CHAPTER-2

INTERPOLATION

- Interpolation formula
- Interpolation in numerical mathematics
- The approximate representation and calculation of functions
- Numerically solving systems of non-linear equations
- Numerical integration
- Numerical differentiation
- Numerically solving integral equations
- Numerically solving differential equations
- Interpolation of operators and some general approaches to the construction of numerical methods.
- Interpolation process
- References
Interpolation formula

A formula for the approximate calculation of values of a function \( f(x) \) by replacing it by a function

\[
g(x) = g(x; \alpha_0, \ldots, \alpha_n)
\]

that is simple in a certain sense and belongs to a certain class. The parameters \( \alpha_i \), \( i = 0, \ldots, n \), are chosen in such a way that the values of \( g(x) \) coincide with the known values of \( f(x) \) on a given set of \( n + 1 \) distinct values of the argument:

\[
g(x_k) = f(x_k), \quad k = 0, \ldots, n.
\] (1)

This method of approximately representing a function is called interpolation and the points \( x_k \) at which (1) should hold are called interpolation nodes\(^{[1]}\). Instead of the simplest condition (1), the values of some quantity related to \( f(x) \) may also be given, e.g. the values of a derivative of \( f(x) \) at interpolation nodes.

The method of linear interpolation is the most widespread among the interpolation methods. The approximation is now looked for in the class of (generalized) polynomials

\[
g(x; \alpha_0, \ldots, \alpha_n) = \sum_{i=0}^{n} \alpha_i \phi_i(x)
\] (2)

in some fixed system of functions \( \phi_0(x), \ldots, \phi_n(x) \). In order for the interpolation polynomial (2) to exist for any function \( f(x) \) defined on an interval \( [a, b] \), and for any choice of \( n + 1 \) nodes \( x_0, \ldots, x_n \in [a, b] \), \( x_i \neq x_j \) if \( i \neq j \), it is necessary and sufficient that \( \{ \phi_i(x) \} \) is a Chebyshev system of functions on \( [a, b] \). The interpolation polynomial will, moreover, be unique and its coefficients \( \alpha_i \) can be found by directly solving (1).

For \( \{ \phi_i(x) \} \) one often takes: the sequences of powers of \( x \),

36
the sequence of trigonometric functions,

1, \sin x, \cos x, \sin 2x, \cos 2x, \ldots,

or the sequence of exponential functions\(^{[19]}\),

1, e^{\alpha_1 x}, e^{\alpha_2 x}, \ldots,

where \( \{ \alpha_i \} \) is a sequence of distinct real numbers.

When interpolating by algebraic polynomials

\[
\sum_{i=0}^{n} \alpha_i x^i
\]  \tag{3}

the system \( \{ \phi_i (x) \} \) is

\[
\phi_i (x) = x^i, \quad i = 0, \ldots, n \tag{4}
\]

while (1) has the form

\[
\sum_{i=0}^{n} \alpha_i x_k^i = f(x_k), \quad k = 0, \ldots, n \tag{5}
\]

The system (4) is a Chebyshev system, which ensures the existence and uniqueness of the interpolation polynomial (3). A property of (4) is the possibility of obtaining an explicit representation of the interpolation polynomial (3) without immediately having to solve (5). One of the explicit forms of (3),

\[
g_n (x) = L_n (x) = \sum_{i=0}^{n} f(x_i) \prod_{j \neq i} \frac{x-x_j}{x_i-x_j}, \tag{6}
\]

37
is called the Lagrange interpolation polynomial. If the derivative \( f^{(n+1)}(x) \) is continuous, the remainder of (6) can be written as

\[
f(x) - g_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_n(x),
\]

\( \xi \in [y_1, y_2] \),  \( \omega_n(x) = \prod_{i=0}^{n} (x-x_i) \),

where \( y_1 = \min(x_0, \ldots, x_n, x) \), \( y_2 = \max(x_0, \ldots, x_n, x) \). The value of the remainder (7) depends, in particular, on the values of \( \omega_n(x) \). The choice of interpolation nodes\(^4\) for which \( \sup_{[a,b]} |\omega_n(x)| \) is minimal, is of interest. The distribution of the nodes is optimal in this sense if the roots

\[ x_k = \frac{b + a}{2} + \frac{b - a}{2} \cos \frac{2k+1}{2n+2} \pi, \quad k = 0, \ldots, n, \]

of the polynomial

\[ T_{n+1}^{[a,b]}(x) = (b-a)^{n+1} 2^{-1-2n} T_{n+1} \left( \frac{2x - (b+a)}{b-a} \right), \]

which deviates least from zero on \([a, b]\), are taken as the nodes. Here \( T_{n+1}(z) \) is the Chebyshev polynomial of degree.

There is a number of other explicit representations of (3) that are more useful for solving this or another practical interpolation problem\(^5\) (e.g., Bessel interpolation formula; Gauss interpolation formula; Newton interpolation formula; Stirling interpolation formula; Steffensen interpolation formula; Everett interpolation formula). If it is difficult to estimate in advance the degree of the interpolation polynomial that is necessary for attaining the error desired (e.g., when interpolating a table), then one takes recourse to the Aitken scheme. In this scheme interpolation polynomials of increasing degrees are constructed sequentially, thus making it possible to control the accuracy in
the computational process[6]. Another approach to the construction of interpolation formulas can be found in Fraser diagram.

The Hermite interpolation formula gives the solution to the problem of the algebraic interpolation of the values of a function and its derivatives at interpolation nodes.

Consider the interpolation problem of finding a polynomial \( P_N \) of degree \( \leq N \) satisfying the conditions

\[
P_N^{(k)}(x_i) = c_{i,k}, \quad (a1)
\]

where the \( x_1, ..., x_m \) are \( m \) distinct knots, and there are precisely \( N + 1 \) equations in (a1). If for each \( i \) the orders of the derivatives occurring in (a1) form an unbroken series \( k = 0, ..., k_i \), one has Hermite interpolation. (In case \( k_i = 0 \) for all \( i \), i.e. if no interpolation conditions involving derivatives occur in (a1), one has Lagrange interpolation.) If gaps (lacunae) occur, one speaks of lacunary interpolation or Birkhoff interpolation[24]. The pairs \((i, k)\) which occur in (a1) are conveniently described in terms of an interpolation matrix \( E \) of size \( m \times (n + 1) \), \( E = (e_{i,k}), i = 1, ..., m, k = 0, ..., n \), where \( e_{i,k} = 1 \) if \((i, k)\) does occur in (a1) and \( e_{i,k} = 0 \) otherwise. The matrix \( E \) is called regular if (a1) is solvable for all choices of the \( x_i \) and \( c_{i,k} \) and singular otherwise.

More generally, let \( G = \{g_0, ..., g_N\} \) be a system of linearly independent \( n \)-times continuously-differentiable real-valued functions on an interval \([a, b]\) or on the circle[2]. Instead of polynomials, now consider linear combinations \( P = \sum_{j=0}^{N} a_j g_j \). A matrix of zeros and ones \( E = (e_{i,k}), i = 1, ..., m, k = 0, ..., n \), is an interpolation matrix if there are precisely \( N + 1 \) ones in \( E \) (and, usually, if there are no rows of zeros in \( E \), this means that all knots do occur at least once in an interpolation condition). Let \( K = \{x_1, ..., x_m\} \) be a set of knots, i.e. \( m \) distinct points of the interval or circle. Finally,
for each \((i, k)\) such that \(c_{i,k} = 1\) let there be given a number \(c_{i,k}\). These data \((G, E, K, c_{i,k})\) define a Birkhoff interpolation problem:

\[
P^{(k)}(x_i) = c_{i,k}
\]

for all \((i, k)\) such that \(c_{i,k} = 1\).

The pair \((E, K)\) is called regular if (a2) is solvable for all choices of the \(c_{i,k}\).

For each \((i, k)\) such that \(c_{i,k} = 1\), consider the row vector of length \(N + 1\),

\[
\mathbf{g}_0^{(k)}(x_i), \ldots, \mathbf{g}_N^{(k)}(x_i).
\]

For varying \((i, k)\) such that \(c_{i,k} = 1\) one thus finds \(N + 1\) row vectors which together make up an \((N + 1) \times (N + 1)\) matrix. The pair \((E, K)\) is regular if and only if this matrix is invertible. Its determinant, where the pairs \((i, k)\) with \(c_{i,k} = 1\) are ordered lexicographically, is denoted \(D(E, K)\).

Suppose that the \(c_{i,k}\) are the values \(f^{(k)}(x_i)\) of the derivatives of some function \(f\) at the knots. Then a simple formula for the solution of the interpolation problem (a2) follows from Cramer's rule\(^{[20]}\). Indeed, if \(D^{(j)}(E, K)\) denotes the determinant obtained by replacing \(g_j\) with \(f\) in the formula for \(D(E, K)\), then

\[
P(t) = \sum_{j=0}^{N} \frac{D^{(j)}(E, K)}{D(E, K)} g_j(t)
\]

Interpolation in numerical mathematics

A method for approximating\(^{[7]}\) or precisely finding some quantity by known individual values of it or of other quantities related to it. On the basis of interpolation a
whole series of approximate methods for solving mathematical problems has been developed.

Most significant in numerical mathematics is the problem of constructing means for the interpolation of functions. The interpolation of functionals and operators is also widely used in constructing numerical methods\textsuperscript{[9-11]}

The approximate representation and calculation of functions.

Interpolation of functions is considered as one of the ways of approximating them. Interpolating a function \( f(x) \) on a segment \([a, b]\) by its values at the nodes \( x_k \) of a grid \( \Delta_n = \{a \leq x_0 < \ldots < x_n \leq b\} \) means constructing another function \( L_n(x) \equiv L_n(f; x) \) such that \( L_n'(x_k) = f(x_k), k = 0, \ldots, n \). In a more general setting, the problem of interpolating a function \( f(x) \) consists of constructing \( L_n(x) \) not only by prescribing values on a grid \( \Delta_n \), but also derivatives at individual nodes, up to a certain order, or by describing some other relation connecting \( f(x) \) and \( L_n(x) \).

Usually \( L_n(x) \) is constructed in the form

\[
L_n(x) = \sum_{i=0}^{n} a_i \phi_i(x),
\]

where \( \{ \phi_i(x) \} \) is a certain system of linearly independent functions, chosen in advance. Such an interpolation is called linear with respect to \( \{ \phi_i(x) \} \), while \( L_n(x) \) is called an interpolation polynomial in the system \( \{ \phi_i(x) \} \) or an interpolation function\textsuperscript{[21]}.

The choice of \( \{ \phi_i(x) \} \) is determined by the properties of the function class for which the interpolation formula is constructed. E.g., for the approximation of \( 2\pi \)-periodic functions on \([0, 2\pi]\) one naturally takes the trigonometric system for \( \{ \phi_i(x) \} \), for the approximation of bounded or increasing functions on \([0, \infty)\) one takes the system of rational or exponential functions\textsuperscript{[19]}, taking into account the behaviour of the functions to be approximated at infinity, etc.
Most often one uses algebraic interpolation: \( \phi_i(x) = x^i \); its simplest variant (linear interpolation with two nodes \( x_k \) and \( x_{k+1} \)) is defined by the formula

\[
L_1(x) = \frac{x - x_k}{x_{k+1} - x_k} \left[ f(x_{k+1}) - f(x_k) \right] + f(x_k),
\]

\( x_k \leq x \leq x_{k+1} \). 

Algebraic interpolation of a very high order is seldom used in practice in the problem of approximating functions on an entire segment \([a, b]\). One usually restricts oneself to linear interpolation by (1) or to quadratic interpolation with three nodes on particular segments of the grid, by the formula

\[
L_2(x) = \frac{(x - x_k)(x - x_{k+1})}{(x_{k+1} - x_k)(x_{k+1} - x_{k+2})} f(x_{k+1}) + \frac{(x - x_k - 1)(x - x_k)}{(x_{k+1} - x_k - 1)(x_{k+2} - x_k)} f(x_k) + \frac{(x - x_k - 1)(x - x_{k+1})}{(x_{k+1} - x_k - 1)(x_{k+2} - x_k)} f(x_k + 1),
\]

\( x_k - 1 \leq x \leq x_{k+1} \).

There are several ways of writing the algebraic interpolation polynomials. Interpolation by splines gains increasing application.

Parabolic or cubic splines are most often used in practice. An interpolation spline of defect 1 for a function \( f(x) \) with respect to a given grid \( \Delta_n \) is a function \( S_3(x) \equiv S_3(f; x) \) that is a polynomial of degree three on each segment \([x_k, x_{k+1}]\), belongs to the class of twice continuously-differentiable functions, and satisfies the conditions

\( S_3(x_k) = f(x_k), \quad k = 0, \ldots, n; \quad n \geq 2 \).

There are still two free parameters in this definition; these are determined by additional boundary conditions: \( S_3^{(i)}(a) = S_3^{(i)}(b), \quad i = 1, 2, \quad S_3^{(3)}(a) = \alpha, \quad S_3^{(3)}(b) = b \), or other conditions.

As well as immediately in the problem of approximating functions, splines are also used in solving other problems; the splines are required then not only to coincide on
a grid $\Delta_n$ with the values of a function $f(x)$, but also with those of the derivatives of this function, up to a certain order.

When processing empirical data $\{y_k\}$ one often determines the coefficients $a_i$ in $L_n(x)$ by requiring

$$S = \sum_{k=1}^{m} [y_k - L_n(x_k)]^2,$$

$m \geq n$,

to be minimal. The function $L_n(x)$ thus constructed is called the interpolation function in the sense of least squares.

The interpolation of functions in several variables meets with a number of principal and numerical difficulties.\textsuperscript{10} E.g., in the case of algebraic interpolation the Lagrange interpolation polynomial of fixed degree need not, generally speaking, exist for an arbitrary system of different nodes. In particular, for a function $f(x, y)$ in two variables such a polynomials $L_n(x, y)$ of total degree at most $n$ can be constructed for nodes $(x_k, y_k)$ only if these nodes do not lie on an algebraic curve of order $n$.

Another approach to the interpolation of functions $f(x_1, \ldots, x_m)$ in several variables is that one first interpolates the function with respect to $x_1$ for fixed $x_k$, $k = 2, \ldots, m$, then with respect to the next variable $x_2$ for fixed remaining nodes, etc.

Now the interpolation polynomial $L_{n_1 \ldots n_m}(x_1, \ldots, x_m)$ for $f(x_1, \ldots, x_m)$ with nodes

$$(x_1^1, \ldots, x_m^m),$$

$x_j^\nu \neq x_j^\mu$, $\nu \neq \mu$;

$k_j = 0, \ldots, n_j$; $j = 1, \ldots, m$,

has the form:

$$L_{n_1 \ldots n_m}(x_1, \ldots, x_m) =$$

$$\sum_{k_1, \ldots, k_m = 0}^{n_1 \ldots n_m} \frac{\omega(x_1^1) \ldots \omega(x_m^m) f(x_1^{k_1}, \ldots, x_m^{k_m})}{\omega(x_1^1)^{k_1} \ldots \omega(x_m^m)^{k_m} (x_1 - x_1^1) \ldots (x_m - x_m^{k_m})},$$

43
where
\[ \omega_{n,j} (x_j) = \prod_{k_j=0}^{n_j} (x_j - x_j^{k_j}), \quad j = 1, \ldots, m. \]

Interpolation splines for functions of several variables are defined on a multi-dimensional grid, with corresponding changes, in analogy with the one-dimensional case. Interpolation of functions is used for replacing complicate functions by simpler ones that can be calculated quicker; for the approximate recovery of functions on their entire domain of values by their values at individual points; and for obtaining smoother functions described by a running process. This kind of problems is of both independent interest and arises in an auxiliary fashion in many branches of science and technology in solving complex problems. Interpolation of functions is also used in approximately finding limit values of functions, in problems of accelerating the convergence of series or sequences, etc.

**Numerically solving systems of non-linear equations.**

The general ideas for constructing interpolation methods for solving an equation \( f(x) = 0 \) or a system of equations \( f_i(x_1, \ldots, x_m) = 0, i = 1, \ldots, m \), are the same. The difficulties of the problem of interpolating functions of several variables especially arise when investigating and practically using this kind of methods for a large number of equations. The basic principle for obtaining interpolation methods for solving an equation \( f(x) = 0 \) is the replacement of \( f(x) \) by its interpolation polynomials \( L_n(x) \) and subsequently solving \( L_n(x) = 0 \). The roots of \( L_n(x) = 0 \) are taken as approximate values of those of \( f(x) = 0 \). The interpolation polynomial \( L_n(x) \) is also used in constructing iteration methods for solving \( f(x) = 0 \). e.g., taking for \( x_0 + 1 \) the root of the linear algebraic interpolation polynomial constructed with respect to the values \( f(x_n) \) and \( f'(x_n) \) at \( x_n \), or with respect to the values \( f(x_n - 1) \) and \( f(x_n) \) at \( x_n - 1 \) and \( x_n \), leads to the method of Newton.
\[ x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad x_{n+1} = x_n - \frac{f(x_n)}{f(x_{n-1}, x_n)}, \]

where \( f(x_{n-1}, x_n) \) is the divided difference of \( f(x) \) at \( x_{n-1} \) and \( x_n \). Under certain conditions, the sequence \( \{x_n\} \) converges, as \( n \to \infty \), to a solution of \( f(x) = 0 \).

Another way to construct methods for solving an equation \( f(x) = 0 \) is based on interpolating the inverse function \( x = g(y) \). Suppose that for interpolating \( g(y) \) the algebraic Lagrange interpolation polynomial

\[ L_n(y) = \sum_{k=0}^{n} \frac{\omega_n(y)}{\omega_n'(y_k)} (y - y_k) g(y_k), \quad \omega_n(y) = \prod_{k=0}^{n} (y - y_k), \]

is taken. It is assumed that the inverse function exists in a neighbourhood of the required root of \( f(x) = 0 \) and that a table of values \( x_k \) and \( y_k = f(x_k), \quad k = 0, \ldots, n \), is available. The next approximate value \( x_{n+1} \) is the value of the interpolation polynomial at zero:

\[ x_{n+1} = -\frac{\omega_n(0)}{\omega_n'(y_k)y_k} \sum_{k=0}^{n} \frac{x_k}{\omega_n'(y_k)y_k}. \]

**Numerical integration.**

The apparatus of interpolation lies at the basis of constructing a lot of quadrature and cubature formulas. Such formulas are constructed by replacing the functions in the integrands on the entire domain, or on a part of it, by interpolation polynomials of a certain kind and integrating these. e.g., quadrature formulas of highest algebraic accuracy, also called Gauss quadrature formulas)

\[ \int_{\alpha}^{b} p(x)f(x)dx \approx \sum_{k=1}^{n} A_k f(x_k), \]

where \( p(x) \) is a sign-definite weight function, are obtained by replacing \( f(x) \) by algebraic interpolation polynomials constructed with respect to the roots \( x_k \) of the polynomial of degree \( n \) that is orthogonal to \( p(x) \).
If one partitions the complete integration interval \([a, b]\) in an even number \(n\) of equal parts of length \(h = (b - a) / n\) and if on each double part one replaces \(f(x)\) by its quadratic interpolation polynomials with nodes at the end and middle points, then one is led to the compound Simpson formula

\[
\int_a^b f(x) \, dx \approx \frac{b - a}{3n} \times \\
\times [f_0 + f_n + 2(f_2 + f_4 + \ldots + f_{n-2}) + 4(f_1 + f_3 + \ldots + f_{n-1})],
\]

where \(f_k = f(a + kh), k = 0, \ldots, n\).

One may also take interpolation splines of some fixed degree as basis. The scheme for constructing formulas for the approximate computation of integrals explained above can be used in the multi-dimensional case as well.

**Numerical differentiation.**

Formulas for numerical differentiation are obtained as results of differentiating interpolation formulas. Here, as a rule, certain a priori information is available about the function to be differentiated, related to its smoothness.

Let \(L_n(x)\) be a certain interpolation polynomial of a function \(f(x)\), and let \(R_n(x)\) be the remainder in the interpolation formula

\[
f(x) = L_n(x) + R_n(x).
\]

If in

\[
f^{(i)}(x) = L_n^{(i)}(x) + R_n^{(i)}(x)
\]

the quantity \(R_n^{(i)}(x)\) is neglected, one obtains a formula for the approximate computation of the \(i\)-th derivative of \(f(x)\):

\[
f^{(i)}(x) \approx L_n^{(i)}(x).
\]
obtained by differentiation of the formulas of linear and quadratic interpolation (1) and (2), respectively. Then for a second-order ordinary differential equation

\[ F(x, y, y', y'') = 0 \]

one obtains, under certain additional conditions, taking (6) into account, the finite-difference equation

\[ F(x_k, y_k, \frac{1}{h} \Delta y_k, \frac{1}{h^2} \Delta^2 y_k) = 0. \]

In it, together with the equations obtained from the additional conditions, the approximate values \( y_k \) of the solution \( y(x) \) at nodes \( x_k \) occur.

Often, reduction of partial differential equations to the corresponding finite-difference equations is often also carried out using formulas for numerical differentiation.

Interpolation methods are applied to solve differential equations written in integral form, e.g., to find an approximate solution of the Cauchy problem

\[ y' = f(x, y), \quad y(x_0) = y_0, \tag{7} \]

at points \( x_k = x_0 + kh, \ k = 0, 1, \ldots \), one uses difference formulas of the form

\[ y_{n+1} = y_n + h \sum_{j=1}^{n} B_j f(x_{n-j}, y_{n-j}), \]

obtained by replacing the integrand in

\[ y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y) \, dx \]

by an interpolation polynomial and subsequent integration. In particular, Adams' formula for first-order equations (7), Störmer's formula for second-order equations, etc., are obtained this way.
This approach makes it possible to construct numerical algorithms for a wide class of differential equations, including partial differential equations. The study of the solvability, exactness and stability of solutions of finite-difference equations that arise constitutes a fundamental and difficult part of the theory of numerically solving differential equations.

**Interpolation of operators and some general approaches to the construction of numerical methods.**

The construction of numerical methods for solving mathematical problems written as $Ax = y$, where $x$ and $y$ are elements of certain sets $X$ and $Y$ and $A$ is a given operator, consists of replacing $X$, $Y$ and $A$, or only some of these three objects, by other objects that are convenient for calculating purposes. This replacement should be carried out in such a way that the solution to the new problem

$$y = \tilde{A}x,$$

consisting of finding $\tilde{Y}$ or $\tilde{x}$, is in some sense close to the solution of the original problem. One of the methods for replacing $A$ by an approximation $\tilde{A}$ is the use of interpolation of operators. The problem of the interpolation of operators has various formulations. A linear interpolation operator $L_1 (F; x)$ for a given operator is written as

$$L_1 (F; x) = F(x_0 ) + F(x_0 , x_1)(x - x_0 ),$$

(8)

where $x_0 , x_1$ are the interpolation nodes and $F(x_0 , x_1)$ is the first-order divided-difference operator. The latter is defined as the linear operator for which

$$F(x_0 , x_1)(x_0 - x_1 ) = F(x_0 ) - F(x_1 ).$$

The given definition of the divided-difference operator can be made concrete in a number of cases. Using linear interpolation (8), the "secant method" for the equation $F(x) = 0$ can be written as

$$x_{n+1} = x_n - F^{-1}(x_{n-1} , x_n )F(x_n ),$$

50
where $F^{-1}(x_{n-1}, x_n)$ is the operator inverse to $F(x_{n-1}, x_n)$.

The formulation of the problem of interpolation of functionals, which is of interest in the theory of approximation methods, is as follows. Let $\{\psi_i(x)\}_{i=0}^n$ be some fixed functionals, or classes of functionals, defined on $X$. A functional $L_n[F; x]$ is called an interpolation functional polynomial for a given functional $F(x)$ and system of points $\{x_k\}$ in $X$ if the relations

$$L_n[F; x_k] = F(x_k), \quad L_n[\psi_i; x] \equiv \psi_i(x), \quad i = 0, \ldots, n,$$

hold.

Interpolation of functionals is used in the construction of approximate methods for computing continual integrals, in finding extrema of functionals, and in a number of other cases.

E.g., approximate interpolation formulas for computing continual integrals have the form

$$\int_X F(x) \, d \mu(x) \approx \int_X L_n[F; x] \, d \mu(x),$$

where the integral over the interpolation polynomial $L_n[F; x]$ with respect to a certain measure $\mu$ can be computed exactly or can be reduced to a finite-dimensional integral.

When $X$ is the space $C[a, b]$ of continuous functions on an interval $[a, b]$, then $L_1[F; x]$ can be represented by a Stieltjes integral,

$$L_1[F; x] = F(x_0) + \int_a^b \left( \frac{x(\tau) - x_0(\tau)}{x_1(\tau) - x_0(\tau)} \right) d \tau \left[ F(x_0(\tau)) + \chi(\tau \tau)(x_1(\tau) - x(\tau)) \right],$$

where $x_0(\tau), x_1(\tau)$ are the interpolation nodes while

$$\chi(\tau, \tau) = \begin{cases} 1, & \tau > \tau, \\ 0, & \tau \leq \tau. \end{cases}$$

51
If $F(x)$ is a constant or a linear functional, then $L_1[F; x] = F(x)$.

The use of interpolation in finding extremal values of functionals may be illustrated by two interpolation analogues of the gradient method for finding a local unconditional minimum of a functional $F(x)$ that is defined on some Hilbert space. The first analogue is obtained if, in the gradient method, $\text{grad } F(x_n)$ is replaced by $F(x_n - 1, x_n)$, i.e.

$$x_{n+1} = x_n - \epsilon_n F(x_n - 1, x_n), \quad \epsilon_n > 0, \quad n = 1, 2, \ldots \quad (9)$$

The second analogue uses the gradient of the interpolation polynomials. From approximations $x_n - 2, x_n - 1, x_n$ to an extremum $x^*$ of $F(x)$ one constructs the quadratic interpolation polynomial

$$L_2[F; x] = F(x_n) + F(x_n - 1, x_n)(x - x_n) + F(x_n - 2, x_n - 1, x_n)(x - x_n - 1)(x - x_n),$$

where $F(x_n - 2, x_n - 1, x_n)$ is the second-order divided difference of $F(x)$ with respect to $x_n - 2, x_n - 1, x_n$. The new approximation $x_{n+1}$ is determined by

$$x_{n+1} = x_n - \epsilon_n \text{ grad } L_2[F; x_n], \quad \epsilon_n > 0, \quad n = 2, 3, \ldots \quad (10)$$

The interpolation methods (9), (10) use two, respectively, there, initial approximations.

The use of interpolation of operators and functionals in the construction of computational algorithms for solving concrete problems is based on the use of interpolation formulas with a small error. Such a kind or formulas must be constructed for individual classes of functionals and operators, taking specific features of these classes into account.
Interpolation process

A process for obtaining a sequence of interpolation functions \( \{ f_n(z) \} \) for an indefinitely-growing number \( n \) of interpolation conditions. If the interpolation functions \( f_n(z) \) are represented by the partial sums of some series of functions, the series is sometimes called an interpolation series. The aim of an interpolation process often is, at least in the simplest basic problems of interpolating, the approximation (in some sense) by means of interpolation functions \( f_n(z) \) of an initial function \( f(z) \) about which one only has either incomplete information or whose form is too complicated to deal with directly.

A sufficiently general situation related to constructing interpolation processes is described in what follows. Let \( (a_{jk}), 0 \leq k \leq j, j = 0, 1, \ldots, \) be an infinite triangular table of arbitrary but fixed complex numbers:

\[
\begin{array}{cccc}
a_{00} & \square & \square & \square \\
\alpha & a_{10} & \square & \square \\
\cdots & \cdots & \cdots & \square \\
a_{n0} & a_{n1} & \cdots & a_{nn} \\
\cdots & \cdots & \cdots & \cdots \\
\end{array}
\]  

(1)

called interpolation nodes or interpolation knots. Suppose that next to (1) there is an analogous table \( (w_{jk}), 0 \leq k \leq j, j = 0, 1, \ldots, \) also consisting of arbitrary fixed complex numbers.

If the \( n \)-th row \( a_{nk}, k = 0, \ldots, n \), of (1) consists of different numbers, or, otherwise said, if this row consists of simple nodes, then, using e.g. the Lagrange interpolation formula, one constructs the (unique) algebraic interpolation polynomial \( p_n(z) \) of degree at most \( n \) satisfying the simple interpolation condition

\[
p_n(a_{nk}) = w_{nk}, \quad k = 0, \ldots, n.
\]  

(2)

If, on the other hand, the point \( a_{n0} \) is a multiple node of multiplicity \( \nu_0 > 1 \) in the \( n \)-th row, i.e. if it is encountered \( \nu_0 \) times in the \( n \)-th row:
\[ \alpha_0 = \alpha, k_1 = \ldots = \alpha, k_{(\nu_0 - 1)}, \text{ then the corresponding multiple interpolation condition at } \alpha, 0 \text{ has the form} \]

\[ p_n(\alpha_0) = p_n^{(0)}(\alpha_0) = w_{\alpha_0}, \]
\[ p_n'(\alpha_0) = w_{\alpha_0}k_1, \ldots, p_n^{(\nu_0 - 1)}(\alpha_0) = w_{\alpha_0}k_{(\nu_0 - 1)} \cdot \tag{3} \]

In the general case in the presence of multiple nodes the (unique) algebraic interpolation polynomial \( p_n(z) \) of degree at most \( n \) is constructed using, e.g., the Hermite interpolation formula. As an example, the system (1) may consist of the systems of \( j + 1, j = 0, 1 \), \ldots, equally-spaced nodes \( \alpha_j = e^{2\pi i k/(j + 1)} \) on the unit circle. This situation is so-called interpolation at roots of unity.

As a result of the interpolation process described one obtains a sequence of interpolation polynomials \( \{p_n(z)\} \) defined by the tables \( (\alpha_jk) \) and \( (w_jk) \). The main questions that arise here are: to determine the set \( E \subset C \) of points of convergence of the sequence \( \{p_n(z)\} \), at which \( \lim_{n \to \infty} p_n(z) = g(z) \) exists, in dependence on \( (\alpha_jk) \) and \( (w_jk) \), to determine the character of the limit function \( g(z) \), to determine the set \( F \subset E \) of uniform convergence \( p_n(x) \to g(z) \), etc.

In the theory of functions of a complex variable the case where the table \( (w_jk) \) is constructed from the values of a regular analytic function \( f(z) \) and its derivatives at the interpolation nodes, such that (applied to a node \( \alpha_0 \) of multiplicity \( \nu_0 \geq 1 \)) \:[14,17] \]

\[ p_n(\alpha_0) = p_n^{(0)}(\alpha_0) = f(\alpha_0), \]
\[ p_n'(\alpha_0) = f'(\alpha_0), \ldots, p_n^{(\nu_0 - 1)}(\alpha_0) = f^{(\nu_0 - 1)}(\alpha_0), \]

has been well-studied. In this case the interpolation polynomial \( P_n(z) \) can be written, by Hermite's formula, as a contour integral over a contour \( \Gamma \) encircling the nodes \( \alpha_n k \), \( k = 0, \ldots, n \), on and inside which \( f(z) \) is regular:

54
\[ p_n(z) = \frac{1}{2\pi i} \oint_{\Gamma} \frac{\omega_n(t) - \omega_n(z)}{\omega_n(t)(t-z)} f(t) \, dt, \]  

(4)

where

\[ \omega_n(z) = (z - \alpha_n) \ldots (z - \alpha_{nn}). \]

Formula (4) easily implies an integral representation for the remainder term of interpolation \( R_n(z) = f(z) - p_n(z) \). Generally speaking, the sequence \( \{ p_n(z) \} \) constructed from \( f(z) \) may diverge. If, however, it converges, then the limit function \( g(z) \) need not coincide with \( f(z) \). The fundamental question is the study of the convergence of \( \{ p_n(z) \} \) to \( f(z) \), and the determination of those systems of nodes \( \{ \alpha_{jk} \} \) for which this convergence is optimal in a certain sense. Suppose, e.g., that \( f(z) \) is a regular function on a continuum \( K \subset \mathbb{C} \) containing at least two points and whose complement in the extended complex plane \( \overline{\mathbb{C}} \) is a simply-connected domain containing the point at infinity. Let the nodes \( \{ \alpha_{jk} \} \) belong to \( K \). Then \( \{ p_n(z) \} \) converges uniformly to \( f(z) \) on \( K \) if and only if

\[ \lim_{n \to \infty} M_n^{1/(n+1)} = c, \]

where \( M_n = \sup \{ \omega_n(z) : z \in \partial K \} \), and \( c \) is the capacity of \( K \) (cf. [4]).

The classical variant of an interpolation process is obtained if the \( \alpha_{jk} = \alpha_k \), \( 0 \leq k \leq j, j = 0, 1, \ldots \) form a sequence \( \{ \alpha_{jk} \} \) for which at the \( n \)-th step the \( n \)-th nodes \( \alpha_0, \ldots, \alpha_n \) are used to construct \( p_n(z) \). For a regular function \( f(z) \) the polynomials \( p_n(z) \) are in this case the partial sums of the Newton interpolation series.

\[ q(z) = \sum_{n=0}^{\infty} c_n \omega_n(z), \]

\[ \omega_n(z) = (z - \alpha_0) \ldots (z - \alpha_n), \]

\[ c_n = \frac{1}{2\pi i} \oint_{\Gamma} \frac{f(t) \, dt}{\omega_n(t)}. \]  

(5)
In calculations an interpolation series of the form (5) has the advantage over the sequence \( \{p_n(z)\} \) that in the transition from the known polynomial \( p_n(z) \) to \( p_{n+1}(z) \) only one coefficient \( c_{n+1} \) of the series has to be computed. Depending on the nodes \( \alpha_k \) and coefficients \( c_k \), the domain of convergence of (5) can be any simply-connected domain in \( \mathbb{C} \) with analytic boundary. In particular, if \( \{\alpha_k\} \) has only a limit point at infinity, if \( \sum_{k=m}^{\infty} 1/|\alpha_k| < \infty \), and if (5) converges at least one point \( \alpha_0 \neq \alpha_k, k = 0, 1, \ldots \), then (5) converges uniformly in any disc \( |z| \leq R \) and, hence, its sum \( q(z) \) is an entire function. Stirling's interpolation series is a particular case of Newton's, for the sequence of nodes \( \alpha_0 = 0, \alpha_1 = -1, \alpha_2 = 1, \ldots, \alpha_2k-1 = -k \), \( \alpha_2k = k, \ldots \). Other, similar, interpolation series have been investigated.

Interpolation processes with non-algebraic interpolation polynomials \( p_n(z) = \sum_{\nu=0}^{\nu=n} b_\nu \phi_\nu(z) \), constructed in systems of functions \( \{\phi_\nu(z)\} \) other than \( \{z^\nu\} \), e.g. in \( \{\alpha^\nu \nu^z\} \), are also an object of study.

The study of interpolation processes in the real domain has its own specifics, both in the formulation of problems as in the results. These specifics, first of all, are brought about by the natural (in the real domain) requirement of regularity of the function \( f(z) \) to be interpolated. It is known, e.g., that there is no system of nodes on \([a, b]\) that would guarantee the convergence of the interpolation processes for arbitrary continuous functions \( f(x) \), \( x \in [a, b] \). On the other hand, if a continuous function \( f(x) \) is given in advance, it is always possible to choose a system of nodes such that the interpolation process converges to \( f(x) \).

Besides interpolation processes with polynomials \( p_n(z) \), interpolation processes with rational functions \( r_n(z) \), e.g. of the form \( r_n(z) = q_n(z) / \omega_{n-1}(z) \), where \( \omega_{n-1}(z) = (z - b_0) \ldots (z - b_{n-1}) \) and \( q_n(z) \) is a polynomial of degree at most \( n \), have drawn the attention of researchers. The interpolation conditions (1)–(3) remain in force, but conditions at the poles \( b_k, k = 0, 1, \ldots \), which in the simple case are given by a triangular table \( \{b_{jk}\}, 0 \leq k \leq j, j = 0, 1, \ldots \), similar to (1) must be given.
Formulas for numerical differentiation, with interpolation at their basis, are obtained from (3) depending on the choice of $L_n(x)$. For numerical differentiation one uses, as a rule, approximate values of the function at nodes; the error in a formula of numerical differentiation not only depends on the manner of interpolating and the interpolation step, but also on the errors in the values of the function at the nodes which are used. E.g., in the case of the linear approximation (1)

$$f'(x_k) \approx \frac{1}{h} [f(x_{k+1}) - f(x_k)], \quad (4)$$

and the remainder $R_1^f(x)$ has the representation

$$R_1^f(x_k) = -\frac{1}{2} f''(\xi) h, \quad \xi \in (x_k, x_{k+1}), \quad h = x_{k+1} - x_k.$$

If $f(x_{k+1})$ and $f(x_k)$ are known with respective errors $\epsilon_k + 1, \epsilon_k, \epsilon_{k+1} \neq \epsilon_k$, then the error in (4) will contain yet another term $(\epsilon_{k+1} - \epsilon_k)/h$, which decreases as the step $h$ increases. When using formulas for numerical differentiation, the interpolation step must be in accordance with the errors in the values of the functions. Therefore, in practice it often happens that a function known to have some error on a dense grid is interpolated on a more coarse grid only.

**Numerically solving integral equations.**

The unknown function $\phi(x)$ in the integral equation is replaced by some interpolation formula (an interpolation polynomial, an interpolation spline, etc.) with nodes $x_k$, and an approximate value $\phi(x_k)$ is determined from the system obtained after replacing the independent variable $x$ by the nodes $x_k$. E.g., for the linear Fredholm integral equation of the second kind

$$\phi(x) = \lambda \int_a^b K(x, s) \phi(s) \, ds + f(x) \quad (5)$$

one can use the Lagrange form of the interpolation polynomial,
\[ \phi(x) = \sum_{k=0}^{n} l_k(x) \phi(x_k) + R_n(x) = L_n(x) + R_n(x), \]

Where \( R_n(x) \) is the remainder and

\[ l_k(x) = \frac{\omega_n(x)(x-x_k)}{\omega'_n(x_k)}, \quad \omega_n(x) = \prod_{k=0}^{n} (x-x_k). \]

Replacing \( \phi(x) \) in (5) by its interpolation polynomial \( L_n(x) \) and \( x \) by \( x_i \), one obtains the linear system of equations

\[ \phi_i = \lambda \sum_{k=0}^{n} M_k(x_i) \phi_k + f(x_i), \]

\[ M_k(x_i) = \int_{x}^{b} K(x_i, s) l_k(s) \, ds, \quad i = 0, \ldots, n, \]

for determining the approximate value \( \phi_i \) of \( \phi(x) \) at \( x_i \). In the case of non-linear integral equations the approximate value \( \phi_i \) has to be determined from the corresponding non-linear system.

**Numerically solving differential equations.**

One of the possibilities for constructing numerical methods for solving differential equations consists of replacing the derivatives of the unknown functions by interpolation formulas for numerical differentiation, and in a number of cases also in replacing by interpolation other functions and expressions occurring in the equation.

Suppose one has the following formula for numerical differentiation with equally-spaced nodes \( x_k = x_0 + nh \):

\[ y'(x_k) \approx \frac{y(x_k+1) - y(x_k)}{h} = \frac{1}{h} \Delta y(x_k), \]

\[ y''(x_k) \approx \frac{1}{h^2} [y(x_{k-1}) - 2y(x_k) + y(x_{k+1})] = \frac{1}{h^2} \Delta^2 y(x_k), \]

(6)
References


