Newton Polynomial Interpolation (Collocation)

The method of divided differences

1. The problem of collocation

The following example of visualized data originates from a former industrial project with von Roll AG.

Example 1.1: Five measurements of flow rates (volume per second = [cm^3/sec]) consist of $\boldsymbol{n} + 1 =$ **231** + 1 data each (there are 232 measurements). For the purpose of integration and presentation each data set has to be interpolated by a single curve that meets all the corresponding measurement points.

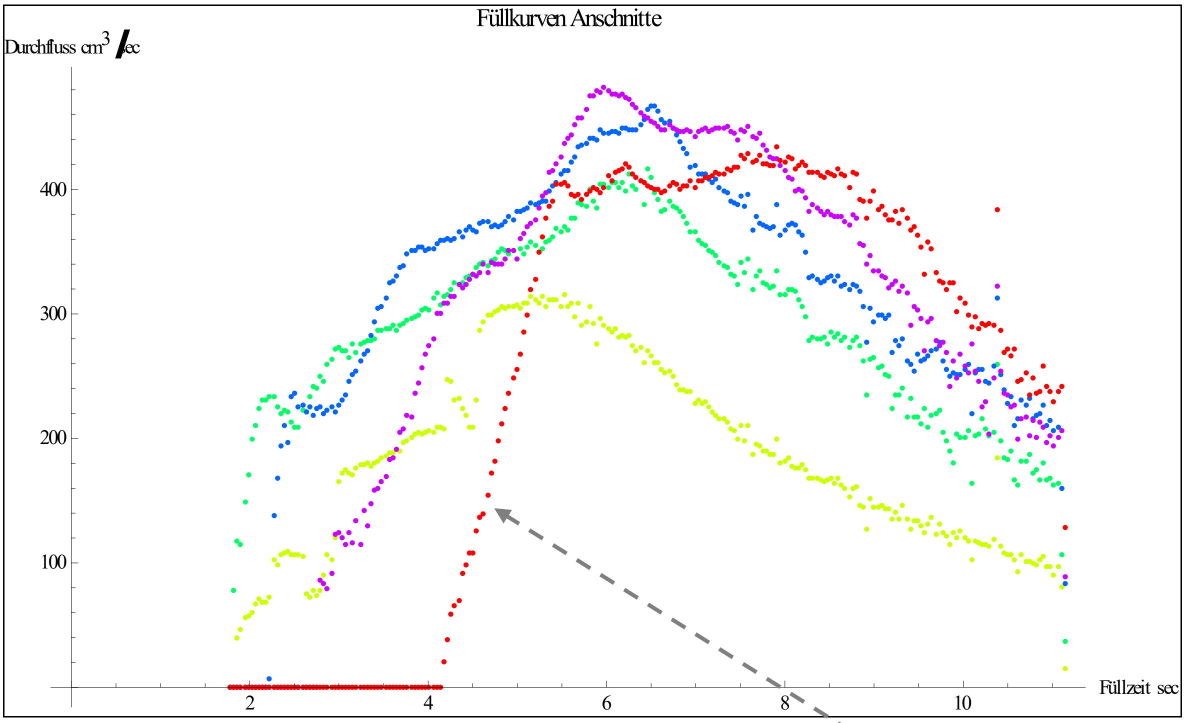


Figure 1.1: Point plots of five data sets (measurements) of flow rates.

This kind of interpolation problem is called collocation[1]. Concentrating now on a single measurement, e.g. the red one #5, and introducing the usual symbols x and y for the abscissa and ordinate axis, respectively, we have to find a polynomial formula[2]

$$\boxed{y(x) = p(x) = c_0 + c_1 x^1 + c_2 x^2 + + c_3 x^3 + \cdots + c_{m-1} x^{m-1} + c_m x^m \qquad (c_0, c_1, \ldots \in \mathbb{R})} \qquad \textbf{(1.1)}$$

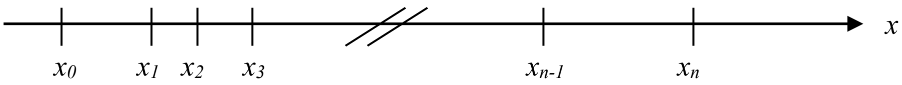
such that the degree m of the polynomial is minimal and the collocation conditions

[1] There are other methods of interpolation often used in practice like osculation (where in addition derivative conditions have to be met) or spline interpolation. Approximation methods, contrary to interpolation, relaxe the condition that all the data points have to be met. The most important approximation methods are called Least-Squares and Chebyshev (Min-Max), respectively. These methods can be combined, of course.

[2] There are other choices than polynomials, of course, like trigonometric polynomials or exponential functions. Polynomials have advantages concerning operations like evaluation, differentiation, integration and implementation of these operations.

$$\boxed{y(x_k) = p(x_k) = y_k \quad (k = 0,1,2,\ldots,n)} \qquad \textbf{(1.2)}$$

are met. The formula (1.2) is a short description of a linear system of n+1 equations in the (unknown) coefficients $c_0, c_1, \ldots, c_m$ of the polynomial. The data (measurement points) are given by a set of indexed coordinate pairs $(x_{k,}, y_k) \quad (k = 0,1,\ldots,n)$, usually beginning with index 0. The ordinate values y_k are called (function or measurement) values and the abscissa values x_k are called arguments:



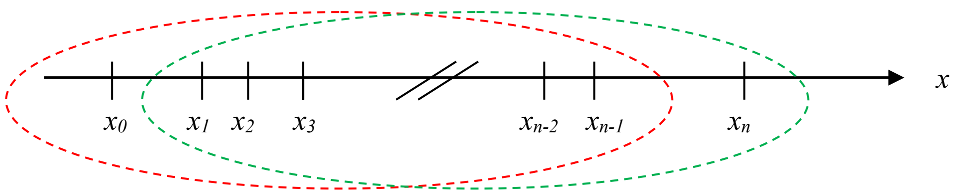
The linear system (1.2) written in schematic form reads as

$$\boxed{\begin{array}{lcl} y_0 & = & c_0 + c_1 x_0^{\;1} + c_2 x_0^{\;2} + + c_3 x_0^{\;3} + \cdots + c_{m-1} x_0^{\;m-1} + c_m x_0^{\;m} \\ y_1 & = & c_0 + c_1 x_1^{\;1} + c_2 x_1^{\;2} + + c_3 x_1^{\;3} + \cdots + c_{m-1} x_1^{\;m-1} + c_m x_1^{\;m} \\ \vdots & \vdots & \\ y_n & = & c_0 + c_1 x_n^{\;1} + c_2 x_n^{\;2} + + c_3 x_n^{\;3} + \cdots + c_{m-1} x_n^{\;m-1} + c_m x_n^{\;m} \end{array}} \qquad \textbf{(1.2')}$$

It consists of n+1 equations in the (unknown) coefficients $c_0, c_1, \ldots, c_m$ ("degrees of freedom"). By elementary linear algebra it is generally uniquely solvable if m = n. But normally, the problem of collocation (1.1) is not attacked by solving (1.2') because of significant numeric instability and inefficiency of computations. The form (1.2') results from the polynomial form (1.1) where a polynomial function is represented as a weighted sum of powers of x and thus $p(x)$ is a linear combination of the basis polynomials $1, x^1, x^2, \ldots, x^m$ (powers).

2. The Aitken-Neville recursion formula

Without explicit solving (1.2) it is possible to establish relations among different interpolating polynomials by partitioning the data set in two (overlapping) parts as indicated in the figure below.



The dashed red and green ovals, respectively, indicate the partition of the data arguments in two overlapping sets: *red* = $\{x_0, x_1, \ldots, x_{n-1}\}$ and *green* = $\{x_1, x_2, \ldots, x_n\}$. Assuming now that the indexed term $p_{0,1,2,\ldots,n-1}(x)$ indicates a polynomial interpolating the red data subset and the indexed term $p_{1,2,\ldots,n-1,n}(x)$ does the same for the green data set then the following relation holds between these "partial" polynomial terms and the "global" polynomial $p(x) = p_{0,1,2,\ldots,n-1,n}(x)$ solving the collocation problem (1.2):

$$\boxed{p(x) = p_{0,1,2,\ldots,n-1,n}(x) = \frac{(x-x_0)\,\overbrace{p_{1,2,\ldots,n-1,n}}^{green}(x) - (x-x_n)\,\overbrace{p_{0,1,2,\ldots,n-1}}^{red}(x)}{(x_n - x_0)}} \tag{1.3}$$

The Aitken-Neville formula (1.3) describes how the "global" interpolating polynomial is combined from the "partial" interpolating polynomials $p_{0,1,2,\ldots,n-1}(x)$ (red) and $p_{1,2,\ldots,n-1,n}(x)$ ($green$).

(Proof: The proof of the Aitken-Neville recursion formula is easily done by sequentially replacing x by the arguments x_k and evaluating $p(x_k)$:

$$p(x_0) = \frac{\overbrace{(x_0 - x_0)}^{0}\,p_{1,2,\ldots,n-1,n}(x_0) - (x_0 - x_n)\,\overbrace{p_{0,1,2,\ldots,n-1}(x_0)}^{y_0}}{(x_n - x_0)} = y_0$$

$$p(x_n) = \frac{(x_n - x_0)\,\overbrace{p_{1,2,\ldots,n-1,n}(x_n)}^{y_n} - \overbrace{(x_n - x_n)}^{0}\,p_{0,1,2,\ldots,n-1}(x_0)}{(x_n - x_0)} = y_n$$

$$p(x_k) = \frac{(x_k - x_0)\,\overbrace{p_{1,2,\ldots,n-1,n}(x_k)}^{y_k} - (x_k - x_n)\,\overbrace{p_{0,1,2,\ldots,n-1}(x_k)}^{y_k}}{(x_n - x_0)} = \frac{y_k(x_k - x_0 - x_k + x_n)}{(x_n - x_0)} = y_k$$

$$(k = 1,\ldots,n-1)$$

□)

A tabular representation of (1.3) appropriate for computational purposes is given below.

$$\begin{array}{l|lllll}
x_0 & y_0 = p_0 & & & & \\
x_1 & y_1 = p_1 & p_{01} & & & \\
x_2 & y_2 = p_2 & p_{12} & p_{012} & & \\
x_3 & y_3 = p_3 & p_{23} & p_{123} & p_{0123} & \\
x_4 & y_4 = p_4 & p_{34} & p_{234} & p_{1234} & \underline{p_{01234} = p_4(x)}.
\end{array}$$

Table 1.1: Tabular representation of the Aitken-Neville recursion formula. Assuming n = 4 and that $p_{0,1,2,3,4}$ has to be found ….

Bottom-up description of Table 1.1

The scheme begins with the trivial partial solutions $y_0 = p_0, y_1 = p_1, y_2 = p_2, y_3 = p_3, y_4 = p_4$. These are constants (order zero polynomials), one for each measurement point. Applying (1.3) repeatedly for consecutive pairs yields $p_{0,1}$ from p_0 and p_1, $p_{1,2}$ from p_1 and p_2, …. and so on.

Graphically, these are straight lines joining two consecutive measurement points; as polynomials $p_{0,1}$, $p_{1,2}$ ... are of order 1.

From $p_{0,1}$ and $p_{1,2}$ then the next column entry to the right, $p_{0,1,2}$, is determined again by (1.3). Graphically, this is a parabola joining the first three (consecutive) measurement points. As a polynomial it is of order 2. In this manner a parabola or 2nd order polynomial is computed by (1.3) for any three consecutive measurement points yielding $p_{0,1,2}$, $p_{1,2,3}$, $p_{2,3,4}$, ...

By continuation of this procedure rightwards from one column to the next one ... the scheme terminates with $p_{0,1,2,3,4}$. This is the final 4th order polynomial covering the whole measurement of five points. It is computed by (1.3) from the cubic polynomials $p_{0,1,2,3}$ and $p_{1,2,3,4}$.

3. The Newton basis polynomials

As mentioned at the end of the first section the form (1.2') is not appropriate for a numerical solution of the collocation problem (1.2), or, equivalently, the basis polynomials of powers $1, x^1, x^2, \dots, x^m, \dots$ are not a good choice for resolving (1.2) numerically. Newton found another basis of polynomials that leads to a lower-triangular form of (1.2) and allows for a surprisingly simple and efficient computational scheme resolving the collocation problem (1.2).

Definition 1.1: The Newton basis polynomials $\pi_k(x) \quad (k = 0,1,2,\dots,n)$ are defined by

$$\boxed{\begin{aligned} &\pi_0(x) = 1 \\ &\pi_1(x) = (x - x_0) \\ &\pi_2(x) = (x - x_0)(x - x_1) \\ &\vdots \\ &\pi_k(x) = (x - x_0)(x - x_1)\cdots(x - x_{k-1}) \\ &\vdots \\ &\pi_n(x) = (x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{aligned}} \tag{1.4}$$

It is obvious that $\pi_k(x) \quad (k = 0,1,2,\dots,n)$ is of degree k and that every polynomial $p(x)$ of degree m in the form (1.1) has a unique representation as a linear combination of Newton polynomials:

$$p(x) = a_0\pi_0(x) + a_1\pi_1(x) + a_2\pi_2(x) + \cdots + a_m\pi_m(x) \tag{1.5}$$

Moreover, when using (1.5) the linear collocation system (1.2) takes the following lower-triangular form:

$$\boxed{\begin{array}{lcl} y_0 & = & a_0 \\ y_1 & = & a_0 + a_1(x_1 - x_0) \\ y_2 & = & a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ \vdots & \vdots & \vdots \\ y_n & = & a_0 + a_1\pi_1(x_n) + a_2\pi_2(x_n) + \cdots + a_n\pi_n(x_n) \end{array}} \tag{1.6}$$

From (1.6) – by step-wise solution row-by-row – we immediately get the

Property 1.1: Stability of Newton coefficients against new data elements

The coefficient a_k in (1.6) is determined by the first k arguments $x_0, x_1, \ldots, x_k \quad (k = 0,1,2,\ldots,n)$ and thus adding a new data argument x_{n+1} to the given, original data set $x_0, x_1, \ldots, x_n$ does not alter the coefficients $a_0, a_1, \ldots, a_n$ of the original collocation polynomial.

At first sight the lower-triangular form of (1.6) looks attractive for a step-wise resolution row-by-row as indicated before Property 1.1 but still this is not optimal as far as efficiency is concerned. Bringing the form (1.6) together with the Aitken-Neville recursion formula allows for an elegant and fast computation of the Newton coefficients $a_0, a_1, \ldots, a_n$.

4. The "elegant" and fast resolution by divided differences

Due to Property 1.1 we can write the Newton coefficient a_k as $a(x_0, x_1, \ldots, x_k)$, a formula depending only on the arguments $x_0, x_1, \ldots, x_k \quad (k = 0,1,2,\ldots,n)$.

The computations below aim at finding a recursive formula for the Newton coefficients $a(x_0, x_1, \ldots, x_k)$, especially the leading coefficient $a(x_0, x_1, \ldots, x_n) \quad (n \in \mathbb{N})$.

Applying the Aitken-Neville recursion formula (1.3) to the collocation polynomial $p(x)$ in the Newton form (1.5) yields

$$p(x) = p_{0,1,2,\ldots,n-1,n}(x) = \frac{(x-x_0)\,p_{1,2,\ldots,n-1,n}(x) - (x-x_n)\,p_{0,1,2,\ldots,n-1}(x)}{(x_n - x_0)} =$$

$$\frac{\begin{array}{c}(x-x_0)(a(x_1) + a(x_1,x_2)(x-x_1) + \cdots + a(x_1,\ldots,x_n)(x-x_1)\cdots(x-x_{n-1})) - \\ (x-x_n)(a(x_0) + a(x_0,x_1)(x-x_0) + \cdots + a(x_0,\ldots,x_{n-1})(x-x_0)\cdots(x-x_{n-2}))\end{array}}{(x_n - x_0)} =$$

... splitting the factor $(x-x_n)$ into $(x-x_{n-1}) + (x_{n-1} - x_n)$ and correspondingly expanding in the nominator above we continue with ...

$$= \frac{\begin{array}{c}(x-x_0)(a(x_1) + a(x_1,x_2)(x-x_1) + \cdots + a(x_1,\ldots,x_n)(x-x_1)\cdots(x-x_{n-1})) - \\ ((x-x_{n-1}) + (x_{n-1}-x_n))(a(x_0) + a(x_0,x_1)(x-x_0) + \cdots \\ + a(x_0,\ldots,x_{n-1})(x-x_0)\cdots(x-x_{n-2}))\end{array}}{(x_n - x_0)} =$$

$$= \frac{\begin{array}{c}a(x_1)(x-x_0) + a(x_1,x_2)(x-x_0)(x-x_1) + \cdots + a(x_1,\ldots,x_n)(x-x_0)(x-x_1)\cdots(x-x_{n-1}) - \\ a(x_0)(x-x_{n-1}) - a(x_0,x_1)(x-x_0)(x-x_{n-1}) - \cdots - a(x_0,\ldots,x_{n-1})(x-x_0)\cdots(x-x_{n-2})(x-x_{n-1}) \\ -(x_{n-1}-x_n)(a(x_0) + a(x_0,x_1)(x-x_0) + \cdots + a(x_0,\ldots,x_{n-1})(x-x_0)\cdots(x-x_{n-2}))\end{array}}{(x_n - x_0)} =$$

The only terms with maximal degree n are encircled in the blue oval, all other parts are of lower degree and do not concern $a(x_0, x_1, \ldots, x_n)$. From this it is obvious that the leading coefficient $a_n = a(x_0, x_1, \ldots, x_n)$ is equal to the <u>divided difference</u> $\dfrac{a(x_1, x_2, \ldots, x_n) - a(x_0, x_1, \ldots, x_{n-1})}{(x_n - x_0)}$. Renaming $a(x_0, x_1, \ldots, x_n)$ by $y(x_0, x_1, \ldots, x_n)$ as is usual in Newton interpolation and considering that

n was arbitrary in $\mathbb{N}$ we can write the recursion formula for the Newton coefficients which are also called divided differences:

$$\boxed{y(x_0,x_1,\ldots,x_k) = \frac{y(x_1,x_2,\ldots,x_k) - y(x_0,x_1,\ldots,x_{k-1})}{(x_k - x_0)} \quad (k = 0,1,\ldots,n)} \tag{1.7}$$

The table below shows the divided differences $y(x_0,x_1,\ldots,x_k)$ for k = 0,1,2,3:

k = 0	$y(x_0)$
k = 1	$\dfrac{y(x_1)-y(x_0)}{(x_1-x_0)}$
k = 2	$\dfrac{y(x_1,x_2)-y(x_0,x_1)}{(x_2-x_0)}$
k = 3	$\dfrac{y(x_1,x_2,x_3)-y(x_0,x_1,x_2)}{(x_3-x_0)}$

Table 1.2: Some divided differences up to order 3

There is a recursive tabular scheme for computing divided differences analoguous the the one in Table 1.1. This is illustrated in the next example.

Example 1.2: Collocation polynomial for the data set
{(0,1), (1,1), (2,2), (4,5)} = $\{(x_k, y_k) \mid k = 0,1,\ldots,n = 3\}$.

x_k	y_k			
0	1			
		0		
1	1		$\frac{1}{2}$	
		1		$-\frac{1}{12}$
2	2		$\frac{1}{6}$	
		$\frac{3}{2}$		
4	5			

Table 1.3: Tabular scheme for computing divided differences column-by-column. The encircled values in the top diagonal correspond to the divided differences (Newton coefficients)

$y(x_0)$, $y(x_0,x_1)$, , $y(x_0,x_1,x_2,x_3)$.

From these the collocation polynomial is determined in Newton form:

$$\boxed{\begin{aligned}&y(x) = p(x) = 1 + 0\pi_1(x) + \frac{1}{2}\pi_2(x) + \frac{-1}{12}\pi_3(x) = \\ &1 + 0(x-0) + \frac{1}{2}(x-0)(x-1)) + \frac{-1}{12}(x-0)(x-1)(x-2)\end{aligned}}$$

Example 1.3: Polynomial interpolation of flow data (cont. Example 1.1)

The collocation polynomial (of expected degree 231) in expanded short form[1] for the red data set (#5) is

[1] In most computer implementations the collocation polynomial is internally represented in Horner form reducing significantly the number of multiplications for numerical evaluation. This is advantageous for numerical evaluation (and thus plotting, e.g.) but disadvantageous for operations like addition, integration and differentiation.

```
1.48918×10^118-6.70261×10^119 x+ ... +8.22523×10^-53 x^229-1.1149×10^-55
x^230+7.51708×10^-59 x^231
```

Plotting the data set together with the collocating polynomial reveals the Runge phenomenon (oscillations with high frequencies and amplitudes towards the boundaries of the arguments range).

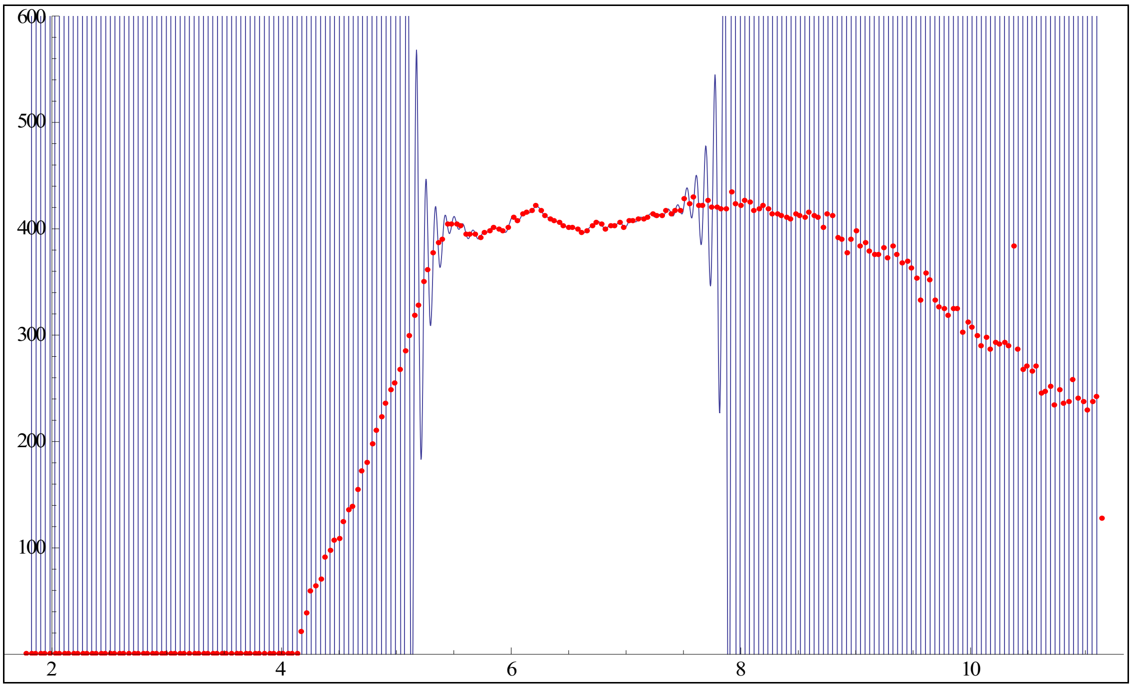


Figure 1.2: Point plot of data set #5 together with a restricted plot of the interpolating collocation polynomial of degree 231. High degrees are a potential risk for oscillations with high frequencies and amplitudes towards the boundaries of the arguments range.

A very useful property of divided differences is symmetry with respect to the arguments:

Property 1.2: Symmetry of divided differences

Any divided difference $y(x_0, x_1, \ldots, x_k)$ is independent from the order of its arguments $x_0, x_1, \ldots, x_k \quad (k \in \mathbb{N})$.

(Proof: This is a consequence of Property 1.1 and the fact that the collocation polynomial is unique and independent from the order of arguments. When interpolating only with respect to $x_0, x_1 \cdots, x_k$ only $\pi_k(x)$ contributes to the power x^k, independently of the ordering of $x_0, x_1 \cdots, x_k$. But the coefficient of the leading power x^k is unique, thus this is always equal to $y(x_0, x_1, \cdots, x_k)$.

$\square$)

5. The collocation error formula

Interpolating a (model) function $y = f(x)$ by a polynomial $p(x)$ collocating with the function f at arguments $x_0, x_1, \ldots, x_n$ means that there is some kind of error between the function f and the collocation polynomial p. The next theorem gives an error measure and a theoretically interesting description of the error by higher derivatives of f.

Theorem 1.1: Collocation error formula ($y = f(x)$)

$$\boxed{y(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_{n-1})(x-x_n) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x)} \qquad \textbf{(1.8)}$$

whereas $x \in \left[\min(x_k), \max(x_k)\right] \qquad \xi \in \left(\min(x_k), \max(x_k)\right)$

(Proof: Schaum's Outline of Numerical Analysis Chapter 2, Problem 2.7. □)

The error formula (1.8) shows that the error $y(x) - p(x)$ is determined by the derivative of order n+1 - evaluated at some (normally unknown) intermediate value ξ depending on x - and the Newton basis polynomial of order n+1: $\pi_{n+1}(x)$.

If n is large and the the derivative $\frac{f^{(n+1)}(x)}{(n+1)!}$ takes large values in the arguments range then there is a potential risk for oscillations towards the boundaries of the arguments range (Runge phenomenon), especially for uniformly distributed arguments. The section on Runge's phenomen below gives more details on that topic.

Another corollary from Theorem 1.1 is the representation of divided differences as derivatives:

Corollary 1.1: Divided differences represented as derivatives ($y = f(x)$

If $x \in \left(\min(x_i), \max(x_i)\right)$ and $x \neq x_i \quad (i = 0,1,\ldots,n \in \mathbb{N})$ then

$$\boxed{y(\underbrace{x, x_0, x_1, \ldots, x_n}_{n+2}) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \qquad \xi \in \left(\min(x_i), \max(x_i)\right)} \qquad \textbf{(1.9)}$$

$$\boxed{y(\underbrace{x_0, x_1, \ldots, x_n}_{n+1}) = \frac{f^{(n)}(\xi)}{n!} \qquad \xi \in \left(\min(x_i), \max(x_i)\right)} \qquad \textbf{(1.9')}$$

(Proof: From Theorem 1.1 we find

$$y(x) = p(x) + \frac{y^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x) \qquad x, \xi \in (\min x_i, \max x_i)_{i=0,1,\ldots,n}$$

Considering now $x = x_{n+1} \neq x_i \quad (i = 0, \cdots, n)$ as the argument of a further measurement the coefficient $\frac{y^{(n+1)}(\xi)}{(n+1)!}$ of the Newton polynomial $\pi_{n+1}(x)$ by (1.7) must equal the divided difference $y(x_0, \ldots, x_n, x) = y(x_0, \ldots, x_n, x_{n+1})$. Altogether we have

$$y(\underbrace{x_0, \ldots, x_n, x}_{(n+2)}) = \frac{y^{(n+1)}(\xi)}{(n+1)!} \qquad x = x_{n+1}, \xi \in (\min x_i, \max x_i)_{i=0,1,\ldots,n}$$

and the assertion (1.9) follows by Proposition 1.2 (symmetry). Finally, (1.9') is an application of (1.9) □)

Corollary 1.1 says that a divided difference can be considered as a derivative. For the case n = 1 this is not a surprise, because $y(x_0,x_1)=\dfrac{\Delta y_0}{\Delta x_0}=\dfrac{y(x_1)-y(x_0)}{x_1-x_0}=y'(\xi) \qquad \xi\in(x_0,x_1)$ by the classical mean value theorem of calculus in one variable.

6. The Runge phenomenon and Chebyshev arguments

The so-called Runge phenomenon of polynomial collocation, i.e. oscillation of errors towards the boundaries of arguments range, was discovered approximately 100 years ago (!). The next example serves as an illustration.

Example 1.4: Runge phenomenon

Interpolating the non-polynomial (model) function $y=f(x)=\dfrac{1}{1+25x^2}$ with n+1 uniformly distributed arguments on the interval [-1, 1] leads to error oscillations towards the boundary values -1 and 1.

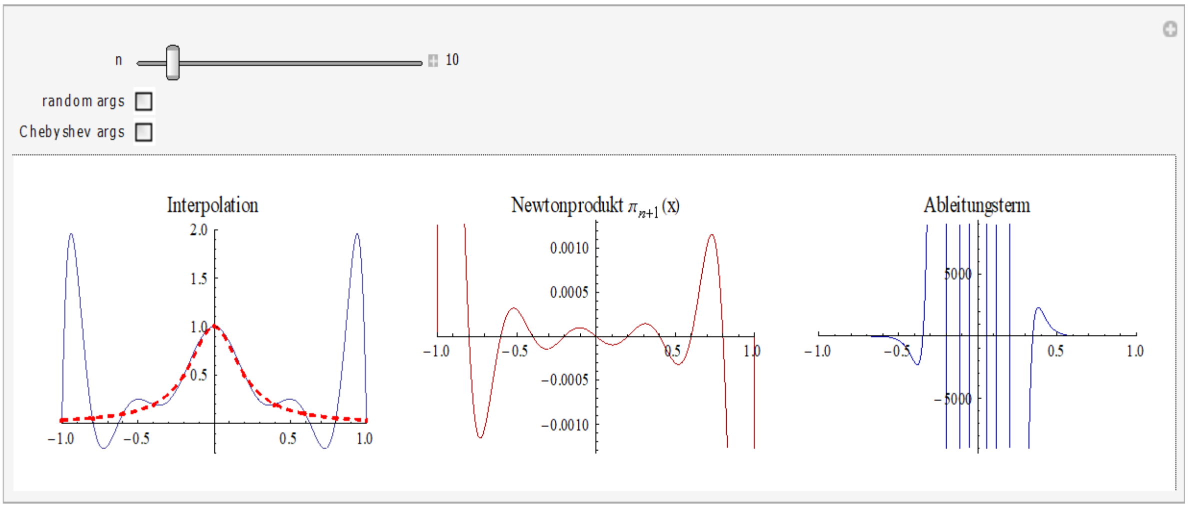


Figure 1.3abc: The first plot (from left) shows the model function y and the collocation polynomial for $n=10$. The middle plot shows the Newton polynomial $\pi_{n+1}(x)$. On the right the derivation term $\dfrac{y^{(n+1)}(x)}{(n+1)!}$ is shown. The derivation term oscillates with large amplitudes as does $\pi_{n+1}(x)$ (middle) towards the boundary values. Looking at this plots it is not surprising that the error term $\dfrac{y^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x)$ of Theorem 1.1 (formula 1.8) itself oscillates towards the boundaries.

A remedy against the error oscillations in the Example 1.4 must be based on another, <u>non-uniform distribution</u> of the arguments since we cannot change the derivatives of y. It is promising to look for arguments $x_k \quad (k=0,1,2,\ldots,n)$ minimizing the maximum absolute amplitude of $\pi_{n+1}(x)$, i.e. minimizing $\max\limits_{-1\le x\le 1}\left|\pi_{n+1}(x)\right|$ (such optimization problems are briefly called <u>min-max</u>). The problem has an unique solution: Choosing Chebyshev arguments minimizes the maximum amplitude $\max_{-1\le x\le 1}|\pi_{n+1}(x)|$ to the value $\frac{1}{2^n}$.

Theorem 1.2: Optimality of Chebyshev arguments[1]

Choosing the Chebyshev arguments

$$\boxed{x_k = \cos\left(\frac{2k+1}{2(n+1)}\pi\right) \quad (k=0,1,\ldots,n)} \tag{1.10}$$

in the interval [-1, 1] minimizes the maximum absolute amplitude $\max_{-1\le x\le 1}|\pi_{n+1}(x)|$ to the value $\frac{1}{2^n}$.

The proof of Theorem 1.2 will be given in a later chapter on least-squares approximation and is omitted here.

The next figures illustrates Theorem 1.2.

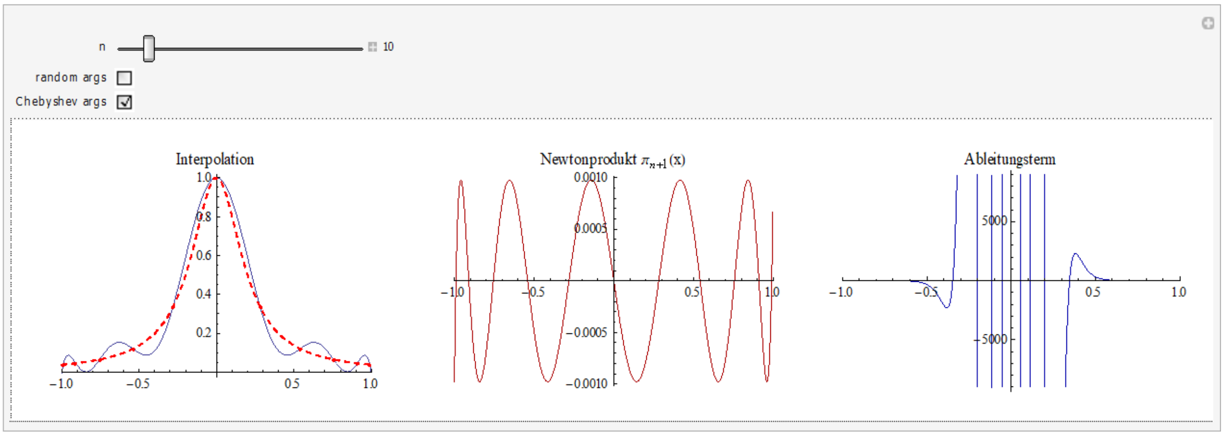


Figure 1.4abc: The first plot (from left) shows the model function y and the collocation polynomial for the Chebyshev arguments $x_k = \cos\left(\frac{2k+1}{2(n+1)}\pi\right)$ $(k=0,1,\ldots,n=10)$. The middle plot shows the optimized Newton polynomial $\pi_{n+1}(x)$ oscillating uniformly between the values $\pm\frac{1}{2^n}$ in the interval [-1, 1].

[1] These arguments are the zeros of the (n+1)th-order Chebyshev polynomial $T_{n+1}(x) = \cos((n+1)\arccos x)$ defined by trigonometry. The Chebyshev polynomials play a crucial role in approximation theory, e.g. least-squares approximation or min-max approximation.

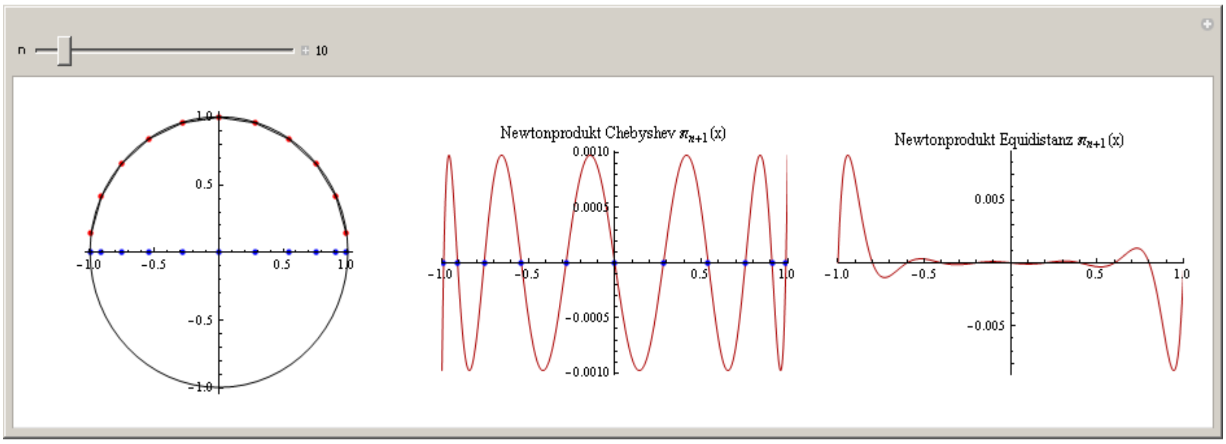


Figure 1.5abc: The first plot (from left) shows a geometric property of the Chebyshev arguments $x_k = \cos\left(\frac{2k+1}{2(n+1)}\pi\right) \quad (k = 0,1,\ldots,n = 10)$. They can be constructed by projection from uniformly distributed, symmetric points on the upper half circle onto the abscissa.

The middle plot shows the optimized Newton polynomial $\pi_{n+1}(x)$ oscillating uniformly between the values $\pm\frac{1}{2^n}$ in the interval [-1, 1] and the plot on the right side, finally, shows a non-optimal $\pi_{n+1}(x)$ corresponding to uniformly distributed arguments.

Example 1.5: Measurement of flow rates with Chebyshev arguments (Ex. 1.1 rev.)

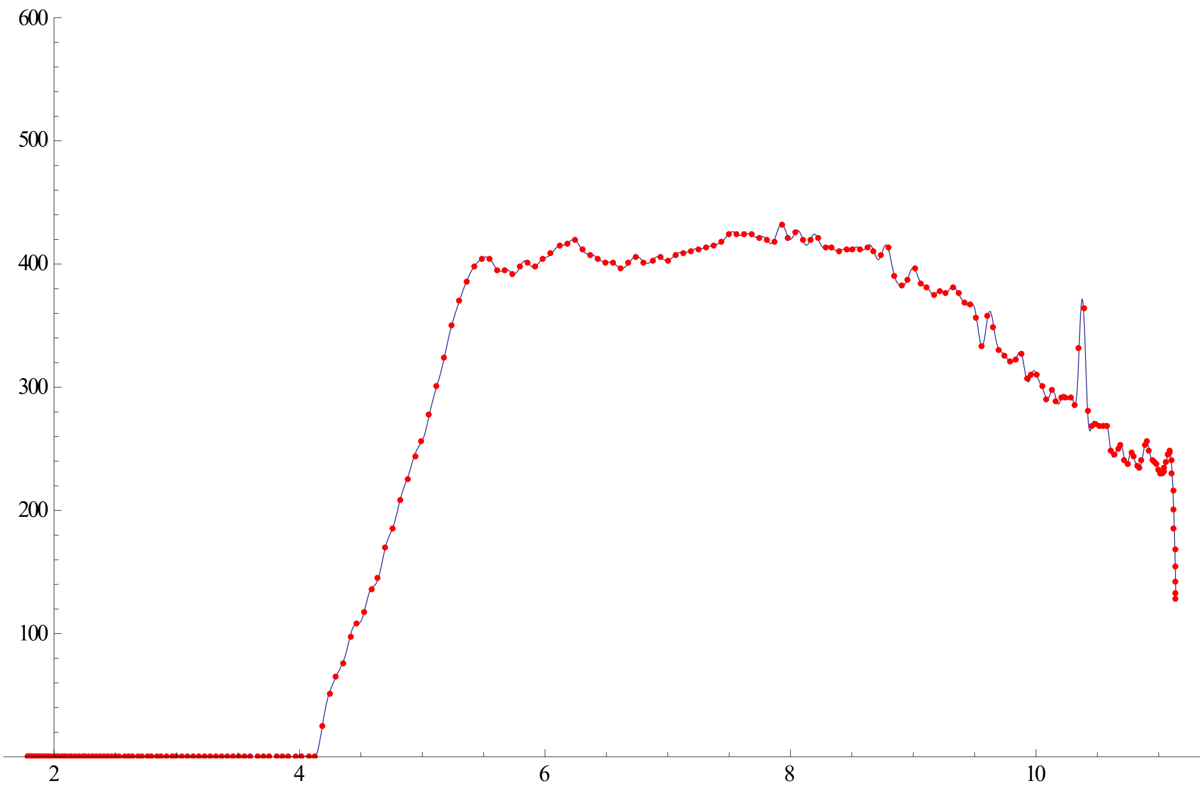


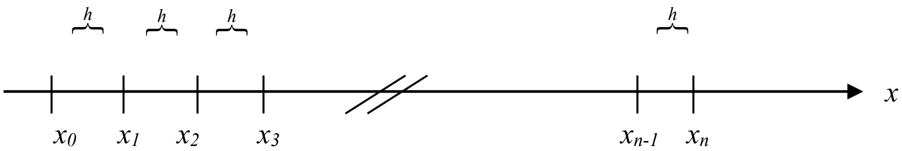
Figure 1.6: The plot shows a measurement of a flow rate (#5) with n+1 = 231+1 Chebyshev arguments in the time interval [a, b] = [1.7819, 11.1399] and the corresponding collocation polynomial. The measurement with Chebyshev arguments provides high resolutions (informations) towards the boundary values of [a, b] and avoids large oscillations of the collocation error.

The affine formula $x \mapsto a + \frac{b-a}{2}(x+1)$ transforms the interval [-1, 1] onto [a, b]; it was applied to the Chebyshev arguments $x_k = \cos\left(\frac{2k+1}{2(n+1)}\pi\right) \quad (k = 0,1,\ldots,n = 231)$ and a = 1.7819, b = 11.1399.

To be fair it has to be said that the choice of Chebyshev arguments has its drawbacks as: Impossibility of measurement, expense of measurement, high density at boundaries and last but not least leads to a non-embedded interpolation scheme. This means that increasing the number n (of measurements minus 1) implies that nearly all arguments and Newton basis polynomials change. The interpolation polynomial has to be recomputed and represented from scratch.

7. Uniformly distributed arguments and Newton's difference formula

The theory of collocation by Newton polynomials developed in the sections above simplifies in notation for the case of uniformly distributed arguments.



It is obvious that $x_k = x_0 + kh \quad (k = 0,1,2,\ldots,n \qquad h > 0)$. Writing the delta-symbol Δ for the difference operator, i.e. $\Delta y_k = y_{k+1} - y_k$ and the abbreviation Δ^m for the m-fold composition of Δ we get by elementary calculation that

$$\boxed{y(x_0, x_1, \ldots, x_k) = \frac{\Delta^k y_0}{h^k k!}} \tag{1.11}$$

From this and (1.7) we deduce

Theorem 1.3: Newton's difference formula for collocation

For uniformly distributed arguments the collocation polynomial p is given by the difference formula

$$\boxed{p(x) = y_0 + \frac{\Delta^1 y_0}{h^1 1!}\pi_1(x) + \frac{\Delta^2 y_0}{h^2 2!}\pi_2(x) + \cdots + \frac{\Delta^n y_0}{h^n n!}\pi_n(x) = \sum_{k=0}^{n} \frac{\Delta^k y_0}{h^k k!}\pi_k(x)} \tag{1.12}$$

By a limiting process $h \to 0$ or $x_k \to x_0 \quad (k = 1,2,\ldots,n)$, respectively, we get a classical result from calculus in one variable:

Corollary 1.2: Taylor-Approximation and Lagrange error term

If $y = f(x)$ is a (model) function with continuous derivatives up to order n on the interval $[x_0, x]$ and y is (n+1)-times differentiable on (x_0, x) then

$$f(x) = \overbrace{y_0 + \frac{y'(x_0)}{1!}(x-x_0) + \frac{y''(x_0)}{2!}(x-x_0)^2 + \cdots + \frac{y^{(n)}(x_0)}{n!}(x-x_0)^n}^{Taylor\ polynomial} + \underbrace{\frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}}_{Lagrange\ error\ term} \qquad \xi \in (x_0, x) \tag{1.13}$$

(Proof: The assertion follows from Theorem 1.1 and Theorem 1.3 when considering that

$$\lim_{h\to 0} \frac{\Delta^k y_0}{h^k k!} = \frac{y^{(k)}(x_0)}{k!} \qquad (k = 0,1,\ldots,n) \quad .$$

which itself is a consequence from Corollary 1.1 (1.9). ∎)

8. Standard monomial polynomials from Newton polynomials – Stirling numbers of the 1st kind

For computational purposes as integration $\int(\ldots)dx$ or derivation $\frac{d(\ldots)}{dx}$ the representation of a polynomial as a linear combination of Newton polynomials $p(x) = a_0\pi_0(x) + a_1\pi_1(x) + \cdots + a_n\pi_n(x)$ is not the best choice.
In the case of equi-distant measurement arguments (as in the previous section) there is a simple and elegant scheme transforming the Newton representation (1.5) into a standard monomial form:

$$p(x) = c_0 + c_1(x-x_0)^1 + c_2(x-x_0)^2 \cdots + c_n(x-x_0)^n \tag{1.14}$$

For the computation of the coefficients $(c_0, c_1, \ldots, c_n)$ from the Newton coefficients $(a_0, a_1, \ldots, a_n)$ there is a sophisticated scheme based on the Stirling numbers of the 1st kind:

$$s(n,k) \text{ with } \left(n = 0,1,2,\ldots \quad k = 0,\ldots,n \quad n \in \mathbb{N}_0\right). \tag{1.15}$$

The next Property introduces the meaning of the Stirling numbers with respect to the equi-distant normalized arguments {0, 1, 2, ..., n}.

Property 1.3: Stirling Numbers of the 1st kind

a) A normalized Newton polynomial $\pi_n(x) = x(x-1)(x-2)\cdots(x-(n-1))$ expands into the standard monomial form $s(n,1)x + s(n,2)x^2 + s(n,3)x^3 + \cdots + s(n,n)x^n$.

b) $s(0,0) = 1, \quad s(n,0) = 0 \quad (n \in \mathbb{N}), \quad s(n,k) = 0 \quad (k > n \geq 0)$

c) The Stirling numbers fulfill the bi-variate recursion:

$$s(n+1,k) = s(n,k-1) - n \cdot s(n,k) \quad (n = 0,1,\ldots \quad k = 1,2,\ldots) \qquad \textbf{(1.16)}$$

(Proof: Part a) actually is a definition of the Stirling numbers: They constitute the coefficients in (1.14). From this b) is a conclusion; there is no other choice. Property c) follows from a), as well, but the details of the purely algebraic computations are omitted here. $\square$)

Example 1.6: Triangle matrix of $s(n,k) \quad (n = 1,2,\ldots \quad k = 1,2,\ldots)$

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & -3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-6 & 11 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
24 & -50 & 35 & -10 & 1 & 0 & 0 & 0 & 0 & 0 \\
-120 & 274 & -225 & 85 & -15 & 1 & 0 & 0 & 0 & 0 \\
720 & -1764 & 1624 & -735 & 175 & -21 & 1 & 0 & 0 & 0 \\
-5040 & 13068 & -13132 & 6769 & -1960 & 322 & -28 & 1 & 0 & 0 \\
40320 & -109584 & 118124 & -67284 & 22449 & -4536 & 546 & -36 & 1 & 0 \\
-362880 & 1026576 & -1172700 & 723680 & -269325 & 63273 & -9450 & 870 & -45 & 1
\end{pmatrix}$$

The matrix represents the Stirling numbers of the 1st kind: $s(n,k)$ with indices running from 1 to 10 $(n = 1,2,\ldots,10 \quad k = 1,2,\ldots,10)$.

It is denoted by $S_1(n)$ and called the Stirling matrix of the 1st kind of order n. The row number corresponds to n, the column number to k and thus it is associated to the power x^k in the monomial expansion (1.14).

The recursion (1.16) is illustrated by the blue and red coloured entries: For n = 3 (row number) and k = 3 (column number) we have: $-6 = s(n+1,k) = s(n,k-1) - n \cdot s(n,k) = -3 - 3 \cdot 1 = -6$.

The recursion allows to expand the matrix row by row. The first column are signed faculty numbers (0! , -1! , 2! , -3! , 4! ,...., $(-1)^n\, n!$). The diagonal consists of 1s and the matrix is triangular with 0s right above the diagonal.

By applying Property 1a) to all rows (n = 1, 2, ...) the connection between the Newton coefficients and the monomial coefficients in (1.14) reveals as a simple matrix operation involving the Stirling matrix of the 1st kind:

Property 1.4: Matrix transformation from $(a_0, a_1, \ldots, a_n)$ to $(c_0, c_1, \ldots, c_n)$

a) $c_0 = a_0$

b) $(c_1, \ldots, c_n) = (a_1, \ldots, a_n) \cdot S_1(n)$

Property 1.4 refers to the normalized equi-distant measurement arguments {0, 1, 2, ..., n}. The general situation of equi-distant arguments is solved by a simple reduction to the normalized distribution. The next example is to reveal the procedure.

Example 1.8: Equi-distant arguments and reduction to normalized arguments {0, 1, 2, ..., n}

The x-arguments are {1, 3, 5, 7} and the measurement values y are {-1, 2, 4, -1}. The Newton-Tableau on the right (the col-

-1			
2	$\frac{3}{2}$		
4	1	$-\frac{1}{8}$	
-1	$-\frac{5}{2}$	$-\frac{7}{8}$	$-\frac{1}{8}$

umn of x arguments is omitted) results in the cubic polynomial $p(x) = -1 + 3/2 \cdot (x-1) - 1/8 \cdot (x-1)(x-3) - 1/8 \cdot (x-1)(x-3)(x-5)$ and the Newton coefficients $\left(a_0, a_1, a_2, a_3\right) = (-1, 3/2, -1/8, -1/8)$.

In order to be connected to Property 1.3 and 1.4 the x-axis is rescaled or normalized by

$$\tilde{x} = \frac{x - x_0}{h} = \frac{x-1}{2} \in [0,3] \ .$$

In these normalized coordinates the Newton tableau on the right yields the cubic polynomial

-1			
2	3		
4	2	$-\frac{1}{2}$	
-1	-5	$-\frac{7}{2}$	-1

$$p(\tilde{x}) = -1 + 3 \cdot (\tilde{x}) + 3 \cdot (\tilde{x})(\tilde{x}-1) - 1/2 \cdot \tilde{x}(\tilde{x}-1)(\tilde{x}-2)$$

and the Newton coefficients $\left(\tilde{a}_0, \tilde{a}_1, \tilde{a}_2, \tilde{a}_3\right) = (-1, 3, -1/2, -1)$.

From Property 1.4 it is concluded that $c_0 = \tilde{a}_0 = -1$ and

$$\left(c_1, c_2, c_3\right) = \left(\tilde{a}_1, \tilde{a}_2, \tilde{a}_3\right) \cdot S_1(3) = (3, -1/2, -1) \cdot \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -3 & 1 \end{pmatrix} = \left(3/2, 5/2, -1\right).$$

Thus

$$p(\tilde{x}) = -1 + 3/2 \cdot \tilde{x} + 5/2 \cdot \tilde{x}^2 - 1 \cdot \tilde{x}^3 = -1 + 3/2 \cdot \left(\frac{x-1}{2}\right) + 5/2 \cdot \left(\frac{x-1}{2}\right)^2 - 1 \cdot \left(\frac{x-1}{2}\right)^3 = p(x).$$

This is the same as $-1 + 3/4 \cdot \left(x-1\right) + 5/8 \cdot \left(x-1\right)^2 - 1/8 \cdot \left(x-1\right)^3 = p(x)$ yielding a monomial representation of the interpolation polynomial p.

Theorem 1.4: Matrix transformation from $\left(a_0, a_1, \ldots, a_n\right)$ to $\left(c_0, c_1, \ldots, c_n\right)$

Assuming equi-distant arguments $x_k = x_0 + k \cdot h \quad (k = 0, \ldots, n)$ with $h = \vartriangle x = const$ and assuming that the normalization $\tilde{x} = \dfrac{x - x_0}{h} \in [0, n]$ is interpolated by $p(\tilde{x}) = \tilde{a}_0 + \tilde{a}_1 \cdot \tilde{x} + \tilde{a}_2 \cdot \tilde{x}(\tilde{x}-1) + \cdots + \tilde{a}_n \cdot \tilde{x}(\tilde{x}-1) \cdot \ldots \cdot (\tilde{x} - n + 1)$ then the monomial representation of the interpolating polynomial $p(x)$ is equal to

$$c_0 + c_1 \cdot \left(\frac{x - x_0}{h}\right) + c_2 \cdot \left(\frac{x - x_0}{h}\right)^2 + \cdots + c_n \cdot \left(\frac{x - x_0}{h}\right)^n =$$

$$\boxed{c_0 + \frac{c_1}{h} \cdot \left(x - x_0\right) + \frac{c_2}{h^2} \cdot \left(x - x_0\right)^2 + \cdots + \frac{c_n}{h^n} \cdot \left(x - x_0\right)^n}$$

whereas $c_0 = \tilde{a}_0 = a_0$ and $\boxed{\left(c_1, \ldots, c_n\right) = \left(\tilde{a}_1, \ldots, \tilde{a}_n\right) \cdot S_1(n) = \left(a_1 h, \ldots, a_n h^n\right) \cdot S_1(n)}$.

Notice that by Theorem 1.3: $\tilde{a}_k = \dfrac{\Delta^k y_0}{k!}$ and $a_k = \dfrac{\Delta^k y_0}{h^k k!} = \dfrac{\tilde{a}_k}{h^k}$. This implies that $a_k = \dfrac{\tilde{a}_k}{h^k}$.