

LECTURES

LESSON VII

7. Finite Difference Calculus. Interpolation of Functions

7.0. Introduction

This lesson is devoted to one of the most important areas of theory of approximation - interpolation of functions. In addition to theoretical importance in construction of numerical methods for solving a lot of problems like numerical differentiation, numerical integration and similar, it has practical application to many engineering problems, including FEM problems.

Theory of approximation deals with replacing of function f defined on some set X by another function Φ . Let function Φ depend on $n + 1$ parameter a_0, a_1, \dots, a_n , i.e.

$$\Phi(x) = \Phi(x; a_0, a_1, \dots, a_n).$$

Problem of approximation of function f by function Φ reduces to determination of parameters $a_i, i = 1, \dots, n$ according to some criterion. Depending on chosen criterion, we differ several sorts of approximations. Generally, depending on form of approximation function, they can be divided to linear and nonlinear approximation functions. The general form of linear approximation function is

$$(7.0.1) \quad \Phi(x) = a_0\Phi_0(x) + a_1\Phi_1(x) + \dots + a_n\Phi_n(x),$$

whereby system of functions $\{\Phi_k\}$ fulfills some given conditions. Linearity of function Φ means linearity regarding parameters a_i ($i = 0, 1, \dots, n$). When $\Phi_k = x^k$ ($k = 0, 1, \dots, n$), i.e.

$$\Phi(x) = a_0 + a_1x + \dots + a_nx^n,$$

we have approximation by algebraic polynomials. In the case when $\{\Phi_k\} = \{1, \cos x, \sin x, \cos 2x, \sin 2x, \dots\}$ we have approximation by trigonometric polynomials, i.e. trigonometric approximation. For the case

$$\Phi(k) = (x - x_k)_+^m = \begin{cases} (x - x_k)^m & (x \geq x_k), \\ 0 & (x < x_k), \end{cases}$$

where m is fixed natural number, we have spline approximation.

We will mention two of nonlinear approximations:

1. Exponential approximation

$$\Phi(x) = \Phi(x; c_0, b_0, \dots, c_r, b_r) = c_0e^{b_0x} + \dots + c_re^{b_rx},$$

where $n + 1 = 2(r + 1)$, i.e. $n = 2r + 1$.

2. Rational approximation

$$\Phi(x) = \Phi(x; b_0, \dots, b_r, c_0, \dots, c_s) = \frac{b_0 + b_1x + \dots + b_r x^r}{c_0 + c_1x + \dots + c_s x^s},$$

where $n = r + s + 1$.

Let function f be given on segment $[a, b]$ by set of pairs (x_k, f_k) ($k = 0, 1, \dots, n$), where $f_k = f(x_k)$. If for approximation of function f by function Φ the criterion for choice of parameters a_0, a_1, \dots, a_n is given by system of equations

$$(7.0.2) \quad \Phi(x_k; a_0, a_1, \dots, a_n) = f_k \quad (k = 0, 1, \dots, n),$$

we have problem of function interpolation. Function Φ is called in this case interpolation function and points x_k ($k = 0, 1, \dots, n$) interpolation nodes.

Problem of interpolation could be more complicated then noted. More general case appears when, in addition to function values in interpolation nodes, the derivatives of function are also included.

7.1. Chebyshev systems

Let function f be given by its values $f_k \equiv f(x_k)$ in points x_k ($x_k \in [a, b]$) ($k = 0, 1, \dots, n$). If we have linear interpolation problem, i.e. interpolation by function (7.0.1), system of equation (7.0.2) reduces to system of linear equations in parameters a_i ($i = 0, 1, \dots, n$),

$$a_0\Phi_0(x_k) + a_1\Phi_1(x_k) + \dots + a_n\Phi_n(x_k) = f_k \quad (k = 0, 1, \dots, n),$$

i.e.

$$(7.1.1) \quad \begin{bmatrix} \Phi_0(x_0) & \Phi_1(x_0) & \dots & \Phi_n(x_0) \\ \Phi_0(x_1) & \Phi_1(x_1) & \dots & \Phi_n(x_1) \\ \vdots & & & \\ \Phi_0(x_n) & \Phi_1(x_n) & \dots & \Phi_n(x_n) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}.$$

In order above given interpolation problem to have unique solution it is necessary that matrix of system (7.1.1) be regular.

To the system of functions (Φ_k) should be intruded such conditions under which there not exists linear combination

$$a_0\Phi_0(x) + a_1\Phi_1(x) + \dots + a_n\Phi_n(x)$$

which has $n + 1$ different zeros on $[a, b]$. System of functions with such characteristic are called Chebyshev (Tchebyshev) systems, or T-systems. There exists extraordinary monograph regarding T-systems [6].

Theorem 7.1.1. *If the functions $\Phi_k : [a, b] \rightarrow R$ ($k = 0, 1, \dots, n$) are $n + 1$ times differentiable and if for every $k = 0, 1, \dots, n$ the Wronsky determinant W_k is different from zero, i.e.*

$$W_k = \begin{vmatrix} \Phi_0(x) & \Phi_1(x) & \dots & \Phi_k(x) \\ \Phi_0'(x) & \Phi_1'(x) & \dots & \Phi_k'(x) \\ \vdots & & & \\ \Phi_0^{(k)}(x) & \Phi_1^{(k)}(x) & \dots & \Phi_k^{(k)}(x) \end{vmatrix} \neq 0,$$

system of functions $\{\Phi_k\}$ is Chebyshev (T) system.

7.2. Lagrange interpolation

Let function f be given by its values $f_k \equiv f(x_k)$ in points x_k ($x_k \in [a, b]$) ($k = 0, 1, \dots, n$). Without decreasing the generality, assume

$$(7.2.1) \quad a \leq x_0 < x_1 < \dots < x_n \leq b.$$

If we take points x_k for interpolation knots and put $\Phi_k(x) = x^k$ ($k = 0, 1, \dots, n$) we have a problem of interpolation of function f by algebraic polynomial. Denote this polynomial with P_n , i.e.

$$P_n(x) = \Phi(x) = a_0 + a_1x + \dots + a_nx^n.$$

Then we have the interpolating polynomial

$$(7.2.2) \quad P_n(x) = \sum_{k=0}^n f(x_k)L_k(x),$$

where

$$\begin{aligned} L_k(x) &= \frac{(x-x_0)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)} \\ &= \frac{\omega(x)}{(x-x_k)\omega'(x_k)}, \\ \omega(x) &= (x-x_0)(x-x_1)\dots(x-x_n), \\ \omega'(x) &= (x_k-x_0)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n). \end{aligned}$$

The formula (7.2.2) is called Lagrange interpolation formula, and polynomial P_n Lagrange interpolation polynomial. When programming, it is suggested to use the following form of Lagrange formula.

$$P_n(x) = \sum_{k=0}^n \left(f(x_k) \prod_{\substack{i=0 \\ i \neq k}}^n \frac{x-x_i}{x_k-x_i} \right).$$

Having in mind that determinant of system (7.1.1) is Vandermonde determinant, i.e.

$$\begin{vmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^n \end{vmatrix} = \prod_{i>j} (x_i - x_j),$$

and the assumption (7.2.1), it follows that the Lagrange polynomial (7.2.2) is unique.

Example 7.2.1. For function values given in tabular form find the Lagrange interpolation polynomial.

x_k	$f(x_k)$
-1	-1.
0	2.
2	10.
3	35.

$$\begin{aligned} P_3(x) &= (-1) \frac{(x-0)(x-2)(x-3)}{(-1-0)(-1-2)(-1-3)} + 2 \frac{(x+1)(x-2)(x-3)}{(0+1)(0-2)(0-3)} \\ &\quad + 10 \frac{(x+1)(x-0)(x-3)}{(2+1)(2-0)(2-3)} + 35 \frac{(x+1)(x-0)(x-2)}{(3+1)(3-0)(3-2)}, \end{aligned}$$

i.e.

$$P_3(x) = \frac{5}{3}x^3 - \frac{4}{3}x^2 + 2.$$

When the general expression for interpolation polynomial is not needed, but only value for some distinct x , the Aitken's scheme is to be used.

Aitken's scheme is of form:

$$\begin{aligned} \mathbf{A}_k &= f(x_k) \quad (k = 0, 1, \dots, n), \\ \mathbf{A}_{k-1,k} &= \frac{1}{x_k - x_{k-1}} \left| \begin{array}{cc} \mathbf{A}_{k-1} & x_{k-1} - x \\ \mathbf{A}_k & x_k - x \end{array} \right| \quad (k = 1, \dots, n), \\ \mathbf{A}_{0,1,\dots,n} &= \frac{1}{x_n - x_0} \left| \begin{array}{cc} \mathbf{A}_{0,1,\dots,n-1} & x_0 - x \\ \mathbf{A}_{1,2,\dots,n} & x_n - x \end{array} \right|, \end{aligned}$$

where $P_n(x) = \mathbf{A}_{0,1,\dots,n}$.

Example 7.2.2. Determine approximately $f(1)$ using data from the following table (see [1], pp. 20-21)

x	14	17	31	35
$f(x)$	68.7	64.0	44.0	39.1

by Aitken's scheme.

Following above given procedure, one gets

$$\begin{aligned} \mathbf{A}_{0,1} &= \frac{1}{17 - 14} \left| \begin{array}{cc} 68.7 & 14 - 27 \\ 64.0 & 17 - 27 \end{array} \right| = 48.33, \\ \mathbf{A}_{1,2} &= \frac{1}{31 - 17} \left| \begin{array}{cc} 64.0 & 17 - 27 \\ 44.0 & 31 - 27 \end{array} \right| = 49.72, \\ \mathbf{A}_{2,3} &= \frac{1}{35 - 31} \left| \begin{array}{cc} 44.0 & 31 - 27 \\ 39.1 & 35 - 27 \end{array} \right| = 48.90, \\ \mathbf{A}_{0,1,2} &= \frac{1}{31 - 14} \left| \begin{array}{cc} 48.33 & 14 - 27 \\ 49.72 & 31 - 27 \end{array} \right| = 49.39, \\ \mathbf{A}_{1,2,3} &= \frac{1}{35 - 17} \left| \begin{array}{cc} 49.72 & 17 - 27 \\ 48.90 & 35 - 27 \end{array} \right| = 49.26, \end{aligned}$$

and, finally,

$$f(27) \cong P_3(27) = \mathbf{A}_{0,1,2,3} = \frac{1}{35 - 14} \left| \begin{array}{cc} 49.39 & 14 - 27 \\ 49.26 & 35 - 27 \end{array} \right| = 49.31$$

There is often problem to determine value of argument for given function value. This problem is usually to be solved using methods of inverse interpolation. If the given function is monotone, the simplest way to apply inverse interpolation is to switch values of function and argument, and then to construct interpolation polynomial. Note that for inverse interpolation it is convenient to use Aitken's interpolation method.

Example 7.2.3. Determine approximately zero of function given in example 7.2.1.

Lagrange interpolation polynomial for function $y \rightarrow f^{-1}(y)$ is

$$\begin{aligned} P_3(y) &= (-1) \frac{(y-2)(y-10)(y-35)}{(-1-2)(-1-10)(-1-35)} + 0 \frac{(y+1)(y-10)(y-35)}{(2+1)(2-10)(2-35)} \\ &\quad + 2 \frac{(y+1)(y-2)(y-35)}{(10+1)(10-2)(10-35)} + 3 \frac{(y+1)(y-2)(y-10)}{(35+1)(35-2)(35-10)}, \end{aligned}$$

wherefrom, for $y = 0$ we get null of function f

$$x \cong P_3(0) = -0.6508$$

7.3. Newton interpolation with divided differences

For function f given by its values $f_k \equiv f(x_k)$ in points x_k ($k = 0, 1, \dots, n$), define first divided differences. The ratio

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

is called divided difference of first order (of function f in points x_0 and x_1) and denoted as $[x_0, x_1; f]$.

Divided difference of order r are defined recursively by

$$(7.3.1) \quad [x_0, x_1, \dots, x_r; f] = \frac{[x_1, \dots, x_r; f] - [x_0, \dots, x_{r-1}; f]}{x_r - x_0},$$

where $[x; f] \equiv f(x)$.

Relation (7.3.1) enables construction of table of divided differences

k	x_k	f_k	Δf_k	$\Delta^2 f_k$	$\Delta^3 f_k$
0	x_0	f_0			
			$[x_0, x_1; f]$		
1	x_1	f_1		$[x_0, x_1, x_2; f]$	
			$[x_1, x_2; f]$		$[x_0, x_1, x_2, x_3; f]$
2	x_2	f_2		$[x_1, x_2, x_3; f]$	
			$[x_2, x_3; f]$		
3	x_3	f_3			
	\vdots	\vdots			

One can show that divided difference of order r has characteristic of linearity, i.e.

$$[x_0, x_1, \dots, x_r; c_1 f + c_2 g] = c_1 [x_0, \dots, x_r; f] + c_2 [x_0, \dots, x_r; g],$$

where c_1 and c_2 are arbitrary constants. Because of, based on (7.3.1),

$$[x_0, x_1; f] = \frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0},$$

one can prove by mathematical induction

$$[x_0, x_1, \dots, x_r; f] = \sum_{i=0}^r \frac{f(x_i)}{\omega'(x_i)},$$

where $\omega(x) = (x - x_0)(x - x_1) \dots (x - x_r)$.

Let $f \in C^n[a, b]$ and the condition (7.2.1) holds. Then, for every $r \leq n$, the formula

$$[x_0, x_1, \dots, x_r; f] = \int_0^1 \int_0^{t_1} \dots \int_0^{t_{r-1}} f^{(r)}(x_0 + \sum_{i=1}^r (x - x_{i-1})t_i) dt_1 dt_2 \dots dt_r$$

holds. This can be proved by mathematical induction.

Applying theorem on integral mean value, from last expression follows

$$\begin{aligned} [x_0, x_1, \dots, x_r; f] &= f^{(r)}(\xi) \int_0^1 \int_0^{t_1} \dots \int_0^{t_{r-1}} dt_1 dt_2 \dots dt_r \\ &= \frac{1}{r!} f^{(r)}(\xi) \quad (a < \xi < b). \end{aligned}$$

Taking $x_i \rightarrow x_0$ ($i = 1, \dots, r$) in last equality, we get

$$[x_0, x_1, \dots, x_r; f] \rightarrow \frac{1}{r!} f^{(r)}(x_0).$$

Let us express now value of function $f(x_r)$ ($r \leq n$) by means of divided differences $[x_0, \dots, x_i; f]$ ($i = 0, 1, \dots, r$).

For $r = 1$, based on definition (7.3.1), we have

$$f(x_1) = f(x_0) + (x_1 - x_0)[x_0, x_1; f].$$

In similar way, for $r = 2$,

$$\begin{aligned} f(x_2) &= f(x_1) + (x_2 - x_1)[x_1, x_2; f] \\ &= (f(x_0) + (x_1 - x_0)[x_0, x_1; f]) + (x_2 - x_1)[x_0, x_1; f] \\ &\quad + (x_2 - x_0)[x_0, x_1, x_2; f], \end{aligned}$$

i.e.

$$f(x_2) = f(x_0) + (x_2 - x_0)[x_0, x_1; f] + (x_2 - x_0)(x_2 - x_1)[x_0, x_1, x_2; f].$$

In general case, it holds

$$\begin{aligned} f(x_r) &= f(x_0) + (x_r - x_0)[x_0, x_1; f] + (x_r - x_0)(x_r - x_1)[x_0, x_1, x_2; f] \\ &\quad + \dots + (x_r - x_0)(x_r - x_1) \dots (x_r - x_{r-1})[x_0, x_1, \dots, x_r; f]. \end{aligned}$$

Using divided differences for set of data $(x_k, f(x_k))$ ($k = 0, \dots, n$) the interpolation polynomial of the following form can be constructed.

$$\begin{aligned} P_n(x) &= f(x_0) + (x - x_0)[x_0, x_1; f] + (x - x_0)(x - x_1)[x_0, x_1, x_2; f] \\ &\quad + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})[x_0, x_1, \dots, x_n; f]. \end{aligned}$$

This polynomial is called Newton's interpolation polynomial.

Having in mind uniqueness of algebraic interpolation polynomial, we conclude that Newton's interpolation polynomial is equivalent to Lagrange polynomial. Note that construction of Newton's interpolation polynomial demands previous forming of table of divided differences, what was not the case with Lagrange polynomial. On the other hand, involving a new interpolation node in order to reduce interpolation error, is more convenient with Newton's polynomial, because do not demand repeating of whole calculation. Namely, at Newton's interpolation we have

$$P_{n+1}(x) = P_n(x) + (x - x_0)(x - x_1) \dots (x - x_n)[x_0, x_1, \dots, x_{n+1}; f].$$

If we put $x_i \rightarrow x_0$ in Newton's interpolation polynomial P_n , based on (7.3.2) it reduces to Taylor polynomial.

Example 7.3.1. Based on table of values of function $x \rightarrow \ln x$ form table of divided differences and write Newton's interpolation polynomial.

k	0	1	2	3
x_k	0.0	0.2	0.5	1.0
$f(x_k)$	1.0000000	1.0200668	1.1276260	1.5430806

k	$[x_k, x_{k+1}; f]$	$[x_k, x_{k+1}, x_{k+2}; f]$	$[x_k, x_{k+1}, x_{k+2}, x_{k+3}; f]$
0	<u>0.1003338</u>		
1	0.3585307	<u>0.5163938</u>	
2	0.8309093	0.5904733	<u>0.0740795</u>
3			

Newton's interpolation polynomial is then

$$P_3(x) = 1. + 0.1003338x + 0.5163938x(x - 0.2) + 0.0740795x(x - 0.2)(x - 0.5).$$

For example, for $x = 0.3$, $ch\ 0.3 \cong P_3(0.3) = 1.0451474$.

7.4. Finite difference calculus

Operator of finite difference on the function set is defined as

$$(7.4.1) \quad \Delta f(x) = f(x + h) - f(x) \quad (h = \text{const.} > 0)$$

and on the series set

$$\Delta y_k = y_{k+1} - y_k.$$

If we want to stress explicitly the step h in definition (7.4.1), then we denote in operator Δh , i.e. Δ_h .

The iterated operator, or degree of operator Δ is denoted as

$$\begin{aligned} \Delta^k f(x) &= \Delta^{k-1} f(x + h) - \Delta^{k-1} f(x) \quad (k \in N), \\ \Delta^0 f(x) &= f(x). \end{aligned}$$

Based on given definition, by means of mathematical induction, it is easy to prove the formulas

$$(7.4.2) \quad \Delta^k = \sum_{i=0}^k (-1)^i \binom{k}{i} f(x + (k - i)h),$$

$$(7.4.3) \quad f(x + kh) = \sum_{i=0}^k \binom{k}{i} \Delta^i f(x).$$

In addition to operator Δ , there are some additional standard operators, known as finite difference operators. They enable formal calculus based on rules of algebra and analysis, being a basis for forming different approximation formulas. This calculus is called **finite difference calculus**.

On the set of continuous functions we will define operators:

1. $E f(x) = f(x + h)$ (shift operator)
2. $\Delta f(x) = f(x + h) - f(x)$ (forward difference operator)
3. $\nabla f(x) = f(x) - f(x - h)$ (backward difference operator)
4. $\delta f(x) = f(x + \frac{h}{2}) - f(x - \frac{h}{2})$ (central difference operator)
5. $\mu f(x) = \frac{1}{2}(f(x + \frac{h}{2}) + f(x - \frac{h}{2}))$ (averaging operator)
6. $l f(x) = f(x)$ (identity operator)
7. $J f(x) = \int f(t) dt$ (integration operator)
8. $D f(x) = f'(x)$ (differentiation operator)

The last one is defined on the set of differentiable functions.

If we denote with L the set of previously defined operators,

$$L = \{E, \Delta, \nabla, \delta, \mu, l, J, D\},$$

then we have the following properties:

- a) Every operator $A \in L$ is linear, i.e.

$$A(\lambda_1 f(x) + \lambda_2 g(x)) = \lambda_1 A f(x) + \lambda_2 A g(x),$$

for every real λ_1 and λ_2 and every f and g from domain of definition of operator A .

b) For $A, B \in L$ it holds the commutativity $AB = BA$, i.e.

$$A(Bf(x)) = B(Af(x)).$$

c) When operator $A \in L$ has its inverse operator A^{-1} , then it holds

$$AA^{-1} = A^{-1}A = 1.$$

If we define sum of operators $C = A + B$ as

$$Cf(x) = (A + B)f(x) = Af(x) + Bf(x),$$

then the following rules hold:

$$\begin{aligned} A + B &= B + A \\ A(B + C) &= AB + AC \\ (A + B) + C &= A + (B + C). \end{aligned}$$

Being $E^k f(x) = E(E^{k-1} f(x)) = f(x + kh)$ ($k \in N$), we can define degree of operator for arbitrary exponent as

$$E^p f(x) = f(x + ph) \quad (p \in R).$$

Let $g(x) = Ef(x) = f(x + h)$. Then $f(x) = g(x - h)$, i.e. $f(x) = E^{-1}g(x)$, wherefrom follows that operator E^{-1} is backward shifting.

In the following part are given formal relations between some operators (see [1], pp. 29-32).

1. Because of

$$\Delta f(x) = f(x + h) - f(x) = Ef(x) - lf(x) = (E - 1)f(x),$$

we have

$$\Delta = E - 1 \quad \text{or} \quad E = 1 + \Delta.$$

By applying binomial formula on right-hand side of last equalities, we get

$$\Delta^k = (E - 1)^k = \sum_{i=0}^k (-1)^i \binom{k}{i} E^{k-i}, \quad E^k = (1 + \Delta)^k = \sum_{i=0}^k \binom{k}{i} \Delta^i,$$

wherefrom follow formulas (7.4.2) and (7.4.3).

2. From

$$E^{1/2} f(x) = f\left(x + \frac{h}{2}\right) \quad \text{and} \quad E^{-1/2} f(x) = f\left(x - \frac{h}{2}\right),$$

the following equalities hold:

$$\delta f(x) = E^{1/2} f(x) - E^{-1/2} f(x)$$

and

$$\mu f(x) = \frac{1}{2}(E^{1/2} f(x) + E^{-1/2} f(x)),$$

or

$$\delta = E^{1/2} - E^{-1/2} \quad \text{and} \quad \mu = \frac{1}{2}(E^{1/2} + E^{-1/2}).$$

3. It holds

$$JDf(x) = J(f'(x)) = \int_x^{x+h} f'(t) dt = f(x + h) - f(x) = \Delta f(x),$$

i.e. $JD = \Delta$. Based on previous equation, we conclude that it holds

$$\Delta D^{-1} = JDD^{-1} = J.$$

4. $\nabla = 1 - E^{-1}$.
5. $\Delta^r = E^r \nabla^r = E^{r/2} \delta^r$.
6. $E^{1/2} = (1 + \frac{1}{4}\delta^2)^{1/2} + \frac{1}{2}\delta$, $\mu = (1 + \frac{1}{4}\delta^2)^{1/2}$.
7. For connection between E and D suppose that function f is arbitrary times differentiable, so that we have

$$Ef(x) = f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \dots,$$

i.e.

$$\begin{aligned} Ef(x) &= l f(x) + hDf(x) + \frac{h^2}{2!}D^2f(x) + \dots \\ &= (1 + hD + \frac{h^2}{2!}D^2 + \dots)f(x) \\ &= e^{hD}f(x), \end{aligned}$$

wherefrom it holds

$$E = e^{hD} \quad \text{and} \quad D = \frac{1}{h} \log(1 + \Delta) = \frac{1}{h}(\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \dots).$$

8. From

$$\delta f(x) = E^{1/2}f(x) - E^{-1/2}f(x) \quad \text{and} \quad E = ehD$$

we have

$$\delta = E^{1/2} - E^{-1/2} = e^{hD/2} - e^{-hD/2} = 2sh \frac{hD}{2}.$$

Similarly,

$$\mu = \frac{1}{2}(E^{1/2} + E^{-1/2}) = ch \frac{hD}{2}.$$

9. $E = (1 - \nabla)^{-1} = 1 + \nabla + \nabla^2 + \dots$
10. $(1 + \frac{1}{4}\delta^2)^{1/2} = 1 + \frac{1}{8}\delta^2 - \frac{1}{128}\delta^4 + \dots$
11. $E^{1/2} = 1 + \frac{1}{2}\delta + \frac{1}{8}\delta^2 + \dots$

7.5. Newton's interpolation formulas

By means of finite difference calculus the several interpolation formulas in equidistant nodes can be evaluated. The oldest one is Newton's interpolation formula.

Let function f be given on $[a, b]$ by pairs of values x_k, f_k , where $f_k = f(x_k)$ and $x_k = x_0 + kh$ ($k = 0, 1, \dots, n$). For given set of data the table of finite differences can be formed. In the following table is used operator Δ .

x_0	f_0				
		Δf_0			
x_1	f_1		$\Delta^2 f_0$		
		Δf_1		$\Delta^3 f_0$	
x_2	f_2		$\Delta^2 f_1$		$\Delta^4 f_0$
		Δf_2		$\Delta^3 f_1$	
x_3	f_3		$\Delta^2 f_2$		
		Δf_3			
x_4	f_4				
\vdots	\vdots				

Let $x = x_0 + ph$ ($0 \leq p \leq n$), i.e. $p = \frac{x - x_0}{h}$. Because of

$$E^p = (1 + \Delta)^p = \sum_{k=0}^n \binom{p}{k} \Delta^k,$$

we have

$$E^p f_0 = \sum_{k=0}^{\infty} \binom{p}{k} \Delta^k f_0 = \sum_{k=0}^n \binom{p}{k} \Delta^k f_0 + R_n(f; x),$$

i.e.

$$(7.5.1) \quad f(x_0 + ph) = \sum_{k=0}^n \binom{p}{k} \Delta^k f_0 + R_n(f; x).$$

The residuum R_n , having in mind the uniqueness of interpolation polynomial, is equal to residuum of Lagrange interpolation formula

$$R_n(f; x) = \frac{h^{n+1}}{(n+1)!} p(p-1) \dots (p-n) f^{(n+1)}(\xi),$$

where ξ is point in interval (x_0, x_n) .

The polynomial

$$(7.5.2) \quad P_n(x) = \sum_{k=0}^n \binom{p}{k} \Delta^k f_0 \quad (ph = x - x_0)$$

obtained in a given way, is called first Newton's interpolation polynomial. This polynomial can be defined recursively as

$$P_k(x) = P_{k-1}(x) + \binom{p}{k} \Delta^k f_0 \quad (k = 1, \dots, n),$$

starting with $P_0(x) = f_0$. The developed form of polynomial is

$$P_n(x) = f_0 + p\Delta f_0 + \frac{p(p-1)}{2!} \Delta^2 f_0 + \dots + \frac{p(p-1) \dots (p-n+1)}{n!} \Delta^n f_0,$$

i.e.

$$P_n(x) = f_0 + \frac{\Delta f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - x_1) + \dots + \frac{\Delta^n f_0}{n!h^n}(x - x_0)(x - x_1) \dots (x - x_{n-1}).$$

Remark. The reader is encouraged to write a code in **Mathematica** for program realization of first Newton's interpolation formula, using all four previously given forms.

First Newton's interpolation polynomial is used for interpolation on begin of interval, i.e. in neighborhood of interval starting point x_0 . Interpolation of function for $x < x_0$ is called extrapolation.

Using operator ∇ we can form the table of finite differences in the following form.

$$\begin{array}{ccccccc} \vdots & & \vdots & & & & \\ x_{n-4} & f_{n-4} & & & & & \\ & & \nabla f_{n-3} & & & & \\ x_{n-3} & f_{n-3} & & \nabla^2 f_{n-2} & & & \\ & & \nabla f_{n-2} & & \Delta^3 f_{n-1} & & \\ x_{n-2} & f_{n-2} & & \nabla^2 f_{n-1} & & \nabla^4 f_n & \\ & & \nabla f_{n-1} & & \nabla^3 f_n & & \\ x_{n-1} & f_{n-1} & & \nabla^2 f_n & & & \\ & & \nabla f_n & & & & \\ x_n & f_n & & & & & \end{array}$$

Let $x = x_n + ph$ ($0 \leq -p \leq n$), i.e. $p = \frac{x-x_n}{h}$. Being

$$E^p = (1 - \nabla)^{-p} = \sum_{k=0}^{+\infty} (-1)^k \binom{-p}{k} \nabla^k,$$

we get

$$\begin{aligned} f(x_n + ph) &= \sum_{k=0}^{+\infty} (-1)^k \binom{-p}{k} \nabla^k f_n \\ &= \sum_{k=0}^{+\infty} \frac{p(p+1)\dots(p+k-1)}{k!} \nabla^k f_n. \end{aligned}$$

For $f(x_n + ph)$ is often used denotation f_{n+p} .

Using differences up to order n , based on previous equality, we get second Newton's interpolation formula

$$P_n(x) = f_n + p\nabla f_n + \frac{p(p+1)}{2!} \nabla^2 f_n + \dots + \frac{p(p+1)\dots(p+n-1)}{n!} \nabla^n f_n,$$

i.e

$$\begin{aligned} P_n(x) &= f_n + \frac{\nabla f_n}{h}(x - x_n) + \frac{\nabla^2 f_n}{2!h^2}(x - x_n)(x - x_{n-1}) + \dots \\ &\quad + \frac{\nabla^n f_n}{n!h^n}(x - x_n)(x - x_{n-1})\dots(x - x_1). \end{aligned}$$

where the residuum is of the form

$$R_n(f; x) = \frac{h^{n+1}}{(n+1)!} p(p+1)\dots(p+n) f^{(n+1)}(\xi) \quad (\xi \in [x_0, x_n]).$$

Example 7.5.1. Based on table of values of function $x \rightarrow f(x)$ form table of divided differences and write first and second Newton's interpolation polynomials.

x_k	-1	0	1	2
$f(x_k)$	-3	-5	1	21

Using operator Δ we form the following table of forward differences.

x_k	f_k	Δf_k	$\Delta^2 f_k$	$\Delta^3 f_k$
-1	<u>-3</u>			
0	-5	<u>-2</u>		
1	1	6	<u>8</u>	
2	<u>21</u>	<u>20</u>	<u>6</u>	

Using forward differences $\Delta f_0, \Delta^2 f_0, \Delta^3 f_0$ (underlined values in table) we develop the first Newton's polynomial

$$\begin{aligned} P_3(x) &= f_0 + \frac{\Delta f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2!h^2}(x - x_0)(x - x_1) \\ &\quad + \frac{\Delta^3 f_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) \\ &= -3 + (-2)(x + 1) + \frac{8}{2}(x + 1)(x - 0) + \frac{6}{3!}(x + 1)(x - 0)(x - 1) \\ &= x^3 + 4x^2 + x - 5. \end{aligned}$$

Having in mind relation between operators Δ and ∇ ,

$$\nabla f_k = f_k - f_{k-1} = \Delta f_{k-1}, \quad \nabla^2 f_k = \Delta^2 E^{-2} f_k = \Delta^2 f_{k-2}, \quad \nabla^3 f_k = \Delta^3 f_{k-3},$$

we can use already formed table of forward differences, and by indices shifting form table of backward differences, i.e. use in table overlined values as $\nabla f_n, \nabla^2 f_n, \nabla^3 f_n$. Then we get second Newton's interpolation polynomial.

$$P_3(x) = 21 + 20(x - 2) + \frac{14}{2}(x - 2)(x - 1) + \frac{6}{3!}(x - 2)(x - 1)(x - 0).$$

For bigger tables of final differences, usually by applying finite element method, it is very important propagation of accidental error of function value in some interpolation nodes. Let us have the value $f_k + \epsilon$ in spite of f_k , where ϵ is error. Then we have the following table obtained using operator Δ .

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$	$\Delta^4 f_i$
x_{k-4}	f_{k-4}				
		Δf_{k-4}			
x_{k-3}	f_{k-3}		$\Delta^2 f_{k-4}$		
		Δf_{k-3}		$\Delta^3 f_{k-4}$	
x_{k-2}	f_{k-2}		$\Delta^2 f_{k-3}$		$\underline{\Delta^4 f_{k-4} + \epsilon}$
		Δf_{k-2}		$\underline{\Delta^3 f_{k-3} + \epsilon}$	
x_{k-1}	f_{k-1}		$\underline{\Delta^2 f_{k-2} + \epsilon}$		$\underline{\Delta^4 f_{k-3} - 4\epsilon}$
		$\underline{\Delta f_{k-1} + \epsilon}$		$\underline{\Delta^3 f_{k-2} - 3\epsilon}$	
x_k	$\underline{f_k + \epsilon}$		$\underline{\Delta^2 f_{k-1} - 2\epsilon}$		$\underline{\Delta^4 f_{k-2} + 6\epsilon}$
		$\underline{\Delta f_k - \epsilon}$		$\underline{\Delta^3 f_{k-1} + 3\epsilon}$	
x_{k+1}	f_{k+1}		$\underline{\Delta^2 f_k + \epsilon}$		$\underline{\Delta^4 f_{k-1} - 4\epsilon}$
		Δf_{k+1}		$\underline{\Delta^3 f_k - \epsilon}$	
x_{k+2}	f_{k+2}		$\Delta^2 f_{k+1}$		$\underline{\Delta^4 f_k + \epsilon}$
		Δf_{k+2}		$\Delta^3 f_{k+1}$	
x_{k+3}	f_{k+3}		$\Delta^2 f_{k+2}$		
		Δf_{k+3}			
x_{k+4}	f_{k+4}				

The erroneous differences in table are underlined. We see the progressive propagation of error, so that error in difference $\Delta^m f_{k-m+i}$ ($i = 0, 1, \dots, m$) is $\binom{m}{i}(-1)^i \epsilon$.

7.6. Interpolation formulas with central differences

From numerical point of view, Newton's interpolation polynomials are not convenient, so that the interpolation polynomial with central differences are usually applied. This polynomials contain the Newton's polynomials as particular cases.

If function f is given in tabular form in the set of points

$$G_m = \{x_0 - mh, \dots, x_0 - 2h, x_0 - h, x_0, x_0 + h, x_0 + 2h, \dots, x_0 + mh\},$$

where m is fixed natural number and $h = const > 0$. Let f_k be function value in point $x_k = x_0 + kh$ ($k = 0, \pm 1, \dots, \pm m$). The table containing all possible differences of function f on set G_m is called central difference table. In set G_3 we can form central difference table of forward differences,

x_{-3}	f_{-3}						
		Δf_{-3}					
x_{-2}	f_{-2}		$\Delta^2 f_{-3}$				
		Δf_{-2}		$\Delta^3 f_{-3}$			
x_{-1}	f_{-1}		$\Delta^2 f_{-2}$		$\Delta^4 f_{-3}$		
		Δf_{-1}		$\Delta^3 f_{-2}$		$\Delta^5 f_{-3}$	
x_0	f_0		$\Delta^2 f_{-1}$		$\Delta^4 f_{-2}$		$\Delta^6 f_{-3}$
		Δf_0		$\Delta^3 f_{-1}$		$\Delta^5 f_{-2}$	
x_1	f_1		$\Delta^2 f_0$		$\Delta^4 f_{-1}$		
		Δf_1		$\Delta^3 f_0$			
x_2	f_2		$\Delta^2 f_1$				
		Δf_2					
x_3	f_3						

and following central difference table of central differences.

x_{-3}	f_{-3}						
		$\delta f_{-5/2}$					
x_{-2}	f_{-2}		$\delta^2 f_{-2}$				
		$\delta f_{-3/2}$		$\delta^3 f_{-3/2}$			
x_{-1}	f_{-1}		$\delta^2 f_{-1}$	$\delta^4 f_{-1}$			
		$\delta f_{-1/2}$		$\delta^3 f_{-1/2}$	$\delta^5 f_{-1/2}$		
x_0	f_0		$\delta^2 f_0$	$\delta^4 f_0$		$\delta^6 f_0$	
		$\delta f_{1/2}$		$\delta^3 f_{1/2}$	$\delta^5 f_{1/2}$		
x_1	f_1		$\delta^2 f_1$	$\delta^4 f_1$			
		$\delta f_{3/2}$		$\delta^3 f_{3/2}$			
x_2	f_2		$\delta^2 f_2$				
		$\delta f_{5/2}$					
x_3	f_3						

In previous example we shifted table of forward difference to table of backward difference in order to save forming one more table. From the equalities

$$\Delta^r f_k = \nabla^r E^r f_k = \nabla^r f_{k+r}, \quad \Delta^r f_k = \delta^r E^{r/2} f_k = \delta^r f_{k+r/2} \quad (r = 1, 2, \dots),$$

we see that from forward difference table it is easy to get backward difference table, or central difference table by index shifting.

By involving $x = x_0 + ph$ and $f_p = f(x)$ we can write the first Newton's interpolation formula using forward difference operator

$$(7.6.1) \quad f_p = f_0 + \binom{p}{1} \Delta f_0 + \binom{p}{2} \Delta^2 f_0 + \binom{p}{3} \Delta^3 f_0 + \dots,$$

and

$$f_p = f_0 + \binom{p}{1} \delta f_{1/2} + \binom{p}{2} \delta^2 f_1 + \binom{p}{3} \delta^3 f_{3/2} + \dots,$$

using central difference operator.

We see the general member in formula (7.6.1). By multiplying the expressions

$$\binom{p}{q} = \binom{p+1}{q+1} - \binom{p}{q+1} \quad \text{and} \quad \Delta^q f_{k+1} - \Delta^q f_k = \Delta^{q+1} f_k,$$

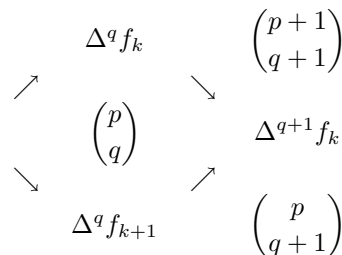
we get

$$\binom{p}{q} \Delta^q f_{k+1} - \binom{p}{q} \Delta^q f_k = \binom{p+1}{q+1} \Delta^{q+1} f_k - \binom{p}{q+1} \Delta^{q+1} f_k,$$

i.e.

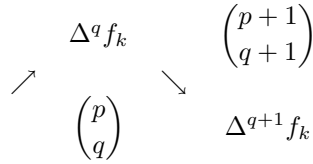
$$(7.6.2) \quad \binom{p}{q} \Delta^q f_{k+1} + \binom{p}{q+1} \Delta^{q+1} f_k = \binom{p}{q} \Delta^q f_k + \binom{p+1}{q+1} \Delta^{q+1} f_k$$

Diagram that can graphically explain the previous equality is the following one.



which is called rhomboid diagram. The idea of this graphic representation is that by moving from left to right of diagram, regardless of which path is used (lower or upper)

the result of collecting expression is the same one. By going up (down) we multiply the difference by binomial coefficient located under (above) the chosen path. For example, to the path



corresponds the expression $\binom{p}{q} \Delta^q f_k + \binom{p+1}{q+1} \Delta^{q+1} f_k$. Using the described rhomboids, it is possible to form rhomboidal network, given on the figure 7.6.1.



Fig. 7.6.1

Beside ascending and descending traces, the horizontal traces can be formed like

$$1, \Delta f_0, \binom{p}{2}, \Delta^3 f_{-1}, \dots$$

When moving this way through table, the member to be taken to sum is equal to product of member on trace and arithmetic mean of member above and below it. For the previously given trace we have

$$\frac{1}{2}(f_1 + f_2) + \frac{1}{2} \left[\binom{p}{1} + \binom{p-1}{1} \right] \Delta f_0 + \frac{1}{2} \binom{p}{2} (\Delta^2 f_{-1} + \Delta^2 f_0) + \dots$$

To every trace in rhomboid network corresponds one interpolation formula. The trace for the first Newton interpolation formula is

$$(7.6.3) \quad f_0, \Delta f_0, \Delta^2 f_0, \Delta^3 f_0, \dots$$

From the numerical point of view, for interpolation of function in point x are convenient formulas that use information on function values from both sides of point x , what was not the case with Newton's formulas.

The survey of more important interpolation formula is given in continuation.

- (1) **First Newton's interpolation formula** corresponds to the path (7.6.3) with formula given in (7.6.1).
- (2) **Second Newton's interpolation formula** corresponds to the path

$$f_0, \Delta f_{-1}, \Delta^2 f_{-2}, \Delta^3 f_{-3}, \dots$$

with formula

$$f_p = f_0 + \binom{p}{1} \Delta f_{-1} + \binom{p+1}{2} \Delta^2 f_{-2} + \binom{p+2}{3} \Delta^3 f_{-3} + \dots$$

Note that second Newton's interpolation formula is constructed in point x_0 and not in x_n , as previously given.

- (3) **First Gauss interpolation formula** corresponds to the path

$$f_0, \Delta f_0, \Delta^2 f_{-1}, \Delta^3 f_{-1}, \Delta^4 f_{-2} \dots$$

with formula of form

$$f_p = f_0 + \binom{p}{1} \Delta f_0 + \binom{p}{2} \Delta^2 f_{-1} + \binom{p+1}{3} \Delta^3 f_{-1} \\ + \binom{p+1}{4} \Delta^4 f_{-2} + \dots$$

- (4) **Second Gauss interpolation formula** corresponds to the path

$$f_0, \Delta f_{-1}, \Delta^2 f_{-1}, \Delta^3 f_{-2}, \Delta^4 f_{-2} \dots$$

and can be represented in form

$$f_p = f_0 + \binom{p}{1} \Delta f_{-1} + \binom{p+1}{2} \Delta^2 f_{-1} + \binom{p+1}{3} \Delta^3 f_{-2} \\ + \binom{p+2}{4} \Delta^4 f_{-2} + \dots$$

- (5) **Stirling's interpolation formula** corresponds to the path

$$f_0, \binom{p}{1}, \Delta^2 f_{-1}, \binom{p+1}{3}, \Delta^4 f_{-2}, \dots$$

and, according to previously given rules can be represented in form

$$f_p = f_0 + \frac{1}{2}(\Delta f_{-1} + \Delta f_0) \binom{p}{1} + \frac{1}{2} \left(\binom{p+1}{2} + \binom{p}{2} \right) \Delta^2 f_{-1} \\ + \frac{1}{2} (\Delta^3 f_{-2} + \Delta^3 f_{-1}) \binom{p+1}{3} \\ + \frac{1}{2} \left(\binom{p+2}{4} + \binom{p+1}{4} \right) \Delta^4 f_{-2} + \dots$$

Note that the Stirling's formula is obtained as arithmetic mean of first and second Gauss interpolation formula.

(6) **Bessel's interpolation formula** corresponds to horizontal path

$$1, \Delta f_0, \binom{p}{2}, \Delta^3 f_{-1}, \binom{p+1}{4}, \dots$$

and has a form

$$\begin{aligned} f_p = & \frac{1}{2}(f_0 + f_1) + \frac{1}{2} \left(\binom{p}{1} + \binom{p-1}{1} \right) \Delta f_0 + \frac{1}{2} (\Delta^2 f_{-1} + \Delta^2 f_0) \binom{p}{2} \\ & + \frac{1}{2} \left(\binom{p+1}{3} + \binom{p}{3} \right) \Delta^3 f_{-1} \\ & + \frac{1}{2} (\Delta^4 f_{-2} + \Delta^4 f_{-1}) \binom{p+1}{4} + \dots \end{aligned}$$

On the figure 7.6.2 are given paths of above given interpolation formulas. For function interpolation are used most often Stirling's and Bessel interpolation formulas. Stirling formula is used when $|p| \leq 0.25$, and Bessel for $0.25 \leq |p| \leq 0.75$.

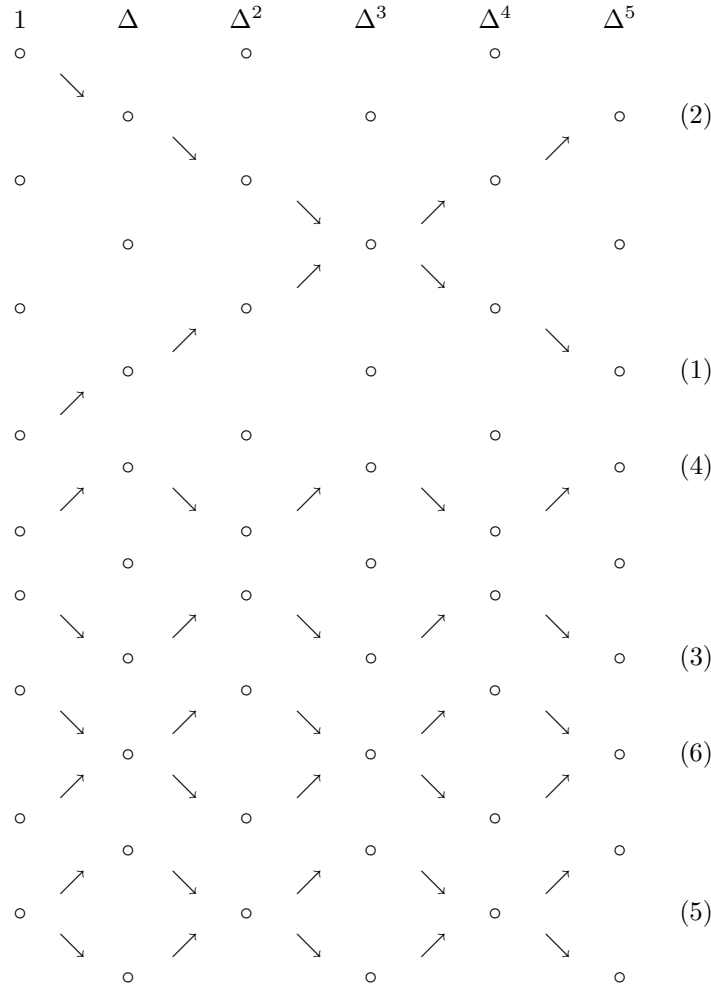


Fig. 7.6.2

Some more complex interpolation of functions involving derivatives of functions is Hermite interpolation (see [1], pp. 51-58).

In the case when one takes for basis function ϕ_k trigonometric basis $\{1, \cos x, \sin x, \cos 2x, \sin 2x, \dots\}$ we have trigonometric approximation. With $n+1 = 2m+1$ basis functions, approximation function is trigonometric polynomial of order m

$$\phi(x) = T_m(x) = \frac{1}{2}a_0 + \sum_{k=1}^m (a_k \cos kx + b_k \sin kx),$$

with approximation parameters $\frac{1}{2}a_0, a_k, b_k, (k = 1, \dots, m)$. On trigonometric interpolation and algorithms for trigonometric sums calculation see ([1], pp. 58-69).

7.7. Spline functions and interpolation by splines

Physical device named spline consists of a long strip fixed in position at a number of points that relaxes to form a smooth curve passing through those points. Before computers were used for creating engineering designs, drafting tools were employed by designers drawing by hand. To draw curves, especially for shipbuilding, draftsmen often used long, thin, flexible strips of wood, plastic, or metal called a spline (or a lath, not to be confused with lathe). The splines were held in place with lead weights (called ducks because of their duck like shape). The elasticity of the spline material combined with the constraint of the control points, or knots, would cause the strip to take the shape which minimizes the energy required for bending it between the fixed points, and thus adopt the smoothest possible shape. One can recreate a draftsman's spline device with weights and a length of thin stiff plastic or rubber tubing. The weights are attached to the tube (by gluing or pinning). The tubing is then placed over drawing paper. Crosses are marked on the paper to designate the knots or control points. The tube is then adjusted so that it passes over the control points. Supposing uniform elasticity of spline, one can say that its potential energy, when bent, is proportional to the integral along it (curvilinear integral along curve) of quadrate of convolution K . Thus, if spline lies along plane curve $y = S(x), a \leq x \leq b$, its potential energy is proportional to the integral

$$(7.7.1) \quad \int_L K(x)^2 ds = \int_a^b \frac{S''(x)^2}{(1 + S'(x)^2)^{5/2}} dx$$

and stabilized shape it takes is such that minimizes (7.10.1) under given limitations.

In the similar way is defined mathematical spline, by discarding $S'(x)^2$ in nominator of (7.7.1), what is close to previous case, when $S'(x) \ll 1$. Thus, now is to minimize the integral

$$(7.7.2) \quad \int_a^b S''(x)^2 dx.$$

Mathematical spline can be more generally defined by using higher derivative then two in (7.7.2).

First results regarding spline functions appeared in papers of Quade and Collatz ([11], 1938) and Courant([12]. In 1946 mathematicians started studying the spline shape, and derived the piecewise polynomial formula known as the spline curve or function (Schoenberg [13]). This has led to the widespread use of such functions in computer-aided design, especially in the surface designs of vehicles. Schoenberg gave the spline function its name after its resemblance to the mechanical spline used by draftsmen. The origins of the spline in wood-working may show in the conjectured etymology which connects the word spline to the word splinter. Later craftsmen have made splines out of rubber, steel, and other elastomeric materials. Spline devices help bend the wood for pianos, violins, violas, etc. The Wright brothers used one to shape the wings of their aircraft.

The extensive development of spline functions and usage of their approximation properties begun in sixties last century. The splines are greatly applied to numerical mathematics, in particular to interpolation, numerical differentiation, numerical integration, differential equations, etc. The extremal and approximative attributes of so known natural cubic spline is given in [1], pp. 81-86.

Let on segment $[a, b]$ given network of nodes

$$(7.7.3) \quad \Delta_n : a = x_0 < x_1 < \dots < x_n = b.$$

Denote with \mathcal{P}_m set of algebraic polynomials of order not greater than m .

Definition 7.7.1. *Function*

$$S_m(x) = S_{m,k}(x, \Delta_n)$$

is called *polynomial spline of degree m and defect k ($1 \leq k \leq m$) with nodes (7.7.3), if satisfies the conditions*

- 1⁰. $S_m \in \mathcal{P}_m$ on every subsegment $[x_{i-1}, x_i]$ ($i = 1, \dots, n$),
- 2⁰. $S_m \in C^{m-k}[a, b]$.

Points x_i are called nodes of spline.

We will further consider polynomial splines of defect 1 and for $S_m(x) = S_{m,1}(x)$ say to be a spline of degree m . Very important kind of splines, interpolation cubic spline, with $m = 3$ will be special considered. Therefore we join to the network nodes Δ_n real numbers f_0, f_1, \dots, f_n .

Definition 7.7.2. *Function $S_3(x) = S_3(x; f)$ is called interpolation cubic spline for function f on the net Δ_n ($n \geq 2$) if the following conditions are fulfilled:*

- 1⁰. $S_3(x, f) \in \mathcal{P}_3$ if $x \in [x_{i-1}, x_i]$ ($i = 1, \dots, n$),
- 2⁰. $S_3(x; f) \in C^2[a, b]$,
- 3⁰. $S_3(x_i; f) = f_i = f(x_i)$ ($i = 0, \dots, n$).

We see that condition 3⁰ does not appear in Definition 7.7.1. The spline defined in this way is called simple cubic spline. It interpolates function f in network nodes (condition 3⁰), it is continuous on $[a, b]$ together with its derivatives $S_3'(x)$ and $S_3''(x)$ (condition 2⁰) and defined on every subsegment between neighbor nodes with polynomial of degree not greater than 3. So, the third derivative of cubic spline is discontinuous, being, part by part of constant value.

Cubic spline has two free parameters, determined usually by some additional contour conditions. The typical ones are:

$$(7.7.4) \quad S_3'(a) = S_3'(b), \quad S_3''(a) = S_3''(b);$$

$$(7.7.5) \quad S_3'(a) = a_n, \quad S_3'(b) = b_n;$$

$$(7.7.6) \quad S_3''(a) = A_n, \quad S_3''(b) = B_n;$$

$$(7.7.7) \quad S_3''(x_1 - 0) = S_3''(x_1 + 0), \quad S_3''(x_{n-1} - 0) = S_3''(x_{n-1} + 0),$$

where a_n, b_n, A_n, B_n are given real numbers.

Conditions (7.7.4) define so known periodic spline. These conditions are used when, for example, interpolating function f is periodic with period $b - a$.

If function f is differentiable and we know values of derivatives in boundary points a and b , then the additional conditions (7.7.5), $a_n = f'(a)$ and $b_n = f'(b)$, or (7.7.6), $A_n = f''(a)$ and $B_n = f''(b)$, are to be used, what is often case at mechanical models. The corresponding spline is called natural cubic spline.

The conditions (7.7.7) are most complicated, but they obtain continuity of third derivatives of spline in points $x = x_1$ and $x = x_{n-1}$.

The most interesting spline approximation is cubic spline interpolation. The algorithm for construction of cubic spline is given in ([1], pp. 73-81). To interested reader is suggested to write a code for construction of spline (taking care of Hermite interpolation) and, if possible, include graphical interface (see Assignment-vii on GAF site). For some programming details, see Fortran subroutines `Spline` and `Splint` in ([5], pp. 109-110). For obtaining a higher order of smoothness in two-dimensional interpolation (applicable in many areas of engineering, and specially in computer graphics), one can use bicubic spline and code given in ([5], pp. 120-121.)

7.8. Prony's interpolation

Dating from 1795, Prony's interpolation ([14]) is often known as Prony's exponential approximation, and until nowadays not applied as it by its sophisticated nature deserves. It is suggested to students and readers to apply the following formulas in developing algorithms for programming of Prony's method. Some benchmarking research for comparison of application of this method, cubic spline, and, for example, least square method to some physical problems is also strongly encouraged.

If we have interpolation function of form

$$\begin{aligned} f(x) &\cong c_1 e^{a_1 x} + c_2 e^{a_2 x} + \dots + c_n e^{a_n x} \\ &= c_1 \mu_1^x + c_2 \mu_2^x + \dots + c_n \mu_n^x, \end{aligned}$$

where $\mu_k = e^{a_k}$. If function f is given on set of equidistant points $\{(x_k, f_k)\}_{k=0,1,\dots,2n-1}$, and $x_k - x_{k-1} = h = \text{const}$ ($k = 1, 2, \dots, 2n-1$), by replacing $x = x_0 + kh$ data set can be replaced by $\{(k, f_k)\}_{k=0,1,\dots,2n-1}$, where $x = 0, 1, \dots, 2n-1$. By setting interpolation problem

$$\Phi(k) = f_k \quad (k = 0, 1, \dots, 2n-1),$$

we get the system of equations

$$\begin{aligned} c_1 + c_2 + \dots + c_n &= f_0 \\ c_1 \mu_1 + c_2 \mu_2 + \dots + c_n \mu_n &= f_1 \\ c_1 \mu_1^2 + c_2 \mu_2^2 + \dots + c_n \mu_n^2 &= f_2 \\ &\vdots \\ c_1 \mu_1^{N-1} + c_2 \mu_2^{N-1} + \dots + c_n \mu_n^{N-1} &= f_{N-1}. \end{aligned} \tag{7.9.1}$$

If μ 's are known and $N = n$, system (7.9.1) is to be solved as system of linear equations, and if $N > n$ by least squares method (see next chapter).

If μ 's are to be determined, we need at least $2n$ equations, but we have system of nonlinear equation, which, as we know, in general case could be unsolvable. Therefore, we can assume that μ 's are the roots of algebraic polynomial of form

$$\begin{aligned} \mu^n + \alpha_1 \mu^{n-1} + \dots + \alpha_{n-1} \mu + \alpha_n &= 0, \\ \text{i.e.} \\ (\mu - \mu_n)(\mu - \mu_{n-1}) + \dots + (\mu - \mu_1) &= 0. \end{aligned} \tag{7.9.2}$$

By multiplying all equations in (7.11.1) by $\alpha_n, \alpha_{n-1}, \dots, \alpha_1, 1$, we get the system

$$\begin{aligned} f_0 \alpha_n + f_1 \alpha_{n-1} + f_2 \alpha_{n-2} + \dots + f_{n-1} \alpha_1 &= -f_n \\ f_1 \alpha_n + f_2 \alpha_{n-1} + f_3 \alpha_{n-2} + \dots + f_n \alpha_1 &= -f_{n+1} \\ &\vdots \\ f_{N-n-1} \alpha_n + f_{N-n} \alpha_{n-1} + f_{N-n+1} \alpha_{n-2} + \dots + f_{N-2} \alpha_1 &= -f_{N-1}. \end{aligned} \tag{7.9.3}$$

If determinant

$$\begin{vmatrix} f_0 & f_1 & \cdots & f_{n-1} \\ f_1 & f_2 & \cdots & f_n \\ \vdots & & & \\ f_{n-1} & f_n & \cdots & f_{N-2} \end{vmatrix} \neq 0,$$

the solution of system (7.9.3) is unique. If $N = 2n$ we get system of linear equations (7.9.3), and if $N > 2n$ we solve this system by least squares method.

7.9. Packages for interpolation of functions

Many libraries and software packages are available for interpolation and extrapolation of functions. Many workstations and mainframe computers have such libraries attached to their operating systems.

Many commercial software packages contain algorithms and programs for interpolation. Some of more prominent packages are `Matlab` and `Mathcad`. Some more sophisticated packages, such as `IMSL`, `MATHEMATICA`, `MACSYMA`, and `MAPLE`, also contain routines for polynomial interpolation and approximation of functions. The book *Numerical Recipes* [5] contains a routines for interpolation (See chap. 3, Interpolation and extrapolation), and the book *Numerical Methods for Engineers and Scientists* ([2]) program code for difference formulas and numerical differentiation. For demands on interpolation in two or more dimensions, higher order for accuracy, and higher order of smoothness (bicubic interpolation, bicubic spline) see code in ([5], pp. 118-122).

Bibliography (Cited references and further reading)

- [1] Milovanović, G.V., *Numerical Analysis II*, Naučna knjiga, Beograd, 1988 (Serbian).
- [2] Hoffman, J.D., *Numerical Methods for Engineers and Scientists*. Taylor & Francis, Boca Raton-London-New York-Singapore, 2001.
- [3] Milovanović, G.V. and Djordjević, Dj.R., *Programiranje numeričkih metoda na FORTRAN jeziku*. Institut za dokumentaciju zaštite na radu "Edvard Kardelj", Niš, 1981 (Serbian).
- [4] Stoer, J., and Bulirsch, R., *Introduction to Numerical Analysis*, Springer, New York, 1980.
- [5] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., *Numerical Recipes - The Art of Scientific Computing*. Cambridge University Press, 1989.
- [6] Karlin, S. and Studden, W.J., *Chebyshev Systems With Applications in Analysis and Statistics*. Interscience Publishers, New York, 1966.
- [7] Milovanović, G.V. and Kovačević, M.A., *Zbirka rešenih zadataka iz numeričke analize*. Naučna knjiga, Beograd, 1985. (Serbian).
- [8] Goertzel, G., *An algorithm for the evaluation of finite trigonometric series*. Amer. Math. Monthly 65(1958).
- [9] Cooley, J.W., Lewis, P.A.W, Welch, P.D., *Historical notes on the fast Fourier transform*. Proc. IEEE 55(1967), 1675-1677.
- [10] Cooley, J.W., Tukey, J.W., *An algorithm for the machine calculation of complex Fourier series*. Math. Comp. 19(1965), 297-301.
- [11] Quade, W. and Collatz, L., *Zur Interpolationstheorie der reellen periodischen Funktionen*. S. -B. Preus. Akad. Wiss. Phys.-Math. Kl. 30(1938), 383-429.
- [12] Courant, R., *Variational methods for the solution of problem of equilibrium and vibrations*. Bull. Amer. Math. Soc. 49(1943), 1-23.
- [13] Schoenberg, I.J., *Contributions to the problem of approximation of equidistant data by analytic functions*. Quart. Appl. Math. 4(1946), 45-99; 112-141.
- [14] Prony, R. De, *Essai expérimentale et analytique...* J. Ec. Polytech. Paris 1(2) (1795), 24-76.

- [15] Ralston,A., *A First Course in Numerical Analysis*.
McGraw-Hill, New York, 1965.
- [16] *IMSL Math/Library Users Manual* , IMSL Inc., 2500 City West Boulevard, Houston
TX 77042
- [17] *NAG Fortran Library*, Numerical Algorithms Group, 256 Banbury Road, Oxford
OX27DE, U.K., Chapter F02.