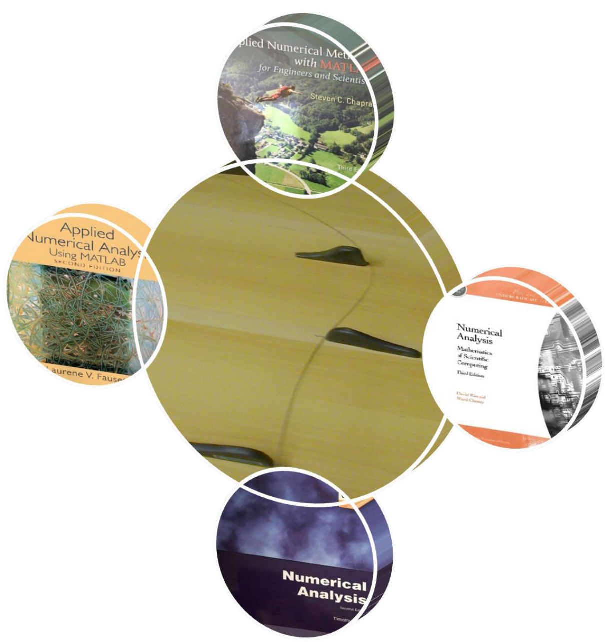
LUND UNIVERSITY
Faculty of Science

MASTER THESIS

Comparison and Evaluation of Didactic Methods in Numerical Analysis for the Teaching of Cubic Spline Interpolation

Abtihal Jaber Chitheer
supervised by Prof. Dr. Carmen Arévalo



May 17, 2017

Lund University

Abstract

Faculty of Science
Centre for Mathematical Sciences
Numerical analysis

Master in Science

Comparison and Evaluation of Didactic Methods in Numerical Analysis for the Teaching of Cubic Spline Interpolation

by Abtihal Jaber Chitheer

In mathematical education it is crucial to have a good teaching plan and to execute it correctly. In particular, this is true in the field of numerical analysis. Every teacher has a different style of teaching. This thesis studies how the basic material of a particular topic in numerical analysis was developed in four different textbooks. We compare and evaluate this process in order to achieve a good teaching strategy. The topic we chose for this research is cubic spline interpolation. Although this topic is a basic one in numerical analysis it may be complicated for students to understand. The aim of the thesis is to analyze the effectiveness of different approaches of teaching cubic spline interpolation and then use this insight to write our own chapter. We intend to channel every-day thinking into a more technical/practical presentation of a topic in numerical analysis. The didactic methodology that we use here can be extended to cover other topics in numerical analysis.

Acknowledgements

I would like to express my gratitude towards my country Iraq for the scholarship. I would like to thank Lund University for giving me the opportunity to do my master degree. I would like to thank my supervisor Carmen Arévalo for giving me this opportunity to do my master thesis in the field of numerical analysis and for the guidance and support throughout the process. I would like also to thank the department of Mathematics for feedback and encouragement. Finally I am very grateful for support from my loving family and friends.

Contents

Chapter 1

Introduction

Many problems in mathematics can be presented in several different ways, but all lead us to the same results. Most problems are complex problems, that might be difficult for students to understand. When I was working in Iraq as a mathematics teacher during the years of 2009-2013 I realized that different ways of teaching will affect the students' level of understanding. Piecewise polynomial interpolation is one topic that students find especially difficult to understand, so we would like to find didactic strategies for teaching this subject. The topic of cubic spline interpolation is a basic topic in the course of numerical analysis and yet students have difficulties in following the process of the construction of a cubic spline based on the conditions a spline must satisfy.

These problems are solved numerically, usually with the help of computers and thus learning how to program this process is also a skill students must develop.

1.1 Aim of the thesis

A mathematical object or concept can be introduced by different sets of equivalent definitions and examples. One fundamental question in this context is that of what teaching method to use. Reading several books on a common topic makes one realize how different similar teachings can be, in style and requirements towards a student. Our goal, in this work of comparison, is to analyze the effectiveness of different approaches and teachings of some particular topics through different course materials, and how to compare and decide between different books with future work in Iraq in mind.

Eventually, the aim of this work is to channel every-day thinking towards a more technical-practical thinking at an earlier stage so that, as a teacher, one can get the expected results from the teaching of a particular topic.

1.2 Spline interpolation

Here we chose to focus on the teaching of spline interpolation. The choice of this topic has two main motivations. First of all, it is one of the basics of numerical analysis. Using an example that is easily understandable by undergraduate students of mathematicians makes the point of the thesis easier to illustrate. Secondly, there are many differences in views and the teaching of this subject, and the study of these differences helps to reach the goal of this thesis. Diversity in teaching strategies gives us tools to reach our goal.

1.3 Didactic reflection:

Variation theory of learning was developed by Ference Marton of the University of Gothenburg in 1997. This theory says that for a teacher to be able to help a student understand a topic, the teacher must first identify the critical aspects of the topic. Once this is done, by varying one aspect while keeping other aspects fixed, the student will learn by observing the differences. Variation theory can be interpreted as "how the same thing or the same situation can be seen, experienced or understood in a limited number of qualitatively different ways" [Maghdid, 2016]. For example, we apply variation theory when we propose several problems that can be solved with the same technique, or when we have one example that can be solved in several different ways. In our development of a chapter for the teaching of cubic spline interpolation we will apply this strategy to try to reach a successful didactics method.

1.4 Additional notes

The four books we have chosen to work with are well-known and used often in the teaching of numerical analysis.

Our goal is not to describe one perfect way of teaching, but to hierichize the methods and teaching strategies by their efficiency.

Any student with basic knowledge of numerical analysis can read this thesis. As we have stated before, our point is not the mathematical aspects but the didactics. Therefore, a basic knowledge is enough and necessary in order to understand most of our points and examples, which will only be clearer if the reader has studied spline interpolation in the past. Furthermore, any person interested in the teaching of mathematics may be interested in reading this work, as successful teaching results from a critical point of view, and especially a self-critical one.

I am a former student of the Department of Mathematics at Mustansiriya University, Iraq, where I have obtained a Bachelor's degree in theoretical mathematics. I was then employed as a mathematics teacher in the schools Fatima Zahra and Ishtar, where I was often confronted with student complains about the lack of easy accessibility to concepts in theoretical mathematics. Their abstract aspect and the specificity of their applications made students unable to identify their own errors when obtaining incorrect solutions to a problem. I therefore got very drawn into constructing proofs and solutions that were more interesting and transparent. I aspire to develop a better teaching method, more comfortable for both teachers and students, by, for example, privileging illustration over expansive calculations.

Chapter 2

Analysis of four different approaches to the teaching of splines

In this thesis I have compared the presentation of the topic *cubic splines* from four different Numerical Analysis books intended to be used as textbooks in regular undergraduate courses.

2.1 Comparison criteria

The four numerical analysis books present the topic under study in considerably different ways. In order to understand their similarities, differences, weaknesses and strengths, we will use the following key points as comparison criteria: *insertion point*, *teaching structure*, and *outcome procurement*. Analysis of the structure of teaching is a logical requirement to explain different effectiveness in teaching outcome. To understand the way an author argues we have to take specific study and comparison criteria. These key points are as follows:

1. *Insertion point*: Here we will focus on how the author introduces the concept of *cubic splines*. We will study the ways in which the authors motivate and direct the reader to cubic splines. For example, how was the presentation initiated? What are the different types of interpolation that are described before starting the cubic spline section? How is the previous material on interpolation used to connect to this subject?

2. *Teaching structure*: This part is divided into two subsections. Firstly, we touch on the *Introduction* process. What are the key points in the initial presentation of cubic splines? What was the strategy for developing the important concepts in the topic? Secondly, we look at at the *elaboration* process. In this part we study how the authors expand on the topic, developing each point so that there is a line of development throughout. For example, how is the derivation of cubic splines done? How many end conditions are mentioned? Were there any Theorems mentioned or were the explanations of a more informal type?

3. *Outcome procurement*: This part deals with the conclusions of the presentation of the subject. For example, were there examples and applications that clarified the concept

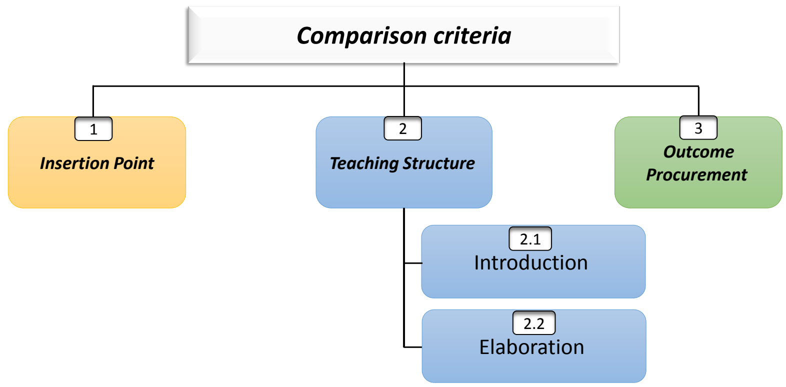


Figure 2.1: Scheme of the comparison criteria keys

of cubic splines? Are there exercises that allow the reader to test his understanding of cubic splines? Were there any summaries at the end of the presentation?

Using these criteria we will be able to study and analyze each book in a systematic manner so that we can use this knowledge in the next chapter to compare the four books and evaluate their didactic methods. The comparison will be made in the form of questions and answers, and this will allow us to go on to the evaluation of the books. Finally, using the results of previous chapters we will come with our conclusion and suggest a didactic method for this topic.

The tables and plots shown in the presentation of the books in the following subsections were taken directly from the books themselves. The last section of the thesis, the *Appendix*, displays a copy of the chapters or sections that have been mentioned in this thesis. This was done in order to make it easier for the reader to compare the original versions with the thesis.

2.2 Book descriptions

The analysis of each book will start with a schematic representation of how the subject of cubic splines is presented and developed. This will be done by using the key comparison criteria mentioned before, that is, *insertion point*, *teaching structure*, and *outcome procurement*. We will then go on to explain each section in detail.

2.2.1 Book I: Applied Numerical Methods with MATLAB for Engineers and Scientists by Steven C. Chapra, 2011.

Steven C. Chapra is professor at the Department of Civil and Environmental Engineering at Tufts University in Massachusetts, U.S.A. He specializes in environmental engineering.

This book focuses on applied numerical analysis using MATLAB. The book can be used for teaching at a basic level and at a more advanced level, and Chapra's book is widely used

Figure 2.2: Steven C.Chapra, author of the book *Applied Numerical Methods with MATLAB for Engineer and Scientists*, 2011.

as textbook in different countries. For example, Florida University ([University of Florida, 2009]) in the United States has been using this book for the course *Introduction of Numerical Methods of Engineering Analysis*. McMaster University in Canada recently used Chapra's book as textbook in the course *Computer Engineering 3 SK3 Numerical Analysis* ([McMaster University, 2014]), and Hacettepe University in Turkey uses this book as textbook in the course *Numerical Analysis with MATLAB* ([Hacettepe Üniversitesi, 2017]). In Italy, it is also used for the course *Numerical Methods* at the online Universita Telematica Internazionale UNINETTUNO. The topic of *cubic splines* is a 10-page section in the chapter *Splines and Piecewise Interpolation*. *Cubic Splines* is mentioned as the last type of spline interpolant and this section also includes a primer on piecewise interpolation in MATLAB.

2.2.1.1 Insertion Point

The chapter titled *Splines and Piecewise Interpolation* in Chapra's book is grounded on its preceding chapter, *Polynomial Interpolation*, where Runge's function was introduced as an example of the dangers of higher-order polynomial interpolation. The motivation starts by mentioning the problems observed in the previous chapter with higher degree polynomials, but without any reference to Runge's function example. In addition, the author proposes an alternative solution to this problem by using piecewise lower order polynomials (spline functions).

The chapter on splines and polynomial interpolation is opened with an example. When a function varies sharply at a point, interpolation polynomials of high degree induce undesired oscillations. This problem is illustrated in Figure 2.4. A linear 3-piece polynomial, as shown in (d), does a much better job that the polynomials of degree 3, 5 and 7, shown in (a, b, c).

The notion of *splines* is introduced as piecewise lower-order interpolating polynomials, and they are suggested as an alternative to higher-order polynomials when the function to interpolate has local, abrupt changes.

After presenting the reader with a motivation to study splines, the author gives a historical account of spline construction. In order to "draw smooth curves through a set of points" when a plan for some construction was drafted, a thin, flexible strip (called a *spline*) was used. The author points to Figure 2.5 to illustrate that the data point (pins) are "interpolated" by

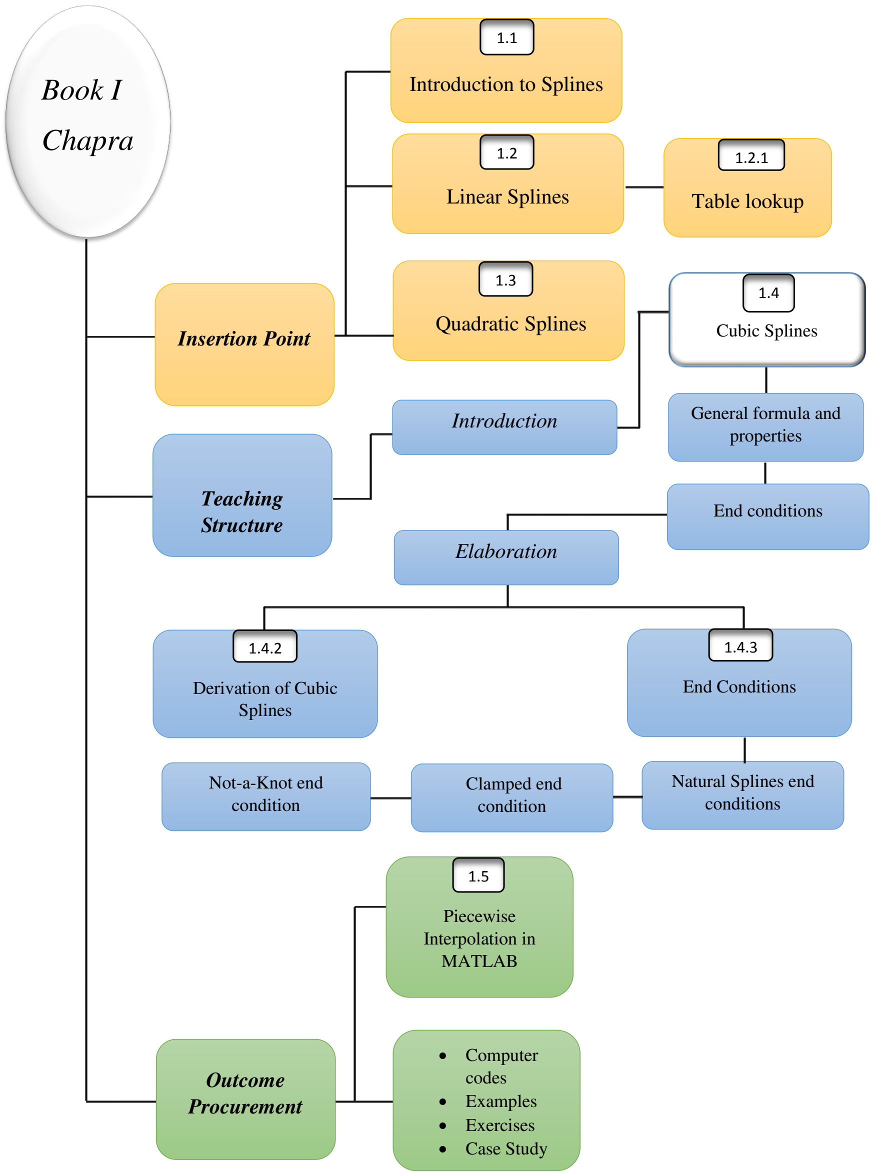


Figure 2.3: Schematic content and organization of Chapter 18, "Splines and Piecewise Interpolation" in Chapra's book [Chapra, 2011].

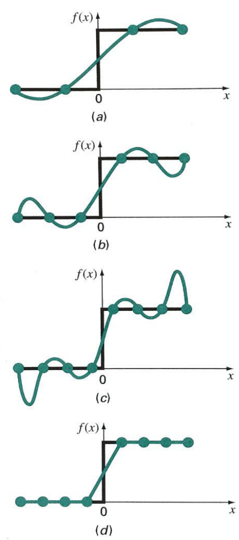


FIGURE 18.1
A visual representation of a situation where splines are superior to higher-order interpolating polynomials. The function to be fit undergoes an abrupt increase at $x = 0$. Parts (*a*) through (*c*) indicate that the abrupt change induces oscillations in interpolating polynomials. In contrast, because it is limited to straight-line connections, a linear spline (*d*) provides a much more acceptable approximation.

Figure 2.4: Spline vs. high-order interpolation ([Chapra, 2011], p. 430).

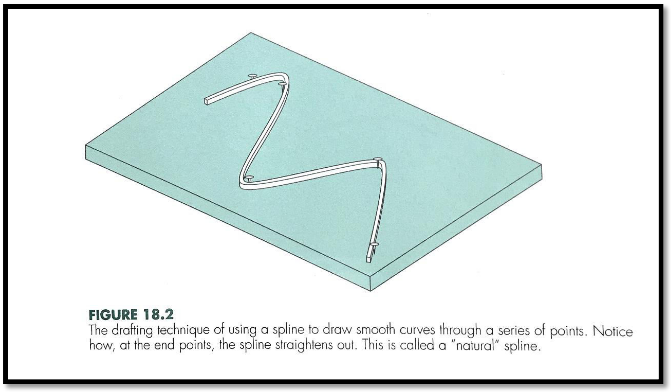


Figure 2.5: Illustration of an early use of splines to construct smooth curves passing through pre-defined points (Chapra [2011], p. 431).

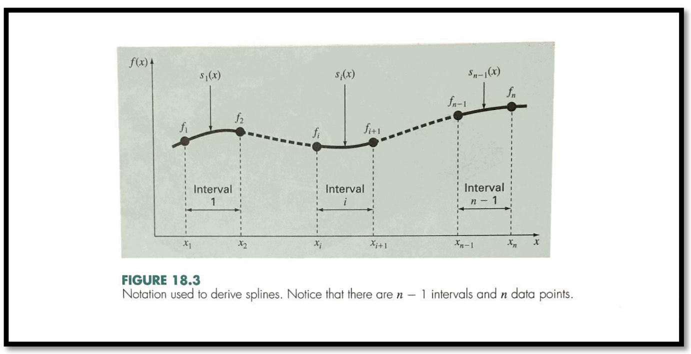


Figure 2.6: This figure displays the notation used to derive splines (Chapra [2011], p. 432).

smooth curves as the strip goes through the pins. In fact, the author asserts that these curves are in fact cubic splines.

At the end of his introductory section, the author gives an overview of the topics that he will discuss in the remainder of the chapter.

Linear Splines

The section on *linear splines* starts with a detailed description of the notation, as portrayed in Figure 2.6.

Firstly, the reader is asked to observe that n distinct data points generate $n-1$ intervals. It is then shown that for each interval i there is a spline function $S_i(x)$, and that a linear spline has a straight line connection between the two end points of the intervals. Thus, it is easy to formulate the equations for these splines. Then, the practical aspect of this formulation

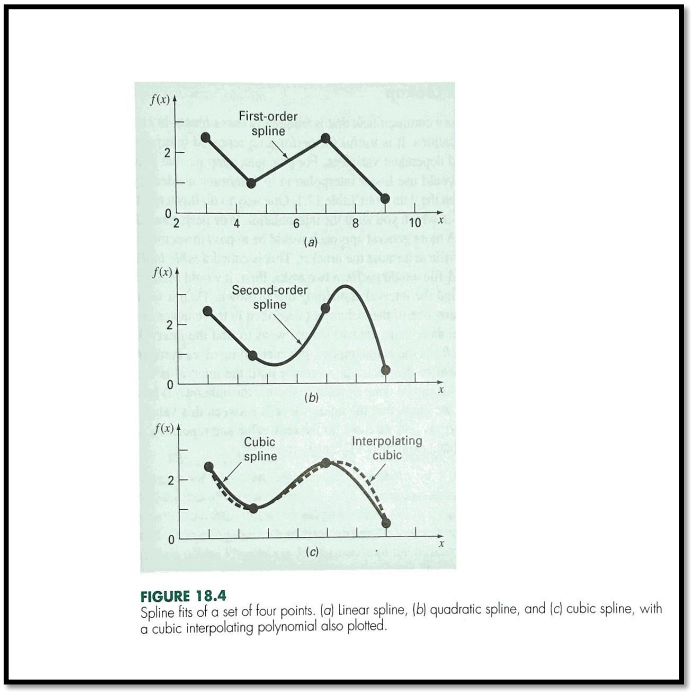


Figure 2.7: (a) Linear spline, (b) Quadratic spline, (c) Cubic spline, with a cubic interpolation polynomial, ([Chapra, 2011, p. 433].

is exposed: they can be used to evaluate the function at any point in the interval containing the data points. The author explains that there are two steps in doing so: (1) locate the subinterval where the point lies, and (2) use the appropriate equation to evaluate the spline. An example is given to illustrate these steps.

Chapra points out that using linear splines is equivalent to "using Newton's first-order polynomial to interpolate within each interval". An example shows the first-order spline, the second-order spline, and the cubic spline interpolations of four data points (Figure 2.7). The aim of Figure 2.7, is to demonstrate that the same data points can be interpolated by splines of different orders: a linear spline, a quadratic spline, and a cubic spline, and that the resulting splines will yield different function evaluations. It also introduces the concept of smoothness of a curve, and its relation to its derivatives.

Here another new concept is introduced: a *knot* is defined as a data point where two spline pieces meet. The author indicates the main disadvantages of linear splines:

1. First-order splines are not smooth.

2. The slope changes abruptly where the spline segments meet (knots).

3. The first derivative of the function is discontinuous at these specific points.

TABLE 17.1 Density (ρ), dynamic viscosity (μ), and kinematic viscosity (v) as a function of temperature (T) at 1 atm as reported by White (1999).

T, °C	ρ, kg/m³	μ, N · s/m²	v, m²/s
−40	1.52	1.51×10^{-5}	0.99×10^{-5}
0	1.29	1.71×10^{-5}	1.33×10^{-5}
20	1.20	1.80×10^{-5}	1.50×10^{-5}
50	1.09	1.95×10^{-5}	1.79×10^{-5}
100	0.946	2.17×10^{-5}	2.30×10^{-5}
150	0.835	2.38×10^{-5}	2.85×10^{-5}
200	0.746	2.57×10^{-5}	3.45×10^{-5}
250	0.675	2.75×10^{-5}	4.08×10^{-5}
300	0.616	2.93×10^{-5}	4.75×10^{-5}
400	0.525	3.25×10^{-5}	6.20×10^{-5}
500	0.457	3.55×10^{-5}	7.77×10^{-5}

Figure 2.8: Table (17.7) from the previous chapter, showing air density at different temperatures (Chapra [2011], p. 406).

The disadvantage of the non-smoothness of linear splines is a motivation why it is important to consider higher-order splines. Nevertheless, the author states that linear splines are important in their own right, and gives an example of their usefulness.

Table Lookup

Chapra uses the example of *a table lookup* to show why linear spline interpolation may be useful. The idea is to interpolate values from a table of independent and dependent variables. "For example, suppose that you would like to set up an M-file that would use linear interpolation", if programming in MATLAB, with specific data that you have as a table, for example Table 2.8.

Given such a table, to perform a linear interpolation at a particular value of the independent variable, the subinterval containing the required value must be found. The action of finding the appropriate values in the table is called *table lookup*. Here the author explains in detail how to construct a Matlab file that does the interpolation task. He describes the process in two steps:

1. search of the interval where the independent variable is located, and

2. linear interpolation.

The first step is described in detail, and two different types of search are presented: a sequential search, for a small set of data points, and a binary search, for large sets of data.

The next section takes on the aspect of the continuity of the derivatives of the interpolating function.

Quadratic Splines

The author starts this section by claiming that in order to have n-the continuous derivatives, a spline must be of degree $n+1$ or higher. Then he asserts that the most frequently used splines are the cubic ones, because higher derivative discontinuities at the knots in most cases cannot be observed visually. Even though the author's aim is to discuss cubic splines, he chooses to go through the derivation of *quadratic splines* because this derivation is less complicated.

The derivation of quadratic splines is done in detail in order to show what are the important points that must be taken into account. It starts by giving the general form of each spline segment as a quadratic polynomial centered at the starting point of each subinterval,

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 \tag{2.1}$$

Then it proceeds to count the number of intervals, equations, and unknowns, which must correspond to the number of coefficients of the spline. Finally, it gives the conditions that must be satisfied by the unknowns. These conditions are *continuity conditions*, *interpolation conditions* and *end conditions*. By counting the number of continuity and interpolation conditions, it is clear that there must be one end condition so that the number of equations (conditions) equals the number of unknowns.

Finally, an example is given (Example 18.2 in the book) to illustrate this construction. This example uses the same data as the previous example in the section for linear splines. In addition, it refers the reader to Figure 2.7, where (b) shows this quadratic spline, and points to the fact that a straight line connects the first two data points due to the end condition, and also that the curve for the last interval swings too high. These observations serve as a motivation to study cubic splines, as they do not exhibit this undesired behaviour.

2.2.1.2 Teaching Structure

This section is divided into two parts. Firstly, the *Presentation* part refers to the introduction of the basic material defining cubic splines. Secondly, the *Elaboration* part contains further details about cubic splines.

Introduction

The author starts by restating the disadvantages of using linear or quadratic splines, and then states that higher degree splines tend to exhibit inherent instabilities, while cubic splines "exhibit the desired appearance of smoothness" while providing the simplest representation. The steps in which the author displays *cubic splines* follow the lines of his exposition for quadratic splines:

1. General form of *cubic splines*.

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{2.2}$$

2. Number of intervals, unknowns, and continuity and interpolation conditions.

3. Calculation of the number of end conditions that need to be added, and what formulations they can have.

4. Enforcement of the continuity and interpolation conditions.

5. Discussion of the several options for end conditions. One type of end condition is, for example, the conditions for a *natural* spline, which can be observed in Figure 2.5. Additionally, to complement the choice of several options, the author adds two more options of end conditions which are described in Section 18.4.2 of the book.

6. Enforcement of the end conditions.

In the next section we present the details of the derivation of cubic splines in Chapra's book.

Elaboration

There are two main issues in the construction of cubic splines. Firstly, the derivation of cubic splines from the continuity and interpolation conditions. Secondly, the enforcement of end conditions.

Derivation of Cubic Splines

A major reason for the introduction and derivation of quadratic splines was to pave the way to the derivation of cubic splines. Correspondingly, also here the number of conditions and the number of unknowns are determined, and it is shown that two additional *end conditions* are required. The construction of cubic splines is demonstrated step by step.

1. The formula for a *cubic spline* (third-order polynomial for each interval between knots) is given. This formula is centered around the initial point of each subinterval.

2. The interpolation and continuity conditions for the *cubic spline* at the knots are enforced.

3. The continuity conditions for the first derivative of the *cubic spline* at the knots are enforced.

4. The continuity conditions for the second derivative of the *cubic spline* at the knots are enforced.

5. The result obtained in the previous steps are substituted into the original formula, yielding a set of equations containing only the unknowns corresponding to the coefficients of the quadratic terms of the spline sections. These equations hold for all the internal knots.

6. The two end conditions are added to the system. As an example, the conditions taken are those for a natural spline, i.e., that the second derivative of the spline is equal to zero at the endpoints.

7. The linear system is shown in a matrix-vector form, showing that the system matrix is tridiagonal.

Finally, an example of the construction of a natural cubic spline is shown in [p. 442] of Chapra's book ([Chapra, 2011]). The author provides the same data that was used for linear and quadratic splines, and the results are displayed in Figure 2.7.

The first type of end conditions applied to cubic splines resulted in *natural splines*. In the next section, the author describes other end conditions.

End Conditions

TABLE 18.2 The first and last equations needed to specify some commonly used end conditions for cubic splines.

Condition	First and Last Equations
Natural	$c_1 = 0, c_n = 0$
Clamped (where f'_1 and f'_n are the specified first derivatives at the first and last nodes, respectively).	$2h_1c_1 + h_1c_2 = 3f[x_2, x_1] - 3f'_1$ $h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'_n - 3f[x_n, x_{n-1}]$
Not-a-knot	$h_2c_1 - (h_1 + h_2)c_2 + h_1c_3 = 0$ $h_{n-1}c_{n-2} - (h_{n-2} + h_{n-1})c_{n-1} + h_{n-2}c_n = 0$

Figure 2.9: How to specify different end conditions (Chapra [2011], p.443).

I: Natural End Condition

As described earlier, a natural spline is defined by requiring that the second derivatives at the two endpoints are equal to zero. This type of spline results when the drafting spline is allowed to behave naturally, without constraining it at its endpoints.

II: Clamped End Condition

Clamped splines result when a drafting spline is clamped at its ends. The first derivative is specified at the two ends.

III: Not-a-Knot End Condition

This condition implies that the third derivative is required to be continuous at the second and the next-to-last knots. Because the spline segments are cubic polynomials, this condition implies that the first and second segments are a single polynomial, and so are the last and next-to-last segments. Thus, these knots are not true knots any longer.

Table 2.9 shows how to modify the first and last equations in order to construct *natural*, *clamped* and *not-a-knot* splines. Figure 2.10 shows the different end conditions by using the same example that was mentioned last section.

2.2.1.3 Outcome Procurement

In the next section the author describes how the user can take advantage of some of Matlab's built-in functions for piecewise interpolation. Two problems are solved as examples. Some snippets of code are given, and the results are plotted.

Piecewise Interpolation in MATLAB

The author introduces two possible functions for piecewise interpolation.

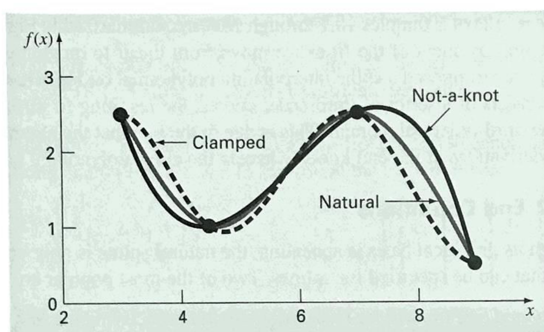


FIGURE 18.5
Comparison of the clamped (with zero first derivatives), not-a-knot, and natural splines for the data from Table 18.1

Figure 2.10: Using the data of the previous example, splines with *natural*, *clamped* and *not-a-knot* end conditions are plotted (Chapra [2011], p. 444).

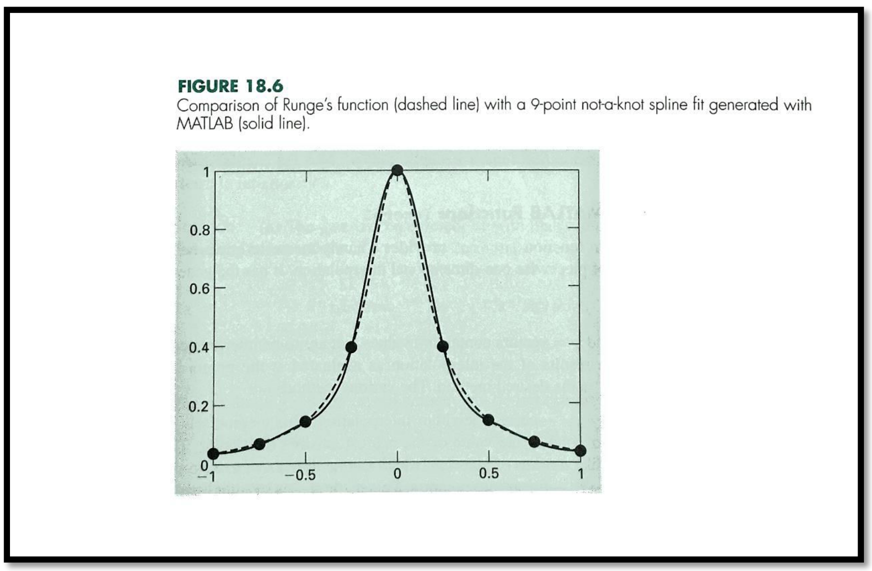


Figure 2.11: Runge's function with nine data points constructed in MATLAB by using a "not-a-knot" end condition ([Chapra, 2011, p. 445]).

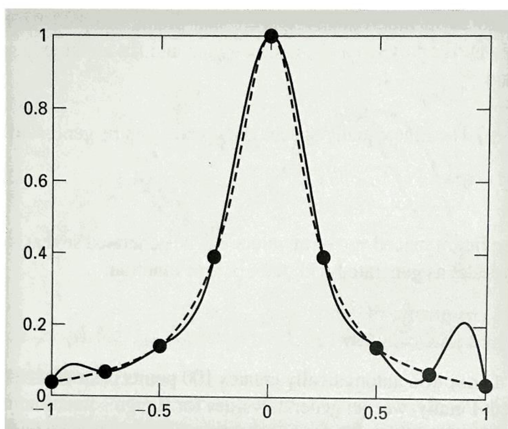


FIGURE 18.7
Comparison of Runge's function (dashed line) with a 9-point clamped end spline fit generated with MATLAB (solid line). Note that first derivatives of 1 and −4 are specified at the left and right boundaries, respectively.

Figure 2.12: Runge's function with nine data points constructed in MATLAB by using a clamped end condition (Chapra [2011], p. 446).

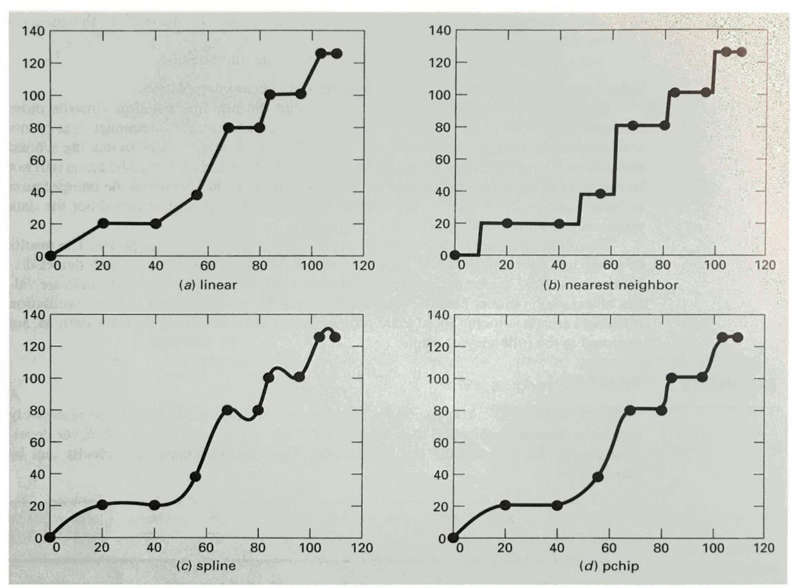


FIGURE 18.8
Use of several options of the `interp1` function to perform piecewise polynomial interpolation on a velocity time series for an automobile.

Figure 2.13: Discrete data fitted with (a) a first degree spline (b) a zero degree spline (c) a standard spline (d) an Hermite spline (Chapra [2011], p. 448).

Figure 2.14: Timothy Sauer, author of the book *Numerical Analysis*, 2012.

1. The `spline` function implements cubic spline interpolation. The author describes the syntax and explains how to choose different end conditions. He then uses Runge's function, already mentioned in the previous chapter (example 17.7), to show how the use of different cubic splines remove the oscillations caused by the Runge phenomenon. The results using different end conditions are presented in Figures 2.11 and 2.12.

2. The function `interp1` implements spline and Hermite interpolation and also other types of piecewise interpolation. Here is an easy way to implement "a number of different types of piecewise one-dimensional interpolations" with a given general formula. There are four different optional functions: `nearest` (for nearest neighbor interpolation), `linear` (for linear interpolation), `spline` (for piecewise cubic spline), and `pchip` (for piecewise cubic Hermite interpolation). Finally, an example is presented to illustrate the trade-offs of using `interp1`, as shown in Figure 2.13.

After having presented the subject of *cubic splines* in detail the author solves a particular real-life heat transfer problem by using cubic splines. The solution is presented in great detail ([Chapra, 2011], p. 452–456).

2.2.2 Book II: Numerical Analysis by Timothy Sauer, 2012.

Timothy Sauer is professor of mathematics at George Mason University. He specializes in dynamical systems.

This book is intended for students of engineering, science, mathematics, and computer science. The book offers an introductory level with some more advanced level topics, and uses MATLAB for its computer examples. Sauer's book is used as textbook in many countries. For example, Washington State University used this book as textbook in the course *Numerical Analysis* during spring 2017 ([Washington State University, 2017]). Lund University in Sweden used it as textbook for its course Numerical Analysis for Computer Science during spring 2017 ([University, 2017]). At Harvard university in Cambridge, Massachusetts, it was used in the 2012 course Numerical Analysis ([Harvard University, 2012]). The topic of *cubic splines* covers a 14-page section of the chapter on Iterpolation.

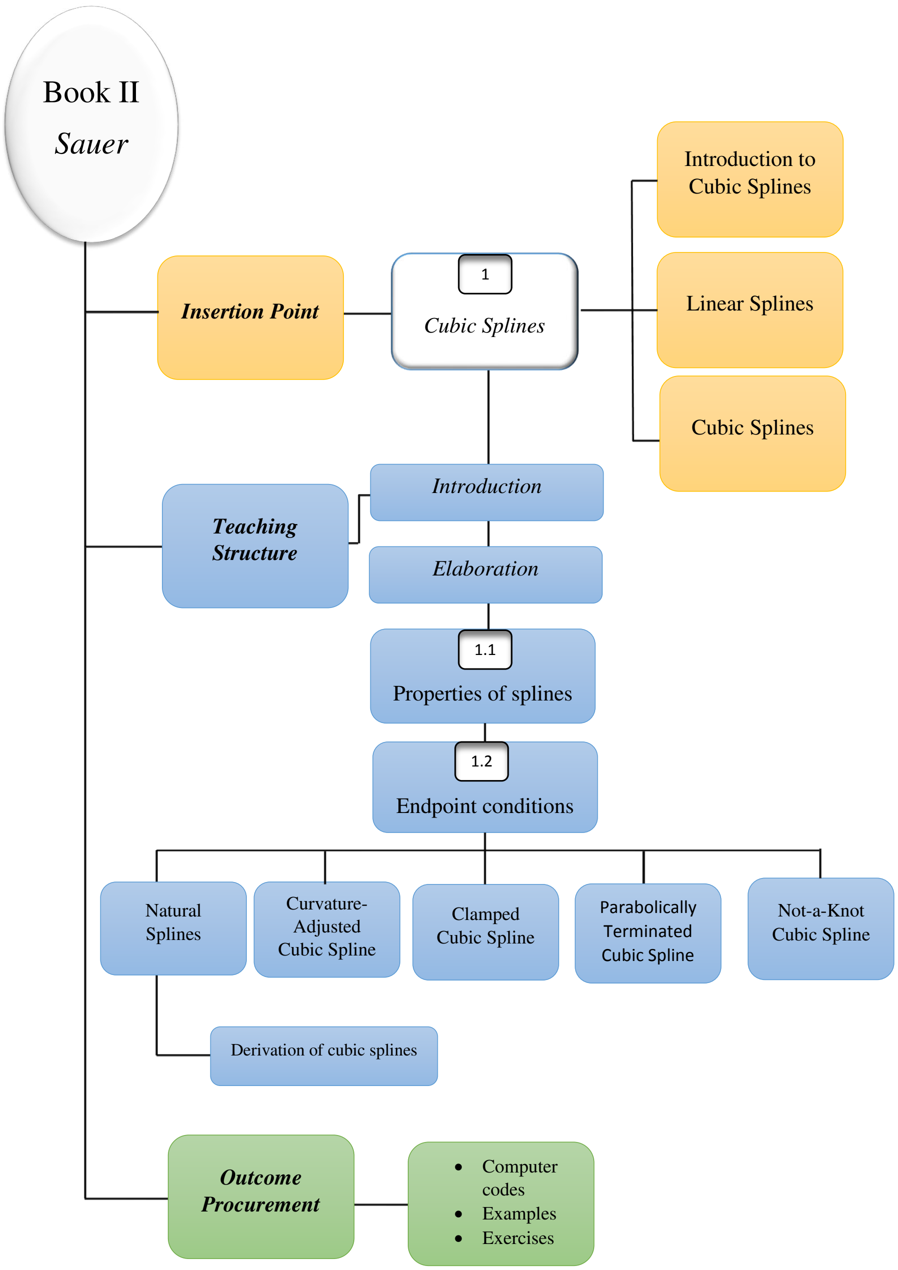


Figure 2.15: Schematic content and organization of Chapter 3, Section 4,“Interpolation – Cubic Splines” in Sauer’s book Sauer [2012].

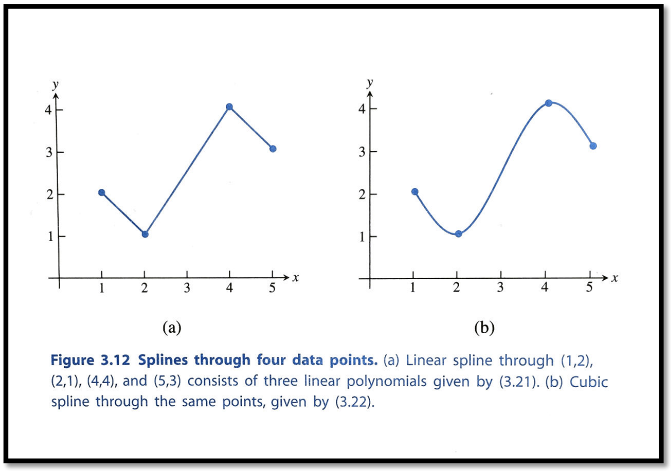


Figure 2.16: (a) Linear spline interpolation (b) cubic spline interpolation (Sauer [2012], p.166).

2.2.2.1 Insertion point

In Sauer's book cubic splines are presented as a section of the chapter titled *Interpolation*, after the sections titled *Data and Interpolating Functions*, *Interpolation Error* and *Chebyshev Interpolation*. After cubic splines there is a section on Bézier curves. In the chapter *Interpolation Error* the author displays the famous *Runge phenomenon* and uses it to motivate Chebyshev interpolation, but does not mentioned it in the section for cubic splines. Instead, the author asserts that "splines represent an alternative approach to data interpolation" (Sauer [2012]), and the idea of splines is to "use several formulas, each with a low-degree polynomial, to pass through the data points" (Sauer [2012]). The first example of these ideas is given by providing an example (four data points) together with a solution that uses the child game concept of "connecting the dots", shown in Figure 2.16. The definition of *linear splines* is introduced only by working out the formulas for this particular example.

The lack of smoothness of *linear splines* is pointed out, and this fact allows the author to introduce *cubic splines*: "cubic splines are meant to address this shortcoming of linear splines" (Sauer [2012]). We are given the equations that define a cubic spline passing through the same given data points, and shown the difference between linear and cubic splines in Figure 2.16. At this point the notion of *knots* is introduced. The author also mentions an important difference between linear splines and cubic splines:

1. There is obviously one and only one linear spline passing through n data points.

2. There are infinitely many cubic splines passing through n data points. Consequently, interpolation cubic splines need extra conditions to be uniquely defined: "extra conditions will be added when it is necessary to nail down a particular spline of interest" (Sauer [2012]).

In the section that follows, Sauer expands on the properties of cubic splines.

2.2.2.2 Teaching structure

The reader is firstly introduced to the basic idea of "connecting the dots" (Sauer [2012]) and is motivated to the introduction of cubic splines. The author then elaborates the topic with precise definitions and properties of cubic splines. Finally, end conditions are introduced, and this leads to the discussion of several different types of cubic splines.

Introduction

The author starts by giving a precise definition of cubic splines. First, he gives the general form of a cubic spline passing through the points $(x_1, y_1), \ldots, (x_n, y_n)$:

$$S_{i-1}(x) = y_{i-1}+b_{i-1}(x-x_{i-1})+c_{i-1}(x-x_{i-1})^2+d_{i-1}(x-x_i-1)^3 \quad x \in [x_{i-1}, x_i], \quad i = 1, \ldots, n-1$$

Then, he gives three properties that a cubic spline must satisfy:

1. **Property 1**: $S_i(x_i) = y_i$ and $S_i(x_{i+1}) = y_{i+1}$ for $i = 1, ..., n-1$.
2. **Property 2**: $S_{i-1}'(x_i) = S_i'(x_i), \quad i = 2, ..., n-1$.
3. **Property 3**: $S_{i-1}''(x_i) = S_i''(x_i), \quad i = 2, ..., n-1$.

The first property is linked to interpolation, while the other two are linked to continuity.

Sauer ends his initial presentation by studying the numerical example he gave in the introduction of the topic, and demonstrates that it satisfies the stated properties.

Elaboration

To show how to construct a cubic spline from a set of data points, the author starts by counting the number of conditions and the number of unknown coefficients. He finds the system of equations is underdetermined, because there are two more unknowns than equations, and states that it is important to arrive at a system containing as many equations as unknowns. It is therefore necessary to add two extra conditions.

Sauer states that while there is an infinite number of ways to add two equations to the system, the easiest way is to require the spline $S(x)$ to have an inflection point at each end of the defining interval $[x_1, x_n]$. Thus, a new property is added:

- **Property 4a**: $S_1''(x_1) = 0$ and $S_{n-1}''(x_n) = 0$.

Splines that satisfy this last condition are called *natural splines*. The conditions that define a spline can be written as a linear system of n equations in n unknowns. These unknowns are the coefficients of the quadratic terms of the spline. The resulting system matrix is strictly diagonally dominant, which means that the system has a unique solution, as stated in the following theorem (Sauer [2012], p. 107):

> **Theorem 2.10**: If the $n \times n$ matrix A is strictly diagonally dominant, then A is a nonsingular matrix.

Once these coefficients are computed, the remaining ones may be calculated by explicit formulas derived from the continuity and interpolation conditions. Then, the author complements the explanation with a pseudo code that calculates the coefficients of a natural cubic

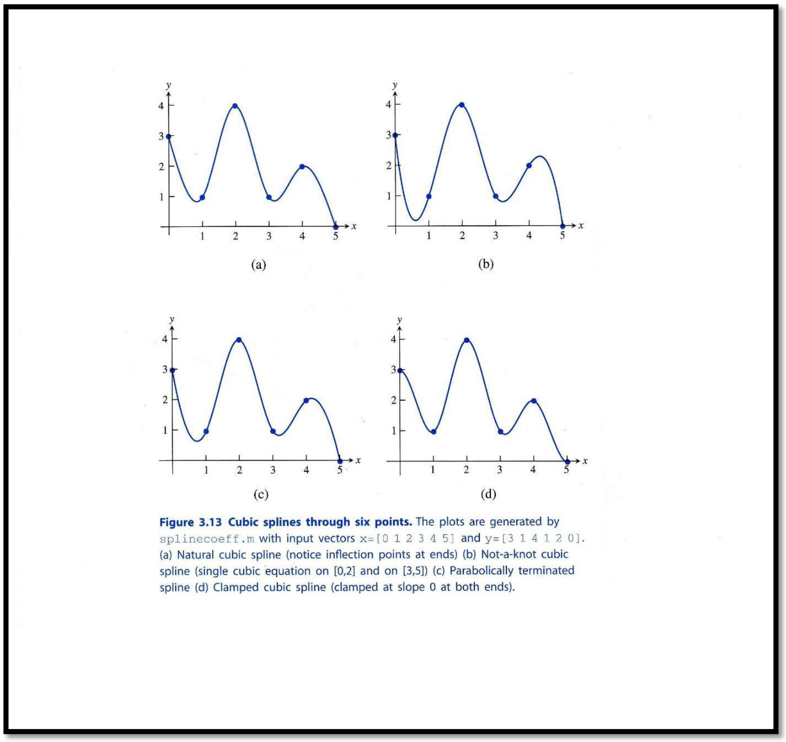


Figure 2.17: As shown above four types of end conditions are demonstrated: (a) Natural cubic spline (b) Not-a-Knot cubic spline (c) Parabolically terminated cubic spline (d) Clamped cubic spline [Sauer, 2012] p.173.

spline, given a set of interpolation points (knots). The use of this code is then illustrated by the calculation of a cubic spline passing through three given points. Finally, he presents a MATLAB code that generates a natural cubic spline and that may be modified to include several other types of cubic splines by choosing different end conditions. This is a topic that will be elaborated in the next section.

The author dedicates a new subsection to the analysis of end conditions. He justifies the addition of end conditions by arguing the need of a square system of equations for the determination of the spline coefficients. The first end condition led to *natural splines*.

Endpoint conditions

"There are several other ways to add two more conditions" (Sauer [2012]) the author has stated. This observation may support the choice of other options. As *natural splines* conditions occur at the left and right ends of the spline, they are called *end conditions*. The author notes that there can be many versions of Property 4a, that is, many different sets of end conditions. The most popular ones are presented.

Property 4a: Natural spline. A cubic interpolatory spline $S(x)$ is called *natural cubic splines* if

$$S_1''(x_1) = 0 \quad , \quad S_{n-1}''(x_n) = 0.$$

The author already mentions *natural splines* in the previous section, after presenting the properties of cubic splines, by adding *natural splines* as a fourth property, "Property 4a".

The follows theorem states the uniqueness of a natural cubic spline:

Theorem 3.7: Let $n \geq 2$. For a set of data points $(x_1, y_1), \ldots, (x_n, y_n)$ with distinct x_i, there is a unique *natural* cubic spline fitting the points.

Property 4b: Curvature-adjusted cubic spline.

A cubic interpolatory spline $S(x)$ is called a *curvature-adjusted* cubic spline if

$$S_1''(x_1) = a \quad , \quad S_{n-1}''(x_n) = b.$$

In the first alternative to natural cubic splines, the user chooses the values of the second derivatives at the endpoints. The author observes that this set of conditions possesses the same qualities as those of the natural spline. He shows how the MATLAB code has to be modified to include this type of cubic spline. He also explains that the new spline coefficient matrix has the same structure as the corresponding matrix for natural splines (the matrix is strictly diagonally dominant).

Property 4c: Clamped cubic spline.

A cubic interpolatory spline $S(x)$ is called a *clamped* cubic spline if

$$S_1'(x_1) = a \quad \text{and} \quad S_{n-1}'(x_n) = b$$

where a, b are arbitrary values. This is similar to the previous Property 4b, but the first derivative is substituted for the second derivative. The author mentions the important point "the slope at the beginning and end of the spline are under the user's control" [Sauer, 2012]. He shows how to replace the MATLAB code with the conditions for the clamped cubic spline and he points to Figure 2.17 to show a plot of a spline interpolation when $a = 0$ and $b = 0$. He points out that *Theorem 3.7* also holds because the "strict diagonal dominance holds also for the revised coefficient".

Property 4d: Parabolically terminated cubic spline.

A cubic interpolatory spline $S(x)$ is called a *parabolically terminated* cubic spline if the first spline polynomial S_1 and the last spline polynomial S_{n-1} are forced to be at most of degree 2. The author claims that when these conditions are applied the system has a strictly diagonally dominant matrix if the dimension of the system is reduced by replacing c_1 by c_2 and c_n by c_{n-1}. Finally, the author shows how to modify the MATLAB code to include the parabolically terminated spline conditions.

Property 4e: Not-a-Knot cubic spline.

A cubic interpolatory spline $S(x)$ is called a *not-a-knot cubic spline* if

$$S_1^{'''}(x_2) = S_2^{'''}(x_2) \qquad \text{and} \qquad S_{n-2}^{'''}(x_{n-1}) = S_{n-1}^{'''}(x_{n-1}).$$

Since $S_1(x)$ and $S_2(x)$ are polynomials of degree 3 or less, the author explains the requirements of *not-a-knot cubic spline* interpolation are equivalent to eliminating the second and the next-to-last knots. "Requiring their third derivatives to agree at x_1, while their zeroth, first, and second derivatives already agree there, causes S_1 and S_2 to be identical cubic polynomials" (Sauer [2012]). Therefore, he notes that x_2 is not needed as a base point and that the spline gives the same formula $S_1 = S_2$, in all of $[x_1, x_3]$. But not only is x_2 not need as a base point, but also x_{n-1}. This is evidently no longer a knot. The author explains how the not-a-knot cubic spline conditions can be inserted in the MATLAB code and he points to Figure 2.17 where we can observe a plot of a not-a-knot spline and compare it with that of a natural spline. Finally, the author states Theorem 3.8, which is similar to Theorem 3.7. These Theorems state that a unique solution for each end conditions exists:

Theorem 3.8: Assume that $n \geq 2$. Then, for a set of data points $(x_1, y_1), \ldots\ldots, (x_n, y_n)$ and for any one of the end conditions given by Properties 4a-4c, there is a unique cubic satisfying the end conditions and fitting the points. The same is true assuming that $n \geq 3$ for Property 4d and $n \geq 4$ for Property 4e.

The section is closed by a short explanation on MATLAB's `spline` command, which uses a not-a-knot cubic spline as a default.

2.2.2.3 Outcome Procurement

The previous section is followed by a list of exercises and computer problems.

The list of exercises includes theoretical as well as more practical questions, such as checking whether a set of equations defines a cubic spline, or finding a particular type of spline that interpolates a set of given points.

In the computer problems section the reader is asked to find and plot a spline, given a set of interpolation points and end conditions.

2.2.3 Book III: Numerical Analysis: Mathematics of Scientific Computing, by David Kincaid and Ward Cheney, 2002.

David Kincaid was professor at the Department of Computer Science and at the Institute for Computational Engineering and Sciences at the University of Texas at Austin. His main subjects were iterative methods and parallel computing. Ward Cheney was a professor at the Department of Mathematics at the University of Texas at Austin. He specialized in numerical analysis and approximation theory.

This book is directed to students with diverse backgrounds, and at a more advanced level. Kincaid and Cheney's book has been used as textbook for higher level courses at several universities. For example, the course *Mathematical Numerical Analysis* at Duke University in Durham, North Carolina, used this book during spring 2015 ([Duke University, 2015]). Also,

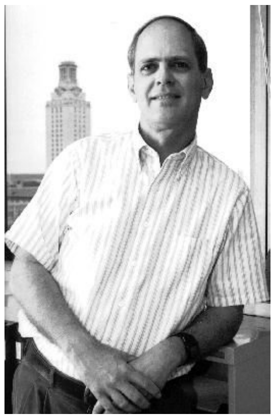


Figure 2.18: David Kincaid

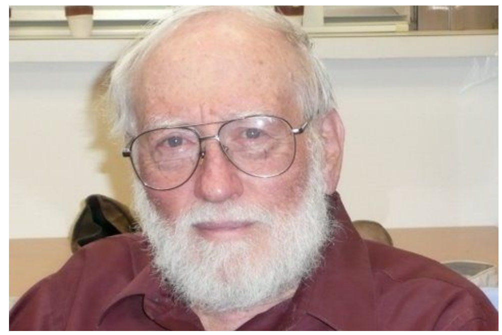


Figure 2.19: Ward Cheney

Figure 2.20: The authors of Numerical Analysis: Mathematics of Scientific Computing, 2002.

this book was used in the course *Numerical Analysis and Differential Equation* during fall 2016 at Cornell University in Ithaca, New York, and in the course *Introduction to Numerical Analysis: Approximation and Nonlinear Equations* in 2014 at University of California San Diego. The cubic spline section in this book is presented at a more advanced level than the other books studied here. The explanations are quite general and only pseudo code is used to describe computer programs. The section on cubic splines is eight pages long.

In Figure 2.21 we give a schematic organization of the chapter on *cubic splines* in Kincaid and Cheney's book.

Chapter 6 of this book is titled *Approximating Functions.* The specific objective of this chapter is to present different types of functions that can be used to approximate other functions or data. The chapter starts with the sections *Polynomial Interpolation*, *Divided Differences*, *Hermite Interpolation* and *Spline Interpolation.* Each of these sections is mostly self-contained. The authors start the section of *Spline Interpolation* directly, without any motivation. Instead, they give a formal definition of a *spline function of degree* k [Kincaid and Cheney, 2002].

To introduce the notion of cubic splines, the authors give the definition of a general spline and then they describe low-degree splines. Then cubic splines are derived, and some theoretic results on optimality are discussed. Then, *tension splines* are described, and finally higher-degree splines are discussed.

2.2.3.1 Insertion point

Kincaid and Cheney start with definition of a spline function of degree k, defining *knots* and the continuity conditions.

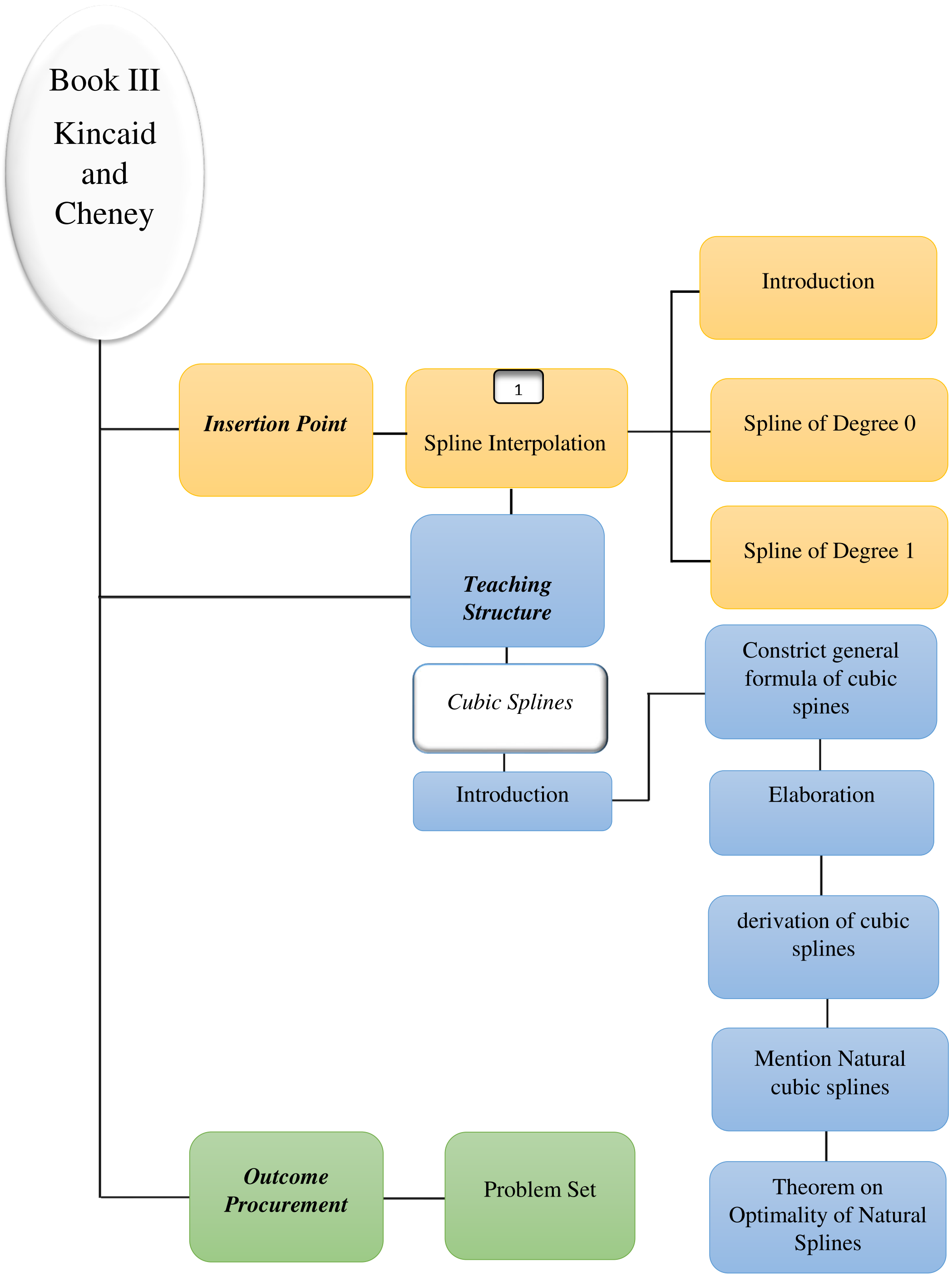


Figure 2.21: Content and organization of Chapter 6, Section 4,"Spline Interpolation" Kincaid and Cheney [2002].

1. S is a continuous piecewise polynomial of degree at most k on each subinterval $[t_0, t_n]$.

2. S has continuous derivatives of all orders up to $k-1$.

No interpolation conditions are given, so that the splines are free to interpolate not only at the knots, but at any given points.

The particular definitions of *splines of degree 0* and *splines of degree 1* are given, together with graphical examples. They do not mention quadratic splines.

The authors present splines of degree 0 as “piecewise constants”,

$$S_{i-1}(x) = c_{i-1} \quad x \in [t_{i-1}, t_i) \quad , i = 1, ..., n \tag{2.3}$$

The authors remark that the intervals for these splines do not intersect each other and so there is no ambiguity in their definition at the knots. To clarify the case he offers a simple example by looking at a spline of degree 0 with six knots, shown in Figure 2.22:

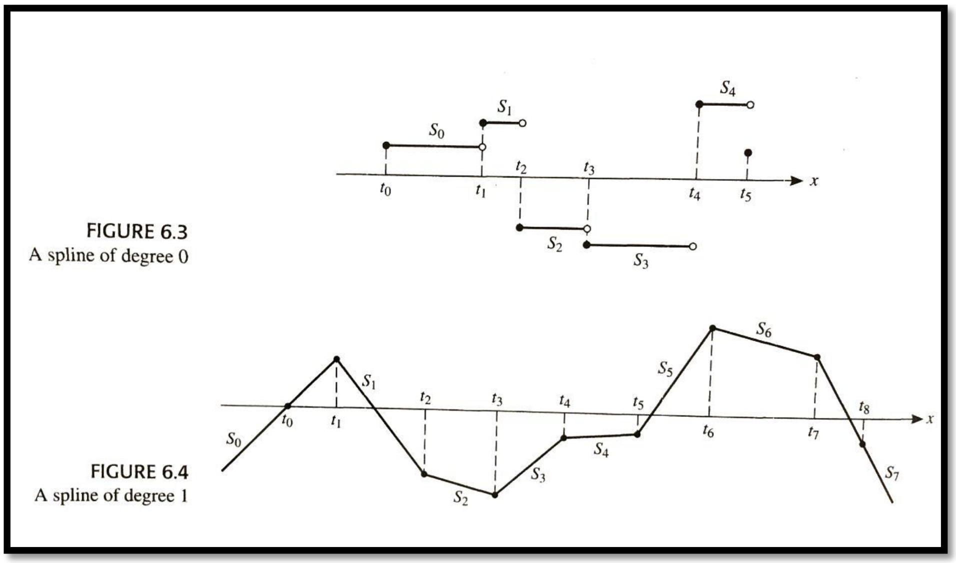


Figure 2.22: Figure 6.3 shows the spline of degree 0 with six knots and Figure 6.4 shows the spline of degree 1 with nine knots ([Kincaid and Cheney, 2002, p.350]).

For splines of degree 1 the authors start by giving Figure 2.22. The linear splines function formula is defined by:

$$S_{i-1}(x) = a_{i-1}x + b_{i-1}, \quad x \in [t_{i-1}, t_i) \tag{2.4}$$

$$S(x) = \begin{cases} S_0(x) = a_0 x + b_0 & x \in [t_0, t_1) \\ S_{i-1}(x) = a_{i-1}x + b_{i-1} & x \in [t_{i-1}, t_i) \\ S_{n-1}(x) = a_{n-1}x + b_{n-1} & x \in [t_{n-1}, t_n] \end{cases} \tag{2.5}$$

In the following steps, they discuss some issues, namely,

1. How can a spline of degree 1 be evaluated?

2. How can express the function S on $(\infty, t_1]$?

3. How can we express the function S on $[t_{n-1}, \infty)$?

Finally, the authors present a pseudocode to evaluate a linear spline at a point x when the knots, the coefficients, and the number of subintervals are given. Figure 2.23 shows this code.

```
input (t_i), (a_i), (b_i), x, n
for i = 1 to n − 1 do
      if x < t_i then
            S(x) = a_{i−1} x + b_{i−1}
            output S(x)
            exit loop
      end if
end do
S(x) = a_{n−1} x + b_{n−1}
output S(x)
```

Figure 2.23: A pseudocode to evalauate a *linear spline* function definition by known t_i and coefficients (a_i, b_i) (Kincaid and Cheney [2002], p.350).

2.2.3.2 Teaching structure

Introduction

The notion of cubic splines is introduced by emphasizing that these are the splines most often used in practice.

The authors motivate the topic by giving a table containing a set of data points:

x	t_0	$\cdots$	t_1	t_n
y	y_0	$\cdots$	y_1	y_n

A cubic spline that interpolates the data is to be constructed. [Kincaid and Cheney, 2002].

What follows is a step-by-step explanation of how the authors develop cubic splines. It must be noted that the authors assume that the interpolation points are the same as the knots.

1. The authors describe the spline by its polynomial pieces:

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1] \\ S_1(x) & x \in [t_1, t_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases} \tag{2.6}$$

   where each S_i is a cubic polynomial on $[t_i, t_{i+1}]$, and the polynomials interpolate the data points so that

$$S_{i-1}(t_i) = y_i = S_i(t_i) \quad (1 \leq i \leq n-1)$$

2. The authors discuss the question: Does the continuity requirements on S, $S^{'}$, and $S^{''}$ as well as the interpolation conditions provide enough conditions to define a *cubic spline*? Altogether these are $4n-2$ conditions for determining $4n$ unknown coefficients. Therefore, there are two degrees of freedom, "and various ways of using them to advantage" [Kincaid and Cheney, 2002], the authors say.

Elaboration

The authors proceed to the derivation of cubic splines immediately after the introduction. When considering end conditions they only point towards *natural end conditions*, describing it as an "excellent choice". No other end condition is mentioned. Now the authors explain the derivation of cubic splines and their end conditions.

Derivation of cubic splines

For the derivation of cubic splines the authors use two conventions,

1. The unknowns are defined as $z_i := S^{''}(t_i), \quad i = 0, ..., n$. They are used to enforce the continuity condition for second derivative of a cubic spline.

2. The authors use the notation $h_i := t_{i+1} - t_i$.

As the second derivative of a cubic polynomial $S_i^{''}$ is a linear function, and because $S_i^{''}(t_i) = z_i$ and $S_i^{''}(t_{i+1}) = z_{i+1}$, it can be written in terms of Lagrange polynomials as

$$S_i^{''}(x) = \frac{z_i}{h_i}(t_{i+1} - x) + \frac{z_{i+1}}{h_i}(x - t_i) \tag{2.7}$$

Integrating twice gives

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1} - x)^3 - \frac{z_{i+1}}{6h_i}(x - t_i)^3 + C(x - t_i) - D(t_{i+1} - x) \tag{2.8}$$

with integration constants C and D. Next, the authors show how these can be determined from interpolation and continuity conditions. Firstly, $S_i(t_i) = y_i$ is imposed which gives

$$D = \frac{y_i}{h_i} - \frac{z_i h_i}{6}.$$

Secondly, $S_i(t_{i+1}) = y_i$ is imposed yielding

$$C = \frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6}.$$

We observe that if the z_i were known, then C and D would be known too, and

$$S_i(x) = \frac{z_i}{6h_i}(t_{i+1}-x)^3 + \frac{z_{i+1}}{6h_i}(x-t_i)^3 + (\frac{y_{i+1}}{h_i} - \frac{z_{i+1}h_i}{6})(x-t_i) + (\frac{y_i}{h_i} - \frac{z_ih_i}{6})(t_{i+1}-x) \quad (2.9)$$

To obtain Equation (2.9) the authors used two conditions:

- $\lim_{x\downarrow t_i} S^{''}(x) = z_i = \lim_{x\uparrow t_i} S^{''}(x)$, which is the continuity condition for $S^{''}$.
- $S_i(t_i) = y_i$, $S_i(t_i+1) = y_{i+1}$, which refers to $S_i(x)$ is continuous function when $x = t_i$ and $x = t_{i+1}$

What about the continuity of first derivative $S^{'}(x)$? The authors observe that "for $S^{'}$ at the interior knots t_i, we must have $S^{'}_{i-1}(t_i) = S^{'}_i(t_i)$" [Kincaid and Cheney, 2002], and so,

$$h_{i-1}z_{i-1} + 2(h_i+h_{i-1})z_i + h_iz_{i+1} = \frac{6}{h_i}(y_{i+1}-y_i) - \frac{6}{h_{i-1}}(y_i-y_{i-1}), \quad i = 1, ..., n-1. \quad (2.10)$$

As these equations result in an underdetermined system, we need to add two more conditions. One choice, $z_0 = z_n = 0$, results in what is called a *natural cubic spline*. Writing the linear system results in a matrix that is a tridiagonal and diagonally dominant. The authors show now to solve this system by *Gaussian elimination* adapted to this structure. The following Figure 2.24 shows the pseudo code for the *Gaussian elimination* process to solve this linear system:

The details of this pseudo code are then examined. The authors clarify what is meant by t_i, y_i and z_i. Then the authors point to the divisions by u_i in the algorithm, and show by induction that these quantities are never zero. Next, the author clarify any value of the cubic spline can be computed in formula (2.6) after the coefficients have been determined on the intervals. In addition, they write the equation (2.9) "a more efficient nested" [Kincaid and Cheney, 2002] by writing the formula (2.9) in a simple way with given the value of the coefficients A_i, B_i and C_i:

$$S_i(x) = y_i + (x-t_i)[C_i + (x-t_i)[B_i + (x-t_i)A_i]] \quad (2.11)$$

Finally, the authors encourage the reader to write subprogram or procedure for this procedure.

To show a simple computation with these routines, the authors solve an example with a particular function and show that the error is zero at each knot because of the properties of cubic spline ($f(t_i) = S(t_i)$). They also show the interpolation errors at 37 equally spaced points.

At this stage, the authors state a theorem where we can see that natural cubic splines produce the smoothest interpolating function.

Theorem 1 (Theorem on Optimality of Natural Cubic Spline):

```
input n, (t_i), (y_i)
for i = 0 to n − 1 do
        h_i ← t_{i+1} − t_i
        b_i ← 6(y_{i+1} − y_i)/h_i
end do
u_1 ← 2(h_0 + h_1)
v_1 ← b_1 − b_0
for i = 2 to n − 1 do
        u_i ← 2(h_i + h_{i−1}) − h_{i−1}^2/u_{i−1}
        v_i ← b_i − b_{i−1} − h_{i−1}v_{i−1}/u_{i−1}
end do
z_n ← 0
for i = n − 1 to 1 step − 1 do
        z_i ← (v_i − h_i z_{i+1})/u_i
end do
z_0 ← 0
output (z_i)
```

Figure 2.24: The procedure to solve the tridiagonal system by *Gaussian elimination* algorithm (Kincaid and Cheney [2002], p.353).

Let $f^{''}$ be continuous in $[a, b]$ and let $a = t_0 < t_1 < .... < t_n = b$. If S is the natural cubic spline interpolating f at the knots t_i for $0 \leq i \leq n$ then

$$\int_a^b [S^{''}(x)]^2 dx \leq \int_a^b [f^{''}(x)]^2 dx$$

The proof of *Theorem 1* is included in the book. This ends the presentation of *cubic splines*. The next part of this section is the *Tension Splines* section, which will not be covered in this thesis. Further information about the *Tension spline* section can be found in the appendix Kincaid and Cheney [2002].

2.2.3.3 Outcome Procurement

The authors have already set some questions and challenges along their development of the theory, as stated above. To complement this, the authors give a list of problems. The Problems section includes these questions posed along the derivation of cubic splines. We observe there is also a question about the determination of whether a given function is a quadratic spline function [Kincaid and Cheney, 2002], but quadratic splines were not mentioned in the previous *spline interpolation* section. The rest of the questions are about *cubic splines* and *natural cubic splines*. The author also includes some computer problems in a separate list. The questions include the proof and numerical test of a formula. The first four questions are about cubic splines and the rest are outside our topic.

Figure 2.25: Laurene V. Fausett, author of Applied Numerical Analysis Using MATLAB, 2008.

2.2.4 Book IV: Applied Numerical Analysis Using MATLAB by Laurene V. Fausett, 2008.

Laurene V. Fausett was a math professor at A&M University in Commerce, Texas. She specialized in applied numerical analysis and neural networks.

This book focuses on the application of numerical analysis and numerical method using MATLAB. The Interpolation chapter in this book covers *cubic splines* under the more general topic of Piecewise Polynomial Interpolation. This section is thirteen pages long. This book was in use at different universities around the globe. For instance, it was used in the online course *MATLAB Programming for Numerical Computation* as textbook in 2016. The course was given by the Indian National Programme on Technology Enhanced learning [A project by the government of India, 2017], intended for engineers who need to learn how to do scientific computations. The University of Liverpool, England used Fausett'a book as second reference in the course *Numerical and Statistical Analysis for Engineering with Programming*, 2010-2011. Also, the University of Arizona in Tulsa has been using this book for its course Numerical Analysis for Civil Engineers.

In Figure 2.26 we show a scheme of how the author presents the section *Cubic Splines* in this book.

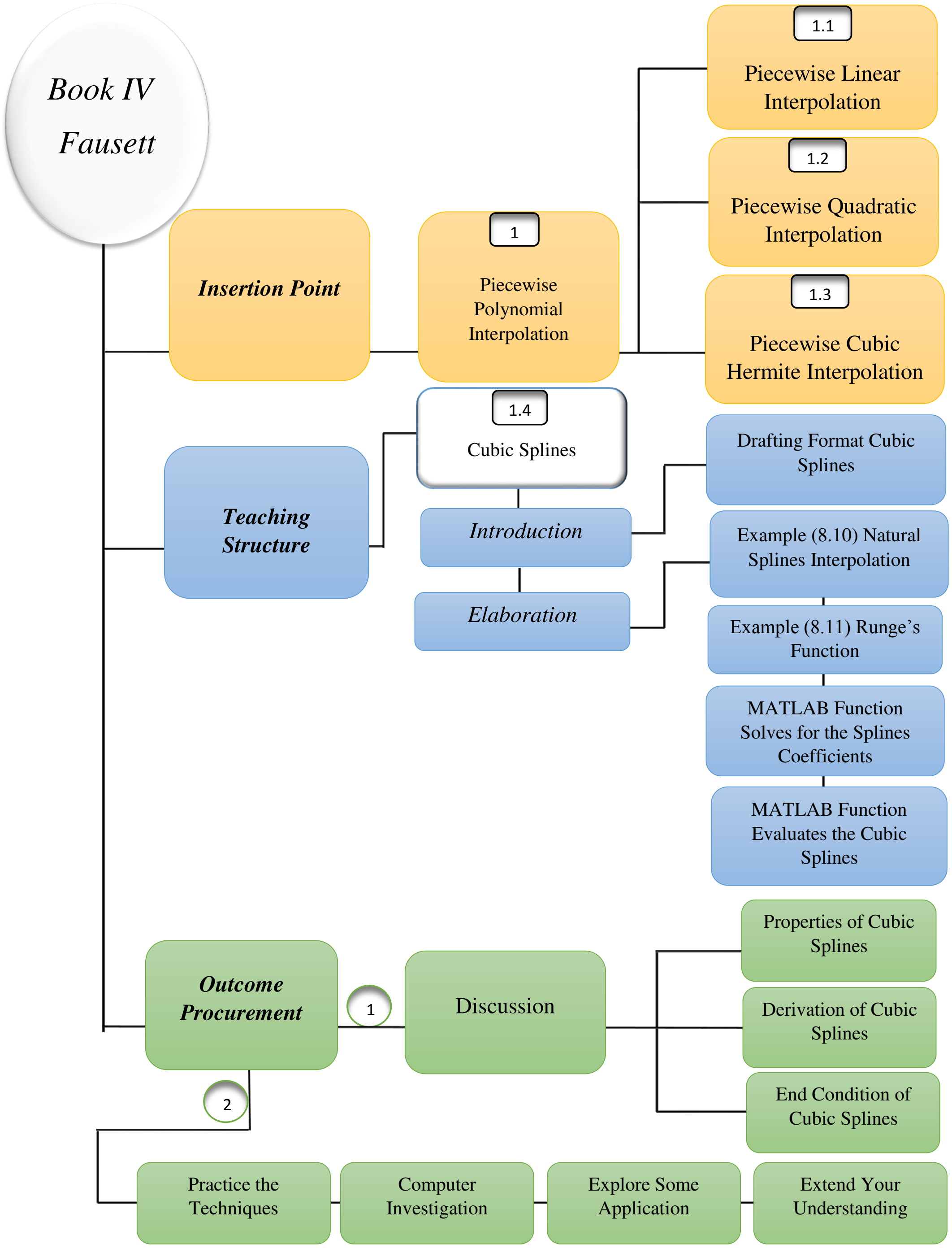


Figure 2.26: Content and organization of Chapter 8, *Interpolation*, section *Piecewise polynomial interpolation* (8.3). Fausett [2008].

2.2.4.1 Insertion Point

Chapter 8 in this book is called *Interpolation.* In the first section of this chapter, the author presents the notion of *polynomial interpolation* introducing Lagrange Interpolation and Newton Interpolation. In the second section Hermite interpolation is introduced. After this she demonstrates the effect of higher degree polynomial interpolation by considering Runge's function. This example shows that increasing the degree of an interpolating polynomial does not necessarily reduce the interpolation error. Instead, high oscillations occur at the endpoints of the interval. This effect gets larger when the number of interpolation points is increased. This is the motivation for introducing piecewise polynomial interpolation in the next section.

In the section *Piecewise Polynomial Interpolation*, the author retakes the disadvantage of high degree interpolation that has been observed in the Runge's function. To avert the problem we can use piecewise polynomials. She motivates the section also by a reference to the historical use of splines as "the design and construction of a ship or aircraft involved the use of full-size models" [Fausett, 2008]. The author refers to the history to show why we might study how to construct a smooth curve and why. Additionally, she explains the details of how spline curves were constructed in the past. The book mentions the relation of the mathematical spline to a bent beam in physics, which takes the shape of a cubic piecewise polynomial. Then, the author presents a spline in a mathematical way by mentioning different types of piecewise polynomial interpolation of degree m. She gives a brief summary for each type of spline that will be mentioned in the next sections:

1. Linear splines, which are continuous functions.

2. Quadratic splines, which have also continuous derivatives.

3. Piecewise cubic Hermite interpolation, which "is useful [...] for shape preserving" [Fausett, 2008]. To program Hermite interpolation in MATLAB the built-in Matleb function `pchip` is described.

She explains the differences between nodes and knots because she distinguishes the two cases. Firstly, she treats the case "nodes=knots", and afterwards, "nodes $\neq$ knots". In her definition, nodes are the points where spline segments meet and knots are the points where interpolation conditions are imposed.

Finally, after having introduced the details about the different kinds of splines, the author draws attention to cubic spline interpolation, mentioning its use in various mathematical problems. To illustrate, a plot of cubic spline interpolation vs. polynomial interpolation is given in Figure 2.27.

Piecewise Linear Interpolation

The author explains linear spline interpolation as the simplest example of piecewise interpolation. The explanation starts by considering a set of data points (x_i, y_i) and includes the details of construction of linear spline interpolation. The description is done for a set of four data points. Then she shows the formula of this spline in terms of Lagrange polynomials. In more general terms, the formula reads

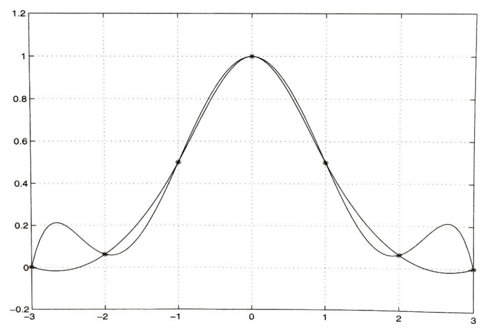


FIGURE 8.18: Polynomial and spline interpolation.

Figure 2.27: Cubic spline and polynomial interpolation for Runge's function (Fausett [2008], p.299).

$$P(x) = \begin{cases} \frac{x-x_2}{x_1-x_2}y_1 + \frac{x-x_1}{x_2-x_1}y_2 & x \in [x_1, x_2) \\ \\ \frac{x-x_{i+1}}{x_i-x_{i+1}}y_i + \frac{x-x_i}{x_{i+1}-x_i}y_{i+1} & x \in [x_i, x_{i+1}) \\ \\ \frac{x-x_n}{x_{n-1}-x_n}y_{n-1} + \frac{x-x_{n-1}}{x_n-x_{n-1}}y_n & x \in [x_{n-1}, x_n] \end{cases} \tag{2.12}$$

In addition, the author mentions the advantage that a linear spline interpolation function is continuous and the disadvantage that it has no further smoothness at the knots. Finally, she presents an example of piecewise linear interpolation of four data points Figure 2.28:

Piecewise Quadratic Interpolation

The author introduces quadratic splines by explaining the construction of a piecewise quadratic function for $n+1$ data points on n intervals. By counting the unknowns of the formula and the equation for each interval it can be seen that there are $3n-1$ equations and $3n$ unknowns. Therefore she concludes that one additional condition is needed in order to have a solvable system. She points to several alternatives to meet this condition and presents an approach that makes it particularly easy to construct quadratic splines. The form of this spline is the following:

$$S_i(x) = y_i + z_i(x - x_i) + \frac{z_{i+1} - z_i}{2(x_{i+1} - x_i)}(x - x_i)^2, \quad \text{on} \quad [x_i, x_{i+1}) \tag{2.13}$$

where z_i is the slope at x_i of the function $S_i(x)$. The author explains:

1. The continuity conditions for the first derivative at the nodes of the quadratic spline formula (2.13) are automatically enforced.

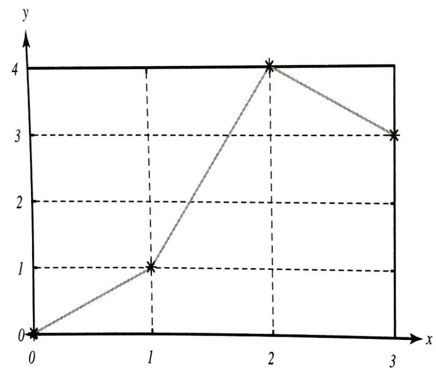


FIGURE 8.19: Piecewise linear interpolation.

Figure 2.28: Linear piecewise interpolation of the data $x = [0, 1, 2, 3]$ and $y = [0, 1, 4, 3]$ ([Fausett, 2008, p. 300]).

2. The interpolation and continuity conditions for the quadratic spline at the knots must also be enforced.

3. The slope z_i can be obtained by using continuity property of the function at the nodes. The slopes are found to satisfy

$$z_{i+1} = 2\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - z_i \tag{2.14}$$

   Thus, specifying z_1 will render the other slopes, $z_2, \ldots, z_n$.

4. By giving four data points $(x_i, y_i), i = 1, 2, 3, 4$, and computing the coefficients of the quadratic spline by using the slopes z_i with the condition "nodes=knots", the author shows that the value of the slope at x_1 has considerable influence on the resulting curve.

5. Another alternative would be to choose the knots to be the midpoints of the nodes. In this case ("nodes$\neq$knots") we obtain six equations for eight unknowns, meaning the value for two of the slopes must be given. The author says this is a "more balanced" [Fausett, 2008] approach, as there can be end conditions at each end of the interval. The slopes z_i are, in this case,

   - $z_1 = x_1$
   - $z_i = \frac{x_{i-1}+x_i}{2}$
   - $z_{n+1} = x_n$

   If the data points are spaced evenly we may define $h_i = x_{i+1} - x_i$ when $i = 1, ..., n$, and then

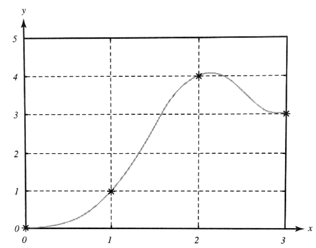


FIGURE 8.20: Piecewise quadratic interpolation of four data points.

Figure 2.29: By considering the data points $(0, 0), (1, 1), (2, 4), \text{and}(3, 3)$ and solving a linear system using MATLAB we get a piecewise quadratic interpolation function for the given data points. ([Fausett, 2008, p. 303]).

- $z_2 - x_1 = \frac{h_1}{2}$
- $z_i - x_{i-1} = \frac{h_{i-1}}{2}$
- $z_n - x_{n-1} = \frac{h_{n-1}}{2}$

and

- $z_2 - x_2 = \frac{-h_1}{2}$
- $z_i - x_i = \frac{-h_{i-1}}{2}$
- $z_n - x_n = \frac{-h_{n-1}}{2}$

The author uses this example to illustrate the properties of quadratic splines. She points to the first property (continuity conditions at the interior nodes) of the quadratic spline function and to the second property (continuity conditions at the first derivative). A linear system with six equations and eight unknowns is obtained. The unknowns are the coefficients of the quadratic spline. Furthermore, the author demonstrates an example to evaluate the this kind of quadratic spline. Example 8.8 in the book uses the same data points as for the previous case, i.e., "nodes=knots". Figure 2.29 shows the plot obtained for the case "nodes≠knots". The disadvantage of using this approach is that the the matrix is not tridiagonal and thus system is more expensive to compute.

The author then turns her attention to piecewise polynomials. In the next sections better smoothness results will be obtained by Hermite interpolation and cubic spline interpolation.

Piecewise Cubic Hermite Interpolation

At the beginning of this section the author emphasizes that "one important use of Hermite interpolation is in the setting of piecewise interpolation:[it] can be used to preserve monotonicity" [Fausett, 2008]. It is important to explain cubic Hermite interpolation in order to understand cubic splines interpolation: a disadvantage of cubic Hermite interpolation – its low degree of smoothness – will serve as motivation for the study of cubic splines interpolation.

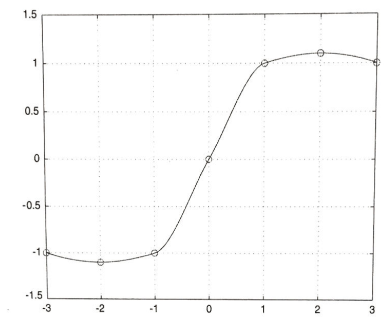


FIGURE 8.21: Piecewise Hermite interpolation.

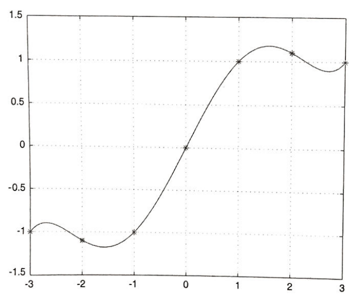


FIGURE 8.22: Polynomial interpolation.

Figure 2.30: By considering some data point of $x = [-3, -2, -1, 0, 1, 2, 3]$ and $y = [-1, -1.1, -1, 0, 1, 1.1, 1]$, a) Figure 8.21: piecewise Hermite interpolation by used `pchip` function in MATLAB. b) Figure 8.22: polynomial interpolation of same data ([Fausett, 2008, p.304]).

By means of a plot (Figure 2.30) using the built-in MATLAB function `pchip`, the author shows that cubic Hermite piecewise interpolation polynomial has continuous first derivative at the interior nodes. The author includes in the figure a plot of polynomial interpolation of the same data in order to point out important differences between them. The polynomial interpolant is smoother but oscillates (overshoots) more.

2.2.4.2 Teaching Structure

In this section we illustrate the details of the section *Cubic Spline Interpolation*. In the *Introduction* we will display the motivation and foundations Fausett uses to introduce this topic. Then, in the *Elaboration* section we will study how the author develops the theory of cubic splines.

Introduction

The author starts by restating the advantage of working with cubic splines, which is to get the maximum smoothness possible. This can be calculated to be the continuity of first and second derivatives. She also points to the simplicity of calculating the required coefficients of the cubic spline after a suitable choice of basis functions.

The way this author presents this material is quite different from that of previously reviewed books. She starts by considering the form of each spline piece:

$$P_i(x) = a_i \frac{(x_{i+1} - x)^3}{h_i} + a_{i+1} \frac{(x - x_i)^3}{h_i} + b_i(x_{i+1} - x) + c_i(x - x_i) \tag{2.15}$$

where $x \in [x_i, x_{i+1}]$ and $h_i = x_{i+1} - x_i$. This form is justified by observing that the second derivative of the spline will be a piecewise linear function that is continuous at the knots.

The following points are made by studying the formula (2.15):

1. The continuity condition for the second derivative is verified.

2. By using the interpolation property of the spline, $P_i(x_i) = y_i$ and $P_i(x_{i+1}) = y_{i+1}$, we can calculate the coefficients b_i and c_i in terms of the a_i:

$$b_i = \frac{y_i}{h_i} - a_i h_i$$

$$c_i = \frac{y_{i+1}}{h_i} - a_{i+1} h_i$$

3. By using the continuity conditions of the first derivative at the knots, with the use of the coefficients b_i, c_i we get

$$h_i a_i + 2(h_i + h_{i+1}) a_{i+1} + h_{i+1} a_{i+2} = \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \tag{2.16}$$

for $i = 1, ..., n-2$

The author discusses how to use formula (2.16) to compute the coefficients of the cubic spline interpolant. In order to get a square system there are different choices for the required additional conditions on the second derivative at the endpoints. Then she suggests "the simplest choice" is the natural cubic spline, for which $a_1 = a_n = 0$.

Elaboration

By means of Example 8.10, the author illustrates the construction of a natural cubic spline interpolant. The example chooses five data points with equally spaced abscissae. She asks the reader to observe the resulting smoothness in this example and compare it to the examples for piecewise linear interpolation and piecewise quadratic spline interpolation. Figure 2.31 shows the graph of the cubic spline.

Following this example, the author describes Runge's function in another example (Example 8.11), using the same data of Example 8.6 from the section on polynomial interpolation.

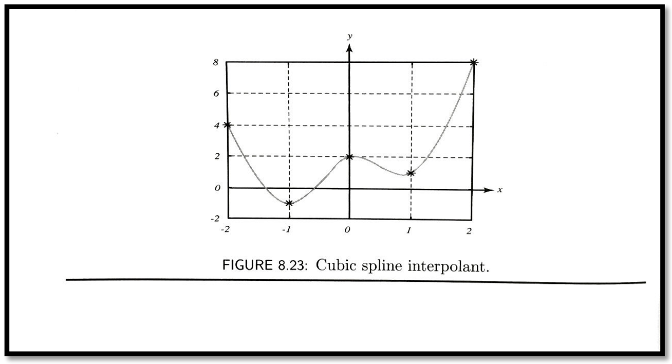


Figure 2.31: Natural cubic spline interpolating the data points $(-2,4),(-1,-1),(0,2),(1,1)$ and $(2,8)$ ([Fausett, 2008, p.306]).

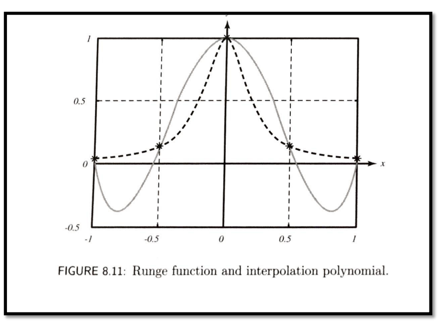


Figure 2.32: a) [Fausett, 2008, p.292].

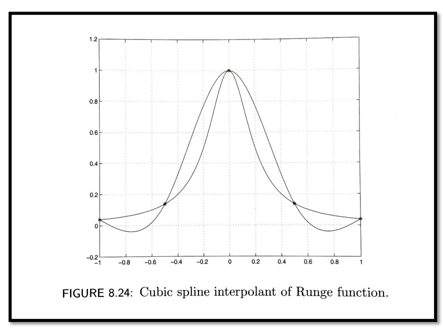


Figure 2.33: b) [Fausett, 2008, p.307].

Figure 2.34: a) Runge's function and interpolation polynomial, b) Cubic spline interpolation of Runge's function

The aims is to show a difference between the use of cubic spline interpolation and polynomial interpolation. One can observe in Figure 2.34 that the cubic spline does not exhibit the undesired behavior of the interpolation polynomial.

Additionally, the author gives a MATLAB code to show how to program the Runge function example by using the `spline` function in MATLAB. After that the author shows the code of a MATLAB function for calculating the spline coefficients by using the *Gauss-Thomas* method for tridiagonal systems. Finally, the author presents a code for the evaluation of a cubic spline (Fausett [2008], p. 309).

In a section called *Discussion*, the author shows some theoretical aspects of the derivation of cubic splines in more detail. The author developed the following points:

1. Choice of “nodes=knots”.
2. Computation of the number of end conditions that need to added, and what formulation they can have.
3. By looking at the previous form 2.15, enforcement of the properties of the *cubic splines*.
4. Explanation of the steps needed to calculate the coefficients b_i and c_i in the previous section.
5. Discussion of possible options for end conditions: natural cubic splines and clamped splines (which specifies the first derivatives at the endpoints).
6. Study of the error between the spline and the original function by looking at the general formula:

$$\mid S(x) - g(x) \mid < kh^4 G = \bigcirc(h^4) \tag{2.17}$$

2.2.4.3 Outcome Procurement

At the end of the chapter on interpolation, the author presents a summary where the main results of polynomial interpolation, Hermite interpolation, and cubic spline interpolation are outlined. Then comes a list of suggested reading, and finally some exercises. These are divided into four sections. *Practice the Techniques* section has exercises to be done by hand, by interpolating with piecewise linear, piecewise quadratic and cubic splines, as well as using different bases to express the polynomials. *Computer Investigations* has interpolation exercises that are to be solved using a calculator or a computer. *Explore Some Applications* has applied problems, some within mathematics, as statistical functions or calculation of integrals, and others outside mathematics, for instance chemical and engineering problems. The last part of this section, *Extend Your Understanding*, is targeted to students who wish to go a step further on the topic of Interpolation. There are some theoretical questions and some implementation issues.

Chapter 3

Comparison and Evaluation

The intention of this chapter is to go into the details of the procedures, strategies, methods and topics that the authors have used and developed for the purpose of teaching cubic splines. In order to compare the different books, we have based our study on questions and answers. These questions have been selected to cover all the issues that need to be discussed in order to make a proper of evaluation of the material. The idea of the question-and-answer scheme is to develop the central topic of this thesis in a clear and easy manner for the reader.

Moreover, each question will have an answer for each particular book. At the end of each section we will present a table giving the details that have been discussed. The aim of these tables is to make it easy to observe the differences and similarities between these books, and to have a basis for the evaluation.

After each question is presented, we look into the facts provided by the answers and draw our own conclusions and make comparisons. Sometimes two or more books have very similar answers to a question, and other times there is a sharp contrast between books. In this section we will evaluate each book in relation to each of the posed questions.

Question 1 **How is the topic of *cubic splines* initially presented?**

I. Steven C. Chapra's book (Chapra [2011]).

Figure 2.3 shows the organization of the topics in the chapter *Splines and Piecewise Interpolation*. Where the subject of cubic splines is introduced and explained. A plan of the chapter is displayed in Figure 2.3.

As observed in Figure 2.3 the *insertion point* rests on three introductory sections.

Firstly, *Introduction to Splines*, the main thrust of this section is to show a difference between different kinds of interpolation splines by increasing data points, and giving a simple example and showing Figure 2.7 to make the idea clear.

Additionally there is a brief summary, looking into its historical development. Secondly, linear splines are explained in precise details. An introduction to linear splines includes an example to explain the formula and how one can calculate the linear splines that interpolate some data. Moreover, the book mentions that data points can be interpolated by splines of different orders. Furthermore, advantages and disadvantages of linear splines are listed. Then an expanded linear spline section is given by the Table Lookup section.

Next, quadratic splines are explained in detail, before going on to the cubic splines section.

II. Timothy Sauer's book (Sauer [2012]).

In Sauer's book, the introduction to cubic splines starts the chapter *Interpolation*. *Cubic splines* is inserted in three steps. Firstly, there is an introduction to general splines. Secondly, the first simple spline (linear spline) is given by means of a simple example. Here it is shown how to write down a formula for linear splines. There is a reference to Figure 2.16, where we can observe a plot of linear splines.

Finally, using the same data, cubic splines and their formulas are displayed. Both types of splines were included in the same Figure 2.16.

III. David Kincaid and Ward Cheney (Kincaid and Cheney [2002]).

The access to cubic splines in Kincaid and Cheney's book starts by introducing the concept of *spline interpolation* in Chapter 6 (Approximating function) of the book. This section is divided into three parts. Firstly, there is an introduction to spline functions and how one can construct this function by satisfying the conditions that define a spline function of degree k. Secondly, a spline of degree zero is shown. Then a disadvantage of this type of spline serves to introduce linear splines and shown in Figure 2.4. Additionally, a pseudo code for the construction of linear splines is given. The next step will be to present cubic splines.

IV. Laurene V. Fausett's book (Fausett [2008]).

The presentation of cubic splines in Fausett 's book starts with the *Piecewise polynomial interpolation* section. The section includes three subsections, *piecewise linear interpolation*, *piecewise quadratic interpolation*, and *piecewise cubic Hermite interpolation*, as observed in Figure 2.26. The display shows three stages of presentation in the *piecewise polynomial interpolation* section. First, there is a motivation for why we are interested in studying this topic. Secondly, a historical text presents the topic idea. Finally, the section mentions a plan for the next steps.

Piecewise linear interpolation explains the derivation of a formula for linear interpolation and shows a plot of linear interpolation by giving an example. *Piecewise quadratic interpolation* also offers all the details of the derivation of this type of spline. Finally, the *Hermite interpolation* section clarifies what is a cubic Hermite by including an example and its plot.

This introduction is followed by *Cubic Splines Interpolation*.

Evaluation 1.

Chapra's book offers extensively detailed steps in the presentation of spline interpolation. Students get a scientific presentation that gradually leads them to understand spline interpolation before the introduction of cubic spline interpolation. There is certainly enough material for students to understand the topic, but tone can ask if all the information is really relevant and important for a presentation of cubic spline interpolation. For example, there is a subsection called “table lookup” inside the section on linear splines. But looking at this kind of table does not contribute to a better knowledge or understanding of this subject, and on the contrary, distracts the student from the most important issues. The section on Quadratic Spline Interpolation offers precise details on this topic but mentions that actually cubic splines are most frequently used. If cubic splines are the most popular, why dedicate so must detail to quadratic splines? Students judge the importance of a topic by how much effort is put into explaining it, so allowing so much time to quadratic splines gives a wrong impression and also suggests to the student that he should spend valuable time on this subject.

In Sauer’s book, the introduction to cubic splines was simplified by doing it in several steps. The cubic spline section introduces linear as well as cubic splines. Although this section is called cubic splines, it is actually a section on spline interpolation. Therefore, the title of the section does not reflect the true content, and is not accessible to a reader who has no knowledge of cubic splines. A student who starts reading a section called “cubic splines” will be confused by reading instead about linear splines . The presentation of the differences and similarities of linear and cubic splines was not done in a scientific manner. Nevertheless, the fact that Sauer’s book called the section “Cubic Splines” marks the importance of the subject.

The presentation in Kincaid and Cheney’s book starts by explaining interpolation by splines, and in this section they mention two types of interpolation before cubic splines are introduced. The manner of presentation was quite understandable, but showing first a spline of degree 0 gives the idea of a non-continuous spline. This goes in opposition to the main idea of splines, which is that they are continuous functions, and furthermore as smooth as possible. A spline of degree 0 is not a good first example of the notion of splines. In addition, the authors completely ignore quadratic splines, and yet in the exercises section they posed some questions about quadratic splines but not about splines of degree 0. Also, spline of degree 1 are usually referred to in the literature as *linear* but here this common name was not used.

The presentation of cubic splines in Fausett’s book appears as a good presentation by explaining different types of interpolation step by step. Going from piecewise linear to piecewise quadratic, to Hermite interpolation, the author makes the presentation in a smooth way so that the student understands the particularities of cubic splines interpolation. The author starts by giving some motivation and history of cubic splines in the introductory section called *Piecewise polynomial interpolation*. Then goes on to define what a spline is, and to describe the content of each of the subsequent subsections. Secondly, it looks at splines of increasing order in a step by step manner. Finally, in the subsection dedicated to cubic splines interpolation the author refers to examples mentioned in the previous sections.

My assessment is that Fausett’s book has the best introduction to *cubic splines interpolation*.

Question 2 What is the connection between the previous chapter or section and the cubic splines section?

I. Steven C. Chapra’s book (Chapra [2011]).

In Chapra's book the bond between the previous chapter and the *splines and piecewise interpolation* chapter is the Runge function. At the end of the previous chapter, the author displays Runge's function example, which displays a problem of doing high order polynomial interpolation. It includes the details of how we can observe the error by increasing the interpolation points and shows a plot of these errors. The Runge function example does not appear in the introduction of the *cubic spline* section, but rather this example is shown in the section *Piecewise interpolation in Matlab*, which came after the section on cubic splines and before the section on implementation. The connection point is the *Runge function* example. Runge's example indicates that some functions cannot be fit well with polynomials, and thus the connection to the *cubic spline* section for interpolating with high order polynomials.

II. Timothy Sauer's book (Sauer [2012]).

In Sauer's book, there is no bond between a previous chapter or section to *cubic spline* section. The section on interpolation error shows Runge's function, but it is a motivation for Chebyshev interpolation and not for cubic splines. There is no reference to this example in the sections on splines.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

In Kincaid and Cheney's book there is no connection between previous chapters or sections and the cubic splines section. The presentation of cubic splines was done independently, without referring to previous sections or chapters.

IV. Laurene V. Fausett's book (Fausett [2008]).

In Fausett's book there is a bond between a previous section and the cubic splines section. The *Runge function* example was described in the section that precedes the *piecewise polynomial interpolation* section. The example is mentioned in the introduction of this section as a motivation for the topic. The example is displayed again in the cubic splines subsection of the *piecewise polynomial interpolation* section and the author points to previous examples in previous sections and solves them using *cubic splines*.

To summarize question 2 Figure 3.1 shows the details of the bond between previous chapters or sections and the cubic splines section.

Evaluation 2.

In Chapra's book, the binding glue between chapters is Runge's function. This example addresses the need to use cubic splines, but the author does not indicate this example as a motivation for the development of cubic splines. The point where Runge's function example is discussed is at the end of the chapter, and thus is not something that will motivate students to the subject.

In Sauer's book, Runge's function example was given to motivate Chebyshev interpolation, but not to motivate splines. Had this example been brought up again in the introduction of cubic splines as motivation, it would have stimulated students to read on.

Kincaid and Cheney's book go directly to explain the theory of cubic splines without any motivation. This makes it hard for students to understand why they might want to read on. Especially difficult is the case when this book is used as a textbook at a basic level.

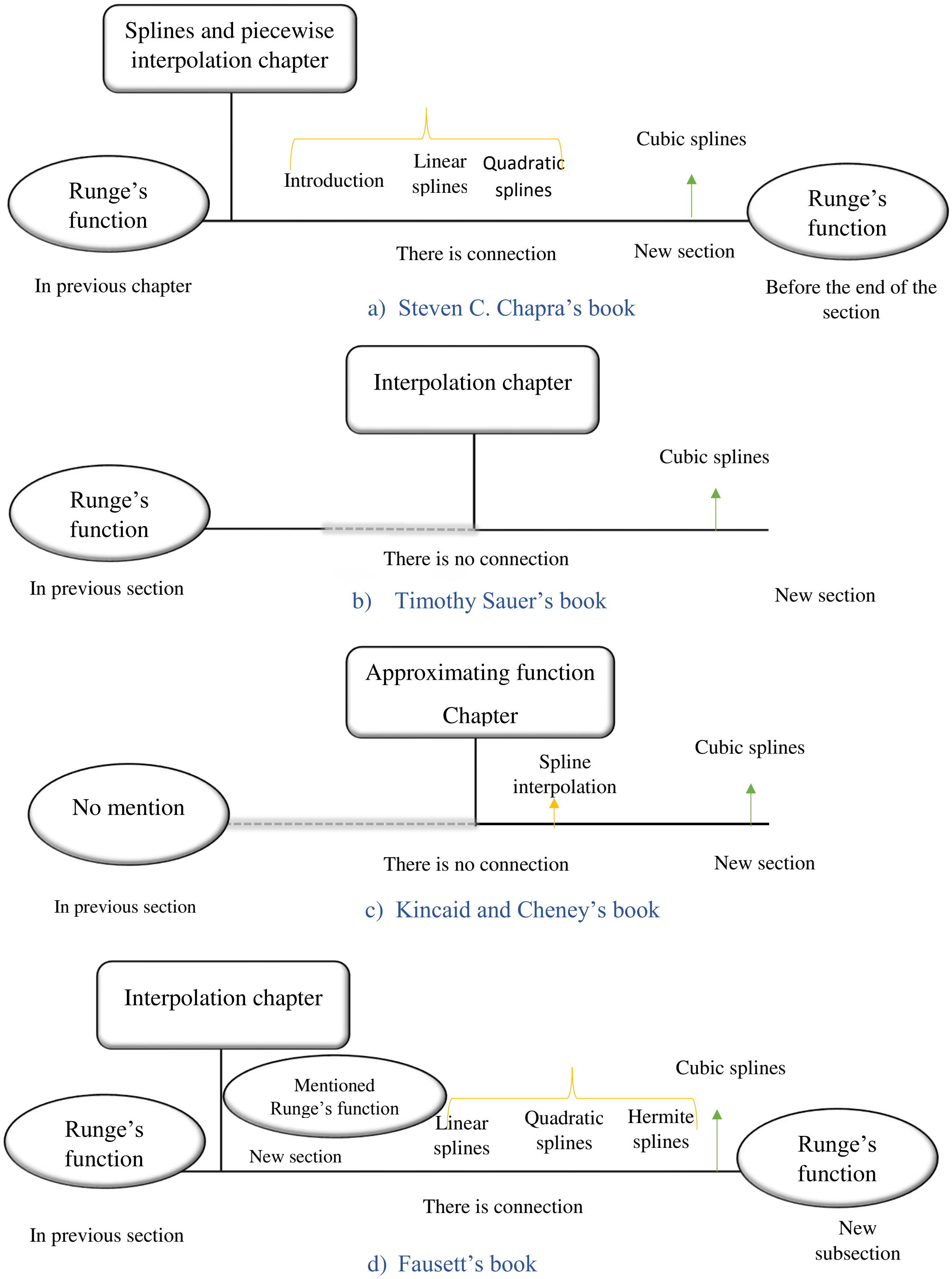


Figure 3.1: Scheme of the connection between the previous chapter or sections and cubic splines

By mentioning Runge's function example at the end of the previous section, Fausett bonds the new section to the previous ones and thus makes the presentation some consistent and motivates students to study the subject. This example is mentioned at the introduction of the *piecewise polynomial interpolation* section and then again when this type of interpolation is compared to cubic splines interpolation. By using the same example in different circumstances, the author employs a technique for *variation theory* Maghdid [2016], and this results in enhanced clarity for basic students as to why the study of cubic splines is interesting and how numerical analysis can offer different ways of handling a particular problem.

My assessment is that Fausett's book shows the greatest connection between previous parts of the book and the section on cubic splines.

Question 3 What is the main justification for the study of cubic splines?

I. Steven C. Chapra's book (Chapra [2011]).

Chapra's book focuses on a practical reason for studying *cubic splines*: "cubic splines are most frequently used" [Chapra, 2011]. Also, he says that "cubic splines are preferred because they provide the simplest representation that exhibits the desired appearance of smoothness."

II. Timothy Sauer's book (Sauer [2012]).

Sauer's book presents the idea of splines as an "alternative approach to data interpolation," where even with a large number of data points one can use lower degree polynomials.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

Kincaid and Cheney's book starts by saying that the theory and construction of cubic splines will be developed because "these are often used by practice."

IV. Laurene V. Fausett's book (Fausett [2008]).

Fausett's book states that "we can do better, without much more work" [Fausett, 2008]. This is followed by the assertion that there is enough information to require continuity of the first two derivatives.

Evaluation 3.

In Chapra's book, the introduction starts by explaining how higher degree polynomials can cause large errors and oscillations. Splines are offered as an alternative, and then the reader is given further reasons to study cubic splines in particular because of their extended use and practicality.

Sauer begins by explaining why the study cubic splines is important, and he does this by giving an example of a linear spline that interpolates a set of data points. I believe this strategy will not make it clear for students at all levels why a linear spline is a good alternative to the solution of this interpolation problem. The introduction of a topic is of utmost importance, as a good introduction will attract the student and will also attract teachers to use the book as textbook for the course. In my opinion, this book lacks strong and engaging introductions.

Kincaid and Cheney's start on the subject of splines gives the impression that it is a difficult topic, as the section begins by stating a formal definition of a general spline and no atempt to rely on intuition is made. This approach can make it difficult for students to understand and to feel motivated to pursue the subject.

The section of Fausett's book on *piecewise polynomial interpolation* has a good explanation of why splines avoid some problems caused by higher degree polynomial interpolation, and even explains the physical construction of splines. Later, in the section on cubic spline interpolation, the author explains the reason to study cubic splines by the phrase " We can do better without much more work" [Fausett, 2008].

My assessment is that Chapra's book motivates the study of cubic splines in a way that awakes the interest of most readers.

Question 4 How are cubic splines defined?

I. Steven C. Chapra's book (Chapra [2011]).

Chapra's book clarifies a definition of *cubic spline* in an introduction to the cubic spline section. The definition starts by giving a general formula for a cubic polynomial. Then he proceeds to enumerate the conditions that cubic splines must satisfy. Moreover, the definition also states the need to add two end conditions.

II. Timothy Sauer's book (Sauer [2012]).

Sauer's book gives a definition of *cubic splines* by stating its form and properties in a systematic manner. The definition starts by giving a general formula for cubic polynomials, and then the properties of cubic splines are described step by step. Then, the need to add two end conditions is explained and several alternatives are provided.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

In Kincaid and Cheney's book the definition is given without showing the general formula of a cubic spline. Rather, the form (linear) of the second derivative is given, and then this formula is twice integrated to get the form of the cubic spline. The definition is observed to have two degrees of freedom, and thus there is a choice of two end conditions.

IV. Laurene V. Fausett's book (Fausett [2008]).

Faust defines splines as piecewise polynomials with "maximum possible smoothness." There is no explicit definition of a cubic spline in Fausett's book, but rather an informal derivation of the spline ends in the more formal definition, where the form of the polynomial pieces is assumed in order to simplify the calculations. There is no mention that splines of other degrees can be also defined.

Evaluation 4.

In Chapra's book, there is no clear definition of cubic splines, but the definition is developed along the text. Although a general formula of cubic spline is given, the definition is embedded in a half page text. This lengthy presentation includes all conditions that define a cubic spline but it is not made clear to the student that they are an integral part of the definition.

Sauer's book also starts by giving a general formula of cubic spline, but then the properties are clearly stated and enumerated. This makes it easier for the reader to comprehend the definition of cubic splines.

Cheney's book offers a definition of a cubic spline as a particular case of a general spline of degree k. In the cubic splines section the general definition is applied to the cubic case, but this might not be clear enough for basic students.

Fausett's book define splines in a loose way, and do not give any formal or even clear definition of cubic splines. Rather, after assuming the form of a cubic spline the author shows how a cubic spline develops by demanding as much continuity as possible. It is unclear that the author chose this way with her students in mind. When the definition starts with an assumed form for the cubic spline, without an explanation of why this choice was made, some students will find it difficult and confusing, and wonder how one can obtain a cubic spline if one does not figure out first this particular form of the spline.

My assessment is that Sauer's book has the best strategy of introducing the definition of cubic splines.

Question 5 What strategy is adopted to present the properties of a general spline of degree k?

Given the data set $\{(x_0, y_0), \dots, (x_{n-1}, y_{n-1}))\}$, the spline S of degree k satisfies

1. $S(x_i) = y_i$;
2. $S \in \mathcal{C}^{k-1}$.

I. Steven C. Chapra's book (Chapra [2011]).

In Chapra's book the properties of cubic spline mentioned in the introduction are quite vague. When quadratic splines are introduced, the author asserts that "to ensure that the nth derivatives are continuous at the knots, a spline of at least $n + 1$ order must be used." Each type of spline is defined separately, but there is no presentation of a general definition of splines of degree k.

II. Timothy Sauer's book (Sauer [2012]).

In Sauer's book only the properties of *cubic splines* are displayed in great detail, but there is no definition of a general spline.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

In Kincaid and Cheney's book, the section opens with a formal definition of a general spline of degree k. Afterwards, this definition is applied to construct linear splines as well as cubic splines.

IV. Laurene V. Fausett's book (Fausett [2008]).

Fausett's book develops the concept of cubic splines without defining a general spline. The properties of cubic splines are not clearly stated, and it is only noted that one can ask for continuity conditions because there is enough information to do so.

Evaluation 5.

In Chapra's book, the properties of splines are presented in three different places in the book. Firstly, in the quadratic splines section, the author point to the properties of general splines and mentions the cubic case before enumerating the quadratic splines properties. This was somewhat confusing, because the point was to discuss the properties of quadratic splines. The fact that cubic splines are the ones most frequently used, was mentioned when describing quadratic splines but the reader cannot be supposed to have previous knowledge of cubic splines. Then, the properties of cubic splines are mentioned within the definition of cubic splines in a shortened form. The details of the properties of cubic splines are not mentioned in the cubic spline section. Finally, the properties of cubic splines are used for the derivation of cubic splines. Thus, the properties of cubic splines are given in a piecewise manner that might not make it easier for a student to understand these properties.

Sauer's book displays the properties of cubic splines in well organized steps and then applies them in the cubic spline formula. This procedure increases the ease of understanding of the topic. In addition, the book mentions all end conditions as forming part of a property that the book calls, *Property4a*, *Property4b*, and so on. These properties are discussed in a dedicated section called *End conditions of cubic splines*. Other books do not include end conditions as a property of splines, are it might be confusing for a reader who looks at more than one book.

In Cheney's book, the properties of cubic splines are included in the description of the properties of general splines, done in a detailed and clear way, but the book does not focus on the particular properties of cubic splines in the cubic splines section.As all students do not have the same level, one might wonder if some students are not able to profit from this presentation. Having a cubic splines section, it would have made it clearer for the student to present the details of the properties and derivation of the cubic spline formula.

The properties of splines in Fausett's book, is done in a different way than the rest of the books. The properties are already discussed in the introduction to cubic splines, but tit was done in an unclear way, embedded in the text without pointing out what exactly these properties are. This adds a difficulty to students at the basic level. Later on in the book the properties of cubic splines are retaken, but this time in a clearer way. To present the properties at the end of the section on cubic splines makes it difficult for students to understand the concept since the start of the presentation. Also, the phrase *properties of cubic splines* is never used in this book, but mentioning them by name because makes the student aware of their role in the theory. It seems that the author does not take into account the level of the students that the book addresses.

My assessment is that Sauer's book manages to present the properties of splines in the best way.

Question 6 What is the methodology used for the derivation of cubic splines?

I. Steven C. Chapra's book (Chapra [2011]).

The derivation of cubic splines constitutes a subsection in the section dedicated to cubic splines. This subsection starts by giving a general formula of a cubic spline. The formula is used in the derivation cubic splines by requiring that it satisfy the properties of cubic splines.

After the derivation steps are explained, the author shows one ends up with an undetermined system which can not be solved to find the coefficients in the cubic spline formula. Here he alludes to the need of two end conditions to yield a square linear system. The linear system is then shown as a strictly diagonally dominant tridiagonal matrix and thus the linear system has a unique solution.

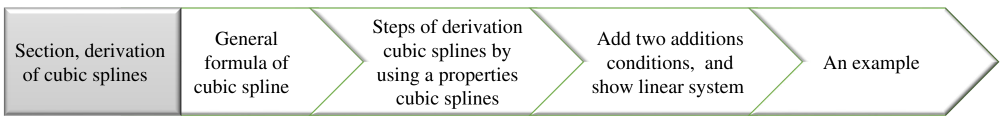


Figure 3.2: Flow of the derivation of cubic splines in Chapra's book.

II. Timothy Sauer's book (Sauer [2012]).

The derivation of cubic splines in Sauer's book is done in particular for a natural spline. It starts by giving a general formula of cubic splines, and then the author shows that by requiring the formula to satisfy smoothness conditions, a linear system for the formula's coefficients is obtained, with n equations and n unknowns. The system matrix is strictly diagonally dominant and thus has a unique solution.

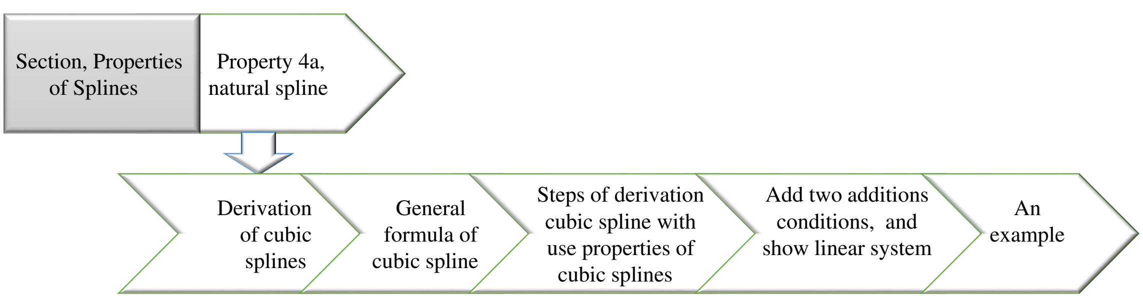


Figure 3.3: Flow of the derivation cubic splines in Sauer's book.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

Kincaid and Cheney's book present a derivation of cubic splines after having given a general definition of splines and having observed that we will need two extra conditions in order to get a solvable system of equations. This book starts with a linear second derivative (2.7), and then integrates this formula twice. The result is a cubic spline. The formula obtained this way has two integration constants that can be determined by imposing interpolation conditions. To obtain a square system of equations for the coefficients of the cubic spline formulas, the author chooses to impose the end conditions for a natural cubic spline. The matrix of this system is shown to be tridiagonal, symmetric and diagonally dominant.

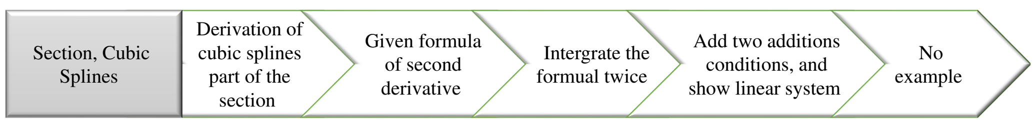


Figure 3.4: Flow of the derivation cubic splines in Kincaid and Cheney's book.

IV. Laurene V. Fausett's book (Fausett [2008]).

In Fausett's book, A derivation of cubic splines is presented at the start of the introduction to cubic splines. A form for the cubic spline on each interval is given (2.15) and shown to satisfy the properties for the second derivative of cubic splines. The author then imposes the rest of the conditions (continuity of the first derivative and interpolation) and shows that a tridiagonal system is obtained to determine all of the unknowns in the formula of cubic splines.

The derivation of cubic splines is revisited in a subsection named *Discussion*. Here the author explains in more detail the process of the derivation of cubic splines. She takes the approach of starting with a linear second derivative and shows that the formula that was used previously can be obtained by integrating this form twice. At this point there is also a discussion of the need for two additional conditions.

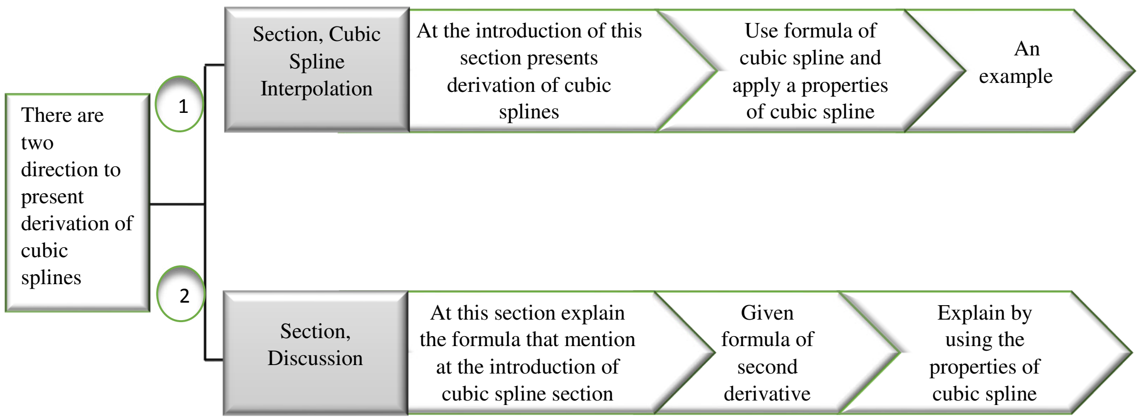


Figure 3.5: Flow of the derivation cubic splines in Fausett's book.

Evaluation 6.

In Chapra's book the derivation of the equations that must be solved in order to calculate the coefficients of a cubic spline is contained in a dedicated subsection inside the cubic splines section. This fact alerts the student of the importance of this derivation in the cubic splines theory. The derivation of the equations for cubic splines starts by presenting a general formula for cubic splines. This first step contributes to a good understanding, and the following steps detail how the properties of cubic splines are used to generate the equations. Additionally, the presentation arrives to a point when there is a need for two additional conditions. Then the linear system is shown to be tridiagonal. The style of this presentation is adequate for students of all levels. Finally, an example is given to show how one can construct a natural cubic spline that interpolates a given set of data (see Example 18.3 (Natural Cubic Splines)).

Sauer's book was similar to Chapra's book in the manner it explained the derivation of the equations for cubic splines. It also starts by giving a general formula, but then proceeds in a different direction, as we show in Figures 3.2 and 3.3. Sauer's book prefers to present the derivation within the property of natural cubic splines, to make it easy for students to understand the steps of derivation and to facilitate the explanation of natural cubic splines as given in a partial example.

Kincaid and Cheney's book present a derivation of cubic spline that differs greatly from the other books already discussed. This book starts the presentation by stating the linearity of second derivative of a cubic spline, and then integrates the formula for the second derivative twice to get the formula for the cubic spline. After this step, it is noted that two additional end conditions are needed. Starting with the linear equation makes derivation quite easy, but it may not be so clear for a basic level. This is a book that is more appropriate for a more advanced level, as mentioned in the preface.

Fausett's book was similar in its derivation to Kincaid and Cheney's book, as it uses the integration of the second derivative formula. Nevertheless, it takes a different direction, as we observed in Figure 3.5. This book first shows, in the introduction of the cubic splines section, how the equations for the cubic spline coefficients are obtained, starting for the general form of a cubic polynomial. Later, in the *Discussion* section, the author explains of the derivation of the equations by integrating the second derivative formula of a cubic spline. It arrives at the same formulas it had already derived in the previous subsection (formula (2.15)). We observed that this author prefers to give the pertinent information, work out the derivation, and then explain afterwards how this information can be justified. This strategy may not be adequate for students at every level. Besides, in mathematics it is important to understand why each step is taken. In this book the derivation of the equations for cubic splines was not clear enough despite the fact that all the details of the derivation were given.

My assessment is that both Chapra's and Sauer's books have equally good strategies for presenting the derivation of the equations that lead to the construction of cubic splines.

Question 7 What is the aim of the examples involving cubic splines?

I. Steven C. Chapra's book (Chapra [2011]).

In Chapra's book, there are three different examples in section of cubic splines:

- Example 18.3, Natural Cubic Splines.
- Example 18.4, Splines in MATLAB.
- Example 18.5, Trade-Offs Using `interp1`.

Firstly, Example 18.3 (Natural Cubic Spline) is given after explaining the derivation of cubic splines. The example shows how to fit a cubic spline to four given data points, and then to evaluate the spline at another point. The intention here is to familiarize the student with the technique presented previously, by solving the exercise "by hand," and also to help understand the definition of splines in the sense that they are piecewise functions, and thus different "formulas" must be used for evaluation, depending on the interval where the value lies.

Secondly, Example 18.4 (spline in MATLAB), and Example 18.5 (Trade-Offs Using `interp1`) are given in the subsection named *Piecewise Interpolation in MATLAB*. The intention here is to teach the reader how to use some built-in MATLAB functions to construct splines, plot them, and evaluate them. Example 18.4 presents Runge's function problem, this example was studied previously to observe the behaviour of polynomials of high degree. The solution of this example includes a MATLAB code. In addition, the example shows a plot of the function when the not-a-knot end conditions are used, and when a clamped end condition was used. The reader can observe that the second set of conditions fits the function badly. This was used to point out the importance of choosing the right end conditions for a particular problem.

Example 18.5 in presented in the *MATLAB Function: `interp1`* subsection. The example is an application to physics. It shows four different types of piecewise interpolation functions, a) linear interpolation (spline of degree 1), b) nearest neighbour (spline of degree 0), c) cubic spline with not-a-knot end conditions, and d) piecewise cubic Hermite interpolation. Plots of the four cases are presented so that the reader can observe and understand the differences between these approaches.

II. Timothy Sauer's book (Sauer [2012]).

In Sauer's book, there are three examples mentioned in the section of cubic splines,

- A simple example at the introduction of cubic splines, Connecting the dots.
- Example 3.13, Checking all cubic spline properties.
- Example 3.14, Calculating a natural cubic spline.

Firstly, an example at the introduction cubic spline section relates a linear spline and a cubic one passing through the same given points. This example includes plots of the two spline functions. Example 3.13 comes after the properties of cubic splines have been detailed. It presents a piecewise cubic polynomial and checks that all properties of cubic splines are satisfied. Example 3.14 appears after presenting the details of the derivation of cubic splines. This example shows how to apply end conditions, construct the system matrix, solve the system, and finally calculate all coefficients of the cubic spline.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

In Kincaid and Cheney's book there is no practical example of cubic splines, but the author presents a table of errors produced by interpolating a given function at 10 equally spaced points using a cubic spline. The example points to two results: no error at interpolation points and largest error at the first subinterval.

IV. Laurene V. Fausett's book (Fausett [2008]).

There were two examples to illustrate cubic splines,

- Example 8.10, Natural Cubic Splines Interpolation.
- Example 8.11, Runge Function.

The introduction to cubic splines consists of the formula and some properties. Then the first example (8.10) calculates the coefficients of a natural cubic spline that interpolates four given points by "by hand.". The second example (8.11) shows a cubic spline that interpolates Runge's function at 5 points. There is no mention of what end condition was used.

Evaluation 7.

Chapra's book offers an example of a natural cubic spline right after the presentation of the derivation of the equations for the coefficients of cubic splines. This simplifies the understanding of the properties and derivation of cubic splines. The example uses the same data points that have been used before in other situations, namely, Example (First-Order Splines) and Example (Quadratic Splines). With this example the author aims to show that using the same data points one obtains different results of spline interpolation. The example illustrates all details that have been mentioned in the cubic splines section and its solution is discussed in a systematic way, which makes it understandable for the reader. There are also more examples given in the subsections called *Splines in MATLAB*, and *Trade-Offs Using* `interp1` . The first example studies the Runge function example, which the author mentioned in the previous chapter. The example shows a satisfactory result obtained with the `spline` function in MATLAB. This example also contributes to motivate the study of cubic spline interpolation by comparing this example with an inadequate polynomial fit. Runge's function example is an important one that should become familiar to students. The second example studies the use of the MATLAB built-in function `interp1`, which can be used for several types of one dimension spline interpolation. The author first explains the uses and syntax of the built-in functions before solving the example. This makes it easy for students to understand the uses and reach of the MATLAB functions. The author uses the functions to do linear interpolation, degree zero spline interpolation, cubic spline interpolation, and Hermite cubic spline interpolation. The author gives the student several options to implement piecewise interpolation of one dimension.

Sauer's book presents three examples in great detail. These examples will be understandable for students of any level. By studying these examples the student gets a clear path to applying cubic splines to interpolate given data.

Fausett's book presents an example of natural cubic splines interpolation in order to show how to interpolate with cubic splines. The example marks three important points to understand cubic splines interpolation, namely, properties of cubic splines, derivation of the equations for cubic splines, and end conditions.

My assessment is that Fausett's, Chapra's, and Sauer's books, all give interesting examples of cubic splines interpolation that help a student to understand the cubic spline topic.

Question 8 How does the choice of end conditions help to achieve the desired outcome?

I. Steven C. Chapra's book (Chapra [2011]).

Chapra's book mentions end conditions early in the derivation of quadratic splines. There he shows that one additional condition is needed in order to match the number of conditions to the number of unknowns. He says "there are a number of different choices that can be made," but does not explain further, and instead makes a particular choice of end condition.

After identifying all cubic spline conditions and properties, Chapra shows that two additional conditions are required to find a unique solution. He refers to the quadratic case, and explains that here one can make a symmetrical choice of conditions, by giving one condition at each end point of the spline interval. He comments that these end conditions can be chosen in different ways. The book first describes *natural cubic splines*, and then two popular ones, *clamped* end condition, and *not-a-knot* end condition.

II. Timothy Sauer's book (Sauer [2012]).

After giving a definition and explaining the conditions and properties of a cubic spline, the author shows the shortage of equations one encounters, as there are more coefficients than equations. At this point he introduces the end conditions to make the system have a unique solution. The first end condition he mentions is for natural cubic splines. The book shows in clear details the derivation of natural cubic splines, gives applications and an example of how to construct a natural cubic spline.

In addition, there are four more end conditions given in the book, and the author points out the differences between them. Optically, the difference is minimal, as described in Figure 2.17, but it becomes clear that the choice of the kind of end conditions depends on the type of application.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

After explaining the properties of cubic splines, the authors show that the system of equations resulting from applying all interpolation and continuity conditions is under-determined. At this point they make it clear that two more conditions are needed, and the end conditions for a natural cubic spline are mentioned. The book does not mention that any other kind of end conditions may also be used, though.

IV. Laurene V. Fausett's book (Fausett [2008]).

Fausett introduces end conditions in the section on quadratic splines. After considering a general form of cubic splines and requiring the necessary conditions and properties, the author suggests the possibility of choosing several different conditions on the second derivatives at the end points, and suggests using a particular one (natural cubic spline). Additionally, in the *Discussion* section, she retakes the subject of end conditions, adding end conditions for the clamped spline, but no elaboration is made.

Evaluation 8.

Chapra's is on a good start, by mentioning the need for two additional conditions at the introduction of the section of cubic splines interpolation. This way he motivates the study of end conditions and motivates the need for two additions conditions in cubic splines interpolation in order to get a unique solution. The first end condition mentioned is the natural cubic spline, because this type is the most commonly used. He also explains the reason for calling this condition "natural," and then the author elaborates more about natural cubic splines by giving examples and showing a plot of this type of spline. Moreover, the author does not limit himself to natural cubic splines but also mentions there are several kinds of end conditions that can be choosen according to the particular application. The author presented all these end conditions in a way that is understandable by giving all the details of each set of end conditions and showing them in a plot. He also shows a table with the different end conditions and how to account for them in the linear system that determines the coefficients of a cubic spline. This will give the student motivation to think about what is the difference between these end conditions. Consequently, by explaining the three most common different end conditions, natural cubic splines, clamped end conditions, and not-a-knot end conditions, the student understands that there are several ways of doing cubic spline interpolation. Even though there are more end conditions commonly used for cubic spline interp.lation, these three give the student the necessary understanding without giving too much detail.

Sauer's book presents five different cubic spline end conditions. It starts with the natural cubic spline end condition. This end condition is given as part of the properties of cubic splines and is named *Property4a: Natural spline*. This end condition is presented in minute detail, but we believe that calling end conditions a *property* of cubic splines can be confusing for students. Especially as the author calls each different end condition a property, students make take some time to understand the difference between properties that are essential for the definition of a cubic spline and end conditions, which can change according to the case at hand. In addition, only the natural end condition is given together with the interpolation and continuity conditions of a cubic spline, and the rest of the end conditions are stated in a different subsection called*Endpoint conditions*, another possible source of confusion. On the other hand, although it is good to present more than one end condition to show the student that there are several possibilities for applying cubic splines to interpolation,at the same time it is not necessary to show some end conditions rarely used, as it serves as a distraction to the student.

Kincaid and Cheney's book presents only one end condition, and so it does not clarify what a natural cubic spline is, nor does it dwell on any details like the other books do. This book contains less information than the other books on end conditions for cubic splines.

Fausett's book offers two cubic spline end conditions, the natural end conditions, and the clamped end conditions. The natural end conditions is introduced in detail, but the clamped cubic spline end conditions was just briefly mentioned in the *Discussion* section.

My assessment is that Chapra's presentation of end conditions is the superior one.

Question 9 Is theory presented informally or by means of theorems?

I. Steven C. Chapra's book (Chapra [2011]).

In Chapra's book there is no mention of any theorem. All the theory is presented in an informal way.

II. Timothy Sauer's book (Sauer [2012]).

Sauer uses Theorem 2.10 from a previous chapter to point out that a strictly diagonally dominant matrix is invertible. With this theorem he proves Theorem 3.7, which states the uniqueness of an interpolating natural cubic spline. The second theorem mentioned in this section is Theorem 3.8, that states that for any set of end conditions specified in previous paragraphs there is a unique cubic spline.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

Kincaid and Cheney prove a theorem on the optimality of natural cubic splines. This theorem shows that the natural cubic spline is the smoothest possible $\mathcal{C}^2$ interpolating function, in the sense of having the smallest curvature.

IV. Laurene V. Fausett's book (Fausett [2008]).

Fausett does not present any theorem, but rather present theory as flowing text.

Evaluation 9.

In Chapra's book, as well as in Fausett's book no theorems are mentioned. The theory is informally introduced.

Sauer's book uses three theorems in the chapter on spline interpolation. The first one, Theorem 2.10, proved in a previous chapter that a square linear system with a strictly diagonally dominant matrix has a unique solution. The matrix of the linear system of equations that gives the coefficients of a natural cubic spline is strictly diagonally dominant. Consequently, the author proves Theorem 3.7 to show that there is a unique natural cubic spline that interpolates a set of given data points. The proof of this theorem precedes the statement of the theorem, and is done is a rigorous but informal way. The student here realizes that he can always construct such a cubic spline. The third theorem, Theorem 3.8, aims to show that cubic splines with other end conditions have a unique solution as well. This theorem helps students to know there is a unique solution in all cases of cubic spline interpolation, but the proof of theorem is not shown. The author focuses more on the content of the theorems than on the proofs.

In Kincaid and Cheney's book present the important Theorem on Optimality of Natural Cubic Splines, that says why natural cubic spline interpolation is preferred to other types of interpolation. The proof is done in rigorous mathematical detail. This theorem gives meaning to the choice of natural cubic splines, apart from just the fact that they are "the most commonly used."

My assessment is that Kincaid and Cheney's approach is the best in terms of how to present a theorem and the reason for doing it.

Question 10 How do the codes for computer programming contribute to support the presentation of the topic?

I. Steven C. Chapra's book (Chapra [2011]).

In Chapra's book, two detailed MATLAB programs are given in the subsection named *Table Lookup*, as an illustration of an application of linear splines. Following these codes are some lines of MATLAB code to show the use of the programs in a particular example.

Section *Piecewise interpolation in Matlab*, following the section on cubic splines, is divided into two subsections, *Matlab Function: spline* and *Matlab Function: interp1*. In the subsection on `spline` the *Runge function* is fitted using nine equally space points in $[-1, 1]$ with a not-a-knot spline and a clamped spline. Some lines of are given to show how this is done. The example includes a plot of the function, showing the two different types of splines. In the subsection on `interp1` an example is presented in which this MATLAB function is used to fit given data with a linear spline, a zero-degree spline, a cubic spline, and Hermite interpolation. Lines of MATLAB code are given to construct these interpolating functions and plot them. The codes and programs given in this book do not contribute to clarify the theory, but are more about how to use MATLAB to get concrete results.

II. Timothy Sauer's book (Sauer [2012]).

Sauer starts by giving a pseudo code for the construction of a natural cubic spline. Based on this, a MATLAB program for the construction of cubic splines is developed in several steps. The parts of the program appear often after a particular end condition of cubic spline has been presented.

The MATLAB program appears after the theory on cubic splines and the natural end conditions are given. The input for the code are the interpolation points and two values defining the end conditions. There is also another program for plotting the spline together with the data points. The code contains the end conditions for natural, curvature-adjusted, clamped, and not-a-knot cubic splines. The user has to comment all conditions except the one he wants to use.

The code calculates the coefficients of the spline by constructing the system, as explained in the theory, and then solving this system. Thus, the codes and programs are very much related to the material presented in the book, and constitute a tool to clarify this theory.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

Kincaid and Cheney do not use a particular programming language, but instead give algorithms as pseudo-code. This was done in two instances:

- A pseudo code to evaluate a linear spline (see Figure 2.23).
- A pseudo code for *Gaussian elimination* (see Figure 2.24).

There is no code or program to address the implementation of cubic splines. This book does not give any importance to the computational part of the topic, even though there are suggested problems that ask the reader to test some algorithms. Thus, it appears that the authors presume advanced knowledge of programming, and focuses only on the theoretical part of splines.

IV. Laurene V. Fausett's book (Fausett [2008]).

Fausett displays two MATLAB programs in *cubic spline* section. One is for the construction of the coefficients of a natural cubic spline, and the other one is for the evaluation of a cubic spline, given its coefficients. In the first code, the author takes advantage of the structure of the system matrix, and uses the Gauss-Thomas method for tridiagonal systems.

A MATLAB script is presented for the interpolation of Runge's function at five equally spaced points. This code uses the two programs previously described and plots the function and the spline.

Thus, the computer examples and codes are a direct application of the theory presented. Nevertheless, there is no mention of how to use MATLAB's built-in functions, nor of how to change end conditions.

Evaluation 10.

In Chapra's book MATLAB codes were presented in a special subsection, *Piecewise interpolation in MATLAB*, dedicated to several built-in functions in MATLAB that implement piecewise interpolation. Here the reader finds a connection between numerical solutions and the theoretical material on splines. In the first part of this section the student encounters MATLAB code that interpolates Runge's function with two different requirement of cubic splines end conditions by using the built-in function `spline`. The book reminds the reader of plots of this function and its polynomial interpolation, which were published in the previous chapter. With this reference the student can understand why cubic splines interpolation is important. Besides, the lines of code given in the book for interpolating the Runge function teaches the student to program cubic spline interpolation. In the previous chapter there is also a MATLAB code that solves the same problem doing polynomial interpolation, so this comparison can teach the student through variation theory Maghdid [2016]. A second MATLAB code shows the use of the built-in function `interp1`. Here a different problem, related to engineering, is solved to show the versatility of the MATLAB function, as it has several options. With this example the author illustrates the difference between the use of linear interpolation, Hermite cubic spline interpolation, and cubic spline interpolation. In the proposed exercises at the end of the section, the student will be able to modify these lines of code to solve other similar problems, again using techniques of variation theory. Finally, a MATLAB code given in an example in the *Case Study* section, where a MATLAB function is constructed to evaluate a spline and its derivative. Also, a MATLAB script that generates an interpolating spline and plots the spline curve is shown. It is clear that students profit from seeing how to program a numerical solution to a problem after studying the mathematical theory. Nevertheless, there is still the question whether giving programming codes will increase the student's capability of programming by himself.

Sauer's first code is a pseudo code which shows how to construct a natural cubic splines given the interpolation points. This code will teach students how to program natural cubic splines using any kind of programming language. Then, the book gives a MATLAB code that calculates the coefficients of a cubic spline. The program includes several different kinds of cubic spline end conditions, and the user must comment the options that are not relevant for the particular problem that is to be solved. The author demonstrates in two different ways how one can program natural cubic splines. In another use of variation theory, the pseudo code teaches how one can program them in a general way, and the MATLAB program will prepare students to program for this particular platform. Giving only a pseudo code might not be enough for some students who are not proficient in a particular programming language. Giving only a MATLAB program may not give the student the tools for understanding the essentials of the algorithms used. Example 3.14 also includes in its MATLAB code different options for end conditions and in this way it motivates students to observe that natural cubic splines are not the only kind the cubic splines one can use. Additionally, in section *Endpoint conditions* the basic code for each type of end conditions is explained. Here arises again the question of how much should be provided to student in terms of computer codes. How much will actually ease their understanding and how much will be a hindrance because there is not much left do be done by the student.

Kincaid and Cheney's book offers only pseudo codes. This will give students a guide in order to program in any language, but if the pseudo code is not clear enough the student will not be able to take this step.

Fausett's book gives three program codes. Firstly, the author gives a MATLAB function that computes the coefficient of the natural spline function. Secondly, she gives a MATLAB function that evaluates the spline. Thirdly, the book shows a MATLAB script that uses these two programs to construct a natural spline that interpolates Runge's function and plots the function and the spline. The author also draws the reader's attention to the previously computed polynomial interpolation a with higher degree polynomial, and mentions the improvement that resulted from applying splines instead. Having given full programs that compute and evaluate splines, students are not given the challenge of programming splines in MATLAB, which could have contributed to a better understanding of the theory.

My assessment is that Chapra does the best job of inserting programming to the chapter.

Question 11 How does a summary help to cement the knowledge?

I. Steven C. Chapra's book (Chapra [2011]).

In Chapra's book, there is no summary for the *cubic spline* section.

II. Timothy Sauer's book (Sauer [2012]).

In Sauer's book, there is no summary for the *cubic spline* section, but there is a summary of software that can be used for calculating and evaluating an interpolating polynomial both in MATLAB and in FORTRAN/C.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

In Kincaid and Cheney's book, there is no summary for the *cubic spline* section.

IV. Laurene V. Fausett's book (Fausett [2008]).

The last section of the chapter is called *Chapter Wrap-Up*, and contains a summary of some of the theory, in particular, the Lagrange form of a polynomial, the Hermite basis, and the construction of a natural cubic spline.

Evaluation 11.

In the three first books, Chapra, Sauer, and Cheney/Kincaid do not present any summary of the cubic splines section. Fausett's book does have a summary section. Here the author presents very briefly the key points that a student should learn.

My assessment is that Fausett's use of a summary is beneficial to students.

Question 12 What applications of cubic splines are given?

I. Steven C. Chapra's book (Chapra [2011]).

   Chapra offers a very detailed application of cubic splines in a section named *Case Study.* The problem is about using cubic splines to determine the thermocline depth and temperature gradient for Lake Platte, Michigan. In the Problems section there are also some proposed problems on the same subject and on other physics problems.

II. Timothy Sauer's book (Sauer [2012]).

   In Sauer's book there was no application described for cubic splines. But the book offered an application of a cubic Bézier spline, which was developed in the section following cubic splines. The application consists on the construction of fonts in PDF files. There are a few applications to population, economy, weather, and chemistry in the Computer Problems section.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

   In Kincaid and Cheney's book, the Computer Problems section has an application of cubic splines consisting on drawing a script letter and reproducing it by means of a cubic spline that interpolates the letter at 11 knots.

IV. Laurene V. Fausett's book (Fausett [2008]).

   In Fausett's book there is a section named *Explore Some Applications.* Here the author gives seven interesting problems from the fields of physics which can be solved using cubic splines.

Evaluation 12.

Chapra's book has a dedicated section, *Case study* ([Chapra, 2011], p. 452–456), where he illustrates an application of cubic splines. In this section a problem from real-life is solved by means of cubic spline interpolation, and that not only requires the calculation and evaluation of the spline, but also using the spline function as an approximation function in order to find derivatives of the underlying function. Apart from showing a direct application of the topic, this section shows how numerical analysis can be used to solve a scientific problem.

Sauer's book presents some applications in the computer exercises section, where there are three questions that show the use cubic splines in real-life.

Kincaid and Cheney has a few interesting applications in its Computer Problem section, especially to geometric design.

Fausett's book dedicates a special section of the exercises to applications. For these problems there is no instruction of using cubic splines interpolation. Consequently, students are left to experiment with different types of interpolation and discover on their own the best way of solving particular problems.

My assessment is that Chpra's book has the most complete and interesting applications that help motivate and capture the student's imagination.

Question 13 **How can the set of proposed exercises be described?**

I. Steven C. Chapra's book (Chapra [2011]).

Chapra proposes 14 exercises for the chapter *Splines and piecewise interpolation*, but most of the questions focus on cubic spline interpolation. The exercises are mostly about fitting a spline to given data. The list of exercises can be found in Problems, [p. 456] [Chapra, 2011].

II. Timothy Sauer's book (Sauer [2012]).

In Sauer's book 21 exercises are displayed immediately after the cubic spline section. The questions involve different kinds of problems and cover all details mentioned in the cubic spline section. There are several exercises intended to check that the student has grasped the concept of splines and their properties. They also focus on different end conditions for cubic splines. For details, see Section (3.4 Exercises) in the book Sauer [2012].

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

This book poses 36 exercises for splines in section (Problems 6.4) Kincaid and Cheney [2002]. Some of the questions ask for mathematical prooofs. Although quadratic splines were not mentioned earlier, there are exercises on this topic. Most of the exercises are about cubic splines and in particular about natural cubic splines.

IV. Laurene V. Fausett's book (Fausett [2008]).

Fausett's book has a list of 16 exercises called *Practice the Techniques* Fausett [2008] at the end of the chapter on *interpolation.* The section is inclusive of all topics that are described in chapter and about half of them are on cubic splines. In these exercises students are given a set of data and are required to interpolate it.

Evaluation 13.

In Chapra's book, the proposed exercises are given in section *Problems* at the end of the chapter. The list is not subdivided into parts, but it is clear from the text of the questions whether they are intended for paper-and-pen or for a computer solution. The first question gives simply data points which to be interpolated by hand, but the rest of the questions give more realistic data points intended to be solved with a calculator or computer, sometimes requiring MATLAB. In addition, within the list of exercises there are several applications for cubic splines, and these question cover different levels of difficulty, going from the easier to the more challenging. Moreover, the questions are not repetitive, but introduce different stages of difficulty, something that will help students to increase their level of understanding. Furthermore, no questions were outside the topic in the sense that they would need some research to solve them. Consequently, Chapra's book contains a list of exercises that support the student in reaching the required level of understanding, but the focus is more on numerical solutions using calculators and computers, specifically MATLAB.

Sauer's book offers a list of questions after the chapter on cubic splines. Thus, all exercises are geared towards cubic spline interpolation. The text of the questions is clear and easy to understand, and the level of the questions goes step by step from basic questions to a higher level. At the end there are a few more theoretical questions. Moreover, the questions cover all the material and are not repetitive. By solving the proposed exercises the student will arrive to a good understanding of all the details of cubic splines. Finally, there is no question that requires more research about cubic splines.

Sauer's book offers questions exercises in a good way as well as Chapra's book.

Kincaid and Cheney's book has an extensive list of proposed exercises at the end of the chapter on spline interpolation, and the list is split into Problems and Computer Problems. The first two exercises require proofs for some aspects of the derivation of the equations for the spline coefficients. The presentation of the exercises is not arranged in terms of difficulty, but the questions, which cover all of the theory on cubic splines in this book, do not seems to have any precise order. There are questions about quadratic splines, which is not a topic discussed in the chapter of spline interpolation. This and other questions require some research on the part of the students, but can be confusing as not support or suggestions are given. Many of the exercises are slight variations of one another, but there is no exact repetition. This should be a good way to lead the student into the more difficult parts of the topic.

Fausett presents several list of exercises, in particular, one called *Practice the Techniques*, another called *Computer Investigations*, then *Explore Some Applications*, and finally *Extend Your Understanding*. The first section lists many exercises of spline interpolation, but for instance the first eight are the same except for different sets of data. In general, the exercises in this section are very repetitive. This strategy does not help the student to understand the topic, but rather seems like a boring repetitive chore. Because there are so many similar exercises, this leads to a lack of coverage of all details of spline interpolation. In the list called *Explore Some Applications* many of the questions are uninteresting exercises of the same nature as in the previous list, that is, a direct exercise on cubic spline interpolation. Finally, the last list of questions has the most interesting exercises, requiring some proofs or some research or numerical experimentation.

My assessment is that Chapra and Sauer have the best lists of exercises, that help students reach the goal of the chapter.

Question 14 Is there a set of computer questions?

I. Steven C. Chapra's book (Chapra [2011]).

   The computer exercises in this book were included in a general list of problems (13). A few questions ask the student to use MATLAB to construct computer codes for the solution of the given problems.

II. Timothy Sauer's book (Sauer [2012]).
   In this book there is a section called *Computer Problems* after the section on cubic splines. It consists of 15 problems, of which eight require the construction of a cubic

spline with some end condition that interpolates a given set of data. The other problems are applications to population, economy, chemistry and weather.

III. David Kincaid and Ward Cheney's book (Kincaid and Cheney [2002]).

In Kincaid and Cheney's book eight computer problems are offered separately. Some of these problems require algorithm construction and their implementation. Half of the proposed problems were about cubic splines.

IV. Laurene V. Fausett's book (Fausett [2008]).

In Fausett's book there is a section called *Computer Investigations*. Here the student is asked to compute the cubic spline interpolant for five given data sets. There are also problems about Hermite interpolation and polynomial interpolation. Finally, eight different sets of data are given, and the student is required to use MATLAB built-in functions to produce and plot interpolating cubic splines.

Evaluation 14.

Chapra's book offers some computer exercises in the *Problems* section. The exercises can be calculated by using a calculator or by using programming. In both cases solving the exercises will reinforce the knowledge students have and also give them the chance to improve their computing skills. The author presents the exercises in a way that allows students to understand all the material of spline interpolation. Even though the computer exercises are included in the same list as other type of exercises, it should be clear for the student when using a calculator or a computer is appropriate.

Sauer's book presents the computer exercises in their own section titled *Computer Problems*. The text of the exercises is clear to understand and the level of the questions go step by step level from beginner level to a higher level. The questions cover all of cubic splines in this book. Some of the questions appear to be somewhat repetitive, but this repetition is more of a variation and will help students to understand the differences between the different cubic spline end conditions. Here there are also some questions that point to applications of cubic spline interpolation, and in this point it is similar to Chapra's book, showing how one can solve problems from real-life by using cubic spline interpolation.

Kincaid and Cheney's book presents computer exercises in a *Computer Problems* section. The questions are well explained and each one is very different from the others, but the level of the questions is often high. This book has a very different style from the two books mentioned previously, as these questions are more intended to be a challenge for students who wish to increase their level of understanding.

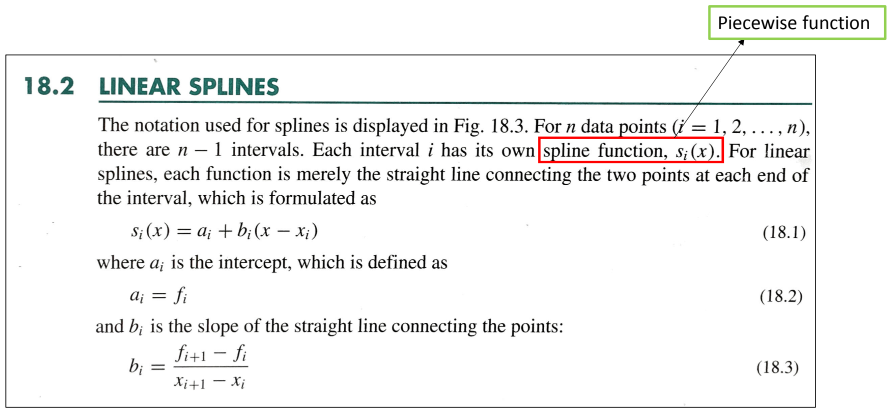
Fausett's book has a section called *Computer Investigations*. Here the text of the questions is quite easy to understand, but there is much repetition. The reader is expected to solve the same problem with different data, and it is the same type of questions posed in *Practice the Techniques*, except that it was a few more data points. The computer questions cover material from the chapter on *Interpolation*, but most questions allude to cubic spline interpolation using MATLAB. There are no questions that challenge the students to do some research or experimentation.

My assessment is that Sauer's book offers the most complete list of computer exercises that will lead students to a better understanding of how to program cubic splines.

To summarize these observations we have constructed a table that summarizes the questions and answers.

Comparison of books on the topic of *cubic splines*					
	Comparison criteria	Steven C. Chapra	Timothy Sauer	David Kincaid, Ward Cheney	Laurene V.Fausett
1	Insertion point	➢ *Splines and Piecewise Interpolation* • Introduction to splines splines • Linear splines • Quadratic splines	➢ *Interpolation* • Lagrange interpolation • Interpolation errors • Chebyshev interpolation	➢ *Approximating functions* • Polynomial interpolation • Divided differences • Hermite interpolation	➢ *Interpolation* • Lagrange interpolation • Hermite interpolation • Piecewise (linear, quadratic) interpolation
2	Interconnection	Runge's function	None	None	Runge's function
3	Motivation	Practical uses	Alternative to Lagrange	Practical uses	Possibility to improve on previous interpolants
4	Definition	Intuitive for general splines	Form and properties	Formal definition for splines of general degree	No explicit definition
5	Properties	No formal stating of properties	Detailed presentation	For splines of general degree	Informally stated along the text
6	Derivation of formulas	Dedicated subsection. Starts with general form and requires conditions to be fulfilled	For the natural cubic spline. Starts with general form and requires conditions to be fulfilled	Starts with second derivative and integrates.	Gives two alternative derivations: starting from the general form, and staring from the second derivative.
7	Examples	• How to fit a cubic spline to data, and then evaluate the spline at a point • Use of Matlab built-in functions	• Same data for linear and cubic splines • Determine if all properties of cubic splines are satisfied • Calculation of an interpolating cubic spline, in detail	No examples	• Calculation for an interpolating cubic spline • Cubic spline for Runge's function

8	End conditions	➢ Quadratic splines: one end condition ➢ Cubic splines • Natural • Clamped • Not-a-knot	• Natural • Curvature-adjusted • Clamped • Parabolically terminated • Not-a-Knot	Natural cubic splines	➢ Quadratic splines ➢ Natural cubic splines • Clamped
9	Theoretical content	Informal, no theorems	➢ Uses theorem from previous chapter ➢ Theorem on uniqueness of natural cubic splines ➢ Theorem on uniqueness of splines with other end conditions	➢ Theorem on the optimality of natural cubic splines	Informal, no theorems
10	Programming	➢ Matlab programs ➢ Lines of Matlab code ➢ Presented as tools for solving particular problems	➢ Matlab programs ➢ Codes that require modifications ➢ Dynamic integration to theory	➢ Pseudocodes not directly related to splines ➢ Secondary value	➢ Matlab programs and scripts ➢ Presented as tools for a particular problem
11	Summary	None	None	None	➢ Lagrange interpolation formula ➢ Hermite basis ➢ Natural cubic spline formula
12	Applications	➢ In *Case study* section, detailed application to fonts ➢ In Problems section	Only as proposed computer problems	One proposed problem about fonts	Dedicated section of proposed applications
13	Proposed exercises	14 questions	21 questions	36 questions	8 questions
14	Proposed computer exercises		15 questions	8 questions	21 questions



18.2 LINEAR SPLINES

The notation used for splines is displayed in Fig. 18.3. For n data points ($i = 1, 2, \ldots, n$), there are $n - 1$ intervals. Each interval i has its own spline function, $s_i(x)$. For linear splines, each function is merely the straight line connecting the two points at each end of the interval, which is formulated as

$$s_i(x) = a_i + b_i(x - x_i) \quad (18.1)$$

where a_i is the intercept, which is defined as

$$a_i = f_i \quad (18.2)$$

and b_i is the slope of the straight line connecting the points:

$$b_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} \quad (18.3)$$

Figure 3.6: linear spline section ([Chapra, 2011], p. 431).

3.1 Points in need of revision

In this section we call attention on issues that may be corrected in the books we have studied. Additionally, we will give our own suggestions on how to improve or correct these presentations. These observations have been used later in the construction of our own chapter on splines.

I. Steven C. Chapra's book Chapra [2011].

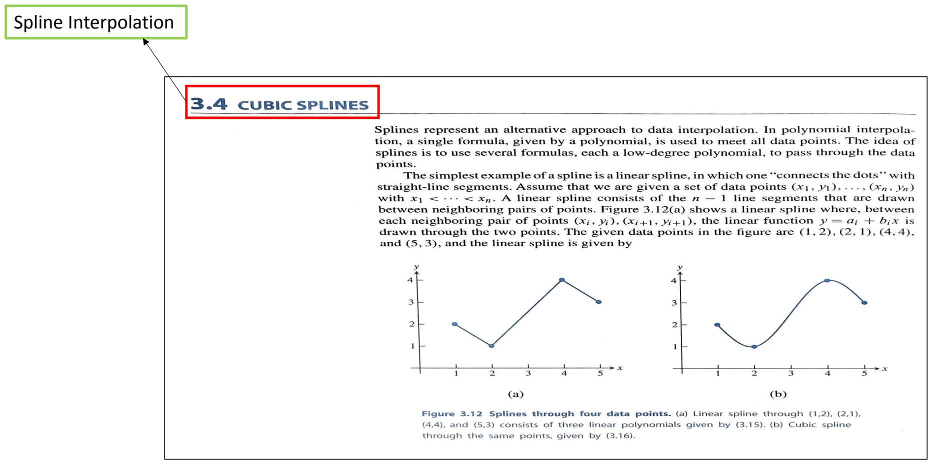
   In the section *Linear Splines* of this book, the author denotes $s_i(x)$ as a spline function, but this is confusing, as it is the entire spline that is a function while s_i is just part of this function in a subinterval, and it is a polynomial. This notation does not help the student to understand the difference between a piecewise polynomial and a polynomial piece of this function.

II. Timothy Sauer's book Sauer [2012].

   In Sauer's book we observe that the section named *Cubic Splines* contains the topic of *linear splines*. The first thing that the student encounters when reading the chapter on cubic splines is the topic of linear splines. We believe that calling the section *Splines* would have been a better choice, and then *Linear Splines* and *Cubic Splines* could have been two different subsections.

III. Davis Kincaid and Ward Cheney's book Kincaid and Cheney [2002].

   In Kincaid and Cheney's book two pseudo codes are presented. The first one evaluates a linear spline function. The code is not easy to understand. The reasons for parenthesis is not clear. Also, the structure of the algorithm is contrived. We modified the pseudo code as is shown in Figure 3.8. Figure (a) shows the pseudo code from the book by Kincaid and Cheney, and Figure (b) shows our modified pseudo code.



3.4 CUBIC SPLINES

Splines represent an alternative approach to data interpolation. In polynomial interpolation, a single formula, given by a polynomial, is used to meet all data points. The idea of splines is to use several formulas, each a low-degree polynomial, to pass through the data points.

The simplest example of a spline is a linear spline, in which one "connects the dots" with straight-line segments. Assume that we are given a set of data points $(x_1, y_1), \dots, (x_n, y_n)$ with $x_1 < \cdots < x_n$. A linear spline consists of the $n - 1$ line segments that are drawn between neighboring pairs of points. Figure 3.12(a) shows a linear spline where, between each neighboring pair of points $(x_i, y_i), (x_{i+1}, y_{i+1})$, the linear function $y = a_i + b_i x$ is drawn through the two points. The given data points in the figure are $(1, 2), (2, 1), (4, 4)$, and $(5, 3)$, and the linear spline is given by

(a) (b)

Figure 3.12 Splines through four data points. (a) Linear spline through (1,2), (2,1), (4,4), and (5,3) consists of three linear polynomials given by (3.15). (b) Cubic spline through the same points, given by (3.16).

Figure 3.7: Cubic spline section ([Sauer, 2012], p. 166)

```
input (t_i), (a_i), (b_i), x, n
for i = 1 to n − 1 do
    if x < t_i then
        S(x) = a_{i−1}x + b_{i−1}
        output S(x)
        exit loop
    end if
end do
S(x) = a_{n−1}x + b_{n−1}
output S(x)
```

(a)

```
input t_i, a_i, b_i, x, n
if x > t_n then
  S = a_{n−1}x + b_{n−1}
else
  for i = 1 to n − 1 do
    if x < t_i then
      S = a_{i−1}x + b_{i−1}
      exit loop
    end if
  end for
end if
output S
```

(b)

Figure 3.8: (a) pseudo code to evaluate a linear spline in ([Kincaid and Cheney, 2002], p. 350). (b) A modification of the pseudo code.

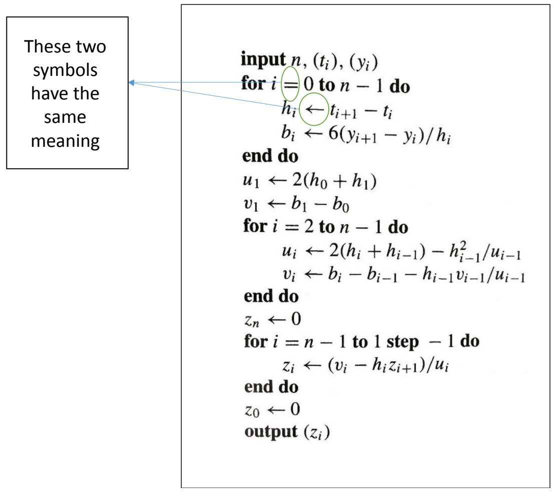


Figure 3.9: Pseudo code of Gaussian elimination for a tridiagonal system ([Kincaid and Cheney, 2002], p.353)

Secondly, the book shows a pseudo code for the Gaussian elimination algorithm of a tridiagonal system. There are different symbols in this pseudo code that may cause confusion in a student, as it is not clear what is meant by the symbol $\longleftarrow$. Our understanding is that it has the same meaning as the equal sign. In Figure 3.9 we can see the use of this symbol.

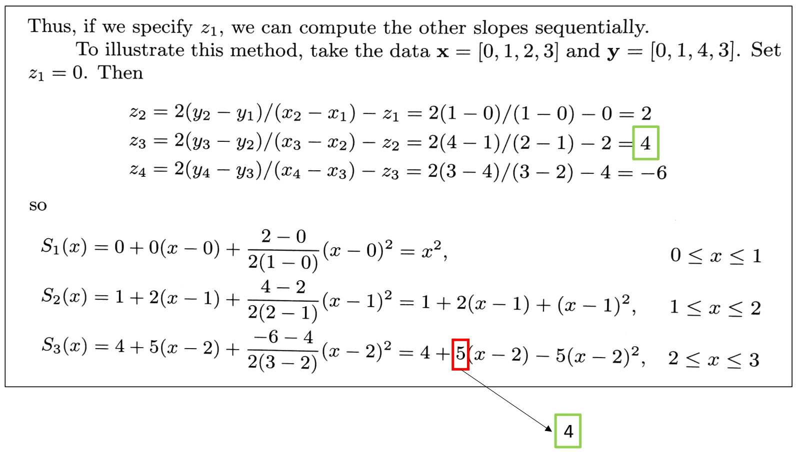
IV. Laurene V. Fausett's book Fausett [2008].

In this book there are several issues that we list below.

(a) In the section *Piecewise Quadratic Interpolation* Fausett shows an example of a quadratic spline when the nodes and the knots are the same. In this example, the author calculates the coefficients of each quadratic piece. There is an error in the formula for the third piece, as shown in Figure 3.10

Students will not notice the error unless they repeat the calculations or plot the spline. The plot of function with the error and the correction is shown in Figure 3.11.

(b) In the presentation of quadratic splines the author explains two different possibilities of defining a quadratic spline, as has been stated before; mainly, "nodes=knots" and "nodes $\neq$ knots." Although Fausett explains these two cases it is not clear why she labels the case "nodes $\neq$ knots" as being *more balanced*. There is no explanation as to the meaning of the word *balance* in this context. We presume she means that the second case will result in two end conditions, as opposed to only one. But when we asked a student that had already studied splines how she interpreted Fausett's assertion, she said that looking at the plots of the two splines she sees that the spline with "nodes $\neq$ knots" is better than the spline with "nodes = knots" because the plot of the first one does not go so high up as that of the



Thus, if we specify z_1, we can compute the other slopes sequentially.

To illustrate this method, take the data $\mathbf{x} = [0, 1, 2, 3]$ and $\mathbf{y} = [0, 1, 4, 3]$. Set $z_1 = 0$. Then

$$z_2 = 2(y_2 - y_1)/(x_2 - x_1) - z_1 = 2(1-0)/(1-0) - 0 = 2$$
$$z_3 = 2(y_3 - y_2)/(x_3 - x_2) - z_2 = 2(4-1)/(2-1) - 2 = 4$$
$$z_4 = 2(y_4 - y_3)/(x_4 - x_3) - z_3 = 2(3-4)/(3-2) - 4 = -6$$

so

$$S_1(x) = 0 + 0(x-0) + \frac{2-0}{2(1-0)}(x-0)^2 = x^2, \quad 0 \le x \le 1$$
$$S_2(x) = 1 + 2(x-1) + \frac{4-2}{2(2-1)}(x-1)^2 = 1 + 2(x-1) + (x-1)^2, \quad 1 \le x \le 2$$
$$S_3(x) = 4 + 5(x-2) + \frac{-6-4}{2(3-2)}(x-2)^2 = 4 + 5(x-2) - 5(x-2)^2, \quad 2 \le x \le 3$$

Figure 3.10: Calculation of quadratic splines ([Fausett, 2008], p. 301).

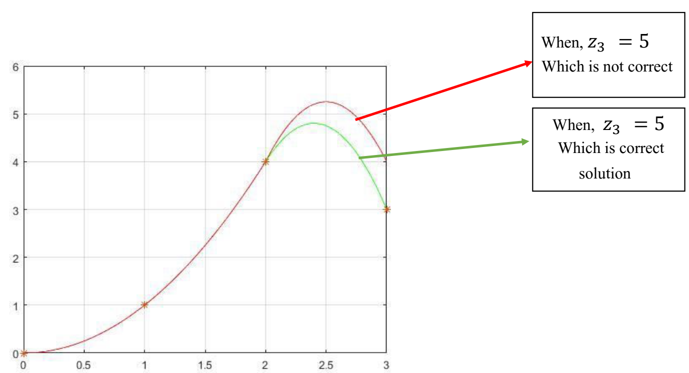


Figure 3.11: Error in the calculation of a quadratic spline in [Fausett, 2008], p. 301

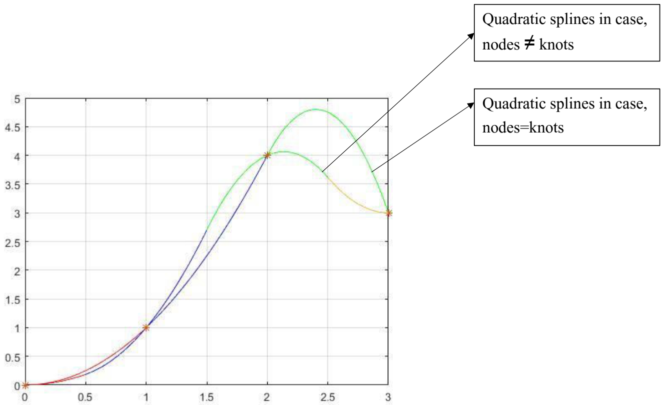


Figure 3.12: Two types of quadratic splines

second one, as shown in Figure 3.12. This answer shows us that the term *balance* does not convey the author's meaning. Consequently, we consider the discussion of these two cases should center on the fact that in the case "nodes = knots" the choice of end conditions has only one degree of freedom while in the other case there are two. This means that in the first case there would be a choice to give the slope for either the initial endpoint or the final endpoint, but not both, while in the second case the slopes at the two endpoints can be given.

(c) In the Piecewise Quadratic Interpolation section the author presents two different types of quadratic splines, mainly, "nodes = knots" and "nodes $\neq$ knots" (Fausett [2008], p. 301–303). In the presentation of the first case the author names the slopes z_i. In the presentation of the second case, z_i represent the knots. The presentation of the two cases appear in the same section, without any clear separation, and thus the use of the same letter to represent two different variables is very confusing. Also, in Example 8.8 the author calculates the coefficients of a quadratic spline without specifying to which of the two types it corresponds.

(d) The author sometimes uses $S(x)$ and some time $P(x)$ to refer to a spline. It would be clearer to use the same letter in all cases.

(e) On page (Fausett [2008] p.302-303) the author explains that the case "nodes $\neq$ knots" is more balanced than the case "nodes = knots", but that the linear system for the first case is not a tridiagonal system nor a banded matrix as shown in Figure 3.13. Nevertheless, when we reorder the equations of this system we get a banded matrix, as shown in equation (X.26).

In the chapter that follows we have made use of our observations as stated in the previous chapters to construct a new chapter that covers the topic of Cubic Spline Interpolation.

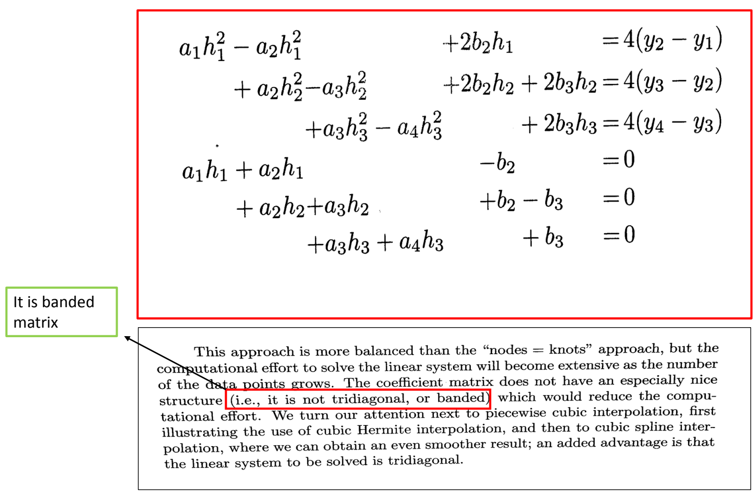


Figure 3.13: The linear system that point to not banded matrix.

Chapter 4

A suggestion of a chapter for teaching cubic spline interpolation

In Chapter 2 of the thesis we studied four different approaches to the teaching of cubic splines. Here we would like offer our own ideas on how to elaborate a chapter on this topic. We will use the same criteria we employed in Chapter 3 to analyze and compare those four books.

The first criterion is *insertion point*. We start by describing what is interpolation in general. Starting with the basic idea will guarantee that the student starts his learning journey with the right foot. We then show the student the different types of interpolation, and how they depend on the types of basis functions that are chosen. In this presentation we use variation theory when we show plots of different types of interpolation for temperature data of the city of Lund. We also use variation theory when we show that choosing different bases will not alter the result. In the second part of the *insertion point* we demonstrate *Runge's phenomenon* in order to illustrate the behavior of polynomials of high degree. The question of how to find solution for this problem will motivate cubic spline interpolation. In showing this phenomenon we also use variation theory by showing two the interpolation of Runge's function with polynomials of two different degrees. In the third part of the *insertion point* we present piecewise polynomial interpolation of different degrees. We will start with linear piecewise polynomial and develop the reasons why we should increase the degree of the splines unril we get to cubic spline interpolation. By presenting an example and showing how spline interpolation of different degrees work for the same data, we also use variation theory. Our overview should lead to the answer of why we are interested in cubic splines.

The second criterion is *teaching structure*. Using variation theory we present this topic using the same style as for the other types of piecewise interpolation. When compared to lower degree spline interpolation, this will call attention to what is similar and what is different for cubic spline interpolation. During the construction of the equations that govern the coefficients of a cubic spline we demonstrate different types of end conditions. Finally, by using the same data that was used before we show a plot a of cubic spline interpolation that can be compared to the other kinds of interpolation.

The third criterion is *outcome procurement*. Here we present some history of cubic splines. We also present some personal history from our experience, using variation theory in this way too. In the course Seminar in Numerical Analysis in 2015 at Lund University, we were shown Figure X.18. This figure show how to construct a natural cubic spline by using "ducks".

Figure X.19 shows a course in 1941 where students were trying to construct a cubic spline by using “ducks.” The teacher of course in Lund University motivated his students by helping them understand how hard it was to construct a cubic spline 74 years ago. Some books mention historical facts at the beginning of the chapter by reminding the reader that cubic splines were used for the construction of ships, but we believe looking at a single application in the past will not heighten the interest of present day students.

Here we should also include exercises for this chapter. To construct these exercises we ask ourselves the following questions:

1. Are the text of the questions offered in way that can be understandable or difficult to understand by students?
2. Are there questions of every level?
3. Do the questions cover all aspects of the details of cubic splines?
4. Are the questions too repetitive, without introducing different aspects?
5. Are there questions outside the topic that need to be researched?
6. What suggestions can we make to improve the list of exercises?

At this stage, we have decided not to include a list of exercises, as this would a task that would prolong the thesis beyond its pretended scope.

The figure in the next page shows a scheme of the content and organization of our proposed chapter for the teaching of cubic splines. What follows is our proposed chapter, which we have called **Chapter X**.

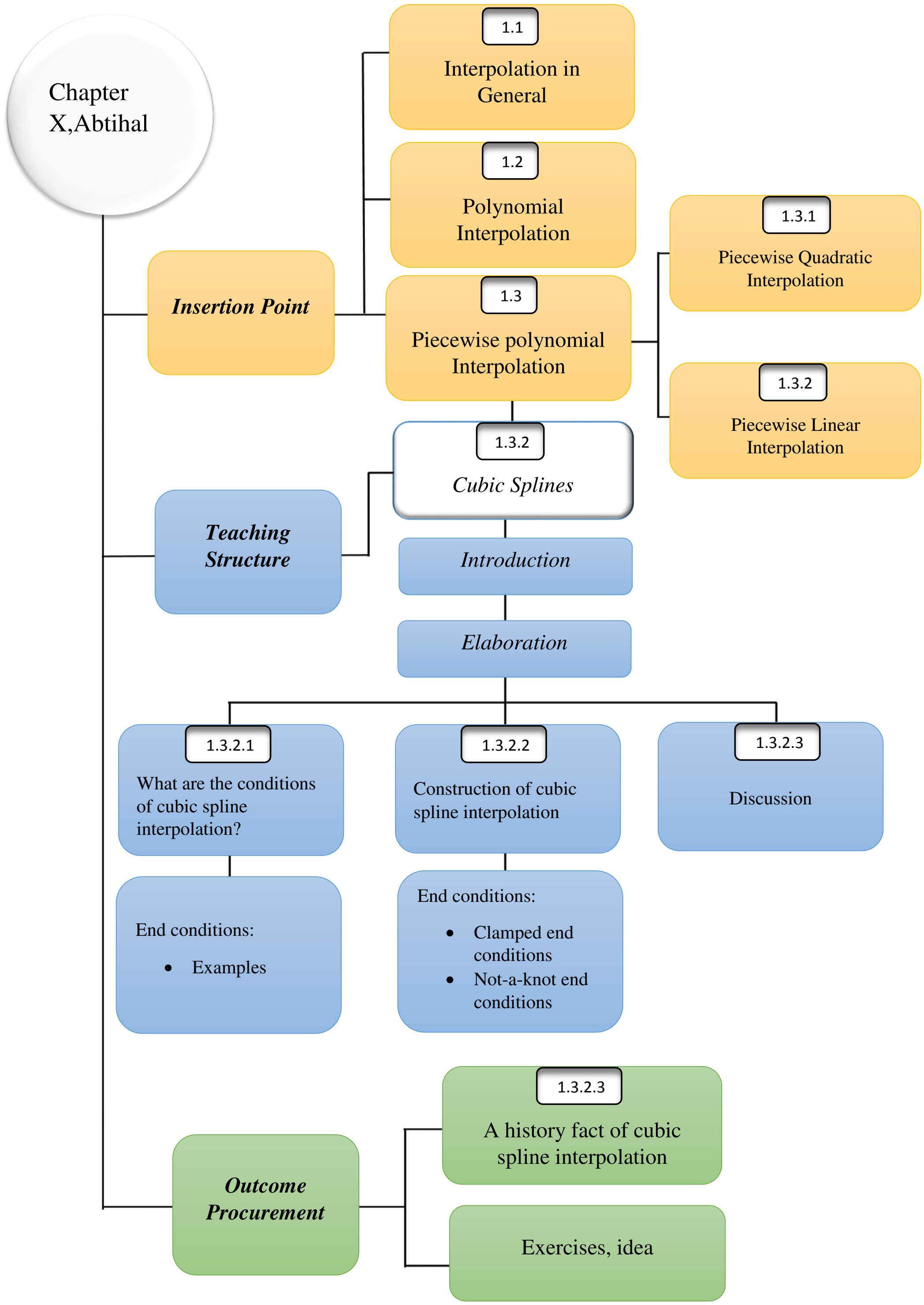


Figure 4.1: Schematic content and organization of Chapter four “ A suggestion of a chapter for teaching cubic spline interpolation.”

CHAPTER X:
PIECEWISE POLYNOMIAL INTERPOLATION

X.1 Interpolation in general

X.1.1 What is interpolation?

Given some data points, e.g., hourly temperature measurements, daily currency values, one looks for a function passing through this data. Such a function gives a continuous representation of the process which might have generated the data. The task to generate such a function and to evaluate it is called interpolation.

Definition 1. *The function $y = f(x)$ is said to* interpolate *the points* $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ *if*

$$f(x_i) = y_i \quad \text{for} \quad i = 0, 1, \ldots, n.$$

Clearly, there is not only one function interpolating given data. In Figure X.2 we show different functions interpolating weather data in Lund, Sweden. The figure shows two curves with different degrees of smoothness. One of these two curves attains also values which are obviously physically incorrect. In the following section we will focus on interpolation with polynomials and with piecewise polynomials.

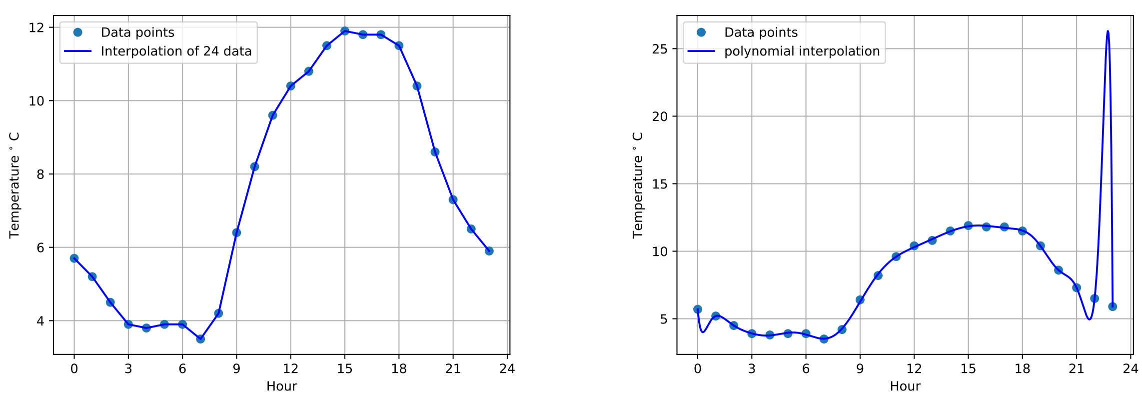


Figure X.2: Different interpolating functions of temperature data in Lund (Sweden) on April 4, 2017.

X.1.2 Why do we study interpolation?

There are many applications that use polynomial interpolation like:

1. **Computer graphics:** plotting a smooth curve that passes through some discrete data points. For example, the shape of tea bottles as show in Figure X.3

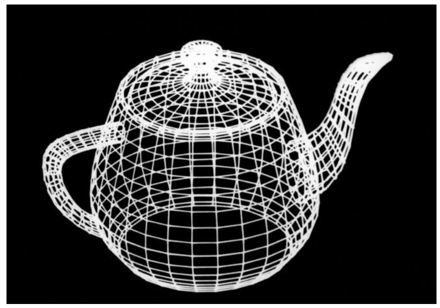


Figure X.3: Example of computer graphics by using interpolation, Introduction to Computer Graphics [2017].

2. **Evaluation of a mathematical function:** evaluation of mathematical functions, for example, sin, cos, log, exp, etc., see Fig. X.4.

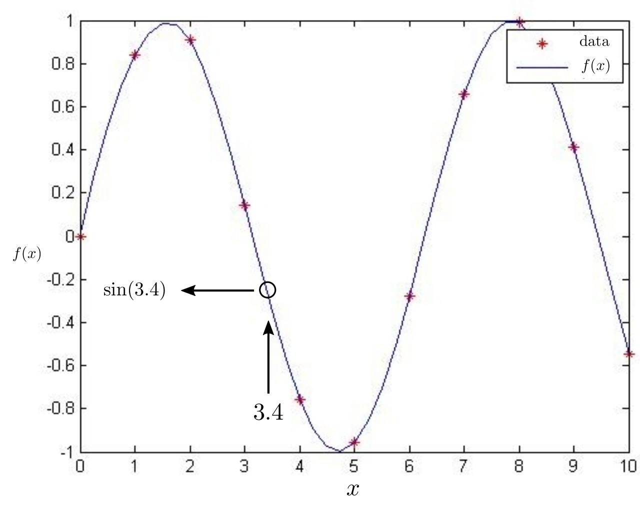


Figure X.4: Evaluating an interpolating function.

3. **Numerical Integration:** substitution of difficult integrands by others that are easier to compute.

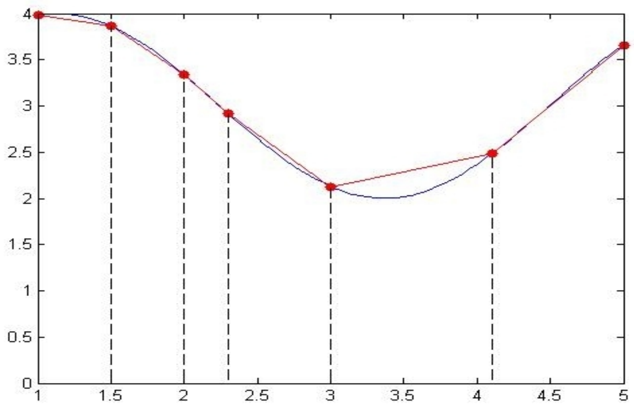


Figure X.5: The approximation of an integral by the area under six straight-line segments.

4. **Extrapolation:** estimation of a value outside the interval where the data points lie in order to predict values. For example, what is the expected temperature for tomorrow.

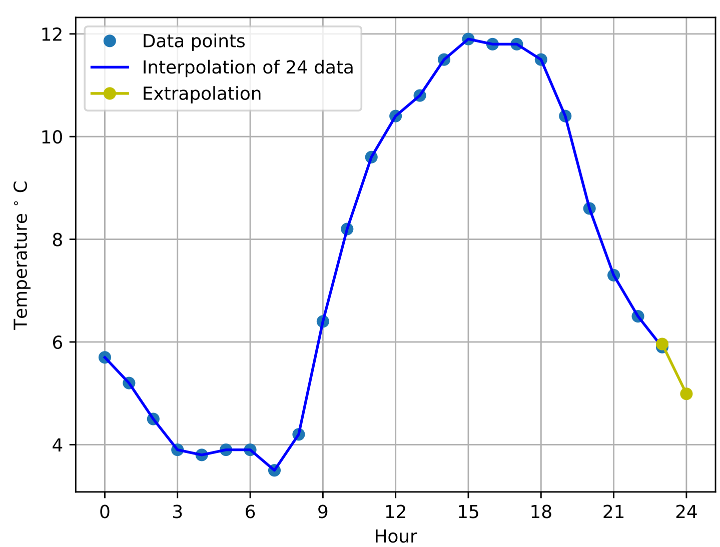


Figure X.6: Extrapolation of temperature data for the city of Lund, Sweden in April 4, 2017.

There are some important points we have to understand when doing interpolation. Firstly, we need to choose an adequate number of data points to meet accuracy requirements. Secondly, a family of interpolation functions, eg., polynomials, has to be chosen to meet requirements like smoothness, easiness of evaluation, periodicity, etc.

X.1.3 Types of interpolating functions

Given a set of data points there are different curves passing through those points. Therefore, there are different types of interpolating functions, and the type of interpolant will depend on the characteristics of the data being fit, the required smoothness of the curve, the trending behavior of the data (periodicity), and how much accuracy we demand, among other things. The most well-used types of interpolating functions are:

1. Polynomials
2. Piecewise polynomials
3. Trigonometric functions
4. Exponential functions
5. Rational functions

We will focus on polynomial and piecewise polynomial interpolation, because these functions are very easy to handle, can be easily differentiated and integrated, and are hopefully flexible enough. Also, there are well-known methods for interpolating with these type of functions.

X.2 Polynomial Interpolation

A set of data points consists of $n+1$ pairs $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$. We assume the x_i to be real and distinct. Polynomial interpolation is a way to fill gaps between known data points by creating a continuous representation. There exists a unique polynomial of degree at most n that passes through these $n+1$ points.

Given a basis $p_0, p_1, ..., p_n$ of the space of polynomials of degree at most n, the *interpolation polynomial* can be written as a linear combination of the basis:

$$P_n(x_j) = \sum_{i=0}^{n} a_i p_i(x_j) \tag{X.1}$$

The coefficients a_i are determined from the interpolation conditions, i.e.,

$$P_n(x_j) = y_j, \quad j = 0, 1, 2, \dots, n \tag{X.2}$$

The meaning of Eq. (X.2) is that the polynomial curve must go through all points in the data set. To determine the coefficients a_i the Eqs. (X.1) and (X.2) are written as a linear equation system with the coefficients a_i as unknowns:

$$\begin{pmatrix} p_n(x_0) & p_{n-1}(x_0) & \cdots & p_0(x_0) \\ p_n(x_1) & p_{n-1}(x_1) & \cdots & p_0(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ p_n(x_n) & p_{n-1}(x_n) & \cdots & p_0(x_n) \end{pmatrix} \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \tag{X.3}$$

If the matrix is non-singular, i.e., invertible, this linear system can be solved uniquely for the coefficients $a_0, a_1, \ldots, a_n$. These values determine the unique polynomial of degree at most n that interpolates the given data.

We consider two different choices of a basis:

1. **Monomial Basis**

$$p_i(x) = M_i(x) = x^i, \qquad i = 0, 1, \ldots, n$$

2. **Lagrange Basis**

$$p_i(x) = L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 0, 1, \ldots, n \tag{X.4}$$

Note that the symbol $\prod$ denotes multiplication in the same way as the usual notation $\sum$ denotes summation.

X.2.1 Monomial Basis

The basis functions are defined as

$$p_i(x) = M_i(x) = x^i, \qquad i = 0, 1, ...n \tag{X.5}$$

Given $n+1$ distinct nodes $x_0, x_1, ..., x_n$, the interpolation conditions (X.2) take the form:

$$\begin{cases} P_n(x_0) = y_0 : \quad a_n x_0^n + a_{n-1} x_0^{n-1} + ... + a_1 x_0 + a_0 = y_0 \\ P_n(x_1) = y_1 : \quad a_n x_1^n + a_{n-1} x_1^{n-1} + ... + a_1 x_1 + a_0 = y_1 \\ \vdots \\ P_n(x_n) = y_n : \quad a_n x_n^n + a_{n-1} x_n^{n-1} + ... + a_1 x_n + a_0 = y_n \end{cases} \tag{X.6}$$

or, alternatively, as a matrix-vector system,

$$\underbrace{\begin{pmatrix} x_0^n & x_0^{n-1} & \cdots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \cdots & x_1 & 1 \\ \vdots & \vdots & \ddots & & \vdots \\ x_n^n & x_n^{n-1} & \cdots & x_n & 1 \end{pmatrix}}_{V} \underbrace{\begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix}}_{a} = \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}}_{y} \tag{X.7}$$

The matrix V is called a *Vandermonde* matrix. The advantage of interpolation with the monomial basis is that the basis is independent of the data, and the system matrix is easy to set up; the disadvantage is that the Vandermode matrix tends to have a large condition number with increasing number of data points and that the linear system $Va = y$ is expensive to be solved as V is a full matrix.

X.2.2 Lagrange Basis

We take now an alternative approach to compute the interpolation polynomial by using the Lagrange basis. The Lagrange polynomials L_i are polynomials of degree n that satisfy

$$L_i(x_j) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

which can easily be verified from (X.4).

Because of this property, the system matrix in (X.3) with the basis functions $p_i(x) = L_i(x)$ becomes the identity matrix

$$\underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix}}_{a} = \underbrace{\begin{pmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_0 \end{pmatrix}}_{y}.$$

Thus, the Lagrange form of the interpolation polynomial can be written as:

$$P_n(x) = \sum_{i=0}^{n} y_i L_i(x) \tag{X.8}$$

with

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 0, 1, \ldots, n$$

It is easy to check the interpolation property of the Lagrange interpolation polynomial:

$$P_n(x_j) = \sum_{i=0}^{n} y_i L_i(x_j) = y_j, \qquad j = 0, 1, \ldots, n$$

The advantage of Lagrange interpolation is that one obtains the interpolation polynomial without solving a linear system of equations, and the Lagrange form is numerical stable. The disadvantage of Lagrange interpolation is that it is expensive to evaluate the polynomials because of their data dependent basis polynomials.

Example 1. *We consider a measured voltage drop V_i across a resistor for a number of different values of the current x_i, and construct a polynomial interpolating this data. This polynomial will then be used to compute the voltage drop at $x = 1.15$ A. The given data is [Chapra, 2011]:*

Table X.1: Measured data across a resistor

x_i	0.25 A	0.75 A	1.25 A	1.5 A	2.0 A
$V(x_i)$	-0.45 V	-0.6 V	0.70 V	1.88 V	6.0 V

Given 5 points we look for a polynomial of degree 4 and express it in the monomial basis $\{1, x, x^2, x^3, x^4\}$ *as*

$$P_4(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0.$$

In order to find the coefficients a_0, a_1, a_2, a_3, a_4 *we set up the interpolation conditions* (**??**).

$$\begin{aligned}
-0.45 &= a_4(0.25)^4 + a_3(0.25)^3 + a_2(0.25)^2 + a_1(0.25)^1 + a_0 \\
-0.6 &= a_4(0.75)^4 + a_3(0.75)^3 + a_2(0.75)^2 + a_1(0.75)^1 + a_0 \\
0.70 &= a_4(1.25)^4 + a_3(1.25)^3 + a_2(1.25)^2 + a_1(1.25)^1 + a_0 \\
1.88 &= a_4(1.5)^4 + a_3(1.5)^3 + a_2(1.5)^2 + a_1(1.5)^1 + a_0 \\
6.0 &= a_4(2.0)^4 + a_3(2.0)^3 + a_2(2.0)^2 + a_1(2.0)^1 + a_0
\end{aligned}$$

Written as a linear equation system in matrix-vector form this gives:

$$\begin{pmatrix} (0.25)^4 & (0.25)^3 & (0.25)^2 & (0.25)^1 & 1 \\ (0.75)^4 & (0.75)^3 & (0.75)^2 & (0.75)^1 & 1 \\ (1.25)^4 & (1.25)^3 & (1.25)^2 & (1.25)^1 & 1 \\ (1.5)^4 & (1.5)^3 & (1.5)^2 & (1.5)^1 & 1 \\ (2.0)^4 & (2.0)^3 & (2.0)^2 & (2.0)^1 & 1 \end{pmatrix} \begin{pmatrix} a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} -0.45 \\ -0.6 \\ 0.70 \\ 1.88 \\ 6.0 \end{pmatrix}$$

This linear system of equations can be uniquely solved for the coefficients a_0, a_1, a_2, a_3, a_4. *As all* x_i *are distinct the Vandermonde matrix is invertible and the coefficients can be computed as*

- $a_0 = 0.88685714$
- $a_1 = -3.38438095$
- $a_2 = 7.3$
- $a_3 = -5.40447619$
- $a_4 = 0.49428571$

and the polynomial is

$$P_4(x) = 0.49428571x^4 - 5.40447619x^3 + 7.3x^2 - 3.38438095x + 0.88685714$$

The polynomial can be evaluated at $x = 1.15$,

$$P_4(1.15) = 0.49428571(1.15)^4 - 5.40447619(1.15)^3 + 7.3(1.15)^2 - 3.38438095(1.15) + 0.88685714 = 0.3372864$$

Example 2. *Now we solve the same problem as in Example 1 by using a Lagrange basis formulation for the same data, see Tab. X.1:*

Firstly, we set up the Langrange basis functions (X.4) *for this example,*

$$\begin{aligned}
L_0(x) &= \left(\frac{x-0.75}{0.25-0.75}\right)\left(\frac{x-1.25}{0.25-1.25}\right)\left(\frac{x-1.5}{0.25-1.5}\right)\left(\frac{x-2.0}{0.25-2.0}\right)\\
L_1(x) &= \left(\frac{x-0.25}{0.75-0.25}\right)\left(\frac{x-1.25}{0.75-1.25}\right)\left(\frac{x-1.5}{0.75-1.5}\right)\left(\frac{x-2.0}{0.75-2.0}\right)\\
L_2(x) &= \left(\frac{x-0.25}{1.25-0.25}\right)\left(\frac{x-0.75}{1.25-0.75}\right)\left(\frac{x-1.5}{1.25-1.5}\right)\left(\frac{x-2.0}{1.25-2.0}\right)\\
L_3(x) &= \left(\frac{x-0.25}{1.5-0.25}\right)\left(\frac{x-0.75}{1.5-0.75}\right)\left(\frac{x-1.25}{1.5-1.25}\right)\left(\frac{x-2.0}{1.5-2.0}\right)\\
L_4(x) &= \left(\frac{x-0.25}{2.0-0.25}\right)\left(\frac{x-0.75}{2.0-0.75}\right)\left(\frac{x-1.25}{2.0-1.25}\right)\left(\frac{x-1.5}{2.0-1.5}\right)
\end{aligned}$$

By using the Lagrange interpolation formula (X.8) *we get:*

$$P_4(x) = -0.45L_0(x) - 0.6L_1(x) + 0.70L_2(x) + 1.88L_3(x) + 6.0L_4(x)$$

For $x_i = 1.15$ *we obtain*

$$\begin{aligned}
P_4(1.15) &= -0.45L_0(1.15) - 0.6L_1(1.15) + 0.70L_2(1.15) + 1.88L_3(1.15) + 6.0L_4(1.15)\\
&= 0.3372864.
\end{aligned}$$

We observe that both examples give us the same result. As the interpolation polynomial is unique this was to be expected.

Polynomial interpolation in Python

For completeness we demonstrate these two examples also by giving a Python code which does the interpolation in both cases:

```
from scipy import *
import scipy.linalg as sl
from matplotlib.pyplot import *

current=[0.25, 0.75, 1.25, 1.5, 2.0]  # current data
voltage=[-0.45,-0.6,0.70,1.88,6.0]    # voltage drop data
plot(current,voltage,'*')
x_plot=linspace(current[0], current[-1] ,100)
# ---------------------Monomial Basis---------------------------
V=vander(current)               # Vandermonde matrix
coef=sl.solve(V,voltage)        # coefficients in monomial basis
y_plot=polyval(coef,x_plot)
plot(x_plot,y_plot)
p_i=polyval(coef,1.15)          # polynomial value at i=1.15
plot(1.15, p_i,'o')
plot([1.15,1.15],[-1,p_i],'-.')# plot the line between point 1.15 and x axis
xlabel('x'); ylabel('y')
print("""
Results
Vandermonde matrix {}
Coefficients {}
Polynomial Value {}""".format(V,coef,p_i))

#---------------------Lagrange Basis------------------------
def L(x,x_data,i):
    """
    Compute the i-th Lagrange basis function
    x_data are the data points
    """
    np1 = len(x_data)
    return prod([(x-x_data[j])/(x_data[i]-x_data[j])
                                   for j in range(np1) if j!=i])


def Lagrangeinterpolation(x,x_data,y_data):
    """
    Evaluate a polynomial at x
    x_data, y_data  interpoilation data
    """
    np1 = len(x_data)
    return sum([y_data[i]*L(x,x_data,i) for i in range(np1)])

order = len(current) - 1             # order of polynomial
y_plot=[Lagrangeinterpolation(x,current, voltage) for x in x_plot]
plot(x_plot,y_plot)
grid()
```

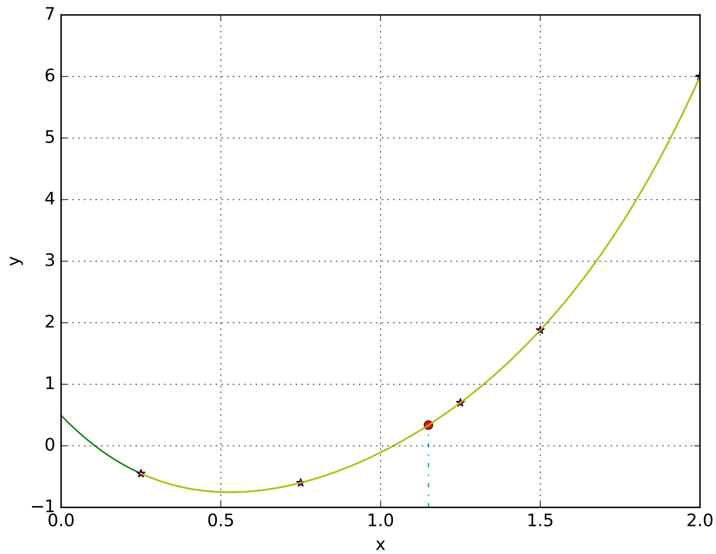


Figure X.7: Plot of the 4$^{\text{th}}$ degree polynomial in different bases: monomial and Lagrange basis give identical results.

Summary: how to construct interpolation polynomials

- The data points $(x_0, y_0), \ldots, (x_n, y_n)$, with x_i distinct, uniquely define a polynomial P_n of degree n that interpolates the data, i.e., $P_n(x_i) = y_i$.

- The same polynomial can be expressed in different bases. It can be represented in the monomial basis

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

or in the Lagrange basis

$$P_n(x) = y_0 \prod_{\substack{j=1 \\ j\neq 0}}^{n} \frac{x - x_j}{x_0 - x_j} + y_1 \prod_{\substack{j=0 \\ j\neq 1}}^{n} \frac{x - x_j}{x_1 - x_j} + \cdots + y_n \prod_{\substack{j=1 \\ j\neq n}}^{n} \frac{x - x_j}{x_n - x_j}.$$

The construction of an interpolation polynomial depends on the chosen basis: the resulting coefficients differ for different bases but the polynomial itself is always the same.

X.2.3 Runge's Phenomenon

This phenomenon was discovered by Carl David Tomle Runge in 1901. It is used to demonstrate a problem of higher order polynomial interpolation by considering the function

$$f(x) = \frac{1}{1 + 25x^2} \tag{X.9}$$

This function is commonly called Runge's function. Data points are sampled on an equidistant grid of x-values in the interval $[-1, 1]$. The more points sampled, the higher is the order of the interpolation polynomial. One observes large oscillations at both ends of the interval, causing a large interpolation error. This phenomenon demonstrates that high degree polynomial interpolation can become unsuitable for polynomial interpolation on equidistant data points. This phenomenon is demonstrated by the following Python code and the related figures.

```
from scipy import *
from matplotlib.pyplot import *


def Runge(x):
    """
    This is an implementation of Runge's function.
    """
    return 1./(1. + 25.*x**2)


knots=int(input('Number of knots\n'))     # asks for the number of knots
x_data = linspace (-1, 1, knots)     # interval [-1,1] in 100 space
y_data = [Runge(x) for x in x_data]

"""
Polynomial interpolation
"""
coef = polyfit (x_data, y_data, knots-1) # polyfit function use to generate-
                                          # coefficients of polynomial
x_plot = linspace (-1, 1, 100)
y_plot = polyval (coef, x_plot)    # polyval function use to generate-
                                 # a polynomial interpolation on x_plot
y_runge = [Runge(x) for x in x_plot]
plot(x_data, y_data, 'o', label="data points")
plot(x_plot, y_plot, label="polynomial interpolation")
plot(x_plot, y_runge, '--', label="Runge's function")
grid()
legend(loc="best")
```

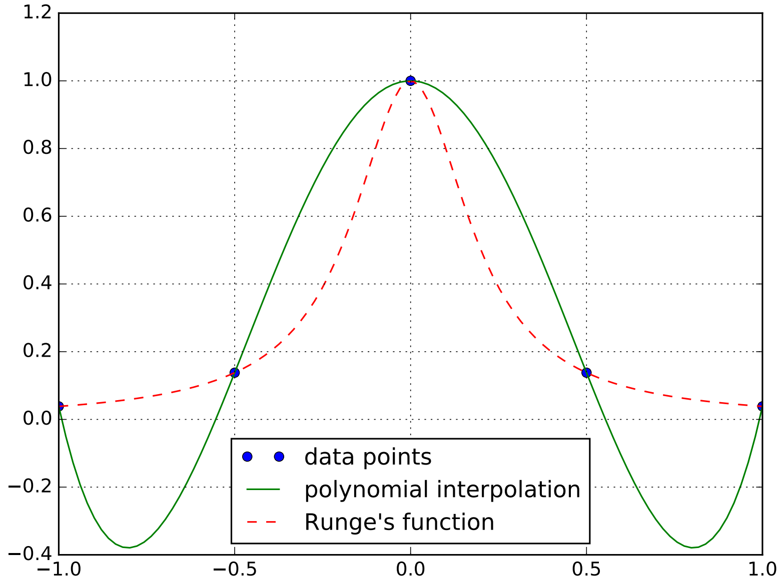


Figure X.8: Runge's example: interpolation using 5 equally spaced data points and a 4^{th} degree polynomial.

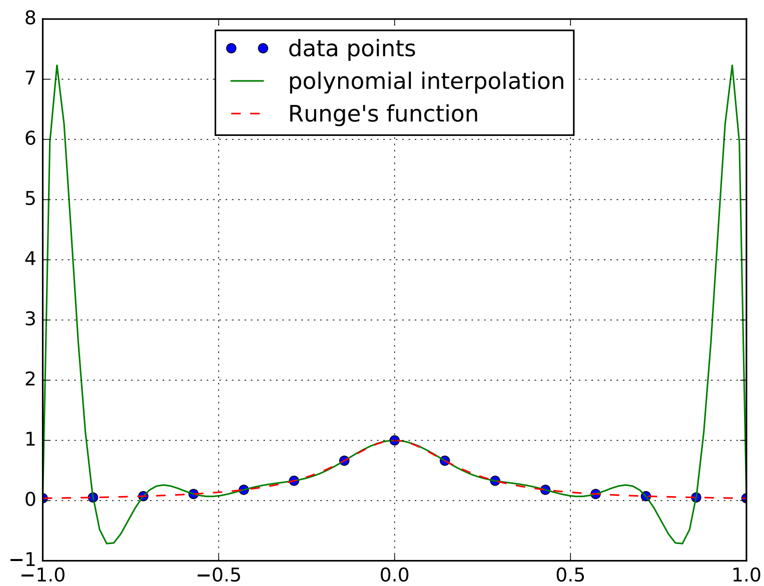


Figure X.9: Runge's example: interpolation using 15 equally spaced data points and a 14^{th} degree polynomial.

Note how the Python functions `polyfit` and `polyval` have been used. The function `polyfit` computes the polynomial coefficients and `polyval` evaluates the polynomial given by these coefficients at particular points.

There is a danger of producing high oscillations when interpolating a large amount of data, because higher degree polynomials tend to have this unwanted behavior. That motivates to introduce splines, and in particular cubic splines. These types of functions will be constructed by using only polynomials of low degree.

X.3 Piecewise polynomial Interpolation

To avoid the effect demonstrated by *Runge's function* in the last section we will use piecewise interpolation polynomials to handle interpolation tasks with a large number of data points.

In this section you will guided gradually starting with the simplest approach – piecewise linear polynomials – up to cubic splines, which are the most frequently used tool for interpolation of a larger set of data points.

X.3.1 Piecewise linear Interpolation

Constructing a curve by connecting two subsequent data points by a straight line is called linear interpolation. Children use this technique already in kindergarten as a training game, cf. Figure X.10.

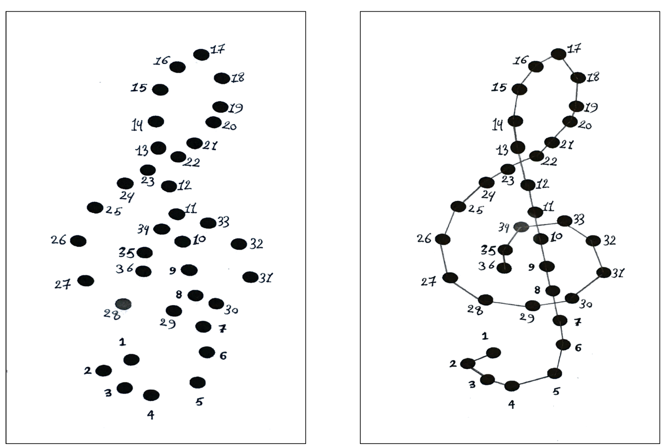


Figure X.10: Example for linear interpolation based on connecting numbered points to obtain the shape of the G-clef.

▶ **Mathematical Formulation of a piecewise linear function**

Let us assume a set of $n + 1$ data points,

$$(x_0, y_0), ..., (x_i, y_i), ...(x_n, y_n)$$

with $x_0 < \cdots < x_i < x_{i+1} < \cdots < x_n$. Then the linear function connecting (x_i, y_i) and (x_{i+1}, y_{i+1}) is

$$S_i(x) = a_i x + b_i \text{ on } [x_i, x_{i+1}) \tag{X.10}$$

where a_i, and b_i are coefficients determined by the interpolation conditions $S_i(x_i) = y_i$ and $S_i(x_{i+1}) = y_{i+1}$,

$$a_i x_i + b_i = y_i \tag{X.11}$$

$$a_i x_{i+1} + b_i = y_{i+1} \tag{X.12}$$

This gives

$$b_i = y_i - a_i x_i \tag{X.13}$$

Then, substituting (X.13) into (X.12) gives

$$a_i x_{i+1} + y_i - a_i x_i = y_{i+1} \tag{X.14}$$

and finally

$$a_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad b_i = \frac{y_i x_{i+1} - y_{i+1} x_i}{x_{i+1} - x_i}.$$

The function S defined by these polynomials,

$$S(x) = S_i(x) \text{ for } x \in [x_i, x_{i+1})$$

is called a linear piecewise polynomial.

By construction these piecewise polynomials are continuous. Continuous linear piecewise polynomials are called *linear splines*.

Substituting the values of the coefficients a_i, and b_i into Eq. (X.10) gives

$$S_i(x) = y_i \frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1} \frac{x - x_i}{x_{i+1} - x_i} \tag{X.15}$$

This is the formula for each linear spline segment. Written in a general way, for the entire interval,

$$S(x) = y_i \frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1} \frac{x - x_i}{x_{i+1} - x_i} \text{ for } x \in [x_i, x_{i+1}). \tag{X.16}$$

We see how easy the calculation of a linear interpolating spline is. The spline is fully determined by the data. No extra conditions are required to set up this function.

Example 3. *Again, we consider the weather data reported from Lund city(Sweden) on April, 4, 2017 given in Table X.2. The linear spline interpolating this data is displayed in Fig. X.11*

Table X.2: Temperature data in Lund on April 4, 2017

Hour	0	1	2	3	4	5	6	7	8	9	10	11
Temperature°C	5.7	5.1	4.5	3.9	3.8	3.8	3.9	3.5	4.2	6.4	8.2	9.6

Hour	12	13	14	15	16	17	18	19	20	21	22	23
Temperature°C	10.4	10.8	11.5	11.9	11.8	11.8	11.5	10.4	8.6	7.3	6.5	5.9

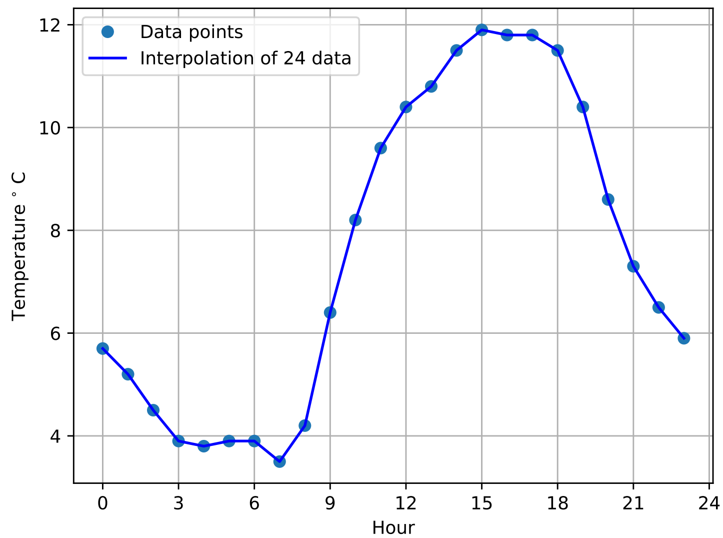


Figure X.11: Linear spline interpolating weather data given in Tab. X.2.

By construction, linear spline interpolation gives us a continuous function, but at the knots the first derivative is generally not continuous, as can be observed in this example. For many applications this curve is not smooth enough. This motivates us to study quadratic spline interpolation and cubic spline interpolation to gain more smoothness.

X.3.2 Quadratic Spline Interpolation

A quadratic spline segment is given by the formula

$$S_i(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i \quad \text{on} \quad [x_i, x_{i+1}), \tag{X.17}$$

which is written – as in the linear case – in a form centered around x_i. We have three unknowns per segment, namely a_i, b_i, c_i. If there are $n+1$ data points, $(x_1, y_1), \ldots, (x_{n+1}, y_{n+1})$, then the number of segments is n, so we have to find conditions to determine a total of $3n$ unknown coefficients.

There are $n+1$ interpolation conditions. Furthermore, asking for continuous and continuously differentiated segment connections we get $2n-2$ additional conditions. There is a total of $3n-1$ conditions for $3n$ unknowns, which allows us one free "wish". We can choose an extra condition in many different ways. Often, the slope at the left or at the right boundary is prescribed. This gives the following relations:

- Interpolation: $S_i(x_i) = y_i$, for $i = 1, \ldots, n$, and interpolation at the endpoint $S_n(x_{n+1}) = y_{n+1}$.
- Continuity: $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ for $i = 1, \ldots, n-1$.
- Continuous derivative: $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$ for $i = 1, \ldots, n-1$.

Thus, to find the coefficients of a quadratic spline segment we make use of these conditions. From the interpolation condition and the form of the spline segment function we obtain directly $c_i = y_i$ for $i = 1, \dots, n$. Then, we set $z_i = S_i'(x_i)$. As $S_i'(x_i) = b_i$ we obtain $b_i = z_i$ for $i = 1, \dots, n$. So, the continuity conditions for the first derivative gives

$$a_i = \frac{z_{i+1} - z_i}{2(x_{i+1} - x_i)}, \quad \text{for} \quad i = 1, \dots, n-1 \tag{X.18}$$

while the coefficient a_n of the last segment is obtained from the end interpolation condition:

$$a_n = \frac{\frac{y_{n+1}-y_n}{x_{n+1}-x_n} - z_n}{2(x_{n+1} - x_n)}. \tag{X.19}$$

To determine the remaining unknowns, z_i, we make use of the continuity condition which results in

$$z_{i+1} + z_i = 2\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad \text{for} \quad i = 1 \dots, n-1. \tag{X.20}$$

To simplify these expressions we introduce $h_i = x_{i+1} - x_i$ and write (X.20) in matrix-vector form

$$\begin{pmatrix} h_1 & & & & & 0 \\ h_2 & h_2 & & & & \\ & \ddots & \ddots & & & \\ & & h_j & h_j & & \\ & & & \ddots & \ddots & \\ 0 & & & & h_{n-1} & h_{n-1} \end{pmatrix} \begin{pmatrix} z_2 \\ z_3 \\ \vdots \\ z_{j+1} \\ \vdots \\ z_n \end{pmatrix} = 2 \begin{pmatrix} y_2 - y_1 - \frac{1}{2}h_1 z_1 \\ y_3 - y_2 \\ \vdots \\ y_{j+1} - y_j \\ \vdots \\ y_n - y_{n-1} \end{pmatrix} \tag{X.21}$$

You see that we left in the right-hand side of this equation the unknown z_1. We choose this as the additional degree of freedom, the "wish" we mentioned before. The formulation simplifies if we set the slope at the left boundary to zero, $z_1 = w_1 = 0$. If we furthermore assume equidistant data, i.e., $h = h_1 = \cdots = h_n$ we obtain from (X.22) the simplified form:

$$\begin{pmatrix} 1 & & & & & 0 \\ 1 & 1 & & & & \\ & \ddots & \ddots & & & \\ & & 1 & 1 & & \\ & & & \ddots & \ddots & \\ 0 & & & & 1 & 1 \end{pmatrix} \begin{pmatrix} z_2 \\ z_3 \\ \vdots \\ z_{j+1} \\ \vdots \\ z_n \end{pmatrix} = \frac{2}{h} \begin{pmatrix} y_2 - y_1 \\ y_3 - y_2 \\ \vdots \\ y_{j+1} - y_j \\ \vdots \\ y_n - y_{n-1} \end{pmatrix} \tag{X.22}$$

Note the special structure of the matrix. It has only entries on two diagonals – the main diagonal and one subdiagonal. This type of matrix is called *banded*. Linear systems with banded matrices can be cheaply solved by special algorithms. Another, perhaps even more important observation is the way how we used the extra degree of freedom. We fixed it by a condition on the left boundary. Why at the left? We could as well have fixed it at the right boundary. The situation would have been even better if we would have a free wish for both boundaries. This would not favor (or bind) a particular boundary. This unsymmetrical behavior is typical for splines with an even degree, here degree two.

This approach, like the approach for linear splines and, later in this chapter, the approach for cubic splines, uses the same points for nodes and knots. Recall that the nodes are the interpolation points and the knots are the points where spline segments join. We refer to this approach as "nodes=knots".

Example 4. *Again, we consider the temperature data from Lund, cf. Tab. X.2 and interpolate it now with quadratic splines of the type "nodes=knots". The result is displayed in Fig. X.12.*

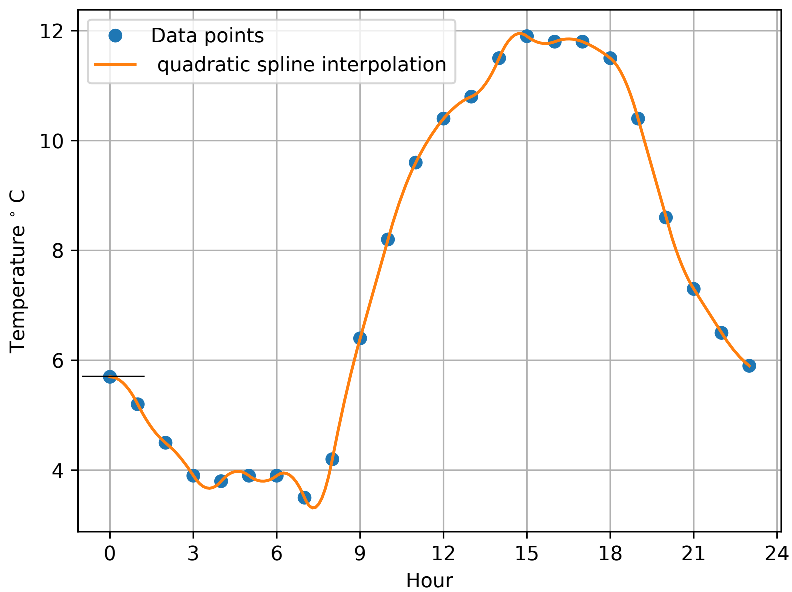


Figure X.12: "Nodes=knots"-type quadratic spline interpolation of temperature data from Lund, Sweden. The boundary condition $w_1 = 0$ at the left boundary is indicated.

There is a second approach to quadratic spline interpolation, which has one more degree of freedom. This extra degree of freedom is used as a second boundary condition in order to treat both boundaries in a more balanced way. For this approach we allow the nodes and knots to be different points and therefore refer to this approach by the acronym "knots≠nodes". The knots k_i are chosen to be the midpoints of the node intervals. To develop this case we assume data points $(x_1, y_1), \ldots, (x_{n+1}, y_{n+1})$ and define $n+1$ knots k_i by

$$k_1 = x_1, \quad k_i = \frac{x_{i-1} + x_i}{2}, \quad k_{n+1} = x_n,$$

As before we express a spline segment as

$$S_i(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i \quad \text{on } [k_i, k_{i+1}) \tag{X.23}$$

but note now the different roles of x_i and k_i.

The conditions for defining the unknown coefficients are now

- Interpolation: $S_i(x_i) = y_i$, for $i = 1, \ldots, n$ and interpolation at the end point $S_n(x_{n+1}) = y_{n+1}$.
- Continuity: $S_i(k_{i+1}) = S_{i+1}(k_{i+1})$, for $i = 1, \ldots, n-1$
- Continuity derivative: $S_i^{'}(k_{i+1}) = S_{i+1}^{'}(k_{i+1})$ for $i = 1, \ldots, n-1$

Thus, the first interpolation property gives $c_i = y_i$. The continuity property results in

$$S_i(k_{i+1}) = S_{i+1}(k_{i+1}) : h_i^2 a_i - h_i^2 a_{i+1} + 2h_i b_i + 2h_i b_{i+1} = 2(y_{i+1} - y_i) \qquad \text{(X.24)}$$

From continuity of the first derivative requirement we obtain

$$S_i^{'}(k_{i+1}) = S_{i+1}^{'}(k_{i+1}) : h_i a_i + h_i a_{i+1} + b_i - b_{i+1} = 0 \qquad \text{(X.25)}$$

To simplify the expressions we set $h_i = x_{i+1} - x_i$ and write a linear system of equations from Eqs. (X.24) and (X.25) to solve for the unknown coefficients of the quadratic spline segments. In addition, by finding the first derivative of quadratic spline segment we obtain $S_i(x) = 2a_i(x - x_i) + b_i$. Finally we use boundary conditions at each boundary by prescribing slopes $S_1^{'}(x_1) = w_1$ $S_n^{'}(x_n) = w_n$. This results in $b_1 = w_1$ and $b_n = w_n$. In total,

$$\begin{pmatrix} C_1 & & \\ H_1 & & \\ & H_i & \\ & & H_{n-1} \\ & & C_n \end{pmatrix} \begin{pmatrix} a_1 \\ b_1 \\ a_2 \\ b_2 \\ \vdots \\ a_i \\ b_i \\ a_{i+1} \\ b_{i+1} \\ \vdots \\ a_{n-1} \\ b_{n-1} \\ a_n \\ b_n \end{pmatrix} = 4 \begin{pmatrix} w_1 \\ y_2 - y_1 \\ 0 \\ y_3 - y_2 \\ 0 \\ \vdots \\ y_{i+1} - y_i \\ 0 \\ \vdots \\ y_n - y_{n-1} \\ 0 \\ w_n \end{pmatrix} \qquad \text{(X.26)}$$

with

$$H_j = \begin{pmatrix} h_j^2 & 2h_j & -h_j^2 & 2h_j \\ h_j & 1 & h_j & -1 \end{pmatrix}, \quad C_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}, \text{ and } C_n = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}$$

The matrix is banded matrix as the one in the previous case. Thus, the linear system is not expansive to compute.

Example 5. *Quadratic spline interpolation of the type "knots≠nodes" applied to the weather data from Tab. X.2 gives the curve displayed in Fig. X.13.*

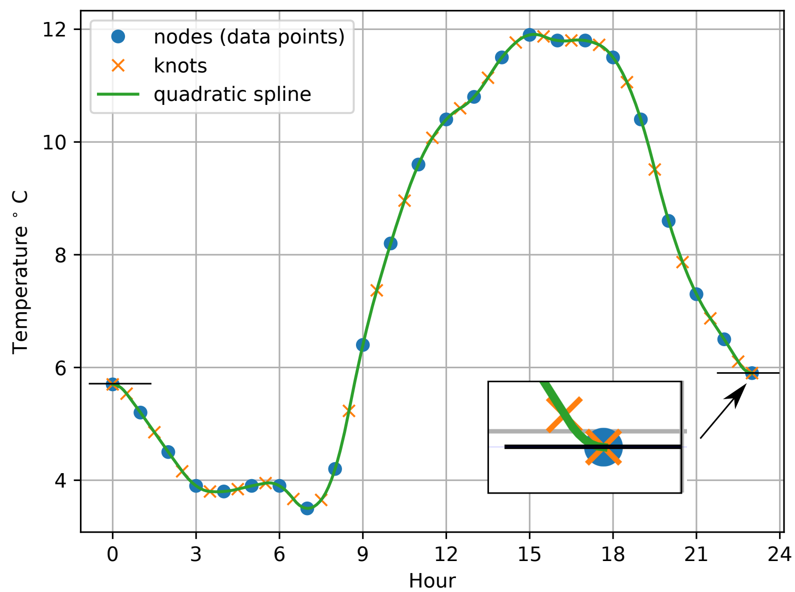


Figure X.13: Quadratic spline interpolation of temperature data from Lund, by using the approach "nodes ≠knots" and boundary values $w_1 = 0$ and $w_n = 0$. These are the slopes indicated at the left and right boundary of the figure.

X.3.3 Cubic Spline Interpolation

In previously presented types of splines interpolation, linear spline interpolation, and quadratic spline interpolation, we saw the advantages and disadvantages of those splines for interpolating given data. Mainly, these lower order splines do not have continuous second derivatives. This motivates to study cubic splines as a tool of smooth interpolation of a big amount of data. To contrast it to polynomial interpolation of many data points, we consider again Runge's function and the related *Runge's phenomenon* which was used to demonstrate the problems with high degree polynomial interpolation. We will see how cubic splines interpolation behave much better on that problem than polynomial interpolation or lower order spline interpolation.

A cubic spline segment is given by the formula:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad \text{on} \ [x_i, x_{i+1}) \tag{X.27}$$

We have four unknowns per segment, namely a_i, b_i, c_i, d_i. If we have $n + 1$ data points, $(x_1, y_1), \ldots, (x_{n+1}, y_{n+1})$, there are n segments. Therefore, we have to find conditions to determined in total $4n$ unknowns coefficients.

X.3.3.1 What are the conditions to determine an interpolation cubic spline?

The interpolation and continuity conditions determine an interpolatory cubic spline

- Interpolation: $S_i(x_i) = y_i$, for $i = 1, \ldots, n$ and interpolation at the end point $S_n(x_{n+1}) = y_{n+1}$.

- Continuity: $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ for $i = 1, \ldots, n-1$.

- Continuous first derivative: $S_i^{'}(x_{i+1}) = S_{i+1}^{'}(x_{i+1})$ for $i = 1, \ldots, n-1$.

- Continuous second derivative: $S_i^{''}(x_{i+1}) = S_{i+1}^{''}(x_{i+1})$ for $i = 1, \ldots, n-1$.

As each of the n cubic spline segments has four unknown coefficients, our description of the function S involves $4n$ unknowns coefficients. The interpolation properties impose $n+1$ linear constraints and the continuity properties impose $3(n-1)$ linear constraints on its coefficients. Therefore, there are a total of $n+1+3(n-1) = 4n-2$ linear constraints for the $4n$ unknown coefficients. As the problem is not fully determined by these conditions there are infinity many cubic splines passing through a given set of data points $(x_1, y_1), ..., (x_n, y_n)$. In order to uniquely define a cubic spline we have to arrive at a linear system, that has the same number of equations as unknowns. Therefor we need to add two more linear constraints (two free wishes). There are various ways of specifying these two additional constraints leading to different splines, e.g., *natural cubic splines*, *curvature-adjusted cubic splines*, and *clamped cubic splines*. These differ by the way how end conditions are selected. This selection is often motivated by the actual case of application. The simplest way of adding two more linear constraints is to put them as conditions at x_1 and x_n. That's why we call them *end conditions*.

▶ **Natural cubic splines** The most commonly used end conditions are:

$$S_1^{''}(x_1) = 0 \quad , \quad S_n^{''}(x_n) = 0.$$

A spline satisfying these conditions is called a "natural" spline because - as we will see later it plays an important role in the elasticity of beams.

Example 6 (Natural cubic splines)**.** *Consider again the the problem to interpolate Runge's function, see Sec. X.2.3:*

$$f(x) = \frac{1}{1 + 25x^2} \tag{X.28}$$

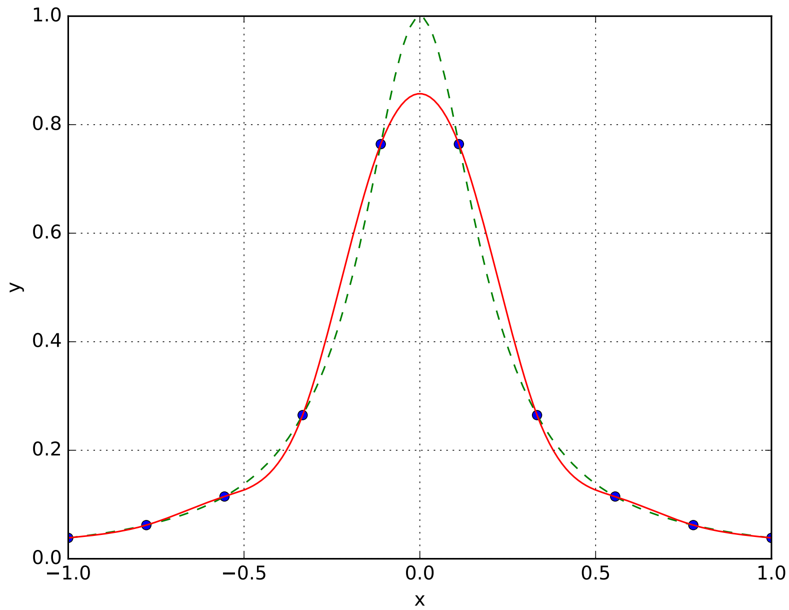


Figure X.14: Natural cubic splines interpolation of Runge's function.

Figure X.14 shows how nicely a natural cubic spline describes Runge's function without oscillation at the boundary regions of the interval. By comparing with the polynomial counterpart in Sec. X.2.3 we see why cubic spline interpolation is to be prefered.

▶ **Curvature-adjusted cubic spline** The alternative approach to a natural cubic spline requires setting:

$$S_1^{''}(x_1) = a \quad , \quad S_n^{''}(x_n) = b$$

As we observe at the second derivative we have arbitrary choice a, and b these instead of zero in natural cubic spline interpolation.

▶ **Clamped cubic splines** A clamped cubic splines is defined by its first derivative at the first and last nodes:

$$S_1^{'}(x_1) = a \quad , \quad S_n^{'}(x_n) = b$$

It is called a "clamped" spline because "it is what occurs when you clamp the end of a drafting spline so that is has a desired slope" Chapra [2011]. For example, when we drive on a bridge the ends of bridge have been such, that their slopes fit to the slope of the connecting streets, cf., Figure X.15. Ideally, a clamped cubic spline interpolating certain given supporting points is the way to describe such a bridge. Thus, the slope at the beginning and end of the spline are under users control.

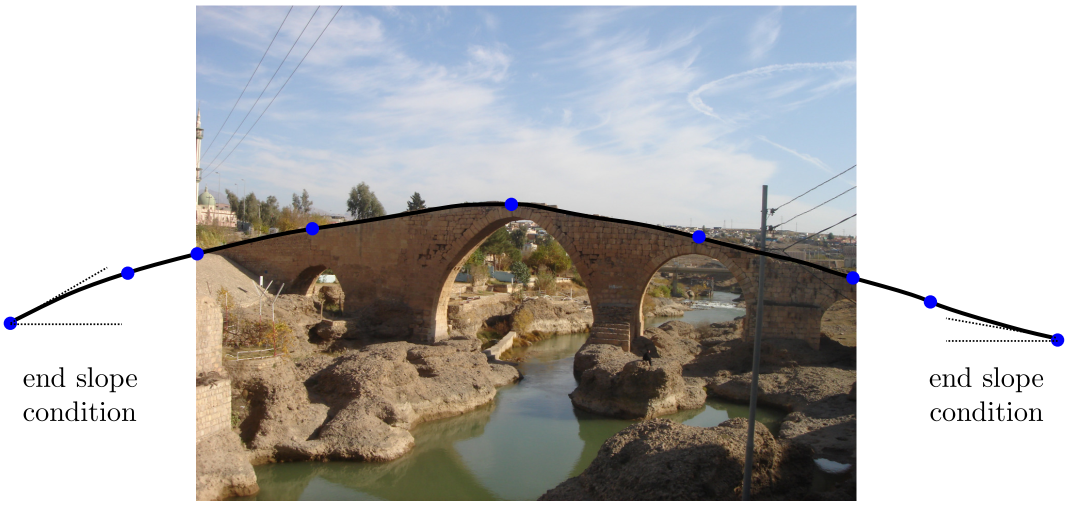


Figure X.15: Dalal bridge in Zakho, Iraq, which dates to the Roman era. The picture behind the clamped curve is taken from Ancient Art [2017].

Additionally, there are several more cases of end conditions cubic splines for example *parabolically terminated* cubic splines, and *not-a-knot* cubic spline. Here the end conditions are not set by physical or technical consideration. Often, when the interpolation result is not statisfactory a change of the type of end conditions is made. For example in Runge's example we observe that natural cubic spline interpolation results in a good description of Runge's function. If we use different types of end conditions maybe we do not get qualitatively the same result as if natural cubic spline were chosen. The following Figure X.16 shows Runge's function interpolated with a clamped spline with the choice $S_1^{'} = 1$ and $S_{n-1}^{'} = -1$.

X.3.3.2 Construction of an interpolatory cubic spline.

Here we will show how to compute the coefficients of a spline segment:

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \text{ on } [x_i, x_{i+1}) \tag{X.29}$$

The continuity and interpolation conditions are

$$\left.\begin{array}{llr} & \text{condition} & \text{number} \\ (1) & S_i(x_i) = y_i, \text{and} \quad S_n(x_{n+1}) = y_{n+1} \quad i = 1, \dots, n & n+1 \\ (2) & S_i(x_{i+1}) = S_{i+1}(x_{i+1}), \quad i = 1, \dots, n-1 & n-1 \\ (3) & S_i^{'}(x_{i+1}) = S_{i+1}^{'}(x_{i+1}), \quad i = 1, \dots, n-1 & n-1 \\ (4) & S_i^{''}(x_{i+1}) = S_{i+1}^{''}(x_{i+1}), \quad i = 1, \dots, n-1 & n-1 \\ (5) & S_1^{''}(x_1) = 0, & 1 \\ (6) & S_n^{''}(x_{n+1}) = 0, & 1 \end{array}\right\} \quad \text{total=4n} \tag{X.30}$$

We have two boundary end conditions (5), and (6) in Eq. (X.30). In total we have now a fully determined system with $4n$ conditions for the $4n$ unknowns.From them we can compute the coefficients of the cubic spline segment (X.29):

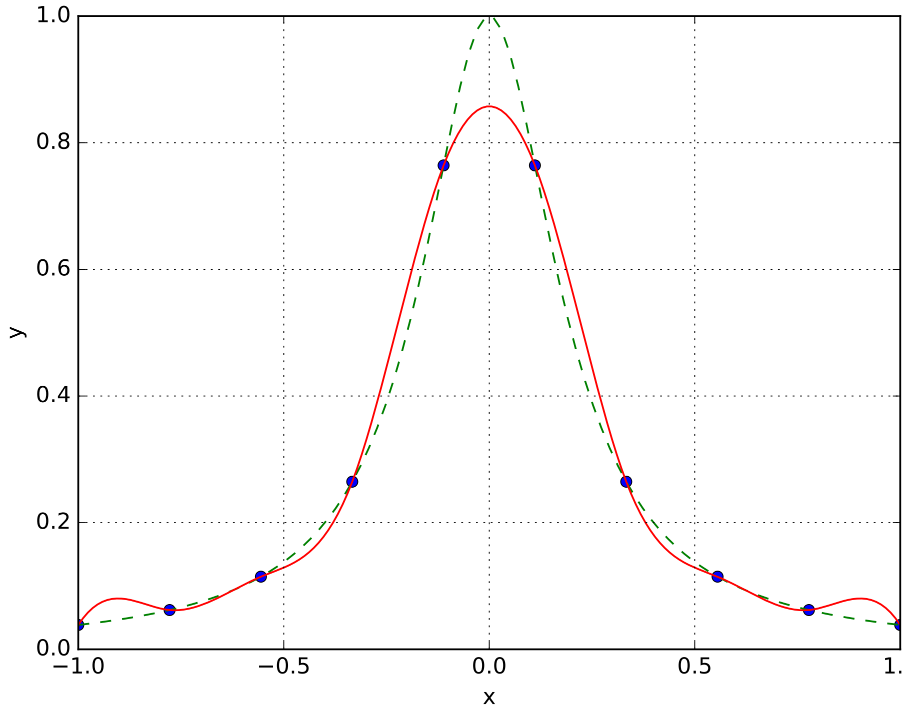


Figure X.16: Runge's function with 10 points and clamped end conditions stating that first derivatives at the left and right boundaries being +1 and -1, respectively.

1. The first property, $S_i(x_i) = y_i$, which is an interpolation condition, gives:

$$d_i = y_i \tag{X.31}$$

2. The second property, $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$, states that curve a cubic spline is continuous at at the nodes. From this and by setting $h_i = x_{i+1} - x_i$ we obtain:

$$a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i = d_{i+1} \tag{X.32}$$

   Additionally, the first derivatives at the interior nodes of cubic spline segment must by equal.

$$S_i'(x_{i+1}) = 3a_i h_i^2 + 2b_i h_i + c_i = c_{i+1} = S_{i+1}'(x_{i+1}) \tag{X.33}$$

   Correspondingly, for the second derivatives we get

$$S_i''(x_{i+1}) = 6a_i h_i + 2b_i = 2b_{i+1} = S_{i+1}''(x_{i+1}). \tag{X.34}$$

3. Now, we will introduce new variables for second derivatives at x_i:

$$\psi_i := S_i''(x_i) = 2b_i \tag{X.35}$$

and express b_i by

$$b_i = \frac{\psi_i}{2} \tag{X.36}$$

4. From Eq. (X.34) we conclude

$$a_i = \frac{b_{i+1} - b_i}{3h_i} \tag{X.37}$$

By substituting (X.36) into (X.37) we get:

$$a_i = \frac{\psi_{i+1} - \psi_i}{6h_i} \tag{X.38}$$

and by substituting Eq.(X.36) and Eq.(X.38) in Eq.(X.32) we get:

$$c_i = \frac{y_{i+1} - y_i}{h_i} - h_i \frac{(2\psi_i + \psi_{i+1})}{6} \tag{X.39}$$

As we observe all unknowns coefficients can be expressed in terms of the ψ.

5. From Eq. (X.33) we obtain now by inserting the expressions for a_i, b_i and c_i:
$3\left(\frac{\psi_i - \psi_{i-1}}{6h_{i-1}}\right) h_{i-1}^2 + 2\frac{\psi_{i-1}}{2} h_{i-1} + \left(\frac{y_i - y_{i-1}}{h_{i-1}}\right) - h_{i-1}\frac{(2\psi_{i-1}+\psi_i)}{6} = \left(\frac{y_{i+1}-y_i}{h_i} - h_i\frac{(2\psi_i+\psi_{i+1})}{6}\right)$
and finally,

$$h_{i-1}\psi_{i-1} + 2(h_{i-1} + h_i)\psi_i + h_i\psi_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right) \tag{X.40}$$

6. By then using conditions (5) and (6) Eq. (X.30) obtain have a linear system to compute the vector ψ from which we determine all coefficients of the cubic spline segment formula.

   Consequently, we can write the linear system with the two boundary conditions as free "wishes" which can be expressed as $\psi_1 = w_1$, and $\psi_{n+1} = w_2$.

   In the case of natural splines these "wishes" are $w_1 = 0$ and $w_{n+1} = 0$. For natural cubic splines, $S_n''(x_{n+1}) = 0$ implies that $\psi_{n+1} = 0$.

   The following system what we need to solve is:

$$\begin{cases} h_{i-1}\psi_{i-1} + 2(h_{i-1} + h_i)\psi_i + h_i\psi_{i+1} = 6\left(\frac{y_{i+1}-y_i}{h_i} - \frac{y_i-y_{i-1}}{h_{i-1}}\right) \\ \psi_1 = \psi_{n+1} = 0 \end{cases}$$

   or more compactly,

$$H\psi = y.$$

$$\begin{pmatrix} 1 & & & & & & & 0 \\ h_1 & 2(h_1+h_2) & h_2 & & & & & \\ & h_2 & 2(h_2+h_3) & h_3 & & & & \\ & & h_3 & 2(h_3+h_4) & h_4 & & & \\ & & & & \ddots & & & \\ & & & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & & & & & & & 1 \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \\ \vdots \\ \vdots \\ \psi_n \\ \psi_{n+1} \end{pmatrix} =$$

$$= 6 \begin{pmatrix} w_1 \\ \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \frac{y_4-y_3}{h_3} - \frac{y_3-y_2}{h_2} \\ \vdots \\ \vdots \\ \frac{y_{n+1}-y_n}{h_n} - \frac{y_n-y_{n-1}}{h_{n-1}} \\ w_{n+1} \end{pmatrix} \tag{X.41}$$

The system matrix is strictly diagonally dominant as its diagonal elements are larger than the sum of the off-diagonal elements. This property guarantees that a matrix is invertible. Consequently the system has a unique solution ψ.

Furthermore, the matrix is banded – more precisely it is tridiagonal. Solving linear systems with tridiagonal matrices is fast.

Finaly we note, that H can easily be constructed in Python from an array containing the step sizes h_i:

```
h=diff(array(x))

H=diag([1]+list(2*(h[:-1]+h[1:]))+[1])+diag(list(h[:-1])+[0],-1)+\
          diag([0]+list(h[1:]),+1)
```

As pointed out before the definition of cubic spline leaves two degree of freedom "wish". In our case we choose natural cubic spline end conditions, but we have different kind of end conditions can by satisfies our linear system as well, for example:

- Clamped end conditions:

  These end conditions prescribe the slopes at the boundary points. So, $S'(x_0)$ and $S'(x_n)$ are given. Thus, the conditions of ψ_0 and ψ_n can be derived and a linear system corresponding to Eq. (X.41) can be set up.

- Not-a-knot end conditions:

  This end conditions enforces continuity of the third derivative at the second and the next to last knots:

$$S_1'''(x_2) = S_2'''(x_2), \quad S_{n-1}'''(x_n) = S_n'''(x_n) \tag{X.42}$$

Common third derivatives make the first and second spline segment a common third degree polynomial, which is as if the second knot would be replaced and just a node for the interpolation would be left over at that place. That's why this end-condition is called a "not-a-knot" condition.

► We sum up all steps to find the coefficients of a cubic interpolatory spline:

1. Condition, *interpolation*: $S_i(x_i) = y_i$ we get: $\boxed{d_i = y_i}$
2. Condition, *continuity of the second derivative* $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$ we get: $\boxed{b_i = \frac{\psi_i}{2}}$, and $\boxed{a_i = \frac{\psi_{i+1} - \psi_i}{6h_i}}$
3. Condition, *continuity of the function* $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ we get:
$$\boxed{c_i = \frac{y_{i+1} - y_i}{h_i} - h_i \frac{(2\psi_i + \psi_{i+1})}{6}}$$
4. Condition, *continuity of the first derivative* $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$ we get:
$$\boxed{h_{i-1}\psi_{i-1} + 2(h_{i-1} + h_i)\psi_i + h_i\psi_{i+1} = 6\left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right)}.$$
Additionally two boundary conditions are used to fix the remaining two degrees of freedom.

Example 7. *We use the same data points as in Example 1 which describes voltage drops V_i across a resistor for a number of different current values x_i. From this data we now construct a cubic spline and compute the voltage drop at $x = 1.15\ A$.*

In the first step we use Eq. (X.41) *to determine the coefficients ψ. So, we have to solve linear system of equations :*

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.5 & 2. & 0.5 & 0 & 0 \\ 0 & 0.5 & 1.5 & 0.25 & 0 \\ 0 & 0 & 0.25 & 1.5 & 0.5 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \psi_0 \\ \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 17.4 \\ 12.72 \\ 21.12 \\ 0 \end{pmatrix}$$

This equation can be solved by using Python or Matlab. The result is

$$\psi = \begin{pmatrix} 0. & 7.790625 & 3.6375 & 13.47375 & 0 \end{pmatrix}$$

Now we can find the coefficients of each cubic spline segment S_1, S_2, S_3 to be

i	a_i	b_i	c_i	d_i
1	*2.596875*	*0*	*-0.94921875*	*-0.45*
2	*-1.384375*	*3.8953125*	*0.9984375*	*-0.6*
3	*6.5575*	*1.81875*	*3.85546875*	*0.70*
4	*-4.49125*	*6.736875*	*5.994375*	*1.88*

Then we will have:

$$
\begin{aligned}
S_1(x) &= 2.596875(x-0.25)^3 - 0.94921875(x-0.25) - 0.45 \\
S_2(x) &= -1.384375(x-0.75)^3 + 3.8953125(x-0.75)^2 + 0.9984375(x-0.75) - 0.6 \\
S_3(x) &= 6.5575(x-1.25)^3 + 1.81875(x-1.25)^2 + 3.85546875(x-1.25) + 0.70 \\
S_4(x) &= -4.49125.(x-1.25)^3 + 6.736875(x-1.25)^2 + 5.994375(x-1.25) + 1.88
\end{aligned}
$$

To evaluate the spline, at for example $x = 1.15$, one first has to find the knot interval to which x belongs, here the second interval, and then one evaluates the corresponding spline segment:

$$
\begin{aligned}
S_2(1.15) &= -1.384375(1.15-0.75)^3 + 3.8953125(1.15-0.75)^2 + 0.9984375(1.15-0.75) - 0.6 \\
&= 0.33402499
\end{aligned}
$$

Example 8. *By using the same data temperature data reported from Lund city (Sweden) on April, 4, 2017 X.2. The cubic spline interpolation this data display in Figure*

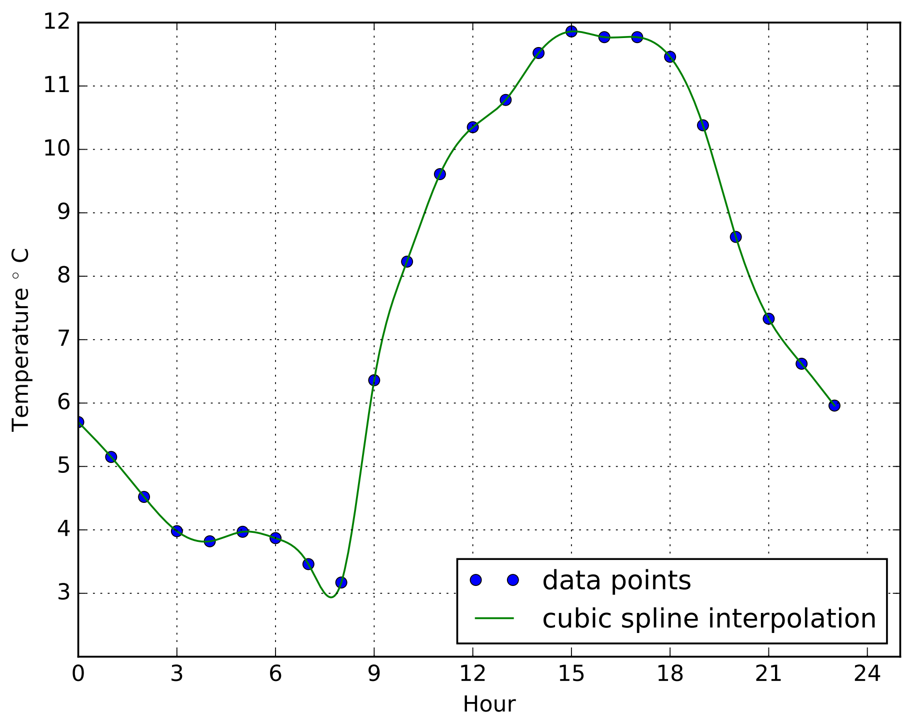


Figure X.17: Cubic spline interpolation of temperature Lund city in Sweden.

The plot of the cubic spline interpolating the temperature Lund city shows that a cubic spline results in a smoother curve than the one obtained by quadratic spline interpolation and linear spline interpolation. Also, cubic spline interpolation is the better choice for interpolation then interpolation by higher degree polynomials.

X.3.3.3 Why cubic spline interpolation?

The term smoothness normally refers to continuity of the function and its derivatives. In the context of splines we also often speak about smoothness, when we want to emphasize that the curve has no "unnecessary oscillations." In this section we demonstrate why cubic splines have optimal smoothness also in this sense.

We consider as a measure of smoothness the integral $\int_a^b [f^{''}(x)]^2 dx$ which measures accumulated curvature.

The following theorem states a minimality property of cubic splines with respect to this measure:

Theorem (Optimality of Natural Splines)**.**

Let $f^{''}$ be continuous in $[a, b]$ and let $a = x_0 < x_1 < \ldots < x_n = b$. If S is the natural cubic spline interpolating f at the knots x_i for $0 \le i \le n$ then:

$$\int_a^b [S^{''}(x)]^2 dx \le \int_a^b [f^{''}(x)]^2 dx$$

Proof:

Let consider the difference between S and f

$$g(x) = f(x) - S(x) \tag{X.43}$$

Consequently,

$$(f^{''})^2 = (S^{''})^2 + (g^{''})^2 + 2S^{''}g^{''} \tag{X.44}$$

and

$$\int_a^b (f^{''})^2 dx = \int_a^b (S^{''}) dx + \int_a^b (g^{''})^2 + \int_a^b 2S^{''}g^{''} dx \tag{X.45}$$

If we can show

$$\int_a^b S^{''}g^{''} dx = 0 \tag{X.46}$$

than we conclude

$$\int_a^b (f^{''})^2 dx \ge \int_a^b (S^{''})^2 dx \tag{X.47}$$

which would complete the proof.

Let us proof no why (X.46) holds. Firstly, we integrate by parts,

$$\int_a bS^{''}g^{''} dx = S^{''}g^{'}\Big|_a^b - \int_a bS^{'''}g^{'} dx$$

Since $S^{''}(a) = S^{''}(b) = 0$ by natural cubic spline conditions, the first term zero. Additionally, in the second term, since $S^{'''}$ segment is constant we set

$$c_i = S'''(x), \quad \text{for} x \in [t_i, t_{i+1}].$$

Then

$$\int_a^b S^{''} g^{'} dx = \sum_{i=0}^{n-1} c_i \int_{x_i}^{x_{i+1}} g^{'}(x) dx$$

and

$$\sum_{i=0}^{n-1} c_i [g(x_{i+1} - g(x_i))] = 0.$$

X.3.3.4 Discussion

We have seen that linear spline interpolation is easy to calculate and it is continuous, but is not smooth. We have looked at two types of quadratic spline interpolation. In the first case, "nodes=knots," we obtain one boundary condition, and in the second case, "nodes≠knots," we get two boundary conditions. This second case seemed to yield a better result. The fact that we have two boundary conditions means that we have more freedom too choose, and thus more control of both sides of the spline curve. Cubic spline interpolation shows smoother results than the other kinds of spline interpolation because in cubic spline interpolation we get two boundary conditions and because of the higher degree we get a smoother curve.

We notice that if we consider splines with "nodes=knots," those of even degree will have an odd number of boundary conditions, and those of odd degree will have an even number of boundary conditions, which translates to having equal control over each end of the curve. The proof of this statement is as follows:

Consider a polynomial of degree k, and $n+1$ interpolation points. Then there will be n spline segments. To find the coefficients of the spline we must have an equal number of conditions and unknowns. We have $(k+1)$ coefficients for each polynomial segment, so we will have $n(k+1)$ coefficients for the entire spline. From the properties of cubic splines we get $n+1$ interpolation conditions and $k(n-1)$ continuity conditions. Thus, the total number of conditions given by the properties of the spline is $(nk+1)+k(n-1)$. Consequently, we need additional $n(k+1)-k(n-1)-(n+1) = k-1$ end conditions. Thus, when the degree of the polynomial k is even we will have an odd number of boundary conditions and viceversa.

X.3.4 A historical fact of cubic splines interpolation.

The first picture, Figure X.18, was shown to students of the course Seminar in Numerical Analysis in 2015 at Lund University during a lecture on spline interpolation. The picture shows the construction of a spline by using "duck" lead weights. It was our personal experience that this picture increased our enthusiasm discover more about the history of cubic splines by researching how the "ducks" were used to construct a cubic spline.

Figure X.19 shows a course in airplane design in 1941. A cubic spline is constructed in order to shape a turbine blade. The "ducks" are used to arrive to the smoothest possible shape. The lead weights are "called ducks because of their duck-like shape" [spline, 2017]. It is commonly accepted that the first mathematical reference to *splines* is 1946. The word "spline" was used in connection with smooth, piecewise polynomial approximation.

Figure X.18: The drafting technique of a spline by using duck lead weights. In 1997 at Rolls Royce in Karlshamn, Sweden.

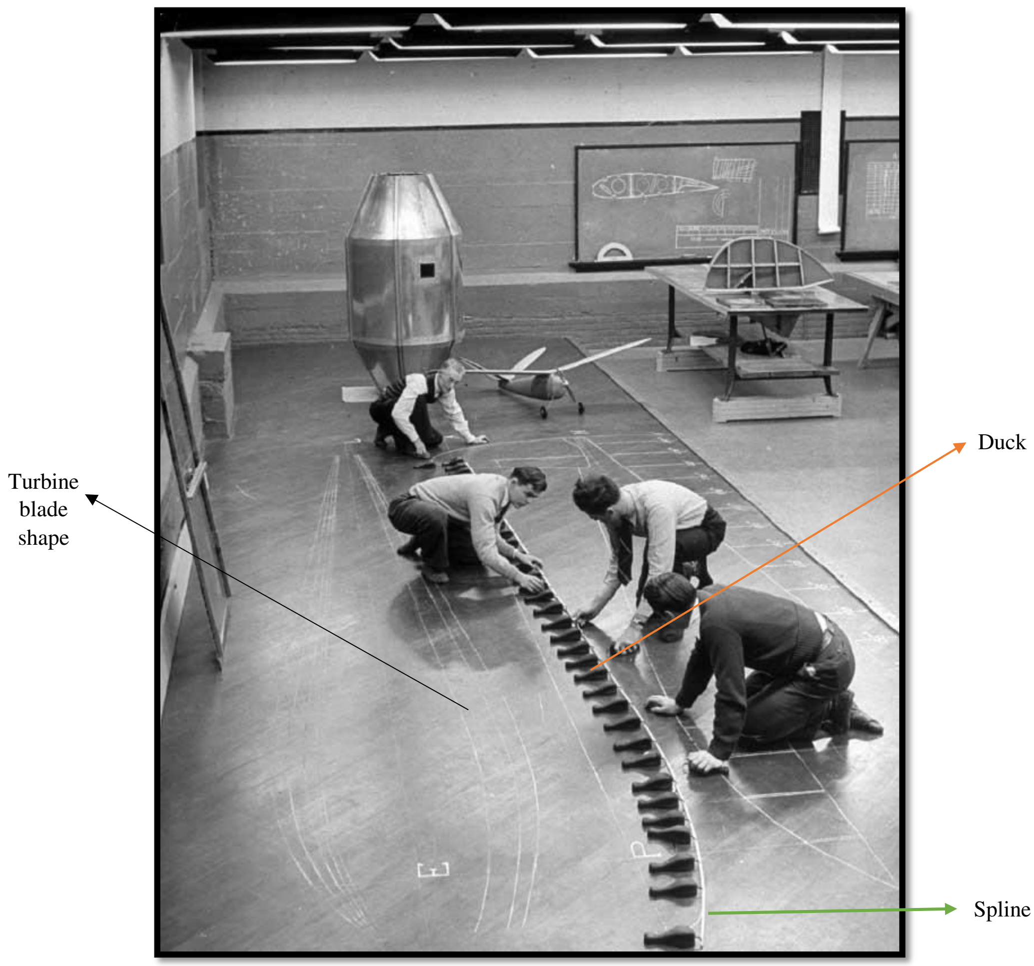


Figure X.19: Course in Airplane Lofting, Burgard High School, Buffalo, NY, USA, January 1, 1941 (Digital Remediation [2017])

Chapter 5

Conclusion and Future Work

The aim of this thesis is to describe and study different methods of teaching in several numerical analysis books to arrive to a good strategy for the teaching of cubic spline interpolation. As future work we would like to further the development of the didactic methods for the teaching of other topics in a numerical analysis course. When we add exercises in any topic we must take into account the critical points as we discussed in the introduction of Chapter 4. The list of exercises should be such that after solving them a student can feel certain that he has understood well the topic. This will give students a motivation to develop their scientific skills in the subject. It is also important to develop the practical aspect of teaching. Even when using the same textbook, different teachers will approach the topic differently when they are in front of a class. What a teacher does and says in front of the audience will clarify the contents of the textbook. A teacher in front of the blackboard will affect students more than the actual textbook contents. Consequently, by using the comparison criteria in 2.1 we can develop various topics in numerical analysis in order to find a good teaching strategy.

Bibliography

A project by the government of India (2017). https://onlinecourses.nptel.ac.in/noc17_ge02/preview. [Online; accessed 11 March 2017].

Ancient Art (2017). http://ancientart.tumblr.com/page/66. [Online; accessed 29 April 2017].

Chapra, S. C. (2011). *Applied Numerical Methods with MATLAB for Engineers and Scientists.* McGraw-Hill Education / Asia, 3nd edition.

Digital Remediation (2017). http://cornelljournalofarchitecture.cornell.edu/read.html?id=74. [Online; accessed 05 March 2017].

Duke University (2015). https://services.math.duke.edu/∼djl/teaching/math361spring2015/syllabus.pdf. [Online; accessed 11 March 2017].

Fausett, L. V. (2008). *Applied Numerical Analysis Using MATLAB.* Pearson Prentice Hall, NJ, 3nd edition.

Hacettepe Üniversitesi (2017). http://yunus.hacettepe.edu.tr/∼edacelik/231-web/info.htm. [Online; accessed 11 March 2017].

Harvard University (2012). http://isites.harvard.edu/icb/icb.do?keyword=k89160&pageid=icb.page522356. [Online; accessed 11 March 2017].

Introduction to Computer Graphics (2017). http://www.kharwal.com/introduction-to-computer-graphics/. [Online; accessed 23 April 2017].

Kincaid, D. and Cheney, W. (2002). *Numerical Analysis.* American Mathematical Society, Providence, RI, 3nd edition.

Maghdid, D. (2016). *Comparative study of teaching computational mathematics in two different learning environments.* PhD thesis.

McMaster University (2014). http://www.ece.mcmaster.ca/∼xwu/outline3SK3.htm. [Online; accessed 11 March 2017].

Sauer, T. (2012). *Numerical Analysis.* Pearson Education, Inc, 2nd edition.

spline, F. (2017). http://en.wikipedia.org/wiki/Flat_spline. [Online; accessed 05 March 2017].

University, L. (2017). http://www.ctr.maths.lu.se/na/courses/FMN050/. [Online; accessed 12 March 2017].

University of Florida (2009). http://www2.mae.ufl.edu/haftka/numerical/. [Online; accessed 11 March 2017].

Washington State University (2017). http://www.math.wsu.edu/faculty/genz/448/info.html. [Online; accessed 11 March 2017].

Appendix A

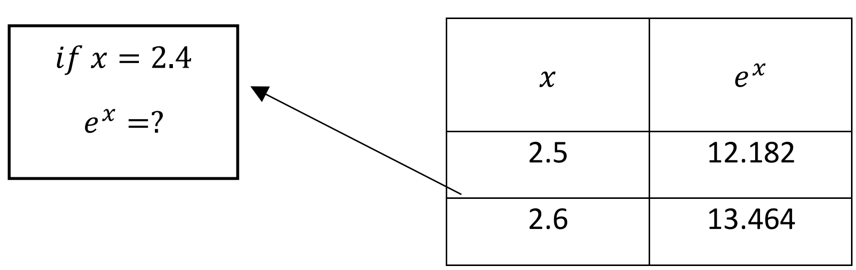
Example (Table lookup):

What is table lookup? And how can observe type of work called table lookup? What follows is a brief summary of table lookup using data from exponential functions:

x	e^x	e^{-x}	x	e^x	e^{-x}
0.00	1.000	1.000	2.5	12.182	0.0821
0.05	1.0513	0.9512	2.6	13.464	0.0743
0.10	1.1052	0.9048	2.7	14.880	0.0672
0.15	1.1618	0.8607	2.8	16.445	0.0608
0.20	1.2214	0.8187	2.9	18.174	0.0550

Figure A.1: Table exponential functions

In order to calculate the function e^x when $x = 2.4$ is not in the previous table.



By using linear spline formula. The following are steps of solving:

$$S(x) = ax + b \tag{A.1}$$

$$S(2.5) = a(2.5) + b \Rightarrow 12.182 = a(2.5) + b \tag{A.2}$$

$$S(2.6) = a(2.6) + b \Rightarrow 13.464 = a(2.6) + b \tag{A.3}$$

By Subtracting the equations (2) from (3) will get:
$a = 12.82$ and $b = -19.868$.

To find the function e^x when $x = 2.4$:

$$S(2.4) = (12.82)(2.4) - 19.868 = 10.9 \tag{A.4}$$

This work is called *table lookup* (what we are doing by using table whether the work by using computer or hands called *table look up*)