ESCUELA DE INGENIERÍA DE FUENLABRADA

DEGREE IN BIOMEDICAL ENGINEERING

BACHELOR'S THESIS

B-SPLINES-BASED SOLUTIONS OF DIFFERENTIAL EQUATIONS IN BIOENGINEERING: MOTION MONITORING AND IMAGE PROCESSING APPLICATIONS

by

Antonio Manuel Ramírez Márquez

Tutor: David Casillas Pérez
Co-tutor: Sara García de Villa

Academic Year 2023/2024

A good decision is based on knowledge and not on numbers. - Plato

Acknowledgements

At first place, to my tutor David Casillas who had guided me in the process of developing this work. Thanks for introducing me to the world of research and concretely to the world of applied mathematics. Also, thanks for the dedication and for instructing me so well.

At second place, to my co-tutor Sara García who had introduced me in the world of motion monitoring in the field of rehabilitation. Thank you for your teaching and for your dedication.

At third place, to my family, specially to my parents. Thanks for giving me the opportunity of keep developing my career and for all the support you have gave me along the process.

At fourth place, I would like to mention professors Pizarro and Fuentes from University of Alcala for introducing me to the problematic of Liver Laparoscopic procedure.

Finally, to all the people that had been with me during the developing of this work or during my bachelor's degree and I had not mentioned. Without all of you, I would not be able of doing this.

Abstract

The bachelor's thesis objective consists of developing a quintic spline numerical algorithm and study its potential for solving some different bioengineering problems. The numerical algorithm will use collocation methods for finding the approximation of the mathematical analysis problem. This approximation will be estimated with a quintic spline.

In general terms, mathematical analysis problems are found in different areas, where it could be highlighted physical sciences and engineering, moreover it also can be found in social sciences, business, medicine, biology, etc. This method can be used for any of these areas if they fulfil the specific requirements given by the design constraints and by the numerical dynamics.

Although the algorithm will be prepared for solving a great amount of numerical problems, along this work, it will be more focused on the applications related to bioengineering and medicine. The algorithm is designed to solve problems either formulated as an initial value problem or formulated as boundary value problems. It can be adapted to solve differential equations and non-linear differential equations. It is possible to scale the algorithm for solving multivariate functions and multidimensional function.

The work will have two parts. The first part is oriented to evaluate the algorithm in the different conditions where it is projected to work, this part also will illustrate the way that collocation methods work. It is useful as it will aid to design the hyper-parameters of the rest of the parts. At this part it will be carried some simple synthetic experiments with known solution but with poor applicability in areas of interest.

In this work, a final applied synthetic experiments will be carried out. This experiment will approximate the Gompertz model to estimate the evolution of the population of tumour cells along the time from an initial population of time and with some experimented parameters. As this equation is analytically solved, this experiment is also useful to validate the algorithm as the previous one.

During this part of the work, synthetic experiments had been carried out. In the work, we define a synthetic experiment as an approximation of a function that can be fully

expressed as a differential equation. Then, it is defined a practical equation as an experiment where an acquired value vary along the time. In this two experiments, acquisition and experimental parameters are needed but do change along the independent variable.

Finally, at the second part of this work, two practical experiments are carried out. One application, in the field of motion monitoring applied to the rehabilitation field and, the second application in the field of image processing applied to laparoscopic liver resection.

In rehabilitation, the estimation of the position and it could be done using the angular velocity and the acceleration given by an IMU. It can also be given by an optical position system that in the case of this experiment allow to estimate the performance of an algorithm.

In different surgical procedure, pre-operative models are essential to incorporate information from other imaging process to the procedure. The objective of the last application is to represent a tumour preoperative model as a derivable spline.

Resumen en Castellano

El objetivo del trabajo consiste en desarrollar algoritmos numéricos basados principalmente en splines quínticos y estudiar su potencial para resolver diferentes problemas en bioingeniería. Se usarán métodos de colocación para aproximar los problemas de análisis matemáticos. Las aproximaciones se desarrollaran con spline quínticos.

En términos generales, estos problemas de análisis matemático se encuentran diferentes áreas, donde destacan las ciencias físicas y las ingenierías, aunque además se puede encontrar en las ciencias sociales, negocios, medicina, biología, etc. El método puede ser usado en cualquiera de estas áreas si cumplen los requisitos específicos dados por las restricciones de diseño y por la dinámica numérica.

Aunque el algoritmo estará preparado para resolver una gran cantidad de problemas numéricos problemas, a lo largo de este trabajo, se centrará más en las aplicaciones relacionadas con la bioingeniería y la medicina. El algoritmo está diseñado para resolver problemas formulados como un problema de valor inicial o formulado como problemas de contorno. Se puede adaptar para resolver ecuaciones diferenciales lineales y ecuaciones diferenciales no lineales. Es posible escalar el algoritmo para resolver funciones multivariadas y funciones multidimensionales.

El trabajo tendrá dos partes. La primera parte está orientada a evaluar el algoritmo en las diferentes condiciones y como los métodos de colocación trabajan. Será útil para mostrar como hiper-parámetros afectan a su dinámica. En esta parte se llevarán a cabo experimetnos sintéticos con soluciones conocidad pero con poca aplicabilidad en áreas de interés.

En esta parte del trabajo, un experimento sintético aplicado se llevara a cabo. En este experimento se aproximará el modelo de Gompertz para estimar la evolución de la población de células tumorales a lo largo del tiempo, partiendo de una población inicial de células y otros parámetros experimentales. Como el modelo tiene solución analítica, este experimento será útil para validar el método como el resto de experimentos sintéticos.

Finalmente, en la segunda parte de este trabajo se realizan dos experimentos prácticos. Una aplicación, en el campo de la monitorización del movimiento aplicada al campo de

la rehabilitación y, la segunda aplicación en el campo del procesamiento de imágenes aplicada a la resección hepática laparoscópica.

En rehabilitación, la estimación de la posición se obtiene utilizando la velocidad angular y la aceleración dadas por una IMU. También puede estar dado por un sistema óptico de posición que, en el caso de este experimento, permite estimar el desempeño de un algoritmo.

En diferentes procedimientos quirúrgicos, los modelos preoperatorios son esenciales para incorporar información de otros procesos de imagen al procedimiento. El objetivo de la última aplicación es representar como un spline derivable un modelo preoperatorio de tumor.

Índice general

List of Figures

List of Table

Chapter 1

Introduction

The medical technology sector provides products, services or solutions to improve population health. This sector currently represents the 7,6 % of the total health expenditure in Europe allowing to meliorate the life expectancy and the quality of life of the population. In addition, this sector is characterised by a constant innovation representing around the 8 % of the average global R&D investment. In 2021, 15321 patents applications were field with the European Patent Office (EPO) representing the 8,1 % of the applications and second highest industrial sector in number of applications, just below digital communication with 15400 patent applications [2].

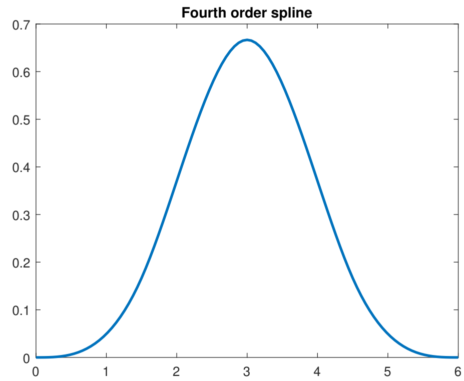
In bioengineering, same as other engineering fields and physics, it is essential developing numerical tools that models different phenomenons. The numerical tool developed in this work will allow to estimate any solution of a mathematical problem modelled by an equation, although the work is mainly focused on differential equations where the unknown function and some of its derivatives are involved [3]. Along this work, solution functions will be approximated with splines by using collocation methods. Moreover, some algorithms will be developed to solve some bioengineering problems and other ones to show the performance of the methods and illustrate the different hyper-parameters choice implications.

The aim of this chapter is to introduce the work and develop the structure of the essay. This chapter is divided in three sections. Section 1.1 explains the motivation of the work, going deeper in the mathematical view. Section 1.2, it shows the objectives of this work, enhancing the most technical part of the work. Section 1.3, the structure of the essay is presented.

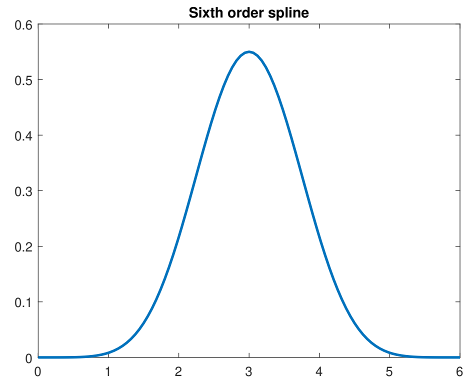
1.1. Motivation

From a mathematical point of view, the objective of this work is not only to show the feasibility of using splines to solve these equations, but also to show the viability and performance of higher-order splines. The technical aspects of the splines will be discussed in Chapter 3, but to introduce the motivation, it is necessary to previously explain how the spline order impacts on the quality of the approximations.

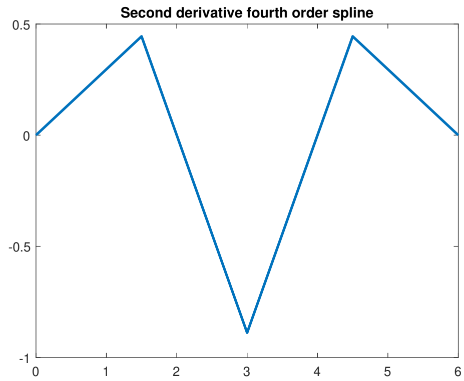
The order of the spline, that corresponds to the order of the piecewise polynomials that shape the spline. In polynomial estimation, it is accepted that the higher the order the greater the smoothness but also the greatest amount of undesired oscillations. As a convention, it is also accepted that even-order polynomials had a better performance than odd-order polynomials [4]. That is the reason why fourth-order polynomials are the standard when it is being worked with splines. As, the order is even and it is low enough to not cause oscillation and it has enough smoothness to return acceptable estimations. At Figure 1.1 two basis splines with even-order and their derivatives are displayed.



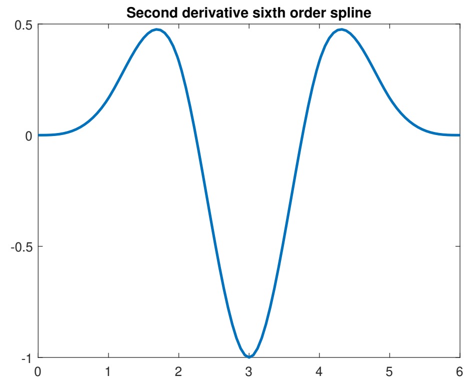
(a) Fourth-order spline.



(b) Sixth-order spline.



(c) Second derivative of fourth-order spline.



(d) Second derivative of sixth-order spline.

Figure 1.1: Basis splines and their second derivatives.

The main problem of fourth-order polynomials arrives when it is desired, not only approximating a specific function, but additionally obtaining a good estimation of some of its derivatives. At Figures 1.1a and 1.1b, the fourth and sixth-order basis splines are represented respectively and, at Figures 1.1c and 1.1d, their second derivatives. In that case, there could be a good approximation of the function as happen in Figure 1.1a but there would not be enough smoothness at the derivative spline to have an acceptable performance at their derivatives as happen in Figure 1.1c. It can be corrected by increasing the order of the spline as happen in Figure 1.1b being the second derivative Figure 1.1d an adequate approximating spline.

Along this work, different spline-based algorithms have been developed. We divide the experimentation in two categories, synthetic and application algorithms. The last ones consist on applications oriented to solve different bioengineering problems that will be explained extensively below. Although, for developing these applications, it is developed the first group of experiments. They are created to go deeper into the method and being able to get an idea of how different decisions taken during the process impact on the approximation. These experiments consists on estimating the solution of several differential equations and obtaining some error metrics by comparing them with their analytical solutions. These error metrics are compared in different frameworks. Also, the aim of these experiments is not only to be used in the development of this work applications but to be used as a guide for the reader in the estimation of its own applications.

The last synthetic experiment that is also a simple practical application. It consists on a non-linear ODE that depends on some parameters. This equation also has an analytical solution so as the other experiments some error metrics could be taken. It models the tumour proliferation with the number of cells at the time the estimation begins, the intrinsic growth of cells and the carrying capacity of the tumour [5]. The Gompertz equation is the model that allows estimating the tumour proliferation. Values from the literature and/or values frequently used are employed for the estimation. In a more practical scenario, at least the initial number of cells should be obtained from the tumour of study.

On the other hand, two different bioengineering applications are developed. One of the applications is is a rehabilitation application in the area of motion estimation and the second one in the field of image processing. Below, it can be found two sections devoted to explain in detail the motivation behind the two bioengineering applications. First, Section 1.1.1 explains the motion monitoring application. Second, Section 1.1.1 explains the image processing application.

1.1.1. Motion monitoring application

This application is developed in the field of rehabilitation that can be defined as the designing of intervention oriented to reduce disability and optimise functioning of individuals in their interactions with the environment [6]. There are different branches in the field of rehabilitation including language training, some physiological therapies, etc. In this application, it is being worked in the field of physical exercise training that is important to improve muscle strength, voluntary movements, balance, etc.

In this kind of therapies, it is required to supervise the correct development of the exercise to ensure the effectiveness. A possible way to supervise the exercise is being supervised by a second person. The main problem with this approach is that hiring a personal coach is expensive. Another problem that could arise is the availability of trained stuff in some places. In this context, virtual coaches are an alternative to perform the supervision. Virtual coaches could support by using the displacement in the three dimensional axis and interpret this information [7].

Generally, the process of obtaining the displacement directly is not a functional approach. It could be done through an optical reference system although it has the inconvenience of installing the equipment. Also, inertial measurement can be used to obtain this displacement through solving some differential equations. At this work, these displacements will be represented as splines. Then, the estimation will be compared with the displacements acquires through the optical system.

The exercises to monitor consists on a set of movements commonly prescribed in different physical therapies, that have demonstrated functional improvements in studies [8]. Concretely, three exercises focused on the lower-limbs will be monitored.

Knee flex-extension (KFE): The subject is sat with a position of 90º of knee flexion. The subject extends the knee in the sagital plane until the maximum. It is performed between ten and twenty repetitions.

Squats (SQT): The subject is standing, sits on a chair and return to a standing position. The subject avoids lateral bending of knee or hip. It is performed between eight and fifteen repetitions.

Hip abduction (HAA): The subject is standing up and moves one of the legs in the frontal plane plane while the other leg remains straight. It is performed between ten and twenty repetitions.

The IMU device will be located at the shin and the reference to the optical system is also in the IMU.

1.1.2. Image processing application

The image processing application is developed in the field of surgical planning. Surgical planning are the method previous to a surgical intervention to define the surgical steps and to collect information in the context of computer-assisted surgery. In this field, it is useful performing some image acquisition before the surgery [9]. These acquisition will give an initial approach of the problematic to threat.

Pre-operative model: The pre-operative model consists on a virtual duplicate of a body component. It is created by segmenting an image acquisition, generally, it is extracted from a CT (Computed Tomography) or from a MR (Magnetic Resonance). It represents the boundaries of the component in a three dimensional way, returning the shape information at the time of the acquisition. Moreover, it is an enabling technology for advanced surgical systems [10].

Liver and liver tumour: The work is realised in the context of liver surgery and in liver tumour. The liver is an organ that can be found in vertebrate animals. Among its function, it can be found the detoxification of the organism and the synthesis of proteins [11]. When a tumour grows in the liver, it is needed to eliminate the tumour and laparoscopic surgery is an option that allows the surgeon to access without large incisions in the skin.

Registration techniques: This procedure is characterised by introducing a light source and a camera in a tube called laparoscope. Registration techniques are needed to fuse the information from the pre-operative imaging and the laparoscope images. Concretely, it is useful these reconstruction techniques to recover the shape from the laparoscopic image. As the laparoscopic image is not enough informative itself, it is useful to use it to estimate the deformation from a given pre-operative model [12].

Generally, these models are geometry objects including a set of points and its triangulation that shape a mesh. This could limit the registration technique as intermediate points are not available with a good estimation and also it does not provide derivatives with enough quality. This is relevant since in registration method, differential equation are a powerful tool to achieve these alignments.

This registration methods were compared in the field of Laparoscopic Liver Resection [13]. It was provided a clinical Data-set that include pre-operative models of fourth patients livers and tumours. In this work the tumour models will be represented as spline and it will be evaluated by using the points provided by the model.

1.2. Objectives

As previously explained, a numerical algorithm will be developed to estimate functions through splines. These functions will be solutions of an equation that models a problem and, with the exception of the image processing application explained at Section 1.1.2, the work will be focused on the resolution of differential equations. Along this work, error metrics along the domain will be taken and the method will be evaluated in its different framework. Along Chapter 3 the method will be explained in detail, introducing hyper-parameters related to the basis and different properties. At the beginning, these characteristics are evaluated. We present the following specific objectives:

1. This work will show that the method is scalable in terms of variate nature, being able to approximate univariate functions and multivariate functions which domain has higher number of subset. Multivariate functions will be approximated at the synthetic experiments to show this property.

2. It will be shown the ability if the method not only solving differential equation, but also, solving systems of differential equation. Other synthetic experiments will be developed showing this property and the solution will be expressed as a a multidimensional spline, corresponding each of the dimensions to the different solutions of the system.

3. Also, at the synthetic experiments, the linearity of the equation is explored. In these experiments, linear and non-linear equations are solved illustrating the capability of the method for reaching a solution in both scenarios.

4. Following with the nature of the problem, another target of the synthetic experiments is intercalating initial value problems and boundary problems indicating the capability of the method of solving different problems formulations. Moreover, these experiments will measure the impact of other decisions taken.

5. The Gompertz equation will consists in a non-linear differential equation. Also, the function is univariate and unidimensional. It will be solved with a sixth-order spline with the parameters extracted from the literature and it will be compared as the other prove of work experiments with its analytical solution. It will show the feasibility of using splines to estimate the tumour growth along time.

6. **Motion monitoring:** The application illustrates the feasibility of using splines to obtain the displacements from inertial measurements. Then, the position is expressed through a differential equation that depends on the acceleration and that assumes that the object moves from an initial position in zeros and from being quiet. Then,

the position is compared with the one provided directly from the optical reference system that is used for comparison. This estimation requires to obtain the orientation of the accelerometers provided from the angular velocity that represents how fast the angular position of an object changes with time. A non-linear three dimensional equation matches the angular velocity with the Euler's angles that describe the object's orientation in a fixed coordinate system

7. **Image processing:** The tumours provided in the database [13] are provided as a set of point in a Cartesian coordinate system. So, the set of points is centred in the coordinate centre and it is transformed into spherical coordinates. Then, the tumour is represented as a bivariate spline which domain are azimuth and elevation and its codomain is the radius.

1.3. Memory structure

The bachelor thesis book essay is divided in six chapters presented at Table 1.1.

Table 1.1: Bachelor thesis book.

Chapter	Title	Description
1	Introduction	General presentation that gives an brief overview of the whole project It includes the incentives and goals of the work.
2	State-of-the-art	It is shown the highest and most recent developments regarding these applications. It includes recent development with splines as other ways of solving these kind of problems.
3	Mathematical background	The technical basis of the tool is specified. It is deeply explained the mathematical nature of each of the elements included at this work, including the spline, the collocation method, etc.
4	Methodology and development	A procedure overview is presented. It includes all the tools used to develop this work, the information used and it presents the general way of handling a problem.
5	Results and discussion	Each of the individual experiments that shape this work are detailed. The discussions of each of the results are developed.
6	Conclusion and Future Lines	A general view of the outcomes and its interpretation is presented. It is included the next steps that should be addressed to keep improving the work.

Chapter 2

State-of-the-art

To begin with an historic perspective, the term spline comes from the East Engalia, where in the 18th century the Oxford English Dictionary found its first recorded usage. The term spline refers to a wide class of functions that are essentially defined piecewise by polynomials. It is agreed that he term spline in connection with smoothness and piecewise polynomial approximation was first presented by the mathematician Isaac Jacob Schoenberg in his article a first class of analytic approximation formulas in 1946 [14].

The ideas behind have their roots at the aircraft and shipbuilding industries. It was desired to design in small model and then create larger ones. Lofting was technique used by the British Aircraft industry that creates this larger models by passing thin wooden strips through the discrete points generated by the smaller model.

Then, in the late 1950s and in the early 1960, the mathematical spline was developed in the context of the automotive industry. Some parallel independent beginnings occurred with Paul De Casteljau at Citroën, Pierre Bézier at Renault, Garrett Birkhoff, Paul Garabedian and Carl De Boor at General Motors. One of the De Casteljau paper was published in 1959, De Boor published several papers in early 1960 from its work at General Motors and the work at General Motors was also described by Birkshof.

At this work, the splines are used in the context of differential equations, an approach that was introduced by Carl the Boor at the book 'A practical guide to splines' [15] where it is presented the Collocation methods to solve differential equations. These splines are estimated from the basis splines which are explained in Boor's book 'Box Splines' [16].

At this chapter, we present some similar works in the literature. The articles has been found with the browser Google Scholar. At the research some keywords including 'Spline', 'Bioengineering', 'Motion monitoring' and 'Image processing' are used to find the articles. The criterion for including it at the state-of-the-art has been the closeness to the application developed, the impact expressed by the number of citation, how recent it is prioritising articles which publication date was less than five years and the how innovative the article is.

Due to the nature of this work, the chapter is divided in two sections. Section 2.1 presents other work similar to the motion monitoring application applications. Section 2.2 contains parallel approaches to the image processing application.

2.1. Motion monitoring

Motion monitoring is not only relevant in the field of rehabilitation, it is also key in fields like robotics' automated driving technologies. In all these fields, it is essential to have a good estimation of the position to feed the system determining its good performance. As in the work presented, when it is worked with inertial measurements, the first objective is to provide a good orientation estimation, which also allows integrating the acceleration and obtaining the position.

It is proposed a uniform B-spline fusion based approach to fusion data in [17]. In the same way that in our work, angular velocities and linear acceleration are fused to obtain pose, velocity and acceleration. In contrast to our work, it is used a axis-based approach for obtaining the rotation representation. Once the rotation is obtained, it is computed the acceleration in a similar way than this work. Finally, it compares the results with the ones obtained with pose-graph-based methods that requires pre-integration showing the feasibility of the method.

A most expanded approach was proposed in [18] where it is proposed not only the IMU data fusion but to the fusion with Light Detection and Ranging (LiDAR) to improve modelling and location accuracy. In this case, neither Angel-Basis, neither Euler angles but Quaternions based approach was used to estimate the rotations. Splines are also used to model the trajectories of the IMU information and then the information is corrected with the LiDAR data. The authors concluded that with the simulation performed, their algorithm can solve effectively the parameter calibration and the fusion localisation.

Another expanded approach was proposed in [19] which fuses the IMU data with visual information. The work estimates the position of the camera with the objective of obtaining an scale factor without the need of a second camera. One interesting aspect is that it uses the rotation derived from the position without scale obtained from the vision algorithm and complementing the one provided by the angular velocity and reducing the drift. At the article, both approaches are compared one spline-based and another discrete approach founded on an extended Kalman filter. Both approaches provide accurate results in the quality of scale estimation.

Following the same line, the work [20] also proposes a camera-IMU fusion system to compensate the Rolling Shutter that is an image distortion effect caused by the motion between the camera and the scene. The calibration method uses splines to allow considering the Rolling Shutter continuous in the time domain and it introduced the IMU information in the derivatives of the spline as an axis-based approach. This approach leads to a more accurate and consistent calibration.

The inverted problem using splines is threaded at [21]. It is tried to find information of motion parameters from the estimation of the position to be used in the motion compensation of remote sensing loads. It is concluded that it can successfully obtain the information and obtain better performances than other methods actually used as transfer alignment.

2.2. Image processing

Splines had been successfully applied in computer vision discipline, that is the field that acquires, processes and analyses digital images or videos and, among its sub-fields, it can be found object detection, motion estimation, image restoration, 3D pose estimation, etc. Besides splines allow solving differential equations, splines decrease the computational capacity needed by removing in many instances the necessity of additional smoothness constraints and the correlation windows centred at each pixel [22].

As previously explained, splines are useful in the field of image registration. In [23] a B-spline mesh was used to model the deforming geometry of the intestine where the surface deformation were computed through Green-Lagrange strain fields. A non-linear optimization scheme was used to mitigate the tracking errors. It was obtained that the Root-Mean-Square error stays under 1 % and it was reduced the rate of strain error on 1 97 % during the synthetic tests.

Also in the field of registration, and also in the field of registration images of livers it is proposed a non-rigid registration scheme in [24] to fusion Computed Tomography and Ultrasound. The main difference with respect our application is that the registration is oriented directly to the CT, not the pre-operative model from the CT. It is successfully compared with an approach that uses gradient orientation information.

Another non-rigid registration deformation, in this case oriented to 2D/3D registration is proposed [25] where lungs and phantom data were used to test the method. It allows the alignment with two orthogonal angle projections. The splines are used to create Thin Plate Spline that is used to interpolate the small deformations. It was obtained a Dice greater than 0.97 and a cross-correlation greater than 0.92, showing the ability of the model to track lung tumours.

Also, in the field of geometry representation it can be found a work [26] where splines are used to design a geometry kernel of Group convolutional neural network (G-CNN). These kind of networks are equipped with the geometry and the spline is used to create a basis for any arbitrary Lie group. This allow to solve the limitations that implies other approaches like using a mesh limited to discrete groups that leads the grid intact or compact groups. It was studies in two datasets, cancer detection and histopathology improving the results of traditional approaches in both cases.

Moreover, in the field of clinical image segmentation at the work [27] splines are proposed to be used for Transrectal Ultrasound (TRUS) segmentation. Also, a Magnetic Resonance Image is segmented through nnU-Net. The aim of the article is to be able to compare both images in prostate cancer. The splines are used for fitting a super-ellipse and segmenting the TRUS image, obtaining a DICE coefficient greater than 87 % in all cases.

At [28] it is developed an application in the field of the reconstruction of the k-spaces called B-spline Parameterised Joint Optimisation of Reconstruction and K-space Trajectories (BJORK). It is an MRI application where splines are used to parameterise trajectories. It is used to reduce the number of parameters and it makes local improvements possible through enabling multilevel optimization. The algorithm presented also includes a neural-network based reconstruction that makes improvements on image quality through compensating sensing-based reconstruction. It also leaves for a future line to define the objective function for the trajectory optimization. Finally, at the work [29] the BJORK approach is used for comparing their projection-based scheme for obtaining the k-trajectories in non-Cartesian acquisition settings.

After an overview of splines in the literature and the recent advances in the topics covered at this work it is important to highlight that splines are not the main state-of-the-art tools when working on this problems. Other types of numerical tools are currently more used and more covered.

Chapter 3

Mathematical Background

This chapter is devoted to explain the fundamentals needed for developing this work. It is an overhaul of the mathematical base used. Section 3.1 gives an introduction of approximations problems. Section 3.2 details the spline theory that is the heart of this work. Section 3.3 gives an overview of differential equation and the ways to solve them. Section 3.4 explains the background needed in the motion monitoring application.

3.1. Introduction to the problem of unknown functions

The use of spline fits in a problematic where it is desired to expand over a predetermined domain some information that previously it was not completely known over that domain. In fact, it is desired to find a function that turnouts the information [16].

There are different reasons that explained the necessity of estimating this function. One reason is it does not exist an analytical solution to the problem it is being faced. Another reason could be that there exist a function that fulfils the requirements but it is too complex and it is not computationally efficient to work with it.

Among the practical situations where this problem emerge is data fitting. In data fitting, it is obtained a collection of information of an unknown function and from the data obtained it is desired to obtain a function over a desired domain. Another practical situation, is to find a solution to a differential equation where the initial values or the contour are known.

3.1.1. Polynomial interpolation and approximation

A first initial approach to find this solution is to fit a polynomial according the data known. A polynomial is an expression where a dependent variable depends on independent variables, independent coefficients and the operations of addition, subtraction, multiplication and positive integers [4].

Example of polynomial:

$$p(x) = a_0 + a_1 x^1 + a_2 x^2 + ... + a_n x^n \tag{3.1}$$

The polynomial is characterised by the coefficients that are constants that are not indeterminate and in the example are represented with the letter a. The polynomial is also characterised by its order, that it is given by the largest variable exponents, being in the example a polynomial of grade n and order $n + 1$.

The order of the polynomial gives the degrees of freedom of the polynomial. It entails that the order gives the number of independent constraints that a polynomial can fully comply. For example, in a data fitting problem where six samples are given, if it is desired to ensure that the polynomial fits exactly the six points a sixth-order polynomial should be found. This kind of estimation is called interpolation.

It is not feasible to find an unique polynomial with less constraints than degrees of freedom. Nevertheless, it is possible to find a polynomial with higher constraints than degrees of freedom. In this case, instead of finding a polynomial that fits exactly the restrictions given, it is find a polynomial that makes the error of the fitted data minimum. This kind of estimation is called approximation.

The residuals are the difference between the given constraint and the results of the estimation. There are different methods to perform the approximation depending on the way it handles the residuals and minimises them. The most standardised is the method of Least Squares. In this method, the sum of the squares residuals is minimised by setting the gradient of the loss to zero [30].

In the case of Linear Least Square the regression model is set as:

$$f(x, \beta) = \sum_{j=1}^{m} \beta_j \phi_j(x) \tag{3.2}$$

Then, letting:

$$X_{ij} = \phi_j(x_i) \tag{3.3}$$

The loss function of the data is computed as:

$$L(D, \beta) = \|Y - X\beta\|^2 = (Y - X\beta)^T (Y - X\beta) = Y^T Y - Y^T X\beta - \beta^T X^T Y + \beta^T X^T X\beta \tag{3.4}$$

Then, the loss gradient is:

$$\frac{dL(D,\beta)}{d\beta} = \frac{d(Y^TY - Y^TX\beta - \beta^TX^TY + \beta^TX^TX\beta)}{d\beta} = -2X^TY + 2X^TX\beta \quad (3.5)$$

Finally, setting the gradient to zero as in Equation (3.6):

$$-2X^TY + 2X^TX\beta = 0 \Longrightarrow X^TY = X^TX\beta \Longrightarrow \hat{\beta} = (X^TX)^{-1}XY \quad (3.6)$$

3.1.2. Advantages and disadvantages of polynomials

As a summary of this kind of estimation, it can be highlighted some advantages. These advantages mainly relies in the simple nature of polynomials developed before. However, this simple nature can also lead to some characteristics that makes polynomials a poor estimator [4].

Among the advantages, the computational cost that implies working with them is relatively low. Due to its nature, the estimation of the coefficients is computationally simple, in the case of interpolation, it is just required to solve a linear systems. Also, in the case of approximation, some methods like, the explained before Linear Least Squared allows to compute them efficiently.

Moreover, the evaluation of the polynomials is also easily performed since it only requires simple basic operation as multiplications and additions. Other operations could be more computationally inefficient, for example, the evaluation of trigonometric functions have a higher computational cost.

Also related with the nature of the polynomials, it is really easy to takes integrals or derivatives of the polynomials. This property is really useful and it is essential for the development of this work. Therefore, it is important to highlight that this property is restricted by the order of the polynomial, since when a derivative is taken the estimation polynomial is of a first lower-order.

On the other hand, the simplicity of these polynomials could involve some disadvantages that made the estimation worse as it is needed to increase the domain or work. One case is the estimation of bounded function, as the limit of polynomials tends to infinity or minus infinity when x tends to infinity. Another issue consists on the fact that even-order polynomials cannot be only positive or only negative functions although this property could be necessary in the estimation.

3.1.3. Runge's phenomenon

Another well studied problem derived from polynomial approximation is the Runge's phenomenon. It consists on the oscillation of edges that occurs in higher-order polynomials [31]. It is a typical problem that occurs in interpolation with equidistant evaluation points. These phenomena leads to obtain an incongruous estimation with significant errors out of the evaluation points.

Among the possible solutions it could be considered changing the evaluation points to the Chebyshev nodes that are the ones that minimises the Runge's phenomenon effects. If it is not to possible due to the sampling nature an alternative could be using S-Runge algorithm without resampling that map the original set of nodes on the set of Chebyshev nodes [32].

Another intuitive solution could be performing approximation where this kind of phenomenon is easier to handle. It can be performed approximation with a lower-order polynomial than the one that presents the oscillations. Also, the Bernstein polynomials can uniformly approximate functions in closed intervals but they are computationally more expensive.

Other families of approaches try to solve this problem by working with additional constraints. The external fake constraints interpolation introduces additional constraint of type $P''(x) = 0$ outside and at the extremes of the work interval [33]. As well, it can be performed constrained minimisation, an example could be interpolating a polynomial with a first derivative that has a minimal L^2 norm.

These are interesting ways of handling oscillation caused by Runge's phenomena since these methods keep most of the advantages of working with polynomials. Nevertheless, they are less useful with the first family of problems explained derived of the necessity of working with a great domain of X.

Moreover, there is another approach that allows to keep most of the advantages of polynomials at the same time that it is able to handle the Runge's phenomenon while it works well in large range of X. This method consists on using piece-wise polynomial function instead of using a single polynomial function. This function is called spline.

3.2. Spline theory

A spline consists on a numeric function determined by piece-wise polynomial functions. Then, for characterising a spline, it is required to have a strictly non decreasing set of point in the domain of x that divide the domain in sub-intervals called sequence of breaks.

The spline function turnout a value of zero outside the interval and in each sub-interval the spline is defined by a specific polynomial function [15].

One way to describe a spline, called Piece-wise Polynomial form (ppform), is in term the breaks and the local coefficients. So, to construct a spline with l pieces, sequence of breaks $\varepsilon_1, ..., \varepsilon_{l+1}$, a local coefficients matrix of size lxk, where each row contain the coefficients of each local polynomial, and order k, order of the components polynomials, it could be described as:

$$p_j(x) = \sum_{i=1}^{k} (x - \xi_j)^{k-i} c_{ji}, \quad 1 \leq j \leq l \tag{3.7}$$

This way of representing the spline could be really interesting as it makes really easy and intuitive to interpret the curve to the human mind. Nevertheless, it is computationally inefficient since it requires more, or in the best case equal, coefficients than degree of freedom of the curve.

3.2.1. Smoothness of the spline

The reason is that the spline is not just a set of polynomials merge together. The spline is a curve where continuity and differentiability could be desired and it is characterised by the sequence of knots.

This smoothness, inside the sub-interval, is given by the nature of the polynomials selected. A polynomial is by definition a continuous function and it is possible to take derivatives until the order of the polynomial where the derivative becomes zero. All these previous derivatives will have the property of continuity.

By the nature of the spline, the maximum smoothness it is achieved at the breaks is until $k - 1$ order. If k order derivative were reached, then the adjacent polynomials will be merged. When it is forced this smoothness, some degrees of freedom are lost. For example, to achieve continuity over all the curve, $l + 1$ degree of freedom are lost since at each break it is given the constraint that both polynomials should be equal. Moreover, if it is desired to be continuous and one time differentiable, $2(l + 1)$ degrees of freedom will be lost by the same reason.

In the spline literature, this smoothness constraint are defined by the knots sequence. The knots sequence consists in a non-decreasing sequence of points in the domain x where only the breaks points can belong to that sequence and all the breaks belongs to that sequence. These smoothness constraints are specified through the number of times a break is in the knots sequence. If a break is only once it means that at that breaks there is the maximum smoothness possible, and each time the break is repeated a smoothness constraint is lost. The order in which this constraints are lost are from the deepest derivative until the continuity.

Due to this, each breaks should be at least once in the knots sequence and a maximum of k times that means that at that break the function is neither differentiable neither continuous. One common scenario is that the curve is desired to have the maximum smoothness possible along all the intervals and to not have any kind of smoothness at the extreme. In this case, the knots sequence will consist on all the breaks inside the interval repeated one time and the external breaks repeated k times.

Then, to calculate the degrees of freedom of the spline, it is needed to take into account that without the smoothing constraints each spline has k degrees of freedom. That as there is one polynomial between each sub-interval there are l polynomials. The knots sequence could be interpreted as the real degrees of freedom that each polynomial gives to the spline, as the number of times it is present are smoothness constraints that are not given. Although, as there is one more break than polynomial k must be subtracted to the number of knots to obtain the real degrees of freedom of the spline.

3.2.2. B-spline

Instead of using Piece-wise Polynomial form, that will require lxk coefficients independently of the degrees of freedom of the polynomial. Basis spline (B-spline) functions can be used being able to work with a coefficient matrix that have the same number of coefficients than the spline degrees of freedom.

B-spline of order k with sequence of knots $t_0, ..., t_j$ is a piece-wise polynomial function that is unique for a given sequence of knots and satisfy the following conditions:

$$B_{j,k}(x) = \begin{cases} 0 \ if \ x < t_j \ or \ x > t_{j+k} \\ non - zero \ otherwise \end{cases} \tag{3.8}$$

$$\sum_i B_{j,k}(x) = 1 \tag{3.9}$$

These B-splines are basis function in a way that all possible spline function with the same-order and the same knots sequences can be constructed in an unique way as a linear combination of B-splines. Then, this curve is constructed multiplying the spline function by a vector-valued constants a_i that can be called control points or also B-coefficients. It is obtained a resulted B-spline form (B-form):

$$f(x) = \sum_{j=1}^{n} B_{j,k} a_j \tag{3.10}$$

These functions can be evaluated by using the called De Boor's algorithm. It consists on a generalisation of De Casteljau's algorithm for Bézier Curves that allows to evaluate numerically stable splines in B-form. There are faster variants of the algorithm that suffer from lower stability.

It is performed the following recursion formula:

$$B_{j,0} = \begin{cases} 1 & t_j < t < t_{j+1} \\ 0 & otherwise \end{cases} \tag{3.11}$$

$$B_{j,k}(x) = \frac{x - t_j}{t_{j+k-1} - t_j} B_{j.k-1}(x) + \frac{t_{j+k} - x}{t_{j+k} - t_{j+1}} B_{j+1,k-1}(x) \tag{3.12}$$

In contrast to ppform, B-form is less intuitive to interpret but fundamentally they are two different ways to represent a spline and each spline has an unique way to be represented in ppform and in B-form. So each way of representing the spline has an equivalent way of representing it in the other form.

The main different relies in the fact that while in ppform the smoothness information is in the coefficients matrix as it is defined by its breaks, as the B-form allow to represent by its knots the coefficient matrix could be smaller.

The stable recurrent relations in the Boor's algorithm also allow to obtain the pp-form coefficients from the B-form and it is possible to obtain the B-coefficients from the coefficients in ppform by using the following conversion:

$$a_j(s) = \sum_{i<k} (-D)^{k-i-1} \frac{(y_{j+1} - \tau)...(t_{j+k-1} - \tau)}{(k-1)!} D^i s(\tau) \tag{3.13}$$

Along the work, it has been worked with splines in B-form so throughout the essay when using the term spline it will be considered that B-form is being used unless the opposite is specified.

As it had been explained, one of the most interesting properties of polynomials is the fact that it can be taken derivatives of the polynomial and keep a simple polynomial expression. In the same way it is possible to take derivative of the spline obtaining another spline of order one less.

The derivative of the B-spline:

$$DB_{j,k}(x) = k\left(\frac{B_{k,k-1}(x)}{t_{j+k} - t_j} - \frac{B_{k+1,k-1}(x)}{t_{j+k+1} - t_{j+1}}\right) \tag{3.14}$$

Then the derivative of the spline will be:

$$Df(x) = D\sum_{j=1}^{n} B_{j,k}a_j = \sum_{j=1}^{n} k\frac{a_j - a_{j-1}}{t_{j+k} - t_j}B_{j,k-1} \tag{3.15}$$

With respect to the spline of the derivative it is important to highlight that it could have the same number of knots if the original spline was continuous over all x or could have less nodes if it was not continuous over some breaks. As well the order changes, it is one less, the degrees of freedom of the spline and in consequence the number of B-coefficients changes.

3.2.3. Tensor product splines

Splines are mainly used to estimate functions. During the essay, it has been considered univariate splines that are useful to estimate univariate functions. Then, it is needed a solution for being able to estimate multivariate functions.

These multivariate splines came in the form of tensor products of univariate splines. The background idea consist on creating a multivariate function with a product of univariate function so if $f_1(x)$ is a function of x and $f_2(y)$ is a function of y, it can be obtained function of x and y as $f(x, y) = f_1(x)f_2(y)$.

In this way, instead of having a vector of coefficients as in the univariate spline. An array with a dimension equal to the number of variables is used to define the spline.

Then a multivariate spline could be defined as:

$$f(x_1, ..., x_v) = \sum_{i=1}^{m} ... \sum_{j=1}^{n} B_{i,k}...B_{j,k}a_{i,...,j} \tag{3.16}$$

In that case, it has been treated one dimensional splines, but it could exist the necessity of approximating functions that have more than one dimension. In this case, it is used a similar approach than the one used before by generating each of the dimension with a spline.

By convention, the arrays of coefficients of the different are ensemble so a final array of coefficients with a first dimension corresponding to the dimensions of the spline and the rest of the dimensions corresponding to the variate nature.

To sum up, splines are a powerful mathematical tool commonly used in the estimation of unknown functions like in interpolation, approximation, noise reduction, etc. During this work, the main area of work is numerical analysis for solving differential equations.

3.3. Differential equations

Differential equations consists on equations that can be dependent on functions and on the derivative of those functions. If the function is univariate, it is called Ordinary Differential Equation (ODE). If the function is multivariate, it is called Partial Differential Equation (PDE). Solving this differential equations consist on finding an analytical solution or an approximation of the function that satisfy the equation [34].

Prior to finding the function, it is important to verify that the problem is well-posed. It means that the solution exists, it is unique and change continuously with the constraints given. Then, it is established that the function $f(x)$ and its derivative functions involved in the differential equation are continuous in an open interval in R and that the points x_0 given as a conditions belong to R.

The differential equation can also be characterised by its order. The order of the differential equation is the highest order of the function derivative that arises at the equation. Along the work, first order and second order differentials equations are solved. The order of the differential equation determines the constraints needed to verify that the solution exists and that it is unique. Along this work two kind of differential equations problems are threaten Initial Value problems and Boundary Value problems.

Initial value problem: In these problems, an unique point in the domain, that correspond to the lower boundary, is given. The Picard-Lindelöf theorem verifies the existence of an unique solution if f is continuous in the region and y_0 verifies the Lipschitz condition [35]. In an univariate first-order differential equation with a solution can be expressed as:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases} \tag{3.17}$$

It can be expanded to multivariate equations: This kind of problem can be expanded to multivariate functions and one of the boundaries must be given, it can be expressed:

$$\begin{cases} y'(t, x) = f(t, x, y(t)) \\ y(t_0, x_0) = y_0 \end{cases} \tag{3.18}$$

It can be expanded to higher-dimensions:

$$\begin{cases} y'(t) = f(t, y_1(t), ..., y_n(t)) \\ y_1(t_0), ..., y_n(t_0) = y_{1,0}, ..., y_{n,0} \end{cases} \tag{3.19}$$

It can be expanded to higher-orders:

$$\begin{cases} y^n(t) = f(t, y(t), y'(t), ..., y^{(n-1)}(t)) \\ y(t_0), y'(t_0), ..., y^{(n-1)}(t_0) = y_0, y_0'..., y_0^{(n-1)} \end{cases} \tag{3.20}$$

Boundary value problem: In these problems, the way of working is similar to the initial value problems. Nevertheless, the differential equation fulfils the constraints given at the extreme of the working zone. As initial value problems, it can be expanded to multivariate functions, to higher dimension functions and to higher-order equations. With respect to the order, it can be expressed as a boundary value problem from second-order differential equations since first-order only requires one constraint [36].

By extrapolation an example of a second-order ordinary differential equation. In initial value problems, two constraints in the lower boundary are needed, one corresponding to the value of the function and another corresponding to the value of the derivative of the function. In parallel, the problem can be rewritten as boundary value problem with two constraints that correspond to value of the function in both extremes.

$$\begin{cases} y''(t) = f(t, y(t), y'(t)) \\ y(t_0), y(t_1) = y_0, y_1 \end{cases} \tag{3.21}$$

Other property that defines a differential equation, and that has impact on the way it is solved, is the linearity. A linear differential equation consists on a linear polynomial on its function and its derivative functions. It can be expressed in the following way:

$$a_0(x)y + a_1(x)y' + ... + a_n(x)y^{(n)} = b(x) \tag{3.22}$$

3.3.1. Solutions to differential equation

Above all, there are two ways of finding a solution to a differential equation. The first one consists on finding an analytical solution to that function. The other one consists on processing the problem as an estimation problem. The last approach is called finding a numerical solution [37].

The main advantage of the first approach is that the error, given by the function at its evaluation, could be done unlimited small as there is not an estimation error. The main disadvantage is that this kind of functions are often either not possible to calculate, either computationally really expensive to evaluate, either not handy to calculate the analytical solutions.

Numerical methods consist on an approximate computer tool designed for solving a mathematical problem. The numerical methods are generally used when this problem either does not have an analytical solution or either this solution requires a high computational cost and a numerical method can provide a enough quality solution with a much lower computational cost. That is the reason why the most common approach is the second one.

In this context, several numerical methods are used for different purposes like computing values of functions, interpolation, regression, extrapolation, solving equations and system of equations, solving eigenvalues or singular value problems, optimization, evaluating integrals or solving differential equations.

In the case of numerical methods, an approximation of the ordinary differential equation, with only a single independent variable, or the partial differential equation, multivariate function with partial derivatives, is found. Spline based methods allow to find an approximation that is composed of several piecewise polynomial functions. These methods are widely used in computational fluid dynamics and in engineering disciplines like computer vision.

In this approach, the numerical method refers to the mathematical tool created to solve the problem. The numerical algorithm is the implementation of a numerical method in a programming language. For a general numerical methods, at least, two properties are desired stability and convergence [38].

1. **Stability:** The numerical method is considered stable if the errors do not add up along the domain.

2. **Convergence:** The numerical method is considered convergent if when the steps or the interval sizes tend to zero the solution converges to the analytical one.

The method is also characterised by the way it performs the estimation, the kind of problems it is able to solve and if it requires refinement iterations. During this work collocation method presented at Section 3.3.2 are selected. It performs the estimation using splines and it is able to solve initial value problems and boundary problems in ODE and PDE. It is a method that does not require iteration when solving linear differential equations and requires them when non-linear differential equations are solved. Previously, a brief overview of the most common methods is presented.

Euler's method The Euler method is the simplest method and it allows solving ordinary differential equations in the case of initial value problems. Mainly, first-order differential equation. It obtains the estimation iterative in a set of points separated by a distance h. It is interpreted as a differential equation consisting on a formula by which the slope of the tangent line to the curve [39]. There are four variants presented.

The points can be expressed as:

$$t_{n+1} = t_n + h \quad (3.23)$$

Table 3.1: Variants of Euler method.

Method	x_{n+1}
Explicit method:	$x_{n+1} = x_n + hf(t_n, x_n)$
Implicit method:	$x_{n+1} = x_n + hf(t_{n+1}, x_{n+1})$
Improved method:	$x_{n+1} = x_n + hf(t_n + \frac{h}{2}, x_n + hf(t_n, x_n))$
Modified method:	$x_{n+1} = x_n + \frac{h}{2}(f(t_n, x_n) + f(t_{n+1}, x_n + hf(t_n, x_n)))$

Taylor's method The Explicit Euler method can be seen as a first-order Taylor expansion of $x(t)$. In the same way, if the function is differentiable enough, it can be obtained a method of a higher-order than one [40].

$$x_{n+1} = x_n + hx'_n + \frac{h^2}{2}x''_n + ... + \frac{h^p}{p!}x_n^{(p)} \quad (3.24)$$

Runge-Kutta's method The main problem of Taylor's method is that it is not practical to evaluate in derivatives higher than the third-order. To solve this in a different way, Runge-Kutta's method is based in evaluating the function in several point of each interval so it is equivalent to higher-order Taylor's methods [41]. The classical method is the fourth evaluation one:

Table 3.2: 4th Taylor's method.

x_{n+1}	$x_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$
K_1	$f(t_n, x_n)$
K_2	$f(t_n + \frac{h}{2}, x_n + \frac{h}{2}K_1)$
K_3	$f(t_n + \frac{h}{2}, x_n + \frac{h}{2}K_2)$
K_4	$f(t_n + h, x_n + hK_3)$

3.3.2. Collocation method for solving DE

In contrast to this methods, Carl de Boor presented an spline based method called collocation's method [15]. At a first mention, it is needed to highlighted that some authors considerate collocation's method as an alternative way to achieve implicit Runge-Kutta's method. Although, the great differences in the way it is computed and the way the outcome is expressed it is commonly treated separately. To establish some parallel lines, the order of the spline would be the parallel to the order of the RK method and the domain of each sub-spline will the h. The improvements with respect to more classic RK and other methods like the one previously explained are numerous.

Continuous domain: It provides an estimation based on a continuous function in all the domain. It reduces the computational cost in general terms in the evaluations. Specially, since it allows evaluating outside the nodes, it allows estimating for an h that is enough small to establish a good estimation but it is not constricted to be small by the evaluations needed.

Easily extrapolated: In contrast to the family of methods, where the previous belong, it does not required a great amount of variant when it is changed the nature of the function, the order of the differential equation, the type of problem, etc. For example, regarding the order, it is only needed to be taken into account that the spline must be at least first-order higher than the differential equation. It provides a general framework that could be intuitive changed for any kind of problem.

This work can be seen as an extension that goes in depth in the method and it allows to have in a more condensed and practical way the nature of splines and the way to estimate through them any function that depends on an equation. Among the ways the work it is extended:

Nature of the functions: De Boor's books only provided an example to solve a non-linear differential equation. Partial differential equations that leads to multivariate functions and systems of equations that leads to multidimensional functions are also discussed on this work.

Engineering applied works: The work developed by De Boor does not included any kind of application and it does not provide any support for equation depending on sampled information. At this work, two different applications are developed including a motion monitoring and an image processing. It also relatively easy to extrapolate to another applications if it desired.

Collocation method consists on a spline-based numerical method that allows giving a solution to ODE, PDE and integrals. It also allows different kind of numerical problems including initial value problems and boundary problems. It is able to solve different differential equations independently of its order and of its linearity.

The method is based on deforming a spline by choosing adequately the coefficient until that spline provides a solution to the equation. The first step consists on creating a spline, setting initially its coefficients to ones, that fulfil the requirements of the function to be estimated. That spline is uniquely defined by its sequence of knots.

It has been previously explained previously what a sequence of knots is and how to create it. The first think it should be taken into account is that the sequence of knots must be coherent with the domain and the image of the function, it must have the same variate nature and the same dimensional nature. The most important thing of the sequence of knots is that it should have the less smoothness possible at the extremes of the function domain. Regarding, the smoothness of the breaks and the distance between breaks, there is no a general rule. By default, it is common that the breaks are selected equidistantly and the number of breaks depend on the flexibility the spline need to estimate the function. Regarding the smoothness of the breaks, it is also not closed but at the work the maximum smoothness is given.

Regarding the order, the unique thing that is closed is that the spline must be at least one-order higher that the differential equation. At the work, higher-order splines are used to improve the estimation of the estimation derivatives.

The collocation sites are the points where the differential equation is fulfilled if the coefficient will be interpolated or quasi-fulfilled if they are approximated. The selection of this points is critical in the quality of the estimation. If they can be selected, the optimal convergence is given by the Legendre Quadrature as in Equation 3.25 being δ the zeros of the Legendre polynomial of order equal to the number of collocation points per interval. Those are also the sites used in the standard Gauss quadrature rule.

$$\tau = \frac{\varepsilon_{i+1} - \varepsilon_i + \delta(\varepsilon_{i+1} - \varepsilon_i)}{2} \tag{3.25}$$

The number of points per interval is also a non-closed topic. Parallel to the number of breaks, increasing the number of collocation points improves the quality of estimation, but at certain point the improvement becomes neglectuble while the computational cost keeps increasing in a linear way.

Then, to be able to shape the spline, it is needed to estimate the coefficients. By evaluating the spline, or the derivative splines, in the constriction points but taking the coefficients as ones, it is possible to express the equation in function of the coefficients. In the case of tensor product splines it could be done through the Cartesian product in each of the domains. In case of the multidimensional splines, the coefficients must be concatenated, so each equation depends on all the coefficients. Then, a system of equation is generated through the constraint of the differential equation at each collocation points and the others restrictions provided by the problem.

The system is transformed into a linear equation in function of the coefficient. If a linear differential equation is being solved, the system will be linear by its nature. If not, a linear viscosity term is added and the newton method is applied to the coefficients. It consists on a first-order Taylor expansion with respect to this term [42]. The linear system must have at least as many equations as coefficients and it must be ensured that it is possible to estimate all the coefficients.

The linealization process in the non-linear differential equation required an initial guess of the coefficients. Obtaining the initial guess is challenging and key for obtaining an estimation that converges, although the literature leaves it open. At this work, it is mainly worked by expanding the additional constraints of the differential equation.

Viscosity term: In the literature the viscosity term is usually vanished in an iterative way but it also can be set directly to zero in some equations. As the initial guess, there is not a closed guide to what value should be set or how it should be vanished if it is needed [42].

Once the coefficients are estimated, to have the estimating spline they should be returned to the shape that correspond to the nature of the function. Once, the coefficient matrix is obtained and reassembled, the spline is fully constructed and the solution to the differential equation is been achieved.

For obtaining the value of the differential equation as a point, the point is evaluated in De Boor's recursive formula and it is used the coefficients previously estimated.

3.4. Motion monitoring estimation

One of the application developed at this work is the motion monitoring using the information from Inertial Measurement Units (IMUs). An IMU consists in an electronic device composed by accelerometers, gyroscopes and sometimes magnetometers. It provides information of specific forces and angular rate and orientation (if the magnetometers are included) in three orthogonal axis. At Figure 3.1 there is a 3D IMU gyroscope.

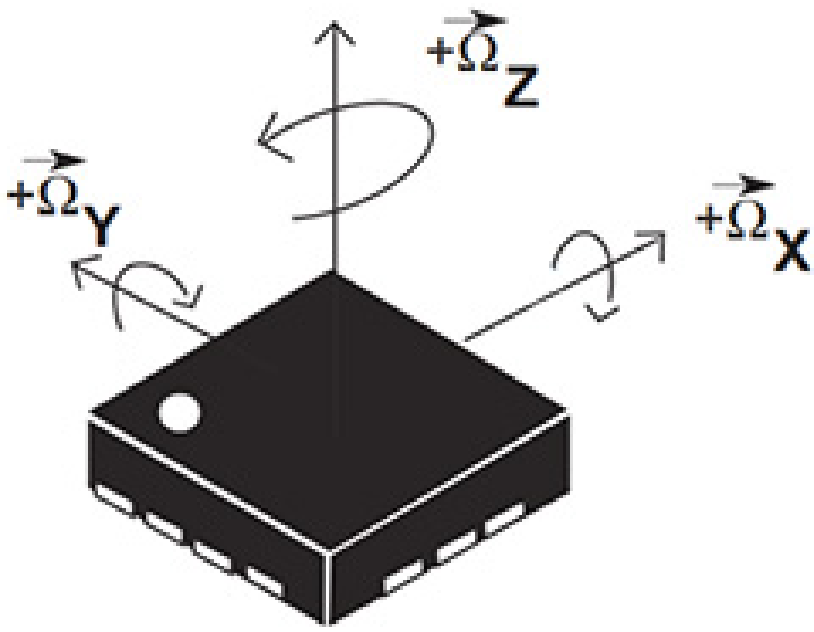


Figure 3.1: IMU gyroscope [1].

3.4.1. Orientation estimation

In the motion monitoring estimation a first step consists on the estimation of the orientation. At this process, the orientation at a time point is set as a reference and along the time the orientation is estimated with respect to that point.

The gyroscope is a mechanical device that measures the angular velocity. It consists on a spinning disc where an spin axis is free to assume of any orientation by itself and it is not affected by the tilting or the rotation of the mountain based on the conservation of the angular momentum. At the IMU, there is a mounted three gyroscopes measuring the angular velocity in each of the axis. The gyroscope allows to know the orientation of the system with respect to the initial frame at all the time.

In a 3D Cartesian's coordinate system, a change in the the relative orientation is described by an orthogonal three times rotation matrix. This matrix can be parameterised through the Euler angles. It uses three successive rotation to describe the relative orientation. To understand the fundamentals of Euler angles, it is needed to take into account that the rotations over the three axis affect the three axis and only in an infinitesimal way it could be that the rate of change of the Euler angles equals the body-referenced rotation rate. Then, the way that the Euler angles relates to the angular velocity is described as [43]:

$$\vec{\omega} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\psi & \sin\psi\cos\theta \\ 0 & -\sin\psi & \cos\psi\cos\theta \end{bmatrix} \begin{pmatrix} \psi' \\ \theta' \\ \phi' \end{pmatrix} \tag{3.26}$$

Those angles are obtained through solving the non-linear differential equation. At the work, collocation methods are used to estimate in a spline the orientation. The system is solved as a non-linear multidimensional univariate spline that depends on the time. Then, at any point the rotation can be obtained by calculating the total matrix rotation of the system. This matrix is obtained by multiplying the three individual inverse rotation matrices of each of the angles.

$$M = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ -\sin\psi\cos\theta + \cos\psi\sin\theta\sin\phi & \cos\psi\cos\theta + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\theta + \cos\psi\sin\theta\cos\phi & -\sin\psi\sin\theta + \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \tag{3.27}$$

3.4.2. Motion estimation

Once the rotation is estimated, it is possible to estimate the motion displacement. The second step consists of integrating the acceleration using the body reference of time zero to estimate the position.

The accelerometer is a device that measure the combination of forces under it is subjected. Concretely, it measured the linear acceleration that is a linear combination of the acceleration associated to the movement and the gravity. The IMU is composed by three accelerometers aligned with the gyroscopes and they allow the estimation of the position through the integration of the movement acceleration [44].

The first step to estimate the position is being able to to bring all the accelerations obtained to one body-reference. The accelerations are multiplied by the rotation matrix obtained form the Euler angles also obtained at the time the acceleration has been sampled. In this way, it is possible to obtain the acceleration referenced at the initial time.

As the gravity is also measured in the accelerometer, it must be estimated and subtracted from the linear acceleration. Obtaining precise estimation of the acceleration is challenging. A first approach could be assume that at the beginning the IMU is not subjected to any force and subtract the offset. Another approach is to take the mean of a period where the IMU is not subjected to a great displacement and assume it as the gravity.

The position is estimated as the second integral of the clean acceleration. At this work it is proposed to use a bias estimator, concretely the Ridge estimator, to obtain the coefficients allowing to interpret the bias as an estimation of the gravity and taking advantage of the stability constraint it imposes to the estimation to mitigate the effect of the error in the gravity estimation. The coefficients are calculated by using a Ridge regression and it is added one standard deviation of the coefficients to them assuming that statistically hat part does not correspond to the gravity.

Chapter 4

Methodology and development

This chapter specifies the tools and the methodology used in this work. Section 4.1 explains the tools going from the software tools to the mathematical ones. Section 4.2 explains both databases used for the practical experiments. Section 4.3 shows in general terms the methods in a general overview. Section 4.4 gives the information related to the codes provided.

4.1. Tools

At an algorithmic level, the different tools used to create the estimation spline will be explained. The object of this section is to aid the reader in the interpretation of the different scripts created along the work and understanding the data nature of the different arrays present in middle steps.

4.1.1. MATLAB

MATLAB (MATrix LABoratory) consists on a multi-paradigm programming language widely used in numerical analysis. MATLAB is also a proprietary software that allows running MATLAB scripts. MATLAB is a weakly typed programming language that allows declaring variables without declaring their type, it also allows working with vectors and matrices. Mainly, two type of files extension are distinguished, '.m' extensions with script files and '.mat' extensions with data files. It exists other types of extensions connected with the environment that are not used at this work. As other programming languages MATLAB allow loops, where 'While' loops remain iterating until a condition become false and 'For' loops that iterates over a given vector.

Functions: These functions return outputs from given inputs. There are a great amount of established functions that improves the time of the work development at the time it helps avoiding errors. Function can also be declared by the developer, and it should be declared in a script with the same name of the function and located in the same folder it is going to be called.

Data structures: It consists on structures of different arrays. Each array in the structure is attached to a name and this array can be called by calling the data structure and its name connected by a point. These data structure can also be saved in a '.mat' extension file mentioned before.

Visualisation tools: MATLAB provides some visualisation tools that are useful to present the different results. The visualisation tools displays some array of results against an array that represents the domain. Among the different functions, it is highlighted plot that represents a vector y in the domain of a vector x and surf a three dimensional plot that represents a dimensional array against two bi-dimensional arrays x and y.

4.1.2. Spline MATLAB structure

Each spline in MATLAB consists on a data structure that describes an unique spline that can be evaluated and differentiated. There are some functions provided by MATLAB that allow creating spline. In this work it is used the function 'spmak' that creates the spline from the knots sequence and the coefficient matrix. This data structure is composed by six fields, three that define the spline and other three with information.

Form: It only accept 'B-' meaning that the spline us defined in basis form or 'pp' meaning that the spline is in piece-wise polynomial form. In this work, it is only worked with basis form which is also the way 'spmak' returns the spline.

Knots: The spline knots sequence, it is represented with the knots vector in each of the variables. It is the first argument in the 'spmak' function, in the case of univariate splines the vector is given and in the case of multivariate spline all the vector are between bracelets '' and separated by comas.

Coefficients matrix: The array corresponding to the coefficient matrix of the spline. It must be one dimension higher than the number of knots vector, keeping the first dimension to assemble the coefficients corresponding to each dimension of the spline and the rest of the dimension to each of the variables. It is the second argument in the spmak function.

Number: It is a vector with length of number of variables in the spline that gives the information of the number of coefficients that has each variable.

Order: It is a vector with length of number of variables in the spline that gives the information of the order of the spline at each variable.

Dim: It informs of the dimensionality of the spline.

4.1.3. Knots

The knots sequence is conditioned by the break sequence and the break sequence is also conditioned by the knots sequence. In the case of synthetic experiment, the breaks sequence is chosen first with equidistant breaks along the domain. The function 'linspace' performs this vector introducing as a first argument the initial point in the domain, the last point as a second argument and the number of breaks as a third argument. Then the knots sequence is chosen so there is a maximum smoothness in the internal breaks and no smoothness in the external one. The function 'augknt' allow creating this vector by introducing the sequence of breaks as a first argument and as a second argument the order of the spline.

The knots can also be obtained from a spline, generally previously created in a previous iteration, that it is assumed to be closed to the estimation spline. The function 'newknt' return a sequence of knots that makes the image of the last derivative of the spline given as a first argument equidistant. With an optional second number allows changing the number of knots. The function 'knt2brk' gives a breaks sequence corresponding to a knots sequence.

In synthetic experiments, the collocation points are calculated from the breaks sequence but in the case of practical experiments, where this point are constrained by a sampling, the breaks sequence is calculated from these points. The initial and the final points give the boundary of the work zone. The breaks are selected from a set of points composed by the midpoints between collocation points and an equal number of collocation points per B-spline are given.

4.1.4. Collocation points

The collocation points consists on a vector where the spline is constrained with the desired equation. If the collocation points can be selected, the roots of the Legendre polynomial are used. A own creation function called 'functiontau' is used to calculate the collocation points by introducing the vector of breaks as a first argument and the number of collocation points per spline as a second argument.

The function calculate the zeros of the Legendre Polynomial of order number of points per B-spline plus one. One the roots are calculated, they are scaled and centred at each of the intervals given by the breaks sequence. Finally, the initial and the final point of the work zone are added to the list.

4.1.5. Collocation matrix

The collocation matrix is mainly defined by a spline and a set of collocation points, it is the matrix that when multiplied by the coefficient matrix return the value of the spline, or its derivative at the collocation point. It is calculated by evaluating the natural spline, spline with one's coefficient matrix corresponding to a given set of knots and order, at the collocation points.

he function 'spcol' return a collocation matrix of the spline and its desired derivatives. So, each column correspond to a coefficient and each row correspond to a collocation point, being the first row the spline evaluation, ans the next rows the successive derivative chosen, and then, the next collocation point. The knots are introduced as a first argument, the order as the second one and the collocation points as a third one specifying the derivatives desired by repeating the collocation points for each derivative desired.

4.1.6. Other functions used:

System solver ('mldivide'): It is the function chosen to solve the coefficients. It is able to solve linear systems, in our case as it is overdetermined it provides the least squares solution.

Derivative spline ('fnder'): It is a function that perform the derivative of a spline, it is introduced the spline as a first argument and as a second argument it is introduced the vector with the order of the derivatives with respect each of the variables.

Point evaluation ('fnval'): It is the function that returns the evaluation of the spline at given points. As the function 'spcol' it used the De Cox-Boor algorithm with the difference that it uses the coefficient matrix of the spline. The spline is introduced as a first argument and the points where the spline is evaluated as the second argument.

4.2. Databases

Another family of tools used consists on the databases used in this work. Both experiment have been developed by using data from external databases. At this section, it is presented the PHYTMO database used in the motion monitoring experiment at Section 4.2.1 and the Liver database used in the image processing experiment at Section 4.2.2.

4.2.1. PHYTMO database

The PHYTMO database [8] is a database of physical therapies exercises used for obtaining the data for the motion monitoring exercises. It includes inertial measurement information and an optical reference systems. It includes the exercises performed by thirty volunteers including six exercises and three gait variations of each volunteer. The volunteers were aged between twenty and seventy years old and the age is specified with the height, the weight, the sex and the presence of any motor condition.

The acquisition was performed in the in the Motion Capture Laboratory of the University of Alcala. It was included the information of four IMUs, two situated at the two upper limbs and two situated at the two lower limbs. The IMUs used are NGIMU23 IMUs (X-io Technology, Bristol, UK), the information provided is presented at Table 4.1. The optical system is placed with the Y-axis pointing to the ceiling and the XZ-plane parallel to the floor plane. The optical system used is the OptiTrack24 system (NaturalPoint Inc), the information provided is presented at Table 4.2.

Table 4.1: Information provided by the inertial system.

Label	Content
Time (s)	It is the time assuming the zero the start of the recording.
Gyroscope (º/s)	It is the angular velocity at each of the coordinates.
Accelerometer (g)	It is the linear acceleration at which the IMU is subjected in each of the coordinates.
Magnetometer (μT)	It is the magnetic force at which the IMU is subjected in each of the coordinates.

Table 4.2: Information provided by the optical system.

Label	Content
Object position (m)	It is the Cartesian position of the IMU.
Object orientation	Quaternion orientation of the IMU in the 3D space.
Skeleton position (m)	It is the Cartesian position of the skeleton segment.
Skeleton orientation	Quaternion orientation of the skeleton in the 3D space.

Among the exercises, there are three legs exercises including 'Knee flexion-extension (KFE)', 'Squats (SQT)' and 'Hip abduction (HAA)'; there are also three are three arm exercises including 'Elbow flex-extension (EFE)', 'Extension of arms over head (EAH)' and 'Squeezing (SQZ)'. Among the gait variation it is found ' Gait (GAT)', 'Gait describing infinity symbol (GIS)' and ' Gait with heel-tiptoe (GHT)'. For this work it is chosen the three legs exercises of one participant.

Regarding the information obtained for performing the estimation, it is selected from the IMU the information of the time, for having the time domain, the gyroscope for estimating the orientation and, the acceleration for obtaining the position. From the ground truth, it is extracted from the optical system the object position and the orientation for contrasting their estimation.

4.2.2. Laparoscopic liver resection database

The Laparoscopic liver resection dataset [13] consists on a clinical dataset created for evaluating registration method to register preoperative data in laparoscopic liver registration and laparoscopic in general. The dataset consists on four patients with liver tumour. Those four patients has been undergone a pre-operative CT and a laparoscopic liver resection where it has been acquired a laparoscopic image and a laparoscopic ultrasound.

At the dataset it is included some script for the registration techniques and for evaluating the registration techniques. It is also presented the results of the registration techniques presented at the article. The main part of interest for the project is the dataset with the data used to achieve perform the registration. That data consists on four folders, corresponding to each of the patients and at each folder it is provided the following information presented at Table 4.3.

Table 4.3: Information provided in the dataset.

Label	Content
Lap Images	The images taken by the camera during the LLR in png extension and a file with the camera parameters.
Lap augmented by LUS	The laparoscopic image augmented with the laparoscopic ultrasound in png extension.
LUS images	The laparoscopic ultrasound images in jpg extension.
LUS segmentation	The human segmented tumour in the laparoscopic ultrasound in jpg format. An automated segmentation is also included but it does not report any information.
LUS calibration and POSE	The laparoscopic ultrasound parameters of calibration and position in mat extension.
Preoperative model	Preoperative model of the liver and the tumour generated from the CT. The vertex collection is achieved in ply format and the geometry including the triangulation in stl format.

At this work, it is used the tumour preoperative models of the four patients. It includes the set of vertex used to recreate the model and the geometry that can be used to generate the preoperative model mesh.

4.3. Methodology

This section is devoted to explain the general procedure of the construction of the spline. The development of the experiments, the error metrics obtaining and the singularities behind each experiment are expanded in Chapter 5.

The main events in the assembling of a spline are summarised in the flux diagram in Figure 4.1. It is deeply explained with the intermediate steps and expanded in Section 4.3.1 and Section 4.3.2.

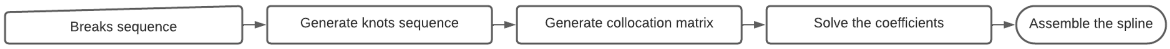


Figure 4.1: Critical events in the spline assembling.

4.3.1. Spline design

The first step in the process is creating a basis spline that cal fulfil the function properties. It is important to analyse the function to estimate and design the spline accordingly. The basis spline is defined by a sequence of knots, at the same time defined by a sequence of breaks. The first two things to take into account to create the sequence of breaks are the variate nature and the domain.

Variate nature: It defines the number of sequences of breaks, it is needed as many sequences of breaks as independent variable the function had.

Domain: It defines the extremes of the sequences of breaks, the extremes of the sequence of breaks must lie on the extremes of the function domain.

To end the breaks sequences, it is needed to define the internal breaks. Taking this decision is not trivial, it should be taken into account that it should be taken as many breaks as elasticity it is needed to transfer to the spline. In contrast, the number of breaks increase considerable the computational cost especially in multivariate functions. This is one of the decisions where the synthetic experiments could assist to design other experiments. It also must be taken into account when the function depend on external information to ensure that the spline is well conditioned.

Spline order: Prior to generate the sequence of knots, the spline-order should be selected. The order is other decision that it is explored along the work. Along the work, the most used basis spline is the sixth-order one. As general recommendation, it is recommended to use even-order splines to approximate the function and it is also desirable to use the lower-order possible but ensuring that the lower derivative to use is approximated at least with a fourth-order spline.

Sequence of knots: A sequence of knots with the maximum smoothness at the internal breaks and without smoothness at the external breaks is generated by using the Matlab function augknt explained at Section 4.1.3.

4.3.2. Coefficient system

Once the basis spline is designed, it is needed to obtain the coefficients that deform the spline to obtain the desired estimator. Then, it is necessary to select the collocation points, that is the points where the spline is constricted, according to the criteria explained at Section 4.1.4 if they can be selected and, if the function is constrained in the time domain, those points will be the collocation points.

In the case of multivariate functions, each of the points should be dependent on each of the domain variables. If the collocation points are not restricted, the tensor product of the collocation points of each breaks sequence is selected.

By using the collocation points, the collocation matrix from Section 4.1.5 is computed. In the case of multivariate spline, the collocation matrix correspond to the Cartesian's product of the collocation matrices of the spline at each variable. If the collocation points is selected as the tensor product of the collocation points, the tensor product of the collocation matrices is used. In the same way that the one variable collocation matrix provides the evaluation at the derivatives, the evaluation at the partial derivatives is obtained by performing introducing these derivatives at the product.

To illustrate with an example, let F be a three variable function and (x_p, y_p, z_p) the collocation point. The second first partial derivative respect to y and second respect to z can be computed in this way:

$$\frac{d^3F(x_p, y_p, z_p)}{dydz^2} = (F(x_p) \times \frac{dF(y_p)}{dy} \times \frac{d^2F(z_p)}{dz^2}) \tag{4.1}$$

If the differential equation is linear, then, the coefficients can be obtained, by solving the linear system composed by the function at all the constriction points and the other constrains introduced by the equation. Two things should be highlighted. Due to the linear nature of the coefficients, although the function depends on several derivatives the collocation matrix can be added as $ac + bc = (a + b)c$ and in the case of multidimensional splines the coefficients are merged in a same system to allow the multidimensional constrictions and solved together. By default, mldivide is used to solve the system although other way of solving can be considered depending on the application.

With respect to the non-linear differential equations, a previous step of linearization is needed. At this linearization, an additional linear viscosity term is added to the equation. This term is first-order higher than the differential equation order. Then, the Taylor expansion of this term with respect to the coefficients is developed until a linear polynomial.

$$\varepsilon d^{n+1}y = F(x, ..., y, ..., y^n) \approx F(c_k) + \frac{dF(c_k)}{dy}(c - c_k) + ... + \frac{dF(c_k)}{dy^n}(c - c_k) \tag{4.2}$$

Iterative procedure: In contrast to linear equations, it requires an iterative procedure. First, an initial guess is set, then, the coefficients are solved with respect to that initial guess. The next steps, the system is solved with respect to the previous step. Analysing the number of steps needed to solve an equation could be complex, convergence criteria consisting in takes error metrics using the next step as a ground truth could be illustrative of the solution obtaining. In general cases, around five iteration could be a good choice it it is preferred to avoid the analysing of the convergence. At each iteration, optimising the number of knots with respect to the previous spline with 'optknt' is recommended.

Viscosity: The role mainly depends on the nature of the equation. Since the non-linear nature of the equation, extraordinary elasticity demand should be taken into account. There is not a general rule, but discontinuities, other lack of smoothness and asymptotes are the main cause of these extraordinary demand. If there is not extraordinary elasticity demand, the viscosity could be set to zero and solve the coefficients in the iterative procedure. When there is an extraordinary demand of elasticity, viscosity that allows obtaining a result should be selected, the system is solved for that viscosity and it is used as a initial guess for a lower viscosity. The viscosity should be vanished until obtaining a desired solution but taking into account that it would not be possible to solve for a $\varepsilon = 0$. The function 'optknt' with an increase in the number of breaks is recommended when vanishing the viscosity to increase the elasticity of the spline.

4.3.3. Spline setup

In the case of these iterative procedures, or in the case it is desired to work with the estimation solution, it is necessary to assemble the spline. The function 'spmak' can assembled the spline in 'B-form' as explained in Section 4.1.2. Then, that Matlab structure representing an unique spline.

Before assembling the spline, the coefficients should be transformed from the vector form that is returned from solving the system to the dimension of the coefficients arrays corresponding to the Spline. It is important to understand the way Matlab handle this information to avoid errors from discordant criteria. It is considered the knots matrix as the tensor product of the knots in the order introduced. It had repercussions on the way product collocation matrices are computed and the way the coefficients are restored to the coefficient matrix.

Collocation matrices: When computing the tensor or the Cartesian's product of some collocation matrices the product order must be the same than the order the knots are introduced at 'spmak'.

Coefficient matrix rearrangement: At the rearrangement, the first step is to chop the concatenated coefficients corresponding to the different dimensions. Then, at the variate nature, the function ‘reshape’ can be used. It should be taken into account, that the first new dimension correspond to the expected dimension, number of knots minus order, of the variable which knots were introduced first. Then, the different dimensions are concatenated at the first dimension of the coefficient matrix.

Finally, as explained at Section 4.1.6, the function ‘fnder’ can be used to obtain the derivative spline and the function ‘fnval’ can be used to evaluate the spline at different points.

4.3.4. Error metrics:

Once the experiment is completed, it is important taking the error metrics that allow to to evaluate the experiment. Obtaining these error metrics is challenging, and the first thing needed is a ground truth. In the synthetic experiments, the analytical solutions will be used as a ground truth; in the motion monitoring experiment, the position given by the optical system will be used as a ground truth and; the image processing experiment will have the same pre-operative model as a ground truth as no other external sample is provided although as it is a fitting experiment it is enough to evaluate part of its performance.

Sampling of the domain: In both practical systems the points of error evaluation are constricted, in one case by the optical system sampling and in the other by the pre-operative models. In the synthetic experiments, a vector or a mesh is generated as a set of the points of error evaluation. In the univariate experiments, a vector which extremes lie on the domain extreme and with one thousand equidistant points is used. In the multivariate spline, a grid of vectors with fifty equidistant points in the domain of each variable used. In the case of bivariate splines, the ones used at this work, two thousand five hundreds points per splines are used.

Error at each point: The error at each point is taken as the absolute value of the subtraction of the estimation evaluated at that point and the analytical function at that point exposed in Equation (4.3). In some experiments, where the image is in values far from an intuitive range, the relative error is used exposed in Equation (4.4).

$$e(t) = |s(t) - f(t)| \tag{4.3}$$

$$e(t) = \frac{|s(t) - f(t)|}{|f(t)|} \tag{4.4}$$

Finally, to characterise the estimation, three global metrics are computed.

Mean error: The mean error estimated at Equation (4.5) allows to understand the error in which the estimation is moving.

$$\mu_e = \frac{\sum_{i=0}^{N-1} e(i)}{N} \tag{4.5}$$

Standard deviation error: The standard deviation error computed at Equation (4.6) allows a parameter of the dispersion of the error. It should be taken into account that for considering an estimation the sum of the mean error and the standard deviation error should be considered acceptable.

$$\sigma_e = \sqrt{\frac{\sum_{i=0}^{N-1}(e(i) - \mu_e)^2}{N}} \tag{4.6}$$

Maximum error: The maximum error at Equation (4.7) is the highest value among the errors computed. If the sampling criteria is enough it could give an idea of the quality of the estimation since it is sensible to higher errors in small localised regions of the domain.

$$max(e(i)) \tag{4.7}$$

4.4. Experiments dataset

At this Section, it is explained how it is structure the GitHub repository where it is updated the work performed by this work. There is an initial division in three folders corresponding to the three families of experiments realised at this work. One folder called 'Synthetic experiments', a second folder called 'Motion monitoring experiment' and a third folder called 'Image processing experiment'.

Table 4.4: Folders content of the repository.

Folder name	Content
Synthetic experiments	It contains all the experiments performed as synthetic experiments.
Motion monitoring experiment	It contain the three experiments that consist on the motion estimation of three different exercises.
Image processing experiment	The four experiments of the four preoperative models of the fourth patients.

At each of the folder intuitive names are given to the experiments following the criteria from Chapter 5. The synthetic experiments are called according to its properties

(_/N)L(1/M)D(1/M)V representing the first part if it is linear or non-linear, the second part if its unidimensional or multidimensional and, the third part if it is univariate or multivariate. If there is more than one experiment its numerated. The Gompertz experiment it is called by its name.

The motion monitoring experiment is called by the name of the exercises monitored 'KFE' for the Knee flexion-extension, 'SQT' for the squats and 'HAA' for the Hip Abduction. The Image processing experiments are called Tumour(1/2/3/4) according to the number of patient.

Each one of the folders is at the same time composed by four folders that structure each of the experiments:

Table 4.5: Folders content of the experiments.

Folder name	Content
Codes	It contains the codes where it is developed the different experiments of the work. It consists on Matlab codes that following the requirements of each experiments generate the approximating splines. The obtaining of the error metrics and results displaying are also included.
Results	The results are Matlab structures generated by each experiment codes where it is displayed the comparison with the ground truth. It is included the domain in which it is compared, the evaluation in the ground truth and in the approximation, the error along the domain and the error metrics computed. The files are 'mat' extensions.
B-splines	It consists on the Matlab structures, presented at the Section 4.1.2 and generated by the code. So, if it is desired to import the Spline it could be done and it is possible to work in Matlab with it. The files are 'mat' extensions.
Graphs	It is composed by the graphic content of each experiment that is also displayed at the essay. They are generated in the code and compressed in EPS format.

The codes can be found in the following GitHub repository: `https://github.com/RealARamirez/Bachelor_thesis.git` and the display of the results of each experiment can be consulted at Chapter 5.

Chapter 5

Results and Discussion

At this chapter, all the experiments including the results and its interpretation will be presented. Among the different characteristics that refers to the experiments, the first distinction is done respect to the ones restricted with sampled information. At Section 5.1 all the synthetic experiments, that correspond to the experiment without this restriction, are developed. The next two sections correspond to the two practical cases, that have this restrictions, corresponding Section 5.2 to the motion monitoring experiment and Section 5.3 correspond to the image processing experiment.

5.1. Synthetic experiments

This section will include all the synthetic experiments. The experiments will be divided in sections corresponding to the other criteria respected to the nature of the equation and the function. The first characteristic is the nature of the equation, the second the dimension of the function and, the third, the variate nature of the function. That is the criteria to subdivide the different sections. Although, the Gompertz model is a nonlinear unidimensional univariate equation, it is left separately at the end in the Section 5.1.7 so it is explained in detail also as an application.

5.1.1. Linear-unidimensional-univariate experiments:

At this section it is explored the most simple cases that consists on solving linear equations with unidimensional univariate functions. At this section is where most hyper-parameters are explored. It is divided in three different experiments.

Experiment 1: This experiment consists on a simple initial value problem of a first-order differential equation where it is evaluated the selection of collocation points. Both results are compared with the analytical solution.

The differential equation is presented on Equation (5.1):

$$\begin{cases} y - y' = 0; \\ y(0) = 1 \end{cases} \tag{5.1}$$

The solution of the differential equation of Equation (5.1) is set on Equation (5.2):

$$y(x) = e^x \tag{5.2}$$

With respect to the characteristics of the spline, it is a fourth-order spline with four breaks in the domain [0, 1].

This experiment shows the great difference caused in the approximation caused by the location of the collocation points. Figure 5.1 exhibits the performance of the exponential approximation by using the break sequence as collocation sites. It is visually assessed that the approximation is worse than the one in Figure 5.2 where the same approximation is estimated only changing the location of the collocation points to the Legendre zeros.

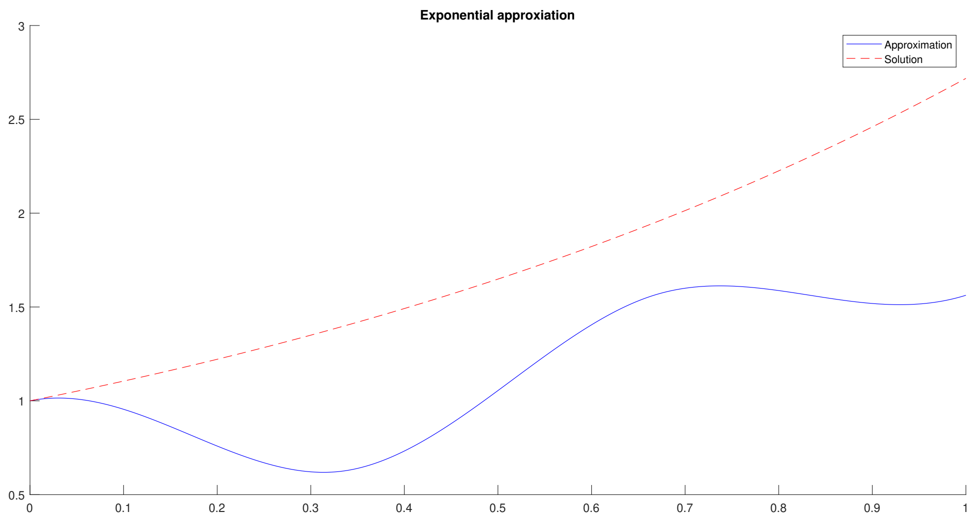


Figure 5.1: Exponential approximated using breaks as collocation points.

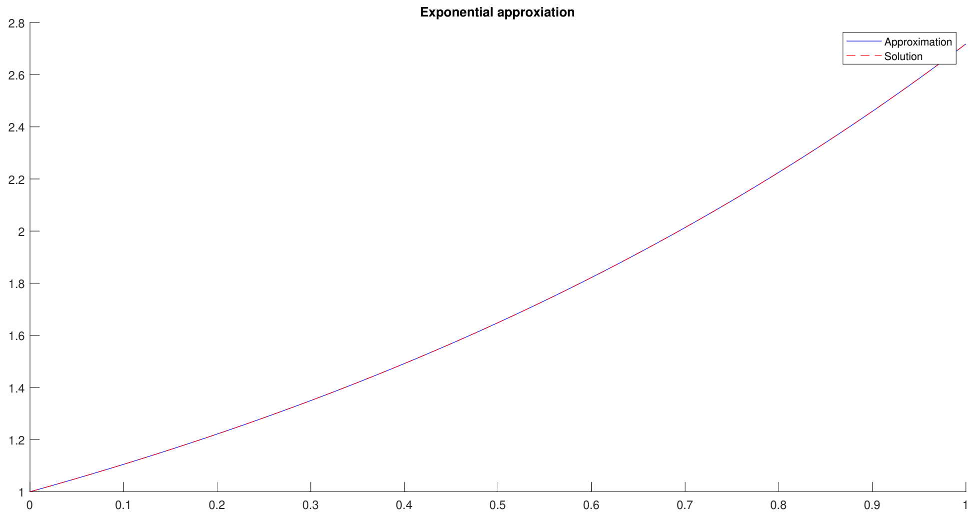


Figure 5.2: Exponential approximated using legendre zeros as collocation points.

The results from the graph are supported by the error metrics that are displayed at Table 5.1. The results states that at this estimation there is an improvement around 10^4 times consistent at all the metrics. It is caused by a better choice in the location of the collocation points.

Table 5.1: Error metrics with respect to the place of collocation points.

Collocation points	Mean error	Max error	Std error
Breaks	0.56	1.15	0.27
Legendre zeros	7.91e-05	1.86e-04	5.65e-05

Experiment 2: This experiment consists on an initial value problem of a second-order ordinary differential equation where it is evaluated the number of breaks and collocation points. The results are compared with the analytical solution and the best solution is exposed.

The differential equation is presented on Equation (5.3):

$$\begin{cases} y + y'' = 0; \\ y(0) = 0; \\ y'(0) = 1; \end{cases} \tag{5.3}$$

The solution of the differential equation of Equation (5.3) is set on Equation (5.4):

$$y(x) = \sin x \tag{5.4}$$

With respect to the characteristics, it is a sixth-order spline in the domain $[0, 2\pi]$ with one collocation point per interval.

The first part of the experiment is devoted to orientate the number of breaks needed to approximate a sine in a period. At Table 5.2, the sine approximation results are displayed. The results exhibits how the error metrics improve when increasing the number of breaks and it concludes that, in this case, five breaks per period is an acceptable choice.

Table 5.2: Error metrics with respect to the number of breaks.

Number of breaks	Mean error	Max error	Std error
3	0.35	0.89	0.29
4	0.19	0.56	0.16
5	0.01	0.03	0.01

Then, the characteristics of the spline are, it is a sixth-order spline with five breaks in the domain $[0, 2\pi]$.

The second part of the experiment finds the error related to the number of collocation points while it maintains the previously concluded five breaks per period. Table 5.3 provides the results concluding that the error decreases until three collocation points per interval and then there is not an improvement. In similar cases, four collocation points per interval is considered a good choice.

Table 5.3: Error metrics with respect to the number of collocation points.

Number of collocation points	Mean error	Max error	Std error
1	0.01	0.03	0.01
2	0.0028	0.0076	0.0023
3	0.0025	0.0071	0.0024
4	0.0028	0.0078	0.0027

Figure 5.3 portraits the sine estimation and the sine function over a period. The estimation uses the values indicated, five breaks per period and four collocation points per interval between two breaks.

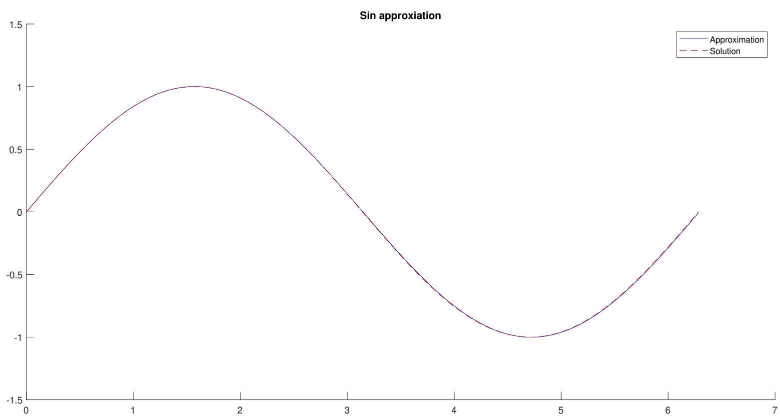


Figure 5.3: Sine approximated with five breaks per period and four collocation points per interval.

Experiment 3: This experiment consists on an boundary value problem of a second-order differential equation where it is evaluated the impact of the order. The problem is solved with splines with different order and the results are compared.

The differential equation is presented on Equation (5.5):

$$\begin{cases} y + y'' = 0; \\ y(0) = 1 \\ y(2\pi) = 1 \end{cases} \tag{5.5}$$

The solution of the differential equation of Equation (5.5) is set on Equation(5.6):

$$y(x) = \cos x \tag{5.6}$$

With respect to the characteristics, it is a spline with five breaks in the domain $[0, 2\pi]$ and with four collocation point per interval.

The Table 5.4 exhibits the error metrics in the fourth-order spline and how it increases when the derivative is higher. It appreciates that the results could be acceptable at the cosine approximation, bad at its first derivative and useless in the case of the second derivative. The results corroborates the hypothesis that third-order and specially second-order splines does not have the required smoothness for having acceptable performances.

Table 5.4: Error metrics with respect to the successive derivatives of a fourth-order spline.

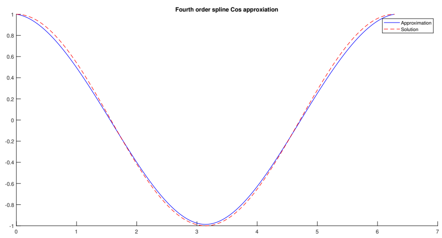
Spline derivative	Mean error	Max error	Std error
0	0.02	0.04	0.01
1	0.04	0.08	0.02
2	0.09	0.18	0.05

In Table 5.5 the criteria is switched so fourth-order spline is guaranteed at the second derivative by choosing a sixth-order spline for the cosine approximation. The results are much better than the ones displayed at Table 5.4 and both derivatives have acceptable performances. By default, at this work the other approximations use sixth-order splines but the main criteria is guaranteeing the last target derivative at least a fourth-order spline.

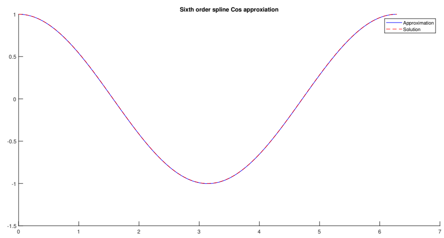
Table 5.5: Error metrics with respect to the successive derivatives of a sixth-order spline.

Spline derivative	Mean error	Max error	Std error
0	0.0014	0.0026	7.14e-04
1	0.0018	0.0032	0.06e-04
2	0.0053	0.0053	0.0027

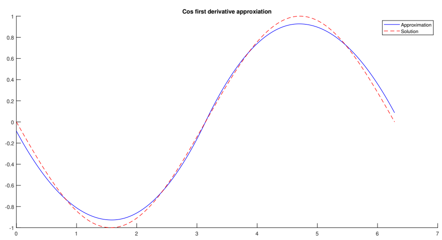
At Figure 5.4 the results can be appreciated in a more visual way. The left figures correspond to the fourth-order spline and the right to the sixth. The first row is the approximation, the second its first derivatives and the the third the second derivatives. It is seen, in concordance with the results, that the right figures approximate better the function, increasing the improvement as the derivative increases.



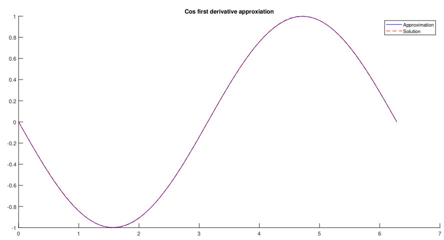
(a) Fourth-order spline cosine.



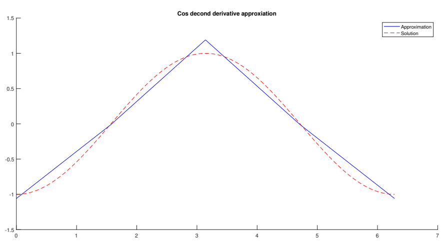
(b) Sixth-order spline cosine.



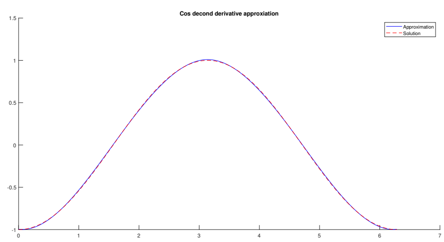
(c) First derivative of fourth-order spline.



(d) First derivative of sixth-order spline.



(e) Second derivative of fourth-order spline.



(f) Second derivative of sixth-order spline.

Figure 5.4: Sine splines and successive derivatives.

As a conclusion of these experiments. It is shown the feasibility to approximate linear differential equation with one dimensional one variable function. Also, at Section 5.1.1 it has been shown that choosing the Legendre zeros is a good choices for the collocation point location. At Section 5.1.1, it is shown that four points of collocation points per interval is a good choice and an association of how the number of breaks affect the error is also shown. At Section 5.1.1, it is shown that, selecting an even-order spline that left the last derivative to approximate in a fourth-order spline is a good choice to ensure that all the derivatives are well estimated.

5.1.2. Linear-unidimensional-multivariate experiments:

At this section it is explored the cases that consists on solving linear equations with unidimensional multivariate functions. At this section, three bivariate functions that comes from solving the 2D Poisson equation with different conditions and comparing with the analytical solution that were provided by other authors [45].

The characteristic of the splines are sixth-order with ten breaks and four collocation points per interval in the domain x:[0, 1] and others in y:[0, 1].

The general differential equation is presented on Equation (5.7):

$$\frac{d^2 z}{dx^2} + \frac{d^2 z}{dy^2} = f(x, y) \tag{5.7}$$

The specifications for the first experiment are presented on Equation (5.8):

$$\begin{cases} f(x, y) = 10(x-1)\cos(5y) - 25(x-1)(y-1)\sin(5y) \\ z(0, y) = (1-y)5y \\ z(1, y) = 0 \\ z(x, 0) = 0 \\ z(x, 1) = 0 \end{cases} \tag{5.8}$$

The solution for the Equation (5.8) is set on Equation (5.9):

$$z(x, y) = (1-x)(1-y)\sin(5y) \tag{5.9}$$

The specifications for the second experiment are presented on Equation (5.10):

$$\begin{cases} f(x,y) = f(x,y) = \sin x\pi \sin y\pi \\ z(0,y) = 0 \\ z(1,y) = 0 \\ z(x,0) = 0 \\ z(x,1) = 0 \end{cases} \tag{5.10}$$

The solution for the Equation (5.10) is set on Equation (5.11):

$$z(x,y) = \frac{\sin(x\pi)\sin(y\pi)}{2\pi^2} \tag{5.11}$$

The specifications for the third experiment are presented on Equation (5.12):

$$\begin{cases} f(x,y) = 0 \\ z(0,y) = 0 \\ \frac{dz}{dx}(1,y) = 0 \\ z(x,0) = 0 \\ z(x,1) = \sin(\frac{x\pi}{2}) \end{cases} \tag{5.12}$$

The solution for the Equation (5.12) is set on Equation (5.13):

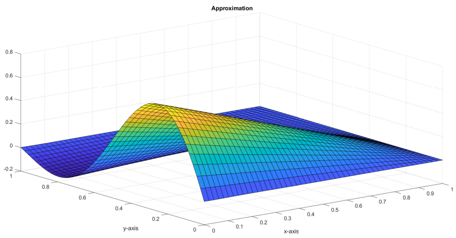
$$z(x,y) = \sinh(\frac{y\pi}{2})\frac{\sin(x\pi/2)\sinh(\pi/2)}{2\pi^2} \tag{5.13}$$

Table 5.6 exhibits the error metrics obtained from each of the experiments. In general terms, the three experiments had resulted in satisfactory estimations. They provide a reliable substitute of the analytical solutions.

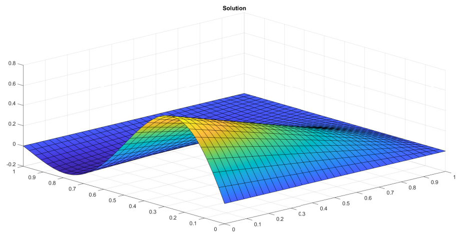
Table 5.6: Error metrics of the three experiments.

Experiment	Equation	Mean error	Max error	Std error
1	(5.8)	1.51e-04	0.0029	7.14e-04
2	(5.10)	1.16e-07	3.95e-07	6.14e-08
3	(5.12)	4.31e-08	9.68e-07	9.26e-08

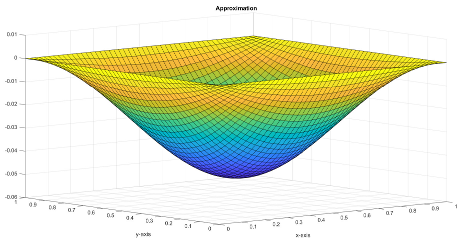
At Figure 5.5 the three approximation at the left and the three analytical solutions at the right are shown. Each of the rows correspond to one experiment, the first row to the first experiment, solving Equation (5.8), being Figure 5.5a the estimation provided of the first experiment and Figure 5.5b its analytical solution. In the same way, solving Equation (5.10), Figure 5.5c corresponds to the second experiment and Figure 5.5d. Finally, solving Equation (5.12), Figure 5.5e shows the third experiment and Figure 5.5f its analytical solution. The graphs corroborate the results obtained and they aid to visualise the bivariate approximation performance. At these experiments it has been shown the feasibility to use multivariate splines fir estimating multivariate functions.



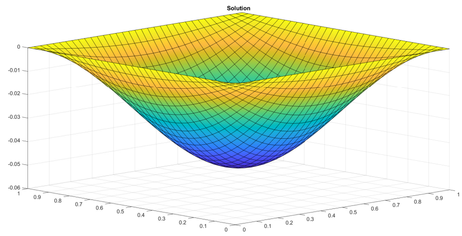
(a) Estimation of Equation (5.8).



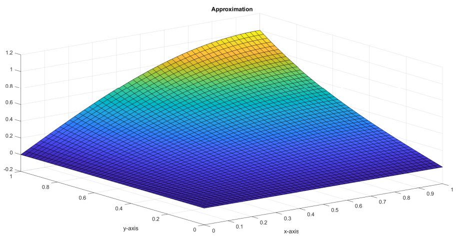
(b) Solution Equation (5.9).



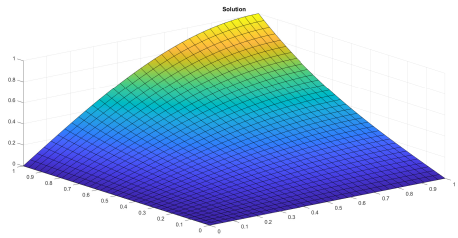
(c) Estimation of Equation (5.10).



(d) Solution Equation (5.11).



(e) Estimation of Equation (5.12).



(f) Solution Equation (5.13).

Figure 5.5: Laplace functions approximation and analytical solutions.

5.1.3. Linear-multidimensional-univariate experiments:

At this Section it is explored the cases that consists on solving linear systems equations that leads to multidimensional univariate functions. Concretely, it is solved a bidimensional function, the error at each dimensions are computed and it is also shown the result of representing one dimension in function of the other one.

The differential equation system is presented on System (5.14):

$$\begin{cases} x' = x + y \\ y' = 4x - 2y \\ x(0) = 2 \\ y(0) = -3 \end{cases} \tag{5.14}$$

With respect to the characteristics of the spline, it is a sixth-order spline with five breaks in the domain [0, 1] and four collocation points per interval.

The results displayed at Table 5.7 shows that the method provides coherent results with the ones obtained from single equations when approximating a system of equation. It concludes that the estimation is successful.

Table 5.7: Error metrics with respect to the spline dimensions.

Dimension	Mean error	Max error	Std error
1	1.69e-06	4.02e-06	1.11e-06
2	5.79e-06	1.37e-05	3.74e-06

At Figure 5.6 it is shown both the y with respect t and the x with respect t. It is compared with their analytical solution. It visually assesses the fact that both estimation have a good performance.

(a) Spline first dimension. (b) Spline second dimension.

Figure 5.6: Spline system independent solutions compared with analytical solutions.

In this cases it is frequent to demand the result as a representation of the dimensions of the spline. In this case, the spline compacted with $y(t)$ with respect $x(t)$ is represented at Figure 5.7.

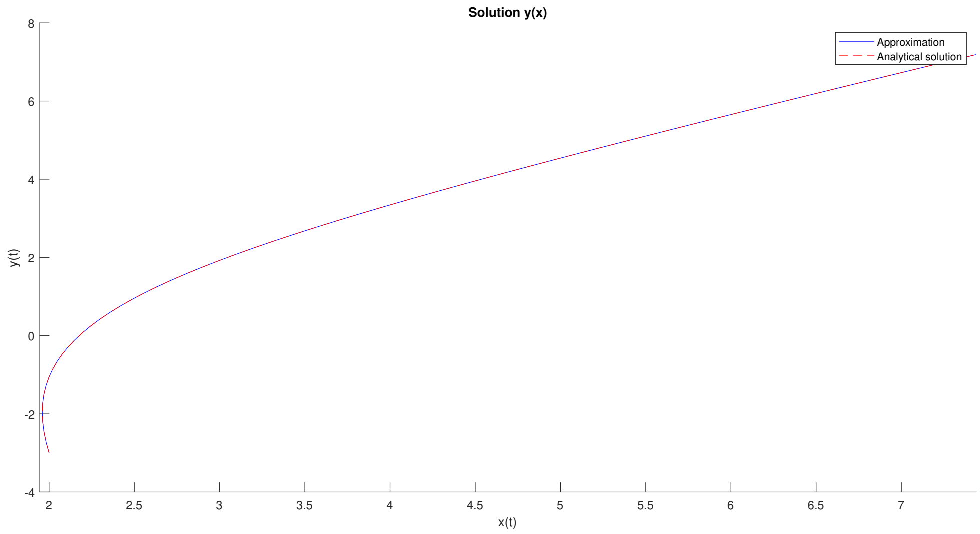


Figure 5.7: Spline compact system solution compared with the analytical solution.

5.1.4. Linear-multidimensional-multivariate experiments:

At this section, it is explored the case of multidimensional multivariate splines. In this case, it is not solved differential equation, it is solved some functions to corroborate the properties of the spline. At the section there are two experiments.

Experiment 1: The goal of this experiment is to create an spline that transforms cardinal coordinates into polar coordinates. It will be a bidimensional bivariate spline that it is obtained through solving with spline the equations that perform the transformation.

The equation for obtaining the radius is presented on Equation (5.15):

$$r = \sqrt{u^2 + v^2} \tag{5.15}$$

The equation for obtaining the angle is presented on Equation (5.16):

$$\omega = \arctan \frac{v}{u} \tag{5.16}$$

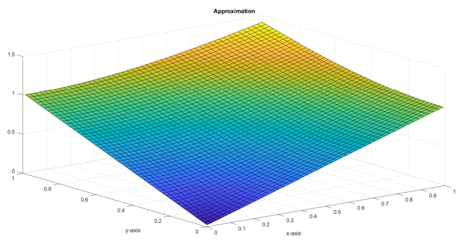
With respect to the characteristic of the splines, it is a sixth order spline with ten breaks and four collocations point per interval in the domain of x:[0, 1] and another in y:[0, 1].

Table 5.8 gives the error metrics for both of the bivariate dimensions. The results are good and coherent with the past results concluding that the approximation of multidimensional multivariate functions is feasible.

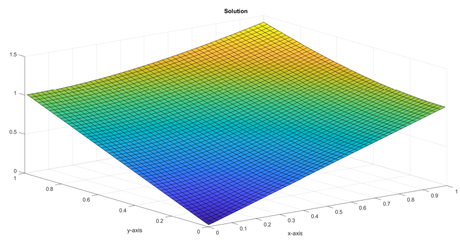
Table 5.8: Error metrics with respect to the spline dimensions.

Dimension	Mean error	Max error	Std error
1	2.26e-06	2.93e-04	1.13e-05
2	1.48e-04	0.02	0.0011

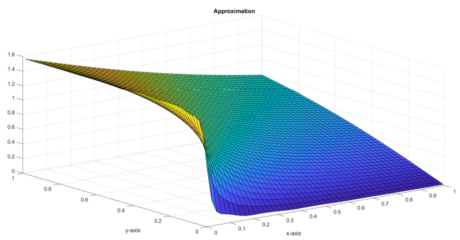
Figure 5.8 displays graphically the results. With the same criteria of Section 5.1.2 at the left column it is seen the approximation and at the right column the analytical solution. The first row represent the radius dimension being Figure 5.8a our approximation and the second row the angle dimension being Figure 5.8c our approximation.



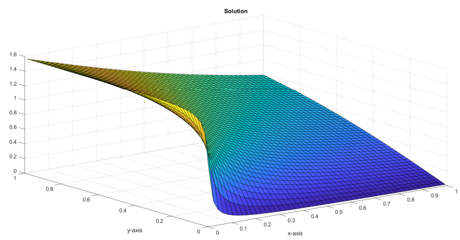
(a) First dimension spline approximation.



(b) First dimension solution.



(c) Second dimension spline approximation.



(d) First dimension solution.

Figure 5.8: Both spline dimensions with its solutions.

Experiment 2: At this experiment a plane is represented. The peculiarity of this experiment is that the domain of the function is also represented as an outcome of the spline. Although this application is not much explored at this work, it is interesting since allow working with the domain.

With respect to the characteristic of the splines, it is a sixth order spline with ten breaks and four collocations point per interval in the domain of x:[0, 1] and another in y:[0, 1].

At Table 5.9 it can be seen that the approximation error allows perfectly to express the domain as an outcome of the spline. The first two dimension represents the domain of the plane and the third dimension the outcome of the plane. At Figure 5.8 it is shown the result of the plane.

Table 5.9: Error metrics with respect to the spline dimensions.

Dimension	Mean error	Max error	Std error
1	1.94e-16	1.66e-15	2.16e-16
2	1.82e-16	1.55e-15	1.97e-16
3	1.91e-16	1.88e-15	2.29e-16

At Figure 5.9 the plane is displayed. It can be seen that the plane result is the equivalent than the plane that could be obtained by generating the grids as it had been generated at the previous bivariate experiments.

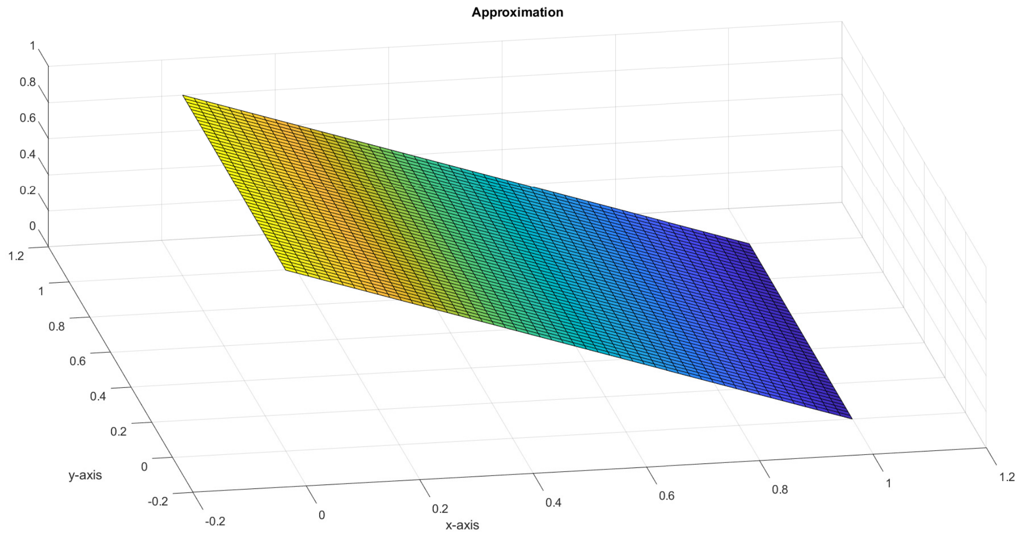


Figure 5.9: Plane represented compactly as a spline.

In this section, it has been provided the feasibility of multidimensional spline, and how it can be worked with them. Although, it has not been used dimensional equation, the collocation method for solving will be the same, but it would probable have error metrics more similar to Section 5.1.3 since this error metrics are closer to the approximation of functions not constricted by a differential equation.

5.1.5. Non-linear-unidimensional-univariate experiments:

At this section, it is started with the most simple case regarding non-linear differential equation solving. As it has been previously explained, this solving is more complex since it requires a procedure to make the function linear, an initial guess and an iterative process. Also, in some cases it requires another iterative procedure to vanish in case it cannot be set to zero. Two experiments are developed, one to show a case where the viscosity can be set to zero and other where it cannot be. Also, the second experiment is used to show the potential of knots redistribution that was not used previously.

Experiment 1: This experiment consists on a non-linear differential equation in which a unidimensional univariable function is approximated. It comes from a boundary problem. At this experiment, the viscosity is directly set to zero.

The differential equation is presented on Equation (5.17):

$$\begin{cases} (y')^2 + y^2 = \frac{50}{(1+\sin 2x)^2} \\ y(0) = 5 \\ y(\pi/2) = 5 \end{cases} \tag{5.17}$$

The introduction of the viscosity term is presented on Equation (5.18):

$$\varepsilon y'' = \frac{50}{(1+\sin 2x)^2} - (y')^2 - y^2 \tag{5.18}$$

Linearization with respect is presented on Equation (5.19):

$$\varepsilon y'' + 2y_k' y' + 2y_k y = \frac{50}{(1+\sin 2x)^2} + (y_k')^2 + y_k^2 \tag{5.19}$$

The analytical solution is set on Equation (5.20):

$$y(x) = 5\sqrt{\frac{1}{1+\sin 2x}} \tag{5.20}$$

With respect to the characteristics of the spline, it is a sixth-order spline with five breaks in the domain of x:[0, pi/2] and four collocation points per interval.

Table 5.10 shows the results of the non-linear differential equation. It expresses the concordance of the result for the non-linear differential equation with the linear differential equation. It demonstrates that a function without an special necessity of elasticity does not require the introduction of a viscosity.

Table 5.10: Error metrics of the spline.

Spline	Mean error	Max error	Std error
1	1.47e-04	4.87e-04	1.44e-04

At Figure 5.10 the approximation and the analytical solution are represented. It provides the same results than the error metrics, obtaining a successful approximation of the function.

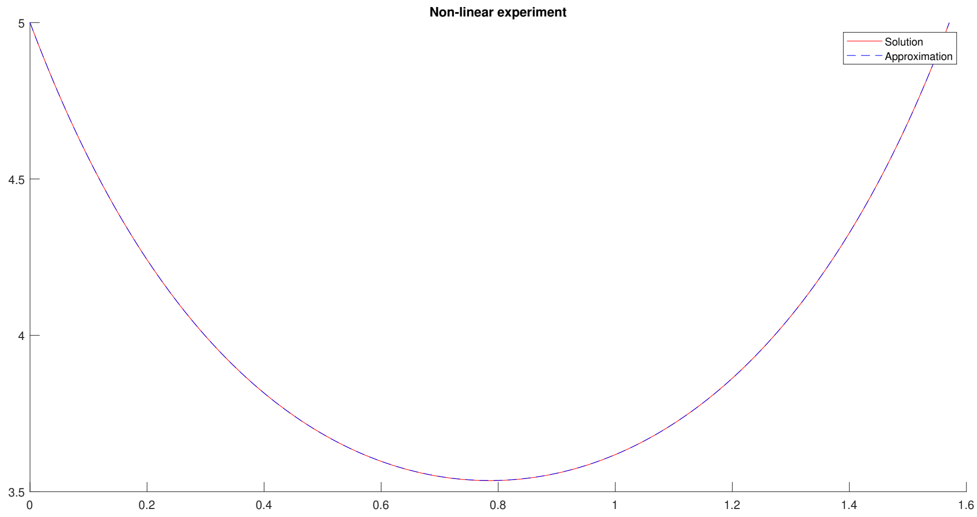


Figure 5.10: Solution to the non-linear differential equation.

Experiment 2 (Eikonal): This experiment consists on other non-linear differential equation which comes from a boundary problem. The Eikonal solved consists on a unidimensional univariable function. At this experiment, the elasticity demand of the problem is high due to the fact that it has a point that is not differentiable at its domain. It is shown the different option regarding the role of viscosity.

The differential equation is presented on Equation (5.21):

$$\begin{cases} (y')^2 = 1 \\ y(-1) = 0 \\ y(1) = 0 \end{cases} \tag{5.21}$$

The introduction of the viscosity term is presented on Equation (5.22):

$$\varepsilon y'' = (y')^2 - 1 \tag{5.22}$$

The linearization with respect to the viscosity term is presented on Equation (5.23):

$$2y_k' y' - \varepsilon y'' = 1 \tag{5.23}$$

The analytical solution is set on Equation (5.24):

$$y(x) = 1 - |x| \tag{5.24}$$

With respect to the characteristics of the splines, it is a sixth-order spline with five breaks and four collocation points per interval in the domain x:[-1, 1].

Set $\varepsilon = 0$: At a first approach the viscosity is set to zero.

At Table 5.11 is perceived that an initial viscosity of zero in a function with a high demand of elasticity and without a good initial guess is a bad choice. The results provided compares the estimation with the solution showing the estimation is useless.

Table 5.11: Error metrics of the first guess.

Viscosity	Mean error	Max error	Std error
0	0.49	1.04	0.31

The Figure 5.11 provides the initial guess and the first trial of the estimation. It shows the way the estimation does not get closer to the solution.

(a) Initial guess. (b) Spline approximation epsilon=0.

Figure 5.11: Spline initial guess and approximation.

Set $\varepsilon = 0{,}25$: At a second approach with the same initial guess.

The analytical solution is set on Equation (5.25):

$$y(x) = 0{,}25(\log(\cosh(\frac{1}{0{,}25})) - \log(\cosh(\frac{x}{0{,}25}))) \tag{5.25}$$

At Table 5.12 the results the comes from comparing the approximation with the solution provided for that viscosity are shown.

Table 5.12: Error metrics of a first viscosity introduction.

Viscosity	Mean error	Max error	Std error
0.25	0.001	0.004	0.001

At Figure 5.12 it is shown graphically that introducing a Viscosity makes possible to have an approximation of the problem that was not possible with the same spline characteristics and initial guess.

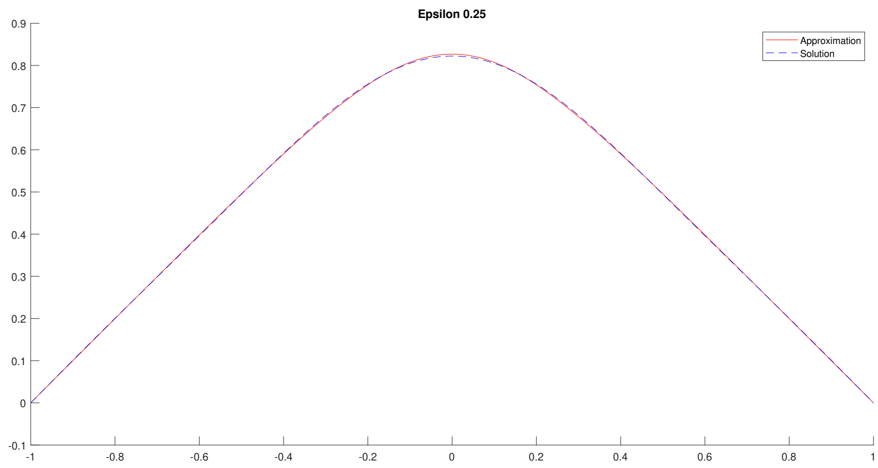


Figure 5.12: Spline approximation epsilon=0.25.

Set $\varepsilon = 0{,}125$**:** With the exposed spline, the previous viscosity is lower the one that allows convergence with the Initial guess. Now the Initial guess is changed to the previous solution. A second approach to increase elasticity by using knots distribution is given.

The analytical solution is set on Equation (5.26):

$$y(x) = 0{,}125(\log(\cosh(\frac{1}{0{,}125})) - \log(\cosh(\frac{x}{0{,}125}))) \tag{5.26}$$

At Table 5.13 the results the comes from comparing both approximation with the solution provided for that viscosity are shown. The first approximation uses as initial guess the result from the previous section and the second this first approximation. The improvement from the knots redistribution is significant.

Table 5.13: Error metrics with a viscosity of 0.125.

Viscosity	Knots redistribution	Mean error	Max error	Std error
0.125	No	0.006	0.02	0.006
0.125	Yes	0.001	0.006	0.001

At Figure 5.13 it is shown graphically that introducing a better initial guess makes possible to have an approximation of the problem with a lower viscosity. Then, it is shown the potential of the knots redistribution to improve the elasticity of the basis.

(a) Spline without knots redistribution. (b) Spline with knots redistribution.

Figure 5.13: Spline approximation with epsilon=0.125.

Set $\varepsilon = 0$**:** Using the previous initial guess and the knots redistribution.

Table 5.14 the results the estimation performed without viscosity but with the initial guess obtained from the previous step. It is a more acceptable results than the first one realised without viscosity.

Table 5.14: Error metrics without viscosity and a good initial guess.

Viscosity	Mean error	Max error	Std error
0	0.004	0.06	0.009

At Figure 5.14 it is shown how when using an initial guess from a viscous solution while solving the equation without viscosity it tends to converge to the given viscous solution.

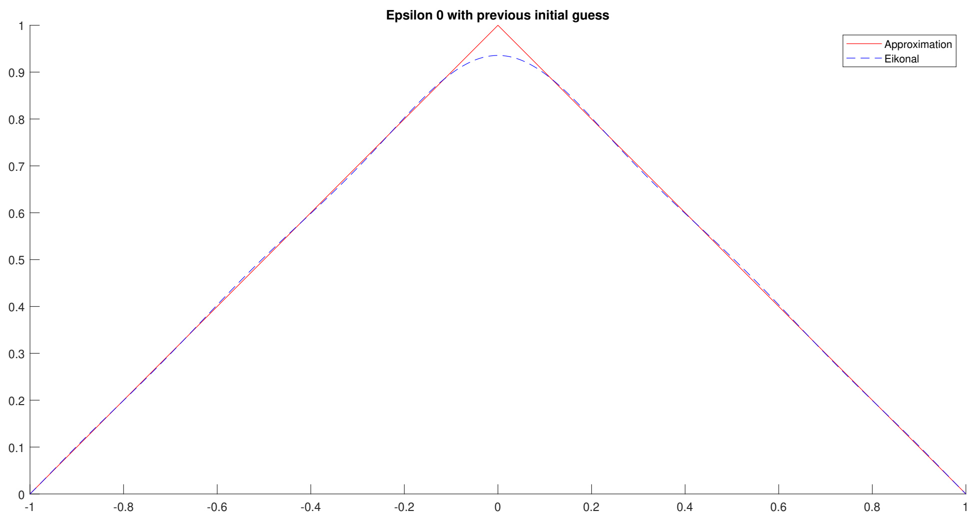


Figure 5.14: Spline approximation of the Eikonal with the best initial guess.

At this Section 5.1.5, it is developed how to proceed in the case of non-linear functions. First, introduce a linear viscosity term and linearize the function respect to that term. Second to analyse, the elasticity demand of the function if it is assumable solve iterative for a zero viscosity. If not, find a viscosity that allow convergence, and vanish the viscosity using previous solution as a initial guess. If, a better solution is needed, introduce more elasticity through increasing the number of breaks and redistributing knots to be able to keep decreasing the viscosity until the solution fulfil the expectations.

5.1.6. Non-linear-unidimensional-multivariate experiments:

A multivariate function that comes from a non-linear differential equation is solved. It consists on an initial value problem called the inviscous Burger equation, also it is evaluated one of its partial derivatives. The viscosity id directly set to zero and there is no knots redistribution. The partial derivative explored is the same evaluated during the iterative process.

The differential equation (Burger equation) is presented on Equation 5.27.

$$\frac{dy}{dt} + y\frac{dy}{dx} = 0 \tag{5.27}$$

The initial condition is presented on Equation (5.28):

$$u(0, x) = x \tag{5.28}$$

The viscosity term is introduced on Equation (5.29):

$$\varepsilon\frac{d^4y}{dt^2dx^2} = -\frac{dy}{dt} - y\frac{dy}{dx} \tag{5.29}$$

The linearization is set on Equation (5.30):

$$\varepsilon\frac{d^4y}{dt^2dx^2} + \frac{dy}{dt} + y_k\frac{dy}{dx} + \frac{dy_k}{dx}y = y_k\frac{dy_k}{dx} \tag{5.30}$$

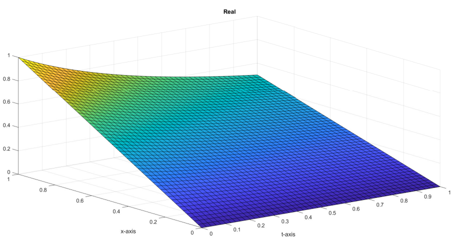
With respect to the characteristics of the spline, it is a sixth-order spline with five breaks and four collocation points per interval at the domain t:[0,1] and another at x:[-1, 1].

It is used an initial guess that expand the initial condition over t and five iterations for obtaining the solution. At this experiment, it is not only evaluated the solution, it is also evaluated the partial derivative respect to x that it is needed at the approximation. The approximation is successfully achieved obtaining the results of Table 5.15 that can be visualised at Figure 5.15.

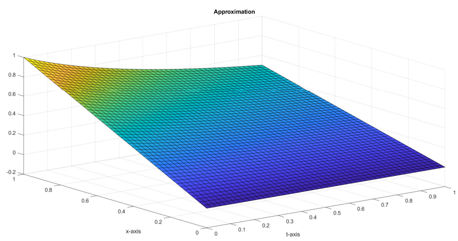
Table 5.15: Error metrics of the function and one partial derivative.

Derivative respect to t	Derivative respect to x	Mean error	Max error	Std error
0	0	2.77e-07	1.38e-06	2.49e-07
0	1	8.67e-07	7.80e-06	8.67e-07

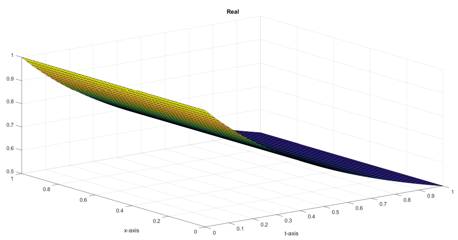
Figure 5.10 shows in the first row the function approximation in Figure 5.15a with its analytical solution in Figure 5.15b. Then, it shows that the sixth-order spline returns a good partial derivative respect to x in Figure 5.15c with its analytical solution in Figure 5.15d. This experiment shows the feasibility of scaling in variate nature a non-linear differential equation.



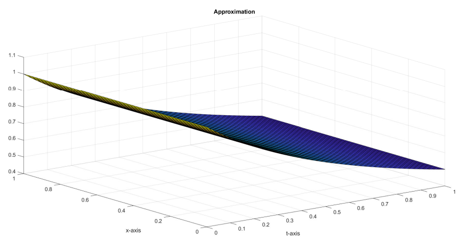
(a) Burger equation spline approximation.



(b) Burger equation solution.



(c) First partial derivative respect to x approximation.



(d) First partial derivative respect to x solution.

Figure 5.15: Burger equation.

5.1.7. Gompertz model:

Finishing with these synthetic applications, a more realistic synthetic application is developed. Tumour consists on a mass provoked by the uncontrolled proliferation of cells. At the time the tumour exceed a given volume, it needs to grow new vessels in-order to feed and oxygenate itself. This formation process it is called angiogenesis and if the vascularization is not enough cells can enter in degeneration and necrosis. This process leads to a diminishing growth of the tumour as the mass increases and it gets closer to the carrying capacity. This capacity can be defined as the maximum size the tumour can achieve with the available nutrients. In a more practical scenario, the mass tumour and the carrying capacity will be estimated with an imaging technique and the intrinsic growth will also be considered non-linear.

It is going to be estimated the number of tumour cells when it is at angiogenesis stage. The model consists on a non-linear differential equation which function has the time as domain and it returns the number of tumour cells. The model depend on three main parameters that is the initial number of cells, the carrying capacity and the growth rate. At this approximation, some values from the literature has been used but in a real scenario it will have particular parameters.

Initial number of cells: It is the number of cells measured at the tumour. It will correspond to the number of cells that will be assigned at time zero, at the beginning of the estimation. In real scenario, it is obtained from the tumour and at this estimation a reliable initial size where a Tumour has started angiogenesis is used.

Intrinsic growth rate: It is a linear constant multiplying the growth equation in the adaptation model of the Gompertz equation used [46]. It defines the growth of the tumour with respect to the time and at this experiment it is used the one provided at [47].

Carrying capacity: The carrying capacity is the maximum number of cells a tumour can maintain with the available nutrients and other conditions. When the carrying capacity is achieved, the growth in the model becomes zero. As the initial number of cells, it should be obtained by evaluating each case and at the example given it is used a general value from the literature [48].

For simplification, a constant intrinsic growth and a frequent carrying capacity will be extracted from the literature and a synthetic initial mass will be used. Concretely, an intrinsic growth of $r = 0{,}006$ and a carrying capacity of $k = 10^{13}$ cells [48].

The differential equation Model [46] is presented on Equation (5.31).

$$N'(t) = rN \log\left(\frac{C}{N}\right) \tag{5.31}$$

The linearization is introduced at Equation (5.32):

$$\varepsilon N'' = N' - r \cdot (\log \frac{C}{N_k} - 1) \tag{5.32}$$

The parameters of the model are set at Table 5.16.

Table 5.16: Parameters used at the Gompertz model.

Viscosity	Carrying capacity: C	Intrinsic growth rate: r	Initial n of cells: N
0	1e13(n of cells)	0.006(cells/t)	1e9(n of cells)

The equation model will be approximated using a spline with ten breaks, coming from an initial spline with equidistant breaks, but redistributing them in each iteration. At the non-linear procedure five iterations will be used at for performing the estimation. As the model has an analytical solution it will be taken error metrics. It will be taken absolute error metrics and relative ones since the last ones will be more informative due to the scale variability.

The analytical solution is set at Equation (5.33):

$$N(t) = C \cdot \exp\left\{\log \frac{N_0}{C} \cdot \exp(-r \cdot t)\right\} \tag{5.33}$$

Table 5.17 displays the error metrics. Due to the range nature of the function, the error metrics are also taken with respect to the relative error. It shows an acceptable performance of the estimation with a mean error similar to the comparable experiments.

Table 5.17: Error metrics of the Gompertz model.

Type of error	Mean error	Max error	Std error
Absolute	2.73e10	1.02e10	0.92e10
Relative	0.01	0.04	0.04

Figure 5.16 shows the approximation to the Gompertz model with its analytical solution. We observe that the estimation is pretty accurate and at the centre of the model is perceived the higher demand of elasticity that justify the improvements by the redistribution of knots.

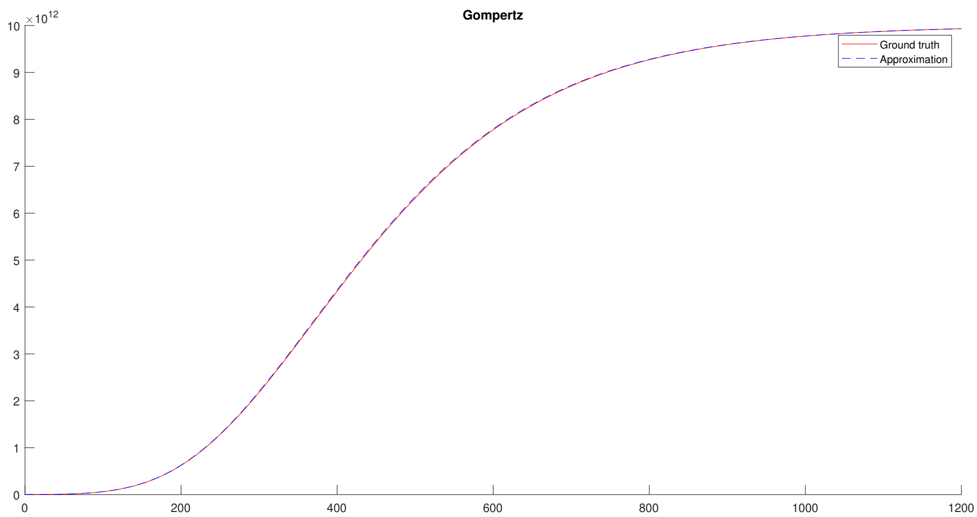


Figure 5.16: Gompertz model.

5.2. Motion monitoring experiment

The motion monitoring experiment tries to estimate the angles and displacements in three rehabilitation exercises. We use information from an IMU that provides the linear acceleration and the angular velocity of the motions. These experiments comes from leg rehabilitation exercises and the information provided by the optical system is used as a reference.

The estimation of the orientation is done through the integration of the angular velocity provided by the gyroscope that are aligned in the axis of the accelerometer projected into the fixed frame. The linear displacement estimation is obtained from integrating the linear acceleration also projected into the global fixed frame. The orientation of the device is obtained through the gyroscope measurement integration. Then, a first multidimensional spline where the orientation is dependent on the time is estimated. Then, with these results a second multidimensional spline is assembled with the motion depending on the time.

For solving the rotation the Euler angles differential systems is used to map the angular velocities in the IMU frame with the rotation angles. The mapping consists on a non-linear differential equation systems presented in Equation (5.34) that is solved with sixth order splines. The system is linealized following the same criteria than in the synthetic experiments, the viscosity and the initial condition are set to zero.

$$\overrightarrow{\omega} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\psi & \sin\psi\cos\theta \\ 0 & -\sin\psi & \cos\psi\cos\theta \end{bmatrix} \begin{pmatrix} \psi' \\ \theta' \\ \phi' \end{pmatrix} \tag{5.34}$$

Then the acceleration is multiplied by the rotation matrix feed with the obtained angles and twice integrated. At Equation (5.35) is presented the equation corresponding to the x axis, at Equation (5.36) with respect to the y axis and at Equation (5.37) with respect to the z axis.

$$a_x(t) \cdot M(\psi, \theta, \phi) - \frac{d^2x(t)}{dt^2} = 0 \tag{5.35}$$

$$a_y(t) \cdot M(\psi, \theta, \phi) - \frac{d^2y(t)}{dt^2} = 0 \tag{5.36}$$

$$a_z(t) \cdot M(\psi, \theta, \phi) - \frac{d^2z(t)}{dt^2} = 0 \tag{5.37}$$

The initial condition is assumed to be a initial position and a linear velocity of zero. Since it is a noisy condition, we use a Ridge regression or to estimate the coefficients and correct the estimation by adding an standard deviation of the coefficients in the opposite direction of the Ridge bias.

Each of the following sections correspond to one leg exercises perform by the volunteer. Section 5.2.1 provides the error metrics and the graphical support in the estimation of the KFE kinematics. Sections 5.2.2 and 5.2.3 includes the graphical results of the SQT and HAA exercises are provided.

5.2.1. Knee flexion-extension (KFE)

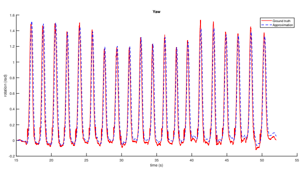
The KFE is an exercise where the patient, starting in a position with a position of 90º of knee flexion, is required to perform an extension exercise to increases the degree as much as possible and then a flexion exercise to return to the position. The exercises will be estimated using the information from the IMU and compared with the one provided by the optical reference system by taking the error metrics of mean error, maximum error and standard deviation of the error at Tables 5.18 and 5.19.

Table 5.18 shows the error estimation at the KFE Euler angles. The errors are expressed in radians and the metrics are coherent with mean and standard deviation error around 0.1 rad and maximum error around 0.5 rad. These errors are acceptable for the orientation angle monitoring with IMUs. The results are displayed at Figure 5.17 with the reference from the optical system.

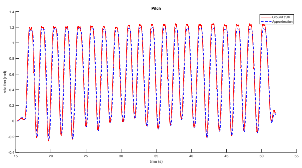
Table 5.18: Error metrics of KFE orientation.

Angle (rad)	Mean error	Max error	Std error
Yaw (ψ)	0.12	0.57	0.12
Pitch (θ)	0.11	0.30	0.07
Roll (ϕ)	0.16	0.55	0.12

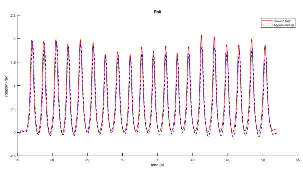
Figure 5.17 shows the estimation and the angles from the optical reference system. It corroborates that in this exercise the estimation of the three angles is satisfactory. It contextualises the maximum errors, nor as big errors in the estimation but smalls delays in parts of the estimation where there is fast changes.



(a) KFE yaw approximation.



(b) KFE pitch approximation.



(c) KFE roll approximation.

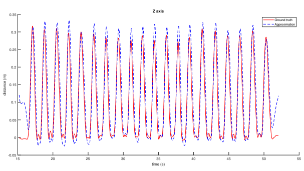
Figure 5.17: Exercise KFE angles of rotation with estimation (blue) and reference (red)..

Table 5.19 shows the error estimation at the KFE linear displacement. The errors are expressed in metres and the metrics are also coherent with mean and standard deviation error around 0.03 m and maximum error around 0.1 m. They are displayed at Figure 5.17.

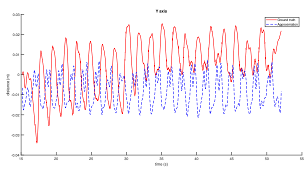
Table 5.19: Error metrics of KFE linear displacement.

Axis (m)	Mean error	Max error	Std error
Z	0.03	0.12	0.02
Y	0.01	0.04	0.01
X	0.05	0.10	0.02

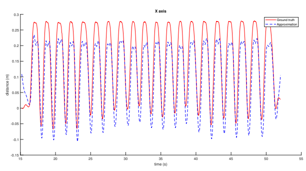
Figure 5.18 also shows the estimation of the linear displacement and the reference linear displacement. The linear displacement is more challenging since it is more affected by noise including errors in the estimation of the gravity and other forces acting on the IMU. It shows that the estimation of the Z axis at Figure 5.18a is satisfactory, the the estimation of the Y axis at Figure 5.18b is useless and the estimation of the X axis at Figure 5.18c is acceptable. The reason behind not being able ti distinguish at the error metrics is that at this exercise the error is bounded at a range and the quality of the estimation depends on how bigger is the exercise range of movement with respect to this range. It is also perceived that the bigger errors are performed at the beginning and at the ending when the patient is not performing the exercise.



(a) KFE Z axis approximation.



(b) KFE Y axis approximation.



(c) KFE X axis approximation.

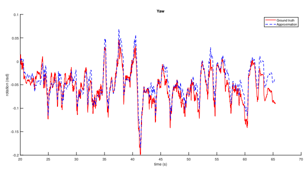
Figure 5.18: Exercise KFE linear displacement with estimations (blue) and references (red).

At the results provided in Tables 5.18 and 5.19 and shown the Figure 5.17 is concluded that the algorithm is successful for approximating the orientation and for the estimation of the movement at the Z and X axis. It fails to approximate the motion at the Y axis where there is no movement at the exercise and the range is much smaller.

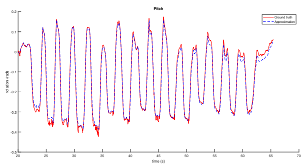
5.2.2. Squats (SQT)

The SQT is an exercises where the patient, starting in a standing position, is required to perform an flexion of the knee and hip joints to descend and then an extension of the these joints to return to the standing position. The exercises are estimated using the information from the IMU and compared with the one provided by the optical reference system.

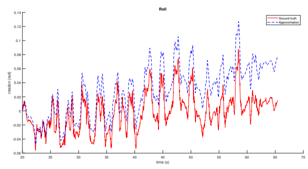
Figure 5.19 shows the graphical results for the orientation and the linear displacement in the SQT exercise. The first row is composed by the angle estimation that is close to the results obtained in Section 5.2.1 with the exception of the roll angle at Figure 5.19c where an small drift at the middle of the time leads to some error. Although, there is some error that is aggregated from that point it still keeps the shape. With respect to the linear displacement, only an acceptable result in the Z axis at Figure 5.19d. It is also coherent with the previous results since this exercise is an smaller range and the fact that there should be error accumulation from the drift error at the orientation.



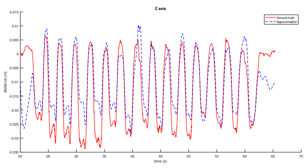
(a) SQT yaw approximation.



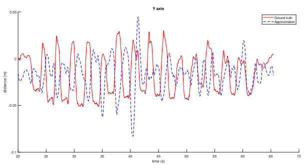
(b) SQT pitch approximation.



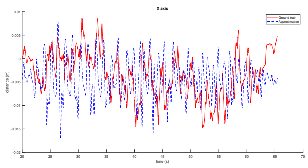
(c) SQT roll approximation.



(d) SQT Z axis approximation.



(e) SQT Y axis approximation.



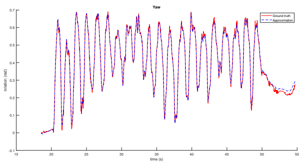
(f) SQT X axis approximation.

Figure 5.19: Exercise SQT orientation and motion with estimations (blue) and references (red).

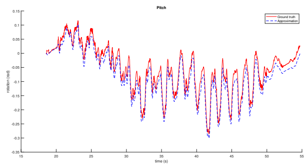
5.2.3. Hip abduction (HAA)

The HAA is an exercises where the patient, starting in a standing position, is required to perform an movement of the leg away from the mid-line of the body and then returning to the initial position. The exercises will be estimated using the information from the IMU and compared with the one provided by the optical reference system.

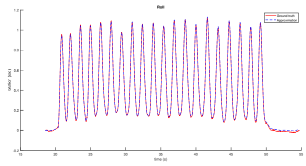
Figure 5.20 presents the information of the HAA exercises in a similar way. The conclusions are similar to the one obtained in the Squats exercise. At this exercise, the estimation of the orientation is satisfactory for the three angles. In the case of the axes it is not archived a good estimation. At the Z axis at Figure 5.20d is perceived that although it follows the movement it shows additional displacements that does not occur. At the other ones the estimation is completely incoherent showing that additional approaches to mitigate noise should be tried.



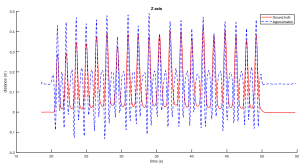
(a) HAA yaw approximation.



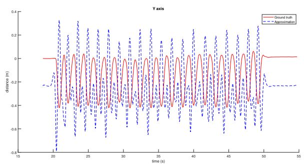
(b) HAA pitch approximation.



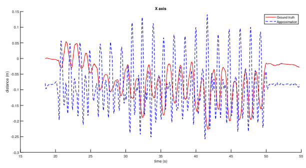
(c) HAA roll approximation.



(d) HAA Z axis approximation.



(e) HAA Y axis approximation.



(f) HAA X axis approximation.

Figure 5.20: Exercise HAA orientation and motion with estimations (blue) and references (red).

5.3. Image processing experiment

At this experiment, the pre-operative model of the tumours prepared for the Laparoscopic Liver Resection are represented as splines. Since at the database there are four patients, four section with the results of each of the approximations are developed. Each of the section corresponds to each of the patients.

The tumour are represented as bivariate functions that are approximated with a bivariate spline. To perform this approximation, we need to find a way to obtain a subset of the domain of the tumour without overlapping with respect to the other two. That is the reason why the tumours are centred at the origin of coordinates and then mapped into spherical coordinates. These two operations is undone for comparing the representation. It is important to highlight that the approach chosen restricts the pre-operative model to approximate to the ones which topology allows this assumption.

The model is assembled in Equation (5.38):

$$f(\alpha, \gamma) = r \tag{5.38}$$

The spline represents a bivariate equation that map the angles azimuth and elevation with the radius. Sixth order splines are selected so there is a good quality until the second derivative approximation. The intervals are selected in an univariate way with equals number of points. It is important to highlight that this approach is possible since the points are quasi uniformly distributed. In contrast to the motion monitoring experiment, it is not chosen the number of points per interval, it is chosen the number of breaks. The criteria for choosing the number of breaks is to chose the higher number of breaks that allows a well conditioning matrix coefficient that means the higher one that makes the matrix different from zero.

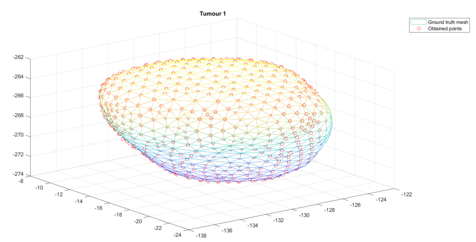
The error metrics are taken as relative error metrics. In this case, the radius provided by the spline is compared with the empirical radius. The metrics are computed in a similar way than the rest of the work including the mean error, the maximum error and the standard deviation of the error. At each patient the result is given.

The results Table 5.20 provide the results approximating each tumour. It concludes that the method perform well in general lines providing stable mean errors and standard deviation of the error at the fourth case. Both values round 1 % in the fourth case although at one tumour the maximum value reaches the 8 %.

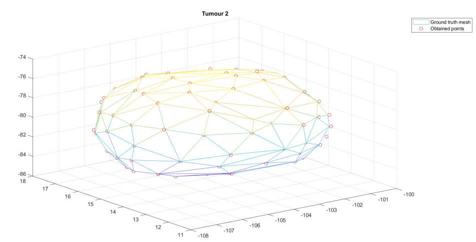
Table 5.20: Relative error metrics of the tumour spline fitting in the four patients.

Patient	Mean error	Max error	Std error
1	0.011	0.081	0.012
2	0.009	0.033	0.009
3	0.007	0.033	0.007
4	0.01	0.01	0.008

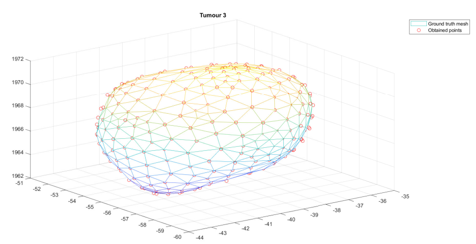
Graphically, four figures are presented to visually inspect the performance of the algorithm at each patient. At the Figure 5.20, it is shown how the spline outcome unconverted to Cartesian coordinated lies in the tumour geometry. We can appreciate how the red points lies well on the geometry mesh of each of the tumour backing the error metrics and showing the feasibility of representing the pre-operative model in a spline.



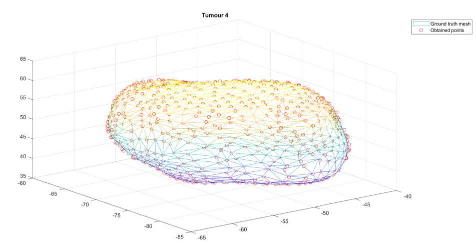
(a) Patient one.



(b) Patient two.



(c) Patient three.



(d) Patient four.

Figure 5.21: Tumour spline outcome laying in ground-truth mesh.

Chapter 6

Conclusion and Future Lines

At this chapter the work closes and the final results are explained. Section 6.1 exhibits the conclusions obtained from the experiments while discussing the technical aspects. Section 6.2 is oriented to discuss the impact of the project and Section 6.3 connects it with the Sustainable Development Goals (SDGs). Section 6.4 shows the limitation of the project and comments the future lines of the investigation.

6.1. General conclusions

The work has successfully approximated several differential equation and it has applied the methodology for developing two bioengineering application, one in the field of motion monitoring and another in the field of image processing. At the project, three group of experiments has been developed, the first one oriented to illustrate how the method works and the other two experiments devoted to develop the two applications that had been proposed. Some different objectives were proposed for this work and these objectives had been covered in one or more of the group of experiments. At Section 1.2 seven objectives, which the first five objectives had been achieved at the Synthetic experiments at Section 5.1, the sixth objective in the motion monitoring experiment at Section 5.2 and the seventh objective in the image processing experiment at Section 5.3.

Starting with objectives archived at the Synthetic experiment. The first objective of scaling the methodology in the variate nature has been explored at Sections 5.1.2, 5.1.4 and 5.1.6 where it has been succeed in the objective of using multivariate spline to solve multivariate function at Section 5.3 this knowledge has been applied for approximating tumours.

The second objective was being able to scale the solver to system of equation. The work successfully solve system of equations and assemble the solutions as a multidimensional spline as as Sections 5.1.3 and 5.1.4 shows. The knowledge is successfully applied at Section

5.2 where the system provided by the Euler angles is solved to estimate the rotation of the IMU.

The third objective it being able to adapt the methodology for solving non-linear equations. Sections 5.1.5 and 5.1.6 tested the adaptation obtaining good results. It is applied also at Section 5.2 since the Euler Angles consists on a non-linear systems of equation.

The fourth objective was to test in the same way as the other three objectives characteristic of the splines and methodology taken during the approximation. At Section 5.1.1 the location and the number of the collocation points, the order of the spline and the number of breaks are tested. The strong conclusions are that the approximation improves using the location points at the Legendre quadrature and that the result improves if the last derivative to take into account corresponds to a fourth order spline, leading to need a higher order spline for the general approximation. Some more weak conclusions, are extracted from this experiment, if the rigidity needed is similar to the needed to a sinus , between five and ten breaks is a good choice and four collocation points per interval in sixth order approximation is a good choice. At section 5.1.5, at the Eikonal approximation, the work shows that introducing a non-zero viscosity is useful when the approximation demands a great unknown elasticity and that the redistribution of knots is useful for increasing the elasticity. This last property is also used to improve the Gompertz approximation at Section 5.1.7.

Closely related, the fifth objective consisted on being able to estimate the number of cells in the tumour proliferation when the tumour has achieved the stage of angiogenesis. Section 5.1.7 proves in a synthetic experiment that the model can be approximated with an acceptable error making its performance comparable to the model given by the analytical resolution.

The sixth objective was to estimate the motion suffered by an IMU in the context of rehabilitation exercise. This objective can be divided in two objectives, the first being able to obtain the orientation of the IMU from the angular velocity and then, incorporate the information to integrate the acceleration so the motion is obtained. Both objective had been tested and achieved in several exercises at Section 5.2, the non-linear differential equation systems provided by the Euler angles had been used to obtain the rotation. Then, it has been developed a double integration for obtaining the motion and it has been successfully employed the Ridge regression to treat the noise induced by gravity and to stabilise the approximation.

The seventh objective was to represent the pre-operative model as a spline. At Section 5.3 the work shows that different tumour pre-operative models can be represented in spherical coordinates as a Spline. It leads to a representation where the tumour is expressed as a function that related the angles azimuth and elevation with the radius.

6.2. Social, economic and environmental impact

Although, at the work we have developed two bioengineering applications, the essence of the work is an updating of the De Boor works that can guide not to only the applications developed, but any other the reader wants. The work has provided the codes in Matlab against the FORTRAN code in the books and it has provided a wider framework of experimentation that the one provided in De Boor books and the Matlab guide. One of the greater impact is that it could be used by researchers and engineers to guide their own applications. In addition to this social dimension, this characteristic has a potentially economic impact since it can reduce the needed of specialised human resources at this field.

The fact that the work lies in the field of analysis with splines makes it have a good impact in the computation efficiency. Splines are considered in different fields as really efficient tool [22] since it allows to set a function in a compressed and easy to evaluate way. This fact has an enormous impact at the economic level since it allows to decrease the computational cost and at the environmental level since it requires less resources to achieve the same objective.

At the motion monitoring experiment, it has been shown the capability of estimation the motion in rehabilitation exercises. This knowledge can be incorporated to the effort to create virtual coaches that does not require human supervision. At the social level, that will increase the accessibility to more people to this kind of exercise since it eliminates the restriction imposed by the need of trained personnel. At the economic level, it can decrease the cost of these kind of treatment and at the environmental level it will reduce unnecessary displacement to perform the treatment and increase the personal mobility of the patients.

The image processing experiment is more versatile but it is also at a more primary stage. At a social level it has the potential to improve the current pre-operative stages allowing the utility of more advanced technology and improving the outcome of Laparoscopic Liver Resection. At the economic level, it has the potential of making this kind of technology cheaper and at the environmental level to reduce the resources needed in this kind of intervention if it is successfully applied.

6.3. Impact related with SDGs

Once the impact of the project is established, it is connected to the Sustainable Development Goals (SDGs). They are a set of goals established by the United Nations (UN) in 2015 that includes the most urgent challenges to give an approach by the 2030. It is composed by seventeen goals related to poverty, social justice, environmental protection and, social justice and equality promotion.

Due to the versatility of the work, it could potentially aid in the achievement of several goals but, since the most developed part of the work corresponds to the experiments in the bioengineering title the work fits perfectly in the third goal (good-health and well being). The objective is to ensure healthy loves and promote well-being for all at all ages.

Table 6.1: Impact related to the target of SDG goal three.

Target	Statement	Alignment
Target 3.4	"By 2030, reduce by one third premature mortality from non-communicable diseases through prevention and treatment and promote mental health and well-being."	This objective aligns with the essence of the motion monitoring experiments since it allows the expansion of the rehabilitation that promotes well-being and prevents non-communicable diseases.
Target 3.8	"To achieve universal health coverage, including financial risk protection, access to quality essential health-care services and access to safe, effective, quality and affordable essential medicines and vaccines for all."	Since the orientation of both experiments look for making health assistance cheaper it has a positive impact in the access of quality essential health-care.
Target 3.D	"Strengthen the capacity of all countries, in particular developing countries, for early warning, risk reduction and management of national and global health risks."	The image processing experiments could lead to an improvement in the management of liver risks.

In general terms, a positive impact aligned with the SDGs can be extracted from this work. Highlighting the fact that this work cannot only have the impact of the most developed parts but it can have an impact in other goals and target by using this work to develop other applications.

6.4. Limitations and future lines

This work has several limitations introduced to develop the experiments that should be discussed. These discussion should lead to set some future lines to keep improving in the experiments and in the technical part. The first and the most general limitation is that the work is developed in a context of imposed limited computation requirements. It has an impact on the way the collocation matrix and the coefficients is computed. In an scenario of increasing needs De Boor recursion formula could be inefficient for evaluating or computing the collocation matrix. Several algorithms has been developed to solve this problem although they are less stable. In the future, it is interesting involving this algorithm and study how to implement them. In a parallel way, the solver used could be not practical when great amount of coefficients must be estimated and alternatives ways should be explored for this scenario.

Following with the computational aspects, we have highlighted the fact that splines are considered a computationally efficient framework but at the work it has not been compared with comparable frameworks. Establishing comparisons with other ways to evaluate the computational efficiency can be interesting to take into account this aspect when taking the decision of selecting and designing the approach. Continuing in this branch, the second experiment has been tested without taking into account that it should be used in a near real-time scenario so as another important future line it is to evaluate practical time implementation to fulfil the objectives.

Also at the first experiment, the main limitation is that the ridge regression requires a parameter that at this level has been developed in an intuitive way. Studying deeper the knowledge and looking robust way of determining this parameter is necessary to improve the work. Also, the statistical rectification that is done at this step needs to be debugged and deeply explored. The third, limitation that had been found at this application is the selection of the sub-interval that had been done without a robust criteria. In fourth order fitting function, Matlab provides the possibility of using a tolerance criteria where the sub-interval length are selected to not allow the fitting points exceed the error. Extending that development to this work could be useful to make the criteria stronger and establishing acceptable tolerance that gives small error at that point but guarantying a coherent estimation at all the domain.

At the second experiment, the main limitation is given by the approach used to model the spline. It requires that the topology does not have any overlapping with respect to the chosen centre of coordinates. It limits the pre-operative models that can be represented and returns problems if the model does not fulfil this condition at any part. A future scope is finding stronger assumptions that allows to represent the model as a spline but that are not so restrictive. The other limitation that arises in the second experiment is that closed path domain are represented with open path domain, it leads to a great computational cost to close the path with weak constrictions or to not imposing them. Designing close path basis splines to approximate these functions is an interesting way of improving the work.

Bibliografía

[1] Bonnie Beker. Aplicar la fusión de sensores a acelerómetros y giroscopios, 2018.

[2] Mateo Aboy, Kathleen Liddell, Johnathon Liddicoat, Cristina Crespo, and Matthew Jordan. Mapping the european patent landscape for medical uses of known products. *Nature Biotechnology*, 39(11):1336–1343, 2021.

[3] Claude Brezinski and Luc Wuytack. *Numerical analysis: Historical developments in the 20th century*. Elsevier, 2012.

[4] Jeffrey Humpherys and Tyler J Jarvis. *Foundations of Applied Mathematics Volume 2: Algorithms, Approximation, Optimization*. SIAM, 2020.

[5] RK Sachs, LR Hlatky, and P Hahnfeldt. Simple ode models of tumor growth and anti-angiogenic or radiation treatment. *Mathematical and Computer Modelling*, 33(12-13):1297–1305, 2001.

[6] World Health Organization. Rehabilitation. `https://www.who.int/news-room/fact-sheets/detail/rehabilitation`.

[7] Kirsten Jack, Sionnadh Mairi McLean, Jennifer Klaber Moffett, and Eric Gardiner. Barriers to treatment adherence in physiotherapy outpatient clinics: a systematic review. *Manual therapy*, 15(3):220–228, 2010.

[8] Sara García-de Villa, Ana Jiménez-Martín, and Juan Jesús García-Domínguez. A database of physical therapy exercises with variability of execution collected by wearable sensors. *Scientific Data*, 9(1):266, 2022.

[9] Yinoussa Adagolodjo, Raffaella Trivisonne, Nazim Haouchine, Stéphane Cotin, and Hadrien Courtecuisse. Silhouette-based pose estimation for deformable organs application to surgical augmented reality. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 539–544. IEEE, 2017.

[10] Prashanth Dumpuri, Logan W Clements, Benoit M Dawant, and Michael I Miga. Model-updated image-guided liver surgery: preliminary results using surface characterization. *Progress in biophysics and molecular biology*, 103(2-3):197–207, 2010.

[11] Thomas M Nosek. Section 6/6ch2/s6ch2_30. *Essentials of Human Physiology. Archived from the original on*, pages 03–24, 2016.

[12] Lena Maier-Hein, Anja Groch, Adrien Bartoli, Sebastian Bodenstedt, Guillaume Boissonnat, P-L Chang, Neil T Clancy, Daniel S Elson, Sven Haase, Eric Heim, et al. Comparative validation of single-shot optical techniques for laparoscopic 3-d surface reconstruction. *IEEE transactions on medical imaging*, 33(10):1913–1930, 2014.

[13] N Rabbani, Lilian Calvet, Y Espinel, B Le Roy, M Ribeiro, Emmanuel Buc, and Adrien Bartoli. A methodology and clinical dataset with ground-truth to evaluate registration accuracy quantitatively in computer-assisted laparoscopic liver resection. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 10(4):441–450, 2022.

[14] IJ Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. part a. on the problem of smoothing or graduation. a first class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4(1):45–99, 1946.

[15] Carl De Boor and Carl De Boor. *A practical guide to splines*, volume 27. springer-verlag New York, 1978.

[16] Carl De Boor, Klaus Höllig, and Sherman Riemenschneider. *Box splines*, volume 98. Springer Science & Business Media, 2013.

[17] Haohao Hu, Johannes Beck, Martin Lauer, and Christoph Stiller. Continuous fusion of motion data using an axis-angle rotation representation with uniform b-spline. *Sensors*, 21(15):5004, 2021.

[18] Yunhui Li, Shize Yang, Xianchao Xiu, and Zhonghua Miao. A spatiotemporal calibration algorithm for imu–lidar navigation system based on similarity of motion trajectories. *Sensors*, 22(19):7637, 2022.

[19] Gabriel Nützi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of imu and vision for absolute scale estimation in monocular slam. *Journal of intelligent & robotic systems*, 61(1-4):287–299, 2011.

[20] Jianzhu Huai, Yuan Zhuang, Yukai Lin, Grzegorz Jozkow, Qicheng Yuan, and Dong Chen. Continuous-time spatiotemporal calibration of a rolling shutter camera-imu system. *IEEE Sensors Journal*, 22(8):7920–7930, 2022.

[21] Junfang Bao, Jianli Li, Chunyu Qu, and Yunzhu Li. Multi-node motion estimation method based on b-spline of array position and orientation system. *Remote Sensing*, 15(11):2892, 2023.

[22] Richard Szeliski and James Coughlan. Spline-based image registration. *International Journal of Computer Vision*, 22:199–218, 1997.

[23] Sachira Kuruppu, Leo K Cheng, Poul MF Nielsen, Thiranja P Babarenda Gamage, Recep Avci, Timothy R Angeli, and Niranchan Paskaranandavadivel. High-resolution spatiotemporal quantification of intestinal motility with free-form deformation. *IEEE Transactions on Biomedical Engineering*, 69(6):2077–2086, 2021.

[24] Romel Bhattacharjee, Ashish Verma, Neeraj Sharma, and Shiru Sharma. Non-rigid registration (computed tomography-ultrasound) of liver using b-splines and free form deformation. *International Journal of Biomedical Engineering and Technology*, 39(4):327–346, 2022.

[25] Guoya Dong, Jingjing Dai, Na Li, Chulong Zhang, Wenfeng He, Lin Liu, Yinping Chan, Yunhui Li, Yaoqin Xie, and Xiaokun Liang. 2d/3d non-rigid image registration via two orthogonal x-ray projection images for lung tumor tracking. *Bioengineering*, 10(2):144, 2023.

[26] Erik J Bekkers. B-spline cnns on lie groups. *arXiv preprint arXiv:1909.12057*, 2019.

[27] Nicola Altini, Antonio Brunetti, Valeria Pia Napoletano, Francesca Girardi, Emanuela Allegretti, Sardar Mehboob Hussain, Gioacchino Brunetti, Vito Triggiani, Vitoantonio Bevilacqua, and Domenico Buongiorno. A fusion biopsy framework for prostate cancer based on deformable superellipses and nnu-net. *Bioengineering*, 9(8):343, 2022.

[28] Guanhua Wang, Tianrui Luo, Jon-Fredrik Nielsen, Douglas C Noll, and Jeffrey A Fessler. B-spline parameterized joint optimization of reconstruction and k-space trajectories (bjork) for accelerated 2d mri. *IEEE Transactions on Medical Imaging*, 41(9):2318–2330, 2022.

[29] Chaithya Giliyar Radhakrishna and Philippe Ciuciu. Jointly learning non-cartesian k-space trajectories and reconstruction networks for 2d and 3d mr imaging through projection. *Bioengineering*, 10(2):158, 2023.

[30] T L_etc Lai, Herbert Robbins, and C Zi Wei. Strong consistency of least squares estimates in multiple regression ii. *Journal of multivariate analysis*, 9(3):343–361, 1979.

[31] Carl Runge. Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten. *Zeitschrift für Mathematik und Physik*, 46(224-243):20, 1901.

[32] Stefano De Marchi, Francesco Marchetti, Emma Perracchione, and Davide Poggiali. Polynomial interpolation via mapped bases without resampling. *Journal of Computational and Applied Mathematics*, 364:112347, 2020.

[33] Nicolas Belanger. External fake constraints interpolation: the end of runge phenomenon with high degree polynomials relying on equispaced nodes–application to aerial robotics motion planning. *indicator*, 2:1–9, 2017.

[34] Dennis G Zill. *A first course in differential equations with modeling applications*. Cengage Learning, 2012.

[35] A Earl and Norman Levinson. Theory of ordinary differential equations, 1955.

[36] Andrei D Polyanin and Valentin F Zaitsev. Handbook of exact solutions for ordinary differential. 1995.

[37] Arieh Iserles. *A first course in the numerical analysis of differential equations*. Number 44. Cambridge university press, 2009.

[38] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.

[39] JD Blanton et al. *Foundations of differential calculus*. Springer Science & Business Media, 2006.

[40] Victor Fairen, Vicente Lopez, and Luis Conde. Power series approximation to solutions of nonlinear systems of differential equations. *American Journal of Physics*, 56(1):57–61, 1988.

[41] Carl Runge. Über die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.

[42] Boling Guo, Dongfen Bian, Fangfang Li, and Xiaoyu Xi. *Vanishing viscosity method: solutions to nonlinear systems*. Walter de Gruyter GmbH & Co KG, 2016.

[43] Franz S Hover and Michael S Triantafyllou. System design for uncertainty. *Mass. Inst. Technol*, 2009.

[44] Tadej Beravs, Janez Podobnik, and Marko Munih. Three-axial accelerometer calibration using kalman filter covariance matrix for online estimation of optimal sensor orientation. *IEEE Transactions on Instrumentation and Measurement*, 61(9):2501–2511, 2012.

[45] Ahmed M Elsherbeny, Rabab MI El-hassani, Hesham El-badry, and Mohamed I Abdallah. Solving 2d-poisson equation using modified cubic b-spline differential quadrature method. *Ain Shams Engineering Journal*, 9(4):2879–2885, 2018.

[46] William E Boyce and Richard C DiPrima. *Elementary differential equations and boundary value problems*. Wiley, 2020.

[47] José Sérgio Domingues. Gompertz model: Resolution and analysis for tumors. *Journal of Mathematical Modelling and Application*, 1(7):70–77, 2012.

[48] Sabrina L Spencer, Matthew J Berryman, José A García, and Derek Abbott. An ordinary differential equation model for the multistep transformation to cancer. *Journal of Theoretical Biology*, 231(4):515–524, 2004.