Spline Interpolation

Alnura Abdyrova, Bence Pór, Mariia Avdokhina

Fall term 2024

Contents

1 Introduction **3**

2 Mathematical Foundation of Splines **3**
2.1 Linear 3
2.2 Quadratic 4
2.3 Cubic 4
2.4 Derivatives at the Endpoints 5
2.5 Selecting Knots 6

3 Implementation **6**
3.1 General 6
3.2 Feature Engineering 6

4 Evaluation of the Method **7**
4.1 Advantages 7
4.2 Disadvantages 8

5 Conclusion **8**

6 References **8**

1 Introduction

In the mathematical field of numerical analysis, interpolation is an estimation method where the goal is to find and construct new data points based on a discrete set of known data points of a function f.

Spline interpolation is a specific interpolation technique where, instead of fitting a single high-degree polynomial to all the values at once, an alternative process is implemented: to approximate the function f, the interval [a, b] is divided into smaller subintervals; afterwards, a piecewise approximation by low-degree polynomials is found. Such polynomials are called "splines", giving the method its name; the endpoints of the subintervals are referred to as "knots".

The focus of this paper is to discuss spline interpolation in detail, as well as analyze its types and provide real-life application examples.

Spline interpolation is usually preferred over polynomial interpolation: the latter is often not sufficiently flexible to yield useful, direct procedures. Instead, polynomial interpolation provides a starting point for other, more precise techniques. Spline interpolation, on the other hand, balances computational efficiency, numerical stability and precision, while also introducing smoothness and avoiding oscillatory behavior near the edges.

Said oscillatory behavior near the edges of an interval is called Runge's phenomenon. In 1901, a German mathematician Carl Runge demonstrated that when one uses polynomial interpolation with high-degree polynomials over a set of equidistant interpolation points, a problem of oscillation near the edges (endpoints) of the interval occurs; as the order of the polynomial approaches infinity, the polynomial diverges even more. Ultimately, Runge's discovery illustrates that going to higher degrees does not always guarantee higher accuracy.

Through fitting low-degree polynomials over smaller subintervals and piecing these polynomials together, spline interpolation ensures smoothness and avoids oscillations. Furthermore, the method provides numerical stability due to lower-degree polynomials; it is also noteworthy that a change in one data point only affects the splines in adjacent subintervals, not in the whole interval. Another conclusion derived from the splines' lower degrees is higher computational efficiency: regardless of the number of data points, the degree of the splines remains low, making managing large datasets easier.

Returning to smoothness as an aspect of spline interpolation, it can be stated that since the method ensures continuity of function and its first and second derivatives at the interval boundaries, the resulting curve provides a better fit for real-life applications such as computer graphics, robotics and machine learning. Practical applications of spline interpolation will be expanded on in Section 3.

2 Mathematical Foundation of Splines

2.1 Linear

This section is built on material from paper [1].

Linear spline interpolation is considered the simplest form of spline interpolation because it uses first-degree polynomials. This method is used to approximate a function by connecting a series of known data points (x_i, y_i) with straight-line segments. If the relationship of two data points is considered $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$ in the subinterval $[x_i, x_{i+1}]$, the line joining these two points is given by:

$$\frac{s_i(x) - f(x_i)}{x - x_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

$$\implies s_i(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i)$$

This can also be written as

$$s_i(x) = a_i + b_i(x - x_i)$$

where

$$a_i = f(x_i) \quad \text{and} \quad b_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}.$$

It creates a piecewise linear function that is continuous over the range of the data but not necessarily differentiable at the data points. This implies that different linear equations for each interval between consecutive data points define the interpolated function. The key properties of linear spline are:

(i) $s(x_i) = f(x_i)$ for $i = 1, 2, \ldots, n$.

(ii) $s(x)$ is a polynomial of degree 1, denoted by $s_i(x)$, on each subinterval $[x_i, x_{i+1}]$ where $i = 1, 2, \ldots, n - 1$.

(iii) $s(x)$ is a continuous function on (a, b).

However, because it uses straight line segments, the function is not smooth, and therefore is not differentiable at the points where the line segments meet knots. While it does not provide the smoothness of higher-order splines, it is computationally efficient and easy to implement. This makes it a practical choice for situations where the data is roughly linear or when simplicity is more important than precision or smoothness.

2.2 Quadratic

This section Quadratic spline interpolation creates a smooth curve to fit the data. While ensuring continuity not simply of the function itself, but also of its first derivative at the knots (a key difference between linear and higher-order splines), the technique falls short in comparison to cubic spline interpolation and thus is used less widely. Although quadratic spline interpolation offers a balance between linear and cubic methods, it lacks simplicity of the former and smoothness of the latter, and therefore will not be discussed as thoroughly in this paper.

2.3 Cubic

This section uses information from books [2],[3],[5].

Cubic spline is the most widely used spline interpolation method. The idea is very similar to that of quadratic spline, but more advanced. Cubic spline creates a smooth curve that is continuous in both the first and second-degree derivatives. Similar to the previously mentioned lower-degree splines, construction of the cubic spline requires $n + 1$ data points that the curve crosses. Each pair of points is connected by a cubic equation:

$$S_0 = a_1x^3 + b_1x^2 + c_1x + d_1 = f(x), \text{ for } x_0 \leq x \leq x_1$$

$$\vdots$$

$$S_n = a_nx^3 + b_nx^2 + c_nx + d_n = f(x), \text{ for } x_{n-1} \leq x \leq x_n$$

n such equations are needed for the construction of the whole curve; each equation contains 4 coefficients. Overall, the total number of coefficients one has to find equals $4n$. To find these coefficients, one has to use information given by the definition of the cubic spline to construct a system of $4n$ equations.

For the spline to be continuous, the curves between the knots have to be equal at the endpoints. For the arbitrary i^{th} knot x_i (the right endpoint of the i^{th} polynomial and the left endpoint of the $i+1^{\text{th}}$ polynomial), we have $S_i(x_i) = f(x_i) = S_{i+1}(x_i)$:

$$a_i x_i^3 + b_i x_i^2 + c_i x_i + d_i = a_{i+1} x_i^3 + b_{i+1} x_i^2 + c_{i+1} x_i + d_{i+1} \quad = f(x_i)$$

This yields $2n$ equations (there are n curves and each gives 2 equations, or from another approach each interior point gives 2 equations as we can see above and the endpoints give 1).

For the first derivative of the spline to be continuous, the first derivatives of two consecutive polynomials have to match as well. The first derivative of the i^{th} polynomial is

$$3a_i x_i^2 + 2b_i x_i + c_i$$

The first derivative has to equal the first derivative of the $i+1^{\text{th}}$ polynomial at x_i. $S_i'(x_i) - S_{i+1}'(x_i) = 0$:

$$3a_i x_i^2 + 2b_i x_i + c_i = 3a_{i+1} x_{i+1}^2 + 2b_{i+1} x_{i+1} + c_{i+1}$$
$$3a_i x_i^2 + 2b_i x_i + c_i - 3a_{i+1} x_{i+1}^2 - 2b_{i+1} x_{i+1} - c_{i+1} = 0$$

There are $n+1$ knots, but two of these are the endpoints: $n-1$ equations are obtained.

Similarly, for the second derivative of the spline to be continuous, the second derivatives of two consecutive polynomials have to match. The second derivative of the i^{th} polynomial is:

$$2a_i x_i + b_i$$

The second derivative has to be equal to the second derivative of the $i+1^{\text{th}}$ polynomial at x_i. $S_i''(x_i) - S_{i+1}''(x_i) = 0$:

$$2a_i x_i + b_i = 2a_{i+1} x_{i+1} + b_{i+1}$$
$$2a_i x_i + b_i - 2a_{i+1} x_{i+1} - b_{i+1} = 0$$

This, again, yields $n-1$ additional equations.
So far, there are $2n + 2(n-1)$ equations. There are multiple methods to derive the last two equations; all of them rely on specific assumptions about the first and last cubic splines and their derivatives.

After obtaining the necessary $4n$ equations one can create the matrix form and solve it to get the coefficients.

2.4 Derivatives at the Endpoints

In this section, references such as [2],[4],[5] were used.

As mentioned, coming up with derivatives at the endpoints is a special step in the construction of the spline. How the problem should be approached depends on the data and how much information one has about the endpoints or how the curve behaves beyond them. In the end, in most cases, a certain assumption has to be made: even without sufficient information, the two derivative values at the endpoints are necessary for a sufficient number of equations. Choosing these two values will have an impact on the whole spline (if the knots are around equally distant, this effect becomes smaller as we approach the middle of the spline). In the following, only a few methods will be listed:

Natural Spline: The most simple approach: it sets the second derivative equal to 0 at both endpoints. The result is a flatter spline, especially around the endpoints. The approach is useful when no information is available about the endpoints.

Not-a-Knot Spline: Forces third-degree continuity at the second and second to last points. Useful when the data points are dense.

Periodic Spline: Assumes that the first and second derivatives are equal at the endpoints, resulting in a repeating curve. Provides an ideal choice for cyclical data.

Extrapolated Spline: Estimates the first derivatives at the endpoints using finite differences. For example, forward difference can be used for the left endpoint (with the help of the second point) and the backward difference for the right endpoint.

2.5 Selecting Knots

For this section, information from book [2] is used.

Sometimes, one does not work with predefined data points and has the chance to choose the knots for the spline interpolation process. There are multiple methods to choose the best knots, depending mainly on the level of precision one wants to achieve with the approximation. One of the main principles is to place more points in regions where the function exhibits high variability, thus attempting to capture the "difficult" shape there. One can save on computations with placing fewer knots to parts where the function has low variability. More advanced methods include cross-validation for the selection of knots where the place of knots is adjusted based on the error compared to the data.

3 Implementation

3.1 General

Spline interpolation is implemented in a wide range of fields. An example of such a field is CNC (Computer Numerical Control) machines, or laser cutting machines (toolpath generation and contour smoothing). As mentioned, spline interpolation is essential for generating toolpaths that guide the cutting or machining process. Another real-life application example is computer graphics and animation, where the method is used to create smooth curves. Furthermore, the technique plays a vital role in medical imaging, particularly when reconstructing 3D images from a series of 2D scans. Moreover, spline interpolation is also used in Machine Learning (ML), particularly Feature Engineering, on which will be expanded in the following subsection.

3.2 Feature Engineering

Feature Engineering (the process of selecting, manipulating and transforming raw data into features that can be used in supervised learning) is highly applicable in Machine Learning. The difference between a good ML model and an exceptional one often lies in the quality of the Feature Engineering part.

When engineering new features, spline interpolation is used widely: for example, when the relationship between input features and the target variable is nonlinear, the method can be used to transform features into a new space where these relationships are more linear and better captured, thus providing higher accuracy and better results. In regression problems, splines are used to model complex relationships without needing to specify the functional form explicitly. Spline interpolation is critical for many ML models that cannot handle nonlinear variables effectively. Numerous ML models would not be able to work with these types of variables effectively themselves. Thus, spline interpolation proves to be a highly useful and powerful tool even in Machine Learning.

4 Evaluation of the Method

4.1 Advantages

In this section, materials from papers [7], [8] are used.

Spline interpolation ensures that both the function and its derivatives are continuous, making the curve visually appealing and mathematically smooth. Also, different boundary conditions can be applied to control the behavior of the spline at its endpoints:

1) Natural Spline: The second derivative at both endpoints is set to zero:

$$S''(x_1) = S''(x_n) = 0.$$

This results in a "freely bending" curve with minimal curvature at the boundaries.

2) Clamped Spline: The first derivative (slope) at the endpoints is specified:

$$S'(x_1) = f'(x_1), \quad S'(x_n) = f'(x_n).$$

This ensures the spline has a specified tangent at the endpoints.

3) Not-a-Knot Condition: The third derivative continuity is enforced at the second and second-to-last points:

$$S'''(x_2) = S'''(x_{n-1}).$$

This makes the second and second-to-last points behave as if they are not "knots," reducing overfitting.

4) Periodic Spline: The function and its first and second derivatives are continuous between the first and last points:

$$S(x_1) = S(x_n), \quad S'(x_1) = S'(x_n), \quad S''(x_1) = S''(x_n).$$

Used for cyclic or repeating data.

Spline interpolation can be used to minimize the integral-square measure of the second derivative to achieve the best possible approximation for given conditions, minimizing oscillations and errors. According to the theorem, if $F(x)$ and its first two derivatives are continuous, $-\infty < x < \infty$, and the period L of a function (length of the interval where the function is defined and repeats itself) are given, then the values of x_k where $0 = x_0 < x_1 < \cdots < x_n = L$ satisfy these conditions.

$$E = \int_0^L \left[F'' - y''\right]^2 dx.$$

This is an unique approximation for $F''(x)$ within an additive constant.

Also, it offers flexibility in selecting the degree of polynomials used for interpolation, such as linear, quadratic, cubic, or higher degrees. This allows to adjust the level of complexity based on the nature of the data and the desired accuracy. Lower-degree polynomials are simpler and computationally efficient, while higher-degree polynomials can model more complex data.

4.2 Disadvantages

Spline interpolation can cause oscillations or overshooting, especially when the data points are unevenly spaced or when higher-degree splines are used. Additionally, the method faces difficulties when the x-values are very large and far apart. In such cases, the interpolation can become unstable, producing results that are significantly inaccurate. This makes cubic spline interpolation less reliable for data sets with large gaps or extreme values.

Another disadvantage is that Spline interpolation can sometimes produce negative values, even when the original data is entirely positive. This happens because spline algorithms construct piecewise polynomials between consecutive points, assuming the points are ordered. This becomes a serious problem if the values being interpolated, like mass, density, or population size, cannot be negative. It gets even worse if the data is used in ways that require positive values, such as taking square roots or using a logarithmic scale.

Moreover, cubic spline interpolation is sensitive to the choice of boundary conditions, which affect how the spline behaves near the endpoints of the data. Different boundary conditions, like setting the slope or assuming a flat curve at the edges, can change the interpolation results. If the wrong boundary conditions are used, the spline may overshoot, oscillate, or produce inaccurate values near the endpoints. Choosing the right boundary conditions is important for accurate and reliable interpolation.

5 Conclusion

All points considered, spline interpolation is a powerful technique in numerical analysis. By dividing the interval and fitting piecewise lower-degree polynomials to the data points, the method offers several advantages over polynomial interpolation.

Cubic spline interpolation is the most often implemented spline interpolation type. Due to the method's ability to ensure the continuity of both the function and its first and second derivatives, the smoothness provides an advantage in various real-life application scenarios.

However, spline interpolation does not come without its limitations. Diverse challenges discussed in this paper highlight the importance of consideration of problem-specific constraints when choosing the right interpolation method.

In conclusion, spline interpolation is an essential method for approximating functions and datasets with precision and smoothness. Its importance in the mathematical field as well as numerous real-life application examples underscore its value in both theory and practice. At this point in time, it can be stated with certainty that spline interpolation will remain a key technique for years to come.

6 References

[1] "SPLINE INTERPOLATION." n.d. https://redox-college.s3.ap-south-1.amazonaws.com/kmc/2020/Mar/29/ZUAJ5W5IIaaZmQQ17QrH.pdf.

[2] Endre Süli, David F. Mayers: An Introduction to Numerical Analysis. Cambridge University Press, 2003

[3]Ascher, U M, and Chen Greif. 2011. A First Course in Numerical Methods. Philadelphia: Society For Industrial And Applied Mathematics.

[4]Erdogan Kaya: Spline Interpolation Techniques. Journal of Technical Science and Technologies; ISSN 2298-0032, 2014

[5]McKinley, Sky, and Megan Levine. n.d. *Cubic Spline Interpolation.* `https://www.rajgunesh.com/resources/downloads/numerical/cubicsplineinterpol.pdf`.

[6]Kythe, Prem K., and Michael R. Schäferkotter. 2004. *Handbook of Computational Methods for Integration.* CRC Press. `https://jtst.ibsu.edu.ge/jms/index.php/jtst/article/view/56`.

[7] "Best Approximation Properties of the Spline Fit." *Journal of Mathematics and Mechanics* 11, no. 2 (1962): 225–234. `http://www.jstor.org/stable/24900920`. "Interpolation – Spline Interpolation."

[8]Pollock, D. n.d. *Smoothing with Cubic Splines.* `https://monoceros.physics.muni.cz/~jancely/NM/Texty/Numerika/CubicSmoothingSpline.pdf`.

Walsh, J. L., J. H. Ahlberg, and E. N. Nilson.