of a Matrix

The future depends on what you do today.
Mahatma Gandhi

This chapter could be considered as part of the future work chapter, but due to it is not directly related with the thesis we write it in an appendix.

12.1 Spectrum of a Matrix and Linear System of Differential Equations (Change the title)

A linear system of differential equations can be solved using analytically using the exponential of a matrix [22], i.e.,

$$\dot{X} = AX;\ X(0) = X_0 \tag{12.1.1}$$

The solution of Equation (12.1.1) is given as

$$\begin{aligned} X(t) &= X_0 e^{At}, \text{ where} \\ e^{At} &= I + At + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots + \frac{(At)^n}{n!} + \cdots \end{aligned} \tag{12.1.2}$$

At this point is where the concept of eigenvalues and eigenvectors play a important to solve a a linear system of differential equations.

Definition 14. *A matrix* $A_{n\times n}$ *said to be diagonalizable if there exists a diagonal matriz* D *and an invertible matrix* V *sucth that*

$$A = VDV^{-1}. \tag{12.1.3}$$

The columns of V are the eigenvectors of A and $D = diag([\lambda_1, \lambda_2, \ldots, \lambda_n])$ is the diagonal matrix whose elements are the eigenvalues. Substituting Equation (12.1.3) in Equation (12.1.2)

$$e^{At} = Ve^{Dt}V^{-1} \tag{12.1.4}$$

where e^{Dt} is a diagonal matrix whose elements $(e^{Dt})_{ii} = e^{\lambda_i t}$.

12.1.1 Reliable Computations of the Spectrum of a Non-defective Matrix (Change the title)

The classical way to find the eigenvalues and eigenvectors of A is to find the roots of the characteristic polynomial

$$p(\lambda) = det(A - \lambda I) = 0, \tag{12.1.5}$$

and then we have to find a basis for the null space of the matrix $A - \lambda I$

$$(A - \lambda I)v = 0. \tag{12.1.6}$$

Given a matrix A, if v is a eigenvector corresponding to the eigenvalue λ, then for each $\beta \neq 0$

$$\begin{aligned} A(\beta v) &= \beta A v \\ &= \beta(\lambda v) \\ &= \lambda(\beta v), \end{aligned} \tag{12.1.7}$$

so βv is also a eigenvector of λ. A direct consequence of Equation (12.1.7) is that Equation (12.1.6) infinite solutions.

The following theorem allows us to determine where the eigenvalues are located.

Theorem 6. *Let A a matrix and $R_i = \sum_{j \neq i} |a_{ij}|$. Consider the $B(a_{ii}, R_i)$ (Gershgorin circle). Each eigenvalue of A lies within at least one of the Gershgorin circle.*

Proof. Let λ be an eigenvalue of A, and $x = (x_1, x_2, \ldots, x_i, \ldots, x_n)$ an eigenvector associate to λ such that $x_i = 1$ and $|x_j| \leq 1;\ \forall j \neq i$.

$$\sum_{j=1} a_{ij} x_j = \sum_{j=1,\, j \neq i} a_{ij} x_j + a_{ii} x_i = \lambda x_i = \lambda \tag{12.1.8}$$

$$|\lambda - a_{ii}| = \left| \sum_{j=1,\, j \neq i} a_{ij} x_j \right| \leq \sum_{j=1,\, j \neq i} |a_{ij}| = R_i. \tag{12.1.9}$$

Therefore, $\lambda \in B(a_{ii}, R_i)$. ■

We can formulate Equation (12.1.6) using interval constraint solving techniques:

$$a_{i1} x_1 + a_{i2} x_2 + \cdots + a_{in} x_n = \lambda x_i; i = \{1, 2, \ldots, n\} \tag{12.1.10}$$

$$x_1^2 + x_2^2 + \cdots + x_n^2 = 1 \tag{12.1.11}$$

$$x_1 \geq 0 \tag{12.1.12}$$

Equations (12.1.10) are the constraints define by (12.1.6), and Equations (12.1.11)-(12.1.12) are used to guarantee that the eigenvectors form a basis.

Example 16. *Find the eigenvalues and eigenvectors of the following matrix*

$$A = \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix}$$

Constraints:

$$\begin{aligned} x_2 &= \lambda x_1 \\ -2x_1 - 3x_2 &= \lambda x_2 \\ x_1^2 + x_2^2 &= 1 \\ x_1 &\geq 0 \end{aligned}$$

Initial Box:

$$\begin{aligned} x_1 &\in [-1, 1] \\ x_2 &\in [-1, 1] \\ \lambda &\in [-5, 1] \end{aligned}$$

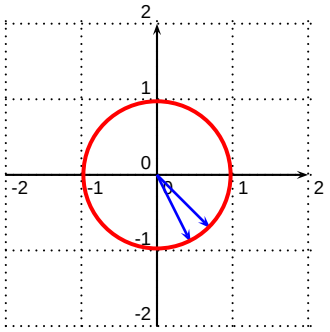


Figure 12.1: Eigenvectors of norm 1

In Figure 12.1, we can see the eigenvectors of the matrix A.

We have shown in this appendix how Interval Constraint Solving Techniques can be used to find in a reliable way the spectrum of a matrix.

References

[1] S. Abdallah. Numerical Solutions for the Pressure Poisson Equation with Neumann Boundary Conditions Using a Non-Staggered Grid, I. *Journal of Computational Physics,* Elseiver, Amsterdam, Netherlands, 70(1):182–192, 1987.

[2] R. A. Adams and J. J. Fournier. *Sobolev Spaces*, volume 140. Academic Press, Kidlington, Oxford, UK, 2003.

[3] N. M. Amato, M. T. Goodrich, and E. A. Ramos. A Randomized Algorithm for Triangulating a Simple Polygon in Linear Time. *Discrete & Computational Geometry: An International Journal of Mathematics and Computer Science,* New York, NY, 26(2):245–265, 2001.

[4] L. Ambrosio and G. Dal Maso. A General Chain Rule for Distributional Derivatives. *Proceedings of the American Mathematical Society,* Providence, RI, USA, 108(3):691–702, 1990.

[5] T. M. Apostol. *Mathematical Analysis.* Addison Wesley Publishing Company, Boston, USA, 1974.

[6] T. M. Apostol. *Calculus, volume I*, volume 1. John Wiley & Sons, Hoboken, NJ, USA, 2007.

[7] Y. Aregbesola. Numerical Solution of Bratu Problem Using the Method of Weighted Residual. *Electron Journal South African Mathematical Science,* Palapye, Botswana, 2003.

[8] M. Argáez, H. Florez, and O. Méndez. A Model Reduction for Highly Non-linear Problems Using Wavelets and the Gauss-Newton Method. In *2016 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS),* El Paso, TX, USA, pages 1–7. IEEE, 2016.

[9] M. Artin. *Algebra.* Always learning. Pearson Education Limited, New York, NY, 2013.

[10] Z. Bai, K. Meerbergen, and Y. Su. Arnoldi Methods for Structure-Preserving Dimension Reduction of Second-Order Dynamical Systems. In *Dimension Reduction of Large-Scale Systems*, pages 173–189. Springer, Berlin, Heidelberg, 2005.

[11] B. R. Baliga and S. V. Patankar. A New Finite-Element Formulation for Convection-Diffusion Problems. *Numerical Heat Transfer,* New York, NY, 3(4):393–409, 1980.

[12] A. Bejan. *Convection Heat Transfer.* John Wiley & Sons, New York, NY, 2013.

[13] R. Bellman, B. G. Kashef, and J. Casti. Differential Quadrature: A Technique for the Rapid Solution of Nonlinear Partial Differential Equations. *Journal of Computational Physics,* Los Angeles, CA, 10(1):40–52, 1972.

[14] F. Benhamou, F. Goualard, V. Granvilliers, and J. Puget. Revising Hull and Box Consistency. In *Logic Programming: Proceedings of the 1999 International Conference on Logic Programming*, page 230. MIT press, Cambridge, MA, USA, 1999.

[15] P. Benner and E. S. Quintana-Ortíz. Model Reduction Based on Spectral Projection Methods. *Dimension Reduction of Large-Scale Systems,* Springer, Berlin, Heidelberg, 45:5–45, 2005.

[16] J. L. Berggren. A lacuna in Book I of Archimedes' Sphere and Cylinder. *Historia Mathematica,* Elseiver, Amsterdam, Netherlands, 4(1):1–6, 1977.

[17] D. M. Bergstrom, A. Lucieer, K. Kiefer, J. Wasley, L. Belbin, T. K. Pedersen, and S. L. Chown. Indirect Effects of Invasive Species Removal Devastate World Heritage Island. *Journal of Applied Ecology,* Oxford, England, 46(1):73–81, 2009.

[18] G. Berkooz, P. Holmes, and J. L. Lumley. The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. *Annual Review of Fluid Mechanics,* Palo Alto, CA, 25(1):539–575, 1993.

[19] D. S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*, volume 41. Princeton University Press, Princeton, 2005.

[20] L. Bers, S. Bochner, and F. John. *Contributions to the Theory of Partial Differential Equations.* Number 33. Princeton University Press, Princeton, 1954.

[21] A. Björck. *Numerical Methods for Least Squares Problems.* Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, January 1996.

[22] W. E. Boyce, R. C. DiPrima, and C. W. Haines. *Elementary Differential Equations and Boundary Value Problems*, volume 9. Wiley, New York, 1969.

[23] A. L. Brown and A. Page. Elements of Functional Analysis. *New University Mathematics Series,* Van Nostrand Reinhold, New York, NY, 1970.

[24] R. L. Burden and J. D. Faires. 2.1 the Bisection Algorithm. Numerical Analysis. *Prindle, Weber & Schmidt,* Boston, MA, 1985.

[25] A. J. Burton and G. F. Miller. The Application of Integral Equation Methods to the Numerical Solution of Some Exterior Boundary-Value Problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences,* London, 323(1553):201–210, 1971.

[26] L. Cai and R. E. White. Reduction of Model Order Based on Proper Orthogonal Decomposition for Lithium-ion Battery Simulations. *Journal of the Electrochemical Society,* Philadelphia, PA, 156(3):A154–A161, 2009.

[27] K. M. Case. Elementary Solutions of the Transport Equation and their Applications. *Annals of Physics,* New York, NY, 9(1):1–23, 1960.

[28] M. Ceberio and L. Granvilliers. Horner's Rule for Interval Evaluation Revisited. *Computing,* Austria, 69(1):51–81, 2002.

[29] M. Ceberio and L. Granvilliers. Solving Nonlinear Equations by Abstraction, Gaussian Elimination, and Interval Methods. In *International Workshop on Frontiers of Combining Systems*, pages 117–131. Springer, Berlin, 2002.

[30] B. Chazelle. Triangulating a Simple Polygon in Linear Time. *Discrete & Computational Geometry,* Princeton, NJ, 6(3):485–524, 1990.

[31] G. Comini, S. Del Guidice, R. W. Lewis, and O. C. Zienkiewicz. Finite Element Solution of Non-Linear Heat Conduction Problems with Special Reference to Phase Change. *International Journal for Numerical Methods in Engineering,* Wiley Online Library, New York, NY, 8(3):613–624, 1974.

[32] K. R. Davidson and A. P. Donsig. *Real Analysis and Applications: Theory in Practice.* Springer Science & Business Media, Dordrecht, Heidelberg, London, 2009.

[33] L. C. De Barros, R. Bassanezi, and W. Lodwick. *First Course in Fuzzy Logic, Fuzzy Dynamical Systems, and Biomathematics*. Springer, Berlin, Heidelberg, 2016.

[34] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. Polygon Triangulation: Guarding an Art Gallery. *Computational Geometry: Algorithms and Applications,* Berlin, pages 45–61, 2008.

[35] J. Dugundji. *Topology*. Allyn and Bacon Series in Advanced Mathematics. Allyn and Bacon, Inc., Boston, 1978.

[36] H. W. Eves. *An Introduction to the History of Mathematics : With Cultural Connections*. Holt, Rinehart and Winston, Forth Worth, TX, 1992.

[37] H. Flórez and M. Argáez. Applications and Comparison of Model-Order Reduction Methods Based on Wavelets and POD. In *2016 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS),* El Paso, TX, USA, pages 1–8. IEEE, 2016.

[38] A. Fournier and D. Y. Montuno. Triangulating Simple Polygons and Equivalent Problems. *ACM Transactions on Graphics (TOG),* New York, NY, 3(2):153–174, 1984.

[39] I. M. Gelfand. Some problems in the theory of quasi-linear equations. *Uspekhi Matematicheskikh Nauk,* Vaduz, Liechtenstein, 14(2):87–158, 1959.

[40] E. A. Gonzalez-Velasco. Connections in Mathematical Analysis: The Case of Fourier Series. *The American Mathematical Monthly,* JSTOR, New York, NY, 99(5):427–441, 1992.

[41] T. R. Goodman. Application of Integral Methods to Transient Nonlinear Heat Transfer. In *Advances in Heat Transfer*, volume 1, pages 51–122. Elsevier, Amsterdam, Netherlands, 1964.

[42] L Granvilliers. Realpaver User's Manual: Solving Nonlinear Constraints by Interval Computations. *University of Nantes,* Nantes, FR, 2003.

[43] L. Granvilliers and F. Benhamou. Realpaver: an Interval Solver Using Constraint Satisfaction Techniques. *ACM Transactions on Mathematical Software (TOMS),* New York, NY, 32(1):138–156, 2006.

[44] W. Hackbusch. The Poisson Equation. *Elliptic Differential Equations,* Springer, Berlin, Heidelberg, pages 27–37, 2010.

[45] E. Hansen and G. W. Walster. *Global Optimization Using Interval Analysis: Revised and Expanded*, volume 264. CRC Press, New York, NY, 2003.

[46] E. R. Hansen and R.I. Greenberg. An Interval Newton Method. *Applied Mathematics and Computation,* New York, NY, 12(2-3):89–98, 1983.

[47] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers.* Oxford University Press, New York, NY, 1979.

[48] A. Hastings. Global Stability in Lotka-Volterra Systems with Diffusion. *Journal of Mathematical Biology,* New York, NY, 6(2):163–168, 1978.

[49] A. Hastings and T. Powell. Chaos in a Three-Species Food Chain. *Ecology,* New York, NY, 72(3):896–903, 1991.

[50] J. Hefferon. *Linear Algebra. 29.2. 2012.* Mathematics Department, St. Michael's College, Colchester, Vt., 2012.

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

[52] M. W. Hirsch, S. Smale, and R. L. Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Academic press, Cambridge, MA, USA, 2012.

[53] K. Hoffman and R. Kunze. *Linear Algebra*. Prentice Hall of India, New Delhi, 1990.

[54] W. G. Horner. A New Method of Solving Numerical Equations of All Orders, by Continuous Approximation. *Philosophical Transactions of the Royal Society of London,* New York, NY, 109:308–335, 1819.

[55] S. Hsu, T. Hwang, and Y. Kuang. A Ratio-Dependent Food Chain Model and Its Applications to Biological Control. *Mathematical Biosciences,* Elseiver, Amsterdam, Netherlands, 181(1):55–83, 2003.

[56] L. Jaulin. *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics*, volume 1. Springer Science & Business Media, London, 2001.

[57] C. K. R. T Jones. Stability of the travelling wave solution of the fitzhugh-nagumo system. *Transactions of the American Mathematical Society,* Providence, RI, 286(2):431–469, 1984.

[58] D. W. Jordan and P. Smith. *Nonlinear Ordinary Differential Equations: An Introduction to Dynamical Systems*, volume 2. Oxford University Press, New York, NY, 1999.

[59] Y. Katznelson. *An Introduction to Harmonic Analysis*. Cambridge University Press, Cambridge, MA, USA, 2004.

[60] J. L. Kelley. *General Topology*. Courier Dover Publications, New York, NY, 2017.

[61] G. Kerschen and J. Golinval. Physical Interpretation of the Proper Orthogonal Modes Using the Singular Value Decomposition. *Journal of Sound and Vibration,* Elsevier, Amsterdam, Netherlands, 249(5):849–865, 2002.

[62] W. R. Knorr. Archimedes and the Measurement of the Circle: a New Interpretation. *Archive for history of exact sciences,* Berlin, 15(2):115–140, 1976.

[63] S. Kutluay, A. R. Bahadir, and A. Özdeş. Numerical Ssolution of One-Dimensional Burgers Equation: Explicit and Exact-Explicit Finite Difference Methods. *Journal of Computational and Applied Mathematics,* New York, NY, 103(2):251–261, 1999.

[64] S. Lang. *Algebraic Number Theory*, volume 110. Springer Science & Business Media, New York, NY, 2013.

[65] C. E. Lee. Evolutionary Genetics of Invasive Species. *Trends in ecology & evolution,* Elseiver, Amsterdam, Netherlands, 17(8):386–391, 2002.

[66] A. C. Lees and D. J. Bell. A conservation Paradox for the 21st Century: The European Wild Rabbit Oryctolagus Cuniculus, an Invasive Alien and an Endangered Native Species. *Mammal Review,* Blackwell Publishin, Oxford, 38(4):304–320, 2008.

[67] S. J. Leon. *Linear Algebra with Applications*. Macmillan, New York, NY, 1980.

[68] R. J. LeVeque. Conservative Methods for Nonlinear Problems. In *Numerical Methods for Conservation Laws,* Cham, Switzerland, pages 122–135. Springer, 1990.

[69] Y. C. Liang, H. P. Lee, S.P. Lim, W. Z. Lin, K. H. Lee, and C. G. Wu. Proper Orthogonal Decomposition and its Applications, Part I: Theory. *Journal of Sound and Vibration,* Elsevier, Amsterdam, Netherlands, 252(3):527–544, 2002.

[70] B. Lindner and L. Schimansky-Geier. Analytical Approach to the Stochastic Fitzhugh-Nagumo System and Coherence Resonance. *Physical Review E,* Melville, NY, 60(6):7270, 1999.

[71] T. Liu. Nonlinear Stability of Shock Waves for Viscous Conservation Laws. *Bulletin of the American Mathematical Society,* New York, NY, 12(2):233–236, 1985.

[72] X Magnus and H. Neudecker. Matrix Differential Calculus with Applications in Statistics and Econometrics. *Wiley Series in Probability and Mathematical Statistics,* Wiley, Chichester, England, 1988.

[73] X. Mao, S. Sabanis, and E. Renshaw. Asymptotic Behaviour of the Stochastic LotkaVolterra model. *Journal of Mathematical Analysis and Applications,* Academic Press, Orlando, 287(1):141–156, 2003.

[74] P. A. Markowich and P. Szmolyan. A System of Convection-Diffusion Equations with Small Diffusion Coefficient Arising in Semiconductor Physics. *Journal of Differential Equations,* Elsevier, Amsterdam, Netherlands, 81(2):234–254, 1989.

[75] A. M. Mathai. *Jacobians of Matrix Transformations and Functions of Matrix Arguments.* World Scientific Publishing Company, Singapore, 1997.

[76] K. Mccann and P. Yodzis. Bifurcation Structure of a Three-Species Food-Chain Model. *Theoretical population biology,* Elseiver, Amsterdam, Netherlands, 48(2):93–125, 1995.

[77] P. Melby, N. Weber, and A. Hübler. Dynamics of Self-Adjusting Systems with Noise. *Chaos: An Interdisciplinary Journal of Nonlinear Science,* Woodbury, NY , 15(3):033902, 2005.

[78] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*, volume 71. SIAM, Philadelphia, PA, 2000.

[79] B. Moore. Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction. *IEEE Transactions on Automatic Control,* New York, NY, 26(1):17 – 32, 1981.

[80] R. E. Moore. *Interval Analysis*, volume 4. Prentice-Hall, Englewood Cliffs, Philadelfia, PA, 1966.

[81] E. Nabil. A Modified Flower Pollination Algorithm for Global Optimization. *Expert Systems with Applications,* Elseiver, Amsterdam, Netherlands, 57:192–203, 2016.

[82] M. C. Nucci and P. A. Clarkson. The Nonclassical Method is More General than the Direct Method for Symmetry Reductions. An Example of the Fitzhugh-Nagumo Equation. *Physics Letters A,* New York, NY, 164(1):49–56, 1992.

[83] S. Odejide and Y. Aregbesola. A Note on Two Dimensional Bratu Problem. *Kragujevac Journal of Mathematics,* Kragujevac, Serbia, 29(29):49–56, 2006.

[84] C. C. Paige and M. A. Saunders. Solution of Sparse Indefinite Systems of Linear Equations. *SIAM Journal on Numerical Analysis,* Philadelphia, PA, 12(4):617–629, 1975.

[85] A. Panfilov and P. Hogeweg. Spiral Breakup in a Modified Fitzhugh-Nagumo Model. *Physics Letters A,* New York, NY, 176(5):295–299, 1993.

[86] A. Pekalski and D. Stauffer. Three Species Lotka-Volterra Model. *International Journal of Modern Physics C,* World Scientific Pub, Singapore, 9(05):777–783, 1998.

[87] Lawrence Perko. *Differential Equations and Dynamical Systems*, volume 7. Springer Science & Business Media, New York, NY, 2013.

[88] N. Piskunov, K.P. Medkov, et al. *Cálculo Diferencial e Integral*, volume 1. Mir, Moscou, 1983.

[89] T. Reichenbach, M. Mobilia, and E. Frey. Coexistence Versus Extinction in the Stochastic Cyclic Lotka-Volterra Model. *Physical Review E,* American Physical Society, Melville, NY, 74(5):051907, 2006.

[90] M. Rewienski and J. White. A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* New York, NY, 22(2):155–170, 2003.

[91] F. J. Rizzo and D. J. Shippy. A Method of Solution for Certain Problems of Transient Heat Conduction. *American Institute of Aeronautics and Astronautics Journal,* AIAA Journal, New York, NY, 8(11):2004–2009, 1970.

[92] S. M. Robinson. False Numerical Convergence in Some Generalized Newton Methods. In *Equilibrium Problems and Variational Models*, pages 401–416. Springer, Boston, MA, 2003.

[93] S. Roman, S. Axler, and F. W. Gehring. *Advanced Linear Algebra*, volume 3. Springer, New York, NY, 2005.

[94] W. Rudin et al. *Principles of Mathematical Analysis*, volume 3. McGraw-hill, New York, NY, 1964.

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

[96] D. Sam-Haroud and B. Faltings. Consistency Techniques for Continuous Constraints. *Constraints,* Zürich, Switzerland, 1(1):85–118, 1996.

[97] F. Scheck. Probabilities, States, Statistics. In *Statistical Theory of Heat*, pages 105–140. Springer, Switzerland, 2016.

[98] W. H. A. Schilders, H. A. Van der Vorst, and J. Rommes. *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13. Springer, Berlin, 2008.

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

[100] V. Stahl. *Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations*. Johannes-Kepler-Universität, Linz, 1995.

[101] I. Stakgold and M. J. Holst. *Green's Functions and Boundary Value Problems*, volume 99. John Wiley & Sons, New York, NY, 2011.

[102] J. Starck, F. Murtagh, and J. M. Fadili. *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*. Cambridge University Press, Cambridge, MA, USA, New York, NY, May 2010.

[103] G. Strang. *Introduction to Linear Algebra*, volume 3. Wellesley-Cambridge Press, Wellesley, MA, 1993.

[104] S. H. Strogatz. *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering.* CRC Press, Boulder, CO, 2018.

[105] M. I Syam and A. Hamdan. An Efficient Method for Solving Bratu Equations. *Applied Mathematics and Computation,* New York, NY, 176(2):704–713, 2006.

[106] A. E. Taylor and D. C. Lay. *Introduction to Functional Analysis*, volume 2. Wiley, New York, NY, 1958.

[107] M. E. Taylor. *Measure Theory and Integration.* American Mathematical Society, New York, NY, 2006.

[108] J. M. T. Thompson and H B. Stewart. *Nonlinear Dynamics and Chaos.* John Wiley & Sons, New York, NY, 2002.

[109] L. Valera and M. Ceberio. Model-Order Reduction Using Interval Constraint Solving Techniques. In *Proceedings of the 7th International Workshop on Reliable Engineering Computing REC'2016,* Bochum, Germany, 2016.

[110] L. Valera, A. Garcia, and M. Ceberio. "On-the-fly" Parameter Identification for Dynamic Systems Control, Using Interval Computations and Reduced-Order Modeling. In *North American Fuzzy Information Processing Society Annual Conference,* Cancun, Mexico, pages 293–299. Springer, 2017.

[111] L. Valera, A. Garcia, M. Afshin G. Ceberio, and H. Florez. Towards Predictions of Large Dynamic Systems' Behavior Using Reduced-Order Modeling and Interval Computations. In *The 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC2017),* Banff, Canada, 2017.

[112] F. Verhulst. *Nonlinear Differential Equations and Dynamical Systems*. Springer Science & Business Media, New York, NY, 2006.

[113] X. Wang, A. F. Garcia, M. Ceberio, C. Del Hoyo, and L. C. Gutierrez. A Speculative Algorithm to Extract Fuzzy Measures from Sample Data. In *2012 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, Brisbane, Australia, 2012.

[114] Abdul-Majid Wazwaz. A Reliable Study for Extensions of the Bratu Problem with Boundary Conditions. *Mathematical Methods in the Applied Sciences,* New York, NY, 35(7):845–856, 2012.

[115] W. Weibull et al. A Statistical Distribution Function of Wide Applicability. *Journal of Applied Mechanics,* American Society of Mechanical Engineers, New York, NY, 18(3):293–297, 1951.

[116] W. Wendi and L. Zhengyi. Global Stability of Discrete Models of Lotka-Volterra Type. *Nonlinear Analysis: Theory, Methods & Applications,* Pergamon Press, New York, NY, 35(8):1019–1030, 1999.

[117] K. Willcox and J. Peraire. Balanced Model Reduction Via the Proper Orthogonal Decomposition. *AIAA journal,* Berlin, 40(11):2323–2330, 2002.

[118] K. Yamamura, H. Kawata, and A. Tokue. Interval Solution of Nonlinear Equations Using Linear Programming. *BIT Numerical Mathematics,* Dordrecht, Netherlands , 38(1):186–199, 1998.

[119] K. Yee. Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media. *IEEE Transactions on antennas and propagation,* Piscataway, NJ, 14(3):302–307, 1966.

[120] X. Zhang. *Matrix Analysis and Applications*. Cambridge University Press, Cambridge, MA, USA, 2017.

[121] C. Zheng, G. D Bennett, et al. *Applied Contaminant Transport Modeling*, volume 2. Wiley-Interscience, New York, NY, 2002.

[122] A. Zien, R. Zimmer, and T. Lengauer. A Simple Iterative Approach to Parameter Optimization. *Journal of Computational Biology,* Mary Ann Liebert, Inc., New York, NY, 7(3-4):483–501, 2000.

Curriculum Vitae

Leobardo Valera was born in Zaraza, Venezuela. After obtaining his Bachelor's degree, he managed to accomplish a Master's degree in Multi-Linear Algebra. He got hired as university lecturer by one of the most prestigious internationally-recognized universities in Venezuela; La Universidad Metropolitana (Metropolitan University). In his academic life. He continued his career until spring 2013 when he moved to El Paso, Texas, to pursue a Ph.D. degree in computational science at the University of Texas at El Paso (UTEP). Once at UTEP, I started to work as a Research Assistant in CR2G under the supervision of Dr. Martine Ceberio.

He has had the opportunity to present his work in several conferences and two of his articles have received the Outstanding-Paper award at the Joint Annual Conference of the North American Fuzzy Information processing Society NAFIPS'2015 and 5th World Conference on Soft Computing.

In November 2015, he defended his dissertation proposal in the area of Numerical Optimization and he completed his Master's degree in computational science. His work was recognized for **Academic and Research Excellence by the Computational Science Program** at UTEP in December 2015. He obtained his doctoral degree in Spring 2018.

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