MAD exercise session 5

Cubic splines

Pascal Auf der Maur

Recap

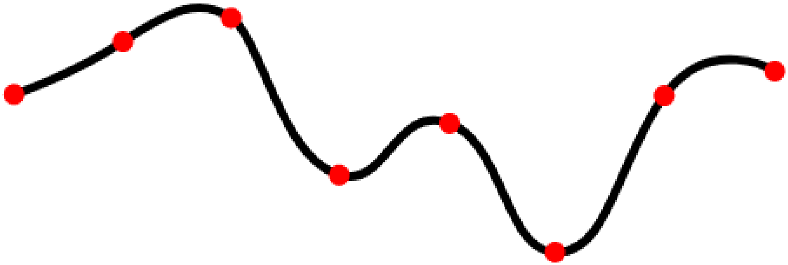
- Last time we discussed Lagrange polynomials to interpolate between datapoints
- We observed that this fit leads to unwanted oscillations because every point influences local behaviour

We want a method that only takes local points into consideration for the fit

Cubic Splines

Idea:

- We fit piecewise cubic polynomials between datapoints
- We require the polynomials to go through each point and have C1 continouity as boundryconditions



Cubic Splines

- Since the polynomials are cubic, they are linear in their second derivative:

$$f''(x) = f_i'' \frac{x_{i+1} - x}{\Delta_i} + f_{i+1}'' \frac{x - x_i}{\Delta_i}$$

- We interpolate the second derivative between points
- The local second derivatives are unknown which have to be determined

Cubic Splines

- By integrating twice we get the the following equation, where C_i and D_i are integrationconstants

$$f''(x) = f_i'' \frac{(x_{i+1} - x)^3}{6\Delta_i} + f_{i+1}'' \frac{(x - x_i)^3}{6\Delta_i} + C_i(x - x_i) + D_i$$

- Then we enforce the aforementioned boundryconditions (see lecturenotes)

Cubic Splines

$$\frac{\Delta_{i-1}}{6} f''_{i-1} + \left(\frac{\Delta_{i-1} + \Delta_i}{3}\right) f''_i + \frac{\Delta_i}{6} f''_{i+1} = \frac{y_{i+1} - y_i}{\Delta_i} - \frac{y_i - y_{y-1}}{\Delta_{i-1}}$$

- This is the resulting equation from which we can assemble a system of equation
- Because of the structure the resulting matrix will be tridiagonal which is easier to solve than a dense matrix

Cubic Splines

- By inspecting the equation we see that we cannot apply this equation to the first and last interval directly.
- Common boundryconditions for the first and last interval are:
  - Natural spline: $f_1'' = f_N'' = 0$
  - Parabolic runout: $f_1'' = f_2''$ and $f_N'' = f_{N-1}''$
  - Clamping: $f'(x_1) = f'(x_N) = 0$

B-Splines

- This concept can be generalized to fit polynomials of arbitrary degree
- The knot vector together with the desired degree determine how to derive the basisfunctions
- The basisfunctions determine the contribution of each point to the interpolated point
- An example program can be found on my gitlab account