Theme: Cubic Spline Interpolation

Focus points: Motivation, Goal, Transition Conditions and Transition Conditions as Equations

Prerequisites

Polynomial Interpolation, Hermite Interpolation, Solving of linear systems of equations, Continuity and First and Second order Derivatives ...

Concept description

The video is structured in 3 sections. The motivation behind cubic spline interpolation and a comparison between it and cubic hermite interpolation is being handled in the first section, as well as the conditions required to circumvent the main disadvantage of hermite interpolation. The second section illustrates hands-on the mathematical mechanism behind the cubic spline interpolation, as well as solving a common misunderstanding about monomial bases in Python. The third and last section of the video shows a couple of examples generated by natural cubic spline interpolation, and presents the viewer with the theme comprehension questions. I chose to structure the video in this way because everything builds upon previous notions and is connected together.

Answers for the theme comprehension questions

1. **Why do we use cubic polynomials to interpolate our data points?**
   The reason we use cubic polynomials in our interpolation is that it gives us just enough degrees of freedom to interpolate our data. Remember that the conditions for a smooth transition are that the first and the second order derivatives of our polynomials should be the same. If we would've used linear polynomials, we could have fulfilled the conditions only if the points were situated on the same line. The same applies to quadratic polynomials, the conditions would be fulfilled only if the points were being found on the same parabola. Using cubic (and higher degree) polynomials allows us to fulfill these conditions for a smooth transition, no matter how the points are situated.

2. **Why is cubic spline interpolation preferred over higher degree polynomial interpolation?**
   Using higher degree polynomial interpolation results in chaotic oscillations of polynomial curves because of the so called "Runge's Phenomenon". This results in an inaccurate interpolation of the data points. This behaviour is avoided when using cubic spline interpolation, as Runge's Phenomenon only occurs when using higher degree polynomials, so our generated interpolation is smoother.

Table of Contents

- **Focus point 1** 00:03
- **Focus point 2** 04:46
- **Python Example 1** 08:04, Function
- **Comprehension questions** 08:39, 08:47 `natural_cubic_interpolation()`

Sources and References

- WR Skript Seiten 69-74
- Tutorial 4 part 3 hermite splines Jupyter Notebook from ISIS
- Cubic spline image
- Cubic hermite image

Veröffentlichung

Ich bin einverstanden, dass mein Video als Unterrichtsmaterial für Studierende der TU im Kurs Wissenschaftliches Rechnen in zukünftigen Veranstaltungen verwendet wird: ☑
Falls das Video verwendet wird, soll die Verwendung anonym (ohne Nennung des Namens) erfolgen: ☑