Polynomial approximation

You want to create an equation that is as accurate as possible for a large number of values (x,y) by **polynomial approximation** .

Polynomials calculate deviating values yi for all values y. The error for each value is r = y - yi.

The sum of the error squares must be small and reach the value of 1.00 as well as possible. Each error is squared to avoid negative signs and to give greater weight to larger errors.

The polynomial should **not have any oscillations (ripples),** especially if the polynomial is to be differentiable, which provides information about the slope of the polynomial.

The differences are shown using the example of 2 polynomial approximations (y1,y2).

Value x	Value y	Equation y1	Equation y2
40.00	30.00	30.21	38.72
55.00	45.00	43.74	44.58
70.00	50.00	52.42	50.48
85.00	60.00	59.78	56.38
100.00	70.00	65.27	62.20
115.00	65.00	69.08	67.84
130.00	70.00	72.40	73.22
145.00	80.00	76.36	78.21
160.00	80.00	81.22	82.69
175.00	90.00	86.24	86.53
190.00	85.00	90.30	89.60
205.00	95.00	92.52	91.77
220.00	90.00	92.83	92.91
235.00	95.00	91.90	92.94
250.00	90.00	90.64	91.78
265.00	90.00	89.43	89.41
280.00	85.00	87.67	85.84
295.00	85.00	83.88	81.11
310.00	75.00	76.86	75.32
325.00	70.00	67.33	68.58
340.00	55.00	58.81	61.05
355.00	55.00	54.66	52.89
370.00	45.00	46.39	44.27

$$y1 = a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6 + hx^7 + ix^8 + jx^9 + kx^{10}$$

a	-3.9017042E+02
b	3.3047293E+01
c	-1.1267919E+00
d	2.1989777E-02
e	-2.6550143E-04
f	2.0654021E-06
g	-1.0507566E-08
h	3.4675302E-11
i	-7.1408468E-14
j	8.3289887E-17
k	-4.1978445E-20

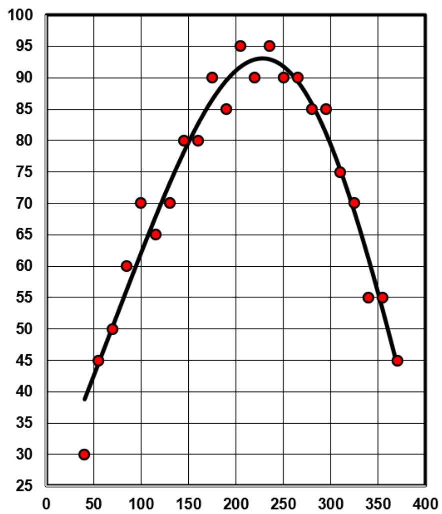
r^2=0.98

The sum of the squares of error is better than in polynomial y2. The waviness is too great and is not suitable for a differentiated derivative to provide information about the slope.

$$y2 = (a + cx + ex^2)/(1 + bx + dx^2)$$

a	2.3680357E+01
b	-3.5563061E-03
c	2.7797812E-01
d	6.3461946E-06
e	-7.4544055E-04

r^2=0.96



The sum of the squares of error is worse than in polynomial y1. However, the ripple is eliminated and is much more suitable for a derivation to provide information about the gradient.

This is the reason why we only use this polynomial type, but of a higher degree, for all thermodynamic values in our heat exchanger calculation software.

$$y2 = (a + cx + ex^2 + gx^3 + ix^4)/(1 + bx + dx^2 + fx^3 + hx^4)$$

Spline interpolation

Spline interpolation is a type of function approximation in which a new function is created that runs exactly through a given set of data points.

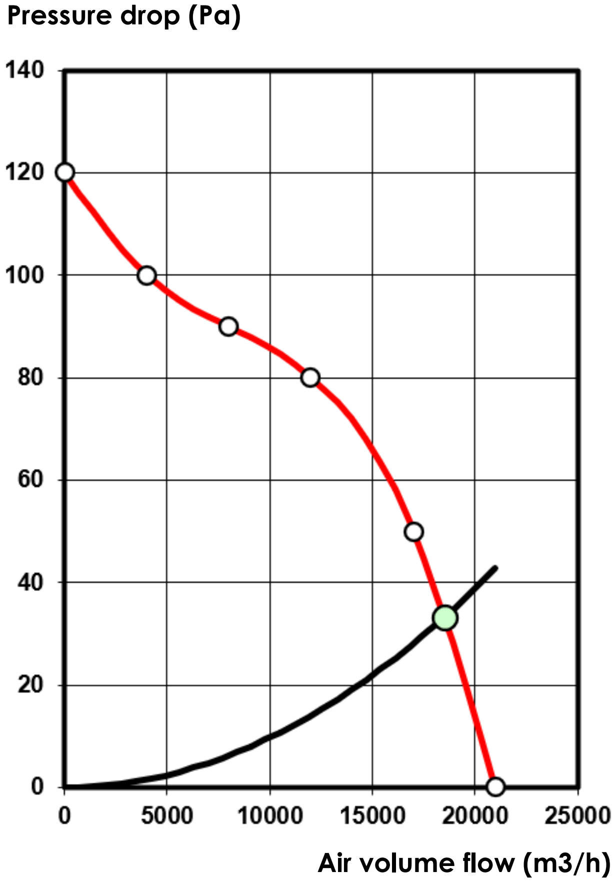
A spline of nth degree is a function that is composed of polynomials of no more than nth degree.

At the points where two polynomial pieces collide, certain conditions are set, such as that the spline (n-1) times is continuously differentiable.

Due to the piecewise definition, splines are more flexible than polynomials and yet relatively simple and smooth. As a result, spline interpolation does not have the disadvantages that arise from the strong oscillation (waviness) of polynomial approximations of higher degrees.

We use cubic splines, for example for fan characteristics with 6 data points, i.e. a spline interpolation with 5 cubic equations for a fan characteristic curve, a square pressure drop characteristic curve of a finned heat exchanger and the intersection between fan and heat exchanger.

$y = a + bx + cx^2 + dx^3$



Polynomial differentiate

Finally, we come back to the reason why we only use this fractional polynomial type 4 degrees **without oscillations (ripples)** for all thermodynamic values in our heat exchanger calculation software and how this can be differentiated.

$$y = \frac{a + cx + ex^2 + gx^3 + ix^4}{1 + bx + dx^2 + fx^3 + hx^4} \rightarrow y = \frac{m(x)}{n(x)}$$

$$\mathrm{m} = a + cx + ex^2 + gx^3 + ix^4 \rightarrow m' = c + 2ex + 3gx^2 + 4ix^3$$

$$\mathrm{n} = 1 + bx + dx^2 + fx^3 + hx^4 \rightarrow n' = b + 2dx + 3fx^2 + 4hx^3$$

$$\mathrm{y}' = \frac{nm' - mn'}{n^2}$$