NUMERICAL METHODS II.

Csaba J. Hegedüs

ELTE, Faculty of Informatics

Budapest, 2016 January

"Financed from the financial support ELTE won from the Higher Education Restructuring Fund of the Hungarian Government"

Referee: Dr. Levente Lócsy, ELTE

Contents

11. Lagrange interpolation and its error

Interpolation is a simple way of approximating functions by demanding that the interpolant function assumes the values of the approximated function at specified places. Collect the support points of the interpolation into the set $\Omega_n = \{x_0, x_1, \ldots, x_n\}$, where x_i-s are not necessarily ordered. We shall consider interpolation in the interval $[a,b]$. The relation $[a,b] = [\min_i x_i, \max_i x_i]$ holds in many cases, but all support points may also be inner points of $[a,b]$.

11.1. Interpolating function with linear parameters

Let n be a natural number, $n \in \mathbb{N}$ and assume the values of the function $f(x)$ are known at the points $x_k \in \mathbb{R}$, $k = 0,1,...,n$. In the case of a linear interpolation problem, we choose the interpolant as

$$\Phi(x) = \sum_{i=0}^{n} a_i \varphi_i(x), \tag{11.1}$$

where $\varphi_i(x)$-s are base functions, and the unknown coefficients a_i are determined from the conditions

$$f(x_i) = \Phi(x_i), \quad i = 0,...,n. \tag{11.2}$$

The functions $\varphi_i(x)$ in (11.1) may be powers of x: $\varphi_i(x) = x^i$, which will lead to a polynomial, but it is also possible to choose other functions: $\varphi_i(x) = \sin(i\omega x)$, $\varphi_i(x) = \cos(i\omega x)$, $\varphi_i(x) = \exp(i\omega x)$. In the case of $n = 2$ the interpolation problem (11.2) leads to the linear system of equations

$$\begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \varphi_2(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \varphi_2(x_1) \\ \varphi_0(x_2) & \varphi_1(x_2) & \varphi_2(x_2) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \end{pmatrix}. \tag{11.3}$$

The obtained system can be solved uniquely if the coefficient matrix has an inverse.

11.2. Polynomial interpolation

This time the choice $\varphi_i(x) = x^i$ leads to the transpose of a Vandermonde matrix in (11.2)

$$V^T(\Omega_n) = \begin{pmatrix} 1 & x_0 & \ldots & x_0^n \\ 1 & x_1 & \ldots & x_1^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \ldots & x_n^n \end{pmatrix}, \tag{11.4}$$

which is nonsingular if the support points x_i are different. It follows that the polynomial interpolation problem is solvable uniquely if no two support abscissas are equal.

11.3. Interpolation with Lagrange base polynomials

Associate the following polynomial with the support abscissas in Ω_n:

$$\omega_n(x) = \prod_{j=0}^{n} (x - x_j). \tag{11.5}$$

Observe it is of degree $n+1$ and it vanishes at the support points. Further, introduce the ith Lagrange base polynomial of order n, which vanishes at all support points with the exception of x_i, where it takes 1:

$$l_i(x) = \frac{\omega_n(x)}{(x - x_i)\prod_{j=0, j\neq i}^{n} (x_i - x_j)} = \frac{\omega_n(x)}{(x - x_i)\omega_n'(x_i)} = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}. \tag{11.6}$$

Here division by $(x - x_i)$ was done for the sake of cancelling it from the numerator and the product in the denominator ensures $l_i(x_i) = 1$. Now choosing $\varphi_i(x) = l_i(x)$ in (11.1) leads to the identity as the coefficient matrix in the linear system because of $l_i(x_j) = \delta_{ij}$, where δ_{ij} denotes the Kronecker delta. Now we have the simple result

$$a_i = f(x_i), \tag{11.7}$$

and the Lagrange interpolating polynomial is:

$$L_n(x) = \sum_{i=0}^{n} f(x_i) l_i(x). \tag{11.8}$$

By using the properties of the base polynomials, one can easily check the relation $L_n(x_i) = f(x_i)$.

11.3.1 Theorem on the interpolation error

Let function $f(x)$ be at least $(n+1)$-times differentiable in $[a,b]$: $f(x) \in C^{n+1}[a,b]$, where the support abscissas are in $[a,b]$. Then for $\forall x \in [a,b]$ there exists $\xi_x \in [a,b]$, such that

$$f(x) - L_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \omega_n(x) \tag{11.9}$$

holds, moreover

$$\left| f(x) - L_n(x) \right| \leq \frac{M_{n+1}}{(n+1)!} \left| \omega_n(x) \right|, \tag{11.10}$$

where $M_k = \left\| f^{(k)} \right\|_\infty = \max_{x\in[a,b]} \left| f^{(k)}(x) \right|$.

Proof. If $x \in \Omega_n$, then both sides of (11.9) are zero and equality holds. Assume in the following that $x \notin \Omega_n$ and introduce function

$$g_x(z) = f(z) - L_n(z) - \frac{\omega_n(z)}{\omega_n(x)} (f(x) - L_n(x)), \quad z \in [a,b]. \tag{11.11}$$

We have $g_x(z) \in C^{n+1}[a,b]$, $g_x(z)=0$, $z \in \Omega_n$, and there is an extra zero at $z=x$ such that there are altogether $n+2$ zero points. Using Rolle's theorem between zero points, we find that $dg_x(z)/dz$ has $n+1$ zeros, $d^2 g_x(z)/dz^2$ has n zeros and after differentiating consecutively $(n+1)$-times, we shall have only one zero of $g_x^{(n+1)}(z)$ at some place $\xi_x \in [a,b]$:

$$g_x^{(n+1)}(\xi_x) = 0 = f^{(n+1)}(\xi_x) - 0 - \frac{(n+1)!}{\omega_n(x)}(f(x) - L_n(x))$$

that can be rearranged into (11.9). For the second statement, take the absolute value of both sides and find an upper bound for the $(n+1)$ th derivative in the interval $[a,b]$. ■

Remark. Rolle's theorem states that if $f(a) = f(b) = 0$ holds and f is differentiable in $[a,b]$ where there exist a point in $[a,b]$, where f' is zero. This theorem is a simple consequence of the Lagrange mean-value theorem, and it is also true if $f(a) = f(b)$ holds.

By finding the minimum of the absolute value of the $(n+1)$ th derivative, a lower bound can also be found as in (1.10).

11.4. Example

Assume the values y_i for the points $\Omega_n = \{x_0, x_1, \ldots, x_n\}$ come from an n th degree polynomial $p_n(x)$. Show that $L_n(x)$ to these support points is the same polynomial: $L_n(x) = p_n(x)$.

Solution. Consider the error of interpolation:

$$p_n(x) - L_n(x) = \frac{p_n^{(n+1)}(\xi_x)}{(n+1)!}\omega_n(x) = 0,$$

where the $(n+1)$-st derivative of an n th order polynomial is identically zero.

11.5. The barycentric Lagrange interpolation

Despite the simplicity and elegance of Lagrange interpolation, it is a common belief that certain shortcomings make it a bad choice for practical computations. Among the shortcomings sometimes claimed are these, Berrut, Trefethen (2004):

1. Each evaluation of $L_n(x)$ requires $O(n^2)$ additions and multiplications.

2. Adding a new data pair (x_{n+1}, f_{n+1}) requires a new computation from scratch.

3. The computation is numerically unstable.

From here it is commonly concluded that the Lagrange interpolation form of $L_n(x)$ is mainly a theoretical tool for proving theorems and for computations one should instead use Newton's formula that will be given later in this text. In fact, it needs $O(n)$ flops for each evaluation of $L_n(x)$ once some numbers, which are independent of the evaluation point x, have been computed.

11.5.1 The first improved Lagrange formula

We shall see that the Lagrange formula (11.8) can be rewritten in such a way that it too can be evaluated and updated in $O(n)$ operations, just like its Newton counterpart. Introduce notation

$$\ell(x) = \omega_n(x) = \prod_{j=0}^{n}(x - x_j).$$

If we define the barycentric weights by

$$w_j = \frac{1}{\prod_{k \neq j}(x_j - x_k)} = \frac{1}{\ell'(x_j)}, \quad j = 0,1,\ldots,n \tag{11.12}$$

then we can write (11.8) in the form:

$$L_n(x) = \ell(x)\sum_{j=0}^{n}\frac{w_j}{(x - x_j)}f_j\,. \tag{11.13}$$

Now Lagrange interpolation is a fomula requiring $O(n^2)$ operations for calculating some quantities independent of x, the numbers w_j, followed by $O(n)$ flops for evaluating $L_n(x)$ once these numbers are known. Rutishauser (1976) called (11.13) the "first form of the barycentric interpolation formula". A remarkable feature of this form is that for compution of the weights w_j, no function data are required.

Now incorporating a new node x_{n+1} entails two calculations:

- Divide each $w_j,\ j = 0,\ldots,n,$ by $x_j - x_{n+1}$ for a cost of $2n+2$ flops.
- Compute w_{n+1} with formula (11.12), for another $n+1$ flops.

11.5.2 The barycentric formula

Equation (11.13) can be modified to an even more elegant formula, the one that is often used in practice. Suppose we interpolate, besides the data f_j, the constant function 1, whose interpolant is of course itself, see also Problem 11.6. Inserting into (11.13), we get

$$1 = \sum_{j=0}^{n}\ell_j(x) = \ell(x)\sum_{j=0}^{n}\frac{w_j}{(x - x_j)}. \tag{11.14}$$

Dividing (11.13) by this expression and cancelling the common factor $\ell(x)$, we obtain the barycentric formula for $L_n(x)$:

$$L_n(x) = \frac{\sum_{j=0}^{n}\frac{w_j}{(x - x_j)}f_j}{\sum_{j=0}^{n}\frac{w_j}{(x - x_j)}}, \tag{11.15}$$

where w_j is still defined by (11.12). Rutishauser (1976) called (11.15) the "second (true) form of the barycentric formula.".

For the numerical stability of barycentric formulas, see Berrut, Trefethen (2004) and Higham (2004).

11.6. Problems

11.1. Three support points (-1,-1), (1,1), (2,3) belong to a function. Obtain the Lagrange base polynomials and the interpolating polynomial $L_2(x)$ that goes through these points.

11.2. The function $f(x)=(x+1)^{-2}$ is interpolated in [0,1] on the support $\Omega=\{0, 0.2, 0.5, 0.8, 1\}$. Estimate the error $\left|f(x)-L_4(x)\right|$ at $x=0.4$!

11.3. The function $f(x)=\cos^2 x$ is interpolated in the interval [0,2]. The support are $\Omega=(0,0.4,0.7,1.3,2)$. Using the error formula, estimate the error of $f(x)-L_4(x)$ at $x=0.5$!

114. Show that $\sum_{j=0}^{n} x_j^k l_j(x)=x^k$, if $k\le n$.

11.5. Give an explanation for $x^{n+1}-\sum_{j=0}^{n} x_j^{n+1} l_j(x)=\omega_n(x)$.

11.6. Introduce the notation $l_i(x)=\sum_{k=0}^{n} c_{k,i}x^k$. Show that matrix $C=\left[c_{k,i}\right]$ is the inverse of the Vandermonde matrix $V^T(\Omega_n)$ belonging to the same support abscissas.

11.7. The partial fraction expansion $1/\omega_n(x)=\sum_{j=0}^{n} b_j/(x-x_j)$ holds if x_j-s are different. Show that the expansion coefficients b_j-s are in the last row of C of the previous problem: $b_j=c_{n,j}$.

11.8. Let $p(x)/q(x)$ be a proper rational function and let $q(x)$ have the simple roots $x_0, x_1, \ldots, x_n$. Explain that

$$\frac{p(x)}{q(x)}=\sum_{j=0}^{n}\frac{p(x_j)}{(x-x_j)q'(x_j)}.$$

In other words, partial fraction expansion can be done by Lagrange interpolation.

11.9. Explain that having done the partial fraction expansion for $1/\omega_n(x)=1/\ell(x)$, we have enough data to apply the barycentric interpolation formula.

12. Some properties of polynomial interpolation

One can pose the natural question: By increasing the degree of the polynomial, will the quality, i.e. the error of the approximation be better? Will the polynomial converge to the function this time? The answer is not always positive, but there are cases when the answer is 'yes'.

12.1. Theorem on uniform convergence

Assume $f \in C^{\infty}[a,b]$ and let $x_k^{(n)}$, $k = 0,1,\ldots,n$; $n = 0,1,2,\ldots$ be a series of support point sets in $[a,b]$. Denote by $L_n(x)$ the Lagrange interpolating polynomial belonging to the nth set $x_0^{(n)}, x_1^{(n)}, \ldots, x_n^{(n)}$, $n = 0,1,2,\ldots$ If $\exists M > 0$ such that $M_n = \left\| f^{(n)} \right\|_{\infty} = \max_{x \in [a,b]} \left| f^{(n)}(x) \right| \le M^n \ \forall n$, then the sequence of $L_n(x)$ polynomials converges uniformly to $f(x)$.

Proof. Apply the maximum norm for the interval $[a,b]$ and find an upper bound for $\|\omega_n\|_{\infty}$:

$$\left| f(x) - L_n(x) \right| \le \left\| f - L_n \right\|_{\infty} \le \frac{M_{n+1}}{(n+1)!} \left\| \omega_n \right\|_{\infty} \le \frac{M^{n+1}(b-a)^{n+1}}{(n+1)!} = \frac{[M(b-a)]^{n+1}}{(n+1)!}.$$

The right hand side here tends to zero with $n \to \infty$ because the factorial in the denominator tends to infinity faster than the power in the numerator, hence uniform convergence follows,

$$\left\| f - L_n \right\|_{\infty} \to 0,$$

that is the maximal absolute difference of the two functions tends to zero. ■

12.2. Lemma on upper bound for $\left|\omega_n(x)\right|$

Let the support points be ordered: $x_{k-1} < x_k$ and introduce the notation $h = \max_{k=1,2,\ldots,n} \left| x_k - x_{k-1} \right|$. Then we have the estimate for $\left|\omega_n(x)\right|$:

$$\left| \omega_n(x) \right| \le \frac{n!}{4} h^{n+1}, \quad x \in [a,b]. \tag{12.1}$$

Proof. We investigate each interval. First let $x \in [x_0, x_1]$. Substituting the value of x at the maximum point, we get the estimate: $\left|(x - x_0)(x - x_1)\right| \le h^2 / 4$. The other multipliers can be bounded by $2h, 3h, \ldots$ such that

$$\left| \omega_n(x) \right| \le (h^2/4)(2h)(3h)\ldots(nh) = \frac{h^{n+1} n!}{4}, \quad x \in [x_0, x_1].$$

Next assume $x \in [x_1, x_2]$. We get similarly:

$$\left| \omega_n(x) \right| \le (2h)(h^2/4)(2h)\ldots((n-1)h) < \frac{h^{n+1} n!}{4}.$$

We get smaller bounds for the other inner intervals as compared to that of the first one. Finally, the last interval yields the same bound as the first one such that (12.1) holds for all $x \in [a,b]$. ■

Remark. The estimate here suggests that if x is in the middle of the interval $[a,b]$, the error is less than it is at closer locations to the ends of $[a,b]$. This proves to be true if the support abscissas are located uniformly. One may guess that the approximation will be better if the support points are more densely located at the ends of the interval.

With the help of 12.2 Lemma, it is possible to give an overall error bound.

12.3. Another theorem on error bound

Let the support points be ordered: $x_{k-1} < x_k$, where $h = \max\limits_{k=1,2,\ldots,n} |x_k - x_{k-1}|$. Then one has the bound

$$|f(x) - L_n(x)| \le \frac{M_{n+1}}{4(n+1)} h^{n+1}, \quad x \in [a,b]. \tag{12.2}$$

Proof. Substituting (12.1) into (11.10) of the error theorem gives the result.

■

12.4. Skilful choice of support abscissas, Chebyshev polynomials

The Chebyshev polynomial of n th order is given by

$$T_n(x) = \cos(n \arccos x), \quad x \in [-1,1], \quad n = 0,1,\ldots \tag{12.3}$$

We show it is a polynomial. Introduce notation $\vartheta = \arccos x$, then

$$\begin{aligned} T_{n\pm1}(x) &= \cos((n\pm1)\vartheta) = \cos(n\vartheta \pm \vartheta) = \\ &= \cos(n\vartheta)\cos\vartheta \mp \sin(n\vartheta)\sin(\vartheta) = xT_n(x) \mp \sin(n\vartheta)\sin\vartheta. \end{aligned}$$

Applying this relation to $T_{n+1}(x) + T_{n-1}(x)$ yields the formula:

$$T_{n+1}(x) + T_{n-1}(x) = 2xT_n(x), \tag{12.4}$$

which leads to the recursion of Chebyshev polynomials:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad x \in [-1,1]. \tag{12.5}$$

Checking some of the first polynomials, we find that $T_n(x) = 2^{n-1}x^n + \ldots$, $n > 0$ so that we can introduce monic Chebyshev polynomials by the relation

$$t_n(x) = 2^{1-n}T_n(x), \quad 0 < n.$$

Now we have the following

12.5. Theorem on the best zero approximating monic polynomial in ∞-norm

Denote by $\mathcal{P}_n^1[-1,1]$ the set of monic polynomials of n th order in $[-1,1]$, then

$$\|t_n\|_\infty \le \|p\|_\infty, \quad p \in \mathcal{P}_n^1[-1,1].$$

In words: it is the monic Chebyshev polynomial that has the smallest maximal absolute value in $[-1,1]$, that is, it gives the best approximating polyniomial in ∞-norm to the zero function in $[-1,1]$.

Proof. The polynomials $T_n(x)$ are oscillating between -1 and 1 due to the nature of *cosine* functions. The extrema are at z_k-s:

$$\cos(n \arccos z_k) = (-1)^k, \quad \text{from where} \quad z_k = \cos\frac{k\pi}{n}, \quad k = 0,1,\ldots n. \tag{12.6}$$

There are altogether $n+1$ different z_k-s. The monic polynomial t_n has extrema at the same places. Now assume indirectly that $\exists p \in \mathcal{P}_n^1[-1,1]$, for which $\|p\|_\infty < \|t_n\|_\infty$. The difference polynomial $r = t_n - p$ is of order at most $n-1$. Observe that $r(z_k)$ is positive if $t_n(z_k)$ is positive, otherwise $p(x)$ is not better than t_n. Similarly, $r(z_k)$ is negative if $t_n(z_k)$ is negative such that $r(z_k)$ should change sign at least n-times between the $n+1$ extremal points. However, this is contradiction, because $r(x)$ should be then at least of order n. ■

The roots of Chebyshev polynomials. $\cos(n \arccos x_k) = 0 \to n \arccos x_k = \frac{\pi}{2} + k\pi \to$

$$x_k = \cos\frac{(2k+1)\pi}{2n}, \quad k = 0,1,\ldots,n-1 \text{ different locations.} \tag{12.7}$$

Corollary. Choose the roots of t_{n+1} for support points of the interpolation in $[-1,1]$. Then $t_{n+1} = \omega_n(x)$ holds and we get the smallest uniform error bound:

$$|f(x) - L_n(x)| \le \frac{M_{n+1}}{(n+1)!}\|\omega_n\|_\infty = \frac{M_{n+1}}{(n+1)!}\|t_{n+1}\|_\infty = \frac{M_{n+1}}{(n+1)!}\frac{1}{2^n}. \tag{12.8}$$

If $x \in [a,b]$, then let $t \in [-1,1]$ and introduce the linear transform

$$x = \frac{a+b}{2} + t\frac{b-a}{2} \tag{12.9}$$

that maps $[-1,1]$ onto $[a,b]$. This relates the Chebyshev abscissas t_i and x_i and one can write $x - x_i$ in terms of $t - t_i$ as $x - x_i = (t - t_i)(b-a)/2$ such that $\omega_n(x) = c^{n+1} t_{n+1}(t)$, $c = (b-a)/2$. Now it is straightforward to give an upper bound as in (12.8) for Chebyshev abscissas in $[a,b]$:

$$|f(x) - L_n(x)| \le \frac{M_{n+1}}{(n+1)!}\|\omega_n\|_\infty = \frac{M_{n+1}c^{n+1}}{(n+1)!}\|t_{n+1}\|_\infty = \frac{M_{n+1}}{(n+1)!}\frac{c^{n+1}}{2^n}. \tag{12.10}$$

12.6. Problems

12.1. Decide if the interpolating polynomials tend uniformly to the functions below, when the number of support points in $[a,b]$ tends to infinity, $n \to \infty$:

a) $f(x)=\sin x,\quad x\in[0,\pi]$ *b*) $f(x)=\cos x,\quad x\in[0,\pi]$

c) $f(x)=e^x,\quad x\in[0,1]$ *d*) $f(x)=(x+2)^{-1},\quad x\in[0,1]$

e) $f(x)=(x+2)^{-1},\quad x\in[-1,1]$

12.2. What happens in case *e*) of the previous example if we choose Chebyshev abscissas, i.e. the roots of t_{n+1}?

12.3. The $\sin x$ function is tabulated in the interval $[0,\pi/2]$. How dense uniform tabulation is needed in order to reach an error bound 10^{-4} for the function when using linear interpolation? Try to estimate the subinterval length h.

12.4. What is the result in the previous problem, if the interpolating polynomial is of second order? Determine the maximal absolute value of $\omega_2(x)$!

12.5. Show that the estimate in (12.1) can be modified to $|\omega_n(x)|\le n!(K_n(b-a))^{n+1}/(4n^{n+1})$, where h is defined in Lemma 12.2 and $1\le K_n=hn/(b-a)$. When will be $K_n=1$?

12.6. According to the Stirling formula $n!\approx\left(\frac{n}{e}\right)^n\sqrt{2\pi n}\left(1+\frac{1}{12n}+\frac{1}{288n^2}-\ldots\right)$. Show with the help of the previous problem that using equidistant support abscissas leads to $\lim_{n\to\infty}\omega_n(x)\to 0$ if $|b-a|\le e$.

12.7. Define the scalar product of two polynomials by $(T_i,T_j)=\int_{-1}^{1}\alpha(x)T_i(x)T_j(x)dx$, where $\alpha(x)=(1-x^2)^{-1/2}$ is the weight function. Show that different Chebyshev polynomials are orthogonal to each other with respect to this scalar product.

13. Iterated interpolations (Neville, Aitken, Newton)

13.1. Neville and Aitken interpolations

An inconvenience may be with Lagrange interpolation is that introducing a newer abscissa results in a need to recalculate the base polynomials. There are also cases when it is not the interpolation polynomial but its value at some places is requested. For such cases iterated interpolation may be more favourable.

Let the support points be $\{(x_i, f_i = f(x_i)\}_{i=0}^{n}$, and denote by $p_{0,1,\ldots,k}(x)$ the polynomial of k th order, which interpolates for abscissas with the same index

$$p_{0,1,\ldots,k}(x_j) = f(x_j),\ \ j = 0,1,\ldots,k\,. \tag{13.1}$$

It will be shown that such polynomials can be build up recursively. Consider the following determinant:

$$p_{0,1,\ldots,k,k+1}(x) = \frac{1}{x_{k+1} - x_0}\begin{vmatrix} x - x_0 & p_{0,1,\ldots,k}(x) \\ x - x_{k+1} & p_{1,\ldots,k+1}(x) \end{vmatrix}. \tag{13.2}$$

It is straightforward to check that the new polynomial interpolates well at points $x = x_0$ and $x = x_{k+1}$. Otherwise, for the intermediate points $0 < j < k+1$, one has

$$p_{0,1,\ldots,k,k+1}(x_j) = \frac{1}{x_{k+1} - x_0}\begin{vmatrix} x_j - x_0 & p_{0,1,\ldots,k}(x_j) \\ x_j - x_{k+1} & p_{1,\ldots,k+1}(x_j) \end{vmatrix} = f(x_j)\frac{x_j - x_0 - (x_j - x_{k+1})}{x_{k+1} - x_0} = f(x_j).$$

Due to this recursion the Neville interpolation calculates the following table of numbers:

	$k=0$	1	2	3
$x - x_0$	$f_0 = p_0(x)$			
$x - x_1$	$f_1 = p_1(x)$	$p_{01}(x)$		
$x - x_2$	$f_2 = p_2(x)$	$p_{12}(x)$	$p_{012}(x)$	
$x - x_3$	$f_3 = p_3(x)$	$p_{23}(x)$	$p_{123}(x)$	$p_{0123}(x)$

Observe now that when introducing a new point (x_4, f_4), it suffices only to calculate a next row in the table. If we compute symbolically by keeping x as a parameter, then we get polynomials. It is easier to compute with numbers. In this case we compute the value of the polynomials at x. To calculate the number in the denominator of the recursion, substract the lower number from the upper one in the leftmost column, e.g. $x - x_0 - (x - x_j)$. We also have these numbers in the left column of the 2×2 determinants. As an example, for the support points

x_i	0	1	3
f_i	1	3	2

the values of the table for $x=2$ can be given as

	$k=0$	1	2
$2-0=2$	1		
$2-1=1$	3	$\frac{1}{2-1}\begin{vmatrix}2 & 1\\ 1 & 3\end{vmatrix}=5$	
$2-3=-1$	2	$\frac{1}{1-(-1)}\begin{vmatrix}1 & 3\\ -1 & 2\end{vmatrix}=5/2$	$\frac{1}{2-(-1)}\begin{vmatrix}2 & 5\\ -1 & 5/2\end{vmatrix}=10/3$

Aitken's interpolation is similar, only the intermediate polynomials differ. The arrangement can be demonstrated by the following table:

	$k=0$	1	2	3
$x-x_0$	$f_0=p_0(x)$			
$x-x_1$	$f_1=p_1(x)$	$p_{01}(x)$		
$x-x_2$	$f_2=p_2(x)$	$p_{02}(x)$	$p_{012}(x)$	
$x-x_3$	$f_3=p_3(x)$	$p_{03}(x)$	$p_{013}(x)$	$p_{0123}(x)$

13.2. Divided differences

Here we introduce divided differences for Newton interpolation. Let the support points be $\{(x_i, f_i=f(x_i)\}_{i=0}^{n}$, then the *divided differences of first order* are*:*

$$f[x_0,x_1]=\frac{f(x_1)-f(x_0)}{x_1-x_0},\quad f[x_i,x_{i+1}]=\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}, \tag{13.3}$$

divided differences of second order:

$$f[x_i,x_{i+1},x_{i+2}]=\frac{f[x_{i+1},x_{i+2}]-f[x_i,x_{i+1}]}{x_{i+2}-x_i}. \tag{13.4}$$

In general, the *divided differences of* k th *order* are based on $k+1$ points*:*

$$f[x_i,x_{i+1},\ldots,x_{i+k}]=\frac{f[x_{i+1},x_{i+2}\ldots,x_{i+k}]-f[x_i,x_{i+1},\ldots,x_{i+k-1}]}{x_{i+k}-x_i}. \tag{13.5}$$

We can arrange the following table of divided differences:

	$k=0$	1	2	3
x_0	$f(x_0)$			
x_1	$f(x_1)$	$f[x_0,x_1]$		
x_2	$f(x_2)$	$f[x_1,x_2]$	$f[x_0,x_1,x_2]$	
x_3	$f(x_3)$	$f[x_2,x_3]$	$f[x_1,x_2,x_3]$	$f[x_0,x_1,x_2,x_3]$

Example. Compute the table of divided differences, if the support points are:

x_i	1/2	1	2	3
f_i	2	1	1/2	1/3

The number on top of a column shows the order of divided differences in that column.

	1	2	3
1/2	2		
1	1	$\frac{1-2}{1-1/2}=-2$	
2	1/2	$\frac{1/2-1}{2-1}=-\frac{1}{2}$	$\frac{-1/2-(-2)}{2-1/2}=1$
3	1/3	$\frac{1/3-1/2}{3-2}=-\frac{1}{6}$	$\frac{-1/6-(-1/2)}{3-1}=\frac{1}{6}$ $\quad\frac{1/6-1}{3-1/2}=\frac{-1}{3}$

13.2.1 Lemma on expansion of divided differences

Divided differences can be expanded as:

$$f[x_0,x_1,\ldots,x_k]=\sum_{j=0}^{k}\frac{f(x_j)}{\omega_k'(x_j)}, \tag{13.6}$$

here $\omega_k(x)$ is the product function of (11.5).

Proof. It can be done by induction. It is true for $k=1$. From k to $k+1$, we write into

$$f[x_0,x_1,\ldots,x_{k+1}]=\frac{f[x_1,\ldots,x_{k+1}]-f[x_0,\ldots,x_k]}{x_{k+1}-x_0}$$

the expanded form of the k th order divided differences with the $(k+1)$ th denominator:

$$f[x_1,\ldots,x_{k+1}]=\sum_{j=1}^{k+1}\frac{f(x_j)(x_j-x_0)}{\omega_{k+1}'(x_j)},\quad f[x_0,\ldots,x_k]=\sum_{j=0}^{k}\frac{f(x_j)(x_j-x_{k+1})}{\omega_{k+1}'(x_j)}.$$

By simplifying and ordering the sums, the statement follows. ■

It is seen that the divided differences are symmetric functions of the base points, that is, its value remains unchanged if the order of base points are varied.

13.3. Recursive Newton interpolation

Denote $N_n(x)$ the (Newton) polynomial that interpolates on the support abscissas $x_0, x_1, \ldots, x_n$. These polynomials can also be computed from the recursion

$$N_n(x) = N_{n-1}(x) + b_n \omega_{n-1}(x) = \sum_{j=0}^{n} b_j \omega_j(x), \quad \omega_{-1}(x) = 1, \tag{13.7}$$

where coefficient b_n may be determined from the condition that $N_n(x)$ interpolates also at x_n:

$$b_n = \frac{f_n - N_{n-1}(x_n)}{\omega_{n-1}(x_n)}.$$

However, there exists another method for the computation of b_n by divided differences.

13.3.1 Theorem on expansion coefficients in Newton's interpolation

$$b_n = f[x_0, x_1, \ldots, x_n]. \tag{13.8}$$

Proof. By definition, $N_n(x) - N_{n-1}(x)$ vanishes at the points $x_0, \ldots, x_{n-1}$, hence we may write $\omega_{n-1}(x)$ in (13.7), and its multiplier comes from the condition that $N_n(x_n) = f(x_n)$ holds. Now use the corresponding Lagrange polynomial in place of $N_{n-1}(x)$:

$$b_n = \frac{f(x_n)}{\omega_{n-1}(x_n)} - \frac{N_{n-1}(x_n)}{\omega_{n-1}(x_n)} = \frac{f(x_n)}{\omega_{n-1}(x_n)} - \sum_{j=0}^{n-1} \frac{f(x_j)\omega_{n-1}(x_n)}{\omega_{n-1}(x_n)(x_n - x_j)\omega'_{n-1}(x_j)} =$$

$$= \frac{f(x_n)}{\omega_{n-1}(x_n)} + \sum_{j=0}^{n-1} \frac{f(x_j)}{(x_j - x_n)\omega'_{n-1}(x_j)} = \sum_{j=0}^{n} \frac{f(x_j)}{\omega'_n(x_j)} = f[x_0, x_1, \ldots, x_n],$$

where we have used (13.6) in the last line. Consequently, the coefficients b_n can be found in the table of divided differences as the last elements of each row. ■

We simply speak about Newton's interpolation, if the b_n coefficients are taken from divided differences. The use of the recursive Newton interpolation may be advantageous in unusual situations; for instance, if we want to get a multidimensional interpolation formula or we interpolate higher derivatives such that certain derivatives of lower order are missing.

13.4. Problems

13.1. Check the proof of Lemma 13.2.1 in detail!

13.2. We want to approximate a function at x with Neville's interpolation. In each row of the table the last number gives the value of an interpolating polynomial. What should be the sequence of support points for a better precision of the last elements?

13.3. Show that the result of the Neville interpolation scheme is unchanged if we write $x_i - x$ in the first column instead of $x - x_i$!

13.4. Determine the second degree polynomial with Neville's interpolation that passes the points (-1,0),(1,1), (2,6) ! (Now x stays in the formulas as a parameter.)

13.5. Compute the same polynomial by Newton's interpolation!

13.6. Find the Newton interpolating polynomial for the points (0,1), (1,3), (-3,2) and give it in Newton interpolation style! What are the three Lagrange base polynomials in this case?

13.7. Having the divided differences at hand, suggest an algorithm that computes substitution values for the Newton form of interpolation!

13.8. Write a Matlab program for computing the Neville interpolation values in a vector coming from polynomials of increasing degree!

13.9. Write a Matlab program for the table of divided differences!

13.10. Identify the base functions of Newton's interpolation and set up the linear system of interpolation conditions!

13.11. Solve the system of linear equations of the former problem such that the steps of divided differences are applied! Compare the result with the table of divided differences!

13.12. Give the matrix that brings vector $[f_0, f_1, \ldots, f_n]^T$ of function values into the vector of first divided differences $[f_0, f[x_0, x_1], \ldots, f[x_{n-1}, x_n]]^T$!

13.13. Show that the error of Lagrange interpolation can be given by the relation:

$$\text{For } \forall x \;\; \exists \xi_0, \xi_1 \in [a,b]: \;\; f(x) - L_n(x) = \frac{f^{(n)}[\xi_0, \xi_1]}{(n+1)!} \omega_n(x)$$

if $f(x) \in C^n[a,b]$, i.e. this time it is not differentiable $(n+1)$-times.

14. Newton and Hermite interpolations

14.1. Theorem, interpolation error with divided differences

Let $x \in [a,b]$, $x \neq x_i$, $i = 0,1,\ldots,n$, then

$$f(x) - L_n(x) = f[x, x_0, x_1, \ldots, x_n]\,\omega_n(x), \tag{13.9}$$

holds, where the support abscissas are in $[a,b]$.

Proof. Let N_{n+1} be such that it assumes the value of $f(x)$ at point x. Applying the fact that $N_n(x) = L_n(x)$, we may write according to the Newton interpolation:

$$f(x) - L_n(x) = N_{n+1}(x) - N_n(x) = f[x, x_0, x_1, \ldots, x_n]\,\omega_n(x)$$

and this shows the statement. ■

14.1.1 Corollary, connection between divided difference and derivative

Let $f(x) \in C^{n+1}[a,b]$, $x \in [a,b]$, $x \neq x_i$, then there exists $\xi_x \in [a,b]$, such that

$$f[x, x_0, x_1, \ldots, x_n] = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \tag{13.10}$$

holds.

To prove this, it is enough to compare formula (11.9) of Theorem 11.3.1 with (13.9). Specifically for $n = 0$ one has $f[x, x_0] = f'(\xi_x)$, and this is nothing else than the Lagrange mean value theorem. Hence (13.10) is the generalization of the mean value theorem for higher divided differences. Observe that x is a formal variable in (13.10), there $n+2$ base points belong to the divided difference of order $n+1$, and derivative is also of order $n+1$.

14.1.2 Another corollary

Formula (13.10) helps us to extend the table of divided differences for the case when a support point occurs more than once. The support points x_i and ξ_x are in the interval $[a,b]$. Now if $[a,b]$ is shrinked to the point x_0, then we get formally in the limiting case

$$f[\underbrace{x_0, x_0, \ldots, x_0}_{n+1 \text{ base points}}] = \frac{f^{(n)}(x_0)}{n!}. \tag{13.11}$$

14.2. Hermite's interpolation

The derivatives of the function are also interpolated in case of the Hermite interpolation. Then we also have function derivatives among support point data:

$$(x_k, f^{(i)}(x_k)),\ i = 0,1,\ldots,m_k - 1),\ m_k \in \mathbb{N}_+ .$$

For instance, if $m_k = 2$, then the zeroth and first derivatives in x_k are returned by the polynomial. The total number of interpolation conditions in general is:

$$\sum_{k=0}^{n} m_k = m+1, \tag{13.12}$$

such that the interpolating polynomial may have degree m: $H_m(x) \in \mathcal{P}_m$. The interpolation conditions are:

$$H_m^{(i)}(x_k) = f^{(i)}(x_k), \quad i = 0,1,\ldots,m_k - 1; \quad k = 0,1,\ldots,n, \tag{13.13}$$

where different support abscissas are assumed.

14.2.1 Theorem on the existence of Hermite's interpolation

The interpolation polynomial $H_m(x)$ satisfying conditions (13.13) uniquely exists.

Proof. Assume $H_m(x) = \sum_{j=0}^{m} a_j x^j$, then the linear system for the expansion coefficients has the form:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^m \\ 0 & 1 & 2x_0 & \ldots & m x_0^{m-1} \\ \vdots & \ldots & \ldots & \ldots & \vdots \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f'(x_0) \\ \vdots \end{bmatrix}$$

which is an $(m+1)\times(m+1)$ system. It has a unique solution if the determinant is nonzero, $\det(A) \neq 0$. Assume indirectly $\det(A) = 0$. Then it follows that the homogenous system (zero right hand vector) has nonzero solution, which refers to a polynomial of degree at most m. Observe that zero right hand vector means that x_k is a root of multiplicity m_k. However, the resulting polynomial should have $m+1$ roots and this contradicts to a polynomial of degree m. Therefore the matrix of the linear system is invertible and the solution is unique. ■

Remark. If some intermediate derivatives are missing, then we speak about the *lacunary* Hermite interpolation. It is not always solvable.

14.2.2 Error theorem on the Hermite interpolatiion

Let $f(x) \in C^{m+1}[a,b]$, $x \in [a,b]$. Then there exists $\xi_x \in [a,b]$, such that

$$f(x) - H_m(x) = \frac{f^{(m+1)}(\xi_x)}{(m+1)!}\omega_m(x) \tag{13.14}$$

holds and $\omega_m(x) = (x-x_0)^{m_0}(x-x_1)^{m_1}\ldots(x-x_n)^{m_n}$.

Proof. It is similar to that of Theorem 11.3.1. For $x = x_k$, $k = 0,1,\ldots,n$ the statement is true. Therefore assume $x \neq x_k$ for all k. Introduce function

$$g_x(z) = f(z) - H_m(z) - \frac{\omega_m(z)}{\omega_m(x)}(f(x) - H_m(x)), \quad z \in [a,b] \tag{13.15}$$

that has $m+2$ roots together with $z = x$. Applying Rolle's theorem $m+1$ times, as in the case of the Lagrangian interpolation, yields the result. ■

In the special case of having the function value and first derivative at all points, we talk about the *Hermite-Fejér interpolation*.

It is worth saying some words about how the divided difference scheme of Newton's interpolation can be carried over to the Hermite interpolation. The interpretation of divided differences for multiple base points were given in (13.11). Accordingly, we have to give point x_0 twice if $f(x_0)$ and $f'(x_0)$ are given. All previously considered base points x_j gives a multiplier $x-x_j$ into the product function $\omega(x)$, and the most recently included point gives a multiplier only in the next step. The order of points is arbitrary, but multiple points should be in one group because of the derivatives. Do not forget, intermediate derivatives may not be missing. For instance, if the second derivative is given, the first one may not be missing.

14.2.3 Example

Find the Hermite-Fejér interpolating polynomial for two points with the scheme of the Newton interpolation. Let the support abscissas be $x_0,\ x_1$.

Solution. The table of divided differences:

	$k=0$	1	2	3
x_0	f_0			
x_0	f_0	f_0'		
x_1	f_1	$f[x_0,x_1]$	$(f[x_0,x_1]-f_0')/(x_1-x_0)$	
x_1	f_1	f_1'	$(f_1'-f[x_0,x_1])/(x_1-x_0)$	$(f_1'-2f[x_0,x_1]+f_0')/(x_1-x_0)^2$

The interpolating polynomial is

$$H_3(x)=f_0+f_0'(x-x_0)+\frac{f[x_0,x_1]-f_0'}{x_1-x_0}(x-x_0)^2+\frac{f_1'-2f[x_0,x_1]+f_0'}{(x_1-x_0)^2}(x-x_0)^2(x-x_1).$$

14.3. Base polynomials for Hermite's interpolation

If intermediate derivatives are not missing, then Hermite base polynomials can always be given just like in the case of the Lagrange base polynomials. The derivatives $f_k^{(i)}$, $i=0,1,\dots,m_k-1$ are given at the point x_k. Associate with x_k the function

$$h_k(x)=\prod_{j=0,\,j\neq k}^{n}\left(\frac{x-x_j}{x_k-x_j}\right)^{m_j},\quad h_k(x_k)=1.$$

The $0,1,\dots,m_j-1$ derivatives vanish at points $x_j(\neq x_k)$ even if they are multiplied with another polynomial. Hence $h_k(x)$ fulfills the expected conditions at the other points $x_j(\neq x_k)$. We look for the base polynomials belonging to x_k in the form:

$$h_{k,i}(x)=p_{k,i}(x)h_k(x),\quad p_{k,i}(x)\in\mathcal{P}_{m_k-1},$$

where $p_{k,i}(x)$ is a polynomial of degree m_k-1, $i=0,1,\dots,m_k-1$. The coefficients of the ith polynomial can be determined from the conditions $(d/dx)^j h_{k,i}(x_k)=\delta_{ij}$, $j=0,1,\dots,m_k-1$.

For the sake of easier understanding, consider the case when the derivatives at x_k are given up to the second derivative, that is $i = 0,1,2$. An easy form of the polynomials is in powers of $x - x_k$. If $i = 0$, then $p_{k,0}(x) = 1 + \alpha_1 (x - x_k) + \alpha_2 (x - x_k)^2$ should be chosen because $h_k(x_k) = 1$ and $h_{k,0}(x_k) = 1$ is fulfilled with $p_{k,0}(x_k) = 1$. Coefficient α_1 comes from the condition that the first derivative vanishes at x_k:

$$[\alpha_1 + 2\alpha_2 (x - x_k)] h_k(x) + p_{k,0}(x) h_k'(x)\Big|_{x=x_k} = 0,$$

hence $\alpha_1 = -h_k'(x_k)$. The zero second derivative yields α_2:

$$2\alpha_2 h_k(x_k) + 2\alpha_1 h_k'(x_k) + h_k''(x_k) = 0$$

and substituting α_1 results in $\alpha_2 = \left(h_k'(x_k)\right)^2 - h_k''(x_k)/2$.

The constant term of $p_{k,2}(x)$ is zero because of $p_{k,2}(x_k) h_k(x_k) = 0$. Similarly, the first power is also missing as the first derivative is zero: $p_{k,2}'(x_k) h_k(x_k) + p_{k,2}(x_k) h_k'(x_k) = 0$. Finally, from $h_{k,2}''(x_k) = 1$ it follows that $p_{k,2}(x) = (x - x_k)^2 / 2$. To show the form of $p_{k,1}(x) = (x - x_k) + \beta_2 (x - x_k)^2$ is left to the reader.

Now we have the standard form of Hermite's interpolation:

$$H_m(x) = \sum_{k=0}^{n} \sum_{i=0}^{m_k - 1} f_k^{(i)} h_{k,i}(x). \tag{13.16}$$

Observe that this form may also serve as the proof for the uniqueness of non-lacunary Hermite interpolation. If we introduce the polynomial

$$p_k(x) = \sum_{i=0}^{m_k - 1} f_k^{(i)} p_{k,i}(x), \tag{13.17}$$

then we can write (13.16) as

$$H_m(x) = \sum_{k=0}^{n} p_k(x) h_k(x), \tag{13.18}$$

$p_k(x)$ being a polynomial of degree $m_k - 1$.

14.4. The Heaviside „cover up" method for partial fraction expansion and interpolation

We have already seen in Chapter 11 that Lagrange interpolation may serve as a tool for partial fraction expansion in the case of simple roots of the divisor polynomial. Now we guess, the same can be done with Hermite interpolation for multiple roots.

The Heaviside method was initially introduced for partial fraction expansion, see e.g. Antoulas *et al* (2003), but as it will be seen here, it is also capable of giving an interpolating polynomial. From (13.18) we have

$$\frac{H_m(x)}{(x - x_k)^{m_k} h_k(x)} = \sum_{j=0}^{n} \frac{p_j(x)}{(x - x_k)^{m_k}} \frac{h_j(x)}{h_k(x)}, \tag{13.19}$$

such that we have the terms for $j = k$:

$$\frac{p_k(x)}{(x-x_k)^{m_k}} = \frac{\gamma_{k,0}}{(x-x_k)^{m_k}} + \frac{\gamma_{k,1}}{(x-x_k)^{m_k-1}} + \ldots + \frac{\gamma_{k,m_k-1}}{(x-x_k)}.$$

Hence multiplying by $(x-x_k)^{m_k}$ and differentiating will give the coefficients

$$\gamma_{k.i} = \frac{1}{i!}\left(\frac{d}{dx}\right)^i \frac{H_m(x)}{h_k(x)}\bigg|_{x=x_k}, \quad i = 0,\ldots,m_k-1. \tag{13.20}$$

Observe that all the other terms in (13.19) will have a multiplier of $(x-x_k)^j$, $j>0$ and when substituting $x = x_k$, they will cancel. When differentiating, we have to substitute function values and derivatives of $H_m(x)$ at x_k. If we put in function data as given, the resulting polynomial (13.18) will be the desired interpolating polynomial.

But we can also get the polynomial coefficients directly from (13.18) by differentiation and equating. In particular, if only the function value $f_k = f(x_k)$ is given, then $h_k(x_k)=1$, $p_k(x)$ is a constant polynomial and

$$f_k = H_m(x_k) = p_k(x_k)$$

as we may confine ourselves only to the term $p_k(x)$ in (13.18).

If the function value and the first derivative are given, then we have from (13.18) the additional condition:

$$H'_m(x_k) = p'_k(x)h_k(x) + p_k(x)h'_k(x) + \left(\sum_{j\neq k}^{n} p_j(x)h_j(x)\right)'\bigg|_{x=x_k},$$
$$= p'_k(x_k) + f_k h'_k(x_k) = f'_k$$

from where $p_k(x) = f_k + (f'_k - f_k h'_k(x_k))(x-x_k)$ follows. Observe that the $j \neq k$ terms will vanish for $x = x_k$.

If we still have the second derivative given, then a next term of $p_k(x)$ can be identified:

$$H''_m(x_k) = p''_k(x)h_k(x) + 2p'_k(x)h'_k(x) + p_k(x)h''_k(x) + \text{Sum}'\big|_{x=x_k}$$
$$= p''_k(x_k) + 2p'_k(x_k)h'_k(x_k) + f_k h''_k(x_k) = f''_k$$

that is,

$$2\gamma_{k,2} + 2(f'_k - f_k h'_k(x_k))h'_k(x_k) + f_k h''_k(x_k) = f''_k$$

and the resulting polynomial is

$$p_k(x) = f_k + (f'_k - f_k h'_k(x_k))(x-x_k) +$$
$$\left[\left(f''_k - f_k h''_k(x_k)\right)/2 - (f'_k - f_k h'_k(x_k))h'_k(x_k)\right](x-x_k)^2. \tag{13.21}$$

This procedure can be continued up to a desired derivative.

A third method, if we have expanded $h_k(x)$ by the powers of $(x-x_k)$, is that we can get $p_k(x)$ by dividing power series (Fourier division) as indicated in (13.20), where the Taylor polynomial of $H_m(x)$ around x_k is given by the incoming data.

Finally we remark that barycentric Hermite interpolation can also be done, see Sadiq *et al* (2013).

14.5. Inverse interpolation

We may interchange the variables x and y in the interpolation, hence getting a polynomial of type $x = x(y)$. This technique is called *inverse interpolation*. It is applied e.g. if we are interested in a location where a given function value is assumed. This happens to be the case if one looks for a zero of the function. Substituting $y = 0$ into the polynomial returns an approximate location for the zero. One has to be careful in inverse interpolation: the function has to be a one-to-one mapping from $[a,b]$ to $[f(a), f(b)]$, otherwise surprising situations may happen.

14.6. Problems

14.1. Find the interpolating polynomial for (x_0, f_0), $(x_1, f_1, f_1^{'}, f_1^{''})$.

14.2. Elaborate Hermite interpolation, if $(x_0; f_0, f_0', f_0'') = (-1; 1, 2, -2)$ and $(x_1; f_1, f_1') = (2; 1, -1)$!

14.3. Having a table for the values of the sine function, the derivatives are also known by the familiar connection to the cosine function. Thus we may use the Hermite-Fejér interpolation, where function values and derivatives are given at all points. Then how densely should the function be tabulated in $[0, \pi/2]$ if we want to to get the values of the function with an error of 10^{-4}?

14.4. Derive the error bound for the Hermite-Fejér interpolation, if we have Chebyshev abscissas.

14.5. Show that $h_k(x) = \left(l_k(x)\right)^2$, $p_{k,0}(x) = 1 - 2l_k'(x_k)(x - x_k)$ and $p_{k,1}(x) = x - x_k$ in the case of the Hermite-Fejér interpolation.

14.6. Derive the coefficients of the multiplier polynomial $p_{k,1}(x) = (x - x_k) + \beta_2 (x - x_k)^2$ for the Hermite base polynomial $h_{k,1}(x)$, (see Sect. 14.3), where data are given up to the second derivatives at point x_k.

14.7. What are the Hermite base polynomials, if the support points are: $(x_0; f_0, f_0')$, and $(x_1; f_1, f_1')$?

14.8. Write a Matlab program for Hermite divided differences!

14.9. A function goes through the points: $(1,-1)$, $(2,1)$, $(3,2)$, $(5,3)$. By choosing inverse interpolation, give an approximate location of zero with Neville's interpolation!

14.10. Let $f(x)=\sum_{j=0}^{n} a_j x^j$ be a polynomial of degree n. What is the value of $f[x_0, x_1, \ldots, x_n]$?

14.11. Find the three base polynomials for the interpolation problem $(x_0, f_0, f_0'),\ (x_1, f_1)$!

15. Splines

If we increase the degree of interpolating polynomials, it is a common experience that polynomials of higher degree may show violent oscillations near the support points. So badly that one may not believe at glance that it approximates the function adequately. Although Weierstrass' theorem assures for $f(x)\in C[a,b]$ that there exists a sequence of support abscissas for which the interpolating polynomials converge uniformly to $f(x)$, Faber's theorem states that for such functions one can also find another sequence of support abscissas for which there is no convergence. From these facts came the idea of changing polynomials to piecewise polynomials which fulfil some continuity conditions. Such functions are called splines.

15.1. Spline functions

15.1.1 Definition of splines

Function $s_l(x)$ is called a spline of degree l if

i) in every subinterval it is a polynomial of degree l and

ii) it is continuously differentiable $l-1$-times in $[a,b]$: $s_l(x)\in C^{l-1}[a,b]$.

The linear combination of such functions will also be a *spline* of degree l and it is straightforward to check that l-th degree splines form a vector space.

Let $\Theta_n=\{x_i, f_i=f(x_i)\}_{i=0}^n$ be the set of support points, where the abscissas or nodes are ordered: $x_{i-1}<x_i,\ 0<i$, and denote by $S_l(\Theta_n)$ the set of splines of order l. For the dimension of this space observe that there are $n(l+1)$ polynomial coefficients in the n subintervals and joining the derivatives at midpoints requires $(n-1)l$ conditions, hence, there remain $n+l$ free parameters. If it is an interpolatory spline, then $n+1$ function values are still given such that there are only $l-1$ free parameters left.

Spline interpolation is simple in terms of *hat functions* $u_i(x)$:

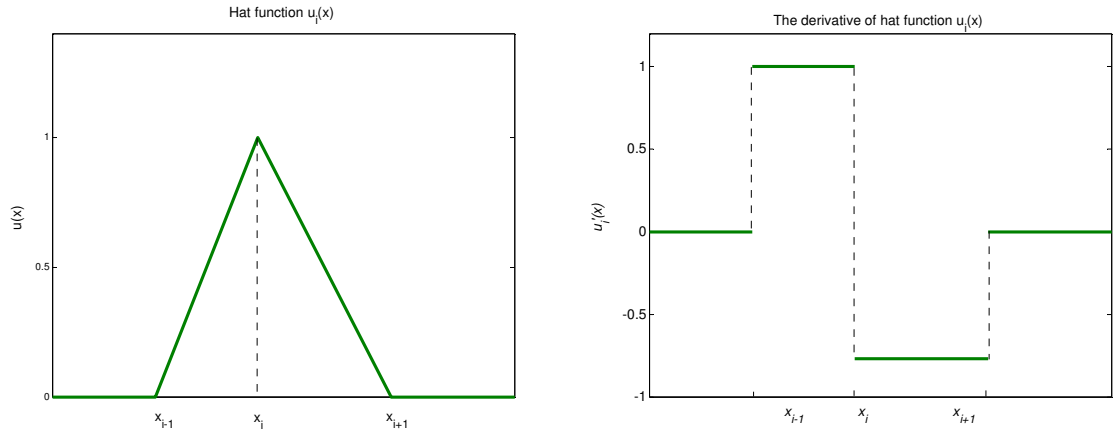

Fig 1. The hat function and its derivative

$$u_i(x)=\begin{cases}\dfrac{x-x_{i-1}}{x_i-x_{i-1}}, & \text{if } x_{i-1}\le x\le x_i\\ \dfrac{x_{i+1}-x}{x_{i+1}-x_i}, & \text{if } x_i\le x\le x_{i+1}\\ 0, & \text{otherwise}\end{cases}\qquad u_i'(x)=\begin{cases}\dfrac{1}{x_i-x_{i-1}}, & \text{if } x_{i-1}<x<x_i\\ \dfrac{-1}{x_{i+1}-x_i}, & \text{if } x_i<x<x_{i+1}\\ 0, & \text{if } x<x_{i-1},\ x_{i+1}<x\end{cases}$$

The first derivative does not exist at the nodes however; one may look for the lower or upper limits there, and that will be enough for the future. The higher derivatives of hat functions all disappear.

15.2. Splines of first degree: $s(x)\in S_1(\Theta_n)$

$$s(x)=f_{i-1}\frac{x_i-x}{x_i-x_{i-1}}+f_i\frac{x-x_{i-1}}{x_i-x_{i-1}},\qquad x\in\left[x_{i-1},x_i\right]. \tag{14.1}$$

The result is a broken line. Our computer usually uses this approximation to draw a function given by the set Θ_n. If support abscissas are dense enough, broken lines are not seen. Function $s(x)$ can be given in Lagrange interpolation style with the aid of hat functions:

$$s(x)=\sum_{i=0}^{n}f_iu_i(x). \tag{14.2}$$

15.3. Splines of second degree: $s(x)\in S_2(\Theta_n)$

For the sake of simplicity, assume that

$f(a)$ and $f'(a)$ are given at the starting point. Then using data

$x_0=a$	x_0	x_1
$f(x_0)$	$f'(x_0)$	$f(x_1)$

one can apply Hermite's interpolation to produce a polynomial of second order that belongs to the first interval. For continuation take the first derivative of this polynomial in x_1, and by adding $f(x_1)$ and $f(x_2)$ we may continue the procedure for interval $\left[x_1,x_2\right]$. Generally the table of divided differences for $\left[x_i,x_{i+1}\right]$ is:

$$\begin{array}{llll} x_i & f(x_i) & & \\ x_i & f(x_i) & s'(x_i) & \\ x_{i+1} & f(x_{i+1}) & f[x_i,x_{i+1}] & \dfrac{f[x_i,x_{i+1}]-s'(x_i)}{x_{i+1}-x_i}, \end{array}$$

and

$$s(x)=f(x_i)+s'(x_i)(x-x_i)+f[x_i,x_i,x_{i+1}](x-x_i)^2,\quad x\in[x_i,x_{i+1}],$$

where the approximation $f'(x_i)\approx s'(x_i)$ is applied when computing the second divided difference.

If the derivative is not known at the beginning, then one may make a second degree polynomial for the first three points and then apply the continuation method given here for subsequent points.

15.4. Splines of third degree: $s(x)\in S_3(\Theta_n)$

For such splines the first and second derivatives also need to join at data points. The $3(n-1)$ conditions set up a linear system of the same order. However, one can derive a simpler approach by performing $2n$ interpolations analytically with a skilful choice of an interpolation subproblem. Therefore we look for a third degree polynomial that interpolates for data $\{x_0; f_0, f_0''\}$, $\{x_1; f_1, f_1''\}$.

Now denote by $l_{ki}(x)$ the Hermite interpolation base polynomials. The first index refers to the node points and the second index indicates the order of the derivative.

First we construct polynomial $l_{00}(x)$. The defining equations are: $l_{00}(x_0)=1$, $l_{00}(x_1)=0$, $l_{00}''(x_0)=0$ and $l_{00}''(x_1)=0$. The second derivative is of degree 1 at most and according to the equations it disappears at both points, hence it is identically 0. As a consequence, $l_{00}(x)$ has degree 1 and it is fully determined by the first two equations:

$$l_{00}(x)=(x_1-x)/(x_1-x_0)=u_0(x),\quad x\in[x_0,x_1]. \tag{14.3}$$

Polynomial $l_{02}(x)$ satisfies the conditions: $l_{02}(x_0)=0$, $l_{02}(x_1)=0$, $l_{02}''(x_0)=1$ and $l_{02}''(x_1)=0$. The second derivative has degree 1 and like $l_{00}(x)$, it is given by $l_{02}''(x)=u_0(x)$, $x\in[x_0,x_1]$. Integrating twice yields

$$l_{02}'(x)=\frac{u_0^2(x)}{2u_0'(x)}+\beta,$$

$$l_{02}(x)=\frac{u_0^3(x)}{6u_0'^2(x)}+\beta\frac{u_0(x)}{u_0'(x)}+\gamma,\quad x\in(x_0,x_1)$$

where we have exploited the fact that $u_0'(x)$ is constant in the interval. Because of $u_0(x_1)=0$, $\gamma=0$ follows from $l_{02}(x_1)=0$. On the other hand, condition $l_{02}(x_0)=0$ gives $\beta=-1/(6u_0'(x))$, such that

$$l_{02}(x)=\frac{u_0^3(x)-u_0(x)}{6u_0'^2(x)}. \tag{14.4}$$

The other two polynomials $l_{10}(x)$ and $l_{12}(x)$ are determined similarly, and the results are:

$$l_{10}(x)=u_1(x),\quad l_{12}(x)=\frac{u_1^3(x)-u_1(x)}{6u_1'^2(x)},\quad x\in(x_0,x_1). \tag{14.5}$$

With these the interpolating polynomial in (x_0,x_1) is

$$p_{0,3}(x) = f_0 l_{00}(x) + f_0'' l_{02}(x) + f_1 l_{10}(x) + f_1'' l_{12}(x) =$$
$$= f_0 u_0(x) + f_0'' \frac{u_0^3(x) - u_0(x)}{6u_0'^2(x)} + f_1 u_1(x) + f_1'' \frac{u_1^3(x) - u_1(x)}{6u_1'^2(x)}.$$

Although the first derivative does not exist at the border points, the multipliers of the second derivatives will tend to zero there, hence we may take that the interpolation is valid in the closed interval $[x_0, x_1]$. Now assume, the function values and the second derivatives are given at points $x_0, x_1, \ldots, x_n$. Then we can write the interpolating polynomial for each subinterval as:

$$p_{i,3}(x) = f_i u_i(x) + f_i'' \frac{u_i^3(x) - u_i(x)}{6u_i'^2(x)} + f_{i+1} u_{i+1}(x) + f_{i+1}'' \frac{u_{i+1}^3(x) - u_{i+1}(x)}{6u_{i+1}'^2(x)}, \quad x \in [x_i, x_{i+1}]$$

but observing the definition of hat functions, the interpolating bundle of polynomials can be given by the sum:

$$p_{[a,b],3}(x) = \sum_{i=0}^{n} f_i u_i(x) + f_i'' \frac{u_i^3(x) - u_i(x)}{6u_i'^2(x)}, \quad x \in [a,b]. \tag{14.6}$$

In this way we have a function which is a third degree polynomial in each subinterval, and it has continuous 0-th and second derivatives in $[a,b]$.

Accordingly we choose the spline of third degree with respect to the support points Θ_n as:

$$s_3(x) = \sum_{i=0}^{n} f_i u_i(x) + w_i \frac{u_i^3(x) - u_i(x)}{6u_i'^2(x)}, \quad x \in [a,b], \quad w_i = s_3''(x_i). \tag{14.7}$$

The second derivatives w_i will be determined from the condition that the first derivatives be continuous at the nodes x_i. The first derivative

$$s_3'(x) = \sum_{i=0}^{n} f_i u_i'(x) + w_i \frac{3u_i^2(x) - 1}{6u_i'(x)}, \quad x \in [a,b] \tag{14.8}$$

is continuous, if lower and upper limits are equal at the inner nodes. Denote by $s_3'(x_i - 0)$ and $s_3'(x_i + 0)$ the lower and upper limits at x_i. Then only two hat functions will give contributions to the limits:

$$s_3'(x_i - 0) = f_{i-1} u_{i-1}'(x_i -) + f_i u_i'(x_i -) + w_{i-1} \frac{3u_{i-1}^2(x_i) - 1}{6u_{i-1}'(x_i -)} + w_i \frac{3u_i^2(x_i) - 1}{6u_i'(x_i -)} =$$
$$= \frac{-f_{i-1} + f_i}{h_{i-1}} + \frac{w_{i-1} + 2w_i}{6} h_{i-1}, \quad \text{where } h_{i-1} = x_i - x_{i-1}$$

and

$$s_3'(x_i + 0) = \sum_{j=i}^{i+1} f_j u_j'(x_i +) + w_j \frac{3u_j^2(x_i) - 1}{6u_j'(x_i +)} = \frac{f_{i+1} - f_i}{h_i} - \frac{w_{i+1} + 2w_i}{6} h_i.$$

Equating the two at x_i:

$$f[x_{i-1}, x_i] + \frac{2w_i + w_{i-1}}{6} h_{i-1} = f[x_i, x_{i+1}] - \frac{2w_i + w_{i+1}}{6} h_i, \quad h_i = x_{i+1} - x_i$$

And collecting unknown terms to the left gives

$$\frac{w_{i-1}h_{i-1}}{6}+\frac{w_i\,(h_{i-1}+h_i)}{3}+\frac{w_{i+1}h_i}{6}=f[x_i,x_{i+1}]-f[x_{i-1},x_i].$$

Further introduce $\sigma_{i-1}=h_{i-1}/(h_{i-1}+h_i)$, which leads to a simple form of the equations

$$w_{i-1}\sigma_{i-1}+2w_i+w_{i+1}(1-\sigma_{i-1})=6f[x_{i-1},x_i,x_{i+1}],\quad i=1,2,\ldots,n-1. \tag{14.9}$$

The resulting linear system has a diagonally dominant tridiagonal matrix, which is advantageous when solving. However, there is no enough number of equations to find the values of w_0 and w_n, hence we have to specify further conditions for the endpoints. One can choose one from the following four cases:

1. Hermite end conditions: the first derivatives $f'(a)$ and $f'(b)$ are given.
2. The second derivatives are given at the endpoints. If they are not known, $w_0=w_n=0$ can be a possible choice. This resulting spline is called *natural spline*.
3. Periodic end condition. If the function is periodic and interpolation is done in the whole period, then $w_0=w_n$ and only one additional equation is needed. For $i=0$ one can identify the following terms for the first equation: $w_{-1}=w_{n-1}$, $h_{-1}=h_{n-1}$, $\sigma_{-1}=h_{-1}/(h_{-1}+h_0)$ and $f_{-1}=f_{n-1}$.
4. We choose continuous third derivatives for the first two and the last two intervals such that the third derivative is constant in these intervals. This spline interpolant has the property that it returns the third degree polynomial for which the function values were given from that polynomial. These third derivatives can be expressed as divided differences:

$$\begin{aligned}(w_1-w_0)/h_0&=(w_2-w_1)/h_1,\\(w_{n-1}-w_{n-2})/h_{n-2}&=(w_n-w_{n-1})/h_{n-1}.\end{aligned} \tag{14.10}$$

15.5. Example

Find the first and last equations that should be added to system (14.9) when Hermite end conditions are applied!

Solution. Take $i=0$ in (14.9) and formally introduce x_{-1}. Now let x_{-1} tend to x_0, hence $\sigma_{-1}\to 0$ and thus

$$2w_0+w_1=6f[x_0,x_{0,}x_1]=6\left(f[x_0,x_1]-f_0'\right)/h_0. \tag{14.11}$$

Fort the last equation substitue $i=n$ in (14.9) and let x_{n+1} tend to x_n:

$$w_{n-1}+2w_n=6f[x_{n-1},x_n,x_n]. \tag{14.12}$$

With these two equations (14.9) is already solvable as there are as many equations as the number of unknowns.

15.6. Problems

15.1. How simple will become (14.9) for equidistant nodes?

15.2. We have at least four support points which came from a third degree polynomial. Show that by using the fourth end conditions, the resulting spline will return that polynomial. In other words, this spline interpolant is exact for cubic polynomials.

15.3. Write a Matlab program to calculate the values of a cubic spline!

15.4. For representig cubic splines, we may use other cubic polynomials which serve automatically continuous 0-th and first derivative at the inner nodes. In order to achieve this, we derive polynomials for the Hermite interpolation problem: $\{x_0; f_0, f_0'\}$, $\{x_1; f_1, f_1'\}$. Now find Hermite base polynomials (see e. g. Problem 14.5) for the interval (x_0, x_1) and show that like in (14.6), one gets the following formula:

$$\tilde{p}_{[a,b],3}(x) = \sum_{i=0}^{n} f_i \left(-2u_i^3(x) + 3u_i^2(x)\right) + f_i' \frac{u_i^3(x) - u_i^2(x)}{u_i'(x)}, \quad x \in [a,b].$$

15.5. Observing the previous problem, we can represent interpolating cubic splines with function values f_i and spline first derivatives t_i at the inner nodes x_i:

$$s_3(x) = \sum_{i=0}^{n} f_i \left(-2u_i^3(x) + 3u_i^2(x)\right) + t_i \frac{u_i^3(x) - u_i^2(x)}{u_i'(x)}, \quad x \in [a,b],$$

where $f_i = s_i(x_i)$, $t_i = s_i'(x_i)$ and $u_i(x)$ are hat functions. State a system of linear equations of the unknowns t_i from the condition that the second derivative is continuous at the inner nodes!

15.6. We should like to approximate the solution of the following differential equation: $-v''(x) = g(x), \quad x \in [0,1], \quad v(0) = v(1) = 0,$ where $g(x)$ is a given function. Divide $[0,1]$ into n equal parts and using (14.9) state a system of equations for the unknown approximate function values $v(x_i),\ i = 1, \ldots, n-1$!

15.7. Write a Matlab program to solve the previous boundary value problem!

15.8. Using the error theorem of interpolation, derive upper bound for the error of first degree spline interpolation!

15.9. Show that the function

$$s\ (x) = \sum_{i=0}^{n} f_i \left(-2u_i^3(x) + 3u_i^2(x)\right), \quad x \in [a,b],$$

has continuous first derivative at x_i, where $u_i(x)$ is the i-th hat function and x_i-s are the support abscissas.

16. Solution of nonlinear equations I.

Hitherto we dealt with the solution of linear systems. But many times one or more zeros of a function are needed by solving the equation

$$f(x)=0\,, \tag{16.1}$$

where $f(x)\in C[a,b]$ is a function in one variable. The value of x^* for which $f(x^*)=0$ holds is called a *root* (in the case of polynomials) or *zero* of $f(x)$. The zero has *multiplicity* m, if the function has the form $f(x)=(x-x^*)^m g(x),\ \ g(x^*)\neq 0$. For $m=1$ we call the zero *simple*. We shall deal with cases where the solution can be found by numerical methods.

16.1. The interval of the root

If there is a sign change: $f(a)f(b)<0$, then there exists at least one root in the interval $[a,b]$ because of continuity. If the first derivative of $f(x)$ also exists and keeps sign in the interval $[a,b]$, then there must be only one root there.

If the function is not monotonic, then it may be helpful to divide $[a,b]$ into subintervals such that there is a sign change there. Thus the odd roots of $f(x)$ can be separated. For differentiable functions the even roots can be sought by the roots of $f'(x)$, because the even roots have turned into odd ones, it may also be possible to look for the roots of $f(x)/f'(x)$, the roots of which are all single.

16.2. Fixed-point iteration

One possible approach is to reformulate $f(x)=0$ into a fixed-pont iteration:

$$x=F\left(x\right) \tag{16.2}$$

and the solution is given by x^* such that $x^*=F(x^*)$. Example: we look for the root of $x^2-\sin(x)=0$. Then we may try the iterative formula $x_{k+1}=\sqrt{\sin(x_k)}$. It is always possible to give a fixed-point iteration, e.g. $x=x+cf(x)$, which is a function, where c is a nonzero constant, but we may also choose a function $c(x)$ such that the convergence properties of the iteration are improved. The next theorem is about the existence of a fixed-pont.

16.2.1 The fixed-point theorem of Brouwer[1]

If $F(x)$ is continuous in $[a,b]$ and $F:[a,b]\to[a,b]$, then there is a fixed-point of F in $[a,b]$.

Proof. Introduce $g(x)=x-F(x)$, then $g(a)\le 0$ and $g(b)\ge 0$, from which $g(a)g(b)\le 0$ follows. If the equality sign holds here, then there is already a zero. Otherwise, in the case of sign change there must be a a root in $[a,b]$ because of continuity. ■

---

[1] The statement of the theorem in many dimensions: if the continuous function $F(x)$ maps the sphere onto itself, then it has a fixed-point.

16.2.2 Theorem on contraction

If $F : S \to \mathbb{R}$ *is continuously differentiable in the closed interval* S *and* $|F'(x)| < 1,\ \forall x \in S$, *then* F *is a contraction.*

Proof. From Lagrange's mean value theorem we have $x, y \in S$-re $\exists \zeta :\ F(x) - F(y) = F'(\zeta)(x - y)$. Taking absolute values and exploiting the fact that $|F'(x)|$ has maximum in S gives the inequality

$$|F(x) - F(y)| \le \max_{x \in S} |F'(x)| |x - y|, \quad x, y \in S .$$

Therefore F is a contraction with $q = \max_{x \in S} |F'(x)| < 1$ Lipschitz constant. ■

16.2.3 Corollary

If $F(x)$ *is a contraction such that* S *is mapped onto itself, then by the Banach's fixed-point theorem there exists one zero in* S *and after one step of iteration it is possible to estimate the distance from the zero with the help of the Lipschitz constant.*

Now returning to the previous example: the derived iteration may be convergent in the domain, where $\left(\sqrt{\sin(x)}\right)' = \dfrac{\cos(x)}{2\sqrt{\sin(x)}} < 1$. As seen, this expression does not have sense for $x = 0$ or $x = \pi$, because we should divide by 0. But for $x = \pi/4$ it returns the value $2^{-5/4}$, which seems better. Drawing the parabola of x^2 and the $\sin(x)$ function, we see that there are two nonnegative roots. One is zero, the other one is close to $x = \pi/4$, so that we may expect that staring with $\pi/4$, the iteration will be convergent. It is also seen that starting with a small positive value, the iteration will not converge to zero, because it will always tend to the larger root.

But if we choose the iteration $x = \arcsin(x^2)$, then it can be checked easily that for small positive starting values the iteration will tend to zero. But if we choose $x = 1$ for the starting value, then we get $\pi/2$ first, followed by complex numbers, because the argument is greater than 1.

Conclusion: we have to pay attention to the mapping of the function, because we may get to places where the necessary convergence properties are missing, or we may get convergence to another root that does not interest us.

If the variable x appears at more places in the equation that is to be solved for zero, then it is possible to find more expressions like $x = F(x)$. For instance, assume it can be found at two places. Then one can show that the two fixed-point iterations may not be convergent at one place. Let $f(x_1, x_2) = 0$ be the equation to be solved, where the two occurrences are identified by x_1 and x_2. Denote by $F_i(x)$ the iteration function gained by expressing x_i. Consequently we have the two equations

$$f(F_1(x), x) = 0 \quad \text{and} \quad f(x, F_2(x)) = 0. \tag{16.3}$$

Let α be a simple root: $f(\alpha, \alpha) = 0$. Differentiating the expressions in (16.3) by x and substituting $x = \alpha$, one gets the equations

$$f_1'(\alpha,\alpha)F_1'(\alpha)+f_2'(\alpha,a)=0,$$
$$f_1'(\alpha,\alpha)+f_2'(\alpha,\alpha)F_2'(\alpha)=0,$$

where the index of f shows the place where the differentiation was done. When obtaining a nonzero solution for a simple root, $f_i'(\alpha,\alpha)$ must be nonzero and the determinant of the 2×2 system should be 0:

$$\begin{vmatrix} F_1'(\alpha) & 1 \\ 1 & F_2'(\alpha) \end{vmatrix} = F_1'(\alpha)F_2'(\alpha)-1=0,$$

from where

$$\left|F_1'(\alpha)\right| = 1/\left|F_2'(\alpha)\right|. \tag{16.4}$$

Unless both derivatives have absolute value 1, one iteration function is convergent; the other one is divergent in the neighbourhood of a root. When there are more than two occurrences of x, one can proceed similarly. However, then the situation is worse, because it may happen that no iteration function will be convergent. Therefore we should organize the occurrences of x into two groups and express x from one group entirely. For instance, having the equation $3x^2-2x-\exp(2.2x)+1=0$, we may look for the roots by taking $\exp(2.2x)+1$ for the constant term of the second degree polynomial:

$$x=\left(2+\sqrt{4+12(\exp(2.2x)-1)}\right)/6=F(x).$$

16.3. Speed of convergence

Let the series $\{x_n\}$ be convergent, $\lim_{n\to\infty} x_n = x^*$. Denote the n-th error by $\varepsilon_n = x_n - x^*$. Then, if there exists constant c and a number $p\ge 1$ such that

$$\left|\varepsilon_{n+1}\right| \le c\left|\varepsilon_n\right|^p, \quad n=0,1,\ldots, \tag{16.5}$$

then the series $\{x_n\}$ has convergence of *p-th order*. If

- $p=1$, then the convergence is *linear* or *first order*,
- $1<p<2$, then the convergence is *superlinear*,
- $p=2$, it is *quadratic* or *second order*,
- $p=3$, *cubic* or *third order*.

Number p characterizes the speed of convergence of an iterational method. For instance, if $p=2$, then roughly saying this would mean that the number of accurate digits is doubled at every iteration step.

The fixed-point iteration doest not have that speed. It can be shown that $p=1$ holds, that is the convergence is of first order if $\left|F'(x^*)\right|\ne 0$. For,

$$\left|\varepsilon_{n+1}\right| = \left|x_{n+1}-x^*\right| = \left|F(x_n)-F(x^*)\right| \le q\left|x_n-x^*\right| = q\left|\varepsilon_n\right|. \tag{16.6}$$

If $F'(x^*)=0$, then the convergence has higher order. This statement is specified in the next

16.3.1 Theorem

Let F be a real function: $F(S)\subset S\subset\mathbb{R}$, S is closed. Assume $F\in C^m(S)$ and $F^{(k)}(x^*)=0$, $k=1,2,\cdots,m-1$. Then the fixed-point iteration given by F has a speed of convergence $p=m$.

Proof. The Taylor-polynomial around x^* with m-th order remainder term is

$$F(x)=F(x^*)+F'(x^*)(x-x^*)+\ldots+\frac{F^{(m-1)}(x^*)}{(m-1)!}(x-x^*)^{m-1}+\frac{F^{(m)}(\xi_x)}{m!}(x-x^*)^m$$

where according to the assumption, the first, second, … , $m-1$-th derivatives vanish. Substitute $x=x_n$ and take into account the relations $x^*=F(x^*)$ and $x_{n+1}=F(x_n)$, with these

$$x_{n+1}-x^*=\frac{F^{(m)}(\xi_x)}{m!}(x_n-x^*)^m,$$

from where

$$\left|\varepsilon_{n+1}\right|=\frac{\left|F^{(m)}(\xi_x)\right|}{m!}\left|x_n-x^*\right|^m\le\frac{M_m}{m!}\left|\varepsilon_n\right|^m,\quad n=0,1,\ldots$$

and $M_k=\max\limits_{x\in S}\left|F^{(k)}(x)\right|$. This shows the m-th order convergence. ■

16.4. Newton iteration (Newton-Raphson method) and the secant method

If the function is differentiable in a neighbourhood of the root, then we may search for it by taking the intersection of the tangent line from x_n to the x axis. Algebraically it can be done by solving the first degree Taylor polynomial around x_n for zero. The result is the next approximate for the root x_{n+1}:

$$0=f(x_n)+f'(x_n)(x_{n+1}-x_n).$$

From here the iteration function of the Newton-Raphson method is:

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)}=F(x_n). \tag{16.7}$$

The *secant method* can be gained by replacing the derivative here with the divided difference for the last two points:

$$x_{n+1}=x_n-\frac{f(x_n)(x_n-x_{n-1})}{f(x_n)-f(x_{n-1})}=\frac{f(x_n)x_{n-1}-f(x_{n-1})x_n}{f(x_n)-f(x_{n-1})}=F(x_{n-1},x_n), \tag{16.8}$$

such that this iteration function uses two previous points. An advantage here over the Newton method is that no derivative is needed, the computation of which can be cumbersome sometimes. On the other hand, a disadvantage is the lower speed of convergence.

16.4.1 Theorem, error of the secant method

Let $f(x)\in C^2[x^*,x_{n-1},x_n]$, then we have for the $n+1$ -th error of the secant method

$$\varepsilon_{n+1} = \varepsilon_n \varepsilon_{n-1} \frac{f''(\xi)}{2f'(\eta)}, \quad \xi \in [x^*, x_{n-1}, x_n], \quad \eta \in [x_{n-1}, x_n], \tag{16.9}$$

where x^* *is the zero of the function and* $[x^*, x_{n-1}, x_n]$ *denotes the interval containing the points given in the brackets.*

Proof. The statement will be shown using (16.8) and divided differences. We take into account the relation $f(x^*) = 0$:

$$\begin{aligned} \varepsilon_{n+1} = x_{n+1} - x^* = x_n - x^* - (x_n - x^*) \frac{f(x_n) - f(x^*)}{x_n - x^*} \frac{1}{f[x_{n-1}, x_n]} = \varepsilon_n \left(1 - \frac{f[x^*, x_n]}{f[x_{n-1}, x_n]} \right) = \\ = \varepsilon_n \left(\frac{f[x_{n-1}, x_n] - f[x^*, x_n]}{f[x_{n-1}, x_n]} \frac{\varepsilon_{n-1}}{x_{n-1} - x^*} \right) = \varepsilon_n \varepsilon_{n-1} \frac{f[x^*, x_{n-1}, x_n]}{f[x_{n-1}, x_n]} \end{aligned} \tag{16.10}$$

and applying Corollary 14.1.1 for the relation between divided differences and derivatives gives the statement. ■

16.4.2 Corollary

Taking the limit $x_{n-1} \to x_n$, we get the corresponding result for the Newton method:

$$\varepsilon_{n+1} = \varepsilon_n^2 \frac{f''(\xi)}{2f'(x_n)}, \quad \xi \in [x_n, x^*], \tag{16.11}$$

It is seen, if there is convergence, then it has order 2, assuming that $f'(x^*) \neq 0$ holds. If, in addition, $f''(x^*) = 0$ holds, then higher order convergence may take place.

16.4.3 Theorem on monotone convergence

Let $f \in C^2[a,b]$, $f(x^*) = 0$, $x^* \in [a,b]$, *and the derivatives* $f'(x), f''(x)$ *do not change sign in* $[a,b]$, *moreover, for the starting point* $x_0 \in [a,b]$ $f(x_0) f''(x_0) > 0$ *is fulfilled. Then the Newton method converges and the generated series* $\{x_n\}$ *tends monotonically to the solution* x^*.

Proof. According to (16.11), the Newton method returns iterates which are located either to the right or to the left of the root, because the sign of f''/f' is constant. Subtracting x^* from (16.7) yields

$$\varepsilon_{n+1} = \varepsilon_n - f(x_n)/f'(x_n). \tag{16.12}$$

Because of $f(x_0) f''(x_0) > 0$, the sign of $f''(x_0)$ and $f(x_0)$ is the same, and this is also true for $f''(x_0)/f'(x_0)$ and $f(x_0)/f'(x_0)$. Choose $n = 0$ in (16.11). Now if $\varepsilon_1 > 0$, then $f''(x_0)/f'(x_0)$ is positive and ε_0 in (16.12) was diminished by the positive $f(x_0)/f'(x_0)$; this is the case in all subsequent steps for $\varepsilon_n > 0$. Similarly, in the case of $\varepsilon_1 < 0$ all further $\varepsilon_n < 0$ is increased. We got the result that either from below or from above the ε_n-s are tending to zero. ■

Corollary. The formula in (16.9) shows that starting the secant method such that for $x_0, x_1 \in [a,b]$, the signs of ε_0, ε_1 and $f''(x_0)/f'(x_0)$ are identical, then the secant method

will also produce a monotone converging sequence on the conditions of the theorem. It is so, because the divided difference in it can always be replaced by a derivative that does not change sign in $[a,b]$.

16.4.4 Theorem on local convergence

Let $f \in C^2[a,b]$, $f(x^*)=0$, $f'(x)\neq 0$, $x, x^* \in [a,b]$, *and assume the condition*

$$\left|x_0 - x^*\right| < \frac{2\min_{[a,b]}\left|f'(x)\right|}{\max_{[a,b]}\left|f''(x)\right|} = \frac{1}{M} \tag{16.13}$$

is fulfilled for the starting point $x_0 \in [a,b]$. *Then starting from* x_0, *the Newton-Raphson method converges to* x^*. *Moreover, the secant method converges, if* x_0 *and* x_1 *satisfy condition* (16.13).

Proof. Considering (16.9) or (16.11), there is contraction in the iteration from the first step if

$$\left|\varepsilon_0 \frac{f''(\xi)}{2f'(\eta)}\right| \le \left|x_0 - x^*\right| \frac{\max_{[a,b]}\left|f''(x)\right|}{2\min_{[a,b]}\left|f'(x)\right|} < 1$$

holds. Then the contraction will be inherited for subsequent steps and the statement comes by rearranging. For the case of the secant method, the same condition still should be demanded for ε_1 in the second step to insure convergence. ■

Having the constant M at hand, then it is possible to estimate the $n+1$ -th error. Introduce $d_k = M\left|\varepsilon_k\right|$, then we can proceed as

$$d_{n+1} = M\left|\varepsilon_{n+1}\right| \le M^2\varepsilon_n^2 \quad \rightarrow \quad d_{n+1} \le d_0^{2^n} \quad \rightarrow \quad \left|\varepsilon_{n+1}\right| \le \left(M\varepsilon_0\right)^{2^n}. \tag{16.14}$$

16.4.5 Theorem, the speed of convergence for the secant method

On the conditions of Theorem 16.4.4 and starting from points x_0, x_1, *the secant method will converge with an asymptotic speed of* $p = (1+\sqrt{5})/2 \approx 1{,}62$ *to* x^*.

Proof. Applying (16.9),

$$\left|\varepsilon_{n+1}\right| \le M\left|\varepsilon_n\right|\left|\varepsilon_{n-1}\right|$$

holds, where M is the same as in (16.13). Again with the notation $d_k = M\left|\varepsilon_k\right|$

$$d_{n+1} \le d_n d_{n-1}, \quad n = 1,2,\ldots$$

At start $\left|x_0 - x^*\right| < 1/M$ and $\left|x_1 - x^*\right| < 1/M$, with these $d_0, d_1 < 1$. Therefore we may state: $\exists d < 1$: $d_0, d_1 \le d$, with the help of that $d_2 \le d^2$, $d_3 \le d^3$, $d_4 \le d^5$ follows, in general

$$d_n \le d^{f_n}, \quad f_0 = f_1 = 1, \quad f_{n+1} = f_{n-1} + f_n, \quad n = 1,2,\ldots$$

Here f_n -s are the elements of the familiar Fibonacci sequence, its elements can be given by a closed formula:

$$f_n = \frac{1}{\sqrt{5}}\left[b_1^{n+1} - b_2^{n+1}\right],\ \ b_1 = \frac{1+\sqrt{5}}{2},\ \ b_2 = \frac{1-\sqrt{5}}{2}. \tag{16.15}$$

Observe that $|b_2| < 1$, consequently the increasing powers will tend to zero. Therefore there exists a positive number K such that for all n $d^{s_{n+1}} \le K$, $s_{n+1} = -b_2^{n+1}/\sqrt{5}$. Therefore we may write

$$d_n \le K\left(d^{b_1/\sqrt{5}}\right)^{b_1^n} = K\left(\tilde{d}\right)^{b_1^n},\ \ \tilde{d} = d^{b_1/\sqrt{5}}.$$

We have got as a result that the errors of the secant method can be bounded by a sequence, which has order of convergence $b_1 = \frac{1+\sqrt{5}}{2} \approx 1{,}62$, in other words, it is *superlinear.* ■

16.5. Examples

1. Let $f \in C^3[a,b]$. Apply a parabola interpolation for three points to derive an iterative method to find a local minimum of $f(x)$ in $[a,b]$!

Solution. Let three points in $[a,b]$ be $(x_{i-2}, f_{i-2}), (x_{i-1}, f_{i-1}), (x_i, f_i)$. Newton's interpolation gives $p_2(x) = f_{i-2} + f[x_{i-2}, x_{i-1}](x - x_{i-2}) + f[x_{i-2}, x_{i-1}, x_i](x - x_{i-2})(x - x_{i-1})$. The zero of the derivative has the form:

$$x_{i+1} = \frac{x_{i-2} + x_{i-1}}{2} - \frac{f[x_{i-2}, x_{i-1}]}{2f[x_{i-2}, x_{i-1}, x_i]}. \tag{16.16}$$

2. How should we detect a local minimum of a function when tabulating with uniform stepsize?

Solution. Denote the stepsize by h and the j-th abscissa by $x_j = a + jh,\ j = 0,1,\ldots$. The triple x_{j-1}, x_j, x_{j+1} suffices if $f[x_{j-1}, x_j] < 0$ and $f[x_j, x_{j+1}] > 0$ hold. Then the local minimum can be estimated by the follwing simple formula, if x_j is the midpoint in (16.16):

$$x_{\min} \approx \frac{x_{j-1} + x_{j+1}}{2} - \frac{f[x_{j-1}, x_{j+1}]}{2f[x_{j-1}, x_j, x_{j+1}]} = x_j - \frac{h}{2}\frac{f_{j+1} - f_{j-1}}{f_{j+1} - 2f_j + f_{j-1}}. \tag{16.17}$$

3. Give a convergence theorem for the iteration (16.16) as with the local convergence theorem of the Newton iteration!

Solution. For the sake of simplicity, consider the case $i = 2$ in (16.16). We try to find connection for the subsequent errors by using the properties of divided differences. Subtract the minimal point x^* from both sides and let $\varepsilon_i = x_i - x^*$. This gives

$$\varepsilon_3 = \frac{\varepsilon_0 + \varepsilon_1}{2} - \frac{f[x_0, x_1]}{2f[x_0, x_1, x_2]}.$$

The divided difference in the numerator will be reshaped by using the fact that $f[x^*, x^*] = 0$ holds and the order of base points is arbitrary.: $f[x_0, x_1] = f[x_0, x_1] - f[x_1, x^*] + f[x_1, x^*] - f[x^*, x^*] = \ = \varepsilon_0 f[x_0, x_1, x^*] + \varepsilon_1 f[x_1, x^*, x^*]$. After substituting and bringing to common denominator, we get:

$$\varepsilon_3 = \frac{\varepsilon_0\left(f[x_0,x_1,x_2]-f[x_0,x_1,x^*]\right)+\varepsilon_1\left(f[x_0,x_1,x_2]-f[x_1,x^*,x^*]\right)}{2f[x_0,x_1,x_2]}.$$

The first to terms may be brought into $\varepsilon_0\varepsilon_2 f[x_0,x_1,x_2,x^*]$. The last two terms need a little more work:
$\varepsilon_1\left(f[x_0,x_1,x_2]-f[x_0,x_1,x^*]+f[x_0,x_1,x^*]-f[x_1,x^*,x^*]\right)=\varepsilon_1\varepsilon_2 f[x_0,x_1,x_2,x^*]+\varepsilon_0\varepsilon_1 f[x_0,x_1,x^*,x^*]$
. With these

$$\varepsilon_3 = \frac{\left(\varepsilon_0+\varepsilon_1\right)\varepsilon_2 f[x_0,x_1,x_2,x^*]}{2f[x_0,x_1,x_2]}+\frac{\varepsilon_0\varepsilon_1 f[x_0,x_1,x^*,x^*]}{2f[x_0,x_1,x_2]}.$$

Introduce $\delta_2 = \max\left\{|\varepsilon_0|,|\varepsilon_1|,|\varepsilon_2|\right\}$ and

$$M = \frac{\max\limits_{x\in[a,b]}\left|f^{(3)}(x)\right|}{2\min\limits_{x\in[a,b]}\left|f''(x)\right|}. \tag{16.18}$$

Then we get by using the relation between the divided differences and the corresponding derivatives:

$$|\varepsilon_3| \le \frac{3}{2}\delta_2^2\frac{2!}{3!}2M = \delta_2^2 M. \tag{16.19}$$

This way $|\varepsilon_3|$ is surely less than the absolute maximum of the three previous ε, if $\delta_2 M < 1$ or alternatively $\delta_2 < 1/M$. Consequently, the resulting method surely converges, if the three starting points are in a neighbourhood of the minimum having radius $1/M$.

16.6. Problems

16.1. We have $f \in C^1[a,b]$, $f(a)f(b)<0$ and $f'(x)$ does not have zero in $[a,b]$. Show that $f(x)$ has only one zero there.

16.2. Explain that the equation $\cos x - 4x + 2 = 0,\ x\in\mathbb{R}$ has only one zero by expressing x from the term $4x$ and the obtained iteration is convergent for all starting points. Apply Banach!s fix point theorem!

16.3. From what neighbourhood of the root will the Newton iteration converge in the previous problem?

16.4. Let $x_0 = 0$ and find x_1 by the fix point iteration in Problem 16.2. Estimate the distance of zero from x_1!

16.5. Solve the equation $f(x) = 1/x - a = 0$ with Newton's iteration! For what starting points will convergence take place? An interesting feature of the obtained iteration function is that it has no division. This was important for old computers not having division operation in their arithmetic.

16.6. Solve $f(x) = x^2 - a = 0,\ a > 0$ with Newton's iteration and discuss convergence!

16.7. By using the solution of the previous problem, develop a method for computing $a^{1/k}$, where a is a positive number and $1 < k$.

16.8. Show that Theorem 16.4.3 may be modified such that $f(x_0)f''(x_0)>0$ is replaced by the following condition: after the first step $x_1 \in [a,b]$.

16.9. Show the iteration with $F(x_n)=x_n-\dfrac{\left(f(x_n)\right)^2}{f(x_n)-f(x_n-f(x_n))}$ has second order convergence!

16.10. Check that the Newton method converges linearly for a multiple root!

16.11. Prove that the corrected formula $x_{n+1}=x_n-rf(x_n)/f'(x_n)$ of the Newton method for a zero of multiplicity r has quadratic convergence.

16.12. Elaborate a stopping criterion for the Newton method to reach a given precision of ε.

16.13. What will happen to the secant method, if the conditions of Theorem 16.4.3 are modified such that $\varepsilon_0>0$ and $\varepsilon_1<0$?

16.14. How to stop the secant method for a prescribed accuracy?

17. Solution of nonlinear equations II.

We continue with some special cases.

17.1. The method of bisection

Assume the interval $[a,b]$ has one zero: $f(a)f(b)<0$ and the function is continuous in $[a,b]$. According to the method of bisection, the interval is halved and the subinterval still containing the zero is kept, i.e. where the sign-change still takes place. Then we have the following algorithm:

1. $\exists f \in C[a,b]$, $f(a)f(b)<0$ and $\in$ is the prescribed accuracy.
2. Start: $[a_0,b_0]=[a,b]$, $x_1=(a+b)/2$.
3. $[a_n,b_n]=\begin{cases}[a_{n-1},x_n], & \text{if } f(a_{n-1})f(x_n)<0,\\ [x_n,b_{n-1}], & \text{otherwise}\end{cases}$
4. $x_{n+1}=(a_n+b_n)/2.$
5. Stop: if $f(x_n)=0$ or $|b_n-a_n|<\in$.

The method is not very fast, but it converges safely. Sign-change does not always imply the presence of a zero. As an example, think on the function $1/x$, where the initial interval is $[-1,2]$.

17.1.1 Theorem

The series x_n, $n=1,2,\ldots$ given by the bisection method has a linear order of convergence and we have for the error

$$|\varepsilon_n| \le \frac{b-a}{2^n}, \quad n=0,1,\ldots \tag{17.1}$$

Proof. Convergence follows from the fact that always the halved interval containing zero is kept. In all steps we have for the error:

$$|\varepsilon_{n+1}| \le \frac{1}{2}|\varepsilon_n|$$

and this means linear convergence. ■

17.2. The method of false position (regula falsi)

The difference now from the bisection method is that the division of the interval is not in the middle but it is chosen to be the zero of the line between points $(a_n, f(a_n))$ and $(b_n, f(b_n))$:

$$x_{n+1} = a_n - f(a_n)\frac{b_n-a_n}{f(b_n)-f(a_n)}. \tag{17.2}$$

It can be shown under similar conditions that the convergence is linear such that it is not faster than the bisection method. Yet, it may happen that it is slower. For instance, this

happens in the case when the values of the function are close to the x-axis and the zero is fairly near to one of the endpoints (a or b).

17.3. Newton iteration for functions of many variables

Let $f:\ \mathbb{R}^n \to \mathbb{R}^n$ be a mapping in n-variables, and we are looking for a vector x such that $f(x)=0$. Assuming differentiability, a two-term power series approximation around $x_k \in \mathbb{R}^n$ gives a next estimate x_{k+1} of the solution:

$$f(x_k)+f'(x_k)(x_{k+1}-x_k)=0, \tag{17.3}$$

where $f'(x)=\left[\partial f_i(x)/\partial x_j\right]\in \mathbb{R}^{n\times n}$ is a matrix – the so-called Jacobi matrix - , which is supposed to be invertible. Solving (17.3) for x_{k+1} gives the iteration:

$$x_{k+1}=x_k-\left[f'(x_k)\right]^{-1} f(x_k). \tag{17.4}$$

If there exists a solution, then we may hope convergence if the vectors are sufficiently close to the solution.

This method needs in all steps the computation of derivatives in the Jacobi matrix and the solution of a linear system. To reduce computational work, many times the following simplification is applied: the $f'(x_k)=LU$ decomposition is formed, and then the simpler

$$x_{k+1}=x_k-\left(LU\right)^{-1} f(x_k) \tag{17.5}$$

iteration is done in the course of some steps. This corresponds to the case in one-dimension where the derivatives are not re-computed for a while. Such methods are called quasi-Newton methods.

17.3.1 Broyden's update formula for the Jacobi matrix

To calculate and invert the Jacobi matrix in all steps of iteration is a tedious task. In order to simplify work, Broyden suggested a rank-one update for the Jacobi matrix. To give the idea, consider the two term power series again to approximate the change in the value of the function :

$$f(x_k)\approx f(x_{k-1})+f'(x_{k-1})(x_k-x_{k-1}). \tag{17.6}$$

Introducing the notations $\Delta f_k=f(x_k)-f(x_{k-1}),\ \Delta x_k=x_k-x_{k-1}$ and $J_{k-1}=f'(x_{k-1})$, this reads

$$\Delta f_k \approx J_{k-1}\Delta x_k. \tag{17.7}$$

Now assume that equality holds for the next Jacobian J_k in this equation:

$$\Delta f_k = J_k \Delta x_k \tag{17.8}$$

and there is a rank-one difference between J_k and J_{k-1}. This equation can be rearranged as

$$\left(J_k-J_{k-1}\right)\Delta x_k=\Delta f_k-J_{k-1}\Delta x_k \tag{17.9}$$

and applying the pseudoinverse of the 'special matrix' Δx_k from the right, gives:

$$J_k=J_{k-1}+\left(\Delta f_k-J_{k-1}\Delta x_k\right)\Delta x_k^T/\left\|\Delta x_k\right\|_2^2. \tag{17.10}$$

This rank-one update has minimal 2-norm due to the pseudoinverse, and the two-norm of a rank-one matrix is equal to its Frobenius norm, therefore the modification of the Jacobian is also minimal in Frobenius norm. Gay (1979) proved the statement: when Broyden's method is applied to a linear system of size $n \times n$, it terminates in $2n$ steps.

17.4. Roots of polynomials

To determine roots of polynomials may be needed most frequently for finding the eigenvalues of matrices from the characteristic polynomial. However, it is not suggested that the characteristic polynomials computed, because matrix algorithms offer more reliable methods. In reality, representing polynomials as the sum of powers for a greater value of n

$$p(x) = \sum_{i=0}^{n} a_i x^i \tag{17.11}$$

is not feasible, because the coefficients may not have enough numerical information to compute roots to a desired accuracy. For instance, Wilkinson took the polynomial with roots 1,2,...,19,20 and computed it in the form of (17.11), and then re-computed the roots. The result was so different that he even got complex pairs of roots. The phenomenon can be explained by the connection between roots and coefficients: as an example, the coefficient of the zeroth degree power is the product of the roots, actually it is equal to 20! which cannot be given accurately with 15 decimal digits. So that many important information may be lost because of the machine number representation.

There is a connection between the form in (17.11) and the so-called *Frobenius companion matrix*, which was already seen in Sec. 7.3:

$$F = \begin{bmatrix} 0 & 0 & \dots & \dots & 0 & -a_0/a_n \\ 1 & 0 & \dots & \dots & 0 & -a_1/a_n \\ & 1 & \ddots & & \vdots & \vdots \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & \ddots & 0 & -a_{n-2}/a_n \\ & & & & 1 & -a_{n-1}/a_n \end{bmatrix}. \tag{17.12}$$

If expanded along the last column, it is easy to show: $\det(\lambda I - F) = \frac{1}{a_n}\sum_{i=0}^{n} a_i \lambda^i$. To know about this matrix is useful from at least two points of view. It shows on one hand that the roots of polynomials may be found by linear algebraic algorithms, which are considered among the most reliable methods. On the other hand, it is possible to give a circle in the complex plane that contains all roots of the polynomial:

$$\|F\|_\infty = \max_{0 \le i < n} \left(1 - \delta_{i0} + |a_i / a_n|\right) = R \ge |x_k|,$$

where δ_{ij} is the Kronecker delta. Number R - being a norm of F - is greater than or equal to the spectral radius of F, which is now the maximal absolute value of the polynomial roots.

It is also possible to give another smaller circle such that all roots are outside of this circle. Introduce the change of variable by $x = 1/y$ and determine the polynomial in terms of y. The result is the *reciprocal polynomial*, where the order of the coefficients are reversed and the roots are the reciprocals of the roots of the first polynomial. Taking the row norm of the Frobenius companion matrix of the reciprocal polynomial gives:

$1/|x_k| \le \max_{0<i\le n}\left(1-\delta_{in}+|a_i/a_0|\right)=1/r$, where it was assumed that $a_0 \neq 0$. Combining the two results,

$$r \le |x_k| \le R, \quad k=1,2,\ldots,n, \tag{17.13}$$

we see that the roots can be found in a ring given by the radii r and R.

For polynomials of (17.11), the Newton method can be easily applied, because the function value and derivatives are easy to compute. For example, if we want to compute them at ξ, it will be enough to divide the polynomial by $(x-\xi)^2$ with remainder:

$$p(x)=q(x)(x-\xi)^2+\alpha x+\beta. \tag{17.14}$$

Then $p(\xi)=\alpha\xi+\beta$ and $p'(\xi)=\alpha$.

To identify multiple roots, Euclid's algorithms may be applied. This time the two starting polynomials are $p_0(x)=p(x), \quad p_1(x)=-p'(x)$ and the $i+1$ -th polynomial is determined by dividing $p_{i-1}(x)$ by $p_i(x)$ and the remainder is formed:

$$p_{i-1}(x)=q_i(x)p_i(x)-c_i p_{i+1}(x), \quad i=1,2,\ldots \tag{17.15}$$

The degree of the polynomials in the series are decreasing and $c_i>0$. The algorithm is ended after $m \le n$ steps:

$$p_{m-1}(x)=q_m(x)p_m(x), \quad p_m(x)\neq 0.$$

The last polynomial is the greatest common divisor of the to beginning polynomials. The derivative has all roots with multiplicity higher than 1, therefore these roots should appear in $p_m(x)$.

If all roots are single and real, then the Eulidean algorithm returns a series of polynomials, that is, a *Sturm-sequence*. Let $V(a)$ be the number of sign changes at a, and $V(b)$ at b, then one can show that the interval $[a,b]$ contains $V(a)-V(b)$ number of roots.

17.5. Problems

17.1. Elaborate the algorithm of the division in (17.14)!

17.2. Give the ring that contains the roots of $4x^5-3x^4+6x^3-5x^2-8x+2$!

18. Numerical quadrature I.

The primitive function is not always known when computing integrals. Even if it is known, sometimes it is so complicated that it is almost useless for computations. For such cases numerical quadrature offers an easier alternative for getting a result of desired accuracy. Next we deal with quadrature formulas that can be gained from interpolation.

We have seen at interpolation that a function in the interval $[a,b]$ can be written as:

$$f = L_n + r_n, \tag{18.1}$$

where L_n is the Lagrange interpolant polynomial and r_n is the error term. (We assume the abscissas are ordered: $x_{i-1} < x_i$ and $x_0 = a$, $x_n = b$.) The principle of deriving quadrature formulas can be given by:

$$\int_a^b f = \int_a^b L_n + \int_a^b r_n = \sum_{i=0}^{n} a_i f(x_i) + R_n, \tag{18.2}$$

where the weights of the quadrature

$$a_i = \int_a^b l_i(x)dx \tag{18.3}$$

come from integrating the Lagrange base polynomials.

Corollary. Such formulas are exact for polynomials of order at most n.

For equidistant abscissas we get the Newton-Cotes formulas.

18.1. Closed and open Newton-Cotes quadrature formulas

18.1.1 Definitions

Denote the set of base points by $\Omega_n = \{x_0,\dots,x_n\}$. For a *closed* quadrature formula, interpolation is done such that $a,b \in \Omega_n$, $h=(b-a)/n$, $x_k = a + k\cdot h$, $k=0,1,\dots,n$. In the case of an *open* formula, we have $a,b \notin \Omega_n$, $h=(b-a)/(n+2)$, $x_k = a+(k+1)\cdot h$, $k=0,1,\dots,n$, $x_{-1}=a$, $x_{n+1}=b$.

Next we turn to the derivation of Newton-Cotes formulas. Lagrange base polynomials yield

$$a_k = \int_a^b l_k(x)dx = \int_a^b \frac{\omega_n(x)}{(x-x_k)\omega_n'(x_k)}dx.$$

Observe that: $x_k - x_j = (k-j)h$ and introduce a new variable: $t=(x-a)/h$. From here $x = a+th$, $x-x_j=(t-j)h$ and

$$a_k = \int_0^n \frac{t(t-1)\overbrace{\ldots\ldots\ldots}^{(t-k)\text{ is missing}}(t-n)h^n}{k(k-1)\ldots 1(-1)(-2)\ldots(-n+k)h^n} h dt = \\ = (b-a)\frac{(-1)^{n-k}}{nk!(n-k)!}\int_0^n \frac{t(t-1)\ldots(t-n)}{t-k}dt = (b-a)B_{k,n}^{\text{closed}}, \tag{18.4}$$

The advantage of this form is that the coefficients $B_{k,n}^{\text{closed}}$ need to be computed only once.

One can get the *open Newton-Cotes* weights similarly:

$$a_k = (b-a)\frac{(-1)^{n-k}}{(n+2)k!(n-k)!}\int_0^{n+2} \frac{(t-1)(t-2)\ldots(t-n-1)}{t-k-1}dt = (b-a)B_{k,n}^{\text{open}}, \tag{18.5}$$

Some of the first Newton-Cotes coefficients can be seen in the following table:

Closed					Open			
1	1			Trapezoidal rule	1			Tangent rule
1	4	1		Simpson	1	1		
1	3	3	1		2	-1	2	
7	32	12	32	7	11	1	1	11

In each row the coefficients should be divided by their sum to get the true numbers. In this way the sum of coefficients is equal to 1 so that a constant function is integrated correctly. For instance, the weights 1 4 1 refer to the actual weights 1/6 4/6 1/6 in the case of the Simpson formula.

18.1.2 Theorem

$$1.\ \sum_{k=0}^{n} B_{k,n} = 1, \qquad 2.\ B_{k,n} = B_{n-k,n}. \tag{18.6}$$

Proof. The first statement comes from the quadrature of the constant function $f(x) \equiv 1$, exploiting the fact that it is exact for polynomials of order 0. One can get the second statement by introducing the new variable $y = n - t$.

■

18.2. Some simple quadrature formulas

1. The *tangent formula* (open Newton-Cotes): $n = 0$, $B_{0,0}^{\text{open}} = \frac{1}{2 \cdot 1 \cdot 1}\int_0^2 1 \cdot dt$ and

$$I_0(f) = (b-a) f\left(\frac{a+b}{2}\right). \tag{18.7}$$

This formula can be interpreted as follows: the function in $[a,b]$ is approximated by the tangent line at the midpoint and integration is done to that line. This shows that the formula is exact for at most first order polynomials.

18.2.1 Theorem, error of the tangent formula

Let $c=(a+b)/2,\ f\in C^2[a,b]$*, then one has the relation:*

$$\int_a^b f(x)dx=(b-a)f(c)+\frac{(b-a)^3}{24}f''(\eta),\quad \eta\in[a,b]. \tag{18.8}$$

Proof. The Taylor expansion around c with remainder term is

$$f(x)=f(c)+(x-c)f'(c)+\frac{(x-c)^2}{2}f''(\xi_x).$$

After integration the first term gives the tangent formula, the second term is zero, such that it is enough to consider the error term:

$$R_1(f)=\frac{1}{2}\int_a^b (x-c)^2 f''(\xi_x)dx.$$

Applying the mean value theorem for integrals gives

$$R_1(f)=\frac{f''(\eta)}{2}\int_a^b (x-c)^2dx=\frac{f''(\eta)}{2}\left[\frac{(x-c)^3}{3}\right]_a^b=\frac{(b-a)^3}{24}f''(\eta). \qquad \blacksquare$$

Usually the integration formula (18.8) is not applied for the whole interval $[a,b]$; instead, it is divided into m equal subintervals, where the tangent formula is applied for integration. For instance, if $m=3$: $h=(b-a)/3$ and we apply (18.7) for each subinterval.

In general we sum up the results of the subintervals and arrive at the *composite tangent rule:*

$$\int_a^b f=\frac{b-a}{m}\sum_{i=1}^{m} f(a-h/2+ih)+\frac{(b-a)^3}{24m^2}f''(\eta), \tag{18.9}$$

where $f''(\eta)=\frac{1}{m}(f''(\eta_1)+f''(\eta_2)+\ldots+f''(\eta_m))$, because f'' assumes this average value somewhere in $[a,b]$ (called the Darboux property).

2. The *trapezoidal formula*. It is a closed Newton-Cotes formula for: $n=1,\ B_0^c=B_1^c=1/2$:

$$I_1(f)=\frac{b-a}{2}(f(a)+f(b)). \tag{18.10}$$

18.2.2 Theorem, error of the trapezoidal formula

Let $f\in C^2[a,b]$*, then*

$$\int_a^b f=\frac{b-a}{2}(f(a)+f(b))-\frac{(b-a)^3}{12}f''(\eta),\quad \eta\in[a,b]. \tag{18.11}$$

Proof. The error term comes by integrating the error of the first order interpolating polynomial:

$$R_1(f)=\int_a^b \frac{f''(\xi_x)}{2}(x-a)(x-b)dx=-\int_a^b \frac{f''(\xi_x)}{2}\underbrace{(x-a)(b-x)}_{\geq 0}dx=-\frac{f''(\eta)}{12}(b-a)^3. \qquad \blacksquare$$

Dividing the whole interval into m parts and summing up the contributions of the subintervals lead to the *composite trapezoidal rule*:

$$\int_a^b f = \frac{b-a}{2m}[f(x_0)+2f(x_1)+\ldots+2f(x_{m-1})+f(x_m)]-\frac{(b-a)^3}{12m^2}f''(\eta). \qquad (18.12)$$

18.2.3 Definition

A quadrature formula is said of order k, if a polynomial of degree k is the polynomial of smallest degree for which the formula is not exact.

Then the tangent and trapezoidal formula is of second order.

3. *Simpson formula*: it is a closed formula coming from second degree polynomial interpolation: $n=2$, $B_0^c=1/6$, $B_1^c=4/6$, $B_2^c=1/6$ and

$$I_2(f)=\frac{b-a}{6}(f(a)+4f(\frac{a+b}{2})+f(b)).$$

We have the function $\omega_2(x)=(x-a)(x-\frac{a+b}{2})(x-b)$ in the error expression. The integral of this function in $[a,b]$ is zero. One can show it simply by transforming the origin of the coordinate system into $(a+b)/2$. Then $\omega_2(x)$ is an odd function that has integral zero for the symmetric integration endpoints. Therefore the error will be derived from Hermite's interpolation,

$$f(x)=H_3(x)+\frac{f^{(4)}(\xi_x)}{4!}(x-a)(x-\frac{a+b}{2})^2(x-b), \qquad (18.13)$$

where it is assumed that the first derivative is still given at the midpoint $(a+b)/2$. Having the divided difference scheme in our mind, we know that the interpolating polynomial now has the form: $H_3(x)=L_2(x)+C(x-a)(x-\frac{a+b}{2})(x-b)$. The coefficient of the second term is irrelevant, because the multiplying function has integral zero such that $\int_a^b H_3=\int_a^b L_2$ holds.

18.2.4 Theorem, error of the Simpson formula

Let $f\in C^4[a,b]$. *Then there exists* $\eta\in[a,b]$, *such that*

$$\begin{aligned}\int_a^b f &= \frac{b-a}{6}\left(f(a)+4f(\frac{a+b}{2})+f(b)\right)-\frac{f^{(4)}(\eta)}{90}\left(\frac{b-a}{2}\right)^5 = \\ &= I_2(f)-\frac{f^{(4)}(\eta)}{90}h^5,\end{aligned} \qquad (18.14)$$

where $h=(b-a)/2.$

Proofs. We start from (18.13) of Hermite's interpolation. Integration by terms gives:

$$I(f)-I_2(f)=\int_a^b \frac{f^{(4)}(\xi_x)}{4!}(x-a)(x-\frac{a+b}{2})^2(x-b)dx.$$

We may apply the mean value theorem of integration if the term beside $f^{(4)}$ is non-negative. This can be assured if we take $(b-x)$ instead of $(x-b)$:

$$I(f)-I_2(f)=-\frac{f^{(4)}(\eta)}{4!}\int_a^b (x-a)(x-\frac{a+b}{2})^2(b-x)dx=-\frac{f^{(4)}(\eta)}{90}\left(\frac{b-a}{2}\right)^5. \qquad \blacksquare$$

Composite Simpson rule. Now the interval $[a,b]$ is divided into an even number of m subintervals and the Simpson formula is applied to the neighbouring interval pairs, the result is the composite Simpson rule. Then the composite formula can be given with point triplets as:

$$\begin{aligned}\int_a^b f &= \sum_{k=1,3,\dots}\left(\frac{2h}{6}(f(x_{k-1})+4f(x_k)+f(x_{k+1}))-\frac{f^{(4)}(\eta_k)}{90}h^5\right)= \\ &=\frac{h}{3}\left(f(x_0)+2\sum_{\substack{k\ \text{páros}\\ \text{belső pont}}} f(x_k)+4\sum_{k\ \text{ptlan}} f(x_k)+f(x_m)\right)-\frac{h^5}{90}\sum_{k\ \text{ptlan}} f^{(4)}(\eta_k).\end{aligned} \tag{18.15}$$

The error term can still be simplified:

$$-\frac{h^5}{90}\sum_{k\ \text{ptlan}} f^{(4)}(\eta_k)=-\frac{(b-a)^5}{180m^4}\left(\frac{\sum f^{(4)}(\eta_k)}{m/2}\right)=-\frac{(b-a)^5}{180m^4}f^{(4)}(\eta), \tag{18.16}$$

because due to the Darboux property there exists an η such that the average is equal to $f^{(4)}(\eta)$.

18.3. Examples

1) We approximate the integral $\int_{-1}^{1}\frac{dx}{2+x}$ with the composite tangent rule. How many subintervals should be chosen such that the error be not greater than 10^{-2}?

Solution. We have to ensure $\frac{(b-a)^3}{24m^2}M_2\le 10^{-2}$, where $b-a=2$ and $M_2=2\max_{x\in[-1,1]}\left|(2+x)^{-3}\right|=2$. Substituting numbers gives $200/3\le m^2$, such that $m=9$ is satisfactory.

2) Give parameters A_0, A_1, A_2 such that the rule $\int_0^2\sqrt{x}f(x)dx\approx I_2(f)=A_0f(0)+A_1f(1)+A_2f(2)$ be exact for at most second degree polynomials!

Solution. There are two sraigthforward approaches. One is to compute weights with the Lagrange base polynomials: $A_i=\int_0^2 l_i(x)\sqrt{x}dx$, where $\sqrt{x}$ can be considered a weight function. The second approach is to set up a system of linear equations according to accuracy demands. The first line below states that the quadrature is exact for the constant polynomial 1, the second line expresses exactnes for polynomial x and the third line is for function x^2:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} \int_0^2 x^{1/2} dx = 4\sqrt{2}/3 \\ \int_0^2 x \cdot x^{1/2} dx = 8\sqrt{2}/5 \\ \int_0^2 x^2 x^{1/2} dx = 16\sqrt{2}/7 \end{bmatrix}.$$

It has the solution: $A_0 = \frac{8\sqrt{2}}{105}$, $A_1 = \frac{32\sqrt{2}}{35}$, $A_2 = \frac{12\sqrt{2}}{35}$.

18.4. Problems

18.1. How many subintervals are needed for computing the integral $\int_0^3 dx/(1+2x)$ with the trapezoidal rule such that the error should not be greater than 0.18?

18.2. The integral $\int_0^2 dx/(1+x)^2$ is approximated by Simpson's rule. How many subintervals of $[0,2]$ is needed for getting an error less than 10^{-3}?

18.3. We compute the integral $\int_0^2 \frac{dx}{3x+2}$ with the tangent rule. How many subintervals are needed for an error not greater than $3.\times 10^{-2}$?

19. Gaussian quadratures

We have seen that a quadrature formula coming from a polynomial of order n is exact for polynomials having degree not greater than n. The Gaussian quadrature formulas are based on the observation that choosing the base points specifically may increase the order of the quadrature formulas. We shall need orthogonal polynomials here. The scalar product of functions f and g is given by

$$(f,g)=\int_a^b w(x)f(x)g(x)dx,$$

where $w(x)$ is the weight function and integrability is assumed. More on orthogonal polynomials can be found in Sec. 9.

19.1. Theorem on the roots of orthogonal polynomials

Let $\{p_k(x)\}$ be a set of orthogonal polynomials. Then for any n the roots of $p_{n+1}(x)$ are real, they have multiplicity 1 and they are in the interval $[a,b]$, where $[a,b]$ is the domain of integration for the scalar product.

Proof. Assume indirectly that the multiplicity may be even greater than 1 and let $x_0,x_1,\dots,x_k$ be the roots of $p_{n+1}(x)$ in $[a,b]$ having odd multiplicity, that is, $p_{n+1}(x)$ changes sign there. If k – the greatest index of such roots – is equal to n, then the theorem is true. If not, then assume $k<n$ and this leads to contradiction. Consider the $(k+1)$-st degree polynomial $q(x)=(x-x_0)(x-x_1)\dots(x-x_k)$. As the inequality $k+1<n+1$ holds, the scalar product (p_{n+1},q) should be zero because of orthogonality, (see the Corollary of Theorem 9.1.2). But the polynomial $p_{n+1}(x)q(x)$ does not change sign in $[a,b]$, because all sign changes of p_{n+1} are extinguished by $q(x)$ such that $(p_{n+1},q)=\int wp_{n+1}q>0$ should follow, and that is contradiction. ■

The Gaussian quadrature formula on $n+1$ base points is derived from interpolation on the roots of $p_{n+1}(x)$ and integrate:

$$\int_a^b wf=\int_a^b wL_n+\int_a^b wr_n=\sum_{i=0}^n a_i f(x_i)+R_n,\quad a_i=\int_a^b l_i(x)w(x)dx. \tag{19.1}$$

19.2. Theorem on the order of the Gaussian quadrature

Let the roots of $p_{n+1}(x)$ be $x_0,x_1,\dots,x_n$, $a_i=\int l_i w$, where l_i is the i-th Lagrange base polynomial belonging to the given base points. Then the Gaussian quadrature formula

$$G_n(f)=\sum_{i=0}^n a_i f(x_i)$$

is exact for all polynomials of degree not greater than $2n+1$: $f\in\mathcal{P}_{2n+1}\to\int f\alpha=G_n(f)$.

Proof. $G_n(f)$ is also an interpolatory quadrature, therefore it is surely accurate for all polynomials of degree n at most. Now assume that $f \in \mathcal{P}_{2n+1}$, $f = p_{n+1} \cdot q + r$, $q, r \in \mathcal{P}_n$ and thus

$$G_n(f) = \sum_{i=0}^{n} a_i f(x_i) = \sum_{i=0}^{n} a_i [\underbrace{p_{n+1}(x_i)}_{=0 \text{ for all } i} \cdot q(x_i) + r(x_i)] =$$

$$= \sum_{i=0}^{n} a_i r(x_i) = G_n(r) = \int rw = \qquad \text{(because it is accurate up to order } n\text{)}$$

$$= \int (p_{n+1} \cdot q + r) w = \qquad \text{(because } q \in \mathcal{P}_n \text{ and thus orthogonal to } p_{n+1}\text{)}$$

$$= \int fw. \qquad \blacksquare$$

19.2.1 Corollary

The weights a_i of the Gaussian quadrature are positive.

Proof. For Lagrange base polynomials we have $l_i(x_j) = l_i^2(x_j) = \delta_{ij}$, where δ_{ij} is the Kronecker-delta; moreover, we have $l_i^2(x) \geq 0$ and $l_i^2(x) \in \mathcal{P}_{2n}$, consequently the Gaussian quadrature is accurate:

$$0 < \int l_i^2 w = G_n(l_i^2) = \sum_{j=0}^{n} a_j l_i^2(x_j) = a_i. \qquad \blacksquare$$

Integrating the function $f(x) = 1$ gives the sum of the weights:

$$\sum_{i=0}^{n} a_i = \int w = \mu_0, \quad \text{where } \mu_i = \int x^i w,$$

μ_0 being the zeroth moment. Observe that it is equal to $b - a$, if the weight function is $w(x) = 1$.

19.2.2 Theorem, error of Gaussian quadrature

Let $f \in C^{2n+2}[a,b]$ and $G_n(f) = \sum_{k=0}^{n} a_k f(x_k)$, and the base points are the roots of $p_{n+1}(x)$. Then

$$I(f) - G_n(f) = \frac{f^{(2n+2)}(\eta)}{(2n+2)!} \left(p_{n+1}, p_{n+1} \right), \tag{19.2}$$

where $p_{n+1}(x)$ is a monic orthogonal polynomial.

Proof. Applying the Hermite-Fejér interpolation (function values and first derivatives are used in the interpolation), we have the following error formula:

$$f(x) = H_{2n+1}(x) + \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} \underbrace{(x - x_0)^2 (x - x_1)^2 \ldots (x - x_n)^2}_{= p_{n+1}^2(x)}.$$

By applying the mean value theorem of integrals, we get the statement from

$$I(f)-G_n(f)=\int_a^b \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!}\underbrace{p_{n+1}^2(x)}_{\geq 0}w(x)dx,$$

if taking into account the fact that the Gaussian quadrature is accurate for $H_{2n+1}(x)$. ■

The following table contains data of some frequently used monic orthogonal polynomials:

Name	$[a,b]$	$w(x)$	μ_0	α_{n+1}	β_n	p_0	p_1	p_2
Legendre	$[-1,1]$	1	2	0	$n^2/(4n^2-1)$	1	x	$x^2-1/3$
Chebyshev	$[-1,1]$	$\frac{1}{\sqrt{1-x^2}}$	π	0	1/4, but $\beta_1=1/2$	1	x	$x^2-1/2$
Laguerre	$[0,\infty]$	e^{-x}	1	$2n+1$	n^2	1	$x-1$	x^2-4x+2
Hermite	$[-\infty,\infty]$	e^{-x^2}	$\sqrt{\pi}$	0	$n/2$	1	x	$x^2-1/2$

19.3. Examples

1. The integral $\int_{-1}^{1}\left(1-x^2\right)^{-1/2}e^{-x}dx$ is approximated by the three-point Gauss-Chebyshev quadrature. Estimate the error!

Solution. For the three-point quadrature $n=2$, which will be applied for (19.2): $M_6=e$ and we should have a monic orthogonal polynomial, thus $p_3(x)=T_3(x)/4$. The result is that the error is not greater than $e\cdot(T_3,T_3)/(16\cdot 6!)=e\pi/(32\cdot 720)$, because $(T_3,T_3)=\pi/2$ (see Problem 7.4).

2. Give the weights of the two-point Gauss-Hermite quadrature! Check that the resulting quadrature formula is accurate up to third degree polynomials!

Solution. For the two-point quadrature, the roots of the second degree Hermite polynomial are $-x_0=x_1=2^{-1/2}$. Hence the integral of the first Lagrange base polynomial is given by $a_0=\int_{-\infty}^{\infty}\frac{(x-x_1)}{(x_0-x_1)}\exp(-x^2)dx=\frac{x_1\mu_0}{2x_1}=\mu_0/2$, because the integral of the first order term is zero as being an odd function. One gets similarly $a_1=a_0$. The resulting quadrature is accurate for the function 1, because the result is μ_0. It is also accurate for functions x and x^3, because they are odd and the result 0 follows for both integrals and quadrature rules. Still it remained to show that the formula is accurate for the second degree polynomial x^2. Its integral can be expressed by (9.8) as $(p_1,p_1)=\beta_1(p_0,p_0)$, and this is equal to $\mu_0/2$ when using the given table. The Gauss-Hermite quadrature yields $\frac{\mu_0}{2}(\frac{1}{2}+\frac{1}{2})$, such that equality holds.

3. Find the value of the following integral:

$$\int_{-1}^{1}\frac{(2x^2+x)dx}{\sqrt{1-x^2}}.$$

Solution 1. We expand the polynomial in the numerator as the linear combination of the first three Chebyshev polynomials: $2x^2 + x = c_0T_0 + c_1T_1 + c_2T_2$. Then the integral can be written as: $(T_0, c_0T_0 + c_1T_1 + c_2T_2) = c_0(T_0, T_0) = c_0\pi$. Using the coefficients of the Chebyshev polynomials, the following linear system of equations can be set up due to (9.5):

$$\begin{bmatrix} 1 & 0 & -1 \\ & 1 & 0 \\ & & 2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix},$$

from where $c_0 = 1$, therefore the value of the integral is π.

Solution 2. Observe that the integral is expressible as $(T_1, 2T_1 + T_0) = 2(T_1, T_1) = 2(\pi/2) = \pi$.

19.4. Problems

19.1. Compute with Hermite-polynomials:

$$\int_{-\infty}^{\infty} e^{-x^2}(2x^2 - x)dx = ?$$

$$H_0 = 1,\ \ H_1 = x,\ \ H_{n+1} = xH_n - (n/2)H_{n-1},\ \ (H_0, H_0) = \sqrt{\pi}.$$

19.2. Give the error formula for the two-point Gauss-Hermite integral!

19.3. $\int_{-1}^{1} \frac{4x^3 + 4x^2 - 3x - 1}{\sqrt{1 - x^2}} dx = ?$ $\{T_{n+1} = 2xT_n - T_{n-1},\ \ (T_n, T_m) = (\delta_{0,m} + \delta_{n,m})\pi/2\}$.

19.4. The tangent formula can be thought of as a Gauss-Legendre quadrature with a polynomial of first degree. Show that the error of the tangent rule can be derived from the error of the Gauss-Legendre quadrature! (Choose $[a,b] = [-1,1]$).

LITERATURE

Antoulas, T. and Slavinsky, JP: (2003) Partial fraction expansion, OpenStax-CNX module: m211, http://cnx.org/content/m2111/2.14/

Berrut, J-P. and Trefethen, L. N. (2004) *Barycentric Lagrange interpolation,* SIAM Review, Vol. 46, No. 3. 501-517.

Gay, D. M. (1979) *Some convergence properties of Broyden's method*, SIAM J. of Numer. Anal. (SIAM) **16** (4) 623-630.

Gergó, L. (2010) *Numerical Methods*, (in Hungarian), Eötvös Kiadó, ELTE, Budapest.

Higham, N. J. (2004) *The numerical stability of barycentric Lagrange interpolation*, IMA J. Numer. Anal., **24,** 547-556.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992) *Numerical Recipes in Fortran 77*, Cambridge Univ. Press, Cambridge.

Rutishauser, H. (1976) *Vorlesungen über numerische Mathematik*, Vol. 1, Birkhäuser, Basel, Stuttgart; English translation, *Lectures on Numerical Mathematics*, Walter Gautschi, ed., Birkhäuser, Boston, 1990.

Sadiq, B. and Viswanath, D. (2013) *Barycentric Hermite interpolation,* SIAM J. Sci. Comput. **36** (3) A1254-A1270

Stoyan, G. and Takó, G. (2002) *Numerical Methods,* 2nd ed. (in Hungarian), Typotex Könykiadó, Budapest.