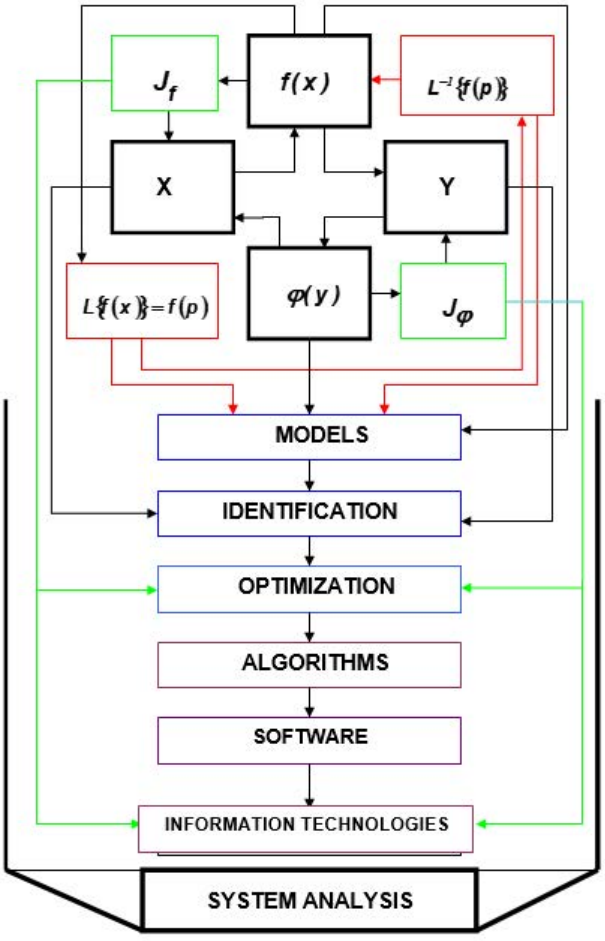
B. I. Mokin, V. B. Mokin, O. B. Mokin

FUNCTIONAL ANALYSIS IN INFORMATION TECHNOLOGIES



Ministry of Education and Science of Ukraine
Vinnytsia National Technical University

B. I. Mokin, V. B. Mokin, O. B. Mokin

FUNCTIONAL ANALYSIS IN INFORMATION TECHNOLOGIES

Textbook

Vinnytsia
VNTU
2024

UDC 517.98
M74

Recommended by the Academic Council of the Vinnytsia National Technical University of the Ministry of Education and Science of Ukraine as a textbook for English-speaking students and postgraduates specializing in the field of information technologies (record No. 9 of 27.02.2024).

Reviewers:
V. Ya. Danilov, Doctor of Technical Sciences, professor (NTUU "KPI named after Sikorsky")
V. I. Klochko, Doctor of Pedagogical Sciences, Professor (VNTU)
O. S. Makarenko, Doctor of Sciences in Physics and Mathematics, professor (NTUU "KPI named after Sikorsky")

Mokin, B. I.
M74 Functional analysis in information technologies: textbook [Electronic resource] / B. I. Mokin, V. B. Mokin, O. B. Mokin, Vinnytsia: VNTU, 2024, (PDF, 129 p.)
ISBN 978-617-8163-06-8 (PDF)
The textbook outlines the basics of functional analysis adapted to the solution of applied problems in the field of information technology using programs implemented in the Python language.
The textbook is recommended for English-taught students, post-graduate students specializing in the IT field in specialties 124 – "System Analysis" and 126 – "Information Systems and Technologies".
UDC 517.98

ISBN 978-617-8163-06-8 (PDF)

CONTENT

INTRODUCTION

Higher mathematics for non-mathematical specialties in higher education institutions contains five mathematical components with different level of detail - linear algebra, mathematical analysis, probability theory, functions of a complex variable and vector algebra. But this is not enough for the effective learning of some special subjects in the field of information technologies, and therefore the curricula of some of these specialties contain other mathematical components, among which an important role is played by the mathematical component called "Functional analysis" and which, in fact, is the "second floor" over the "Mathematical Analysis" component.

It is known from mathematical analysis that a function is a law according to which one numerical set corresponds to another numerical set.

Graphically, this can be displayed as shown in fig. B.1.

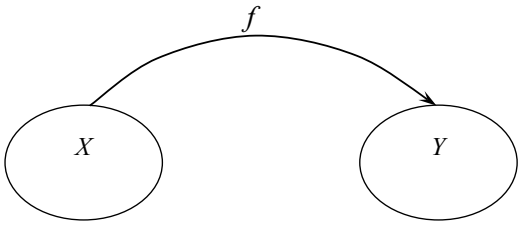


Figure B.1 – Graphical interpretation of the term function

Conventionally, the function is most often written as follows

$$y = f(x),\ \ x \in X,\ \ y \in Y, \tag{1}$$

or

$$y = y(x),\ \ x \in X,\ \ y \in Y,, \tag{2}$$

where $\in$ –is the symbol of the element belonging to the set.

If the function *f* assigns only one number $y \in Y$ to each number $x \in X$, then, as is known from mathematical analysis, such a function is called a *single-valued*, and if the function assigns two or more numbers to each number, then such function is called a *multi-valued*.

A function can be specified in the form of a table, a graph, or one or more formulas.

A function the graph of which has no discontinuities belongs to the continuous class, and a continuous function the graph of which does not contain breaks and therefore has a continuous first derivative belongs to the *smooth* class.

A continuous function whose graph has breaks, and therefore its derivative - the breaks of the 1st type, belongs to the class of piecewise smooth.

From the same subject of "Mathematical analysis" it is known that a functional is a law according to which a set of functions is matched to a set of numbers.

Graphically, it looks as shown in fig. B.2.

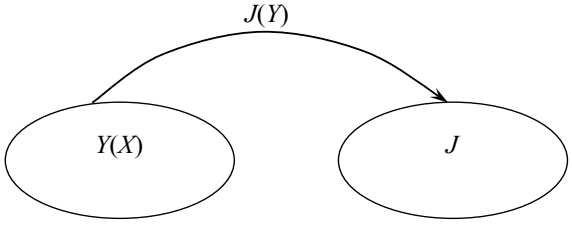


Figure B.2 – Graphical interpretation of the concept of functional

Conventionally, the functional is most often written as follows

$$J_y = J\big(y(x), x\big),\ \ x \in X,\ \ y(x) \in Y(X),\ \ J_y \in J\,. \tag{3}$$

Examples of functional can be definite integrals:

$$J_y = \int_a^b y(x)\,dx \tag{4}$$

or

$$J_y^f = \int_a^b f(x, y)\,dx\,, \tag{5}$$

or

$$J_y^F = \int_a^b F(x, y, y')\,dx\,, \tag{6}$$

in which $f(x, y)$ – is a mathematical expression that is a construction from an independent variable x and its function $y(x)$, and $F(x, y, y')$ – is a mathematical expression that is a construction from an independent variable x, its function $y(x)$, and the first derivative $y'(x)$ of this function; at the same time, the segment $[a, b]$ is the domain of the function $y(x)$, i.e. $x \in [a, b]$.

So the function sets the law according to which each element from one numerical set is matched with some element from another or the same numerical set, and the functional sets the law according to which each element from the set of functions is matched with some element from the set of numbers.

Then, there is a question: "Is it not possible to find a law according to which each element from a set of functions is matched by some element from another or the same set of functions?"

The answer to this question is positive, and the mathematical concept that characterizes such a law is called an *operator* in mathematics.

For example, between a set of continuous functions on a segment $[a, b]$ and a set of derivatives of these functions $f(t), t \in [a, b]$ there is a one-to-one correspondence, which is given by the differentiation operator $D = \dfrac{d}{dt}$, for example, the function

$$y = t^2 \tag{7}$$

corresponds to the derivative

$$\frac{dy}{dt} = 2t, \tag{8}$$

which is also a function of the same independent variable.

Analyzing the program of the educational subject "Mathematical analysis", it is easy to realise that this mathematical discipline is dedicated to the study of the properties of functions and operations with them, the main of which are differentiation and integration. And it does not pay attention at all to the study of the properties of functional and operators as independent mathematical objects. Therefore, a separate mathematical component called "Functional analysis" is dedicated to the study of these mathematical objects, which is studied

by students of all mathematical specialties at universities and which is also included in the list of mandatory mathematical disciplines for students and postgraduates of some IT specialties.

The mathematical discipline “Functional analysis”, which is studied by students of mathematical specialties at universities, is a set of concepts and theorems that combine these concepts into a single mathematical structure, and therefore it contains 90 percent of the material dedicated to the formulation and proof of these theorems . At the same time, it is more important for IT students to be able to use this material in practical applications. That is why we built our study guide using material dedicated mainly to the presentation of the main concepts and final results obtained in the theory of functional analysis and their application to the solution of the applied problems that IT specialists face with. The program material of the subject is presented in six chapters, the first of which is dedicated to sets, metric spaces and their characteristics; the second – theories of measure and integrals of Riemann, Lebesgue and Stieltjes; the third – to functional and methods of finding their unconditional extrema; the fourth – methods of finding conditional extrema of functional; the fifth – theory and applied aspects of the use of operators; the sixth – characteristics and recommendations for the application of several special operators, such as direct and inverse Laplace operators and autoregressive operators, which are widely used in system analysis and applied information technologies.

In conclusion to this brief introduction to functional analysis it is necessary to note that in the English-language version of the textbook the authors used all the references listed in the bibliography, but without specification of the source, which is typical of monographs and scientific papers. And the material which is taken from the Ukrainian-language manuals, written by the authors themselves about the basics of functional analysis and some specific subjects in which the concepts of functional analysis are used, which we use in this textbook to demonstrate the solution of specific applied problems, we present without quotation marks and references.

The differences between this textbook and other study guides on functional analysis is, first of all, in a different structuring of the study material and its selection since this textbook is focused on solving those applied problems, that a specialist in information technologies confronts with. Moreover, each applied problem is accompanied by the developed computer program for implementing its algorithms in the Python language.

Chapter 1. SETS AND METRIC SPACES, THEIR CLASSES AND CHARACTERISTICS

1.1 Sets, subsets and their characteristics

The concept of a **set** in mathematics is understood as a collection of objects of a certain nature, which are called its elements. A **set** is given if all its elements and the rule according to which the elements belong to the **set** are known.

The elements of the **set** can be, for example, all the rivers flowing through Ukraine, or all natural numbers on the number line, or all real numbers located on the segment [0,1] of the number line, or all continuous functions whose arguments are given on this segment of the number axis.

In mathematics, **sets** are denoted by uppercase letters of the Latin or Greek alphabets, and their elements are denoted by lowercase letters from the same alphabets, for example, A, B, X, Z, E, Φ, Ω, Ψ – are **sets**, and a, b, x, z, ε, ϕ, ω, ψ – are elements. A symbolic entry indicates $x \in X$, that an element belongs to a **set**, and a symbolic entry indicates $x \notin X$ – that it does not belong to a **set**. A **set** with a finite number of elements is called a **finite set**, and a **set** with an infinite number of elements is called an **infinite set**. An example of a **finite set** is the **set** of cars registered in Ukraine, and an example of an **infinite set** is the **set** of real numbers on the segment [0,1] of the number axis. If the elements of the **set** A are a finite numerical sequence with n members, then symbolically it can be written as $A = \{a_i\}$, $i = 1, 2, ..., n$. If the elements of this set A are an infinite numerical sequence of members, then it can be symbolically written in the form $A = \{a_i\}$, $i = 1, 2, 3...$. A **Set** that does not contain any element, is called an **empty set** and is denoted by the symbol $\emptyset$ or O, which does not need to be equated with the number "zero".

If the **sets** A and B consist of the same elements, then they are considered equal, as evidenced by the record $A = B$. If not all the elements of the **set** A are included in the **set** B, then the **set** A is called a **subset** of the **set** B, as evidenced by the record $A \subset B$. For example, on the number line, the **set** of rational numbers R, each of which is known to be the ratio of two integers, is a **subset** of the **set** Z of real numbers. If we are not sure that the **subset** of the **set** A contains fewer elements than the **set** B, then we write it like this: $A \subseteq B$.

When two **sets** A and B are combined, a new **set** M is formed, which is called their **sum** and which contains all the elements of both of these **sets**, and each identical element of both **sets** is included in their **sum** M as one element - symbolically, the sum is written as follows:

$$M = A \cup B \tag{1.1}$$

For example, if A and B and are numerical **sets**, where

$$A = \{1,2,3,4,5\}, \quad B = \{4,5,6,7,8\}, \tag{1.2}$$

then, according to (1.1), we will have

$$M = A \cup B = \{1,2,3,4,5\} \cup \{4,5,6,7,8\} = \{1,2,3,4,5,6,7,8\} \tag{1.3}$$

At the intersection of two **sets** A and B, a new **set** P is formed, which is called their intersection and which contains only those elements of both **sets** that are the same, and each of these identical elements of both **sets** is included in their intersection as one element - the intersection is symbolically written as follows:

$$P = A \cap B \tag{1.4}$$

For numerical **sets** (1.2) given in the conditions of the previous example, according to (1.4), we have

$$P = A \cap B = \{1,2,3,4,5\} \cap \{4,5,6,7,8\} = \{4,5\} \tag{1.5}$$

For the sum and the intersection of **sets** $A,\ B,\ C$ the following properties are valid:

Asociality

$$(A \cup B) \cup C = A \cup (B \cup C), \tag{1.6}$$

$$(A \cap B) \cap C = A \cap (B \cap C), \tag{1.7}$$

Commutativity

$$A \cup B = B \cup A, \tag{1.8}$$

$$A \cap B = B \cap A, \tag{1.9}$$

Distributiveness

$$(A \cup B) \cap C = (A \cap C) \cup (B \cap C), \tag{1.10}$$

$$(A \cap B) \cup C = (A \cup C) \cap (B \cup C) \tag{1.11}$$

And for the sum and intersection of a **set** A with itself and with its **subset** B, the relations are valid

$$A \cup A = A, \tag{1.12}$$

$$A \cap A = A, \tag{1.13}$$

$$A \cup B = A, \tag{1.14}$$

$$A \cap B = B \tag{1.15}$$

The **set** Q consisting of the elements of the **set** A that are not included in the **set** B is called the difference of these **sets** and is denoted as $A - B$ or $A \setminus B$, i.e.

$$Q = A - B = A \setminus B \tag{1.16}$$

It is quite obvious that in the general case

$$A - B \neq B - A \tag{1.17}$$

For example, for numerical **sets** (1.2)

$$A - B = \{1,2,3\}, \tag{1.18}$$

$$B - A = \{6,7,8\} \tag{1.19}$$

If the **set** A is a **subset** of the **set** B, then the difference $B - A$ is called the complement of the **set** A to the **set** B and is symbolically denoted as $C_B A$, i.e.

$$C_B A = B - A \tag{1.20}$$

For example, the **set** $\overline{R}$ of irrational numbers on the number line is the complement of the **set** R of rational numbers to the **set** Z of real numbers, i.e.

$$\overline{R} = C_Z R = Z - R \tag{1.21}$$

If the **sets** $A_i,\ i = 1,\ 2,\ ...,\ n$ are **subsets** of the **set** A, then the relations are valid

$$C_A A_1 \cup C_A A_2 \cup ... \cup C_A A_n = C_A (A_1 \cap A_2 \cap ... \cap A_n), \tag{1.22}$$

$$C_A A_1 \cap C_A A_2 \cap ... \cap C_A A_n = C_A (A_1 \cup A_2 \cup ... \cup A_n), \tag{1.23}$$

the correctness of which is easy to verify graphically, for example, for n = 3, if the set A is represented in the figure as a square with three circles inscribed in it, representing the subsets of A_1, A_2, A_3.

An important characteristic of **sets** is their **equivalence**, according to which **sets** *A,B* are considered as **equivalent** if, according to some rule, each element $a \in A$ is matched by one unique element $b \in B$, and each element $b \in B$ is matched by one unique element $a \in A$. For example, a **set** *A* of privately owned passenger cars, each of which is registered to only one owner in a certain settlement, and a **set** *B* of people who own these cars are equivalent. The rule by which the equivalence of these sets is established is the entry of the owner's surname in the vehicle passport.

And in order to compare **non-equivalent sets**, the concept of their **power** is introduced, which for the **set** A is symbolically written as $\overline{\overline{A}}$ and which is determined by something common that occurs in all **sets equivalent** to the one under consideration. It is obvious that **finite sets** of different natures have only the number of their elements in common, and therefore, if a **set** *A* has ***n*** of elements, and a **set** *B* has ***m*** of elements and at the same time $n > m$, then we state that the **set** *A* has a power greater than the **set** *B*.

But there arises a question: "And how to compare the **power of infinite sets**, each of which has an infinite number of elements?".

In mathematics, it is established that of all infinite sequences, the **natural series** *N* approaches infinity the fastest since each of its subsequent numbers is equal to the previous number increased by one, and at the same time, when forming this series, all real numbers that are contained on the number axis in each such unit are omitted. And therefore the **natural series**, which is an infinite series of numbers, is **an infinite set of the lowest power, symbolically denoted by a small Latin letter** a, that is,

$$\overline{\overline{N}} = a, \tag{1.24}$$

and all other **infinite sets** will be compared among themselves, based on how they are related by **power** to the **power of the natural series**, determined by the relation (1.24). And all **infinite sets** with the **power of a natural series** are called **countable sets**, since each of their elements can be assigned an index equal to the corresponding number of the natural series, due to which each of their elements can be **counted**.

And the first fact that was established in mathematics after the agreement regarding the **power of the natural series** is that **the power of the set *Z* of real number**s given on the interval [0,1] is **bigger than** a.

The proof of this fact is simple - if you add a sequence of real numbers x_1, x_2, x_3, ... on the segment [0,1] of the numerical axis so that each subsequent number is three times smaller than the previous one, and divide this segment [0,1] into three equal segments Δ_1 each with a width of at least one a point x_1 will not enter from these segments. Let's divide the segment in which the point x_1 did not enter, also into three equal segments of width Δ_2 each, and choose the one from them in which the point x_2. did not enter. According to this algorithm, we will continue this process ad infinitum. As a result, on the segment [0,1] of the numerical axis, we will receive a **counted set** Δ_1, Δ_2, Δ_3, ...,, the elements of which are smaller and smaller segments of the segment, and next to which, on the same segment, there is a previously calculated set of numbers x_1, x_2, x_2, ...,**,**, none of which falls into any of these segments . And this means that there are more real numbers on the segment [0,1] of the number axis than there are numbers of the natural series on the entire number axis, which allows us to conclude that the **power of an infinite set of real numbers on the segment [0,1] is greater than the power of the natural series, which is a countable set.**

In mathematics, **the power of an infinite set of real numbers on the interval [0,1] is called the power of a continuum denoted by a lowercase Latin letter *c***, therefore, the inequality is valid

$$c > a. \tag{1.25}$$

Moreover, it was established that for powers *c, a* the inequality (1.25), as well as the equality

$$c = 2^a. \tag{1.26}$$

are real.

To prove the equality (1.26), we will use the method of mathematical induction according to the algorithm: we will consider successively which powers n_0, n_1, n_2, n_3 will have sets generated by finite sets $A_0 = \{O\}, A_1 = \{a_1\}, A_2 = \{a_1, a_2\}, A_3 = \{a_1, a_2, a_3\}$, if all **possible subsets** generated by the elements of these **finite sets** are introduced as elements into each of the generated sets. Bearing in mind that the number of combinations of elements of C from the *n* of the elements on *m, as* is known from the high school mathematics, shall be determined according to correlation

$$C_n^m = \frac{n!}{m!\,(n-m)!}, \tag{1.27}$$

we will find that:

$$\begin{cases} n_0 = C_0^0 = \dfrac{0!}{0!0!} = 1 = 2^0, \\ n_1 = C_1^0 + C_1^1 = \dfrac{1!}{0!1!} + \dfrac{1!}{1!0!} = 1 + 1 = 2 = 2^1, \\ n_2 = C_2^0 + C_2^1 + C_2^2 = \dfrac{2!}{0!2!} + \dfrac{2!}{1!1!} + \dfrac{2!}{2!0!} = 1 + 2 + 1 = 4 = 2^2, \\ n_3 = C_3^0 + C_3^1 + C_3^2 + C_3^3 = \dfrac{3!}{0!3!} + \dfrac{3!}{1!2!} + \dfrac{3!}{2!1!} + \dfrac{3!}{3!0!} = 1 + 3 + 3 + 1 = 8 = 2^3. \end{cases} \tag{1.28}$$

According to the **ideology of the method of mathematical induction**, it **follows from relations (1.28) that if a finite set $A_n = \{a_1, a_2, a_3, ..., a_n\}$ has *n* elements, then the power n_n** of **the set generated by it**, which includes **all possible subsets of this set, will be equal to**

$$n_n = 2^n. \tag{1.29}$$

And hence the conclusion that if the **power of the counted set is a, equal to the power of the infinite set of real numbers generated by it on the interval [0, 1], the elements of which are all possible subsets of the elements of this set, will be equal to two to the power of a,** which proves the validity of the equality (1.26).

And now let's return to the expression (1.21), according to which the **set *Z* of real numbers on the segment [0, 1]** of the numerical axis is the **sum of the subset *R* of rational numbers and the subset $\overline{R}$ of irrational numbers** given on the same segment.

As is known, each rational number is the ratio of two integers, and if this rational number is less than one, then its numerator is always an integer that is smaller than the number in the denominator. Since the integers are elements of the natural series, which is a countable power set, then these numbers can be counted both in the numerator and in the denominator, and **therefore the subset of rational numbers on the segment [0, 1] of the**

number axis is also a countable set of power *a*. The above fact has **two consequences**, **the first** of which proves that **the subset of irrational numbers on the specified segment is an infinite set of the power of the continuum** *c*, because only due to this subset the set of real numbers on the specified segment will have the power c that we have already shown above with respect to the set of real numbers. And **the second consequence is the statement that if any counted subset is added to the power set of the continuum, the power of their sum remains equal** *c*.

And then we pay attention to the fact that all unit segments on the number axis, located between adjacent natural numbers, can be counted by assigning to each of them an index equal to the natural number placed on the right border of each such unit segment, so a subset of unit segments, placed between natural numbers on the number axis, is a counted set of power $a,$, which is a smaller power of the continuum of the unit segment [0, 1] of the number axis. So, based on this statement, we can draw an important conclusion that **the entire axis of real numbers is a multiple of the power of the continuum** *c***.**

But, as we have already shown above, the **set that is generated by the union of all possible subsets of the generic set of a certain power has a power equal to two to the power equal to the power of the generic set**. And from this fact we draw the conclusion: **the power** *f* **of the set of all functions** *f(x)* **the argument** *x* **of which is set on the segment [0, 1]** (or on the entire numerical axis) **of the power of the continuum** *c* **is equal to two in the power of** *c***, that is**,

$$\bar{\bar{f}} = 2^c \tag{1.30}$$

This is where we will finish the consideration of the material of the subsection dedicated to sets, subsets and their characteristics, which we will need when explaining the basics of functional analysis. Those who wish to learn more about this area of mathematics are referred to textbooks on set theory or functional analysis, which are used by students of mathematical specialties at universities.

1.2 Metric spaces and their classes and characteristics

Set

$$\Omega = \{x, y, z, \ldots, u, v, \ldots\} \tag{1.31}$$

of the elements of some nature are called **metric space** if each ordered pair of elements $x, y \in \Omega$ is in line with an integral number $\rho(x, y)$, which is called the **metric of space** Ω, if this number satisfies three axioms of metrics:

1) axiom identity

$$\rho(x, y) = 0 \tag{1.32}$$

then and only then, when

$$x = y; \tag{1.33}$$

2) axiom symmetry

$$\rho(x, y) = \rho(y, x); \tag{1.34}$$

3) the triangle axiom

$$\rho(x, y) + \rho(y, z) \geq \rho(x, z). \tag{1.35}$$

Considering these axioms we see that the **metric** $\rho(x, y)$ of **space** Ω **sets the distance between the elements** x , y **of this space.**

The elements of the metric space are called points.

Examples

1. For a three-dimensional Euclidean space E_3 the distance between points $x = \{x_1, x_2, x_3\}$ and $y = \{y_1, y_2, y_3\}$ $(x, y \in E_3)$ is determined by an expression

$$\rho(x, y) = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2}. \tag{1.36}$$

2. For the set $C[0,1]$ of continuous functions $x(t)$, $y(t)$, …, given on the segment $t \in [0,1]$, the distance between the elements $x(t)$ and $y(t)$ is given by the expression

$$\rho(x, y) = \max_t |x(t) - y(t)|. \tag{1.37}$$

If X – an arbitrary metric space then the sequence

$$\{x_n\} \subset X \tag{1.38}$$

coincides to a point $x_0 \in X$, if when $n \to \infty$

$$\rho(x_n, x_0) \to 0, \tag{1.39}$$

or, as written otherwise

$$\lim_{n\to\infty} x_n = x_0. \tag{1.40}$$

The sequence $\{x_n\}$, that coincides to some point x_0, is limited.

If the set contains all its limit points, then it is closed.

Let the metric space X is given, and let there be a sequence of points $\{x_n\}$ in this space that coincides to the point $x_0 \in X$. Then, when $n \to \infty$ the expression (1.39) will be fair as well as the expression

$$\rho(x_{n+p}, x_0) \to 0, \tag{1.41}$$

for and any $p > 0$. And the inequality of the triangle (1.35) using expressions (1.39) and (1.41) takes a form of

$$\rho(x_{n+p}, x_0) + \rho(x_n, x_0) \geq \rho(x_{n+p}, x_n). \tag{1.42}$$

And from expressions (1.39), (1.41) and (1.42) due to the inequality of a triangle for metrics, we will have an expression

$$\rho(x_{n+p}, x_n) \to 0. \tag{1.43}$$

If a condition (1.43) is fulfilled for some sequence $\{x_n\} \subset X$, then it is called a **fundamental sequence** or sequence that coincides in itself or a **sequence of Cauchy**.

If in the metric space ***X***, any sequence $\{x_n\} \subset X$,that coincides to itself, coincides to some limiting point x_0 , which is an element of the same space, that is $x_0 \in X$, then this space ***X*** is called **complete**.

Metric space *X* is called linear if it defines the operations of addition and multiplication by the scalar, which satisfy the following conditions:

1) $$x^* + x^{**} = x^{**} + x^*, \ \forall\, x^*, x^{**} \in X; \tag{1.44}$$

2) $$\left(x^* + x^{**}\right) + x^{***} = x^* + \left(x^{**} + x^{***}\right), \ \forall\, x^*, x^{**}, x^{***} \in X; \tag{1.45}$$

3) $$x + 0 = x, \ \forall\, x \in X, 0 \in X, \tag{1.46}$$

where the element **0** is zero of the set **X**;

4) for $\forall\, x^* \in X$ there is $x^{**} \in X$ such that

$$x^* + x^{**} = 0, \tag{1.47}$$

where the element x^{**} is an element opposite to the element x^*;

5) $$1 \cdot x = x, \ \forall\, x \in X; \tag{1.48}$$

6) $$\alpha \cdot \left(\beta \cdot x\right) = \left(\alpha \cdot \beta\right) \cdot x, \ \forall\, x \in X \text{ and } \forall\, \alpha, \beta; \tag{1.49}$$

7) $$\left(\alpha + \beta\right) \cdot x = \alpha \cdot x + \beta \cdot x, \ \forall\, x \in X \text{ and } \forall\, \alpha, \beta; \tag{1.50}$$

8) $$\alpha \cdot \left(x^* + x^{**}\right) = \alpha \cdot x^* + \alpha \cdot x^{**}, \ \forall\, x^*, x^{**} \in X \text{ and } \forall\, \alpha. \tag{1.51}$$

A linear metric space ***X*** is called normalized if $\forall\, x \in X$ it can be matched by some non-negative number $\| x \|$, which is called the norm and which satisfies the following conditions:

1) $$\| x \| = 0 \ \text{ if and only if } \ x = 0; \tag{1.52}$$

2) $$\| \alpha \cdot x \| = | \alpha | \cdot \| x \|, \quad \alpha \text{ is a scalar}; \tag{1.53}$$

3) $$\| x^* + x^{**} \| \le \| x^* \| + \| x^{**} \|, \ \forall\, x^*, x^{**} \in X. \tag{1.54}$$

It is quite obvious that the norm $\| x \|$ is the distance from the element ***x*** to the zero element of the set ***X***.

Examples of norms:

1) for space $C[0,1]$

$$\| x(t) \| = \max_{t \in [0,1]} | x(t) | \tag{1.55}$$

or

$$\| x(t) \| = \sup_{t \in [0,1]} | x(t) |; \tag{1.56}$$

2) for the Euclidean dimension E_n of the space n

$$\| x \| = \sqrt{\sum_{i=1}^{n} x_i^2}\ , \tag{1.57}$$

where $x = \{x_1,\ x_2,\ \ldots,\ x_n\},\ x \in E_n$.

It is obvious that for any linear normalized space ***X*** the relation is valid

$$\| x^* - x^{**} \| = \rho\left(x^*, x^{**} \right), \tag{1.58}$$

where $x^*, x^{**} \in X$.

A complete linear normalized space is called Banach (after the name of the mathematician who studied this space) and is denoted as B-space.

It is clear that the spaces $C[0,1]$ and E_n are Banach.

Note that the norm in B-space can be introduced in different ways, so long as it meets the conditions (1.52), (1.53), (1.54).

For example, in the space of functions $x(t)$ continuous on a segment $t \in [0,1]$, the norm can be introduced not only in the form (1.55), but also in the form

$$\| x \| = \int_0^1 |x(t)|\, dt\ . \tag{1.59}$$

Such a B-space is called a Lebesgue space which is denoted by $L[0,1]$, to distinguish it from the space $C[0,1]$ of the same functions, but with norm (1.55).

For space E_n as a norm, you can use not only the ratio (1.72), but also a more general one

$$\| x \| = \left(\sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}},\ \ p > 0\ . \tag{1.60}$$

It is clear that (1.57) coincides to (1.60) for $p = 2$.

A Banach space with a scalar product of elements is called a Hilbert space (after the name of the mathematician who studied it) and is denoted as an H-space.

H-space can be finite-dimensional or infinite-dimensional.

The scalar product of the elements $f, g \in H$ is written in the form (f, g) or $\langle f, g \rangle$. The scalar product must be subject to the following conditions:

1) $$\langle f, g \rangle = \langle g, f \rangle, \tag{1.61}$$

2) $$\langle \alpha \cdot f, g \rangle = \alpha \cdot \langle f, g \rangle, \tag{1.62}$$

3) $$\langle f, \alpha \cdot g \rangle = \alpha \cdot \langle f, g \rangle, \tag{1.63}$$

4) $$\langle f_1 + f_2, g \rangle = \langle f_1, g \rangle + \langle f_2, g \rangle; \tag{1.64}$$

5) $$\langle f, g_1 + g_2 \rangle = \langle f, g_1 \rangle + \langle f, g_2 \rangle; \tag{1.65}$$

6) $$\langle f, f \rangle > 0, \text{ якщо } f \neq 0. \tag{1.66}$$

It follows from the expression for the norm that for the H-space

$$\| f \| = \sqrt{\langle f, f \rangle}. \tag{1.67}$$

H-space is often considered in two implementations.

1. Space l_2 of all counted ordered sequences $x \in l_2$

$$x = \{x_1, x_2, \ldots, x_n, \ldots\} \tag{1.68}$$

such that have the property

$$\sum_{i=1}^{\infty} x_i^2 < \infty. \tag{1.69}$$

For elements: $x^*, x^{**} \in l_2$:

$$\rho\left(x^*, x^{**}\right) = \sqrt{\sum_{i=1}^{\infty} \left(x_i^* - x_i^{**}\right)^2}; \tag{1.70}$$

$$\| x^* \| = \sqrt{\sum_{i=1}^{\infty} \left(x_i^*\right)^2}; \tag{1.71}$$

$$\| x^* - x^{**} \| = \rho\left(x^*, x^{**}\right); \tag{1.72}$$

$$\left\langle x^*, x^{**} \right\rangle = \sum_{i=1}^{\infty} x_i^* \cdot x_i^{**}; \tag{1.73}$$

$$\left\| x^* \right\| = \sqrt{\left\langle x^*, x^* \right\rangle} \tag{1.74}$$

It follows from these relations that l_2-space is a generalization of Euclidean E_n-space when $n \to \infty$.

l_2-space is sometimes called a coordinated Hilbert space.

2. **The space $L_2[a, b]$ of functions $f(t)$ with an integrated square**, that is, for which

$$\int_a^b f^2(t)\, dt < \infty. \tag{1.75}$$

The following relations are valid for $f(t), g(t) \in L_2[a, b]$:

$$\rho(f, g) = \sqrt{\int_a^b \left(f(t) - g(t)\right)^2 dt}; \tag{1.76}$$

$$\| f\| = \sqrt{\int_a^b f^2(t)\, dt}\,; \tag{1.77}$$

$$\| f - g\| = \rho(f, g); \tag{1.78}$$

$$\langle f, g\rangle = \int_a^b f(t)\cdot g(t)\, dt\,. \tag{1.79}$$

Let's write two widely used inequalities separately:

the Cauchy–Minkowski inequality

$$\| f + g\| \le \| f\| + \| g\|, \tag{1.80}$$

the Buniakovsky–Schwartz inequality

$$|\langle f, g\rangle| \le \| f\| \cdot \| g\|, \tag{1.81}$$

to prove which it is enough to substitute expressions for all components in them.

1.3 Orthonormal subsets in Hilbert spaces

Consider a functional Hilbert space $H[a,b]$ such that $x(t), y(t) \in H[a,b]$, $t \in [a,b]$.

Let the scalar product $\langle x, y\rangle$ of functions $x(t)$ and $y(t)$ equal to zero, that is,

$$\langle x, y\rangle = \int_a^b x(t)\cdot y(t)\, dt = 0\,. \tag{1.82}$$

If the condition (1.82) is satisfied for the functions $x(t), y(t) \in H[a,b]$, then they are said to be orthogonal on $[a,b]$.

Let us have in the Hilbert space $H[a,b]$ a finite-dimensional or infinite sequence of functions $\{\varphi_k(t)\}$ such that

. $$\{\varphi_k(t)\} \subset H[a,b],\ t \in [a,b]. \tag{1.83}$$

If the condition is true for this sequence $\{\varphi_k(t)\}$

$$\langle \varphi_k, \varphi_m\rangle = \int_a^b \varphi_k(t)\cdot \varphi_m(t)\, dt = 0,\ k \ne m, \tag{1.84}$$

then this sequence is called orthogonal.

If the condition is satisfied for an orthogonal sequence $\{\varphi_k(t)\} \subset H[a,b]$

$$\int_a^b \varphi_k^2(t)\,dt = 1, \tag{1.85}$$

then this sequence is called orthonormal.

A sequence $\{\varphi_k(t)\} \subset H[a,b]$ is called orthogonal with weight $w(t)$, if there exists a function $w(t) \in H[a,b]$, that satisfies the condition

$$\int_a^b \varphi_k(t) \cdot \varphi_m(t) \cdot w(t)\,dt = 0,\ k \neq m. \tag{1.86}$$

It is clear that the sequence $\left\{\sqrt{w(t)} \cdot \varphi_k(t)\right\} \subset H[a,b]$ is simply orthogonal.

A subset of orthogonal functions $\{\varphi_k(t)\} \subset H[a,b]$ is complete in H-space if there is no nonzero function in it that would be orthogonal to any of the functions of this sequence.

A sequence of functions $\{\varphi_k(t)\} \subset H[a,b]$ **is called closed in H-space if for** $\forall\, f(t) \in H[a,b]$ **and for** $\forall\, \varepsilon > 0$ **it is possible to construct such a linear combination of functions** $\varphi_k(t)$, **taken with weight** λ_k, **that the condition is fulfilled**

$$\left\| f(t) - \lambda_1 \cdot \varphi_1(t) - \lambda_2 \cdot \varphi_2(t) - \cdots - \lambda_k \cdot \varphi_k(t) - \cdots \right\| < \varepsilon. \tag{1.87}$$

This means that with an error that does not exceed ε, the function $f(t) \in H[a,b]$ on the segment $[a,b]$ can be presented in the form

$$f(t) \cong \sum_{k=1}^{N} \lambda_k \cdot \varphi_k(t), \tag{1.88}$$

where ***N*** can be either a finite integer or infinity.

Different mathematics for basic functions

$$f_k(t) = t^k,\ k = \overline{0,n} \tag{1.89}$$

obtained various systems of orthonormal polynomials for different weight functions and orthogonalization intervals. Therefore, it is not necessary to build this sequence yourself every time you need to approximate a function $f(t) \in H[a,b]$ using an orthonormal sequence $\{\varphi_k(t)\} \subset H[a,b]$. It is enough to choose one of those built by others, using a reference book on higher mathematics or a manual on the mathematical theory of processing the results of experiments.

Here are examples of orthogonalization intervals, weighting functions, and normalization factors of the most common systems of orthogonal polynomials (Table 1).

Table 1 - Examples of orthogonalization intervals, weighting functions, and normalization factors for the most common systems of orthogonal polynomials

orthogonal polynomials	orthogonalization intervals	weighting functions $W(t)$	normalization factors
Legendre $P_k(t)$	$t \in [-1,1]$	1	$\int_{-1}^{1} \left(P_k(t)\right)^2 dt = \frac{2}{2k+1}$
Chebyshov I $T_k(t)$	$t \in [-1,1]$	$\left(1-t^2\right)^{-\frac{1}{2}}$	$\int_{-1}^{1} \left(T_k(t)\right)^2 \left(1-t^2\right)^{-\frac{1}{2}} dt = \begin{cases} \frac{\pi}{2}, k \neq 0 \\ \pi, k = 0 \end{cases}$
Chebyshov II $U_k(t)$	$t \in [-1,1]$	$\left(1-t^2\right)^{\frac{1}{2}}$	$\int_{-1}^{1} \left(U_k(t)\right)^2 \left(1-t^2\right)^{\frac{1}{2}} dt = \frac{\pi}{2}$
Laguerra $L_k(t)$	$t \in [0,\infty)$	e^{-t}	$\int_{0}^{\infty} \left(L_k(t)\right)^2 e^{-t} dt = 1$
Laguerra attached $L_k^{(i)}(t)$	$t \in [0,\infty)$	$t^i \cdot e^{-t}$	$\int_{0}^{\infty} \left(L_k^{(i)}(t)\right)^2 t^i e^{-t} dt = \frac{(k+1)!}{k!}$
Ermita $H_k(t)$	$t \in (-\infty,\infty)$	e^{-t^2}	$\int_{-\infty}^{\infty} \left(H_k(t)\right)^2 e^{-t^2} dt = 2^k \cdot k! \cdot \sqrt{\pi}$

Thus, in order to approximate the function $f(t) \in H[a,b]$, $t \in [a,b]$ using an orthonormal system of polynomials $\{\varphi_k(t)\} \subset H[a,b]$, it is necessary, based on the interval of orthogonalization $[a,b]$ and the convenience of the weight function $w(t)$, to select one or another orthonormal system of polynomials from the directory and find the ratio for the general member of the selected system, revealing which one to obtain the number of its members, which is sufficient to ensure the given accuracy of the approximation.

For example, we give an expression for a common member –

$$P_n(t) = \frac{1}{2^n n!} \frac{d^n}{dt^n}(t^2-1)^n, \qquad n = 0,1,2,\dots,N \tag{1.90}$$

and the first 7 members of the orthonormal sequence for Legendre polynomials, the weighting function for which is the function $w(t) = 1$, the orthogonalization interval is the segment [-1,1), the normalization factor has the form $2/(2n + 1)$. Therefore, according to expression (1.90), we will have:

$$\left\{\begin{array}{c} P_0(t)=1, \\ P_1(t)=t, \\ P_2(t)=\frac{1}{2}(3t^2-1), \\ P_3(t)=\frac{1}{2}(5t^3-3t), \\ P_4(t)=\frac{1}{8}(35t^4-30t^2+3), \\ P_5(t)=\frac{1}{8}(63t^5-70t^3+15t), \\ P_6(t)=\frac{1}{16}(231t^6-315t^4+105t^2-5), \\ P_7(t)=\frac{1}{16}(429t^7-693t^5+315t^3-35t) \end{array}\right\} \quad (1.91)$$

As a second example, we give the formula for the general member –

$$T_n(t)=\frac{1}{2^n}\left(\left(t+\sqrt{t^2-1}\right)^n+\left(t-\sqrt{t^2-1}\right)^n\right), \qquad n=1,2,\dots,N \quad (1.92)$$

and the first 7 members of the orthonormal sequence for Chebyshov 1 polynomials, the weighting function for which is the function $w(t)=\sqrt{1-t^2}$, the orthogonalization interval is the segment [-1,1), the normalization factor has the form π for $k=0$ and $\pi/2$ for $k\neq 0$. Therefore, according to expression (1.92), we will have:

$$\left\{\begin{array}{c} T_0(t)=1, \\ T_1(t)=t, \\ T_2(t)=\frac{1}{2}(2t^2-1), \\ T_3(t)=\frac{1}{4}(4t^3-3t), \\ T_4(t)=\frac{1}{8}(8t^4-8t^2+1), \\ T_5(t)=\frac{1}{16}(16t^5-20t^3+5t), \\ T_6(t)=\frac{1}{32}(32t^6-48t^4+18t^2-1), \\ T_7(t)=\frac{1}{64}(64t^7-112t^5+56t^3-7t) \end{array}\right\} \quad (1.93)$$

1.4 Approximation of continuous functions in Hilbert spaces

The approximation of continuous functions is understood as the process of finding an analytical description of a function given by the elements of some set, which may not be a subset of the selected space, in the selected space. For example, a polynomial approximation of a function given in the form of a table.

Let $\{\varphi_k(t)\}\subset H[a,b]$, $t\in[a,b]$, be some complete sequence of orthonormal functions that is closed in this space.

Let $H[a,b]=L[a,b]$ is the H-space of functions $f(t)\in L[a,b]$ for which the condition is satisfied

$$\int_a^b |f(t)|\,dt<\infty, \quad (1.94)$$

and the metric $\rho(f_1, f_2)$ is given by the ratio

$$\rho(f_1, f_2) = \int_a^b |f_1(t) - f_2(t)|\, dt. \tag{1.95}$$

Suppose that the series

$$\sum_k \lambda_k \cdot \varphi_k(t), \tag{1.96}$$

where λ_k is some scalar unknown to us, which converges uniformly to some function $f(t) \in L[a, b]$. This means that for $\forall\, \varepsilon > 0$ exists such that ***m*** for $\forall\, t \in [a, b]$ and $\forall\, n \geq m$ the relation holds

$$\int_a^b \left| f(t) - \sum_{k=0}^{n} \lambda_k \cdot \varphi_k(t) \right| dt < \varepsilon, \tag{1.97}$$

from which it $n \to \infty$ follows that

$$f(t) = \sum_{k=0}^{\infty} \lambda_k \cdot \varphi_k(t). \tag{1.98}$$

To determine the weighting coefficients λ_k, $k = \overline{0, \infty}$, multiply both parts of equation (2.41) by $\varphi_j(t)$ and integrate the result in the range from «a» to «b».. As a result, we get:

$$\int_a^b f(t) \cdot \varphi_j(t)\, dt = \sum_{k=0}^{\infty} \lambda_k \cdot \int_a^b \varphi_k(t) \cdot \varphi_j(t)\, dt. \tag{1.99}$$

Since $\{\varphi_k(t)\} \subset L[a, b]$ it is an orthonormal sequence, relations (1.84) and (1.85) hold for it. Taking this into account, from (1.99) we have

$$\lambda_j = \int_a^b f(t) \cdot \varphi_j(t)\, dt, \;\; j = 0, 1, 2, \ldots. \tag{1.100}$$

The weighting coefficients λ_j are called Fourier coefficients, and their complete sequence $\{\lambda_j\}$ is called the Fourier spectrum of the expansion of a function $f(t) \in L[a, b]$ by an orthonormal system of functions $\{\varphi_j(t)\} \subset L[a, b]$.

The requirement (1.97) of uniform convergence of the series (1.96) to the function $f(t)$ is the so-called “strong convergence requirement”.

But it turns out that in H-space the strong convergence is equivalent to “convergence on the average”, which is a weaker requirement and can be written as

$$\lim_{n \to \infty} \int_a^b \left[f(t) - \sum_{k=0}^{n} \lambda_k \cdot \varphi_k(t) \right]^2 dt = 0. \tag{1.101}$$

We consider the process of approximating a function $f(t) \in L_2[a, b]$ in H-space $L_2[a, b]$ using an orthonormal sequence $\{\varphi_k(t)\} \subset L_2[a, b]$.

In this case, the approximation problem can be reduced to such a selection of partial sum coefficients C_k

$$S_n(t) = \sum_{k=0}^{n} C_k \cdot \varphi_k(t) \tag{1.102}$$

in the H-space $L_2[a,b]$ so that this sum approaches the function $f(t) \in L_2[a,b]$ with an error not exceeding the given one, i.e. so that

$$\left\| f(t) - S_n(t) \right\| = \sqrt{\int_a^b \left[f(t) - S_n(t) \right]^2 dt} \to \min_{C_k}. \tag{1.103}$$

To find $\min\limits_{C_k}$ of the expression (1.103), we compose and solve the system of equations

$$\frac{\partial E}{\partial C_k} = 0, \; k = \overline{0,n}, \tag{1.104}$$

where

$$\begin{aligned} E &= \int_a^b \left[f(t) - S_n(t) \right]^2 dt = \\ &= \int_a^b f^2(t)\, dt - 2 \cdot \int_a^b f(t) \cdot S_n(t)\, dt + \int_a^b \left[S_n(t) \right]^2 dt. \end{aligned} \tag{1.105}$$

As a result of solving the system of equations (1.104), we find that

$$C_k = \lambda_k. \tag{1.106}$$

Therefore, in order for the partial sum $S_n(t)$ to approximate the function $f(t)$ with the specified accuracy, it is necessary to choose the Fourier coefficients λ_k of the function $f(t)$ as coefficients C_k.

Substituting (1.106) into (1.105), we will have:

$$E = \int_a^b f^2(t)\, dt - \sum_{k=0}^{n} \lambda_k^2 \geq 0. \tag{1.107}$$

Because

$$\lim_{n \to \infty} E = 0, \tag{1.108}$$

then it follows from the expression (1.107) that

$$\int_a^b f^2(t)\, dt = \sum_{k=0}^{\infty} \lambda_k^2. \tag{1.109}$$

The relation (1.109) is called Parseval's equality. The square root of both its parts can be interpreted as the length of the vector $f(t)$ in the *H*-space $L_2[a,b]$, expressed through its projections on the orthogonal coordinate system $\{\varphi_k(t)\}$, which is a subset of the same H-space $L_2[a,b]$.

Concluding this subsection, we emphasize that in case of using the Legendre orthonormal polynomial function (1.91) for approximation, the Fourier coefficients must be calculated not by the expression (1.100), but by the expression

$$\mu_n = \frac{2n+1}{2}\int_{-1}^{1} f(t)P_n(t)dt, \quad n = 0,1,2,\dots,N, \tag{1.110}$$

and in the case of using Chebyshev 1 (1.93) to approximate the function $f(t)$ of orthonormal polynomials, the Fourier coefficients must be calculated not by the expression (1.100), but by the expressions:

$$\mu_n = \frac{1}{\pi}\int_{-1}^{1} f(t)T_n(t)(1-t^2)^{-\frac{1}{2}}dt, \quad n = 0, \tag{1.111}$$

$$\mu_n = \frac{2}{\pi}\int_{-1}^{1} f(t)T_n(t)(1-t^2)^{-\frac{1}{2}}dt, \quad n = 1,2,\dots,N \tag{1.112}$$

1.5 Programs for implementing operations in metric spaces in Python

A Python program for checking sets for equality, determining their power, and checking for equivalence

(Program 1):

```
In [1]: A={1,2,3,4,5}
In [2]: B={4,5,6,7,8}
In [3]: LA=list(A); LA
Out[3]: [1, 2, 3, 4, 5]
In [4]: LB=list(B); LB
Out[4]: [4, 5, 6, 7, 8]
In [5]: LA==LB
Out[5]: False
In [6]: len(LA)
Out[6]: 5
In [7]: len(LB)
Out[7]: 5
In [8]: len(LA)==len(LB)
Out[8]: True
```

End of program 1

A Python program for finding the sum of sets and their union excluding common elements, as well as for determining the difference and intersection of sets

(Program 2):

```
In [1]: dLA = {}
In [2]: dLA['a']=1
In [3]: dLA['b']=2
In [4]: dLA['c']=3
In [5]: dLA['e']=4
In [6]: dLA['h']=5
In [7]: dLA
Out[7]: {'a': 1, 'b': 2, 'c': 3, 'e': 4, 'h': 5}
In [8]: dLB={}
In [9]: dLB['e']=4;dLB['h']=5;dLB['p']=6;\
        dLB['q']=7;dLB['r']=8
In [10]: dLB
Out[10]: {'e': 4, 'h': 5, 'p': 6, 'q': 7, 'r': 8}
In [11]: dLA.keys() | dLB.keys()
Out[11]: {'a', 'b', 'c', 'e', 'h', 'p', 'q', 'r'}
In [12]: dLA.keys() - dLB.keys()
Out[12]: {'a', 'b', 'c'}
In [13]: dLB.keys() - dLA.keys()
Out[13]: {'p', 'q', 'r'}
In [14]: dLA.keys() & dLB.keys()
Out[14]: {'e', 'h'}
In [15]: dLA.keys() ^ dLB.keys()
Out[15]: {'a', 'b', 'c', 'p', 'q', 'r'}
```

End of program 2.

A Python program for determining the norm and metric of Banach spaces whose elements are numbers

(Program 3):

```
In [1]: import numpy as np
In [2]: a2=np.array([1,2])
In [3]: a3=np.array([1,2,3])
In [4]: a4=np.array([1,2,3,4])
In [5]: c2=np.array([2,1])
In [6]: c3=np.array([3,2,1])
In [7]: c4=np.array([4,3,2,1])
In [8]: e2=a2-c2;e2
Out[8]: array([-1,  1])
In [9]: e3=a3-c3;e3
Out[9]: array([-2,  0,  2])
In [10]: e4=a4-c4;e4
Out[10]: array([-3, -1,  1,  3])
In [11]: import scipy
In [12]: import scipy.linalg as la
In [13]: la.norm(a2)
Out[13]: 2.23606797749979
In [14]: la.norm(a3)
Out[14]: 3.7416573867739413
In [15]: la.norm(a4)
Out[15]: 5.477225575051661
In [16]: la.norm(c2)
Out[16]: 2.23606797749979
In [17]: la.norm(c3)
Out[17]: 3.7416573867739413
In [18]: la.norm(c4)
Out[18]: 5.477225575051661
In [19]: m2=la.norm(e2);m2
Out[19]: 1.4142135623730951
In [20]: m3=la.norm(e3);m3
Out[20]: 2.8284271247461903
In [21]: m4=la.norm(e4);m4
Out[21]: 4.47213595499958
```

End of program 3.

A Python program for determining the norms and metrics of Banach spaces C[0,1] whose elements are functions

(Program 4):

```
In [1]: import numpy as np
In [2]: x=np.linspace(0,1,11)
In [3]: g1=lambda x: -1+3*x-x**2
In [4]: g1vec=np.vectorize(g1)
In [5]: g11=g1vec(x)
In [6]: g11
Out[6]: array([-1.  , -0.71, -0.44, -0.19, 0.04,
                0.25, 0.44,  0.61,  0.76,  0.89,  1.])
In [7]: g111=np.piecewise(g11,[g11<0,g11>=0],\
        [lambda g11:-g11,lambda g11: g11])
In [8]: g111
Out[8]: array([1.  , 0.71, 0.44, 0.19, 0.04, 0.25,
                0.44, 0.61, 0.76, 0.89, 1.  ])
In [9]: ng1=g111.max( );ng1
Out[9]: 1.0
In [10]: ig1=g111.argmax( );ig1
Out[10]: 0
In [11]: g2=lambda x: 5*x-6*x**2
In [12]: g2vec=np.vectorize(g2)
In [13]: g22=g2vec(x);g22
Out[13]: array([ 0.   , 0.44, 0.76, 0.96,1.04,1.   ,
                 0.84,  0.56,  0.16,  -0.36, -1.   ])
In [14]: g222=np.piecewise(g22,[g22<0,\
         g22>=0], [lambda g22:-g22,\
         lambda g22: g22])
In [15]: g222
Out[15]: array([0.   , 0.44, 0.76, 0.96, 1.04,1.   ,
                0.84, 0.56, 0.16, 0.36, 1.   ])
In [16]: ng2=g222.max( ); ng2
Out[16]: 1.0399999999999998
In [17]: ig2=g222.argmax( ); ig2
Out[17]: 4
In [18]: g3=lambda x: -1-2*x+5*x**2
In [19]: g3vec=np.vectorize(g3)
In [20]: g33=g3vec(x);g33
Out[20]: array([-1.   , -1.15, -1.2  , -1.15, -1.   ,
                 -0.75, -0.4  , 0.05, 0.6 , 1.25, 2. ])
In [21]: g333=np.piecewise(g33,[g33<0,\
                g33>=0], [lambda g33:-g33,\
                lambda g33: g33])
In [22]: g333
Out[22]: array([1.  , 1.15, 1.2 , 1.15, 1.  , 0.75,
                 0.4 , 0.05, 0.6 , 1.25, 2.  ])
In [23]: mg3=g333.max( );mg3
Out[23]: 2.0
In [24]: ig3=g333.argmax( );ig3
Out[24]: 10
```

End of program 4 .

A Python program for determining the norms and metrics of Lebesgue spaces L(0,1), whose elements are functions

(Program 5):

```
In [1]: import numpy as np
In [2]: x=np.linspace(0,1,11)
In [3]: g1=lambda x: -1+3*x-x**2
In [4]: g1vec=np.vectorize(g1)
In [5]: g11=g1vec(x)
In [6]: g11
Out[6]: array([-1.  , -0.71, -0.44, -0.19, 0.04,
                0.25, 0.44, 0.61, 0.76, 0.89, 1. ])
In [7]: g111=np.piecewise(g11,[g11<0,g11>=0],\
        [lambda g11:-g11,lambda g11: g11])
In [8]: g111
Out[8]: array([1. , 0.71, 0.44, 0.19, 0.04, 0.25,\
                0.44, 0.61, 0.76, 0.89, 1.  ])
In [9]: c1=g111.sum( )
In [10]: nLg1=0.1*c1;nLg1
Out[10]: 0.633
In [11]: g2=lambda x: 5*x-6*x**2
In [12]: g2vec=np.vectorize(g2)
In [13]: g22=g2vec(x)
In [14]: g22
Out[14]: array([ 0.  ,  0.44,  0.76,  0.96, 1.04,
                 1. , 0.84,  0.56, 0.16, -0.36, -1.  ])
In [15]: g222=np.piecewise(g22,[g22<0,\
          g22>=0],[lambda g22:-g22,\
          lambda g22: g22])
In [16]: g222
Out[16]: array([0.  , 0.44, 0.76, 0.96, 1.04, 1.  ,
                0.84, 0.56, 0.16, 0.36, 1.  ])
In [17]: c2=g222.sum( )
In [18]: nLg2=0.1*c2;nLg2
Out[18]: 0.712
In [19]: g3=lambda x: -1-2*x+5*x**2
In [20]: g3vec=np.vectorize(g3)
In [21]: g33=g3vec(x)
In [22]: g33
Out[22]: array([-1.  , -1.15, -1.2 , -1.15,  -1. ,
                -0.75, -0.4 , 0.05, 0.6, 1.25, 2.  ])
In [23]: g333=np.piecewise(g33,[g33<0,\
          g33>=0], [lambda g33:-g33,\
          lambda g33: g33])
In [24]: g333
Out[24]: array([1.  , 1.15, 1.2 , 1.15, 1.  , 0.75,
                0.4 , 0.05, 0.6 , 1.25, 2.  ])
In [25]: c3=g333.sum( )
In [26]: mLg3=0.1*c3;mLg3
Out[26]: 1.0550000000000002
```

End of program 5.

A Python program for determining the norm, metric, and scalar product in Hilbert spaces whose elements are functions of a real variable

(Program 6):

```
In [1]: import sympy
In [2]: from sympy import *
In [3]: x,y,z=symbols('x y z')
In [4]: f1=5*x**3-3*x**2+2*x-4
In [5]: f2=1+2*x+3*x**3
In [6]: f3=f1*f1;f3
Out[6]: (5*x**3 - 3*x**2 + 2*x - 4)**2
In [7]: f4=expand(f3);f4
Out[7]: 25*x**6 - 30*x**5 + 29*x**4\
          - 52*x**3 + 28*x**2 - 16*x + 16
In [8]: a=symbols('a')
In [9]: a=integrate(f4,(x,0,1))
In [10]: a
Out[10]: 914/105
In [11]: nf1=a**0.5;nf1
Out[11]: 2.9503833487806
In [12]: f5=f2**2;f5
Out[12]: (3*x**3 + 2*x + 1)**2
In [13]: f6=expand(f5);f6
Out[13]: 9*x**6 + 12*x**4 + 6*x**3\
           + 4*x**2 + 4*x + 1
In [13]: b=symbols('b')
In [14]: b=integrate(f6,(x,0,1))
In [15]: b
Out[15]: 1999/210
In [16]: nf2=b**0.5;nf2
Out[16]: 3.08529538602832
In [17]: f7=f1*f2;f7
Out[17]: (3*x**3 + 2*x + 1)*(5*x**3 \
            - 3*x**2 + 2*x - 4)
In [18]: f8=expand(f7);f8
Out[18]: 15*x**6 - 9*x**5 + 16*x**4 \
           - 13*x**3 + x**2 - 6*x - 4
In [19]: sd=integrate(f8,(x,0,1))
```

```
In [20]: sd
Out[20]: -2551/420
In [21]: f9=f1-f2;f9
Out[21]:  2*x**3 - 3*x**2 – 5
In [22]: f10=f9**2;f10
Out[22]: (2*x**3 - 3*x**2 - 5)**2
In [23]: f11=expand(f10);f11
Out[23]: 4*x**6 - 12*x**5 + 9*x**4\
         - 20*x**3 + 30*x**2 + 25
In [24]: d=symbols('d')
In [25]: d=integrate(f11,(x,0,1))
In [26]: d
Out[26]: 1063/35
In [27]: mf1f2=d**0.5;mf1f2
Out[27]: 5.51102790515785
```

End of program 6.

A Python program for approximating a function of a real variable given in the Hilbert space by weighted sums of Legendre polynomials orthonormal to the line segment (Program 7):

```
In [1]: import sympy
In [2]: from sympy import *
In [3]: x,t,P=symbols('x t P')
In [4]: f1x=x**3-3*x**2+2*x-4
In [5]: f1t=27*t**3-27*t**2+6*t-4
In [6]: P0=1
In [7]: P1=t
In [8]: P2=(3*t**2-1)/2
In [9]: P3=(5*t**3-3*t)/2
In [10]: P4=(35*t**4-30*t**2+3)/8
In [11]: P5=(63*t**5-70*t**3+15*t)/8
In [12]: P6=(231*t**6-315*t**4\
                +105*t**2-5)/16
In [13]: P7=(429*t**7-693*t**5+315*t**3\
                -35*t)/16
In [14]: q0=f1t*P0*1/2
In [15]: q1= f1t*P1*3/2
In [16]: q2= f1t*P2*5/2
In [17]: q3= f1t*P3*7/2
In [18]: q4= f1t*P4*9/2
In [19]: q5= f1t*P5*11/2
In [20]: q6= f1t*P6*13/2
In [21]: q7= f1t*P7*15/2
In [22]: mju=symbols('mju')
In [23]: mju0=integrate(q0,(t,-1,1));mju0
Out[23]: -13
In [24]: mju1=integrate(q1,(t,-1,1));mju1
Out[24]: 111/5
In [25]: mju2=integrate(q2,(t,-1,1));mju2
Out[25]: -18
In [26]: mju3=integrate(q3,(t,-1,1));mju3
Out[26]: 54/5
In [27]: mju4=integrate(q4,(t,-1,1));mju4
Out[27]: 0
In [28]: mju5=integrate(q5,(t,-1,1));mju5
Out[28]: 0
In [29]: mju6=integrate(q6,(t,-1,1));mju6
Out[29]: 0
In [30]: mju7=integrate(q7,(t,-1,1));mju7
Out[30]: 0
In [31]: su=symbols('su')
In [32]: su01=mju0*P0+mju1*P1
In [33]: f1t1=expand(su01);f1t1
Out[33]: 111*t/5 – 13
In [34]: su02=su01+mju2*P2
In [35]: f1t2=expand(su02);f1t2
Out[35]: -27*t**2 + 111*t/5 – 4
In [36]: su03=su02+mju3*P3
In [37]: f1t3=expand(su03);f1t3
Out[37]: 27*t**3 - 27*t**2 + 6*t - 4
In [38]: f1t12=f1t-f1t1
In [39]: su11=integrate(f1t12*f1t12,(t,-1,1))
Out[39]: 28512/175
In [40]: nf1t21=su11**0.5;nf1t21
Out[40]: 12.7642357501620
In [41]: f1t22=f1t-f1t2
In [42]: su12=integrate(f1t22*f1t22,(t,-1,1))
In [43]: su12
Out[43]: 5832/175
In [44]: nf1t22=su12**0.5;nf1t22
Out[44]: 5.77284282530837
In [45]: f1t32=f1t-f1t3
In [46]: su13=integrate(f1t32*f1t32,(t,-1,1))
In [47]: su13
Out[47]: 0
In [48]: nf1t32=su13**0.5;nf1t32
Out[48]: 0
In [49]: import numpy as np
In [50]: import matplotlib as mpl
In [51]: import matplotlib.pyplot as plt
In [52]: mpl.rcParams['font.family']='fantasy'
In [53]: mpl.rcParams['font.fantasy'] \
             ='Arial','Times New Roman','Tahoma'
In [54]: t=np.linspace(-1,1,100)
```

```
In [55]: f1t=27*t**3-27*t**2+6*t-4
In [56]: f1t1=111*t/5 - 13
In [57]: f1t2=-27*t**2 + 111*t/5 - 4
In [58]: f1t3=27*t**3 - 27*t**2 + 6*t - 4
In [59]: fig=plt.figure(facecolor='white')
In [60]: plt.plot(t,f1t,'-k',t,f1t1,'-.g',t,f1t2,\
                  ':c',t,f1t3, '--r', linewidth=3)
In [61]: plt.legend(fontsize=18)
In [62]: ax=fig.gca()
In [63]: plt.title(r'Апроксимація  функції' \
                   ' поліномами  Лежандра')
In [64]: plt.text(0.45,.95,r'Графіки наближень'\
                  '  до функції',
                  horizontalalignment='center',\
                  verticalalignment='center',\
                  transform=ax.transAxes,fontsize=16)
In [65]: plt.xlabel(u't-вісь абсцис',{'fontname':\
                   'Times New Roman'})
In [66]: plt.ylabel(r'$f(t)$-ордината')
In [67]: plt.grid(True)
```

Note to program7: As shown in Figure 3, the graph of the function and its approximations can only be obtained if the part of this program, starting with command 49, is typed not using command lines, but in the form of a file like this shown in this figure. This is due to the fact that the matplotlib graphic editor is adapted to work with files, and not with command lines, when using which each subsequent command removes the result of the previous command from the screen, preventing the simultaneous display of the coordinate grid and graphs on the screen, and inscriptions

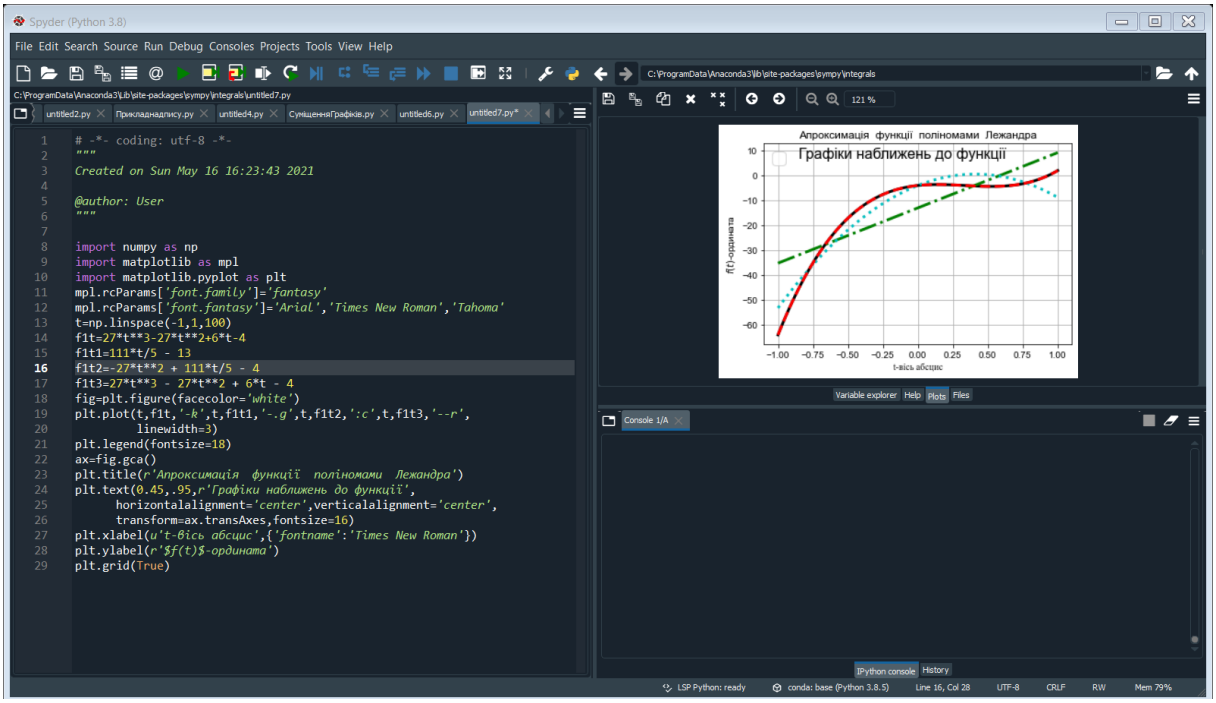


Figure 3. Graph of the function and three approximations to it by Legendre polynomials

End of program 7.

1.6 Tasks for self-testing

1. Define the concept of “set” and give examples of sets.
2. What are the sum, intersection and difference of sets?
3. What sets are equivalent?

4. Define the concept of “power of a set”
5. What is the power of a natural series of numbers?
6. What is the power of the set of rational numbers?
7. What is the power of the set of real numbers?
8. What is the power of the set of irrational numbers?
9. Give the definition of a counted set
10. How is the power of the counted set related to the power of the continuum?
11. What is a metric space?
12. What is a space metric and what conditions should it satisfy? .
13. Which sets are complete and closed?
14. Give the definition of a fundamental sequence in a metric space.
15. Which metric space is complete?
16. How is a linear metric space defined?
17. What is the norm of space? Give examples of norms.
18. Give the definition of Banach space.
19. Give the definition of Hilbert space. Give examples.
20. What conditions must be satisfied by a scalar product in Hilbert space?
21. How are the metric and the norm, the norm and the scalar product related to each other hilbert space?
22. What is an orthonormal sequence of functions in Hilbert spacious?
23. Which orthogonal sequence in Hilbert space is complete? Which is closed?
24. Define the function approximation process.
25. How to approximate a continuous function in space $L\,[a, b]$?
26. How to approximate a continuous function in space $L_2\,[a, b]$?
27. What is Parseval's equality. Give its geometric interpretation.
28. What is the Fourier spectrum of a continuous function in Hilbert space?
29. What orthogonal sequences based on power functions do you know?
30. How to turn a set into a list?
31. How to determine the number of elements in the list?
32. Which operation checks the equality of sets?
33. Which operation checks the equivalence of sets?
34. How to set an empty dictionary?
35. How to fill an empty dictionary with elements with keys?
36. Why do you need the keys() method?
37. How to combine sets?
38. How to find the intersection of sets?
39. How to find the difference of sets?
40. How to find the union of sets without common elements?
41. How to display the obtained result on the screen?
42. What do the functions len(LA), dLA{} and dLA['a']=1 define?
43. What function is used to calculate the norm in the Banach number space?
44. What symbols in the program denote the metrics of the Banach number space between given points?
45. Why do we need to use the lambda( ) function?
46. What role does the vectorize( ) function play?
47. What is the function piecewise( ) for?
48. Why do you need the expand( ) command?
49. How to bring the function defined on the segment [a,b] to the segment [-1,1], on which defined Legendre polynomials in the approximation problem?
50. Show the command in the program that introduces the fifth Legendre polynomial

Chapter 2. LEBEGUE'S MEASURE FOR SETS AND SPACES AND THEIR INTEGRALS

2.1 Lebesgue measure for sets and spaces

On the entire numerical axis or on its segment $[a, b]$, which is also the segment [0,1], on which a certain set is specified, each element of this set is displayed by a point, which, as is known, has zero length. In this connection, two interrelated questions arise: "What is the length of a segment of the number axis with its own length $b-a$ or a segment with unit length occupied by the set on this segment? And how to measure the length of that part of the segment occupied by the points of this set?".

Mathematics gave comprehensive answers to these questions by introducing the concept of Lebesgue's measure and extending it to functions defined on this segment, on this part of the plane or on this volume. Let's reveal these answers in more detail.

As we have already noted in the previous subsections, the power of the set of real numbers both on the entire numerical axis and on its segment $[a, b]$, which can be considered in the version of the segment [0, 1], is equal to the power of the continuum ***c***, and therefore for the convenience of expositions devoted to the theory of Lebesgue measure, we will consider exactly the set ***E*** defined on the segment [0,1].

Let us recall once again that the measure of a segment $[a, b]$, as well as of an interval (a,b), the measure of the sum of intervals α_i, that do not intersect, but each of which is a subset of the segment [0, 1], will be the sum $\sum_i \alpha_i$ of the lengths of these intervals.

Let the limited numerical set ***E*** be a subset of the unit segment [0, 1], of the numerical axis, i.e., $E \subset [0,1]$. We denote the subset that complements the set ***E*** to the unit segment of the number axis by the symbol CE.

We specify the set ***E*** by specifying its structure in the form of a limited one

$$E = \{x_1, x_2, ..., x_n\} \tag{2.1}$$

or counted

$$E = \{x_1, x_2, ..., x_n, ...\} \tag{2.2}$$

of numerical sequence.

Cover each point x_i of sequences (2.1) or (2.2) with an interval α_i, that does not intersect with other intervals that cover other points of these sequences, and find the sum $\sum_i \alpha_i$ of the lengths of these intervals for each of these sequences, which will be equal to, respectively,

$$\sum_{i=1}^{n} \alpha_i = \alpha_1 + \alpha_2 + ... + \alpha_n, \tag{2.3}$$

$$\sum_{i=1}^{\infty} \alpha_i = \alpha_1 + \alpha_2 + ... + \alpha_n + ... \tag{2.4}$$

Since the points on the number axis have no length, the intervals they cover within a unit segment of this axis can be very small, and therefore their sum will always have a lower limit

$$m^* E = \inf \sum_i \alpha_i . \tag{2.5}$$

This lower bound, given by the expression (2.5) and denoted by the symbol $m^* E$, is called the **external measure of the set *E***. It is obvious that the **external measure** $m^* CE$ **of**

the set CE, which complements the set ***E*** to the segment [0,1] with unit length on which they are both defined, can be determined in a similar way.

The difference between the length of a segment [0,1] and the external measure m^*CE of the set CE that complements the set ***E*** to this segment [0,1]

$$m_*E = 1 - m^*CE, \tag{2.6}$$

is called **the internal measure of the set *E*** and is denoted by the symbol m_*E.

Definition: **if the external measure m^*E of the set *E* and its internal measure m_*E coincide and are equal to the same number mE, that is, if**

$$m^*E = m_*E = mE, \tag{2.7}$$

then the set *E* is called Lebesgue measurable, and this number mE is called the Lebesgue measure of the set *E*.

It is obvious that from this statement, as a consequence, it follows that the set CE, which complements the set ***E*** to the unit segment on the number axis, is also measurable according to Lebesgue, and the expression for its measure mCE is also valid

$$m^*CE = m_*CE = mCE, \tag{2.8}$$

similar to (2.7).

For example, let's find Lebesgue measures for sets of real ***Z***, rational ***R*** and irrational $\overline{R}$ numbers given on the interval [0,1], which are related by the relation

$$Z = R \cup \overline{R} \tag{2.9}$$

The segment [0,1] of the numerical axis contains closely spaced points, which are the projections of real numbers onto it, so the Lebesgue measure of the set ***Z***, which has the power of the continuum ***c***, is its length on this segment, i.e.,

$$mZ = 1. \tag{2.10}$$

As we already know, the set ***R*** of rational numbers is a countable set, so it can be written as an infinite sequence (2.2). Let's choose a quantity $\varepsilon > 0$ and cover the numbers of the counted sequence (2.2) with the counted sequence of intervals (2.4), choosing for the first interval the quantity ε, for the second $\frac{\varepsilon}{2}$ – the quantity that can be presented as $\frac{\varepsilon}{2^1}$, for the third $\frac{\varepsilon}{4}$ – the quantity that can be presented as $\frac{\varepsilon}{2^2}$, for the nth – the quantity $\frac{\varepsilon}{2^{n-1}}$. In this case, the expression (2.4) can be rewritten as follows:

$$\sum_{i=1}^{\infty}\alpha_i = \varepsilon + \frac{\varepsilon}{2^1} + \frac{\varepsilon}{2^2} + \ldots + \frac{\varepsilon}{2^{n-1}} + \ldots = \varepsilon\left(1 + \frac{1}{2} + \frac{1}{2^2} + \ldots + \frac{1}{2^{n-1}} + \ldots\right) = \varepsilon\left(\frac{1}{1-\frac{1}{2}}\right) = 2\varepsilon. \tag{2.11}$$

We pay attention to the fact that in the brackets in the middle part of the expression (2.11) we have the sum of the terms of the descending geometric progression, which, as we know from the school mathematics course, is equal to its first term divided by one, minus the denominator, that is, minus the number by which you need to multiply the previous term of the geometric progression to get the next one.

Since, as we have already noted, the point on the numerical axis has no width, the value can also be set in the vicinity of zero, and therefore, taking into account expressions (2.5) and (2.11), we obtain

$$m^*R = \inf\sum_{i=1}^{\infty}\alpha_i = {}_{\varepsilon}\underline{\lim}_{0}(2\varepsilon) = 0 \tag{2.12}$$

The set $\overline{R}$ of irrational numbers given on the segment [0,1], which complements the set ***R*** of rational numbers to the set ***Z*** of real numbers on this unit segment of the number axis, as we already know from the previous subsections, like the set of real numbers, also has the power of the continuum ***c***, as a result of which is that the Lebesgue measure of the set $\overline{R}$ of irrational numbers is also the length of the segment, that is,

$$m\overline{R} = 1 \tag{2.13}$$

And on the basis of expressions (2.7) and (2.8) and expression (2.13), we have the right to write that

$$m\overline{R} = m^*\overline{R} = m_*\overline{R} = 1 = m^*CR \tag{2.14}$$

Taking into account expression (2.14), we obtain from expression (2.6)

$$m_*R = 1 - m^*CR = 1 - 1 = 0\,. \tag{2.15}$$

So it follows from expressions (2.7), (2.12) and (2.15) that the Lebesgue measure of the set *R* of rational numbers on the segment [0,1] is zero, and in connection with the enumerated set of such segments on the entire number axis, this set is zero-dimensional is also on the entire numerical axis.

And as a consequence of the considerations given above, the following definition follows: a set ***E*** is zero-dimensional according to Lebesgue, if for any $\eta > 0$ it can be covered by a finite or countable system of intervals with a total length less than η.

It follows from this definition that any set whose elements are individual points, regardless of whether their number is determined by a specific number or whether these points are the same as the numbers of the natural series, belongs to the zero-dimensional class, that is, to the dimensional class, but with a Lebesgue measure equal to zero. And this, in turn, means that if any function given on a segment $[a,b]$ has on this segment no more than a countable set of points in which the function tolerates discontinuities of the 1st kind, then the set of discontinuities of this function is zero-dimensional.

It follows from Lebesgue's definition of the concepts of set and measure that:

1) if the sets E_1, E_2 are Lebesgue measurable, then the sets $E_1 \cup E_2$, $E_1 \cap E_2$, $E_1 - E_2$ are also Lebesgue measurable;

2) if the sets E_1, E_2 are Lebesgue measurable and have no common interior points, then

$$m(E_1 \cup E_2) = m\,E_1 + m\,E_2; \tag{2.16}$$

3) if the sets E_1, E_2 are Lebesgue measurable and $E_1 \supset E_2$, then

$$m(E_1 - E_2) = m\,E_1 - m\,E_2; \tag{2.17}$$

4) if the set *E* is given in a limited region of the plane, which is a two-dimensional space, then by placing each of its points x_i in an open rectangle with the area α_i^2 and applying the procedure described above for the set whose points are located on the segment of the number axis, we will arrive at the numbers that will determine the external and the internal Lebesgue measure of the set *E* on the plane, which lead to the determination of the Lebesgue measure of this set *E* in two-dimensional space;

5) if the set *E* is given in a limited area of three-dimensional space, then by placing each of its points x_i in an open cube of volume α_i^3 and applying the procedure described above for the set whose points are located on the segment of the numerical axis, we will arrive at the numbers that will determine the external and the internal measure of the Lebesgue set *E* in three-dimensional space, which lead to the determination of the Lebesgue measure of this set *E* in this space.

An important note: after Lebesgue introduced his definition of the concept of measure on a set or in space, in almost all mathematical studies related to the application of the concept of measure, Lebesgue's measure is used, so mathematicians agreed, when using the term "measure", not to indicate every time that this is Lebesgue's measure, and understanding that this is so and without mentioning this surname next to this term.

Now let's extend the notion of measure to functions $f(x)$ of a real variable defined on the measurable sets of its argument ***x***.

Definition: A function $f(x)$ of a real variable *x* defined on the measurable set E of values *x* of its argument belongs to the class of measurable functions, if for any number *A* the subset $E_A \subset E$ for all elements x of which $f(x) > A$ is measurable.

We draw attention to the fact that the dimensionality of a function is determined by the dimensionality of the set of values of its argument. So, if even the set of all values of a function's argument has measure zero, that function will belong to the dimension class. And therefore, as a consequence of this definition of the dimensionality of the function, we have:

1) a function $f(x)$ that is continuous on a closed bounded set $E \subset (-\infty, \infty)$ of its argument x is measurable on this set;

2) the sum, difference, and product of two dimensional functions $f(x), g(x)$ given on the dimensional set E of its argument x is a measurable function on this set, and the quotient from the division of one dimensional function by another dimensional function is also a measurable function, except for the case when the function that is the divisor is zero at any of the points $x \in E$;

3) lattice function $f[k\Delta x] = f(x)\big|_{x=k\Delta x}$, generated by a continuous function $f(x)$ by taking into account its values only on a discrete set $E = \{k\Delta x\}$ of values of the argument x, where $\Delta x -$ the sampling interval a belongs $k = 0,1,2,...,n,$ to the class of dimensional functions, since it is defined on a zero-dimensional set E;

4) if two dimensional functions $f(x), g(x)$, are given on a measurable set E, then the subset $E_> \subset E$,with points $x \in E_>$ of $f(x) > g(x)$ is also measurable;

5) a continuous measurable function $f(x)$ defined on the measurable set E of its argument x has the C-property on this set if for any number $\varepsilon > 0$ there exists such a closed subset $F \subset E$, for which the inequality holds

$$m(E - F) < \varepsilon; \tag{2.18}$$

6) if a function $f(x)$ is defined on a measurable set E with a measure less than infinity, and at all points of the set E, with the possible exception of some zero-dimensional subset, has values less than infinity, then for this function to be measurable, it is sufficient that it has the C-property on this set.

2.2 Riemann and Stieltjes integrals

First, consider the Riemann integral. This integral is one of the main concepts of mathematical analysis, which is studied in the course of higher mathematics according to the curriculum of any specialty of non-mathematical specialization, therefore, in the textbook on functional analysis, which is a superstructure on mathematical analysis, we will only recall it, and then compare it with he further introduced the Stiltjes and Lebesgue integrals according to the program of functional analysis.

So, let a continuous and bounded function $f(x)$ be given on the numerical axis on the segment $[a,b]$ of the argument $\boldsymbol{x}$ values. Let us divide this segment by points $x_0, x_1, x_2, ..., x_n$, and so that $x_0 = a$ and $x_n = b$, on $\boldsymbol{n}$ segments $\Delta_i x$, $i = 1,2,...,n$, such that

$$\Delta_i x = x_i - x_{i-1}, \qquad i = 1,2,...,n \tag{2.19}$$

Let us denote the internal point ξ_i of the segment by $\Delta_i x$, that is, let $\xi_i \in \Delta_i x$. Since the function $f(x)$ is continuous and bounded, it will acquire a maximum M_i and a minimum m_i numerical value on each segment $\Delta_i x$ at some of its internal points or on its boundaries, that is, we will have

$$m_i \le f(\xi_i) \le M_i, \qquad i = 1,2,...,n \tag{2.20}$$

Multiply all terms of inequality (2.20) by $\Delta_i x$ and sum these products, resulting in

$$\sum_{i=1}^{n} m_i \Delta_i x \le \sum_{i=1}^{n} f(\xi_i) \Delta_i x \le \sum_{i=1}^{n} M_i \Delta_i x \tag{2.21}$$

Sums

$$\Sigma_{\max} = \sum_{i=1}^{n} M_i \Delta_i x, \tag{2.22}$$

$$\Sigma_{\min} = \sum_{i=1}^{n} m_i \Delta_i x \tag{2.23}$$

are called, respectively, the upper and lower sums of Darba. It is quite obvious that when reduced $\Delta_i x$ to zero, the superscript $\boldsymbol{n}$ in the Darboux sums will approach infinity, and these sums themselves will approach a common limit in the form of a number from above and below

$$J = \lim_{\Delta_i x \to 0} \sum_{i=1}^{n \to \infty} f(\xi_i) \Delta_i x, \tag{2.24}$$

which is called the Riemann integral and denoted

$$J = \int_a^b f(x) dx. \tag{2.25}$$

Since the operation of finding the sum is linear, for which it is true that

$$\sum_{i=1}^{\infty} f(\xi_i) \Delta_i x = \sum_{i=1}^{n} f(\xi_i) \Delta_i x + \sum_{i=n+1}^{\infty} f(\xi_i) \Delta_i x, \tag{2.26}$$

then the integration operation is also linear, for which it is true that

$$\int_a^b f(x) dx = \int_a^{a_*} f(x) dx + \int_{a_*}^b f(x) dx, \tag{2.27}$$

if $a_* \in [a,b]$.

Proceeding from the fact that each component $M_i \Delta_i x$, $m_i \Delta_i x$ in the Darboux sums (2.22), (2.23) determines the area of a rectangle with base $\Delta_i x$ and height M_i and m_i, respectively, the Riemann integral, which is a definite integral belonging to the class of functionals, can be geometrically interpreted as an expression which is used to determine the

area of a flat figure bounded from the bottom by the segment $[a,b]$ of the number axis, and from the top by the graph of the function $f(x)$ in the range from $f(a)$ to $f(b)$.

And now let's move on to the consideration of the Stiltjes integral.

From the above, it is easy to see that we constructed the Riemann integral (2.25), which is one of the basic concepts of mathematical analysis, not using the measure theory in an explicit form, but using the upper (2.22) and lower (2.23) Darboux sums, to which we applied boundary transition.

It turns out that in the same way it is possible to construct another class of integrals that entered mathematics under the name of the Dutch mathematician Stiltjes, who created their theory in response to a request from the theory of probabilities.

The main difference between the Stiltjes integral and the Riemann integral is that in the Riemann integral, the integration over a segment $[a,b]$ of a function $f(x)$ is carried out using the increments of its argument dx on the same segment of the numerical axis, and in the Stiltjes integral, the integration over a segment $[a,b]$ of the function $f(x)$ is carried out using the increments $dg(x)$ of another function $g(x)$ specified on the same segment of the number axis, and the integrated function $f(x)$ itself is called integrated over the function $g(x)$ on the segment $[a,b]$ of the number axis.

Symbolically, Stiltjes' integral is written as follows:

$$S = \int_a^b f(x)dg(x). \tag{2.28}$$

The algorithm for constructing the integral (2.28), proposed by Stiltjes, differs from the algorithm for constructing the Riemann integral only in details, because first, as in the Riemann algorithm, it is proposed to set a continuous and bounded function $f(x)$ on the numerical axis on the segment $[a,b]$ of the argument x values and to divide this segment into points $x_0, x_1, x_2, ..., x_n$ so that $x_0 = a$ and $x_n = b$ on n segments $\Delta_i x$, $i = 1,2,...,n$, each of which is determined by expression (2.19).

If we mark the internal point of the segment $\Delta_i x$ with the symbol ξ_i since function *f*(*x*) is continuous and limited, on each segment $x\ldots$ in some of its internal points or on its bounds it equired maximum M and minimal m of the digital values, and the expression (2.20) will be valid for it.

Then we consider the bounded function $g(x)$ given on the same $[a,b]$ segment, on which we do not impose the continuity condition and for which on each segment given by the expression (2.19), we find the increases

$$\Delta_i g = g(x_i) - g(x_{i-1}), \quad i = 1,2,...,n. \tag{2.29}$$

Multiplying all terms of inequality (2.20) by $\Delta_i g$ and summing up these products, we receive

$$\sum_{i=1}^{n} m_i \Delta_i g \le \sum_{i=1}^{n} f(\xi_i)\Delta_i g \le \sum_{i=1}^{n} M_i \Delta_i g. \tag{2.30}$$

Sums

$$\Sigma^s_{\max} = \sum_{i=1}^{n} M_i \Delta_i g = \sum_{i=1}^{n} M_i (g(x_i) - g(x_{i-1})), \tag{2.31}$$

$$\Sigma^s_{msn} = \sum_{i=1}^{n} m_i \Delta_i g = \sum_{i=1}^{n} m_i (g(x_i) - g(x_{i-1})) \tag{2.32}$$

are called, respectively, the upper and lower Darboux-Stiltjes sums. It is quite obvious that when decreasing $\Delta_i x$ to zero, the superscript $\boldsymbol{n}$ in the Darboux-Stiltjes sums will approach infinity, and these sums themselves, due to the limitation of increments (2.29), will approach a common limit from above and below in the form of a number

$$S = \underset{\Delta_i x \to 0}{\lim} \sum_{i=1}^{n\to\infty} f(\xi_i)(g(x_i) - g(x_{i-1})), \tag{2.33}$$

which is called the Stiltjes integral and is written in the form (2.28).

It follows from the very definition of the Stiltjes integral that:

1) $$\int_a^b dg(x) = g(b) - g(a), \tag{2.34}$$

2) $$\int_a^b kf(x)dg(x) = k\int_a^b f(x)dg(x), \qquad \forall k = const, \tag{2.35}$$

3) $$\int_a^b f(x)dg(x) = \int_a^{a_*} f(x)dg(x) + \int_{a_*}^b f(x)dg(x), \qquad \forall a_* \in [a,b], \tag{2.36}$$

4) $$\int_a^b (f_1(x) \pm f_2(x))dg(x) = \int_a^b f_1(x)dg(x) \pm \int_a^b f_2(x)dg(x), \tag{2.37}$$

5) $$\int_a^b f(x)d(g_1(x) \pm g_2(x)) = \int_a^b f(x)dg_1(x) \pm \int_a^b f(x)dg_2(x). \tag{2.38}$$

2.3 The Lebesgue integral

When mathematicians saw that there are functions that do not integrate after Riemann, they began to search for such a generalization of the concept of the definite integral, with the help of which these functions could also be integrated. And such a generalization was achieved by Lebesgue, who proposed that the increment of the coordinate, by which the integration of the function $y = f(x)$ given on the segment $[a,b]$ is carried out, be determined not along the axis of the argument ***x***, but along the functional axis *y*, because in this case, even when the coordinate ***x*** is set on a zero-dimensional set E of a finite or countable quantity points on the x axis, the coordinate ***y*** will be an element of the set of real numbers on the segment [m,M] of the y axis, the measure of which is its length, and the left border of which is the real number ***m*** , which is the minimum value of this function on the segment $[a,b]$, and the right border is the real number ***M***, which is the maximum value of this function on the same segment $[a,b]$.

Lebesgue constructed his integral by putting forward the condition that the measurable and limited by the lower ***m*** and upper ***M*** values function $y = f(x)$, given on the segment $[a,b]$ of the axis ***x***, was defined on the set ***E*** with measure

$$mE(m \le y < M). \tag{2.39}$$

He suggested to devide the segment [m, M] of the axis ***y*** into points

$$m = y_0, y_1, y_2, ..., y_{i-1}, y_i, ..., y_n = M \tag{2.40}$$

into *n* segments $[y_{i-1}, y_i]$, $i = 1,2,...,n$, marking the maximal of them

$$\alpha = \max\left[y_{i-1}, y_i\right], \tag{2.41}$$

and making up the sum

$$\Sigma = \sum_{i=1}^{n} y_{i-1} mE_i\left(y_{i-1} \le y < y_i\right), \tag{2.42}$$

using the fact that, according to property (2.16), measures

$$mE\left(m \le y < M\right) = \sum_{i=1}^{n} mE_i\left(y_{i-1} \le y < y_i\right). \tag{2.43}$$

The limit of the sum (2.42) when even the maximum of the segments defined by the expression (2.41) approaches zero

$$L =_{\alpha} \underrightarrow{\lim}_0 \Sigma =_{\alpha} \underrightarrow{\lim}_0 \sum_{i=1}^{n} y_{i-1} mE_i\left(y_{i-1} \le y < y_i\right), \tag{2.44}$$

which has all the properties of an integral, Lebesgue introduced into mathematics as a new interpretation of the definite integral, which other mathematicians named after him the Lebesgue integral, and the sum (2.42) was called the Lebesgue integral sum on the set *E* with the measure determined by the expression (2.39).

If, using the property of monotonicity of the measure, write $mE_i\left(y_{i-1} \le y < y_i\right)$ for the *i*-th segment in the form

$$mE_i\left(y_{i-1} \le y < y_i\right) = mE_i\left(y_i > y\right) + mE_i\left(y \ge y_{i-1}\right) = mE_i\left(y_i > y\right) - mE_i\left(y_{i-1} \ge y\right), \tag{2.45}$$

and substituting expression (2.45) into expression (2.44), we obtain the Lebesgue integral in the form

$$L =_{\alpha} \underrightarrow{\lim}_0 \sum_{i=1}^{n} y_{i-1} mE_i\left(y_{i-1} \le y < y_i\right) =_{\alpha} \underrightarrow{\lim}_0 \sum_{i=1}^{n} y_{i-1} \left\{mE_i\left(y_i > y\right) - mE_i\left(y_{i-1} > y\right)\right\}. \tag{2.46}$$

Since the measure mE_i is a monotone function, we denote it by the symbol *g*,-coordinate *y*, then we can write that

$$\begin{cases} mE_i\left(y_i > y\right) = g\left(y_i\right), \\ mE_i\left(y_{i-1} \ge y\right) = g\left(y_{i-1}\right), \\ \Delta_i g = g\left(y_i\right) - g\left(y_{i-1}\right) \end{cases} \tag{2.47}$$

Substituting expressions (2.47) into (2.46), we get

$$L =_{\alpha} \underrightarrow{\lim}_0 \sum_{i=1}^{n} y_{i-1} \left\{g\left(y_i\right) - g\left(y_{i-1}\right)\right\} =_{\alpha} \underrightarrow{\lim}_0 \sum_{i=1}^{n} y_{i-1} \Delta_i g\,. \tag{2.48}$$

Analyzing the right-hand side of the expression (2.48), we see that it is the Stieltjes integral of the function y by function $g\left(y\right)$ on the segment [m,M], which is the measure of the function y, and therefore the expression (2.48) can be written as follows:

$$L = \int_m^M y dg\left(y\right). \tag{2.49}$$

By reducing the Lebesgue integral (2.44) to the Stieltjes integral (2.49), we actually proved that the limit of the Lebesgue integral sum (2.42) exists, so the expression (2.44) really satisfies the requirements that the definite integral must satisfy.

We draw attention to the fact that if the function of the measure by the coordinate y on the segment [m, M] in the projection onto the segment $[a,b]$ of the axis x is not only monotonic, but also linear, that is, if

$$g(y)=x, \tag{2.50}$$

then the expression (2.49) turns into the expression

$$L=\int_m^M y dg(y)=\int_a^b f(x)dx. \tag{2.51}$$

That is, in this case, the Lebesgue integration on the axis y segment [m, M] and the Riemann integration on the axis x segment $[a,b]$ give the same result, which also connects the Riemann integral with the measure theory, since the Riemann integration over x is an integration over the measure of the segment [a,b].

2.4 Programs for implementing integrals in the Python language

The program for calculating the Riemann integral in case when the limits of integration are real numbers, for example, the limits of the range [0,1], when the body of the function is specified directly, when it is specified parametrically, and also in the case of using a lambda function with specified numerical values of parameters.

(Program 8):

```
In [1]: import scipy
In [2]: from scipy.integrate import quad
In [3]: import numpy as np
In [4]: def f(x):
            return np.exp(-x)**2*np.cos(x)**3
In [5]: q1=quad(f,0,1);q1
Out[5]: (0.3398620054810545, 3.773226e-15)
In [6]: def f(x,a3,a2,a1,a0):
            return a3*x**3+a2*x**2+a1*x+a0
In [7]: q2=quad(f,0,1,args=(4,3,2,1));q2
Out[7]: (4.0, 4.440892098500626e-14)
In [8]: q3=quad(lambda x: f(x,4,3,2,1),0,1);q3
Out[8]: (4.0, 4.440892098500626e-14)
```

End of program 8.

Python program for calculating the Steeltjes integral from the function $f(x)=x^3+3x^2-2x-4$ by the function $g(x)$=2exp(x)-3x of the real variable x specified on the interval [0,2] discretely through the interval $\Delta_i x=x_i-x_{i-1}$= 0.1; i = 1,2,...,20

(Program 9):

```
In [1]: import numpy as np
In [2]: x=np.linspace(0,2,21)
In [3]: def f(x):
            return x**3+3*x**2-2*x-4
In [4]: fvec=np.vectorize(f)
In [5]: f1=fvec(x);f1
Out[5]:
array([-4.   , -4.169, -4.272, -4.303, -4.256,
       -4.125, -3.904, -3.587, -3.168, -2.641,
       -2. , -1.239, -0.352, 0.667, 1.824, 3.125,
        4.576, 6.183, 7.952, 9.889, 12. ])
In [6]: g=lambda x: 2*np.exp(x)-3*x
In [7]: gvec=np.vectorize(g)
In [8]: g1=gvec(x);g1
Out[8]:
array([2.        , 1.91034184, 1.84280552,
       1.79971762, 1.7836494 , 1.79744254,
       1.8442376 , 1.92750541, 2.05108186,
       2.21920622, 2.43656366, 2.70833205,
       3.04023385, 3.43859334, 3.91039993,
       4.46337814, 5.10606485, 5.84789478,
       6.69929493, 7.67178888, 8.7781122])
In [9]: g11=np.diff(g1);g11
Out[9]
array([-0.08965816, -0.06753632,
       -0.0430879, -0.01606822,  0.01379315,
```

```
        0.04679506, 0.08326781, 0.12357644,
        0.16812437, 0.21735743, 0.27176839,
        0.3319018 ,  0.39835949, 0.4718066 ,
        0.55297821, 0.64268671, 0.74182993,
        0.85140015,0.97249396,1.10632331])
In [10]: f11=f1[:-1];f11
Out[10]:
array([-4. ,-4.169,-4.272, -4.303, -4.256, -4.125,
        -3.904,-3.587,-3.168,-2.641,-2. , -1.239,
        -0.352, 0.667,1.824,3.125, 4.576, 6.183,
         7.952,  9.889])
```

```
In [11]: s1=f11*g11;s1
Out[11]:
array([ 0.35863266,  0.28155892,  0.18407151,
         0.06914155,-0.05870363,-0.19302962,
        -0.32507755, -0.4432687, -0.53261799,
        -0.57404098,-0.54353678, -0.41122633,
        -0.14022254, 0.314695, 1.00863225,
         2.00839596, 3.39461378 5.2642071,
         7.73327194, 10.94043125])
In [12]: S=s1.sum( );S
Out[12]: 28.33592779370339
```

End of program 9

A Python program for calculating the Lebesgue integral from the function $f(x) = e^{-x}sin2x$ of the real variable x given on the segment [0,3] discretely at points through the interval $\Delta_i x = x_i - x_{i-1} = 0.15; \quad i = 1,2,...,20$

(Program 10):

```
In [1]: import numpy as np
In [2]: x=np.linspace(0,3,21)
In [3]: def f(x):
            return np.exp(-x)*np.sin(3*x)
In [4]: fvec=np.vectorize(f)
In [5]: f1=fvec(x);f1
Out[5]:
array([ 0.        ,  0.3743783 ,  0.58030285,
        0.62214868,0.53445891,0.36753575,
        0.17375969,-0.00294201,-0.1332846,
       -0.20441749,-0.21811645,-0.1866539,
       -0.12773711,-0.05972154,0.00205897,
        0.0474343,0.07199992,0.07646286,
        0.06518194,0.0443898,0.02051817])
In [6]: M=max(f1);M
Out[6]: 0.6221486811452028
In [7]: m=min(f1);m
Out[7]: -0.21811645170452654
In [8]: mEf=M-m;mEf
Out[8]: 0.8402651328497294
In [9]: f11=np.sort(f1);f11
Out[9]:
array([-0.21811645, -0.20441749, -0.1866539 ,
        -0.1332846,-0.12773711,-0.05972154,
        -0.00294201,  0.        ,  0.00205897,
         0.02051817,0.0443898,0.0474343,
         0.06518194,0.07199992,0.07646286,
         0.17375969,0.36753575,0.3743783 ,
         0.53445891,0.58030285,0.62214868])
In [10]: g=np.diff(f11);g
```

```
Out[10]:
array([0.01369896, 0.0177636 , 0.0533693 ,
       0.00554749, 0.06801557, 0.05677952,
       0.00294201, 0.00205897, 0.0184592 ,
       0.02387163, 0.0030445 , 0.01774765,
       0.00681798, 0.00446294, 0.09729683,
       0.19377606, 0.00684255, 0.16008061,
       0.04584394, 0.04184583])
In [11]: f111=f11[:-1];f111
Out[11]:
array([-0.21811645, -0.20441749, -0.1866539 ,
        -0.1332846 ,-0.12773711,-0.05972154,
        -0.00294201,  0.        ,  0.00205897,
         0.02051817, 0.0443898 ,  0.0474343,
         0.06518194,0.07199992,0.07646286,
         0.17375969, 0.36753575, 0.3743783,
         0.53445891, 0.58030285])
In [12]: I1=f111*g;I1
Out[12]:
array([-2.9879683e-03,-3.6311899e-03,
       -9.96158766e-03, -7.39394368e-04,
       -8.68811299e-03, -3.39096030e-03,
       -8.65544179e-06,  0.00000000e+00,
        3.80069447e-05,  4.89802141e-04,
        1.35144637e-04,  8.41847084e-04,
        4.44409318e-04,  3.21331035e-04,
        7.43959408e-03,  3.36704688e-02,
        2.51488210e-03,  5.99307062e-02,
        2.45017000e-02, 2.42832565e-02])
In [13]: L=np.sum(I1);L
Out[13]: 0.1252032798312037
```

End of program 10

2.5 Self -Testing Task

1. Give a definition of the concept of “measure”.
2. How to determine the measure of Lebesgue and what are its properties?
3. What is a zero-dimensional set?
4. Define a measuring function.
5. Write down the expressions for the lower and upper Sumy Darb.
6. Give a definition of Riman's integral.
7. Write down the expressions for the lower and upper Sumy Darb-Siltyes.
8. Give a definition of the Stiltjes ' integral.
9. Under what condition can Riman integral be obtained from the integral integral?
10. Give a definition of the integral.
11. What is the connection between the integals of Lebeg and the Stiltjes ?
12. Under what conditions are Riman and Lebeg integrated to give the same result?
13. What feature in the program is performed by the command fvec=np.vectorize(f)
14. What is the program g=lambda x: 2*np.exp(x)-3*x?
15. What does the program g11=np.diff(g1)?
16. What we reach the command f11=f1[:-1]?
17. What is the implementation of the team f11=f1[:-1]?
18. Toe command in the program that calculates the integral of the Stiltjes
19. The in the results of the program both vectorized functions
20. Say in the results of the program the numerical value
21. What does the program require f11=np.sort(f1)?
22. What we reach the command f111=f11[:-1]?
23. What gives us the implementation of the l1=f111*g?
24. Show in the results of the program the numerical value of the Lebesgue integral.

Chapter 3. FUNCTIONALS AND METHODS OF SEARCHING FOR THEIR UNCONDITIONAL EXTREMUMS

3.1 Functionalities used in applied IT tasks

As we have already defined in the introduction, a *functional* is a law according to which a set of functions is matched to a set of numbers.

We will also recall that the final stage of the system analysis of a complex object is its optimization, which consists in finding such parameters of this object, according to which its initial coordinate or its spatial structure are characterized in the best way according to the indicators chosen as a measure of this "bestness". It is these indicators, the internal components of which are functions that characterize the spatial structures of complex objects or processes in them, and the external components are numbers that characterize the "best" of these spatial structures or processes, as a rule, are functionals, among which in general of functional analysis methodologies in its application in the form of calculus of variations, the most common are:

1) functional

$$J_y^F = \int_a^b F(x, y, y')\ dx, \tag{3.1}$$

where $F(x, y, y')$ is a mathematical expression, which is a construction of an independent variable ***x***, its function $y(x)$ and the first derivative $y'(x)$ of this function; at the same time, the segment $[a, b]$ is the domain of setting the function $y(x)$, i.e., $x \in [a, b]$;

2) functional

$$J = \int_a^b F\left(x, y, y', y'', \ldots, y^{(n)}\right) dx, \tag{3.2}$$

which connects not only the function $y(x)$ and its derivative $y'(x)$, as in the case of (3.1), but also older derivatives $y''(x), \ldots, y^{(n)}(x)$;

3) functional

$$J = \int_a^b F\left(x, y_1, y_2, \ldots, y_n; y_1', y_2', \ldots, y_n'\right) dx, \tag{3.3}$$

which connects the set of functions $\{y_1(x), y_2(x), \ldots, y_n(x)\}$, defining a surface in ***n***-dimensional space and the set of first derivatives $\{y_1'(x), y_2'(x), \ldots, y_n'(x)\}$ of these functions in the same space.

The study of the conditions under which these functionals acquire extreme values is carried out within the framework of functional analysis, which is called calculus of variations, and the functions on which these functionals acquire extreme values are called extremals.

3.2 Classical problem of calculus of variations, necessary and sufficient conditions for the existence of an unconditional extremum of the functional

The set of methods for finding extrema of functionals of various types, from the number we have given in the first subsection of this section, constitutes the essence of calculus of variations, to understand the basics of which we will conduct research on the extremum of the functional (3.1).

But, before determining the conditions under which the functional (3.1) will acquire an extreme value, let us clarify what we mean by the concepts of absolute and relative extrema of the functional, and what we mean by the concept of a weak relative extremum of the functional.

By analogy with how global and local extrema are determined for a function, absolute and relative extrema are determined for a functional, the first of which specifies the largest (or smallest) value of the functional on the entire set of functions on which this functional is specified, and the second specifies the largest (or the smallest) value of a functional on a subset of close functions that are only a part of the entire set of functions on which this functional is defined.

In turn, strong and weak are distinguished among relative extremes.

To unify the approaches, it was agreed to consider that a strong relative minimum of the functional is reached at the extremal $f_e(x)$, if its value on this curve in the given range $[a, b]$ of values of the argument x is smaller than on all other curves $f_i(x),\ i = 1,\ldots,n$ of the given class of functions, the zero-order distance

$$\Delta_0 = \max\left|f_e(x) - f_i(x)\right|,\quad i = 1,\ldots,n \tag{3.4}$$

to which is small. It is obvious that a strong relative maximum of the functional will be reached at the extremal of the same class of curves, if its value is the largest in the given range of values of the argument.

If the relative minimum (or maximum) of the functional is reached at the extremal, the distance is of the first order

$$\Delta_1 = \max\left|f_e'(x) - f_i'(x)\right|,\quad i = 1,\ldots,n, \tag{3.5}$$

from which to all other curves of this class of functions is small, then a weak relative minimum (or maximum) occurs at this extreme.

Note that in expression (3.5), the symbols denote the first derivatives $f_e'(x)$, $f_i'(x)$ of the functions $f_e(x)$, $f_i(x)$.

It is clear that the absolute extremum is at the same time relative, and the strong relative extremum is at the same time weak.

And therefore, if some condition must be fulfilled with respect to a weak relative extremum, then it must be true for both a strong relative extremum and an absolute one.

Having dealt with the above concepts, we will determine what conditions the function $y = y(x)$, must satisfy in order for it to have a weak relative minimum of the functional (3.1).

To find these conditions, we assume that this weak relative minimum of the functional (3.1) on the function $y = y(x)$ is reached.

This assumption gives us the right to assume that the value of the functional (3.1) on any other function, for example, $y_\eta(x) = y(x) + \alpha \cdot \eta(x)$, where α is a number and $\eta(x)$ is an arbitrary smooth function for which

$$\eta(a) = \eta(b) = 0, \tag{3.6}$$

will not be less than $y = y(x)$, i.e.,

$$\Delta J_y^F = \int_a^b \left(x, y + \alpha\eta, y' + \alpha\eta'\right) dx - \int_a^b F\left(x, y, y'\right) dx = J_y^F\left(\alpha\right) - J_y^F \geq 0. \tag{3.7}$$

Since the definite integral after integration and substitution of boundaries is converted into a number, the value of the increment of the functional, which is given by the difference (3.7) of the definite integrals, will depend only on the value of the parameter α, that is, this increment becomes a continuous function, in which the independent variable is α

$$\Delta J_y^F = \Delta J_y^F(\alpha) = J_y^F(\alpha) - J_y^F. \tag{3.8}$$

As you know from the course of mathematical analysis, a continuous function around any value of its argument, in particular zero, can be expanded into a Taylor series. For the function (3.8) around the point $\alpha = 0$, this series will have the form:

$$\Delta J_y^F(\alpha) = \alpha \cdot \frac{dJ_y^F(\alpha)}{d\alpha} + \frac{\alpha^2}{2!} \cdot \frac{d^2 J_y^F(\alpha)}{d\alpha^2} + \frac{\alpha^3}{3!} \cdot \frac{d^3 J_y^F(\alpha)}{d\alpha^3} + \cdots. \tag{3.9}$$

The first and second members of the series (3.9), i.e. $\alpha \cdot \frac{dJ_y^F(\alpha)}{d\alpha}$ and $\frac{\alpha^2}{2} \cdot \frac{d^2 J_y^F(\alpha)}{d\alpha^2}$, are called, respectively, the first and second variations of the functional J_y^F and are denoted by δJ_y^F and $\delta^2 J_y^F$.

Since the independent variable α enters the second variation $\delta^2 J_y^F$ in the square, then at values close to zero, the second variation becomes much smaller than the first variation δJ_y^F, which is included α in the first degree. And this, in turn, gives us the right to assume that the point is around $\alpha = 0$

$$\Delta J_y^F(\alpha) \approx \delta J_y^F. \tag{3.10}$$

It follows from the expressions (3.7)-(3.10) that

$$\alpha \cdot \frac{dJ_y^F(\alpha)}{d\alpha} \geq 0. \tag{3.11}$$

It is clear that for arbitrary values α (both bigger and less than zero), the expression (3.11) in the vicinity of the point $\alpha = 0$ is fulfilled only in one case, when

$$\frac{dJ_y^F(\alpha)}{d\alpha} = 0, \tag{3.12}$$

or

$$\frac{d}{d\alpha}\left(\int_a^b F(x, y + \alpha\eta, y' + \alpha\eta')\, dx \right) = 0. \tag{3.13}$$

Since the operations of differentiation and integration are linear, they can be interchanged, i.e. (3.13) can be rewritten as follows:

$$\int_a^b \frac{d}{d\alpha} F(x, y + \alpha\eta, y' + \alpha\eta')\, dx = 0. \tag{3.14}$$

Taking the derivative of in α the integrand expression (3.14), by the rule of differentiation of a complex function we will have

$$\int_a^b \left(\frac{\partial F}{\partial y} \cdot \frac{d(y + \alpha\eta)}{d\alpha} + \frac{\partial F}{\partial y'} \cdot \frac{d(y' + \alpha\eta')}{d\alpha} \right) dx = 0, \tag{3.15}$$

or

$$\int_a^b \left(\frac{\partial F}{\partial y} \cdot \eta + \frac{\partial F}{\partial y'} \cdot \eta' \right) dx = 0. \tag{3.16}$$

The expression (4.31) is inconvenient for analysis, since its first component includes the function $\eta(x)$ itself, and its derivative $\eta'(x)$ is included in the second. You can get rid of the derivative $\eta'(x)$ by taking the integral of the second component of the expression by parts. To simplify notation, we will denote partial derivatives as follows:

$$\begin{cases} \dfrac{\partial F}{\partial x} = F_x, \dfrac{\partial F}{\partial y} = F_y, \dfrac{\partial F}{\partial y'} = F_{y'}, \\ \dfrac{\partial^2 F}{\partial y' \partial \mathrm{x}} = F_{y'x}, \dfrac{\partial^2 F}{\partial y' \partial y} = F_{y'y}, \dfrac{\partial^2 F}{\partial y' \partial y'} = F_{y'y'} \end{cases} \tag{3.17}$$

Taking the integral in the second component of equation (3.16), that is, the integral

$$\int_a^b \frac{\partial F}{\partial y'} \cdot \eta' \, dx = \int_a^b F_{y'} \cdot \eta' \, dx, \tag{3.18}$$

using the method of integration by parts, we will have

$$\int_a^b F_{y'} \cdot \eta' \, dx = -\int_a^b \frac{d}{dx} F_{y'} \cdot \eta \, dx. \tag{3.19}$$

Considering (3.19), equation (3.16) can be rewritten as follows:

$$\int_a^b \left(F_y \cdot \eta - \frac{d}{dx} F_{y'} \cdot \eta \right) dx = 0, \tag{3.20}$$

Or

$$\int_a^b \left(F_y - \frac{d}{dx} F_{y'} \right) \cdot \eta \, dx = 0. \tag{3.21}$$

Analyzing the obtained integral equation (3.21), we see that for an arbitrary smooth function $\eta(x)$ it is necessary that it be performed

$$F_y - \frac{d}{dx} F_{y'} = 0. \tag{3.22}$$

This is Euler's well-known equation, which he derived in 1744 by transforming the functional into a function of many variables followed by its minimization.

Summarizing all of the above, it can be asserted that in order for the function $y(x)$ to deliver a weak relative minimum of the functional (3.1), it must be a solution of the Euler equation (3.22). **In this case, the function $y(x)$ is called the extremal of the functional**.

Calculus of variations is actually based on the use of Euler's equation in different interpretations.

The existence of a solution of the Euler equation for an extremal is a necessary condition for the minimum or maximum of the functional (3.1) to be reached on it. But, as in the case of the extremum of a function $y(x)$, the necessary conditions for the extremum of a functional must necessarily be supplemented with sufficient conditions, with the help of which one can recognize both those functions on which the maximum or minimum of the functional is reached, and those on which, despite the fulfillment of the necessary conditions, the functional does not reach the extremum.

To determine the sufficient conditions for the extremum of the functional (3.1), let us return again to the Taylor series expansion (3.9) of the increment of the functional $\Delta J_y^F(\alpha)$.

As already indicated above, in the vicinity of the point $\alpha = 0$ due to the fulfillment of the condition (3.12), the first variation $\delta J_y^F(\alpha)$ approaches zero, therefore the increase of the functional is determined by the second variation $\delta^2 J_y^F(\alpha)$, since the other terms of the Taylor series approach zero faster than the second variation due to higher, but close to zero, degrees α.

It is clear that in the case of a minimum of the functional, both the increase of the functional $\Delta J_y^F(\alpha)$ around the point $\alpha = 0$, and its second variation $\delta^2 J_y^F(\alpha)$, will not be less than zero, because at the point $\alpha = 0$ of the minimum, the value of this functional is the smallest and any displacement from this point will either not lead to a change in the value of the functional, or lead to the growth of its value.

So, heuristically, the minimum of the functional (3.1) is reached at the extremal $y(x)$, for which the expression

$$\delta^2 J_y^F(\alpha) \geq 0, \tag{3.23}$$

where,(we remind)

$$\delta^2 J_y^F(\alpha) = \frac{\alpha^2}{2}\frac{d^2 J_y^F}{d\alpha^2}. \tag{3.24}$$

By analogy, we conclude that the maximum of the functional (3.1) is reached at the extremal $y(x)$, for which the expression

$$\delta^2 J_y^F(\alpha) \leq 0. \tag{3.25}$$

Since expressions (3.23), (3.25) are included in the square according to expression (3.24), the signs of these inequalities are determined exclusively by the signs α of the second derivative of the functional increment, which can be written as follows:

$$\begin{aligned}\frac{d^2 J_y^F}{d\alpha^2} &= \frac{d^2}{d\alpha^2}\left(\int_a^b F(x, y+\alpha\eta, y'+\alpha\eta')\,dx\right) = \\ &= \int_a^b \frac{d^2}{d\alpha^2} F(x, y+\alpha\eta, y'+\alpha\eta')\,dx = \\ &= \int_a^b \left(F_{yy}\eta^2 + \left(F_{yy'} + F_{y'y}\right)\cdot\eta\eta' + F_{y'y'}(\eta')^2\right)dx.\end{aligned} \tag{3.26}$$

In order to obtain the square of the function η, in the middle term on the right-hand side of expression (3.26), which will facilitate the analysis of expression (3.26) as a whole, let's take the integral of this middle term by parts. After substituting the value obtained by integration by parts of the integral into expression (3.26), we obtain

$$\frac{d^2 J_y^F(\alpha)}{d\alpha^2} = \int_a^b \left(\left(F_{yy} - \frac{d}{dx}F_{yy'}\right)\eta^2 + F_{y'y'}(\eta')^2\right)dx. \tag{3.27}$$

We remind that that the auxiliary function $\eta(x)$ is arbitrary, and therefore it can be taken as shown in Fig.4.

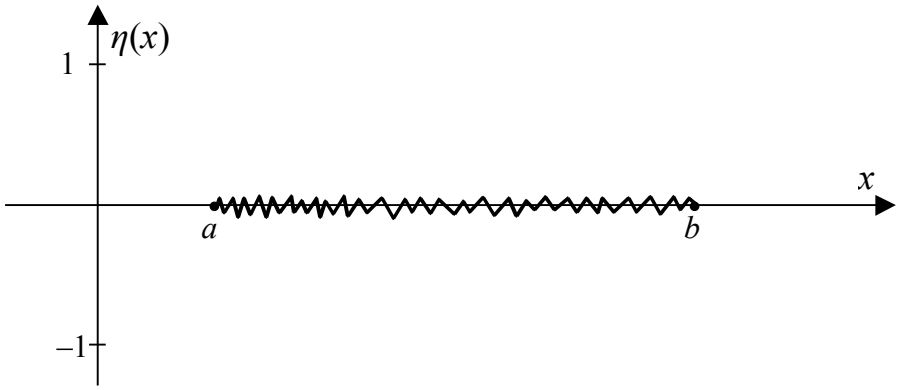


Figure 4 – Graph of the auxiliary function $\eta(x)$ for the expression (3.27)

With such a choice, we will have numerically close to zero values $(\eta)^2$ and hundreds of times larger values $(\eta')^2$. Therefore, the sign of expression (3.27) will be completely determined by the sign of the coefficient at $(\eta')^2$, that is, the sign at $F_{y'y'}$.

Based on the expressions (3.23), (3.25) and all of the above, it can be asserted that the minimum of the functional (3.1) is reached at the extremal $y(x)$ within the limits $x \in [a, b]$, if for all $x \in [a, b]$ we have

$$F_{y'y'} \geq 0\,, \tag{3.28}$$

and maximum if –

$$F_{y'y'} \leq 0\,. \tag{3.29}$$

Conditions (3.28), (3.29) are sufficient conditions for reaching the extremal $y(x)$ within the extremum of the functional (3.1).

These conditions are often called Legendre's conditions, after the mathematician who derived them.

3.3 Euler's equation and its analysis

Let's write Euler's equation (3.22), using the formula of the complete derivative of the function of three variables when differentiating the component $\frac{d}{dx}F_{y'}$. We will get

$$F_y - \frac{\partial F_{y'}}{\partial x}\cdot\frac{dx}{dx} - \frac{\partial F_{y'}}{\partial y}\cdot\frac{dy}{dx} - \frac{\partial F_{y'}}{\partial y'}\cdot\frac{dy'}{dx} = 0\,, \tag{3.30}$$

or

$$F_y - F_{y'x} - F_{y'y}\cdot y' - F_{y'y'}\cdot y'' = 0\,. \tag{3.31}$$

From the expression (3.31), it is clear that the Euler equation is a nonlinear differential equation of the second order, for which there is no single method of solution.

An important case is when the function $F(\bullet)$ in the functional (3.1) clearly does not depend on ***x***, that is, when

$$F(x, y, y') = F(y, y')\,. \tag{3.32}$$

In this case, instead of equation (3.31), we will have the equation

$$F_y - F_{y'y} \cdot y' - F_{y'y'} \cdot y'' = 0, \tag{3.33}$$

which is easily transformed into an equation by multiplying by y' –

$$\frac{d}{dx}\left(F - y' \cdot F_{y'}\right) = 0. \tag{3.34}$$

In turn, it follows from equation (3.34) that

$$F - y' \cdot F_{y'} = C, \tag{3.35}$$

where ***C*** is a constant.

The expression (3.35) obtained in this way is called the first integral of Euler's equation. It is a first-order differential equation that clearly does not depend on , and therefore can always be solved.

In some optimization problems, the Euler equation, which is a nonlinear differential equation of the second order, is convenient to present as a system of two differential equations of the first order, to obtain which a new variable is introduced

$$p = F_{y'}. \tag{3.36}$$

Substituting (3.36) into (3.22), we get

$$F_y = \frac{dp}{dx}. \tag{3.37}$$

Let's introduce the function ***H***, which specifies the first integral of the Euler equation, that is, the function

$$H = F - y' \cdot F_{y'}. \tag{3.38}$$

Substituting (3.36) into (3.38), we get

$$H = F - y' \cdot p. \tag{3.39}$$

Differentiating the function H for y and for p, we get

$$\begin{cases} \dfrac{\partial H}{\partial y} = F_y, \\ \dfrac{\partial H}{\partial p} = -y', \end{cases} \tag{3.40}$$

Given that $y' = \dfrac{dy}{dx}$, and equality (3.37), the system of equations (3.40) can be rewritten as follows:

$$\begin{cases} \dfrac{\partial H}{\partial y} = \dfrac{dp}{dx}, \\ \dfrac{\partial H}{\partial p} = -\dfrac{dy}{dx}. \end{cases} \tag{3.41}$$

The system of equations (3.41) is another form of representation of the Euler equation (3.22), which is called canonical.

It will not be superfluous to remind that the **problem of finding the extremal $y(x)$, which delivers the extremum of the functional (3.1) and is a curve pinched at the ends, is usually called the simplest in the classical calculus of variations.**

To continue the analysis of sufficient conditions for reaching an extremal $y(x)$ within $x \in [a, b]$ the extremum of the functional (3.1), let us return to the expanded notation of the Euler equation (3.31). This expression can be rewritten like this –

$$y'' = \frac{F_y - F_{y'x} - F_{y'y} \cdot y'}{F_{y'y'}}. \tag{3.42}$$

It follows from the expression (3.42) that a function $y(x)$ in order $x \in [a, b]$ to claim the role of an extremal must have a second derivative y'' in the domain, and its first derivative y' must satisfy additional conditions in some cases.

It is clear from (3.42) that at

$$F_{y'y'} > 0 \tag{3.43}$$

for the minimum of the functional (3.1) and at

$$F_{y'y'} < 0 \tag{3.44}$$

for its maximum, no other conditions need to be imposed on the first derivative y' in the domain $x \in [a, b]$ except that it must exist in this domain.

But if in certain points of the region $x \in [a, b]$

$$F_{y'y'} = 0, \tag{3.45}$$

then it is necessary that at these same points the first derivative y' numerically coincides with the value of the expression $\frac{F_y - F_{y'x}}{F_{y'y}}$, that is, that the equality is fulfilled at these points

$$y' = \frac{F_y - F_{y'x}}{F_{y'y}}, \tag{3.46}$$

which follows from the necessity to have, in order to ensure the existence in the domain $x \in [a, b]$ of the second derivative y'', in addition to the fulfillment of the condition (3.45), also the fulfillment of the condition

$$F_y - F_{y'x} - F_{y'y} \cdot y' = 0. \tag{3.47}$$

It is clear that if equality (3.45) holds for all points of the region $x \in [a, b]$, then equality (3.47) must also hold for all points of this region.

It should be noted that if equality (3.47) holds for all points of the region $x \in [a, b]$, then this means that the integrand function $F(x, y, y')$ in the functional (3.1) depends on the first derivative linearly, i.e. that this functional has the form

$$J_y^F = \int_a^b \left(M(x, y) + N(x, y) \cdot y' \right) dx. \tag{3.48}$$

Functionals of this type are called degenerate.

For them, the Euler equation is not differential since

$$F_y - \frac{d}{dx}F_{y'} = \frac{\partial M}{\partial y} + y'\frac{\partial N}{\partial y} - \frac{d}{dx}\big(N(x,y)\big) = \\ = \frac{\partial M}{\partial y} + \frac{\partial N}{\partial y}\cdot y' - \frac{\partial N}{\partial y}\cdot\frac{dy}{dx} - \frac{\partial N}{\partial x}\cdot\frac{dx}{dx} = \frac{\partial M}{\partial y} - \frac{\partial N}{\partial x} = 0. \tag{3.49}$$

It follows from the expression (3.49) that the Euler equation for degenerate functionals does not contain derivatives of the extremal $y(x)$.

Opening the outer brackets in the functional (3.47), it can be written in the form

$$J_y^F = \int_a^b M\,dx + N\,dy, \tag{3.50}$$

that is, it turns into an integral of a complete differential.

And, as is known from the course of mathematical analysis, the value of such an integral does not depend on the path of integration, that is, its value on all functions $y(x)$.on which the functional (3.50) is defined is the same, which in our case, in turn, means that the extremum of such the functional is reached on any function from its domain.

If the functional (3.1) is not degenerate, that is, it cannot be represented in the form (3.50), and the condition (3.47) is fulfilled, then this means that the extremum of the functional cannot be reached in the class of smooth functions. In this case, it should be looked for in the class of piecewise-smooth functions whose graphs have breaks.

3.4 Finding extremums of functionals depending on several functions and their first derivatives

Let's consider the problem of finding the extremum of the functional (3.3), which connects a set of functions

$$\{y_1(x), y_2(x), \ldots, y_n(x)\}, \tag{3.51}$$

which defines the surface in n-dimensional space, and the set $\{y_1'(x), y_2'(x), \ldots, y_n'(x)\}$ first derivatives of these functions in the same space.

It is clear that if the extremum of the functional (3.3) exists, then its extremals are in the set (3.51).

We construct the method for finding extremals of the functional (3.3) in the following way: we set the variation of only the function $y_1(x)$, and we fix all other functions from $y_2(x)$ to $y_n(x)$ inclusive and their derivatives by converting them into constants in this way, forming a set in this way

$$\begin{cases} y_2(x) = C_2,\ y_2'(x) = C_{22}, \\ y_3(x) = C_3,\ y_3'(x) = C_{33}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_n(x) = C_n,\ y_n'(x) = C_{nn}. \end{cases} \tag{3.52}$$

In this case, the functional (3.3) will have the form

$$J_{y_1}^F = \int_a^b F(x, y_1, C_2, C_3, \ldots, C_n; y_1', C_{22}, C_{33}, \ldots, C_{nn})\,dx, \tag{3.53}$$

which formally does not differ from the functional (3.1).

The minimum of such a functional, as we have already established earlier, is reached at the extremal $y_1^*(x)$, which is a solution of the Euler equation

$$\frac{\partial F}{\partial y_1} - \frac{d}{dx}\frac{\partial F}{\partial y_1'} = 0, \tag{3.54}$$

or (using previously introduced notations)

$$F_{y_1} - \frac{d}{dx}F_{y_1'} = 0. \tag{3.55}$$

And then we set the variation of only the function $y_2(x)$, and fix all the other functions $y_1(x), y_3(x), y_4(x), \ldots, y_n(x)$ of the set (3.51) and their derivatives, turning them into constants in this way and forming a set

$$\begin{cases} y_1(x) = C_1, \ y_1'(x) = C_{11}, \\ y_3(x) = C_3, \ y_3'(x) = C_{33}, \\ y_4(x) = C_4, \ y_4'(x) = C_{44}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_n(x) = C_n, \ y_n'(x) = C_{nn}. \end{cases} \tag{3.56}$$

In this case, the functional (3.3) will have the form

$$J_{y_2}^F = \int_a^b F\left(x, C_1, y_2, C_3, C_4, \ldots, C_n; C_{11}, y_2', C_{33}, C_{44}, \ldots, C_{nn}\right) dx, \tag{3.57}$$

which also formally does not differ from the functional (3.1).

The minimum of such a functional, by analogy with the previous case, is reached at the extremal $y_2^*(x)$, which is a solution of the Euler equation

$$F_{y_2} - \frac{d}{dx}F_{y_2'} = 0. \tag{3.58}$$

We will continue this process until we get an extremal $y_n^*(x)$, which is a solution to the Euler equation

$$F_{y_n} - \frac{d}{dx}F_{y_n'} = 0. \tag{3.59}$$

And now let's reduce all Euler's equations, from (3.55) to (3.59), into one system

$$\begin{cases} F_{y_1} - \dfrac{d}{dx}F_{y_1'} = 0, \\ F_{y_2} - \dfrac{d}{dx}F_{y_2'} = 0, \\ \ldots\ldots\ldots\ldots\ldots \\ F_{y_n} - \dfrac{d}{dx}F_{y_n'} = 0. \end{cases} \tag{3.60}$$

It is clear that the set of functions

$$\left\{y_1^*(x), y_2^*(x), \ldots, y_n^*(x)\right\}, \tag{3.61}$$

which is a compatible solution of the system of Euler equations (3.60), and will be the set of extrema of the functional (3.3), at which it reaches an extremum.

As in the case of the functional (3.1), the fact that the set of functions (3.61) is a solution of the system (3.60) provides only a necessary condition for the existence of the extremum of the functional (3.3).

To check the sufficient conditions for the existence of the extremum of the functional (3.3) on the set of extremals (3.61), it is necessary, as in the case of the functional (3.1), to make sure that the Legendre conditions are fulfilled, which for one extremal had the form (3.28) for the minimum and (3.29) for maximum, and for the system of extremals (3.61) will have the form (for the minimum) –

$$\begin{cases} F_{y_1'y_1'} \geq 0, \\ \begin{vmatrix} F_{y_1'y_1'} & F_{y_1'y_2'} \\ F_{y_2'y_1'} & F_{y_2'y_2'} \end{vmatrix} \geq 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \begin{vmatrix} F_{y_1'y_1'} & F_{y_1'y_2'} \ldots F_{y_1'y_n'} \\ F_{y_2'y_1'} & F_{y_2'y_2'} \ldots F_{y_2'y_n'} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ F_{y_n'y_1'} & F_{y_n'y_2'} \ldots F_{y_n'y_n'} \end{vmatrix} \geq 0. \end{cases} \tag{3.62}$$

Sufficient conditions for the existence of the maximum of the functional (3.3) on the set of extremals (3.61) according to Legendre will have a form similar to (3.62), but the signs of the inequalities in them will be opposite.

3.5 Finding extremums of functionals depending on one function and its older derivatives

Consider the functional (3.2), which connects not only the function $y(x)$ and its derivative $y'(x)$, as in the case of (3.1), but also higher derivatives $y''(x), \ldots, y^{(n)}(x)$.

Back in the first half of the eighteenth century, Euler proved that a function $y(x)$ will be an extremal of the functional (3.2) if it is a solution of the equation

$$F_y - \frac{d}{dx}F_{y'} + \frac{d^2}{dx^2}F_{y''} - \cdots + (-1)^n \cdot \frac{d^n}{dx^n}F_{y^{(n)}} = 0. \tag{3.63}$$

Euler obtained this equation by the method of mathematical induction.

He first considered the problem of finding a function $y(x)$ at which the functional (3.1) reaches a minimum. We have described this process in detail in the previous sections. Euler then considered the functional

$$J = \int_a^b F(x, y, y', y'')\,dx, \tag{3.64}$$

for which, following the same path with the selection of the first variation, equating it to zero and taking the second part of the integral by parts, we obtained the condition that the function $y(x)$ is extremal if it is a solution of the equation

$$F_y - \frac{d}{dx}F_{y'} + \frac{d^2}{dx^2}F_{y''} = 0. \tag{3.65}$$

After that, Euler concluded that the functional (3.2) will reach an extremum on functions that are solutions of equation (3.63). After considering several examples with derivatives with an order higher than the second, he was convinced that this was the case. And the strict proof that the extremal of the functional (3.2) is a function $y(x)$ satisfying equation (3.63) was carried out by Poisson - that is why this equation entered the calculus of variations with a double name - the Euler-Poisson equation.

For the functional (3.64), equation (3.63), which reduces to (3.65), is a fourth-order differential equation. Its solution will be the function $y(x, C_1, C_2, C_3, C_4)$, which contains four arbitrary constants C_1, C_2, C_3, C_4, for the definition of which you need to have four equations. Such equations will be the boundary conditions

$$\begin{cases} y(a, C_1, C_2, C_3, C_4) = A_1, \\ y'(a, C_1, C_2, C_3, C_4) = A_2, \\ y(b, C_1, C_2, C_3, C_4) = B_1, \\ y'(b, C_1, C_2, C_3, C_4) = B_2. \end{cases} \tag{3.66}$$

It is clear that the necessary condition for the extremum of the functional (3.2), in which the highest derivative of the unknown function $y(x)$ is the derivative $y'''(x)$ of the third order, will be the condition that this function satisfies the equation (3.63), which in this case will have the sixth order. Six arbitrary constants of this extremal will need to be found from the boundary conditions of the type (3.66), with the difference that there will already be six equations and they will set at the boundaries not only the values of the extremal itself and its first derivative, as in the case of minimizing the functional (3.65), but also of its second derivative, that is, the system of equations (3.66) will be supplemented with more equations

$$\begin{cases} y''(a, C_1, C_2, C_3, C_4, C_5, C_6) = A_3, \\ y''(b, C_1, C_2, C_3, C_4, C_5, C_6) = B_3. \end{cases} \tag{3.67}$$

And, of course, in all equations (3.66) for this case, arbitrary constants C_5 and C_6 shall be added.

Legendre's conditions, which distinguish the minimum of the functional (3.2) at the extremal $y(x)$ from the maximum, turned out to be very simple for this problem.

Studies have shown that, in order for the minimum of the functional (3.2) to be reached at the extremal $y(x)$, it is sufficient to fulfill the condition

$$F_{y^{(n)}y^{(n)}} \geq 0, \tag{3.68}$$

and for the maximum -

$$F_{y^{(n)}y^{(n)}} \leq 0. \tag{3.69}$$

3.6 Python programs for finding unconditional extrema of functionals

The Python program for exploring the unconditional extremum functioned

$$\boldsymbol{J_1 = \int_a^b F_1(t, y, y')dt}$$

in case when $a = 0$, $b = 1$, $F_1(t, y, y') = t^2 + y^2 + ty + (y')^2$, and the extremal $y(t)$ starts at the point $(y(0) = 0,\ y'(0) = 1)$.

(Program 11)

```
In [1]: import sympy
In [2]: from sympy import*
In [3]: from IPython.display import*
In [4]: init_printing(use_latex=True)
In [5]: t,C1,C2=symbols('t C1 C2')
In [6]: y=Function('y')(t)
In [7]: z=Function('z')(t)
In [8]: z=y.diff(t)
In [9]: u=Function('u')(t)
In [10]: u=t**2+y**2+t*y+z**2
In [11]: de=Eq(u.diff(y)-u.diff(z,t),0)
In [12]: display(de)
```

$$t + 2y(t) - 2\frac{d^2}{dt^2}y(t) = 0$$

```
In [13]: des=dsolve(de)
In [14]: display(des)
```

$$y(t) = C_1 e^{-t} + C_2 e^{t} - \frac{t}{2}$$

```
In [15]: eq1=des.rhs.subs(t,0)
In [16]: eq1
```

Out [16]: $C_1 + C_2$

```
In [17]: eq2=des.rhs.diff(t).subs (t,0)
In [18]: eq2
```

Out [18]: $-C_1 + C_2 - \frac{1}{2}$

```
In [19]: seq=solve ([eq1, eq2-1], C1, C2)
In [20]: seq
```

Out [20]: $\left\{C_1: -\frac{3}{4}, C_2: \frac{3}{4}\right\}$

```
In [21]: rez=des.rhs.subs([(C1,seq[C1]),(C2, \
         seq[C2])])
In [22]: rez
```

Out [22]: $-\frac{t}{2} + \frac{3}{4}e^{t} - \frac{3}{4}e^{-t}$

```
In [23]: F=Lambda (t, rez)
In [24]: display(Latex('$y(t)=' \
          +str(latex(F(t)))+'$'))
```

$$y(t) = -\frac{t}{2} + \frac{3e^{t}}{4} - \frac{3e^{-t}}{4}$$

```
In [25]: import numpy as np
In [26]: import matplotlib.pyplot as plt
In [27}: x=symbols('x')
In [28]: expr=-x/2+3*exp(x)/4 \
         -3*exp(-x)/4
In [29]: f=lambdify(x,expr,"numpy")
In [30]: x=np.linspace(0,1,21)
In [31]: f=f(x)
In [32]: fig=plt.figure(facecolor='white')
In [33]: plt.plot(x,f,'-r',linewidth=3)
Out [33]: [<matplotlib.lines.Line2D \
          at 0x1d0a7047970>]
```

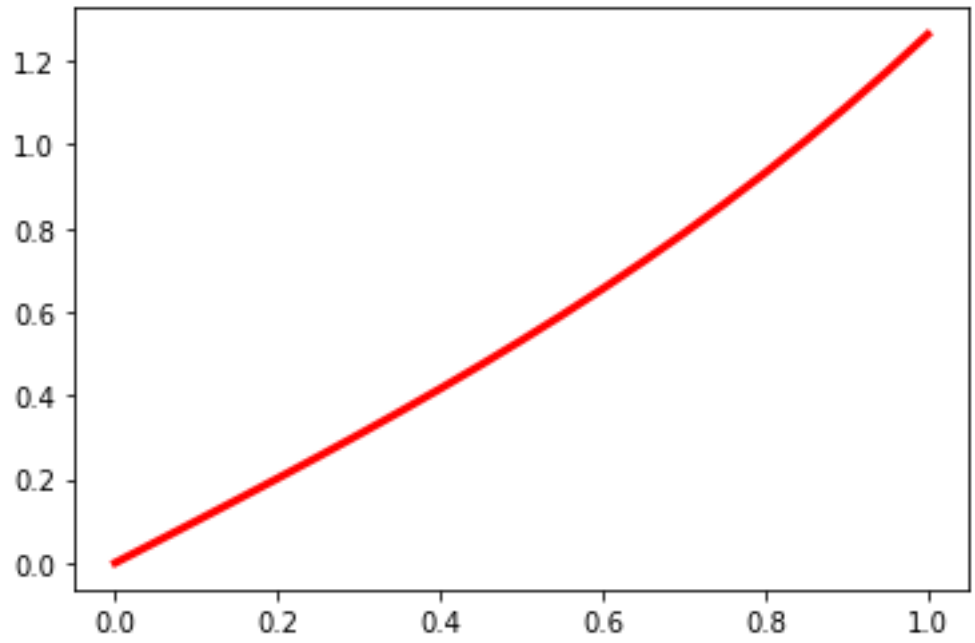


Figure 5. Graph of the extremal of the functional $J_1 = \int_a^b F_1(t, y, y')dt$ in case when a=0, b=1, $F_1(t, y, y') = t^2 + y^2 + ty + (y')^2$, and the extremal $y(t)$ begins in points $(y(0) = 0, y'(0) = 1)$

End of program 11.

The Python program for exploring the unconditional extremum functioned

$$J_1 = \int_a^b F_1(t, y, y', y'')dt$$

in case when $a = 0$, $b = 1$,
$F_1(t, y, y', y'') = t^2 + y^2 + (y')^2 + (y'')^2 + ty + ty' + ty'' + y'y''$, and the extremal $y(t)$ starts at the point $(y(0) = 0,\ y'(0) = 1,\ y''(0) = 0,\ y'''(0) = -1)$
(Program 12)

```
In [1]: import sympy
In [2]: from sympy import*
In [3]: from IPython.display import*
In [4]: init_printing(use_latex=True)
In [5]: t=symbols('t')
In [6]: y=Function('y')(t)
In [7]: z=Function('z')(t)
In [8]: w=Function('w')(t)
In [9]: z=y.diff(t)
In [10]: w=z.diff(t)
In [11]: u=Function('u')(t)
In [12]: u=t**2+y**2+z**2+w**2 \
                +t*y+t*z+t*w+z*w
In [13]: de1=Eq(u.diff(y)-u.diff(z,t) \
                +u.diff(w,1,t,2),0)
```

```
In [14]: display(de1)
```

$$t + 2y(t) - 2\frac{d^2}{dt^2}y(t) + 2\frac{d^4}{dt^4}y(t) - 1 = 0$$

```
In [15]: des1=dsolve(de1)
In [16]: display(des1)
```

$$y(t) = -\frac{t}{2} + \left(C_1 sin\left(\frac{t}{2}\right) + C_2 cos\left(\frac{t}{2}\right)\right)e^{\frac{-\sqrt{3}t}{2}} + \left(C_3 sin\left(\frac{t}{2}\right) + C_4 cos\left(\frac{t}{2}\right)\right)e^{\frac{\sqrt{3}t}{2}} + \frac{1}{2}$$

```
In [17]: eq11=des1.rhs.subs(t,0);eq11
Out[17]:
```

$$C_2 + C_4 + \frac{1}{2}$$

```
In [18]: eq12=des1.rhs.diff(t).subs(t,0)
In [19]: eq12
Out[19]:
```

$$\frac{1}{2}C_1 - \frac{\sqrt{3}}{2}C_2 + \frac{1}{2}C_3 + \frac{\sqrt{3}}{2}C_4 - \frac{1}{2}$$

```
In [20]: eq13=des1.rhs.diff(t,t).subs(t,0)
In [21]: eq13
Out[21]:
```

$$-\frac{\sqrt{3}}{2}C_1 + \frac{1}{2}C_2 + \frac{\sqrt{3}}{2}C_3 + \frac{1}{2}C_4$$

```
In [22]: eq14=des1.rhs.diff(t,3).subs(t,0)
In [23]: eq14
Out[23]:
```

$$C_1 + C_3$$

```
In [24]: var('C1 C2 C3 C4')
In [25]: seq1=solve([eq11,eq12-1,eq13,\
          eq14+1],C1,C2,C3,C4)
In [26]: seq1
```

Out [26]:

$$\left\{C_1 : -\frac{1}{2} - \frac{\sqrt{3}}{12}, C_2 : -\frac{1}{4} - \frac{2\sqrt{3}}{3}, C_3 : -\frac{1}{2} + \frac{\sqrt{3}}{12}, C_4 : -\frac{1}{4} + \frac{2\sqrt{3}}{3}\right\}$$

```
In [27]: rez1=des1.rhs.subs([(C1,seq1[C1]),(C2,seq1[C2]),(C3,seq1[C3]),(C4,seq1[C4])])
In [28]: rez1
Out [28]:
```

$$-\frac{t}{2} + \left(\left(-\frac{1}{2} - \frac{\sqrt{3}}{12}\right)sin\left(\frac{t}{2}\right) + \left(-\frac{1}{4} - \frac{2\sqrt{3}}{3}\right)cos\left(\frac{t}{2}\right)\right)e^{\frac{-\sqrt{3}t}{2}} +$$

$$+\left(\left(-\frac{1}{2} + \frac{\sqrt{3}}{12}\right)sin\left(\frac{t}{2}\right) + \left(-\frac{1}{4} + \frac{2\sqrt{3}}{3}\right)cos\left(\frac{t}{2}\right)\right)e^{\frac{\sqrt{3}t}{2}} + \frac{1}{2}$$

```
In [29]: F1=Lambda(t,rez1)
```

In [30]: display(Latex('$y(t)='+str(latex(F1(t)))+'$'))

$$y(t) = -\frac{t}{2} + \left((-\frac{1}{2} - \frac{\sqrt{3}}{12})sin\left(\frac{t}{2}\right) + (-\frac{1}{4} - \frac{2\sqrt{3}}{3})cos\left(\frac{t}{2}\right)\right)e^{\frac{-\sqrt{3}t}{2}} +$$

$$+\left((-\frac{1}{2} + \frac{\sqrt{3}}{12})sin\left(\frac{t}{2}\right) + (-\frac{1}{4} + \frac{2\sqrt{3}}{3})cos\left(\frac{t}{2}\right)\right)e^{\frac{\sqrt{3}t}{2}} + \frac{1}{2}$$

```
In [31]: import numpy as np
In [32]: import matplotlib.pyplot as plt
In [33]: x=np.linspace(0,1,21)
In [34]: def y(x):
             return -x/2+((-1/2-3**0.5/12)*np.sin(x/2)+(-1/4-2*3**0.5/3)*np.cos(x/2)) \
                        *np.exp(-x*3**0.5/2)+((-1/2+3**0.5/12)*np.sin(x/2) \
                        +(-1/4+2*3**0.5/3)*np.cos(x/2))*np.exp(x*3**0.5/2)+1/2
In [35]: fig=plt.figure(facecolor='white')
In [36]: plt.plot(x,y(x),'-r',linewidth=3)
Out[36]: [<matplotlib.lines.Line2D
at 0x1d0a7247250>]
```

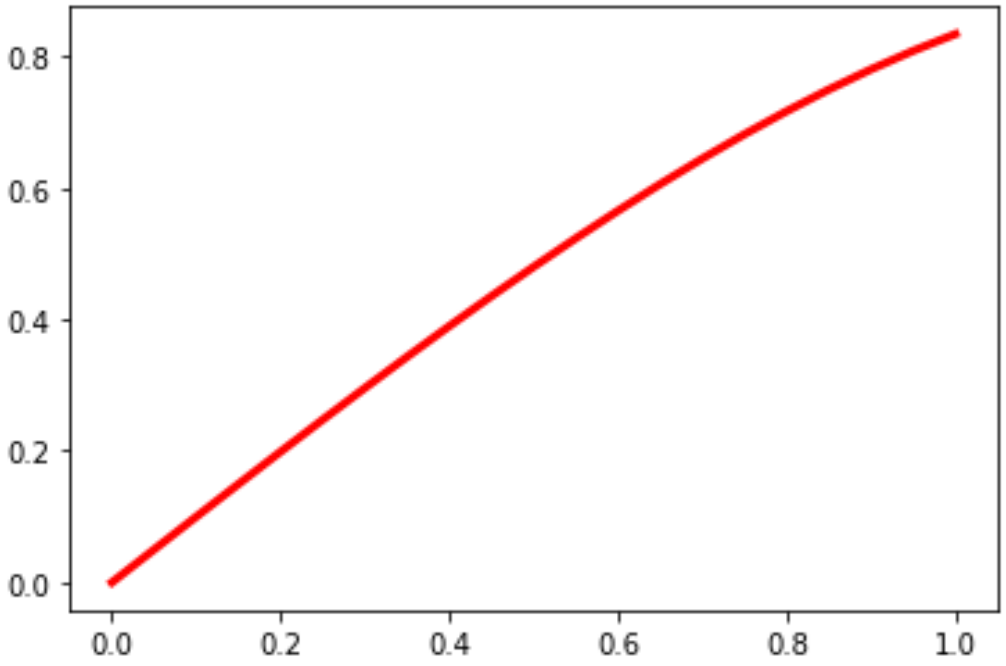


Figure 6. Graph of the extremal of the functional $J_1 = \int_a^b F_1(t, y, y', y'')dt$ in case when a=0, b=1, $F(t, y, y', y'') = t^2 + y^2 + (y')^2 + (y'')^2 + ty + ty' + ty'' + y'y''$, and the extremal y(t) starts at the point ($y(0)$=0, $y'(0) = 1$, $y''(0) = 0$, $y'''(0) = -1$)

End of program 12.

A Python program for exploring the unconditional extremum functioned

$$J_1 = \int_a^b F_1(t, y_1, y_2, y_3, y_1', y_2', y_3')dt$$

in case when a=0, b=1, $F_1(t, y_1, y_2, y_3, y_1', y_2', y_3') = t^2 + y_1^2 + y_2^2 + y_3^2 + 3y_1y_2 + (y_1')^2 + (y_2')^2 + (y_3')^2 + 5y_2y_3$, and the extremals $y_1(t), y_2(t), y_3(t)$ start at the points: $(y_1(0) = 0, y_1'(0) = 1), (y_2(0) = 0, y_2'(0) = 2), (y_3(0) = 0, y_3'(0) = -1)$
(Program 13)

```
In [1]: import sympy
In [2]: from sympy import*
In [3]: from IPython.display import*
In [4]: init_printing(use_latex=True)
In [5]: t=symbols('t')
In [6]: y1=Function('y1')(t)
In [7]: y2=Function('y2')(t)
In [8]: y3=Function('y3')(t)
In [9]: z1=Function('z1')(t)
In [10]: z1=y1.diff(t)
In [11]: z2=Function('z2')(t)
In [12]: z2=y2.diff(t)
In [13]: z3=Function('z3')(t)
In [14]: z3=y3.diff(t)
In [15]: u=Function('u')(t)
In [16]: u=t**2+y1**2+y2**2+y3**2 \
         +3*y1*y2+z1**2+z2**2+z3**2+5*y2*y3
In [17]: de11=Eq(u.diff(y1)-u.diff(z1,t),0)
In [18]: de12=Eq(u.diff(y2)-u.diff(z2,t),0)
In [19]: de13=Eq(u.diff(y3)-u.diff(z3,t),0)
In [20]: display(de11,de12,de13)
```

$$2y_1(t) + 3y_2(t) - 2\frac{d^2}{dt^2}y_1(t) = 0,$$

$$3y_1(t) + 2y_2(t) - 2\frac{d^2}{dt^2}y_2(t) + 5y_3(t) = 0,$$

$$-2\frac{d^2}{dt^2}y_3(t) + 2y_3(t) + 5y_2(t) = 0$$

```
In [21]: eq11=Eq(y1.diff(t)-z1,0)
In [22]: eq12=Eq(y2.diff(t)-z2,0)
In [23]: eq13=Eq(y3.diff(t)-z3,0)
In [24]: eq14=Eq(y1+3*y2/2-z1.diff(t),0)
In [25]: eq15=Eq(3*y1/2+y2+5*y3/2 \
          -z2.diff(t),0)
In [26]: eq16=Eq(-z3.diff(t)+5*y2/2+y3,0)
In [27]: des16=dsolve(eq11,eq12,eq13,\
          eq14,eq15,eq16);des16
Out [27]: ValueError: dsolve() and
 classify_ode() only work with functions
of one variable, not True
In [28]: import numpy as np
In [29]: from scipy.integrate import odeint
In [30]: import matplotlib.pyplot as plt
In [31]: def f(y,t):
                    y1,y2,y3,z1,z2,z3=y
                    return [z1,z2,z3,y1+3*y2/2,\
                    3*y1/2+y2+5*y3/2,\
                    5*y2/2+y3]
In [32]: y0=[0,0,0,1,2,-1]
In [33]: t=np.linspace(0,1,21)
In [34]: [y1,y2,y3,z1,z2,z3]=odeint(f,y0,t,\
          full_output=False).T
In [35]: fig=plt.figure(facecolor='white')
In [36]: plt.plot(t,y1,'-r',t,y2,'-g',t,y3,'-c',\
linewidth=3)
```

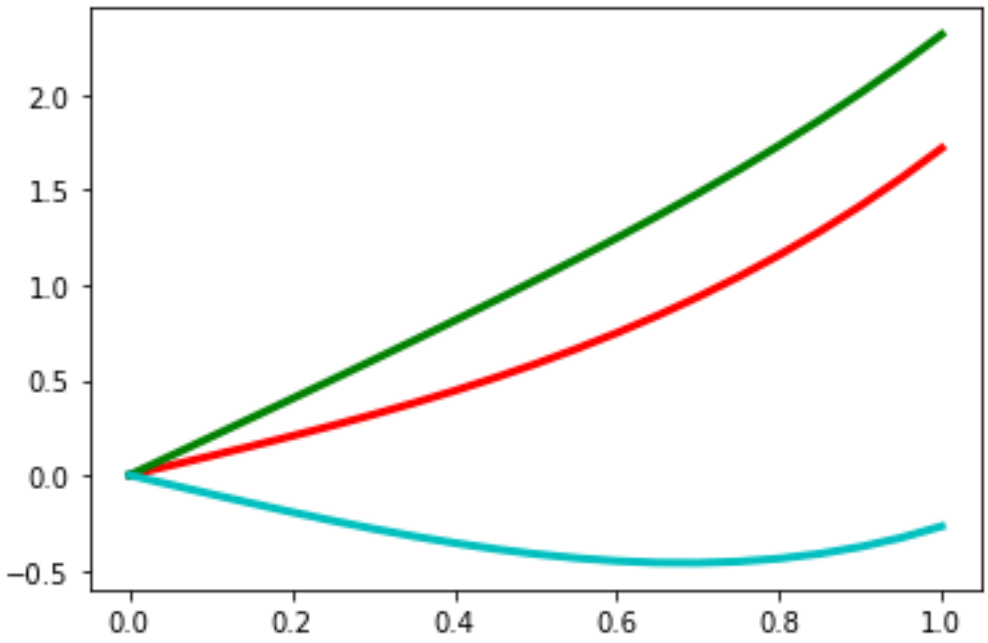


Figure 7. Graph of extremals of the functional $J_1 = \int_a^b F_1(t, y_1, y_2, y_3, y_1', y_2', y_3')dt$ in case when a=0, b=1, $F_1(t, y_1, y_2, y_3, y_1', y_2', y_3') = t^2 + y_1^2 + y_2^2 + y_3^2 + 3y_1y_2 + (y_1')^2 + (y_2')^2 + (y_3')^2 + 5y_2y_3$, and the extremals $y_1(t), y_2(t), y_3(t)$ start at the points: $(y_1(0) = 0,\ y_1'(0) = 1), (y_2(0) = 0, y_2'(0) = 2), (y_3(0) = 0, y_3'(0) = -1)$

End of program 13.

3.7 Tasks for self-testing

1. Explain the concept of "functional" and give examples of its definition.
2. How to meaningfully distinguish the absolute and relative extremes of the functional? Give an example.
3. How to determine the distance of zero and first order between two functions?
4. When is a strong relative extremum of the functional reached at the extreme? When is a weak relative extremum reached?
5. What is the consistent logical connection between the weak, strong and absolute extremes of the function? Is the logical sequence preserved in the reverse direction?
6. How to get the Euler equation? Give its interpretation.
7. How to obtain the Euler equation in the form of a nonlinear differential equation of the second order?
8. Obtain the first integral of the Euler equation. Give its interpretation.
9. Derive Euler's equation in canonical form.
10. What are the sufficient conditions for the existence of an extremum of a functional in the simplest problem of the calculus of variations? How to get them and how to use them to distinguish the maximum and minimum of the functional?
11. How to distinguish the minimum from the maximum of a functional, if $F_{y'y'} = 0$?
12. Which functionals are degenerate and what can be said about their extremals?
13. Derive the necessary conditions for the existence of an extremum of a functional that connects several unknown functions and their first derivatives.
14. What are the sufficient conditions for the existence of an extremum of a functional connecting several unknown functions and their first derivatives?
15. Derive the Euler–Poisson equation heuristically (according to Euler). Give its interpretation.
16. What are the sufficient conditions for the existence of an extremum of a functional that connects the extremal with its higher derivatives? How to distinguish the maximum and minimum of the functional in this case?
17. Show the command by which Euler's equation is formed
18. Show the command that implements the solution of the Euler equation
19. What is achieved by the command `seq=solve([eq1,eq2-1],C1,C2)`
20. What is the command `rez=des.rhs.subs([(C1,seq[C1]),(C2,seq[C2])])` for?
21. What is the command `var('C1 C2 C3 C4')` for?
22. Why is the command `seq1=solve([eq11,eq12-1,eq13,eq14+1],C1,C2,C3,C4)` needed?
23. Why do we need the command `display(de11,de12,de13)`?
24. What is the purpose of calling the command from scipy.integrate import odeint?
25 What command does the program use to determine the extremals of the functional?
26. Which program command transforms the system of three Euler equations of the 2nd order into a system of 6 equations of the 1st order?
27. What will those few commands look like, how should they be added to the program so that it determines what is achieved - maximum or minimum?

Chapter 4. STUDY OF FUNCTIONALS AT THE CONDITIONAL EXTREMUM

4.1 The method of uncertain Lagrange multipliers

Let the functional be given

$$J = \int_a^b F\left(x, y, z, y', z'\right) dx, \tag{4.1}$$

which is another form of notation of the functional (3.3) for n=2, and it is necessary to find such functions

$$y(x), z(x), \tag{4.2}$$

which deliver the extremum of the functional (4.1) under the condition that

$$\varphi(x, y, z) = 0, \tag{4.3}$$

that is, under the condition that all points of the curve given by expressions (4.2) lie on the surface (4.3).

Let's try to find the necessary conditions for the existence of the extremum of the functional (4.1) in the presence of the constraint (4.3).

Suppose that we found such functions $y(x)$ and $z(x)$, which deliver an extremum, for example, a minimum, of the functional (4.1).

Let's add to the functions $y(x)$ and $z(x)$ variations δy and δz, which satisfy the requirements

$$\begin{cases} \delta y > 0 \quad для \quad x \in [x_1, x_2], \\ \delta y = 0 \quad для \quad x \notin [x_1, x_2], \\ \delta z > 0 \quad для \quad x \in [x_1, x_2], \\ \delta z = 0 \quad для \quad x \notin [x_1, x_2], \\ [x_1, x_2] \subset [a, b], \end{cases} \tag{4.4}$$

and find the first variation δJ of the functional (4.1) when passing from the curve on the surface (4.3) described by the functions $y(x)$, $z(x)$, to the curve on the same surface described by the functions

$$\begin{cases} y(x) + \delta y, \\ z(x) + \delta z. \end{cases} \tag{4.5}$$

It is clear that

$$\delta J = \delta J_y + \delta J_z, \tag{4.6}$$

That is, the first variation δJ of the functional (4.1) will be equal to the sum of the first variation δJ_y of the same functional by the coordinate y at the constant coordinate z, and the first variation δJ_z by the coordinate z at the constant coordinate y.

It is obvious that at the minimum point (x_0, y_0, z_0) of the functional (4.1) the relation will be valid

$$\delta J = 0. \tag{4.7}$$

And this is possible due to independence y and z from each other only under the condition that

$$\begin{cases} \delta J_y = 0, \\ \delta J_z = 0. \end{cases} \tag{4.8}$$

But, as we have already seen when deriving Euler's equation, the first variations δJ_y, δJ_z can be presented in the form

$$\delta J_y = \int_a^b \left(F_y - \frac{d}{dx} F_{y'} \right) \cdot \delta y \cdot dx, \tag{4.9}$$

$$\delta J_z = \int_a^b \left(F_z - \frac{d}{dx} F_{z'} \right) \cdot \delta z \cdot dx. \tag{4.10}$$

So, substituting (4.9) and (4.10) into the relation (4.6), we will have

$$\delta J = \int_a^b \left(F_y - \frac{d}{dx} F_{y'} \right) \cdot \delta y \cdot dx + \int_a^b \left(F_z - \frac{d}{dx} F_{z'} \right) \cdot \delta z \cdot dx = 0. \tag{4.11}$$

Let $[x_1, x_2]$ is a small circle at the coordinate x of the point (x_0, y_0, z_0), in which the minimum of the functional (4.1) is reached, and $x_0 \in [x_1, x_2]$.

Then expressions

$$\sigma_1 = \int_{x_1}^{x_2} \delta y \cdot dx, \tag{4.12}$$

$$\sigma_2 = \int_{x_1}^{x_2} \delta z \cdot dx \tag{4.13}$$

will define small rectangular planes σ_1, σ_2 on coordinate planes (x, y), (x, z) in the vicinity of a point x_0 with sides $dx, \delta y$ and dx, δz, and expressions (4.9), (4.10) will define the volumes of prisms with bases dx, δy and dx, δz, which are under the surfaces

$$F_y - \frac{d}{dx} F_{y'}, \tag{4.14}$$

$$F_z - \frac{d}{dx} F_{z'}. \tag{4.15}$$

But we know that the volume of a rectangular prism with a small rectangular plane at the base can be found by multiplying the area of the base by the height of this prism, which gives us the right to rewrite expression (4.11) in the form

$$\delta J = \left(F_y - \frac{d}{dx} F_{y'} \right)\Bigg|_{x_0} \cdot \sigma_1 + \left(F_z - \frac{d}{dx} F_{z'} \right)\Bigg|_{x_0} \cdot \sigma_2 = 0. \tag{4.16}$$

As we have already noted, the curve described by the functions (4.5) also lies on the surface (4.3), so it also has a valid equality (4.3), which in this case takes the form

$$\varphi(x, y + \delta y, z + \delta z) = 0. \tag{4.17}$$

Let's integrate equations (4.3), (4.17) in the domain $x \in [x_1, x_2]$. It is clear that as a result of integration we will get

$$\int_{x_1}^{x_2} \varphi(x, y + \delta y, z + \delta z)\, dx = 0, \tag{4.18}$$

$$\int_{x_1}^{x_2} \varphi(x, y, z)\, dx = 0. \tag{4.19}$$

Subtract equation (4.19) from (4.18). We will get

$$\int_{x_1}^{x_2} \left(\varphi(x, y + \delta y, z + \delta z) - \varphi(x, y, z)\right) dx = 0. \tag{4.20}$$

Formally, under the integral in equation (4.20) is the increment $\Delta\varphi$ of the function $\varphi(x, y, z)$, i.e

$$\Delta\varphi = \varphi(x, y + \delta y, z + \delta z) - \varphi(x, y, z), \tag{4.21}$$

which it receives when moving from a point on the surface with coordinates (x, y, z) to a point on the same surface with coordinates $(x, y + \delta y, z + \delta z)$.

We remind that around a point $x_0 \in [x_1, x_2]$, the increment of the function is equal to its first variation, i.e

$$\Delta\varphi = \delta\varphi. \tag{4.22}$$

But

$$\delta\varphi = \varphi_y \cdot \delta y + \varphi_z \cdot \delta z. \tag{4.23}$$

So, taking into account expressions (4.21), (4.22), (4.23), equation (4.20) can be rewritten as follows:

$$\int_{x_1}^{x_2} \left(\varphi_y \cdot \delta y + \varphi_z \cdot \delta z\right) dx = 0, \tag{4.24}$$

or like this:

$$\int_{x_1}^{x_2} \varphi_y \cdot \delta y \cdot dx + \int_{x_1}^{x_2} \varphi_z \cdot \delta z \cdot dx = 0. \tag{4.25}$$

Taking into account the expressions (4.12), (4.13), the volumes of rectangular prisms given by the integrals in the expression (4.25) can be written as follows:

$$\begin{cases} \int_{x_1}^{x_2} \varphi_y \cdot \delta y \cdot dx \approx \varphi_y\big|_{x_0} \cdot \sigma_1, \\ \int_{x_1}^{x_2} \varphi_z \cdot \delta z \cdot dx \approx \varphi_z\big|_{x_0} \cdot \sigma_2. \end{cases} \tag{4.26}$$

Substituting the expression (4.26) into (4.25), we have in the neighborhood of the point $x_0 \in [x_1, x_2]$ –

$$\varphi_y \cdot \sigma_1 = -\varphi_z \cdot \sigma_2, \tag{4.27}$$

where

$$\sigma_2 = -\frac{\varphi_y}{\varphi_z} \cdot \sigma_1 . \tag{4.28}$$

Substituting the expression (4.28) into (4.16), we have in the neighborhood of the point $x_0 \in [x_1, x_2]$ –

$$\left(F_y - \frac{d}{dx} F_{y'} \right) \cdot \sigma_1 - \frac{\varphi_y}{\varphi_z} \cdot \left(F_z - \frac{d}{dx} F_{z'} \right) \cdot \sigma_1 = 0 . \tag{4.29}$$

It follows from equation (4.29) that

$$\frac{F_y - \frac{d}{dx} F_{y'}}{\varphi_y} = \frac{F_z - \frac{d}{dx} F_{z'}}{\varphi_z} . \tag{4.30}$$

Since we carried out all the operations before obtaining relations (4.30) using the operation of integration by the coordinate $\boldsymbol{x}$, we can equate each relation in the expression (4.30) to an unknown function $\lambda(x)$, i.e.,

$$\begin{cases} \dfrac{F_y - \dfrac{d}{dx} F_{y'}}{\varphi_y} = \lambda(x), \\ \dfrac{F_z - \dfrac{d}{dx} F_{z'}}{\varphi_z} = \lambda(x). \end{cases} \tag{4.31}$$

In turn, relation (4.31) can be rewritten as follows:

$$\begin{cases} F_y - \dfrac{d}{dx} F_{y'} + \lambda(x) \cdot \varphi_y = 0, \\ F_z - \dfrac{d}{dx} F_{z'} + \lambda(x) \cdot \varphi_z = 0. \end{cases} \tag{4.32}$$

After all these explanations, we can state that in order for the functions (4.2) to deliver the extremum of the functional (4.1) in the presence of constraints (4.3), it is necessary that they be a solution of the equations (4.32).

Proceeding from the expression (4.31), formally in the relations (4.32) it would be necessary to put a “minus” sign on the members $\lambda(x)\varphi_y$, $\lambda(x)\varphi_z$, but since the function $\lambda(x)$ is still undefined, any sign can be put on it. Why it is convenient for us to put a “plus” will become clear from the statements that follow. And let's start these calculations by constructing a function

$$L = F(x, y, z, y', z') + \lambda(x) \cdot \varphi(x, y, z) . \tag{4.33}$$

At the minimum point (x_0, y_0, z_0) of the functional (4.1), the expression (4.3) is valid, so it is clear that at this point

$$L = F . \tag{4.34}$$

And therefore we have the right to go from finding the necessary conditions for the existence of the minimum of the functional J (4.1) to finding the conditions for the existence of the minimum of the functional J^L, where

$$J^L = \int_a^b L\,dx = \int_a^b \left(F + \lambda(x)\cdot\varphi(x,y,z)\right)dx\,. \tag{4.35}$$

This transition allows us to transfer the problem of finding extremals that deliver the conditional extremum of the functional (4.1) into the problem of finding extremals that deliver the unconditional extremum of the functional (4.35), which we already know how to solve, since we know that in the absence of restrictions, the minimum of the functional is reached at functions that are the solution of the system of Euler's equations obtained from the integral function of several variables of this functional.

It is quite obvious that for our case this system takes the form:

$$\begin{cases} L_y - \dfrac{d}{dx}L_{y'} = 0, \\ L_z - \dfrac{d}{dx}L_{z'} = 0. \end{cases} \tag{4.36}$$

Finding L_y, L_z, $L_{y'}$, $L_{z'}$, we see that system (4.36) is identical to system (4.32).

But the two equations of the system (4.36) or, which is the same thing, the system (4.32) do not allow us to unambiguously find the three unknown functions

$$y(x), z(x), \lambda(x). \tag{4.37}$$

So they need to be supplemented with a third equation, which should be taken as the constraint equation (4.3). In this case, the system of defining equations is closed and such that it gives an unambiguous solution to the problem of finding a conditional extremum.

Lagrange, who introduced it for the first time, called the function $\lambda(x)$ an indeterminate multiplier, and therefore the method of finding extrema at which functionals acquire a conditional extremum was included in the calculus of variations, which is an integral part of functional analysis, under the name of Lagrange's method of indeterminate multipliers.

Remark. After, by introducing the Lagrange function L in the form (4.33), we transferred the conditional extremum problem to the category of the simplest problem of the calculus of variations on the unconditional extremum, the Legendre sufficient conditions introduced by us earlier, which distinguish the minimum from the maximum, also become valid for it, and the necessary conditions for the existence of an extremum for functionals that depend on higher derivatives. It is only necessary not to forget to substitute the Lagrange function L and its derivatives instead of the function F and its derivatives in the relations by which these conditions are determined. At the same time, if J has the form (3.3), and the restriction is analogous to (4.3), i.e

$$\varphi(x, y_1, y_2, \ldots, y_n) = 0\,, \tag{4.38}$$

then the structure of the Lagrange function L will be similar to (4.33), that is, it will have the form

$$L = F\left(x, y_1, y_2, \ldots, y_n, y_1', y_2', \ldots, y_n'\right) + \lambda(x)\cdot\varphi(x, y_1, \ldots, y_n)\,. \tag{4.39}$$

But if a system of equations acts as a constraint

$$\varphi_j(x, y_1, \ldots, y_n) = C_j,\ \ j = \overline{1, m},\ \ m \le n\,, \tag{4.40}$$

then the Lagrange function L should be taken in the form

$$L = F\left(x, y_1, y_2, \ldots, y_n, y_1', y_2', \ldots, y_n'\right) + \sum_{j=1}^{m}\lambda_j(x)\cdot\left(\varphi_j(x, y_1, \ldots, y_n) - C_j\right). \tag{4.41}$$

In this case, the system of Euler's equations (3.60), in which instead $F_{y_i}, F_{y_i'}, i=\overline{1,n}$ of must be $L_{y_i}, L_{y_i'}, i=\overline{1,n}$, solved together with the system of equations (4.40), since it is necessary to find not only n extremals $y_i(x)$, but also m undetermined Lagrange multipliers $\lambda_j(x)$.

4.2 The isoperimetric problem of finding extremals of functionals

In 1732, Leonard Euler made the first approach to solving the problem of finding extremals of functionals under the conditions of restrictions, also given by functionals, which was called isoperimetric. In 1744, he published the solution to this problem in the most general form.

The isoperimetric problem of finding extremals of functionals was formulated as follows: among curves of the same length, or, what is the same, of the same perimeter, find the curve that bounds the largest area.

Mathematically, it can be written as follows: find the curve $y(x)$, that delivers the extremum of the functional (3.1), i.e., the functional

$$J^F = \int_a^b F(x, y, y')\, dx,$$

which estimates the area of a given figure, provided that another functional

$$J^K = \int_a^b K(x, y, y')\, dx, \tag{4.42}$$

which defines the length of the perimeter of this figure, has a constant value J_0^K, i.e.,

$$\int_a^b K(x, y, y')\, dx = J_0^K. \tag{4.43}$$

Euler solved this problem in an extremely complicated way, which today can be interesting only to specialists in the history of mathematics.

We will present the solution of the isoperimetric problem, which was obtained by Lagrange 15 years later in a much simpler way using the method of uncertain factors.

The essence of the process of solving the problem by Lagrange is as follows: if we omit the upper bound in the functional (4.42), then we get an integral with a variable upper bound, i.e.,

$$\psi(x) = \int_a^x K(x, y, y')\, dx, \tag{4.44}$$

the derivative of which $\psi'(x)$, as is known from the course of mathematical analysis, will be equal to the integrand function, i.e.,

$$\psi'(x) = K(x, y, y'). \tag{4.45}$$

Equation (4.45) can be considered as a restriction of the form (4.40) for $j=1$.

Taking into account this limitation, based on (4.41), the Lagrange function for the isoperimetric problem can be written as follows:

$$L = F\left(x, y, y'\right) + \lambda\left(x\right) \cdot \left(\psi'\left(x\right) - K\left(x, y, y'\right)\right). \tag{4.46}$$

Now we can reformulate the isoperimetric problem as follows: find the functions $y\left(x\right)$ and $\psi\left(x\right)$, which deliver the minimum of the functional

$$J^L = \int_a^b L\left(x, y, \psi, y', \psi'\right) dx\,. \tag{4.47}$$

According to the method of undetermined Lagrange multipliers, the Euler equations for the functional (4.47) will have the form

$$\begin{cases} L_y - \dfrac{d}{dx} L_{y'} = 0, \\ L_\psi - \dfrac{d}{dx} L_{\psi'} = 0. \end{cases} \tag{4.48}$$

Substituting the expression (4.46) into the system of equations (5.48), we obtain a system of equations

$$\begin{cases} F_y - \lambda\left(x\right) \cdot K_y - \dfrac{d}{dx}\left(F_{y'} - \lambda\left(x\right) \cdot K_{y'}\right) = 0, \\ -\dfrac{d}{dx}\lambda\left(x\right) = 0. \end{cases} \tag{4.49}$$

From the second equation of the system (4.49), we find that

$$-\lambda\left(x\right) = C_1\,. \tag{4.50}$$

Substituting the expression (4.50) into the first equation of the system (4.49), we obtain the equation

$$F_y + C_1 \cdot K_y - \frac{d}{dx}\left(F_{y'} + C_1 \cdot K_{y'}\right) = 0\,, \tag{4.51}$$

which in its general form is a nonlinear differential equation of the second order. The solution of this equation $y\left(x, C_1, C_2, C_3\right)$ will depend on three constants C_1, C_2, C_3, the presence of two of which (C_2, C_3) is determined by the second order of the differential equation, and the third (C_1) by substitution (4.50). To determine them, in addition to the equation with three unknowns C_1, C_2, C_3, which we obtain from condition (4.43) after substituting into the function $K\left(x, y, y'\right)$ of the general solution $y\left(x, C_1, C_2, C_3\right)$ and integrating the result of the substitution by x in the range from a to b, in the form

$$Q\left(C_1, C_2, C_3\right) = J_0^K\,, \tag{4.52}$$

it is necessary to synthesize two more equations with the same unknowns, which is easy to do by using boundary conditions that will have the form

$$\begin{cases} y\left(a, C_1, C_2, C_3\right) = y_a, \\ y\left(b, C_1, C_2, C_3\right) = y_b. \end{cases} \tag{4.53}$$

Solving the system of three equations (4.52), (4.53) with three unknowns C_1, C_2, C_3, we find their values

$$\left\{C_1^*, C_2^*, C_3^*\right\}, \tag{4.54}$$

which, after substituting them into the general solution $y\left(x, C_1, C_2, C_3\right)$ of equation (4.51), transform this solution into an extremal

$$y(\mathrm{x}, \mathrm{C}_1^*, C_2^*, C_3^*), \tag{4.55}$$

which delivers the extremum of the functional (3.1) in the presence of the constraint (4.43), and therefore is the solution of the isoperimetric problem of finding the extrema of the functionals under the conditions of the constraints also given by the functionals.

The minimum from the maximum of the functional (3.1) on the extremal (4.55) is distinguished by considering the Legendre conditions for the function *L*.

We remind that if

$$L_{y'y'} > 0, \tag{4.56}$$

then the extremal (4.55) delivers the minimum of the functional (3.1), and if

$$L_{y'y'} < 0 \tag{4.57}$$

– that's the maximum.

Remark. When considering the isoperimetric problem, we minimized the functional (3.1), using the functional (4.42) as a constraint (4.43). But, apparently, it was possible to do the opposite - to minimize the functional (4.42), and to use the functional (3.1) as a constraint. It is clear that the course of solving such a problem would not change, only its substantive interpretation would change, since with such a formulation of the problem, we would have to look for the curve of the smallest perimeter that limits the given area.

So it can be stated that the functions $F\left(x, y, y'\right)$ and $K\left(x, y, y'\right)$ n the Lagrange function are equal. This fact is reflected in mathematics in the form of the principle of reciprocity, according to which the form of extremals when solving an isoperimetric problem will not change depending on which of the two functionals is minimized and which sets the limit.

4.3 Direct method of finding extremals of functionals

We already know how to find extremals of functionals by solving the Euler, Euler–Lagrange, or Euler–Poisson equations.

But along with these methods, there is another class of methods for determining the extremals of functionals, with the help of which this procedure is carried out by direct minimization of the functional under the condition that the extremal is given by the partial sum $S_n\left(t\right)$ (1.102), in which the members of the orthonormal sequence are chosen as functions $\varphi_k\left(t\right)$ from among those considered in subsection 1.3.

The methods of this class are called direct methods for finding extrema of functionals or approximate methods.

One of the most popular methods in this class is the **Ritz method**, proposed at the beginning of the 20th century.

The essence of the Ritz method is as follows.

Let it be necessary to find an extremal $y\left(x\right)$, that minimizes the functional (3.1), which for convenience we will rewrite in the form

$$J(y)=\int_a^b F(x,y,y')\,dx \tag{4.58}$$

provided that on the borders of the region $[a,b]$ we have

$$\begin{cases} y(a)=y_a, \\ y(b)=y_b \end{cases} \tag{4.59}$$

and appearance restrictions are in place

$$g(x,y,y')=0 \tag{4.60}$$

or

$$\int_a^b K(x,y,y')\,dx = J_{1_0}. \tag{4.61}$$

We will look for an extremal $y(x)$ in the form

$$y(x)=\sum_{k=1}^{n} C_k \cdot \varphi_k(x), \tag{4.62}$$

where $\varphi_k(x)$ are known orthonormal polynomials from among those considered in the first section, for example, Laguerre, Legendre, or Chebyshov polynomials, which we deliberately chose and considered in detail in subsection 1.3, based on the convenience of their use for solving our problem.

By choosing the expression (4.62), we actually reduce the task of finding the extrema of the functional to the task of determining the coefficients $C_k, k=\overline{1,n}$. By substituting the expression (4.62) into the equation of the boundary conditions (4.59), at the first stage of solving the problem, we reduce by two the number of unknown coefficients $C_k, k=\overline{1,n}$ that we need to find in order to uniquely determine the extremal (4.62), which delivers the minimum of the functional (4.58).

We manage to do this at the first stage because with the help of two equations (4.59) two coefficients, for example C_1 and C_2, can be expressed in terms of other coefficients $C_k, k=\overline{3,n}$.

After that, the sought extremal (4.62) will already have the form

$$y=f_1(x,C_3,C_4,\ldots,C_n). \tag{4.63}$$

By substituting expression (4.63) into (4.60) or (4.61), we will obtain at the second stage an equation with which one more coefficient C_k can be removed, for example, C_3, by expressing it in terms of other coefficients $C_k, k=\overline{4,n}$.

After that, the sought extremal (4.63) will already have the form

$$y=f_2(x,C_4,\ldots,C_n). \tag{4.64}$$

Substituting the expression (4.64) into the functional (4.58), calculating the function $F(x,y,y')$ and taking the integral, we will obtain the function at the third stage

$$J(y)=f_3(C_4,\ldots,C_n), \tag{4.65}$$

which no longer contains a variable ***x*** and will be a function of coefficients only $C_k, k=\overline{4,n}$.

To find the optimal values of these coefficients, we use the standard method of finding the extremum of the function $(n-3)$ of variables $C_k, k=\overline{4,n}$ at the fourth stage.

For this, we equate the partial derivatives $\frac{\partial J(y)}{\partial C_k}, k=\overline{4,n}$ of the function (4.65) to zero

$$\frac{\partial f_3(C_4,\ldots,C_n)}{\partial C_k}=0,\ k=\overline{4,n}, \tag{4.66}$$

and solve the obtained system of algebraic equations (4.66) with respect to $C_k, k=\overline{4,n}$.

We substitute the found values of the coefficients $C_k, k=\overline{4,n}$ into the function (4.64). This will be the extremal $y(x)$ which, under the conditions of the restrictions (4.60) or (4.61) and the boundary conditions (4.59), delivers the minimum of the functional (4.58).

4.4 Python-Realization Programs Searching for Conditional Functional Extremes

Python language program for research on conditional extremum functioned

$$J_1=\int_a^b F_1(t,y,y')dt$$

in case where $a=0$, $b=1$, $F_1(t,y,y')=y^2+yy'+(y')^2$ and the extremal $y(t)$ begins at the point ($y(0)=0$, $y'(0)=1$) and satisfies the restriction $y=1$

(Program 14)

```
In [1]: import sympy
In [2]: from sympy import*
In [3]: from IPython.display import*
In [4]: init_printing(use_latex=True)
In [5]: t=symbols('t')
In [6]: λ =symbols ('λ')
In [7]: y=Function('y')(t)
In [8]: z=Function('z')(t)
In [9]: z=y.diff(t)
In [10]: u=Function('u')(t)
In [11]: u=y**2+y*z+z**2+λ*(y-1)
In [12]: de=Eq(u.diff(y)-u.diff(z,t),0)
In [13]: display(de)
```

$$\lambda+2y(t)-2\frac{d^2}{dt^2}y(t)=0$$

```
In [14]: des=dsolve(de)
In [15]: display(des)
```

$$y(t)=C_1e^{-t}+C_2e^{t}-\frac{\lambda}{2}$$

```
In [16]: eq1=des.rhs.subs(t,0);eq1
```

Out[16]: $C_1+C_2-\frac{\lambda}{2}$

```
In [17]: eq2=des.rhs.diff(t).subs(t,0);eq2
```

Out[17]: $-C_1+C_2$

```
In [18]: seq=solve([eq1,eq2-1],C1, C2);seq
```

Out[18]: $\left\{C_1:\frac{\lambda}{4}-\frac{1}{2}, C_2:\frac{\lambda}{4}+\frac{1}{2}\right\}$

```
In [19]: rez=des.rhs.subs([(C1,\
         seq[C1]),(C2,seq[C2])]);rez
```

Out[19]: $-\frac{\lambda}{2}+\left(\frac{\lambda}{4}-\frac{1}{2}\right)e^{-t}+\left(\frac{\lambda}{4}+\frac{1}{2}\right)e^{t}$

```
In [20]: eq3=rez.subs(t,1);eq3
```

Out[20]: $-\frac{\lambda}{2}+\left(\frac{\lambda}{4}-\frac{1}{2}\right)e^{-1}+\left(\frac{\lambda}{4}+\frac{1}{2}\right)e^{1}$

```
In [21]: expr1=-λ/2+(λ/4-1/2)/exp(1) \
         +(λ/4+1/2)* exp(1)
In [22]: expr2=1
In [23]: solveset(Eq(expr1,expr2),λ)
Out [23]:
```

$$\frac{-4.0(-1.0e-0.5+0.5e^2)}{(-1.0+1.0e)^2}$$

```
In [24]: import numpy as np
In [25]: λ=-4*(-np.exp(1)-0.5 \
         +0.5*(np.exp(1))**2)/(-1+\
         np.exp(1))**2;λ
Out [25]: -0,6423909789760494
In [26]: λ=np.array(-0.64239097897604)
In [27]: λ=λ.round(3);λ
Out[27]: -0.642
In [28]: d={}
In [29]: d["λ"]=-0.642
In [30]: d
Out [30]:{'λ': -0.642}
In [31]: y=y.subs({"λ":-0.642});y
```

Out [31]: $0.321 - 0.6605e^{-t} + 0.3395e^{t}$
In [32]: F=Lambda(t,y)
In [33]: display(Latex('$y(t)=' \
+str(latex(F(t)))+'$'))
$y(t) = 0.321 - 0.6605e^{-t} + 0.3395e^{t}$
In [34]: import matplotlib
In [35]: import matplotlib.pyplot as plt
In [36]: x=symbols('x')
In [37]: expr=0.321-0.6605*exp(-x) \
+0.3395*exp(x)
In [38]: f=lambdify(x, expr,'numpy')
In [39]: x=np.linspace(0,1,21)
In [40]: fig= plt.figure (facecolor='white')
In [41]: plt.plot(x,f(x),'-r',linewidth=3)

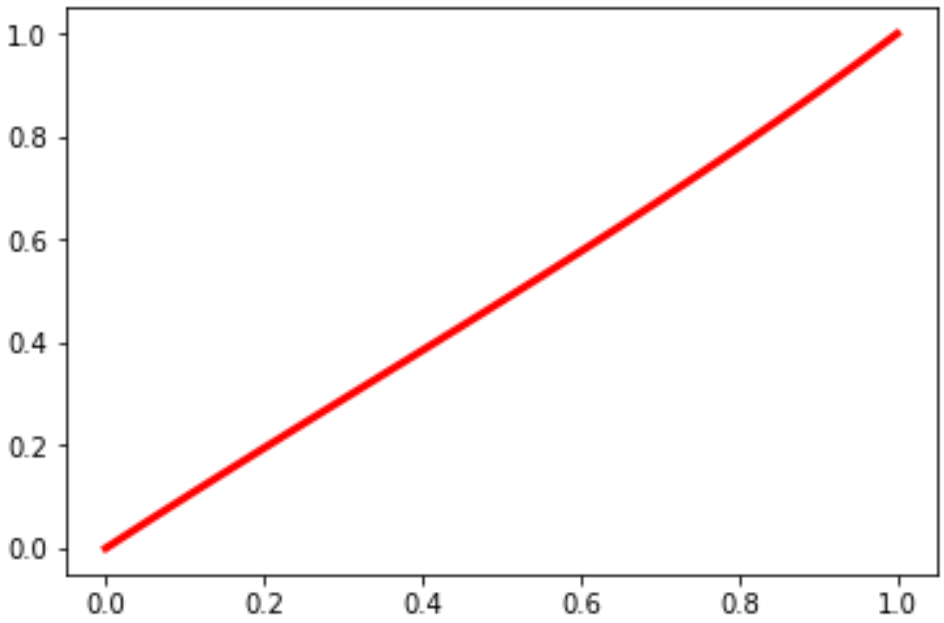


Figure 8. Functional Extremal Graph $J_1 = \int_a^b F_1(t, y, y')dt$ in case when $a = 0$, $b = 1$, $F_1(t, y, y') = y^2 + yy' + (y')^2$ and the extremal $y(t)$ begins in point ($y(0) = 0$, $y'(0) = 1$) and satisfies the restriction $y = 1$

End of program 14.

Python language program for research on conditional functional extremum

$$J_1 = \int_a^b F_1(t, y, y')dt$$

in case where a = 0, b = 1, $F_1(t, y, y') = y + y^2 + y' + (y')^2$, and the extremal y (t) begins at the point ($y(0) = 0$, $y'(0) = 1$) and satisfies the restriction $J_2 = \int_a^b (y + y')dt$, where $J_2 = 0.5$

(Program 15)

In [1]: import sympy
In [2]: from sympy import*
In [3]: from IPython.display import*
In [4]: init_printing(use_latex=True)
In [5]: t=symbols('t')
In [6]: y=Function('y')(t)
In [7]: z=Function('z')(t)
In [8]: z=y.diff(t)
In [9]: y1= Function('y1')(t)
In [10]: z1 = Function('z1')(t)
In [11]: z1=y1.diff(t)
In [12]: C3=symbols('C3')
In [13]: u=Function('u')(t)
In [14]: u=y+y**2+z+z**2+C3*(z1-y-z)
In [15]: de=Eq(u.diff(y)-u.diff(z,t),0)
In [16]: des=dsolve(de)
In [17]: display(des)
$$y(t) = C_1 e^{-t} + C_2 e^{t} + \frac{C_3}{2} - \frac{1}{2}$$
In [18]: eq1=des.rhs.subs(t,0);eq1

Out[18]: $C_1 + C_2 + \frac{C_3}{2} - \frac{1}{2}$

In [19]: eq2=des.rhs.diff(t).subs(t,0);eq2

Out[19]: $-C_1 + C_2$

In [20]: var('C1 C2')

Out[20]: (C1,C2)

In [21]: seq=solve([eq1,eq2-1],C1,C2);seq

Out[21]:

$\left\{C_1: -\frac{C_3}{4} - \frac{1}{4}, C_2: -\frac{C_3}{4} + \frac{3}{4}\right\}$

In [22]: rez=des.rhs.subs([(C1,seq[C1]),\
(C2,seq[C2])]);rez

Out[22]:

$\frac{C_3}{2} + \left(-\frac{C_3}{4} - \frac{1}{4}\right)e^{-t} + \left(-\frac{C_3}{4} + \frac{3}{4}\right)e^{t} - \frac{1}{2}$

In [23]: inte1=integrate(rez \
+rez.diff(t),(t,0,1));inte1

Out[23]: $C_3 + \frac{e(3-C_3)}{2} - 2$

In [24]: solveset(Eq(inte1,0.5),C3)

Out [24]: $\frac{2.5(-1.0+0.6e)}{-1.0+0.5\,e}$

In [25]: import numpy as np

In [26]: C3=2.5*(-1.0+0.6*(np.exp(1))/ \
(-1.0+0.5*np.exp(1)))

Out [26]: 4,39221119

In [27]: C3=np.array(4.39221119)

In [28]: C3=C3.round(3);C3

Out[28]: 4.392

In [29]: d={}

In [30]: d["C3"]=4.392

In [31]: d

Out [31]:{'C3': 4.392}

In [32]: y=rez.subs({'C3':4.392});y

Out [32]: $1.696 - 1.348e^{-t} - 0.348e^{t}$

In [33]: F=Lambda(t,y)

In [34]: display(Latex('$y(t)='+\
str(latex(F(t)))+'$'))

$y(t) = 1.696 - 1.348e^{-t} - 0.348e^{t}$

In [35]: import matplotlib

In [36]: import matplotlib.pyplot as plt

In [37]: x=symbols('x')

In [38]: expr=1.696-1.348*exp(-x) \
-0.348*exp(x)

In [39]: f=lambdify(x, expr,"numpy")

In [40]: x=np.linspace(0,1,21)

In [41]: fig= plt.figure (facecolor='white')

In [42]: plt.plot(x,f(x),'-r',linewidth=3)

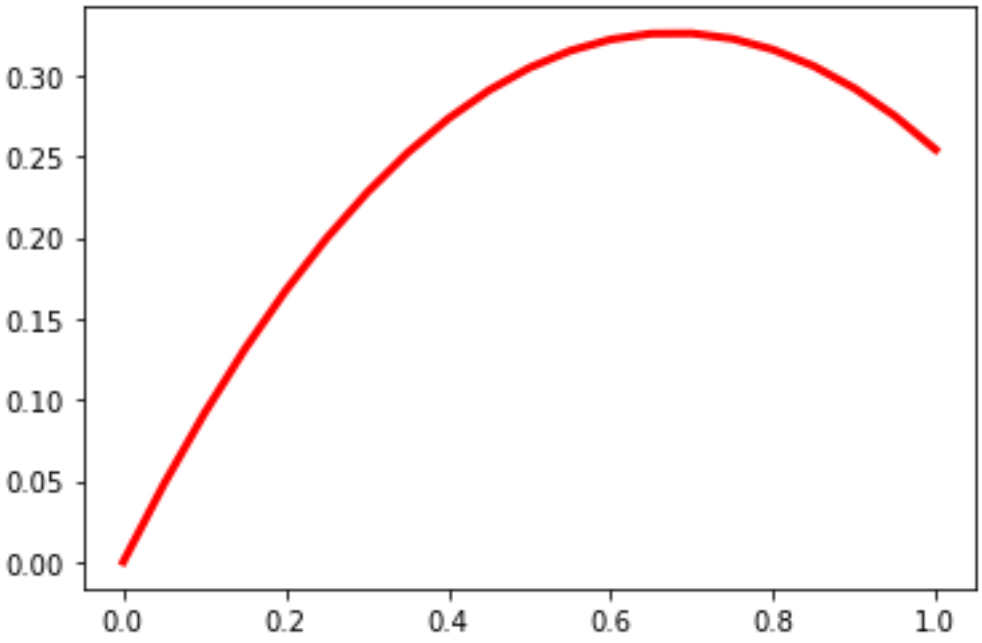


Figure 9. Functional Extremal Graph $J_1 = \int_a^b F_1(t, y, y')dt$, when $a = 0$, $b = 1$, $F_1(t, y, y') = y + y^2 + y' + (y')^2$, and the extremal $y(t)$ begins at the point $(y(0) = 0, y'(0) = 1)$ and satisfies the restriction $J_2 = \int_a^b (y + y')dt$, where $J_2 = 0.5$

End of program 15.

Python language program for research on conditional functional extremum

$$J_1 = \int_a^b F_1(t, y, y')dt$$

the Ritz using the first 5 orthonormized polynomials of the Legendre, when a = -1, b = 1, $F_1(t, y, y') = y + y^2 + y' + (y')^2$, and the extremal $y(t)$ begins at the point ($y(0) = 0$, $y'(0) = 1$) and satisfies the restriction $J_2 = \int_a^b F_2(t, y, y')dt$, where $F_2(t, y, y') = y + y'$, $J_2 = 1$

(Program 16)

```
In [1]: import sympy
In [2]: from sympy import*
In [3]: from IPython.display import*
In [4]: init_printing(use_latex=True)
In [5]: t=symbols ('t')
In [6]: P0=Function('P0')(t)
In [7]: P1=Function('P1')(t)
In [8]: P2=Function('P2')(t)
In [9]: P3=Function('P3')(t)
In [10]: P4=Function('P4')(t)
In [11]: P0=1
In [12]: P1=t
In [13]: P2=(3*t**2-1)/2
In [14]: P3=(5*t**3-3*t)/2
In [15]: P4=(35*t**4-30*t**2+3)/8
In [16]: var('C0 C1 C2 C3 C4')
Out[16]: (C0,C1,C2,C3,C4)
In [17]: y=Function('y') (t)
In [18]: z=Function('z') (t)
In [19]: y=C0*P0+C1*P1+C2*P2+C3*P3+C4*P4
```

In [20]: display(y)

$$C_0 + C_1 t + C_2\left(\frac{3t^2}{2} - \frac{1}{2}\right) + C_3\left(\frac{5t^3}{2} - \frac{3t}{2}\right) + C_4\left(\frac{35t^4}{8} - \frac{15t^2}{4} + \frac{3}{8}\right)$$

In [21]: z=y.diff(t)

In [22]: display(z)

$$C_1 + 3C_2 t + C_3\left(\frac{15t^2}{2} - \frac{3}{2}\right) + C_4\left(\frac{35t^3}{2} - \frac{15t}{2}\right)$$

In [23]: eq1=y.subs(t,0);eq1

Out[23]: $C_0 - \frac{C_2}{2} + 3C_4/8$

In [24]: eq2=z.subs(t,0);eq2

Out[24]: $C_1 - 3C_3/2$

In [25]: seq=solve ([eq1,eq2-1],C0,C1);seq

Out[25]: $\{C_0: \frac{C_2}{2} - \frac{3C_4}{8}, C_1: \frac{3C_3}{2} + 1\}$

```
In [26]: rez=y.subs([(C0,seq[C0]),(C1,seq[C1])])
In [27]: rez1=z.subs([(C0,seq[C0]),(C1,seq[C1])])
In [28]: u=Function('u')(t)
In [29]: u=rez+rez1
In [30]: eq3=integrate(u,(t,-1,1));eq3
```

Out[30]: $C_2 + 5C_3 - \frac{3C_4}{4} + 2$

```
In [31]: seq1=solve([eq3-1],C2);seq1
```

Out[31]: $\{C2: -5C_3 + \frac{3C_4}{4} - 1\}$

```
In [32]: rez2=rez.subs([(C2,seq1[C2])])
In [33]: rez3=rez1.subs([(C2,seq1[C2])])
In [34]: u1=Function ('u1')(t)
In [35]: u1=rez2+rez2**2+rez3+rez3**2
In [36]: w=symbols('w')
In [37]: w=integrate(u1,(t,-1,1));w
```

Out[37]:

$$\frac{2755C_3^2}{14} + 81C_3 - 108C_3C_4 + \frac{1477C_4^2}{45} - \frac{108C_4}{5} + \frac{317}{30}$$

In [38]: eq4=diff(w,C3);eq4

Out[38]: $\frac{2755C_3}{7} - 108C_4 + 81$

In [39]: eq5=diff(w,C4);eq5

Out[39]: $-108C_3 + \frac{2954C_4}{45} - \frac{108}{5}$

In [40]: seq2=solve([eq4,eq5],C3,C4);seq2

Out[40]: $\{C_3: -\frac{67149}{318865}, C_4: -\frac{7776}{446411}\}$

```
In [41]: rez4=rez2.subs([(C3,seq2[C3]),\
                          (C4,seq2[C4])]);rez4
```

Out[41]: $-\frac{4860t^4}{63773} - \frac{67149t^3}{127546} + \frac{7980t^2}{63773} + t$

In [42]: F=Lambda(t,rez4)

In [43]: display(Latex('$rez4(t)=' \
+str(latex(F(t)))+'$'))

$$rez4(t) = -\frac{4860t^4}{63773} - \frac{67149t^3}{127546} + \frac{7980t^2}{63773} + t$$

In [44]: expr=-4860*t**4/63773 \
-67149*t**3/127546+7980*t**2/63773+t

In [45]: expr1=expr.evalf(3);expr1

$$-0.0762t^4 - 0.526t^3 + 0.125t^2 + t$$

In [46]: from sympy.plotting import plot

In [47]: exstremal=plot(expr1,(t,-1,1))

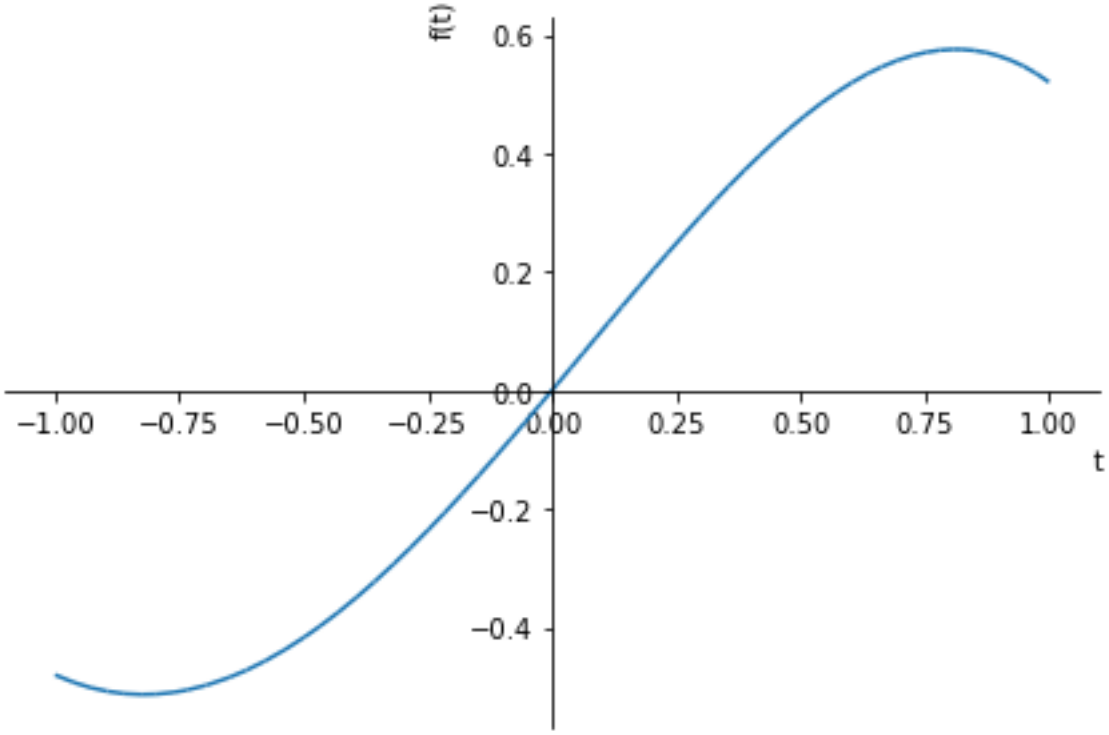


Figure 10. Functional Extremal Graph $J_1 = \int_a^b F_1(t, y, y')dt$ in case when a = -1, b = 1, $F_1(t, y, y') = y + y^2 + y' + (y')^2$, and the extremal y (t) begins at the point (y (0) = 0, $y'(0)$ = 1) and satisfies the restriction $J_2 = \int_a^b F_2(t, y, y')dt$, where $F_2(t, y, y') = y + y'$, $J_2 = 1$

End of program 16.

4.5 Self -Testing Task

1. What is the difference between the search for conditional extremum and the unconditional?
2. Formulate the lagrani problem and the algorithm for solving it using uncertain factors.
3. Prove that the task of minimizing the functional in the presence of restrictions can be transformed into the simplest problem of variational calculus in relation to the functional, in which these restrictions are introduced using indefinite Lagrange multipliers.
4. Write down the algorithm of the Lagrange for the functional, which depends on several functions and their derivatives, in the conditions of restrictions determined by one equation, as well as the system of equations.

5. How is the isopermetric problem of optimization and why is it called isopermetric?
6. Construct an algorithm for solving the isoperimetric optimization problem.
7. How to distinguish the minimum functional from the maximum in the problem of conditional extremum?
8. What is the essence of the principle of reciprocity in isoperimetric problem?
9. What is the essence of direct methods of finding functional extremums? The essence of the Ritz method.
10. In what form is the extremal of functional in the Ritz method found? What orthonormal sequences are used in this method?
11. Expand the essence of the stages of solving the problem of search for an extremum functional by the Ritz method.
12. Show the command that formed the Euler- Lagrange equation
13. Show the command of the application by which Euler- Lagrange equation is summoned to the monitor
14. Why do we need the seq=solve([eq1,eq2-1],C1,C2)
15. What is the purpose of implementing rez = des.rhs.subs ([(C1, Seq [C1]), (C2, Seq [C2]))?
16. What does the command f=lambdify(x, expr3,"numpy")?
17. How to determine what is achieved on extremities - a minimum or a maximum of functionality?
18. What will those few teams look like to add to the program to determine what is achieved – maximum or minimum?
19. Where does the algorithm for determining Fourier coefficients in the program that implements the Ritz method?
20. Show the program command that implements the determination of coefficients C0, C1?
21. How is the C2 coefficient determined and what teams in the program are searching for its value?
22. How are the C3, C4 coefficients determined and what teams in the program are searching for their values?
23. What is reached by the rez4=rez2.subs([(C3,seq2[C3]),(C4,seq2[C4])])?
24. What is the exstremal=plot(expr1,(t,-1,1))?

Chapter 5. OPERATORS AND THEIR APPLIED ASPECTS

5.1 Operator, its linearity and norm

In the third and fourth chapters of this tutorial, we studied the properties of functionals that specify the laws according to which each element of the set of functions is matched by some element of the set of numbers.

And now let's move on to the study of operators that set laws according to which each element from a set of functions is matched with some element from another or the same set of functions.

And let's begin the study of this section of functional analysis by establishing some of the most commonly used characteristics and properties of operators.

The operator ***A***, which transforms a set ***X*** into a set ***Y***, is called linear if, firstly, it is additive, i.e. it satisfies the relation

$$A \cdot (x_1 + x_2) = A \cdot x_1 + A \cdot x_2, \ \forall\, x_1, x_2 \in X, \tag{5.1}$$

and secondly, is continuous, i.e., the relation holds for it

$$A \cdot x_n \to A \cdot x_0, \tag{5.2}$$

if $x_n \to x_0, \ \forall\, x_0, x_n \in X$.

The following is true for a linear operator A:

1) $$A \cdot 0 = 0, \ 0 \in X; \tag{5.3}$$

2) $$A(-x) = -A \cdot x, \ \forall\, x \in X; \tag{5.4}$$

3) $$A \cdot (t \cdot x) = t \cdot A \cdot x, \ \forall\, x \in X, t - \text{ is a scalar.} \tag{5.5}$$

If the relation (5.5) holds for the operator, then it is called homogeneous.

One of the most important properties of the operator $S[0,1] \subset X$, is its boundedness on the unit ball , for which it is performed

$$\| x \| \le 1, \ x \in S. \tag{5.6}$$

The boundedness of the linear operator ***A*** is set as follows.

Let there exist such a constant ***K*** that

$$\| A \cdot x \| \le K \cdot \| x \|, \ \forall\, x \in X. \tag{5.7}$$

Number

$$K_0 = \sup_{\| x \| \le 1} \| A \cdot x \| \tag{5.8}$$

is called the norm of the operator A **and denote**

$$K_0 = \| A \|. \tag{5.9}$$

We remind that the symbol " sup " means the "upper limit" of the expression that is on the right side of this symbol. The set of elements on which the operator acts is indicated under the symbol.

From the triangle inequality for the norm, it follows that for $\forall\, x \in S$ is true

$$\| A \cdot x \| \le \| A \| \cdot \| x \|. \tag{5.10}$$

And therefore it can be claimed that the set of linear operators A that transform a set X into a set Y, which is symbolically denoted as

$$A : (X \to Y) \tag{5.11}$$

or

$$A \in (X \to Y), \tag{5.12}$$

is a linear normalized space $(X \to Y)$.

It follows from relations (5.7) (5.10) that for the norm of the operator

$$\| A \| = \sup_{x \neq 0} \frac{\| A \cdot x \|}{\| x \|}. \tag{5.13}$$

A very important theorem: if an additive and homogeneous operator is bounded, then it is continuous and therefore linear.

Proof.

Let $x_n \to x_0$, $\{x_n\} \subset X$, $x_0 \in X$.

Then, due to additivity and boundedness,

$$\| A \cdot x_n - A \cdot x_0 \| = \| A \cdot (x_n - x_0) \| \le K \cdot \| x_n - x_0 \|. \tag{5.14}$$

But from the fact that $x_n \to x_0$, it follows that

$$\| x_n - x_0 \| = 0. \tag{5.15}$$

Substituting the expression (5.15) into (5.14), we obtain that

$$\| A \cdot x_n - A \cdot x_0 \| = 0. \tag{5.16}$$

The relation (5.16) is the property (5.2) written through the norm, which determines the continuity of the operator.

But, if the operator ***A*** is continuous and additive, then it is linear by definition.

For a Hilbert space, the concept of a continuous operator can also be introduced using a scalar product. That is, in the *H*-space it is such a linear operator A from the set $(X \to Y)$, for which the fact that $x_n \to x_0$, $y_n \to y_0$, implies that

$$\langle A \cdot x_n, y_n \rangle \to \langle A \cdot x_0, y_0 \rangle. \tag{5.17}$$

Due to the presence of a scalar product in H-space, a special class of linear operators is distinguished, which are called symmetrizing operators. In this case, **the operator $A \in (H \to H)$ is called symmetrizing if for** $\forall\, x, y \in H$

$$\langle A \cdot x, y \rangle = \langle x, A \cdot y \rangle \tag{5.18}$$

Consider $\langle Ax, y \rangle$ under the condition that x is a variable element and a y is a fixed element in the *H*-space, that is, we assume that

$$\{x\} \subset H, \; y \in H. \tag{5.19}$$

Under condition (5.19), the scalar product $\langle Ax, y \rangle$ is a linear functional of x, that is, we have

$$F_y(x) = \langle A \cdot x, y \rangle . \tag{5.20}$$

By changing the elements of $y \in H$, we will obtain different values of the functional (5.20), which can be formally denoted by the symbol y^* -

$$y^* = F_y(x), \tag{5.21}$$

where $y^* \in H$.

But, on the other hand, you can formally write that

$$y^* = A^* y, \tag{5.22}$$

where A^* **is a linear operator that transforms elements** $y \in H$ **into elements of** $y^* \in H$. **This operator** A^* **is called conjugate to the operato**r A.

Since for the elements x and y^* it is also possible to define the scalar product as $\langle x, y^* \rangle$, the ratio will also be valid

$$\langle x, y^* \rangle = \langle x, A^* y \rangle, \tag{5.23}$$

which we obtain by substituting into the expression of the relation (5.22).

But, if A is a symmetrizing operator, it follows from the comparison of expressions (5.18) and (5.23) that

$$A = A^*, \tag{5.24}$$

i.e., the symmetrizing operator A coincides with its conjugate A^*. In this regard, **symmetrizing operators are also called self-adjoint.**

Operator A**, for which**

$$A \cdot A^* = A^* \cdot A = I, \tag{5.25}$$

where I is a single operator, i.e. such that

$$I \cdot x = x, \tag{5.26}$$

called a unitary.

5.2 The inverse operator and the resolvent and spectrum of the operator

Let the operator $A \in (X \to Y)$ be given.

The operator $A^{-1} \in (Y \to X)$**,** which satisfies the equation

$$\begin{cases} A^{-1} \cdot (A \cdot x) = x, \ \forall x \in X, \\ A \cdot (A^{-1} \cdot y) = y, \ \forall y \in Y. \end{cases} \tag{5.27}$$

is called the inverse of the operator . It is clear that equations (5.27) can be satisfied only under the condition that

$$A^{-1} \cdot A = A \cdot A^{-1} = I . \tag{5.28}$$

Note that by the product of any operators we mean their consecutive application to the element standing to the right of them.

After introducing the inverse operator A^{-1}, it is clear that to solve the equation

$$A \cdot x = y \tag{5.29}$$

it is necessary to find A^{-1}, since

$$\begin{gathered} A^{-1}\left(A \cdot x\right) = A^{-1} y \\ \Downarrow \\ \left(A^{-1} A\right) \cdot x = A^{-1} y \\ \Downarrow \\ I \cdot x = A^{-1} y \\ \Downarrow \\ x = A^{-1} y. \end{gathered} \tag{5.30}$$

The question arises: "How to solve the equation:

$$A \cdot x - \lambda \cdot x = y, \tag{5.31}$$

$$A \cdot x - \lambda \cdot x = 0, \tag{5.32}$$

where $y \in Y$, $A \in \left(X \to Y\right)$, $x \in X$ and is λ – a scalar?"

Let's rewrite (5.31), (5.32) as follows:

$$\left(A - \lambda \cdot I\right) \cdot x = y, \tag{5.33}$$

$$\left(A - \lambda \cdot I\right) \cdot x = 0. \tag{5.34}$$

Let there exist an operator R_λ, inverse to $\left(A - \lambda \cdot I\right)$, i.e

$$R_\lambda = \left(A - \lambda \cdot I\right)^{-1}. \tag{5.35}$$

In this case, multiplying equation (5.33) on the left by R_λ, we get

$$x = \left(A - \lambda \cdot I\right)^{-1} \cdot y. \tag{5.36}$$

Expression (5.36) will be the solution of equation (5.33).

The inverse operator R_λ, which is defined by the relation (5.35), is called the resolvent of the operator or the solving operator for equation (5.33).

It is clear that the solution of equation (5.34) will have the form

$$x = \left(A - \lambda \cdot I\right)^{-1} \cdot 0 = 0. \tag{5.37}$$

Those values of the parameter λ, that allow having a resolvent R_λ, are called regular values of the operator A. All other values of the parameter λ, which are not regular, make up the spectrum of the operator A.

But, as research has shown, **there are such values of λ, for which the homogeneous equation (5.32) has a solution other than zero**, that is, (5.37) is not fulfilled for them. **Such values λ, are called characteristic numbers or eigenvalues of the operator A. It is clear that they are points of the spectrum of this operator.**

But it should be noted that the spectrum of the operator A may include values that are not its characteristic numbers λ, that is, the power of the spectrum of the operator is greater than the power of the set of its characteristic numbers.

The theorem is useful for finding solutions to many inhomogeneous operator equations of the class (5.31): let $A \in (X \to X)$, where is $X \in$ B-space. Let there also exist some parameter μ, which satisfies the condition

$$|\mu| < \frac{1}{\|A\|}. \tag{5.38}$$

Under these conditions, the operator $(I - \mu A)$ has an inverse operator R_μ and at the same time

$$R_\mu = (I - \mu \cdot A)^{-1} = I + \mu \cdot A + \mu^2 \cdot A^2 + \mu^3 \cdot A^3 + \cdots. \tag{5.39}$$

It is clear that R_μ is also a resolvent of the operator A, normalized to λ.

(5.39) is proved by decomposing R_μ into a power series around the point $\mu = 0$.

Let us give some properties of the resolvent.

1. For the set D of regular points of the operator A

$$R_\lambda - R_\mu = (\lambda - \mu) \cdot R_\lambda \cdot R_\mu, \ \forall \lambda, \mu \in D. \tag{5.40}$$

From equation (5.40) by means of the limit transition $\lambda \to \mu$, we obtain

$$\frac{d^k R_\lambda}{d\lambda^k} = k! \cdot R_\lambda^{k+1}. \tag{5.41}$$

2. If the operator A is bounded, then its entire spectrum lies in a circle

$$\|\lambda\| \le \|A\|, \tag{5.42}$$

and outside this circle, i.e. at

$$\|\lambda\| > \|A\| \tag{5.43}$$

the resolvent R_λ can be decomposed into a series that converges according to the norm of the operator

$$R_\lambda = -\frac{1}{\lambda} \cdot \left(I + \frac{1}{\lambda} \cdot A + \frac{1}{\lambda^2} \cdot A^2 + \cdots \right). \tag{5.44}$$

3. **The radius of the smallest circle r_A with the center at the origin of coordinates, which contains the entire spectrum of the operator A, is called the spectral radius** and can be determined by the Gelfand formula

$$r_A = \lim_{n \to \infty} \sqrt[n]{\|A^n\|}. \tag{5.45}$$

It follows from the expression (5.45) that

$$r_A \le \|A\|. \tag{5.46}$$

4. Let λ be a common regular point of two closed linear operators A and B. If $D_B \supset D_A$, then

$$R_\lambda^B - R_\lambda^A = R_\lambda^B \cdot (A - B) \cdot R_\lambda^A. \tag{5.47}$$

Let us return to equation (5.31) or, which is the same thing, to equation (5.33). Two statements can be made about it.

1. **Equation (5.31) is said to have a unique solution if the corresponding homogeneous equation**

$$A \cdot x = 0 \tag{5.48}$$

has only zero solution.

2. **Equation (5.31) is said to have a correct solution if for** $\forall x \in X$ **the relation**

$$\| x \| \le K \cdot \| A \cdot x \|, \; A \in (X \to X). \tag{5.49}$$

is true.

It follows from the first statement that there is a left inverse operator A^{-1} for the operator $A \in D_A$.

It follows from the second statement that the operator A^{-1} is bounded, and therefore the solution of equation (5.31) depends continuously on the right-hand side.

Let us introduce the notion of a group of operators.

A group $u(t)$ **of operators** A **is understood to mean such a set of them, which have the following properties**:

1) $$u(0) = I; \tag{5.50}$$

2) $$u(t+\tau) = u(t) \cdot u(\tau), \quad -\infty < t, \tau < \infty; \tag{5.51}$$

3) $$\frac{du}{dt} = A \cdot u. \tag{5.52}$$

It is clear that these three properties have only those operators A, that are bounded and satisfy the relation

$$u(t) = e^{t \cdot A}. \tag{5.53}$$

It follows from the expressions (5.50), (5.52) and (5.53) that the operator A can be defined as a derivative of the group $u(t)$ at $t = 0$.

Therefore, the **operator** A **is called the generic operator for the group** $u(t)$ **or, which is the same thing, the infinitesimal operator**.

Expanding the exponent in expression (5.53) into a power series, we obtain

$$u(t) = 1 + t \cdot A + \frac{t^2}{2!} \cdot A^2 + \frac{t^3}{3!} \cdot A^3 + \cdots, \tag{5.54}$$

that is, the group $u(t)$ can be specified not only through the exponent (5.53), but also in the form of a power series (5.54).

5.3 Method of compressed images

This method is one of the key for many applications of functional analysis in applied problems of the IT sphere, but before explaining its essence, **we give a theorem about the only common point** of a sequence of nested closed spheres and is formulated as follows: **a sequence of nested closed spheres** $K_1 \supset K_2 \supset K_3 \supset ...,$ **which are subsets of the complete metric space *R*, such that**

$$K_n = \{x_n, \rho(x, x_n) \le r_n\}, \quad n = 1,2,3,..., \tag{5.55}$$

and the radii r_n **of which approach zero at** $n \to \infty$ **, have a single common point** x_0.

The proof of this theorem is based on the fact that for the sequence (5.55), no matter how small we choose the number $\varepsilon > 0$, it is always possible to construct a closed sphere with a radius $r_k < \varepsilon,$ for which the expression

$$\rho(x, x_k) \le r_k, \tag{5.56}$$

from which, due to the completeness of space R, it follows that no matter which two points of this space we take, it will always be possible to construct a sphere with a radius smaller than the distance between these points, and therefore the sequence $\{x_n\}$ is fundamental, for which it is true

$$\lim_{n \to \infty} x_n = x_0 . \tag{5.57}$$

And it follows from the expressions (5.55), (5.56), (5.57) that with the insertion of a sphere into a sphere, the radius of each of the following ones is smaller and smaller, we will arrive at a single point x_0 of space R, common to all these spheres, in which this sequence the ball is formed.

And now let's imagine that we form a sequence of such "balls" $[a_i, b_i], i = 0,1,2,...,n$ on a segment $[a,b]$ of the number axis, provided that

$$\begin{aligned} &b = b_0 > b_1 > b_2 > ... > b_n > ..., \\ &a = a_0 < a_1 < a_2 < ... < a_n < ... \end{aligned} \tag{5.58}$$

According to the above-formulated and proven theorem, this sequence, contracting, will lead to a single common point on the number axis x. So, if we determine the law $f(x)$, according to which, when compressing a segment $[a,b]$ of the numerical axis x, each point $x_i \ , i = 0,1,2,...,n$ of the previous segment $[a_i, b_i], i = 0,1,2,...,n$ will correspond to a point $x_i^* \ , i = 1,2,...,n$ of the next segment nested in the previous one, then this single common point can be found by solving the equation

$$x = f(x) . \tag{5.59}$$

And then we will write the definition: the mapping given in the metric space R by the operator A is called compressed if for $x_1 \in R$ and $x_2 \in R$ the inequality holds

$$\rho(Ax_1, Ax_2) \le \theta \rho(x_1, x_2), \tag{5.60}$$

where

$$0 < \theta < 1 \tag{5.61}$$

Next, it is appropriate to consider **Banach's theorem**, which, in fact, is an extension to operators of the above-proved theorem about the only common point of a sequence of nested spheres given in a complete metric space, and which is formulated as follows: **the compressed mapping by the operator A of the complete metric space R in itself has one and only one fixed point**. To prove this theorem, we assume that the operator A acts on a point $x_i \in R, \ i = 0,1,2,...,n$ such that

$$x_1 = Ax_0, \quad x_2 = Ax_1, ..., x_n = Ax_{n-1}, ... \tag{5.62}$$

Let's also assume that

$$\begin{aligned} &\rho(x_1, x_2) = \theta \rho(x_0, x_1), \\ &\rho(x_2, x_3) = \theta \rho(x_1, x_2) = \theta^2 \rho(x_0, x_1), \\ &..... \\ &\rho(x_{n-1}, x_n) = \theta^{n-1} \rho(x_0, x_1), \end{aligned} \tag{5.63}$$

where θ is given by the expression (5.61).

Let's ask $q > n$. Then, according to expressions (5.60) and (5.62), we have

$$\begin{aligned}\rho(x_n, x_q) &= \rho(Ax_{n-1}, Ax_{q-1}) \le \theta\rho(x_{n-1}, x_{q-1}) = \theta\rho(Ax_{n-2}, Ax_{q-2}) \le \theta^2\rho(x_{n-2}, x_{q-2}) = \\ &= \theta^2\rho(Ax_{n-3}, Ax_{q-3}) \le \theta^3\rho(x_{n-3}, x_{q-3}) = \ldots = \theta^{n-1}\rho(Ax_0, Ax_{q-n}) \le \theta^n\rho(x_0, x_{q-n})\end{aligned} \tag{5.64}$$

Based on the definition of the metric as the distance between the elements of the metric space $\rho(x_0, x_{q-n})$, we can write that

$$\rho(x_0, x_{q-n}) = \rho(x_0, x_1) + \rho(x_1, x_2) + \ldots + \rho(x_{q-(n-1)}, x_{q-n}) \tag{5.65}$$

Let's leave only the first and last terms in inequality (5.64), which will only strengthen it. As a result of such a step, we get that

$$\rho(x_n, x_q) \le \theta^n\rho(x_0, x_{q-n}). \tag{5.66}$$

Let

$$q, n \to \infty. \tag{5.67}$$

By substituting expressions (5.63) into (5.65), and the result of this substitution into expression (5.66), taking into account the condition (5.66), we obtain

$$\rho(x_n, x_q) \le \theta^n\rho(x_0, x_1)(1 + \theta + \theta^2 + \ldots + \theta^{q-n-1} + \ldots) = \frac{\theta^n}{1-\theta}\rho(x_0, x_1) \tag{5.68}$$

Note that in obtaining the expression (5.68), we used the formula for the sum of the members of
an infinitely decreasing geometric progression with the denominator θ.

It is easy to see that with growth ***n*** the multiplier θ^n on the right-hand side of inequality (5.68) will approach zero due to the fulfillment of condition (5.61), which, in turn, indicates that with growth n, q the left-hand side of this inequality will also approach zero, i.e. that the sequence $\{x_n\}$ is fundamental and therefore for the general term

$$x_n = Ax_{n-1} \tag{5.69}$$

of the sequence (5.62) when $n \to \infty$ the equality holds

$$x = Ax, \tag{5.70}$$

which, according to expression (5.59) of the theorem on the existence of a fixed point, confirms that part of Banach's theorem, which states that the compressed mapping carried out by the operator A in the complete metric space R has a fixed point *x*. And it is quite obvious that this fixed point is a solution of the operator equation (5.70).

It remains to prove that part of Banach's theorem, which states that this fixed point *x* is the only one for the operator A.

This proof can be carried out from the opposite, that is, suppose that the mapping carried out by the operator A in the metric space R has two fixed points *x* and x^*. Then equality (5.70) will hold for each of them, i.e. then we will have for the point x^* as well

$$x^* = Ax^*. \tag{5.71}$$

But then, according to the definition of the concept of compressed mapping and expressions (5.60), (5.70), (5.71), we can write that

$$\rho(x, x^*) = \rho(Ax, Ax^*) \le \theta\rho(x, x^*). \tag{5.72}$$

Expression (5.72) can be fulfilled only under the condition that

$$\theta = 1, \tag{5.73}$$

i.e., further compression by this operator cannot be performed. And this, in turn, proves that

$$x = x^* \tag{5.74}$$

i.e., that these points occupy the same place on the axis - this is the confirmation of that part of Banach's theorem, which states that the fixed point is the only one for the operator A.

As we will show in the next section dedicated to applied aspects of functional analysis, algorithms for solving operator equations, which are both algebraic and integral equations, are easily formed using the method of compressed mappings and the Banach theorem proved above

5.4 Application of the compressed mapping method to prove the existence of a single solution of differential and integral equations

Consider the space C of continuous functions $y = y(x)$ defined on the segment $x \in [a,b]$ of the number axis, with values defined on the segment $y \in [M,N]$ of the number axis, and the metric

$$\rho(y, y_1) = \sup_{x \in [a,b]} |y - y_1|, \tag{5.75}$$

where y, y_1 – are the points of this space.

Let the differential equation be given in the space C

$$y' = f(x,y) \tag{5.76}$$

with the initial condition

$$y(x_0) = y_0 . \tag{5.77}$$

Let's impose a condition on the function $f(x,y)$ so that it satisfies the Lipshitz condition, that is, so that

$$|f(x,y_0) - f(x,y_1)| \le L|y_0 - y_1|, \tag{5.78}$$

where $(x,y_0),(x,y_1)$ are the points of the rectangular area G, bounded by segments $x \in [a,b]$, $y \in [M,N]$, on the plane (x,y), and L is a constant, the numerical value of which will be determined a little later.

Let's integrate the differential equation (5.76) in the range from x_0 to x

$$\int_{x_0}^{x} \frac{dy}{dx} dx = \int_{x_0}^{x} f(x,y) dx . \tag{5.79}$$

We will have

$$y(x) - y(x_0) = \int_{x_0}^{x} f(x,y) dx , \tag{5.80}$$

or

$$y = y_0 + \int_{x_0}^{x} f(x,y) dx . \tag{5.81}$$

It is obvious that the integral equation (5.81) can be rewritten in operator form as follows

$$y = Ay\,. \tag{5.82}$$

If we prove that the operator A carries out a compressed mapping of the space of functions C into itself, then we thereby prove that this mapping defines a single fixed point in this space, which in the functional interpretation is the solution of the integral equation (5.81), which, in in turn, is another form of writing the differential equation (5.76), and therefore this proof will be simultaneously a proof that in the given space both the integral equation (5.81) and the differential equation (5.76) have a solution which, in addition is the only one.

Applying the compressed mapping method, we have the right to write that

$$\begin{aligned}\rho(y_1,y_2)=\rho(Ay_0,Ay_1)=\sup_{x\in[a,b]}|Ay_0-Ay_1|\le\sup_{x\in[a,b]}\int|f(x,y_0)-f(x,y_1)\|dx|\le\\ \le\sup_{x\in[a,b]}\int_{x_0}^{x}L|y_0-y_1\|dx|=L|x-x_0|\sup_{x\in[a,b]}|y_0-y_1|=\theta\,\rho(y_0,y_1),\end{aligned} \tag{5.83}$$

where

$$\theta = L|x - x_0| \tag{5.84}$$

According to the ideology of the method of compressed mappings, in order for the expression (5.83) to realize this ideology, it is necessary that the inequality

$$\theta < 1\,. \tag{5.85}$$

Comparing expressions (5.84) and (5.85), we see that expression (5.83) will implement the ideology of the compressed mapping method and will direct the sequence

$$y_1 = Ay_0,\ y_2 = Ay_1,\ y_3 = Ay_2,...,y_n = Ay_{n-1},... \tag{5.86}$$

to a single fixed point of the domain G, which will be the projection of the function $y = y(x)$ onto the metric functional space C and the solution of the integral equation (5.81) in the case when the constant L satisfies the inequality

$$|x - x_0| < \frac{1}{L}\,. \tag{5.87}$$

Integral equation

$$\varphi(x) = f(x) + \lambda\int_a^b K(x,y)\varphi(y)dy, \tag{5.88}$$

which connects continuous functions $\varphi(x)$ with the norm in the space of continuous functions C

$$\|\varphi(x)\| = \sup_{x\in[a,b]}|\varphi(x)| \tag{5.89}$$

with functions $K(x,y)$ continuous at the points of the plane (x,y), bounded by the boundaries of the rectangle $[a \le x, y \le b]$, with norm

$$\|K(x,y)\| = \sup_{[a\le x,y\le b]}|K(x,y)|\,, \tag{5.90}$$

is called the Fredholm equation of the 2nd kind in honor of the mathematician who constructed it and studied its properties. It is also called the inhomogeneous Fredholm equation.

If the upper bound of the integral in equation (5.88) is set equal to $\boldsymbol{x}$, then we obtain the integral equation

$$\varphi(x) = f(x) + \lambda \int_a^x K(x,y)\varphi(y)dy, \tag{5.91}$$

which connects in the same space of continuous functions C the continuous functions $\varphi(x)$ with the norm (5.89) with the same continuous functions $K(x,y)$ with the norm (5.90) and which is called the Volterra equation of the 2nd kind - also in honor of the mathematician who proposed and studied this structure properties This integral equation is called Volterra's inhomogeneous equation.

If in equations (5.88), (5.91) we put

$$f(x) = 0, \tag{5.92}$$

then these integral equations are called, respectively, the Fredholm equations of the 1st kind and Volterra equations of the 1st kind or, again, respectively, the homogeneous Fredholm equation and the homogeneous Volterra equation.

An operator A that in space C transforms a class of functions φ into itself according to an expression

$$A\varphi = \lambda \int_a^b K(x,y)\varphi(y)dy, \tag{5.93}$$

is called the Fredholm operator with kernel $K(x,y)$.

Taking into account the expression (5.93), the Fredholm equation (5.88) can be rewritten in operator form as

$$\varphi = f + A\varphi, \tag{5.94}$$

and its n-th iteration with successive approximations to the solution will have the form

$$\varphi_n = f + A\varphi_{n-1}. \tag{5.95}$$

It is obvious that the Volterra operator will formally differ from the Fredholm operator only in that the upper limit of the integral in it will not be a constant b, but an independent variable x, but, in fact, due to this, we will have the power of the set of functions formed by the Volterra operator, greater than the power of the set of functions, which is formed by the Fredholm operator. But since the properties of the Fredholm and Volterra operators coincide, we will focus further explanations on the Fredholm operator. And this operator has three such basic properties: firstly, it is linear, secondly, it is continuous, thirdly, it is bounded, which is sufficient to prove that its mapping in metric space has a single fixed point, which is a solution using the Fredholm equation, and to build a resolvable algorithm for obtaining this solution. So,

1) the Fredholm operator is linear because

$$\begin{aligned} A(\varphi_1 + \varphi_2) &= \lambda \int_a^b K(x,y)(\varphi_1(y) + \varphi_2(y))dy = \lambda \int_a^b K(x,y)\varphi_1(y)dy + \\ &+ \lambda \int_a^b K(x,y)\varphi_2(y)dy = A\varphi_1 + A\varphi_2 \end{aligned}; \tag{5.96}$$

2) the Fredholm operator is continuous, because if there is a limit φ of the sequence $\{\varphi_n\}$, i.e.,

$$\lim_{n\to\infty} \varphi_n = \varphi, \tag{5.97}$$

where

$$\varphi_n(x) = f(x) + \lambda \int_a^b K(x,y)\varphi_{n-1}(y)dy = f + A\varphi_{n-1}, \tag{5.98}$$

then

$$\begin{aligned} {}_n\underline{\lim}_{\infty}\varphi_n = {}_n\underline{\lim}_{\infty}\left(f(x) + \lambda \int_a^b K(x,y)\varphi_{n-1}(y)dy \right) = f(x) + \lambda \int_a^b K(x,y)\,{}_n\underline{\lim}_{\infty}\varphi_{n-1}(y)dy = \\ = f(x) + \lambda \int_a^b K(x,y)\varphi(y)dy = f + A\varphi = \varphi \end{aligned} \tag{5.99}$$

3) the Fredholm operator is bounded because

$$\begin{aligned} \|A\varphi\| = \frac{\sup}{[a \le x,y \le b]}|A\varphi| = \frac{\sup}{[a \le x,y \le b]}\left|\lambda \int_a^b K(x,y)\varphi(y)dy\right| \le \\ \le \frac{\sup}{[a \le x,y \le b]}\lambda \int_a^b |K(x,y)|\,|\varphi(y)|\,dy = \\ = \lambda \int_a^b \frac{\sup}{[a \le x,y \le b]}|K(x,y)|\,\frac{\sup}{[a \le y \le b]}|\varphi(y)|\,dy = \lambda \|K(x,y)\|\,\|\varphi(y)\| \int_a^b dy = \\ = \lambda\,\|K(x,y)\|\,\|\varphi(y)\|\,(b-a) \end{aligned} \tag{5.100}$$

We will show that the Fredholm operator performs a compressed mapping of the space of functions φ into itself. To do this, we will apply the standard algorithm of the compressed mapping method, but not with respect to metrics, but with respect to norms, according to which we will have

$$\begin{aligned} \|\varphi_1,\varphi_2\| = \frac{\sup}{[a \le x \le b]}|\varphi_1(x) - \varphi_2(x)| = \frac{\sup}{[a \le x \le b]}|(f + A\varphi_0) - (f + A\varphi_1)| = \\ = \frac{\sup}{[a \le x \le b]}|A(\varphi_0 - \varphi_1)| = \frac{\sup}{[a \le x \le b]}\left|\lambda \int_a^b K(x,y)(\varphi_0(y) - \varphi_1(y))dy\right| \le \\ \le \lambda \int_a^b \frac{\sup}{[a \le x,y \le b]}|K(x,y)|\,\frac{\sup}{[a \le y \le b]}|\varphi_0(y) - \varphi_1(y)|dy = \\ = \lambda \|K(x,y)\|\,\|\varphi_0,\varphi_1\| \int_a^b dy = \lambda \|K(x,y)\|(b-a)\|\varphi_0,\varphi_1\| = \theta\|\varphi_0,\varphi_1\|, \end{aligned} \tag{5.101}$$

$$\begin{aligned} \|\varphi_2,\varphi_3\| = \frac{\sup}{[a \le x \le b]}|\varphi_2(x) - \varphi_3(x)| = \frac{\sup}{[a \le x \le b]}|(f + A(A\varphi_0)) - (f + A(A\varphi_1))| = \\ = \frac{\sup}{[a \le x \le b]}|A^2(\varphi_0 - \varphi_1)| = \frac{\sup}{[a \le x \le b]}\left|\lambda \int_a^b K(x,y)[\lambda \int_a^b K(y,\omega)(\varphi_0(\omega) - \varphi_1(\omega))d\omega]dy\right| \le \\ \le \lambda^2 \int_a^b \{[\frac{\sup}{[a \le x,y \le b]}|K(x,y)|]^2 \frac{\sup}{[a \le \omega \le b]}|\varphi_0(\omega) - \varphi_1(\omega)|\int_a^b d\omega\}dy = \\ = \lambda^2 \|K(x,y)\|^2\,\|\varphi_0,\varphi_1\|(b-a)\int_a^b dy = \lambda^2 \|K(x,y)\|^2\,(b-a)^2\,\|\varphi_0,\varphi_1\| = \theta^2\|\varphi_0,\varphi_1\|, \end{aligned} \tag{5.102}$$

Continuing, by analogy, for the sequence (5.95) we obtain

$$\begin{aligned} \|\varphi_n,\varphi_{n+1}\| = \frac{\sup}{[a \le x \le b]}|\varphi_n(x) - \varphi_{n+1}(x)| = \frac{\sup}{[a \le x \le b]}|(f + A(A^{n-1}\varphi_0)) - (f + A(A^{n-1}\varphi_1))| = \\ = \frac{\sup}{[a \le x \le b]}|A^n(\varphi_0 - \varphi_1)| \le \lambda^n \|K(x,y)\|^n\,(b-a)^n\,\|\varphi_0,\varphi_1\| = \theta^n\|\varphi_0,\varphi_1\| \end{aligned} \tag{5.103}$$

In expressions (5.101), (5.102), (5.103)

$$\theta = \lambda \left\| K(x,y) \right\| (b-a). \tag{5.104}$$

As we remember, in order for the expressions (5.101), (5.102), (5.103) to carry out compression leading to a fixed point, which is a solution of the integral equation (5.88), it is necessary that the inequality

$$\theta < 1. \tag{5.105}$$

And since the norm $\left\| K(x,y) \right\|$ of the kernel $K(x,y)$ of the Fredholm operator and the segment $[a,b]$ of the numerical axis on which the integration is carried out in the integral equation (5.88) are determined by the conditions of the problem for which this equation was synthesized, it follows from the expressions (5.104), (5.105) that in order for the inhomogeneous Fredholm's integral equation (5.88) had a unique solution $\varphi(x)$, it is necessary to choose the parameter λ based on the relation

$$\lambda < \frac{1}{\left\| K(x,y) \right\| (b-a)}. \tag{5.106}$$

Since the norm $\left\| \varphi_n, \varphi_{n+1} \right\|$ specifies the distance between two points φ_n and φ_{n+1} the metric space C of continuous functions with the metric (5.75), and therefore

$$\rho(\varphi_n, \varphi_{n+1}) = \frac{\sup}{[a \le x \le b]} \left| \varphi_n(x) - \varphi_{n+1}(x) \right| = \left\| \varphi_n, \varphi_{n+1} \right\|, \tag{5.107}$$

then the expression (5.103), which we used to prove that the Fredholm operator compresses mappings in a given functional metric space, can also be used to decide at which stage the iterative process can be stopped

$$\begin{aligned}
&\varphi_0(x) = f(x) = A^0 f, \\
&\varphi_1(x) = f(x) + \lambda \int_a^b K(x,y)\varphi_0(y)dy = f(x) + \lambda \int_a^b K(x,y) f(y)dy = A^0 f + Af, \\
&\varphi_2(x) = f(x) + \lambda \int_a^b K(x,y)\varphi_1(y)dy = A^0 f + Af + A^2 f, \\
&\text{.....}, \\
&\varphi_n(x) = f(x) + \lambda \int_a^b K(x,y)\varphi_{n-1}(y)dy = \sum_{i=0}^{n} A^i f(x),
\end{aligned} \tag{5.108}$$

calculation of approximations $\varphi_n(x)$ to the solution $\varphi(x)$ of equation (5.88). To do this, it is necessary to set the permissible value ε of the approximation error and stop the calculation according to the expressions (5.108) at the value ***n*** for which the inequality will be fulfilled

$$\left\| \varphi_n, \varphi_{n+1} \right\| < \varepsilon. \tag{5.109}$$

And at the end of this section, we will construct the resolvent of the Fredholm operator. Let's rewrite the operator equation (5.94) in the form

$$\varphi - A\varphi = f. \tag{5.110}$$

The operator equation (5.110) can also be presented as follows:

$$(I - A)\varphi = f. \tag{5.111}$$

Let's define the inverse operator $(I - A)^{-1}$ and multiply the operator equation (5.111) by it on the left. We will get

$$(I-A)^{-1}(I-A)\varphi=(I-A)^{-1}f, \tag{5.112}$$

or

$$\varphi=(I-A)^{-1}f=R_{\Phi}f \quad . \tag{5.113}$$

According to the expression (5.39) given in subsection 5.2, the operator

$$R_{\Phi}=(I-A)^{-1}=\frac{I}{I-A}=I+A+A^2+A^3+\ldots=\sum_{i=0}^{\infty}A^i \tag{5.114}$$

and is the resolvent of the Fredholm operator equation. And since, in determining it, we used the formula for the sum of members of an infinitely decreasing geometric progression with denominator A, then there will be a resolvent R_{Φ} only if

$$\|A\|<1, \tag{5.115}$$

that is, when

$$\lambda<\frac{1}{\|K(x,y)\|\,(b-a)\|\varphi\|}, \tag{5.116}$$

which is fully consistent with the expression (5.106).

Taking into account the expression (5.114), the solution (5.113) of the operator equation (5.110) can be rewritten as

$$\varphi(x)=\sum_{i=0}^{\infty}A^i f(x) \tag{5.117}$$

And applying the limit transition to the last equation in the system of approximations (5.108), we have

$${}_n\underline{\lim}_{\infty}\varphi_n(x)={}_n\underline{\lim}_{\infty}\sum_{i=0}^{n}A^i f(x)=\sum_{i=0}^{\infty}A^i f(x). \tag{5.118}$$

It follows from expressions (5.117) and (5.118) that

$${}_n\underline{\lim}_{\infty}\varphi_n(x)=\varphi(x). \tag{5.119}$$

With this, we have confirmed that in the process of successive approximations (5.108) we will necessarily arrive at an approximate solution of the Fredholm equation of the 2nd kind, the error of which will not exceed the given value ε.

5.5 Example of solving operator equations

As an example, we will demonstrate how to apply the method of compressed mappings to solve an operator equation that has the form of an integral Fredholm equation of the 2nd kind given in the form (5.88).

Let

$$a=0,\quad b=1,\quad f(x)=x,\quad K(x,y)=x+y,\quad \varepsilon=0{,}1. \tag{5.120}$$

Substituting conditions (120) into expression (5.88), we will have a concretized Fredholm integral equation of the 2nd kind in the form

$$\varphi(x)=x+\lambda\int_0^1(x+y)y\,dy, \tag{5.121}$$

We need to start solving this equation by the method of compressed mappings by determining the admissible value of the parameter λ, for which we should use the inequality (5.106), in the right-hand side of which we must also substitute the norm of the kernel of the Fredholm operator, the numerical value of which is determined by the expression (5.90). For the conditions specified by expressions (5.120), the numerical value of this norm will be equal to

$$\|K(x,y)\| = \frac{\sup}{[a \le x, y \le b]} |K(x,y)| = \frac{\sup}{[0 \le x, y \le 1]} |x + y| = |1+1| = 2 . \tag{5.122}$$

Substituting the numerical values of the corresponding parameters from the expressions (5.120) and (5.122) into the inequality (5.106), we will have

$$\|K(x,y)\| = \frac{\sup}{[a \le x, y \le b]} |K(x,y)| = \frac{\sup}{[0 \le x, y \le 1]} |x + y| = |1+1| = 2 . \tag{5.123}$$

We accept

$$\lambda = 0,4 . \tag{5.124}$$

For our conditions, the iterative process (5.108) of approximations to the solution of equation (5.121) will have the form

$$\begin{cases} \varphi_0(x) = x, \\ \varphi_n(x) = x + 0,4\int_0^1 (x+y)\varphi_{n-1}(y)dy, \qquad n = 1,2,3,... \end{cases} \tag{5.125}$$

We will stop this process and declare the last approximation that satisfies the criterion (5.109) to be the approximate solution of equation (5.121), which for our conditions (5.120) will be

$$\|\varphi_n, \varphi_{n+1}\| < 0,1 \qquad n = 0,1,2,... \tag{5.126}$$

And then we proceed to iterations.

Let $n = 1$.

In this case, from the expression (5.125), we have

$$\begin{aligned} \varphi_1(x) &= x + 0,4\int_0^1 (x+y)ydy = x + 0,4x\int_0^1 ydy + 0,4\int_0^1 y^2 dy = x + 0,4x\left(\frac{y^2}{2}\right)\Big|_0^1 + \\ &+ 0,4\left(\frac{y^3}{3}\right)\Big|_0^1 = x + 0,4x\left(\frac{1}{2}\right) + 0,4\left(\frac{1}{3}\right) = 1,2x + 0,133 \end{aligned} \tag{5.127}$$

Let's check the obtained approximation $\varphi_1(x)$ to the solution $\varphi(x)$ using the criterion (5.126), substituting the value $\varphi_0(x)$ from the expression (5.125) and the value $\varphi_1(x)$ from the expression (5.127) into which and revealing the norm based on the expression (5.107), we will have

$$\begin{aligned} \|\varphi_0, \varphi_1\| &= \frac{\sup}{[0 \le x \le 1]} |\varphi_0(x) - \varphi_1(x)| = \frac{\sup}{[0 \le x \le 1]} |x - 1,2x - 0,133| = \\ &= |(-0,2) \bullet 1 - 0,133| = 0,333 \end{aligned} \tag{5.128}$$

Comparing the numerical value of expression (5.128) with the right-hand side of expression (5.126) when n=0 ,we see that the inequality is not satisfied, so it is impossible to stop the iterative process upon obtaining the approximation (5.127).

And so let it now $n = 2$.

In this case, from expressions (5.125) and (5.127) we have

$$\varphi_2(x) = x + 0{,}4\int_0^1 K(x,y)\varphi_1(y)dy = x + 0{,}4\int_0^1 (x+y)(1{,}2y + 0{,}133)dy = x + 0{,}48x\int_0^1 ydy +$$

$$+ 0{,}48\int_0^1 y^2dy + 0{,}053x\int_0^1 dy + 0{,}053\int_0^1 ydy = x + 0{,}48x\left(\frac{y^2}{2}\right)\Big|_0^1 + 0{,}48\left(\frac{y^3}{3}\right)\Big|_0^1 + 0{,}053x(y)\Big|_0^1 + \quad (5.129)$$

$$+ 0{,}053\left(\frac{y^2}{2}\right)\Big|_0^1 = x + 0{,}24x + 0{,}16 + 0{,}053x + 0{,}026 = 1{,}293x + 0{,}186$$

We will check the obtained approximation $\varphi_2(x)$ to the solution $\varphi(x)$, by using the criterion (5.126), substituting the value $\varphi_1(x)$ from the expression (5.127) and the value $\varphi_2(x)$ from the expression (5.129) into which and revealing the norm based on the expression (5.107), we will have

$$\|\varphi_1, \varphi_2\| = \sup_{[0 \le x \le 1]} |\varphi_1(x) - \varphi_2(x)| = \sup_{[0 \le x \le 1]} |1{,}2x + 0{,}133 - 1{,}293x - 0{,}186| = \quad (5.130)$$

$$= |(-0{,}093) \bullet 1 - 0{,}053| = 0{,}146$$

Comparing the numerical value of the expression (5.130) with the right-hand side of the expression (5.126) when $n = 1$,we see that the inequality is not satisfied, so it is impossible to stop the iterative process upon obtaining the approximation (5.129)

So let it now ***n=3***

In this case, from expressions (5.125) and (5.129), we have

$$\varphi_3(x) = x + 0{,}4\int_0^1 K(x,y)\varphi_2(y)dy = x + 0{,}4\int_0^1 (x+y)(1{,}293y + 0{,}186)dy = x + 0{,}516x\int_0^1 ydy +$$

$$+ 0{,}516\int_0^1 y^2dy + 0{,}074x\int_0^1 dy + 0{,}074\int_0^1 ydy = x + 0{,}516x\left(\frac{y^2}{2}\right)\Big|_0^1 + 0{,}516\left(\frac{y^3}{3}\right)\Big|_0^1 + 0{,}074x(y)\Big|_0^1 + \quad (5.131)$$

$$+ 0{,}074\left(\frac{y^2}{2}\right)\Big|_0^1 = x + 0{,}258x + 0{,}172 + 0{,}074x + 0{,}037 = 1{,}332x + 0{,}209$$

Let's check the obtained approximation $\varphi_3(x)$ to the solution $\varphi(x)$ using the criterion (5.126), substituting the value $\varphi_2(x)$ from the expression (5.129) and the value $\varphi_3(x)$ from the expression (5.131) into which and revealing the norm based on the expression (5.107), we will have

$$\|\varphi_2, \varphi_3\| = \sup_{[0 \le x \le 1]} |\varphi_2(x) - \varphi_3(x)| = \sup_{[0 \le x \le 1]} |1{,}293x + 0{,}186 - 1{,}332x - 0{,}209| = \quad (5.132)$$

$$= |(-0{,}039) \bullet 1 - 0{,}023| = 0{,}062$$

Comparing the numerical value of the expression (5.132) with the right part of the expression (5.126) at $n = 2$, we see that inequality is performed, so the iterative process after receiving the approximation (5.131) can be stopped and assume that with an error not exceeding the specified conditions (5.120) the numerical value 0.1 approximate solution of integral equation (5.121) is an expression

$$\varphi(x) \approx \varphi_3(x) = 1{,}332x + 0{,}209. \quad (5.133)$$

5.6 Python-realization of algorithms of calculating the norm of operators and Solving operator equations

Python program to calculate the norm of the function differentiation operator in the metric functional space $C[a, b]$ for the case when $a = 1$, $b = 3$

(Program 17)

```
In [1]: import sympy
In [2]: from sympy import *
In [3]: import numpy as np
In [4]: t = symbols('t')
In [5]: x = Function('x')(t)
In [6]: y = Function('y')(t)
In [7]: z=Function('z')(t)
In [8]: x=2*t-0.5*t**2+t**0.5
In [9]: y=x.diff();y
Out[9]: 0.5*t**(-0.5) - 1.0*t + 2
In [10]: z=y.diff();z
Out[10]: -0.25*t**(-1.5) - 1.0
In [11]: t=np.linspace(1,3,21)
In [12]: g1=lambda t: 2*t-0.5*t**2+t**0.5
In [13]: g1vec=np.vectorize(g1)
In [14]: g11=g1vec(t)
In [15]: g111=np.piecewise(g11,[g11<0,\
                g11>=0],[lambda g11:-g11,\
                lambda g11:g11])
In [16]: ng1=g111.max();ng1
Out[16]: 3.4715750888103103
In [17]: ng1=np.round_(ng1,3)
In [18]: ng1
Out[18]: 3.472
In [19]: g2=lambda t: 2-1.0*t+\
                        0.5*t**(-0.5)
In [20]: g2vec=np.vectorize(g2)
In [21]: g22=g2vec(t)
In [22]: g222=np.piecewise(g22,[g22<0,\
                g22>=0],[lambda g22:-g22,\
                lambda g22:g22])
In [23]: ng2=g222.max()
In [24]: ng2
Out[24]: 1.5
In [25]: nd_dt=ng2/ng1;nd_dt
Out[25]: 0.43202764976958524
In [26]: np.round_(nd_dt,3)
Out[26]: 0.432
In [27]: g3= lambda t:-0.25*t**(-1.5) -1.0
In [28]: g3vec=np.vectorize(g3)
In [29]: g33=g3vec(t)
In [30]: g333=np.piecewise(g33,[g33<0,\
                g33>=0],[lambda g33:-g33,\
                        lambda g33:g33])
In [31]: ng3=g333.max()
In [32]: ng3
Out[32]: 1.25
In [33]: ndd_dtdt=ng3/ng1;ndd_dtdt
Out[33]: 0.36002304147465436
In [34]: np.round_(ndd_dtdt,3)
Out[34]: 0.36
```

End of program 17

Python program to calculate the norm of the function differentiation operator in the metric functional space $L_2[a,b]$ for the case when $a = 0$, $b = 2$

(Program 18)

```
In [1]: import sympy
In [2]: from sympy import *
In [3]: t = symbols('t')
In [4]: x = Function('x')(t)
In [5]: y = Function('y')(t)
In [6]: z=Function('z')(t)
In [7]: x=1+2*t-0.5*t**2-0.25*t**3
In [8]: y=x.diff();y
Out[8]: -0.75*t**2 - 1.0*t + 2
In [9]: z=y.diff();z
Out[9]: -1.5*t - 1.0
In [10]: f1=x*x;f1
Out[10]: 4*(-0.125*t**3 - 0.25*t**2 +\
          t + 1/2)**2
In [11]: f11=expand(f1);f11
Out[11]:
0.0625*t**6 + 0.25*t**5 - 0.75*t**4 - \
         2.5*t**3 + 3.0*t**2 + 4*t + 1
In [12]: b=integrate(f11,(t,0,2));b
Out[12]: 7.00952380952381
In [13]: nx=b**0.5;nx
Out[13]: 2.64755053011719
In [14]: f2=y*y;f2
Out[14]: 4*(-0.375*t**2 - 0.5*t + 1)**2
```

```
In [15]: f22=expand(f2);f22
Out[15]: 0.5625*t**4 + 1.5*t**3 - \
         2.0*t**2 - 4.0*t + 4
In [16]: b1=integrate(f22,(t,0,2));b1
Out[16]: 4.26666666666667
In [17]: ny=b1**0.5;ny
Out[17]: 2.06559111797729
In [18]: f3=z*z;f3
Out[18]:
         2.25*(-t - 0.666666666666667)**2
In [19]: f33=expand(f3);f33
Out[19]: 2.25*t**2 + 3.0*t + 1.0
In [20]: b2=integrate(f33,(t,0,2));b2
Out[20]: 14.0000000000000
In [21]: nz=b2**0.5;nz
Out[21]: 3.74165738677394
In [22]: nd_dt=ny/nx;nd_dt
Out[22]: 0.780189497605494
In [23]: ndd_dtdt=nz/nx;ndd_dtdt
Out[23]: 1.41325249290268
```

End of program 18

The Python language program for solving the algebraic equation $f(x) = 0$ in the variant $ax^2 + bx + c = 0$, given on the segment of values $x \in [e, h]$ independent variable by compressed reflection method after representing it in the form $x = f(x)$, i.e in the variant $x = -\frac{ax^2}{b} - c/b$

(Program 19)

```
In [1]: import numpy as np
In [2]: import sympy as sm
In [3]: a = 1.0
In [4]: b = -6.0
In [5]: c = 8.75
In [6]: po = 0.05
In [7]: e = 0
In [8]: h = 2
In [9]: N = 21
In [10]: x, y = sm.symbols('x y')
In [11]: fx = -a*x**2/b-c/b
In [12]: fy = -a*y**2/b-c/b
In [13]: expr1 = -a*x**2/b-c/b
In [14]: expr2 = -a*y**2/b-c/b
In [15]: f0 = sm.lambdify(x, expr1,\
                  "numpy")
In [16]: f00 = sm.lambdify(y, expr2, \
                 "numpy")
In [17]: x = np.linspace(e, h, N)
In [18]: print(f"x: {x}")
x: [0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 \
    0.9 1.  1.1 1.2 1.3 1.4 1.5 1.6 \
    1.7 1.8 1.9 2.]
In [19]: f1 = f0(x)
In [20]: print(f"f1: {f1}")
    f1: [1.45833333 1.46     1.465    \
    1.47333333 1.485    1.5      \
    1.51833333 1.54     1.565    \
    1.59333333 1.625    1.66     \
    1.69833333 1.74     1.785    \
    1.83333333 1.885    1.94     \
    1.99833333 2.06     2.125    ]
In [21]: f11 = f1.min()
In [22]: f12 = f1.max()
In [23]: print(f"f11, f12: {f11, f12}")
    f11, f12: (1.45833333333333,\
        2.124999999999998)
In [24]: e1 = float(e); h1 = float(h)

In [25]: if e1 < f11 and f12 < h1:
             print("1st if")
             x[0] = 1.0
             for i in range(N):
                 print(f"i: {i}")
                 x[i+1] = -a*(x[i])**2/b-c/b
                 xx = x[i+1]-x[i]
                 axx = abs(xx)
                 if axx < po:
                     print(x[i+1])
                     break
```

```
                    else:
                        i += 1
                        print('1st else')
        else:
            for j in range(1,4):
                print(f"j: {j}")
                e2 = e-j
                h2 = h+j
                N1=(h1-e1)/0.1+1
                N2=int(N1)
                y = np.linspace(e2, h2, N2)
                f2 = f00(y)
                f21 = f2.min( )
                f22 = f2.max( )
                e3 = float(e2); h3 = float(h2)
                if e3 < f21 and f22 < h3:
                    print("2nd if")
                    y[0] = 1.0
                    for k in range(N2):
                        print(f"k: {k}")
                        y[k+1] = -a*(y[k])**2/b-c/b
                        yy = y[k+1]-y[k]
                        ayy = abs(yy)
                        print(f"ayy: {ayy}")
                        if ayy < po:
                            print(f"y[k+1]:{y[k+1]}")
                            break
                        else:
                            k += 1
                            print('2nd else')
```

Printing the results obtained in the cycle:

```
j: 1
2nd if
k: 0
ayy: 0.625
2nd else
k: 1
ayy: 0.2734375
2nd else
k: 2
ayy: 0.16057332356770804
2nd else
k: 3
ayy: 0.1059101050271205
2nd else
k: 4
ayy: 0.07455950924982035
2nd else
k: 5
ayy: 0.054731667403494555
2nd else
k: 6
ayy: 0.041356092063042915
y[k+1]: 2.3355681973111864
j: 2
j: 3
```

End of program 19.

Python language program to solve the integral equation of Fredholm of the second kind

$$\boldsymbol{\varphi(x) = f(x) + \lambda \int_a^b K(x,y)\varphi(y)dy,}$$

in which: $a = 0$, $b = \pi$, $f(x) = \sin(x)$, , $K(x,y) = xe^{-y}$, $\varepsilon = 0.01$, compressed reflection method

(Program 20)

```
In [1]: import numpy as np
In [2]: import sympy as smp
In [3]: a = 0
In [4]: b = 3
In [5]: po = 0.05
In [6]: N = 31
In [7]: x, y = smp.symbols('x y')
In [8]: fxy = x*smp.exp(-y)
In [9]: expr=fxy
In [10]: f0= smp.lambdify((x,y),expr,\
          "numpy")
In [11]: x = np.linspace(a,b,N)
In [12]: y = np.linspace(a,b,N)
In [13]: f1 = f0(x,y)
In [14]: print(f"f1: {f1}")
f1: [0.         0.09048374 0.16374615
0.22224547 0.26812802 0.30326533
0.32928698 0.34760971 0.35946317
0.36591269 0.36787944 0.36615819
 0.36143305 0.35429133 0.34523575
0.33469524 0.32303443 0.31056199
0.297538   0.28418038 0.27067057
0.2571585  0.24376695 0.23059534
 0.21772309 0.2052125  0.1931113
 0.18145488 0.17026818 0.15956734
0.14936121]

In [15]: f11 = f1.max()
In [16]: print(f"f11: {f11}")
          f11: 0.36787944117144233
In [17]: q = 1/(f11*(b-a))
In [18]: q1 = q - 0.5
In [19]: q1=q1.round(1)
In [20]: print(f"q1: {q1}")
q1: 0.4
In [21]: x, y = smp.symbols('x y')
In [22]: g = smp.Function('g')(x)
In [23]: h = smp.Function('h')(x)
In [24]: f2 = x**2
In [25]: g = f2
In [26]: gg = []
In [27]: gg.append(g)

In [28]: for i in range(20):
             print(f"i: {i}")
             g = f2 + smp.integrate(q1*x*smp.exp(-y)*(gg[i]).subs(x,y),(y,0,3))
             gg.append(g)
             h = gg[i+1] - gg[i]
             mh = smp.integrate(h**2,(x,0,3)).n(3)
             print(f"mh: {mh}, {type(mh)}")
             nh = mh**0.5
             print(f"nh: {nh}, {type(nh)}")
             if nh > po:
                i += 1
             else:
                print(f"gg[i+1]: {gg[i+1].n(3)}")
                break
```

Printing the results obtained in the cycle:

```
i: 0
mh: 1.92, <class 'sympy.core.numbers.Float'>
nh: 1.38437852708354, <class 'sympy.core.numbers.Float'>
i: 1
mh: 0.197, <class 'sympy.core.numbers.Float'>
nh: 0.443458310822449, <class 'sympy.core.numbers.Float'>
i: 2
mh: 0.0202, <class 'sympy.core.numbers.Float'>
nh: 0.142055441765376, <class 'sympy.core.numbers.Float'>
i: 3
mh: 0.00207, <class 'sympy.core.numbers.Float'>
nh: 0.0455071838073526, <class 'sympy.core.numbers.Float'>
gg[i+1]: x**2 + 0.672*x
```

End of program 20.

5.7 Self -Testing Task

1. Give a definition of the operator. Give examples.
2. Which operator is linear? What properties of a linear operator do you know?
3. What is the norm of the operator? Write down the inequality of the triangle for the norm of the operator.
4. Which linear operator is symmetrical?
5. What is a conjugated and self -burned operators?
6. Which operator is called unitary?
7. How to determine the inverse operator?
8. What is the resolve of the heterogeneous operator equation?
9. What are the values of the operator regular and what are its spectrum?
10. What are the characteristic numbers or own values of the operator?
11. Are the characteristic numbers of the operator within the set of its regular values?
12. How can resolvets express through a row by the operator's degrees?
13. What are the properties of the operator resolvents?
14. Under what conditions does a heterogeneous operator equation have a correct solution?
15. What is meant by a group of operators and what properties of such a group do you know?
16. What is the essence of the idea of the compressed reflections of the operator?
17. Formulate and prove Banah's theorem about a single fixed point in the compressed display of the operator.
18. Using the compression method of reflections, prove that the algebraic equation has a single fixed point on a given segment of the numerical axis.
19. If the algebraic equation has several roots, how to find all its fixed points?
20. Using the compression method of reflections, prove the existence of a single fixed point of differential operator.
21. How to transform an integral operator into a differential?
22. Using the compression method of reflections, prove the existence of a single fixed point of Fredholm operator.
23. What is Fredholm's resolution?
24. Construct an algorithm for sequential approximates to a single fixed point of Fredholm operator.
25. What is the criterion for achieving the necessary accuracy of the approximate solution of the heterogeneous integral equation of Fredholm of the 2nd kind?
26. What is the fundamental difference between Fredholm and Voltaire operators?
27. How are the capacity of the sets of integral equations of Fredholm and Voltaire compassionate?

28 Show the commands in the programs that set cycles.

Chapter 6. SPECIAL OPERATORS AND THEIR APPLICATIONS

6.1 Direct and inverse Laplace operators

From the general course of higher mathematics, which is taught to students of technical higher education institutions, it is known that with the **help of the Laplace operator**

$$L\{f(t)\} = F(p) = \int_0^\infty f(t)e^{-pt}dt \tag{6.1}$$

each continuous time function $f(t)$, given on the set of real numbers, which satisfies the condition $f(t) = 0$ at $t < 0$, the Dirichlet condition and is called the original, can be matched with a function *F* of the complex variable $p = \sigma + j\omega$, which is called the image of the original on the complex plane. This correspondence is recorded as follows:

$$f(t) \dot{\Leftrightarrow} F(p). \tag{6.2}$$

For example, the time function of a unit jump

$$x(t) = 1(t) = \begin{cases} 1, & t \geq 0, \\ 0, & t < 0, \end{cases} \tag{6.3}$$

on the complex plane corresponds to the image

$$F(p) = \int_0^\infty 1(t)e^{-pt}dt = \int_0^\infty e^{-pt}dt = \frac{1}{-p} \cdot e^{-pt}\Big|_0^\infty = \frac{1}{p}, \tag{6.4}$$

or

$$1(t) \dot{\Leftrightarrow} \frac{1}{p}. \tag{6.5}$$

Another example – the exponent $e^{-\alpha t}$ at $t \geq 0$ on the complex plane corresponds to the image

$$F(p) = \int_0^\infty e^{-\alpha t} \cdot e^{-pt}dt = \int_0^\infty e^{-(p+\alpha)t}dt = \frac{1}{-(p+\alpha)}e^{-(p+\alpha)t}\Big|_0^\infty = \frac{1}{p+\alpha}. \tag{6.6}$$

The main advantage of the analysis in the area of images $F(p)$, i.e. on the complex plane, compared to the analysis in the area of originals $f(t)$, i.e. in time space, is that under zero initial conditions, the operation of differentiation $\frac{d}{dt}$ of the original $f(t)$ in time space corresponds to the operation of multiplication by a complex variable p of its image $F(p)$ on the complex plane, i.e

$$\frac{df}{dt} = \dot{f}(t) \dot{\Leftrightarrow} p \cdot F(p), \tag{6.7}$$

since

$$L\left\{\frac{df}{dt}\right\} = \int_0^\infty \frac{df}{dt}e^{-pt}dt = \left(f(t)e^{-pt}\right)\Big|_0^\infty - \int_0^\infty f(t)\left(-pe^{-pt}\right)dt = p\int_0^\infty f(t)e^{-pt}dt = pF(p). \tag{6.8}$$

We draw attention to the fact that when integrating according to expression (6.8), we used the well-known method of integration by parts, assigning new variables u, v values $u = e^{-pt}$, $dv = df$, as a result of which we will have $v = f$, $du = -pe^{-pt}dt$ and a zero initial condition $f(0) = 0$.

Applying the same technique, we get that

$$\frac{d^2 f}{dt^2} = \ddot{f}(t) \dot{\Leftrightarrow} p^2 \cdot F(p) \tag{6.9}$$

and

$$\frac{d^n f}{dt^n} = f^{(n)}(t) \dot{\Leftrightarrow} p^n \cdot F(p). \tag{6.10}$$

And the operation of integrating the original $f(t)$ in time space corresponds to the operation of division by the complex variable *p* of its image $F(p)$ on the complex plane, i.e.

$$\int_0^t f(\tau)\,d\tau \dot{\Leftrightarrow} \frac{F(p)}{p}, \tag{6.11}$$

since

$$L\left\{\int_0^t f(\tau)d\tau\right\} = \int_0^\infty \left(\int_0^t f(\tau)d\tau\right)e^{-pt}dt = \left(\frac{1}{-p}e^{-pt}\int_0^t f(\tau)d\tau\right)\Bigg|_0^\infty - \int_0^\infty \left(\frac{1}{-p}e^{-pt}\right)f(t)dt = \\ = \frac{1}{p}\int_0^\infty f(t)e^{-pt}dt = \frac{1}{p}F(p) \tag{6.12}$$

When integrating according to expression (6.12), we also used the method of integration by parts, assigning new variables u, v the values $u = \int_0^t f(\tau)d\tau$, $dv = e^{-pt}dt$, as a result of which we will have $v = \left(\frac{1}{-p}e^{-pt}\right)$, $du = f(t)dt$.

Applying the same technique, we get that

$$\underbrace{\int_0^t \ldots \int_0^t}_{n} f(\tau_1, \tau_2, \ldots, \tau_n)\,d\tau_1\,d\tau_2 \ldots d\tau_n \dot{\Leftrightarrow} \frac{F(p)}{p^n}. \tag{6.13}$$

Deu to the properties (6.7), (6.11) and their consequences (6.10), (6.13), differential and integral equations written in time space correspond to algebraic equations on the complex plane, which are much easier to solve, since this is taught in school.

For example, to the differential equation in the domain of the originals $x(t)$, $y(t)$

$$a_2\frac{d^2y(t)}{dt^2} + a_1\frac{dy(t)}{dt} + a_0 y(t) = b_1\frac{dx(t)}{dt} + b_0 x(t) \tag{6.14}$$

on the complex plane corresponds to the algebraic equation

$$a_2 p^2 Y(p) + a_1 pY(p) + a_0 Y(p) = b_1 pX(p) + b_0 X(p) \tag{6.15}$$

relative to images $X(p)$ and $Y(p)$. Its solution is the function $Y(p)$, which can be determined from equation (6.15) as follows:

$$Y(p)=\frac{b_1 p+b_0}{a_2 p^2+a_1 p+a_0}X(p) \tag{6.16}$$

or

$$Y(p)=W(p)\cdot X(p), \tag{6.17}$$

where

$$W(p)=\frac{b_1 p+b_0}{a_2 p^2+a_1 p+a_0}. \tag{6.18}$$

If we consider $X(p)$ as the image of the signal $x(t)$ which enters the input of a linear dynamic system (LDS), the mathematical model of which can be written in the form (6.14), where $y(t)$ is the response of this system to the input signal $x(t)$, then the function $W(p)$ can be interpreted as a transfer function of the system (Fig. 11).

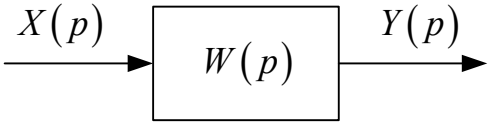


Figure 11 – Generalized structural diagram of a linear dynamic system in the image area

As can be seen from the expressions (6.16)–(6.18), the transfer function $W(p)$ does not depend on the external signals acting on the system, but uniquely characterizes its ability to transfer these signals from its input to the output. And therefore, this function is one of the most important mathematical models of linear dynamic systems, for which the Laplace transform (6.1) specifies a mutually unambiguous correspondence between the originals and their images.

It follows from the expression (6.17) that, knowing the image $X(p)$ of the input signal $x(t)$ and the image $Y(p)$ of the system response to this signal, the transfer function can be obtained by taking their ratio, i.e.

$$W(p)=\frac{Y(p)}{X(p)} \tag{6.19}$$

An inverse operator exists for the Laplace operator defined by expression (6.1)

$$L^{-1}\{F(p)\}=\frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty}F(p)e^{pt}dp=\frac{1}{2\pi j}\oint F(p)e^{pt}dp=f(t), \tag{6.20}$$

according to which the original $f(t)$ can be found from a known image $F(p)$, which, as a rule, is used only for constructing tables of correspondence between $f(t)$ and $F(p)$, and in the practice of analysis, decomposition formulas obtained by applying the remainder theorem when integrating in expression (6.20) are more often used, one of which is for multiple image poles p_i

$$Y(p)=\frac{C(p)}{D(p)}, \tag{6.21}$$

where $C(p)$, $D(p)$ are polynomials in powers p of orders m and n, accordingly, has the form

$$y(t)=\sum_{i=1}^{n}\frac{C(p_i)}{D'(p_i)}e^{p_i t}\,. \tag{6.22}$$

Recall that p_i these are the roots of the equation

$$D(p)=0, \tag{6.23}$$

a which are called the poles of the expression (6.21), and

$$D'(p_i)=\left.\frac{dD}{dp}\right|_{p=p_i}\,. \tag{6.24}$$

We will give an example of the use of the decomposition formula (6.22).
Let us have an image of an unknown original in the form

$$F(p)=\frac{2p+1}{p^3+5p^2+6p}. \tag{6.25}$$

It is necessary to determine its original $f(t)$.

It is obvious that for our example

$$\begin{aligned} &C(p)=2p+1,\\ &D(p)=p^3+5p^2+6p,\\ &\frac{dD}{dp}=3p^2+10p+6. \end{aligned} \tag{6.26}$$

Let's find the poles of the image (6.25), that is, the roots of the equation

$$p^3+5p^2+6p=0\,. \tag{6.27}$$

Bringing equation (6.27) to the form

$$p\left(p^2+5p+6\right)=0, \tag{6.28}$$

it is easy to see that the poles of the image (6.25) are

$$\begin{cases} p_1=0,\\ p_2=\text{-}2,\\ p_3=\text{-}3. \end{cases} \tag{6.29}$$

Substituting (6.26) and (6.29) into the expansion formula (6.22), we obtain

$$\begin{aligned} f(t)&=\frac{C(p_1)}{D'(p_1)}e^{p_1 t}+\frac{C(p_2)}{D'(p_2)}e^{p_2 t}+\frac{C(p_3)}{D'(p_3)}e^{p_3 t}=\\ &=\frac{2\cdot(0)+1}{3\cdot(0)^2+10\cdot(0)+6}e^{0\cdot t}+\frac{2\cdot(\text{-}2)+1}{3\cdot(\text{-}2)^2+10\cdot(\text{-}2)+6}e^{\text{-}2t}+\\ &+\frac{2\cdot(\text{-}3)+1}{3\cdot(\text{-}3)^2+10\cdot(\text{-}3)+6}e^{\text{-}3t}=\frac{1}{6}+\frac{3}{2}e^{\text{-}2t}-\frac{5}{3}e^{\text{-}3t}. \end{aligned} \tag{6.30}$$

Function

$$f(t)=\frac{1}{6}+\frac{3}{2}e^{\text{-}2t}-\frac{5}{3}e^{\text{-}3t} \tag{6.31}$$

and is the original image $F(p)$ given by expression (6.25). We remind you once again that the original is defined only for the values of $t \geq 0$.

Note that if among the poles of the image (6.21), or, that is the same, among the roots of the equation (6.23) of the n-th order there is a multiple root, for example p_1, of multiplicity *k*, i.e., when the equation (6.23) takes the form

$$(p - p_1)^k D_{n-k}(p) = 0, \tag{6.32}$$

then instead of the expansion formula in the form (6.22) we will have the expansion formula in the form

$$y(t) = \frac{1}{(k-1)!} \frac{d^{(k-1)}}{dp^{(k-1)}} \left[\frac{C(p)(p - p_1)^k e^{pt}}{D(p)} \right]_{p=p_1} + \sum_{i=2}^{n-1} \frac{C(p_i)}{D'(p_i)} e^{p_i t} \tag{6.33}$$

or

$$y(t) = \frac{1}{(k-1)!} \frac{d^{(k-1)}}{dp^{(k-1)}} \left[\frac{C(p)e^{pt}}{D_{n-k}(p)} \right]_{p=p_1} + \sum_{i=2}^{n-1} \frac{C(p_i)}{D'(p_i)} e^{p_i t} \tag{6.34}$$

Let's relate the transfer function $W(p)$ of a linear dynamic system with its transient $h(t)$ and impulse transient $g(t)$ characteristics.

The transient characteristic $h(t)$ of the system is its response to the input signal $x(t)$ in the form of a single jump. That is, in case when $x(t) = 1(t)$, we have $y(t) = h(t)$.

The graphic interpretation of this definition is shown in fig. 12.

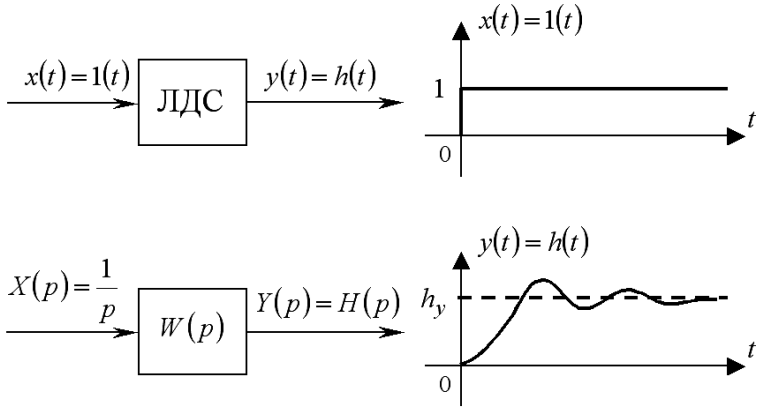


Figure 12 – Response graph h(t) of a linear dynamic system per unit jump 1(t)

The impulse transient or weight characteristic $g(t)$ of the system is its response to a single impulse input signal $x(t)$ in the form of a delta function $\delta(t)$, for which the following is true:

$$\delta(t) = \begin{cases} \infty, & t = 0, \\ 0, & t \neq 0, \end{cases} \tag{6.35}$$

$$\int_{-\infty}^{\infty} \delta(t)\,dt = \int_{0}^{\infty} \delta(t)\,dt = 1. \tag{6.36}$$

It follows from the expressions (6.35), (6.36) that the delta-function is an idealization of the pulse of a unit area with an extremely high height and an extremely short length (Fig. 13).

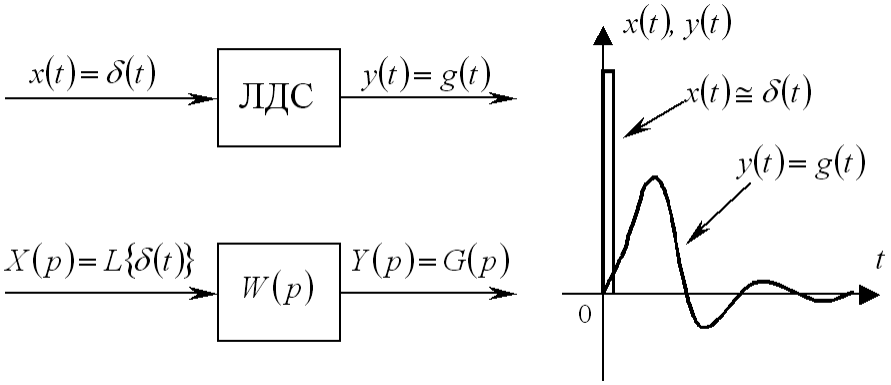


Figure 13 – Response graph g(t) of the linear dynamic system of LDS per unit impulse δ(t)

It is very important that the signal $x(t)$ that acts on the input of a linear dynamic system with an impulse transient characteristic $g(t)$ (Fig. 14) and the response of the system $y(t)$ to this signal are related by the convolution integral

$$y(t)=\int_0^{\infty} x(t-\tau)\,g(\tau)\,d\tau\,, \tag{6.37}$$

which belongs to the class of integral Fredholm equations, which we considered earlier, and which has an extremely transparent meaning - the output signal of a dynamic system is formed by the sum of reactions to each pulse of the input signal during the presentation of this input signal in the form of a sequence of pulses with a height equal to the value of the input signal in appropriate moment in time.

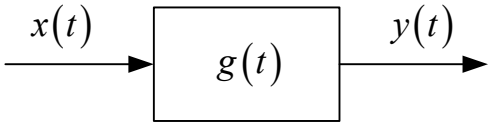


Figure 14 – Generalized structural diagram of a linear dynamic system

Since, according to relation (6.17), the Laplace image $Y(p)$ of the output signal of a linear dynamic system is the product of the transfer function $W(p)$ of this system and the Laplace image $X(p)$ of the input signal of the system, which in case of fulfillment of (6.3), according to (8.5), will be equal to $\frac{1}{p}$, then for the image according to Laplace $H(p)$, we will have the transition characteristics

$$H(p)=\frac{W(p)}{p}, \tag{6.38}$$

which, in turn, gives us the right to record

$$h(t)=L^{-1}\left\{\frac{W(p)}{p}\right\}. \tag{6.39}$$

It is easy to see that the Laplace imae of the delta function is equal to

$$L\{\delta(t)\} = \int_0^\infty \delta(t) e^{-pt} dt = e^{-p\cdot(t=0)} \int_0^\infty \delta(t) dt = e^{-p\cdot 0} \cdot 1 = 1. \tag{6.40}$$

So, if

$$x(t) = \delta(t), \tag{6.41}$$

by definition

$$y(t) = g(t), \tag{6.42}$$

then it follows from (6.17) that

$$G(p) = W(p). \tag{6.43}$$

That is, the transfer function of a linear dynamic system is the Laplace image of its impulse transient characteristic, and vice versa

$$g(t) = L^{-1}\{W(p)\}. \tag{6.44}$$

It follows from relations (6.38) and (6.43) that

$$G(p) = p \cdot H(p). \tag{6.45}$$

And this, in turn, means that the equation is valid in the field of originals

$$g(t) = \frac{dh(t)}{dt}, \tag{6.46}$$

that is, that the impulse transient characteristic $g(t)$ of the system can be obtained by differentiating its transient characteristic $h(t)$.

Summarizing the above, it can be stated that the mathematical model of LDS in the form of a transfer function $W(p)$ can be determined by dividing the Laplace-transformed response of the system $y(t)$ by the Laplace-transformed input signal $x(t)$. It is quite obvious that before the Laplace transformation, both the experimentally recorded input signal $x(t)$ and the experimentally recorded response of the system $y(t)$ to this signal must be approximated by the appropriate functions of the argument t. According to the Weierstrass theorem, this can almost always be done with the help of polynomials in powers of the argument t, the Laplace transformation of which leads to the ratio of polynomials in powers of the argument p. It is also obvious that the simplest task of identifying such a system will be solved by this algorithm if the input signal of the dynamic system is a single jump $1(t)$ or a single pulse $\delta(t)$, from the Laplace-transformed responses of the system to each of which we immediately obtain a transfer function according to the relations (6.38) or (6.43).

At the end of this subsection, we will show how to construct a mathematical model of this system in the form of a differential equation after obtaining the LDS transfer function $W(p)$.

Let the mathematical model of the LDS on the complex plane have the form

$$\cdot\ W(p) = \frac{b_m p^m + b_{m-1} p^{m-1} + \ldots + b_1 p + b_0}{a_n p^n + a_{n-1} p^{n-1} + \ldots + a_1 p + a_0}. \tag{6.47}$$

Substituting the value $W(p)$ from expression (6.47) into expression (6.17) and moving the denominator to the left side of the equality, we obtain

$$\begin{aligned}&\left(a_n p^n + a_{n-1}p^{n-1} + ... + a_1 p + a_0\right)\cdot Y(p) = \\ &= \left(b_m p^m + b_{m-1}p^{m-1} + ... + b_1 p + b_0\right)\cdot X(p).\end{aligned} \tag{6.48}$$

Opening the brackets in equality (6.48) and taking into account that the multiplication of the image by p corresponds to the differentiation of the original with respect to t, we arrive at a differential equation of the ***n***-th order

$$\sum_{i=0}^{n} a_i \frac{d^i y}{dt^i} = \sum_{l=0}^{m} b_l \frac{d^l x}{dt^l}, \tag{6.49}$$

for which the condition is fulfilled

$$m \le n, \tag{6.50}$$

which is due to the ability of the system to be physically implemented under such a model.

It is quite obvious that in order to obtain a solution of a differential equation of the *n*-th order, it must be integrated *n*-once, which causes the appearance n of constant integrations, for the specification of which it is necessary to know at the initial moment of time not only the value of the initial coordinate $y(t)$, but also the value at this initial moment of all its derivatives up to $(n-1)$ order inclusive, that is, the initial conditions for equation (6.49) have the form:

$$\left.\begin{aligned}&y(t)\big|_{t=0} = y(0),\\ &\left.\frac{dy}{dt}\right|_{t=0} = y^1(0),\\ &\cdots\cdots\cdots\cdots,\\ &\left.\frac{d^{n-1}y}{dt^{n-1}}\right|_{t=0} = y^{n-1}(0).\end{aligned}\right\} \tag{6.51}$$

It should be noted that a significant number of LDSs is characterized by the fact that all elements of their structures, which are able to store energy, lose this energy after the system is turned off, which gives reason to consider the initial conditions for the model in the form of (6.49) before the system is restarted zero, that is, in the system of equations (6.51), consider all right-hand sides to be equal to zero. This immediately leads to the advantages of solving this differential equation due to its transformation into a complex plane using the Laplace operator with the subsequent application to the obtained image of the inverse of the Laplace operator in the form of one of the forms of the decomposition theorem.

Summarizing all of the above, we can state that the direct Laplace operator implements the law of mapping a set of continuous functions of a real argument, given on the number axis, into a set of continuous functions of a complex argument, given on a complex plane, the coordinates of the points of which are also real numbers. And the inverse Laplace operator works on the contrary, implementing the inverse process of transforming the specified sets into one another.

And we will conclude the consideration of direct and inverse Laplace operators by referring to the fact that in functional analysis and related sections of mathematics, the theory and practice of applying these operators carries operational calculus.

6.2 Autoregressive operators in time series display problems

During the systematic analysis of dynamic processes that have a random nature and the creation of information technologies suitable for the implementation of this analysis, a significant role is assigned to the prediction of the development of these processes over time.

To date, the most effective mathematical models that can be used to predict the development of processes are those that use time series during their construction.

We remind **that a time series is a set of values of a random process taken at equal time intervals t. Let us denote this set by the symbol z_t.**

In fact, z_t this is a random process discrete in time.

The task of forecasting is that, knowing the value of the process at the moment t, it is necessary to forecast its value at the moment $t+l$, where l is the bias time. To distinguish the forecast value of the process from the actual value, the actual value of the time series at the moment $t+l$ is denoted by the symbol z_{t+l}, and the forecast value by the symbol $z_t(l)$.

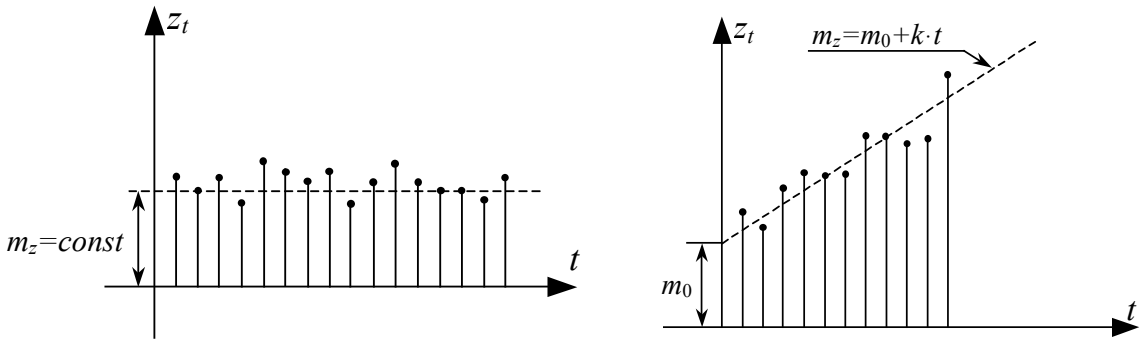
It is clear that it is in principle impossible to accurately predict the value of a random process, which is a time series, and therefore the forecast is carried out by achieving the minimum of some functional chosen as a criterion for the adequacy of the forecast model.

If the value is small t (1, 2 steps), then one of such criteria can be the variance of the deviation $z_t(l)$ from z_{t+l}, which should be minimal for the optimal forecast model, i.e.

$$E\left\{\left(z_{t+l}-z_t(l)\right)^2\right\} \rightarrow \min, \tag{6.52}$$

where E is the symbol of the mathematical waiting operation.

Like any other random process, the time series z_t can be stationary (Fig. 15, a) or non-stationary (Fig. 15, b).



a) b)

Figure 15 – Graphs of realization of stationary (a) and non-stationary (b) time series

A stationary time series is characterized by the equilibrium of its values z_t near the average value m_z, which is a constant, as shown in Fig. 15, a.

For a non-stationary time series, the moving average value $m_z(t)$ of the process is a function of time t, as shown in Fig. 15, b.

We will introduce a number of useful operators that will be needed later.

1. **The operator B shifts back by one time unit**

$$z_{t-1} = Bz_t . \tag{6.53}$$

It is clear that, according to the expression (6.53), the expression is also valid

$$z_{t-2} = Bz_{t-1} . \tag{6.54}$$

Substituting the values z_{t-1} from expression (6.53) into (6.54), we obtain

$$z_{t-2} = B\left(Bz_t\right) = B^2 z_t . \tag{6.55}$$

Generalizing the expression (6.55), we have

$$z_{t-m} = B^m z_t . \tag{6.56}$$

2. **The forward shift operator *F* by one time unit**

$$z_{t+1} = Fz_t . \tag{6.57}$$

It is clear that, according to the expression (6.57), the expression is also valid

$$z_{t+2} = Fz_{t+1} . \tag{6.58}$$

Substituting the values z_{t+1} from expression (6.57) into (6.58), we obtain

$$z_{t+2} = F\left(Fz_t\right) = F^2 z_t . \tag{6.59}$$

Generalizing the expression (6.59), we have

$$z_{t+m} = F^m z_t . \tag{6.60}$$

3. **Difference operator ∇ with a shift back by one time unit**

$$\nabla z_t = z_t - z_{t-1} . \tag{6.61}$$

Substituting the values from expression (6.53) into (6.61), we get

$$\nabla z_t = z_t - z_{t-1} = z_t - Bz_t = \left(1 - B\right) z_t . \tag{6.62}$$

It follows from the expression (6.62) that

$$\nabla = 1 - B . \tag{6.63}$$

4. **Difference operator Δ with forward shift by one time unit**

$$\Delta z_t = z_{t+1} - z_t . \tag{6.64}$$

Substituting the values z_{t+1} from expression (6.57) into (6.64), we obtain

$$\Delta z_t = Fz_t - z_t = \left(F - 1\right) z_t . \tag{6.65}$$

It follows from the expression (6.65) that

$$\Delta = F - 1 . \tag{6.66}$$

5. **Sum operator *S***

$$Sz_t = z_t + z_{t-1} + z_{t-2} + \ \ldots \ = \sum_{j=0}^{\infty} z_{t-j} . \tag{6.67}$$

Substituting the values z_{t-m} from expression (6.56) into (6.67), we get

$$Sz_t = \left(1 + B + B^2 + \ \ldots\right) z_t = \frac{1}{1-B} z_t = \left(1 - B\right)^{-1} z_t . \tag{6.68}$$

We draw attention to the fact that during the derivation of relation (6.68) we used the formula for the sum of the terms of an infinitely decreasing geometric progression with the denominator ***B***, which, under the condition of considering it as a number and the condition of convergence of the series (6.67), must be less than unity.

It follows from relations (6.63) and (6.68) that

$$S\, z_t = \nabla^{-1} z_t , \tag{6.69}$$

or

$$S = \nabla^{-1} . \tag{6.70}$$

Therefore, the sum operator is the inverse of the difference operator with a backward shift.

Next, we remind that **a sequence of uncorrelated and normally distributed random pulses a_t with zero mean and variance**

$$\sigma_a^2 = const \tag{6.71}$$

called discrete white noise.

Let's try to use white noise pulses a_t to build a time series z_t model in the following way

$$z_t = \mu + a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \ \ldots \ , \tag{6.72}$$

where μ is the reference level (average value) of the time series z_t, and ψ_k, $k = 1, 2, \ldots$ are the weight coefficients of the white noise pulses with which they are included in the sum (6.72).

Let's perform the operation of centering the time series z_t by subtracting the average value μ.

For a centered time series

$$\tilde{z}_t = z_t - \mu . \tag{6.73}$$

From the expression (6.72), we obtain

$$\tilde{z}_t = a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \ \ldots \ . \tag{6.74}$$

Using relation (6.56) for pulses a_{t-m}, from (6.74) we have

$$\tilde{z}_t = a_t + \psi_1 B a_t + \psi_2 B^2 a_t + \ \ldots = \left(1 + \psi_1 B + \psi_2 B^2 + \ \ldots\right) \cdot a_t . \tag{6.75}$$

Let's mark

$$\Psi\left(B\right) = 1 + \psi_1 B + \psi_2 B^2 + \ \ldots \ . \tag{6.76}$$

Taking into account (6.76), the relation (6.75) can be written as follows

$$\tilde{z}_t = \Psi\left(B\right) a_t . \tag{6.77}$$

The expression $\Psi\left(B\right)$ in the form (6.76) is a filter operator that transforms a sequence of white noise pulses a_t into a time series with given properties (Fig. 16), i.e., matches a discrete stochastic function from one zero-dimensional set with another discrete stochastic function from another zero-dimensional set.

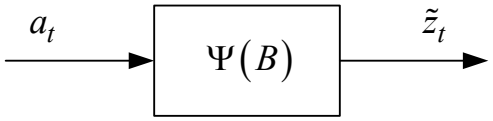


Figure 16 – Block diagram of a linear filter

The coefficients ψ_k, $k = 1, 2, \ldots$ of the filter operator are selected in the procedure of minimizing the criterion (6.52) at $l = 0$.

In the model of the linear filter (6.72), the values of the time series z_t are determined by the weighted sum of the current and previous pulses of white noise a_t.

A characteristic feature of the filter operator $\Psi(B)$ given by expression (6.76) is that it theoretically has an infinite number of members, which creates certain inconveniences in case of its practical use.

Therefore, the proposal to build a time series model z_t based on a finite set of power-q weighted pulses of white noise a_t in the form

$$\tilde{z}_t = \mathrm{a_t} - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ \ldots - \theta_q a_{t-\dot{q}} \tag{6.78}$$

Since the ratio (6.78) uses q of the previous values of white noise a_{t-i}, $i = \overline{1, q}$, which are weightedly subtracted from the current pulse a_t, this ratio actually specifies a "moving average" that "shifts" along the sequence a_t with growth t, keeping the same number of members during the "shift".

Applying the ideology of relation (6.56) to impulses a_{t-m}, from expression (6.78) we obtain

$$\tilde{z}_t = \mathrm{a_t} - \theta_1 B a_t - \theta_2 B^2 a_t - \ \ldots - \theta_q B^q a_t, \tag{6.79}$$

or

$$\tilde{z}_t = \Theta(B) a_t, \tag{6.80}$$

where $\Theta(B)$ is the moving average operator –

$$\Theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ \ldots - \theta_q B^q \tag{6.81}$$

of order q, which is actually also a filter operator, but with a limited number of components, that is, a "shortened" filter operator.

The ratio (6.80) defines a model of a stationary time series z_t, which uses the moving average operator (6.81), and therefore in mathematics it is agreed to call this model the model of the moving average order q (abbreviated: the MA(q) model).

We will show how the filter operator is related to the autoregression operator.

From a philosophical point of view, the regression model is a model "looking back, towards where it came from"; that is, it is a model that sets the value of some process coordinate at a given moment of time based on its independent components determined at a previous moment. The number of components taken into account determines the regression order.

Based on this interpretation, the autoregression model is a model that sets the value of some process coordinate at a given time based on its previous values. The number of taken into account previous values determines the order of autoregression.

For a centered time series $\tilde{z}_t$, the order p autoregression model (abbreviated: AR(p)) can be written as

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \ \ldots + \phi_p \tilde{z}_{t-p} + a_t, \tag{6.82}$$

where a_t is the white noise pulse, the definition of which is given above.

Taking into account relation (6.56), expression (6.82) can be rewritten as follows

$$\tilde{z}_t - \phi_1 B\tilde{z}_t - \phi_2 B^2 \tilde{z}_t - \ldots - \phi_p B^p \tilde{z}_t = a_t,$$

or

$$\Phi(B)\tilde{z}_t = a_t, \tag{6.83}$$

where $\Phi(B)$ is the autoregression operator of order , which has the form

$$\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p \ . \tag{6.84}$$

Let's use the identity further

$$\Phi(B)\Phi^{-1}(B) = \Phi^{-1}(B)\Phi(B) = I, \tag{6.85}$$

in which I is the unit operator; and the operator $\Phi^{-1}(B)$ is the inverse of the operator $\Phi(B)$.

Multiplying by $\Phi^{-1}(B)$ the left side of equation (6.83), we get

$$\Phi^{-1}(B)\Phi(B)\tilde{z}_t = \Phi^{-1}(B)a_t, \tag{6.86}$$

or (taking into account (6.85)) -

$$I\tilde{z}_t = \Phi^{-1}(B)a_t. \tag{6.87}$$

Since multiplication by the unit operator does not change the result, expression (6.87) can be written as follows

$$\tilde{z}_t = \Phi^{-1}(B)a_t. \tag{6.88}$$

Comparing the expression (6.88) with (6.77), it can be stated that

$$\Psi(B) = \Phi^{-1}(B). \tag{6.89}$$

So, by synthesizing the autoregression operator $\Phi(B)$ based on the realization of the studied time series z_t , which is easy to do, as will be shown below, and defining the operator $\Phi^{-1}(B)$ inverse to $\Phi(B)$, which is also quite simple, we simultaneously define the linear filter operator $\Psi(B)$, which forms a time series z_t from white noise a_t with given properties.

We pay attention to the fact that in this case $\Psi(B)$ the criterion (6.52) is not minimized when $l = 0$, which was discussed above.

When solving the problem of identifying a time series z_t model based on the order p autoregression operator, it is necessary to determine $p+2$ the unknowns, which are the coefficients $\phi_1, \phi_2, \ldots, \phi_p$ of the operator $\Phi(B)$, the average value μ of the process z_t and the variance σ_a^2 of white noise a_t.

We will talk about how to solve this problem after we define the concepts of utocovariance and autocorrelation of a time series.

The autocovariance γ_k of a time series z_t with a delay k is called an expression

$$\gamma_k = \operatorname{cov}\{z_t, z_{t+k}\} = E\{(z_t - \mu)\cdot(z_{t+k} - \mu)\}, \tag{6.90}$$

in which E is the symbol for calculating the mathematical expectation from the expression in curly brackets.

It is clear that

$$\gamma_0 = E\left\{\left(z_t - \mu\right)^2\right\} = \sigma_z^2 \tag{6.91}$$

is the variance of the time series z_t.

To obtain a statistical estimate γ_k^* of the autocovariance γ_k defined by the expression (6.90), the expression is used

$$\gamma_k^* = \frac{1}{N-k}\sum_{t=1}^{N-k}\left(z_t - \mu\right)\cdot\left(z_{t+k} - \mu\right)\cdot \tag{6.92}$$

Autocovariance γ_k characterizes the degree of linear relationship between the values of the time series z_t and z_{t+k}.

It is clear that

$$\begin{cases} \left|\gamma_k\right| \le \gamma_0, \\ \gamma_k = \gamma_{-k}. \end{cases} \tag{6.93}$$

The autocorrelation ρ_k of a time series z_t with a delay k is called an expression

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{E\left\{\left(z_t - \mu\right)\cdot\left(z_{t+k} - \mu\right)\right\}}{E\left\{\left(z_t - \mu\right)^2\right\}}. \tag{6.94}$$

The following relations are valid for arbitrary autocorrelation ρ_k:

$$\begin{cases} \left|\rho_k\right| \le \rho_0, \\ \rho_0 = 1, \\ \rho_k = \rho_{-k}. \end{cases} \tag{6.95}$$

The entire possible population γ_k is the autocovariance function of the time series z_t. It belongs to the class of lattice functions. Similarly, the set of all values ρ_k defines the autocorrelation function of the time series z_t.

An example of the graph of the autocorrelation function ρ_k is shown in Fig. 17.

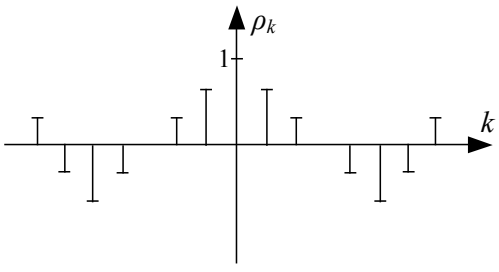


Figure 17 – One of the possible graphs of the autocorrelation function ρ_k, $k = \overline{-N, N}$

And now let's return to the already formulated task of determining the coefficients of the autoregression operator, that is, we will get an answer to the question: "**How to determine the coefficients of the autoregression of the time series model in the form of AR$\left(p\right)$?".**

We will show that the answer to this question is provided by the Yule-Walker equations.

To synthesize them, first multiply the expression (6.82) by $\tilde{z}_{t-k}$. As a result, we get

$$\tilde{z}_{t-k}\tilde{z}_t = \phi_1\tilde{z}_{t-k}\tilde{z}_{t-1} + \phi_2\tilde{z}_{t-k}\tilde{z}_{t-2} + \ \ldots + \phi_p\tilde{z}_{t-k}\tilde{z}_{t-p} + \tilde{z}_{t-k}a_t . \tag{6.96}$$

Let's replace the discrete variable in expression (6.96) by putting

$$t - k = \lambda . \tag{6.97}$$

We will get

$$\tilde{z}_\lambda\tilde{z}_{\lambda+k} = \phi_1\tilde{z}_\lambda\tilde{z}_{\lambda+k-1} + \phi_2\tilde{z}_\lambda\tilde{z}_{\lambda+k-2} + \ \ldots + \phi_p\tilde{z}_\lambda\tilde{z}_{\lambda+k-p} + \tilde{z}_\lambda a_{\lambda+k} . \tag{6.98}$$

Let's find the mathematical expectation from both parts of equation (6.98). We will get

$$\begin{aligned} E\{\tilde{z}_\lambda\tilde{z}_{\lambda+k}\} = \phi_1 E\{\tilde{z}_\lambda\tilde{z}_{\lambda+k-1}\} + \phi_2 E\{\tilde{z}_\lambda\tilde{z}_{\lambda+k-2}\} + \ \ldots + \\ + \phi_p E\{\tilde{z}_\lambda\tilde{z}_{\lambda+k-p}\} + E\{\tilde{z}_\lambda a_{\lambda+k}\}. \end{aligned} \tag{6.99}$$

Considering the expression (6.90), from the expression (6.99) we have

$$\gamma_k = \phi_1\gamma_{k-1} + \phi_2\gamma_{k-2} + \ \ldots + \phi_p\gamma_{k-p} \tag{6.100}$$

for all k from 1 to p.

But with $k = 0$, taking into account the expression (6.93), we get another equatio

$$\gamma_0 = \phi_1\gamma_1 + \phi_2\gamma_2 + \ \ldots + \phi_p\gamma_p + \sigma_a^2 . \tag{6.101}$$

The absence of mathematical expectation $E\{\tilde{z}_\lambda \cdot a_{\lambda+k}\}$ calculation results in equations (6.100) and their presence in equation (6.101) in the form of white noise dispersion σ_a^2 is explained by the fact that, according to the properties of white noise, each of its impulses is correlated (interrelated) only with itself and not at all correlated with no other, even placed in time next to it. So

$$E\{\tilde{z}_\lambda \cdot a_{\lambda+k}\} = \begin{cases} 0, & npu \quad k \neq 0 \\ \sigma_a^2, & npu \quad k = 0 \end{cases}. \tag{6.102}$$

From equation (6.101), taking into account expression (6.91), we have

$$\sigma_a^2 = \sigma_z^2 - \phi_1\gamma_1 - \phi_2\gamma_2 - \ \ldots - \phi_p\gamma_p . \tag{6.103}$$

So, if for the implementation of a stationary time series z_t of length *N* the estimated average value μ and variance σ_z^2 have already been calculated according to known formulas

$$\mu = \frac{1}{N}\sum_{t=1}^{N} z_t , \tag{6.104}$$

$$\sigma_z^2 = \frac{1}{N-1}\sum_{t=1}^{N}(z_t - \mu)^2 , \tag{6.105}$$

the coefficients $\phi_1, \phi_2, \ldots, \phi_p$ are somehow found and calculated according to the expression (6.92) of the autocovariance $\gamma_1, \gamma_2, \ldots, \gamma_p$, then it is not difficult to find the dispersion of white noise σ_a^2 according to the expression (6.103).

Before giving the algorithm for obtaining the numerical values of the coefficients $\phi_1, \phi_2, \ldots, \phi_p$, let's divide all the equations of the system (6.100) by γ_0.

As a result, we will get a system of equations no longer relative to autocovariances γ_k, $k = \overline{1, p}$, but relative to autocorrelations ρ_k, $k = \overline{1, p}$, that is, a system

$$\begin{cases} \phi_1\rho_0 + \phi_2\rho_1 + \phi_3\rho_2 + \ldots + \phi_p\rho_{p-1} = \rho_1, \\ \phi_1\rho_1 + \phi_2\rho_0 + \phi_3\rho_1 + \ldots + \phi_p\rho_{p-2} = \rho_2, \\ \phi_1\rho_2 + \phi_2\rho_1 + \phi_3\rho_0 + \ldots + \phi_p\rho_{p-3} = \rho_3, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \phi_1\rho_{p-1} + \phi_2\rho_{p-2} + \phi_3\rho_{p-3} + \ldots + \phi_p\rho_0 = \rho_p, \end{cases} \tag{6.106}$$

or, in matrix form,

$$M \cdot \Phi = \rho, \tag{6.107}$$

where the matrices $M(p \times p)$, $\Phi(p \times 1)$, $\rho(p \times 1)$ have the form

$$M = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{p-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{p-2} \\ \rho_2 & \rho_1 & 1 & \cdots & \rho_{p-3} \\ . & . & . & . & . \\ \rho_{p-1} & \rho_{p-2} & \rho_{p-3} & \cdots & 1 \end{bmatrix},$$

$$\Phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_p \end{bmatrix}, \qquad \rho = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_p \end{bmatrix}.$$

In equations (6.107), the values of the coefficients $\phi_1, \phi_2, \ldots, \phi_p$ are unknown.

To solve the system (6.107), we first define the matrix M^{-1}, which is the inverse of the matrix M. Then we multiply the matrix equation (6.107) on the left by M^{-1}. As a result, we get

$$M^{-1}M \cdot \Phi = M^{-1}\rho, \tag{6.108}$$

or

$$I \cdot \Phi = M^{-1}\rho, \tag{6.109}$$

and finally

$$\Phi = M^{-1}\rho. \tag{6.110}$$

Equations (6.106), (6.107) are called Yule-Walker equations. Their solution in the form of (6.110) makes it possible to determine the vector $\Phi(p \times 1)$ of coefficients $\phi_1, \phi_2, \ldots, \phi_p$ of the autoregression operator in the AR(p) model based on previously calculated autocorrelations ρ_k, $k = \overline{1, p}$ of the time series z_t.

In the time series z_t model based on order *p* autoregression, only one current pulse of white noise a_t is involved in the formation of the current value of the series. It is natural to assume that if we subtract the weighted sum *q* of previous values of white noise from this impulse a_t, we will get a model that will take into account more "subtle" moments of the random process and will more adequately reflect its properties, since in addition to autoregression, this model will also take into account the moving average of the process.

Such a time series z_t model is called a moving average autoregression model (abbreviated: ARMA (p, q) model) and has the form

$$\begin{aligned}\tilde{z}_t &= \phi_1\tilde{z}_{t-1} + \phi_2\tilde{z}_{t-2} + \ \ldots + \phi_p\tilde{z}_{t-p} + \\ &+ \mathrm{a}_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ \ldots - \theta_q a_{t-q}.\end{aligned} \tag{6.111}$$

By transferring all the members from $\tilde{z}_{t-i},\ i=\overline{1, p}$ to the left side of equation (6.111) and performing already known transformations, we obtain the equation

$$\Phi(B)\tilde{z}_t = \Theta(B)a_t, \tag{6.112}$$

in which the operators $\Phi(B)$ and $\Theta(B)$ are determined by expressions (6.81), (6.84).

Equation (6.112) is the basic form of the time series z_t model based on ARMA(p, q).

To identify this model, we need to determine $p+q+2$ the unknowns, which are the coefficients $\phi_i,\ i=\overline{1, p}$ of the operator $\Phi(B)$, the coefficients $\theta_j,\ j=\overline{1, q}$ of the operator $\Theta(B)$, the mean μ of the process z_t and the variance σ_a^2 of the white noise a_t.

All time series models constructed above were based on the condition of stationarity of these series. But in everyday life we constantly encounter non-stationary random processes. For example, these are the processes of starting or braking any technological equipment that implements a technological process of a stochastic nature.

We will show that such non-stationary random processes, which, when discretized, turn into time series, can be adequately described using a model in which autoregression operators - an integrated moving average - are embedded.

For their synthesis, let us assume that in the ARMA(p, q) model given by expression (6.112), the operator $\Phi(B)$ has d multiple roots equal to unity.

In this case, according to Viett's theorem, the operator $\Phi(B)$ can be written in the form

$$\Phi(B) = (1-B)^d \cdot \left(1 - \phi_1^* B - \phi_2^* B^2 - \ \ldots - \phi_l^* B^l\right), \tag{6.113}$$

where

$$d + l = p. \tag{6.114}$$

Let's mark

$$\Phi^*(B) = 1 - \phi_1^* B - \phi_2^* B^2 - \ \ldots - \phi_l^* B^l. \tag{6.115}$$

Taking into account expressions (6.113) and (6.115), equation (6.112) can be rewritten as

$$\Phi^*(B)\cdot(1-B)^d\,\tilde{z}_t = \Theta(B)a_t. \tag{6.116}$$

Since, according to the expression (6.63)

$$(1-B)^d = \nabla^d, \tag{6.117}$$

i.e., $(1-B)^d$ is a difference operator with backward shift of order d, then

$$(1-B)^d z_t = \nabla^d z_t \tag{6.118}$$

defines a new variable w_t, which is associated with z_t the relation

$$w_t = \nabla^d z_t . \tag{6.119}$$

Substituting expression (6.119) into equation (6.116), we get

$$\Phi^*(B) w_t = \Theta(B) a_t \tag{6.120}$$

It is obvious that the expression (6.120) defines the ARMA (l, q) model with respect to w_t, which can be rewritten as

$$\begin{aligned} w_t = \phi_1^* w_{t-1} + \phi_2^* w_{t-2} + \ \ldots + \phi_l^* w_{t-l} + \\ + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ \ldots - \theta_q a_{t-q}. \end{aligned} \tag{6.121}$$

Equations (6.119), (6.120) specify the model of a non-stationary time series z_t in the form of autoregression – integrated moving average order (l, q, d). Abbreviated: model ARIMA(l, q, d).

We draw attention to the fact that the first difference ∇z_t of values of any non-stationary time series z_t has a lower degree of non-stationarity than the time series z_t itself. The second difference $\nabla^2 z_t$, which is the difference of the first differences $\nabla(\nabla z_t)$ of this time series z_t, will have an even smaller degree of non-stationarity.

Increasing the order d of the difference $\nabla^d z_t$, sooner or later we will reach its value w_t, which will already be a relatively stationary time series w_t. In fig. 18 provides a graphic interpretation of this fact.

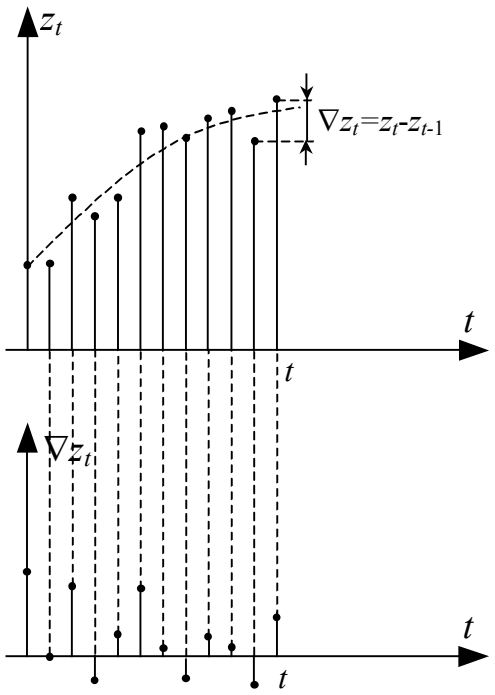


Figure 18 – Graphical interpretation transformation of the non-stationary time series z_t into a stationary time series for its difference ∇z_t

It is clear that from the ARIMA (l, q, d) model (6.119), (6.120) we get the ARMA (p, q) model, if $d = 0$.

We will explain why the name of the ARIMA (l, q, d) model contains the word "integrated" in relation to the moving average.

We remind that the inverse operator for ∇ is the sum operator S (6.70). Therefore, having obtained w_t from equation (6.120), in order to move to the time series z_t, it is necessary to *d* sum the coordinate w_t times, since, multiplying equation (6.119) on the left by ∇^{-d}, we have

$$\nabla^{-d} w_t = \nabla^{-d} \nabla^d z_t, \tag{6.122}$$

or

$$\nabla^{-d} w_t = I \cdot z_t, \tag{6.123}$$

from which, taking into account (6.70), we have

$$z_t = \nabla^{-d} w_t = S^d w_t. \tag{6.124}$$

It is clear that the most difficult task when using the ARIMA (l, q, d) model is to determine the numerical value of the integration parameter *d*, or, in other words, to determine the number of differences that must be successively taken from a non-stationary time series z_t in order to transform it into a stationary series relative to some difference of this series. It is obvious that it must be solved by substituting the value z_t obtained by expression (6.124) and the experimental value of this coordinate into the criterion functional (6.52) and searching for the minimum value of this functional in the approximate interpretation.

6.3 Examples of implementation of special operators

First, we consider an example of the application of the direct (6.1) and inverse (6.20) Laplace operators, which are widely used in the analysis of processes in linear dynamic systems.

Let the process in a linear dynamic system, at the input of which a signal $x(t)$ arrives and whose reaction to this signal is the output coordinate $y(t)$, be described by a differential equation

$$\frac{d^4 y}{dt^4} + 10\frac{d^3 y}{dt^3} + 35\frac{d^2 y}{dt^2} + 50\frac{dy}{dt} + 24y = 2\frac{dx}{dt} + x, \tag{6.125}$$

which is a mathematical model of this process in the time domain under zero initial conditions (6.51).

And let us find out what will be the nature of the response $y(t)$ of this dynamic system to the input signal

$$x(t) = 5e^{-t}. \tag{6.126}$$

As you know, under zero initial conditions, the differential equation (6.125) cannot be solved by the classical method, since the system of equations for determining the integration constants for each of the exponents will have no solutions other than zero. But this differential equation (6.125) is easily solved if, using the direct Laplace operator, it is transformed into a complex plane, i.e., if the transformation

$$L\{\frac{d^4y}{dt^4}\}+L\{10\frac{d^3y}{dt^3}\}+L\{35\frac{d^2y}{dt^2}\}+L\{50\frac{dy}{dt}\}+L\{24y\}=L\{2\frac{dx}{dt}\}+L\{x\},\qquad(6.127)$$

which is admissible since both the differential equation and the Laplace operator are linear. And then, applying the expression (6.10) and putting the operator equation (6.125) in brackets $Y(p)$ and $X(p)$, we get the form

$$(p^4+10p^3+35p^2+50p+24)Y(p)=(2p+1)X(p),\qquad(6.128)$$

which is easily transformed into an expression

$$Y(p)=\frac{2p+1}{p^4+10p^3+35p^2+50p+24}X(p).\qquad(6.129)$$

Laplace-transforming the signal (6.126) using expression (6.6) under the condition that $\alpha=1$, we have

$$X(p)=\frac{5}{p+1}.\qquad(6.130)$$

Substituting expression (6.130) into (6.129), we get

$$Y(p)=\frac{2p+1}{p^4+10p^3+35p^2+50p+24}\ \ \frac{5}{p+1}\qquad(6.131)$$

or

$$Y(p)=\frac{10p+5}{p^5+11p^4+45p^3+85p^2+74p+24}=\frac{C(p)}{D(p)}\qquad(6.132)$$

It is easy to see that the expression (6.132) is the Laplace-transformed response of our dynamic system, defined on the complex plane, to the input signal (6.126). And therefore, in order to find this reaction in the time domain, that is, to determine $y(t)$ it is necessary to use an inverse Laplace operator (6.132), which is also linear and therefore the expression will be fair

$$L^{-1}\{Y(p)\}=L^{-1}\{\frac{10p+5}{p^5+11p^4+45p^3+85p^2+74p+24}\}=L^{-1}\{\frac{C(p)}{D(p)}\}=y(t)\qquad(6.133)$$

The inverse Laplace operator in the expression (6.133) will be applied in the form of the decomposition theorem (6.22) if all the poles of the expression (6.133), i.e., the roots of the equation (6.23), which for the expression (6.133) will have the form

$$p^5+11p^4+45p^3+85p^2+74p+24=0,\qquad(6.134)$$

will be different numbers (real or complex), or in the form (6.34), if there are multiples among the roots of equation (6.134).

So the next step in the algorithm for applying the inverse Laplace operator is to determine the poles of the expression (6.133), or, which is the same thing, to determine the roots of the equation (6.134). It is quite obvious that in order to solve this equation of the 5th order, it is necessary to apply a suitable program in some application program package, for example, in PPP MathCAD or MALAB or Python.

Applying one of these packages, we find that the roots of equation (6.134) will be: multiple root

$$p_1=-1\qquad(6.135)$$

with multiplicity of 2 and three simple roots

$$p_2 = -2, \quad p_3 = -3, \quad p_4 = -4 . \tag{6.136}$$

So, since we have among the roots of the equation (6.134) and one multiple, we will apply the inverse Laplace operator in the form (6.34), for which we need to first find a polynomial $D_{n-k}(p)$ which for our conditions

$$n = 5, \quad k = 2 \tag{6.137}$$

and, according to expression (6.32), will have the form

$$D_3(p) = \frac{p^5 + 11p^4 + 45p^3 + 85p^2 + 74p + 24}{(p+1)^2} = p^3 + 9p^2 + 26p + 24 . \tag{6.138}$$

In addition, according to expressions (6.24) and (6.233), we will have

$$D'(p) = \frac{dD}{dp} = 5p^4 + 44p^3 + 135p^2 + 170p + 74 . \tag{6.139}$$

Taking into account the expressions (6.135)–(6.139) for our example, the inverse Laplace operator in the form (6.34) takes the form

$$y(t) = \frac{d}{dp}\left[\frac{(10p+5)e^{pt}}{p^3 + 9p^2 + 26p + 24}\right]_{p=p_1} + \sum_{i=2}^{4} \frac{10p_i + 5}{5p_i^4 + 44p_i^3 + 135p_i^2 + 170p_i + 74} e^{p_i t} . \tag{6.140}$$

Taking the derivative and writing the sum, from the expression (6.140) we will have

$$\begin{aligned} y(t) = {} & \frac{\left[10e^{p_1 t} + (10p_1 + 5)te^{p_1 t}\right](p_1^3 + 9p_1^2 + 26p_1 + 24)}{(p_1^3 + 9p_1^2 + 26p_1 + 24)^2} - \frac{(10p_1 + 5)e^{p_1 t}(3p_1^2 + 18p_1 + 26)}{(p_1^3 + 9p_1^2 + 26p_1 + 24)^2} + \\ & + \frac{(10p_2 + 5)e^{p_2 t}}{5p_2^4 + 44p_2^3 + 135p_2^2 + 170p_2 + 74} + \\ & + \frac{(10p_3 + 5)e^{p_3 t}}{5p_3^4 + 44p_3^3 + 135p_3^2 + 170p_3 + 74} + \frac{(10p_4 + 5)e^{p_4 t}}{5p_4^4 + 44p_4^3 + 135p_4^2 + 170p_4 + 74} \end{aligned} \tag{6.141}$$

And by substituting the numerical values of the poles from the expressions (6.135), (6.136) into the expression (6.141) and performing the corresponding calculations, we obtain

$$y(t) = \frac{5}{36}(23 - 6t)e^{-t} - \frac{15}{2}e^{-2t} + \frac{25}{4}e^{-3t} - \frac{35}{18}e^{-4t} . \tag{6.142}$$

The second example that we consider is the example of solving the problem of synthesis of autoregression operator using experimentally defined values of the time series.

This example is borrowed from our same textbook on mathematical methods of identification of dynamic systems.

Therefore, let 10 values of the original coordinate of the object, recorded by us at the same intervals in the process of normal operation, were the following as shown in Table 2.

Table 2 - Table Experimentally defined values of the original coordinate of the object

t	1	2	3	4	5	6	7	8	9	10
y_t	20	50	40	20	30	50	10	40	50	20

Let's build a model of this time series, suitable for the forecast of the following values, based on the autoregressions of the 1st, 2nd and 3rd orders of magnitude that look like

$$z_t = \phi_1 z_{t-1} + a_t, \tag{6.143}$$

$$z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + a_t, \tag{6.144}$$

$$z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + \phi_3 z_{t-3} + a_{t,} \tag{6.145}$$

where

$$z_t = y_t - m_y = y_t - \mu \tag{6.146}$$

Before solving the problem of identification of these models and the choice of them, we calculate all the necessary parameters of the time row given by Table 2, using expressions (6.104), (6.105) and (6.92), (6.94).

So,

$$\mu = m_y = \frac{1}{10}\sum_{t=1}^{10} y_t = \frac{1}{10}(20+50+40+20+30+50+10+40+50+20) = 33, \tag{6.147}$$

$$\gamma_z[0] = \sigma_z^2 = \sigma_y^2 = \frac{1}{9}\sum_{t=1}^{10}(y_t - m_y)^2 = \frac{1}{9}[(-13)^2 + (17)^2 + (7)^2 + (-13)^2 + (-3)^2 + \\ + (17)^2 + (-23)^2 + (7)^2 + (17)^2 + (-13)^2] = 221, \tag{6.148}$$

$$\gamma_z[1] = \frac{1}{8}\sum_{t=1}^{9}(y_t - m_y)(y_{t+1} - m_y) = \frac{1}{8}[(-13)\bullet 17 + 17\bullet 7 + 7\bullet(-13) + \\ + (-13)\bullet(-3) + (-3)\bullet 17 + 17\bullet(-23) + (-23)\bullet 7 + 7\bullet 17 + 17\bullet(-13)] = -105, \tag{6.149}$$

$$\gamma_z[2] = \frac{1}{7}\sum_{t=1}^{8}(y_t - m_y)(y_{t+2} - m_y) = \frac{1}{7}[(-13)\bullet 7 + 17\bullet(-13) + 7\bullet(-3) + \\ + (-13)\bullet 17 + (-3)\bullet(-23) + 17\bullet 7 + (-23)\bullet 17 + 7\bullet(-13)] = -116, \tag{6.150}$$

$$\gamma_z[3] = \frac{1}{6}\sum_{t=1}^{7}(y_t - m_y)(y_{t+3} - m_y) = \frac{1}{6}[(-13)\bullet(-13) + 17\bullet(-3) + \\ + 7\bullet 17 + (-13)\bullet(-23) + (-3)\bullet 7 + 17\bullet 17 + (-23)\bullet(-13)] = -173, \tag{6.151}$$

$$\rho_z[1] = \frac{\gamma_z[1]}{\gamma_z[0]} = \frac{-105}{221} = -0{,}475, \tag{6.152}$$

$$\rho_z[2] = \frac{\gamma_z[2]}{\gamma_z[0]} = \frac{-116}{221} = -0{,}525, \tag{6.153}$$

$$\rho_z[3] = \frac{\gamma_z[3]}{\gamma_z[0]} = \frac{-173}{221} = -0{,}783\,. \tag{6.154}$$

Now we have all the necessary data to determine the matrices

$$M, \phi, \rho, \tag{6.155}$$

the use of autoregression models (6.143) - (6.145) is used.

Since the model (6.143) $q = 1$, then these matrices will look like it

$$M = [1], \qquad \phi = [\phi_1], \qquad \rho = [\rho_z[1]]. \tag{6.156}$$

Subituting (6.156) in (6.110) and given that in this case $M^{-1} = M$ we will have

$$\phi_1 = \rho_z[1]\,, \tag{6.157}$$

and taking into account the expression (6.152) –

$$\phi_1 = -0{,}475\,. \tag{6.158}$$

Since the model (6.144) $q = 2,$ the matrices for her will look like

$$M = \begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix}, \qquad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}, \qquad \rho = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}, \tag{6.159}$$

And taking into account expressions (6.152), (6.153) -

$$M = \begin{bmatrix} 1 & -0{,}475 \\ -0{,}475 & 1 \end{bmatrix}, \qquad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}, \qquad \rho = \begin{bmatrix} -0{,}475 \\ -0{,}525 \end{bmatrix} \tag{6.160}$$

Since the model (6.145) $q = 3,$ the matrices (6.155) will look like it

$$M = \begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix}, \qquad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}, \qquad \rho = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix} \tag{6.161}$$

And taking into account expressions (6.152), (6.153), (6.154) -

$$M = \begin{bmatrix} 1 & -0{,}475 & -0{,}525 \\ -0{,}475 & 1 & -0{,}475 \\ -0{,}525 & -0475 & 1 \end{bmatrix}, \qquad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}, \qquad \rho = \begin{bmatrix} -0{,}475 \\ -0{,}525 \\ -0{,}783 \end{bmatrix} \tag{6.162}$$

Comparing the expressions (6.103) and (6.157), we see that to identify the model of autoregression (6.143) we have only to find a variance σ_a^2 of "white noise" $a(t),$ from which a computer program will form pulses $a(t),$ for this model. We will find this variance from the expression (6.103), substituting in which expressions (6.148), (6.149) and (6.158), we will get

$$\sigma_a^2 = \gamma_z[0] - \phi_1 \gamma_z[1] = 221 - (-0{,}475)(-105) = 171{,}125 \approx 171\,. \tag{6.163}$$

So in this case the model of the stokastic component z_t of the original coordinate y_t will look like

$$z_t = \phi_1 z_{t-1} + a_t = -0{,}475 z_{t-1} + a_t, \tag{6.164}$$

And the value of the source coordinate y_t will be found from the expression (6.146), which after substitution of its value μ from the expression (6.147) will look like

$$y_t = 33 + z_t\,. \tag{6.165}$$

Substituting in expression (6.165) value from Table 2, we find that

$$z_{10} = y_{10} - 33 = 20 - 33 = -13\,. \tag{6.166}$$

And if we want to prognoze the value of the source coordinate y_{11} of the object, then, first, we generate the computer program of the «white noise» a_{11}. And let this value a_{11} be equal to the average deviation $\sigma_a = \sqrt{\sigma_a^2} \approx 13$. We substitute this value a_{11} and z_{10} the expression (6.164) and get a numerical value z_{11} that in our case will be equal

$$z_{11} = -0{,}475 z_{10} + a_{11} = -0{,}475(-13) + 13 = 19{,}175 \approx 19\,. \tag{6.167}$$

And then the obtained numerical value is substituted into an expression (6.165) and find that

$$y_{11} = 33 + z_{11} = 33 + 19 = 52\,. \tag{6.168}$$

Similarly, using z_{11}, y_{11} and generating the next white noise pulse, we can predict at time $t = 10$ and value z_{12}, y_{12}, and then, using already obtained z_{13}, y_{13} and, predict and, however, it is clear that the more distant moments of time $t = 10$ we will use, the accuracy the forecast will decrease.

But before using the 1-st order identified in the form of a 1-st order (6.164) to predict the following values of the source coordinate of the object, it is necessary to make sure that this model specifies sufficient accuracy of the forecast. And for this purpose it is necessary, using matrices (6.160), to identify the model of the time row given by table 2, in the form of autoregression of the 2-nd order (6.144) and compare the degree of accuracy of forecasting according to both models.

And, of course, when processing large arrays of the values of the time series for the synthesis of autoregression operator requires all those calculations that we have made for the sake of clearly demonstrating their essence and structure, to perform in some software environment, in our case it is Python.

6.4 Python programs for implementing tasks with special operators

A Python program for solving problems related to the use of the direct Laplace operator to transform functions *f*(*t*) of a real variable *t* into a complex plane in the form of functions *F*(*p*) of a complex variable *p*

(Program 21)

```
In [1]: import sympy
In [2]: from sympy import *
In [3]: t = symbols ('t')
In [4]: p = symbols ('p')
In [5]: f = Function ('f')(t)
In [6]: F = Function ('F')(p)
In [7]: f = t
In [8]: f1 = f*exp(-p*t)
In [9]: F1 = integrate (f1,(t,0,oo))
In [10]: F1
Out[10]:
Piecewise((p**(-2), Abs(arg(p)) < pi/2),
 (Integral(t*exp(-p*t), (t, 0, oo)), True))
In [11]: K = ((p**(-2), Abs(arg(p)) < pi/2), \
         (Integral(t*exp(-p*t), (t, 0, oo)), True))
In [12]: K[0]
Out [12]: (p**(-2), Abs(arg(p)) < pi/2)
In [13]: K[0][0]
Out [13]: p**(-2)
In [14]: F = K[0][0]
In [15]: print(F)
Out[15]: p**(-2)
In [16]: f2 = exp (- 2*t)
In [17]: f3 = f2*exp(-p*t)
In [18]: F3 = integrate (f3,(t,0,oo))
In [19]: F3
Out[19]:
Piecewise((1/(2*(p/2 + 1)), Abs(arg(p)) <= pi/2),
(Integral(exp(-2*t)*exp(-p*t), (t, 0, oo)), True))
In [20]: K1 = ((1/(2*(p/2 + 1)), Abs(arg(p))\
          <= pi/2), (Integral(exp(-2*t)*exp(-p*t), \
          (t, 0, oo)), True))
In [21]: K1[0]
Out [21]: (1/(2*(p/2 + 1)), Abs(arg(p)) <= pi/2)
In [22]: K1[0][0]
Out [22]: 1/(p+2)
In [23]: F2 = K1[0][0]
In [24]: print(F2)
Out[24]: 1/(p+2)
In [25]: f4 = f*f2*exp(-p*t)
In [26]: F5 = integrate (f4,(t,0,oo))
In [27]: F5
Out [27]:
Piecewise((1/(4*(p/2 + 1)**2), Abs(arg(p)) \
<= pi/2),(Integral(t*exp(-2*t)*exp(-p*t),\
(t, 0, oo)), True))
In [28]: K2 = ((1/(4*(p/2 + 1)**2), Abs(arg(p))\
          <= pi/2), (Integral(t*exp(-2*t)*\
          exp(-p*t), (t, 0, oo)), True))
In [29]: K2[0]
Out [29]:
(1/(4*(p/2 + 1)**2), Abs(arg(p)) <= pi/2)
In [30]: K2[0][0]
Out [30]: 1/(p+2)**2
In [31]: F4 = K2[0][0]
In [32]: print(F4)
Out[32]: 1/(p+2)**2
In [33]: f5 = f*f2*sin (3*t)
```

```
In [34]: f6 = f5*exp(-p*t)
In [35]: F6 = integrate (f6,(t,0,oo))
In [36]: F6
Out[36]:
Piecewise((6/((1 + 9/(p + 2)**2)**2*\
(p + 2)**3), 2*Abs(arg(p + 2)) < pi),
(Integral(t*exp(-2*t)* \
exp(-p*t)*sin(3*t), (t, 0, oo)), True))
In [37]: K3 = ((6/((1 + 9/(p + 2)**2)**2*\
        (p + 2)**3), 2*Abs(arg(p + 2)) < pi), \
        (Integral(t*exp(-2*t)*exp(-p*t)*\
        sin(3*t), (t, 0, oo)), True))
In [38]: K3[0]
Out [38]: ((6/((1 + 9/(p + 2)**2)**2*\
(p + 2)**3),2*Abs(arg(p + 2)) < pi))
In [39]: K3[0][0]
Out [39]:
6/((1 + 9/(p + 2)**2)**2*(p + 2)**3)
In [40]: F7 = K3[0][0]
In [41]: print(F7)
Out[41]:
6/((1 + 9/(p + 2)**2)**2*(p + 2)**3)
```

End of program 21

A Python program for solving problems related to the use of the inverse Laplace operator in the form of expansion formulas for the transformation of the functions *F*(*p*) of the complex variable *p* to the axis of the real variable *t* in the form of functions *f*(*t*)

(Program 22)

```
In [1]: import sympy
In [2]: from sympy import *
In [3]: t = symbols ('t')
In [4]: p = symbols ('p')
In [5]: x = Function ('x')(t)
In [6]: y = Function ('y')(t)
In [7]: C = Function ('C')(p)
In [8]: D = Function ('D')(p)
In [9]: W = Function ('W')(p)
In [10]: X = Function ('X')(p)
In [11]: Y = Function ('Y')(p)
In [12]: D1 = Function ('D1')(p)
In [13]: Y1 = Function ('Y1')(p,t)
In [14]: Y2 = Function ('Y2')(p,t)
In [15]: C = 2*p+4
In [16]: D = p**2+7*p+12
In [17]: W = C/D
In [18]: W
Out[18]:
(2*p + 4)/(p**2 + 7*p + 12)
In [19]: x = t
In [20]: x1 = x*exp(-p*t)
In [21]: X1 = Function ('X1')(p)
In [22]: X1 = integrate (x1,(t,0,oo));X1
Out[22]:
Piecewise((p**(-2), Abs(arg(p)) < pi/2),
(Integral(t*exp(-p*t), (t, 0, oo)), True))
In [23]: K1 = ((p**(-2), Abs(arg(p)) < pi/2), \
        (Integral(t*exp(-p*t), (t, 0, oo)), True))
In [24]: K1[0]
Out[24]: (p**(-2), Abs(arg(p)) < pi/2)
In [25]: X = K1[0][0]
In [26]: X
Out[26]:
p**(-2)
In [27]: Y = W*X
In [28]: Y
Out[28]:
(2*p + 4)/(p**2*(p**2 + 7*p + 12))
In [29]: C1 = Function ('C1')(p)
In [29]: expr = Y
In [30]: C1, D1 = fraction (expr)
In [31]: print(C1)
2*p + 4
In [32]: print(D1)
p**2*(p**2 + 7*p + 12)
In [33]: D2 = Function ('D2')(p)
In [34]: D2 = D1.diff(p)
In [35]: D2
Out[35]:
p**2*(2*p + 7) + 2*p*(p**2 + 7*p + 12)
In [36]: solveset (Eq(D1,0), p)
Out[36]:
FiniteSet(-4, -3, 0)
In [37]: roots (Eq(D1,0), p)
Out[37]: {-3: 1, -4: 1, 0: 2}
In [38]: d0 ={ }
In [39]: d0["a"]=-3
In [40]: d0["b"]=-4
In [41]: d0["c"]=0
In [42]: d0
Out[42]: {'a': -3, 'b': -4, 'c': 0}
```

```
In [43]: p1,p2,p3 = symbols('p1 p2 p3')
In [44]: p1 = d0['a']
In [45]: p1
Out[45]: -3
In [46]: p2 = d0['b']
In [47]: p2
Out[47]: -4
In [48]: p3 = d0['c']
In [49]: p3
Out[49]: 0
In [50]: Y1=C1*exp(p*t)/D2
In [51]: Y1
Out[51]:
(2*p + 4)*exp(p*t)/(p**2*(2*p + 7) +\
2*p*(p**2 + 7*p + 12))
In [52]: Y2 = diff(C1*(p-p3)**2*  \
                   exp(p*t)/D1,p)
In [53]: Y2
Out[53]:
t*(2*p + 4)*exp(p*t)/(p**2 + 7*p + 12)\
+ (-2*p - 7)*(2*p + 4)*exp(p*t)/(p**2 +7*p +12)**2\
+ 2*exp(p*t)/(p**2 + 7*p + 12)
In [54]: y = Y1.subs(p,p1)+Y1.subs(p,p2)+\
                   Y2.subs(p,p3)
In [55]: y
Out[55]:
t/3 - 1/36 - 2*exp(-3*t)/9 + exp(-4*t)/4
In [56]: p11 = plot(x,(t,0,2),show=False,\
                   line_color = 'c')
In [57]: p22 = plot(y,(t,0,2),show=False,\
                   line_color = 'r')
In [58]: p11.extend(p22)
In [59]: p11.show( )
```

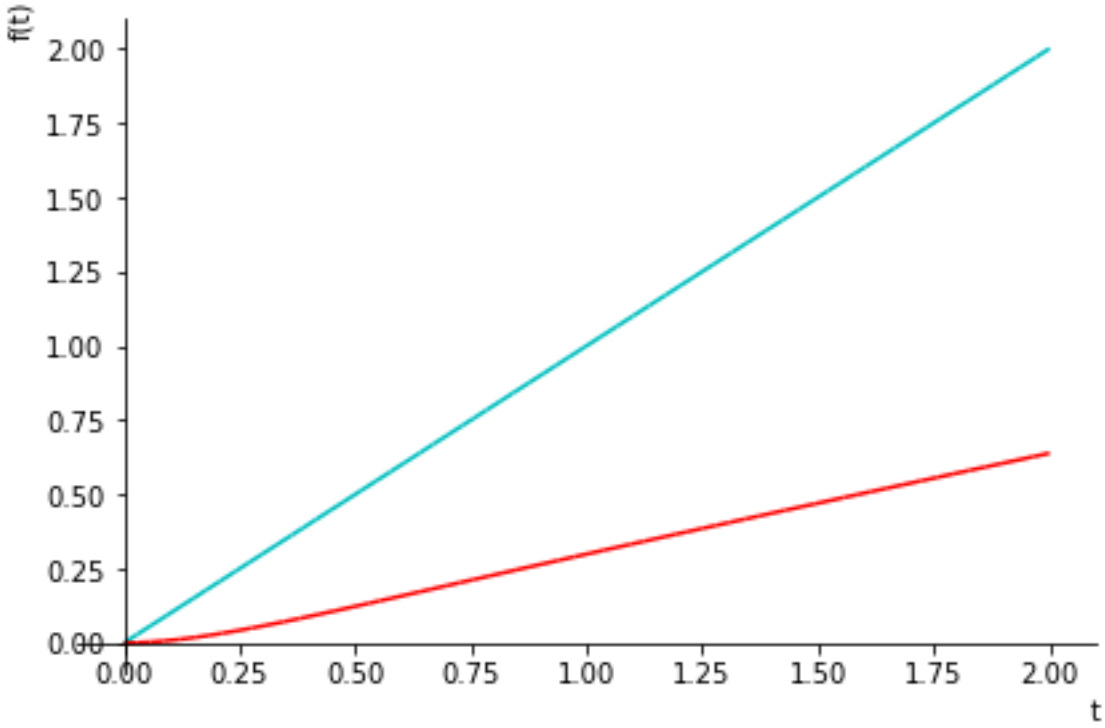


Figure 19. Graphs of the input signal $x(t) = t$ entering the dynamic system with a given transfer function $W(p)$, and its output signal $y(t)$ on time interval $t \in [0,2]$

End of program 22

A Python program to solve a problem related autoregressive identification of the stationary time series
y_t={5.,8.,3.,4.,6.,3.,2.,7.,5.,4.,3.,6.,4.,5.,3.,8.,6.,4.,3.,5.,4.,2.,7.,4.,5.,3.,6.,3.,4.,5.}, given at N points $t \in [0,N-1]$ with N=30, using the AR(3) model with the structure:

$$y_t = b + m_t, \qquad b = \frac{1}{N}\sum_{i=1}^{N} y_i, \qquad m_t = g_1 m_{t-1} + g_2 m_{t-2} + g_3 m_{t-3} + a_t$$

and the Yule-Walker algorithm for determining its parameters

(Program 23):

```
In [1]: import numpy as np
In [2]: L=[5.,8.,3.,4.,6.,3.,2.,7.,5.,4.,3.,6.,4.,5.,
3.,8.,6.,4.,3.,5.,4.,2.,7.,4.,5.,3.,6.,3.,4.,5.]
In [3]: N=30
In [4]: def fun (x):
    return np.sum(x)
In [5]: fun(L)
Out[5]: 137.0
In [6]: b=_/N;b
Out[6]: 4.566666666666666
In [7]: b=b.round(3);b
Out[7]: 4.567
In [8]: L1=L-b;L1
Out[8]:
array([ 0.433,  3.433, -1.567, -0.567,  1.433,
 -1.567, -2.567,  2.433,  0.433, -0.567, -1.567,
  1.433, -0.567,  0.433, -1.567,  3.433,
  1.433, -0.567, -1.567,  0.433, -0.567,
-2.567,  2.433, -0.567,  0.433, -1.567,  1.433,
 -1.567, -0.567,  0.433])
In [9]: def fun (x):
    return np.dot(x,x)
In [10]: fun(L1)
Out[10]:
77.36667
In [11]: q0=_/N; q0
Out[11]:
 2.5788889999999998
In [12]: q0=q0.round(3); q0
Out[12]:
 2.579
In [13]: L2=L1[:-1];L2
Out[13]:
array([ 0.433,  3.433, -1.567, -0.567,  1.433,
-1.567, -2.567,  2.433,  0.433, -0.567, -1.567,
  1.433, -0.567,  0.433, -1.567,  3.433,  1.433,
-0.567, -1.567,  0.433, -0.567, -2.567,  2.433,
-0.567,  0.433, -1.567,  1.433, -1.567, -0.567])
In [14]: L5=L1[1:];L5
Out[14]:
array([ 3.433, -1.567, -0.567,  1.433, -1.567,
-2.567,  2.433,  0.433, -0.567, -1.567,  1.433,
-0.567,  0.433, -1.567,  3.433,  1.433,  -0.567,
 -1.567,  0.433, -0.567, -2.567,  2.433, -0.567,
 0.433, -1.567,  1.433, -1.567, -0.567,  0.433])
In [15]: def fun (x,y):
    return np.dot(x,y)
In [16]: fun(L2,L5)
Out[16]:
 -22.820818999999993
In [17]: q1=_/(N-1); q1
Out[17]:
 -0.786924793103448
In [18]: q1=q1.round(3);q1
Out[18]:
 -0.787
In [19]: L3=L2[:-1];L3
Out[19]:
array([ 0.433,  3.433, -1.567, -0.567,  1.433,
-1.567, -2.567,  2.433,  0.433, -0.567, -1.567,
 1.433, -0.567,  0.433, -1.567,  3.433,  1.433,
-0.567, -1.567,  0.433, -0.567, -2.567,  2.433,
 -0.567,  0.433, -1.567,  1.433, -1.567])
In [20]: L6=L5[1:];L6
Out[20]:
array([-1.567, -0.567,  1.433, -1.567, -2.567,
  2.433,  0.433, -0.567, -1.567,  1.433, -0.567,
  0.433, -1.567,  3.433,  1.433, -0.567, -1.567,
  0.433, -0.567, -2.567,  2.433, -0.567,  0.433,
 -1.567,  1.433, -1.567, -0.567,  0.433])
In [21]: fun(L3,L6)
Out[21]:
-14.874308000000001
In [22]: q2=_/(N-2);q2
Out[22]:
 -0.5312252857142857
In [23]: q2=q2.round(3);q2
Out[23]: -0.531
In [24]: L4=L3[:-1];L4
Out[24]:
array([ 0.433,  3.433, -1.567, -0.567,  1.433,
-1.567, -2.567,  2.433,  0.433, -0.567, -1.567,
 1.433, -0.567,  0.433, -1.567,  3.433,  1.433,
-0.567, -1.567,  0.433, -0.567, -2.567,  2.433,
 -0.567,  0.433, -1.567,  1.433])
In [25]: L7=L6[1:];L7
Out[25]:
array([-0.567,  1.433, -1.567, -2.567,  2.433,
  0.433, -0.567, -1.567,  1.433, -0.567,  0.433,
 -1.567,  3.433,  1.433, -0.567, -1.567,  0.433,
 -0.567, -2.567,  2.433, -0.567,  0.433, -1.567,
  1.433, -1.567, -0.567,  0.433])
In [26]: fun(L4,L7)
Out[26]:
2.6702030000000008
In [27]: q3=_/(N-3); q3
Out[27]:
0.09889640740740743
In [28]: q3=q3.round(3); q3
Out[28]:
0.099
In [29]: r0=q0/q0;r0
Out[29]:
 1.0
In [30]: r1=q1/q0;r1
```

```
Out[30]:
 -0.30515703761147733
In [31]: r1=r1.round(3);r1
Out[31]:
 -0.305
In [32]: r2=q2/q0;r2
Out[32]:
 -0.2058937572702598
In [33]: r2=r2.round(3);r2
Out[33]:
 -0.206
In [34]: r3=q3/q0;r3
Out[34]:
 0.038386971694455214
In [35]: r3=r3.round(3);r3
Out[35]:
 0.038
In [36]: L9=[r0,r1,r2,r3];L9
Out[36]:
 [1.0, -0.305, -0.206, 0.038]
In [37]: import sympy
In [38]: from sympy import*
In [39]: r,r0,r1,r2,r3 = symbols('r r0 r1 r2 r3')
In [40]: M = symbols('M')
In [41]: M = Matrix([[r0,r1,r2],[r1,r0,r1],
[r2,r1,r0]]);M
Out[41]:
Matrix([
[r0, r1, r2],
[r1, r0, r1],
[r2, r1, r0]])
In [42]: g,g1,g2,g3=symbols('g g1 g2 g3')
In [43]: g = Matrix([g1,g2,g3]);g
Out[43]:
Matrix([
[g1],
[g2],
[g3]])
In [44]: M =M.subs([(r0,1),(r1,-0.305),
(r2,-0.206)]);M
Out[44]:
Matrix([
[     1, -0.305, -0.206],
[-0.305,      1, -0.305],
[-0.206, -0.305,      1]])
In [45]: r=Matrix([r1,r2,r3]); r
Out[45]:
Matrix([
[r1],
[r2],
[r3]])
In [46]: r=r.subs([(r1,-0.305),(r2,-0.206),
(r3,0.038)]); r
Out[46]:
Matrix([
[-0.305],
[-0.206],
[ 0.038]])
In [47]: B=simplify(M.inv());B
Out[47]:
Matrix([
[ 1.23702975377247, 0.501685993913973,
  0.40784235742089],[0.501685993913973,
  1.30602845628752, 0.501685993913973],
[ 0.40784235742089, 0.501685993913973,
  1.23702975377247]])
In [48]: g=B*r; g
Out[48]:
Matrix([
[-0.465143380064886],
[-0.402992022370261],
[-0.180732103116296]])
In [49]: g=g.evalf(3); g
Out[49]:
Matrix([
[-0.465],
[-0.403],
[-0.181]])
In [50]: g1=g[0,0]; g1
Out[50]:
-0.465
In [51]: g2=g[1,0]; g2
Out[51]:
-0.403
In [52]: g3=g[2,0]; g3
Out[52]:
-0.181
In [53]: a = symbols ('a')
In [54]: ska = symbols ('ska')
In [55]: ska = q0-g1*q1-g2*q2-g3*q3; ska
Out[55]:
2.01681854248047
In [56]: skv = symbols ('skv')
In [57]: skv = ska**(0.5); skv
Out[57]:
1.42014736646605
In [58]: skv = skv.evalf(3); skv
Out[58]:
1.42
In [59]: a11,a22 = symbols ('a11 a22')
In [60]: a11 = -3*skv; a11
Out[60]:
-4.26
In [61]: a22 = 3*skv; a22
Out[61]:
4.26
```

```
In [62]: import random as rnd
In [63]: m = symbols('m:10');
Out[63]:
(m0, m1, m2, m3, m4, m5, m6, m7, m8, m9)
In [64]: l = symbols ('l:10');l
Out[64]:
(l0, l1, l2, l3, l4, l5, l6, l7, l8, l9)
In [65]: m=list(m);m
Out[65]:
[m0, m1, m2, m3, m4, m5, m6, m7, m8, m9]
In [66]: l=list(l);l
Out[66]:
[l0, l1, l2, l3, l4, l5, l6, l7, l8, l9]
In [67]: d = symbols ('d:10'); d
Out[67]:
(d0,d1,d2,d3,d4,d5,d6,d7,d8,d9)
In [68]: d = list (d); d
Out[68]:
[d0,d1,d2,d3,d4,d5,d6,d7,d8,d9]
In [69]: d[0] = rnd.uniform (-4.26,4.26); d[0]
Out[69]:
3.7878324717658174
In [70]: m[0]=g1*L1[29]+g2*L1[28]+\
         g3*L1[27]+ d[0]; m[0]
Out[70]:
4.09812879744941
In [71]: L1 = np.append(L1,[m[0]]);L1
Out[71]:
array([0.43299999999999983, 3.433,
-1.5670000000000002, -0.5670000000000002,
1.4329999999999998, -1.5670000000000002,
-2.567, 2.433, 0.43299999999999983,
-0.5670000000000002, -1.5670000000000002,
1.4329999999999998, -0.5670000000000002,
0.43299999999999983, -1.5670000000000002,
3.433,  1.4329999999999998,
-0.5670000000000002, -1.5670000000000002,
0.43299999999999983, -0.5670000000000002,
-2.567, 2.433, -0.5670000000000002,
0.43299999999999983, -1.5670000000000002,
1.4329999999999998, -1.5670000000000002,
-0.5670000000000002,  0.43299999999999983
4.09812879744941], dtype=object)
In [72]: l[0]=b+m[0]; l[0]
Out[72]:
8.66512879744941
In [73]: L = np.append(L,[l[0]]);L
Out[73]:
array([5.0, 8.0, 3.0, 4.0, 6.0, 3.0, 2.0, 7.0, 5.0,
4.0, 3.0, 6.0, 4.0, 5.0, 3.0, 8.0, 6.0, 4.0, 3.0,
5.0, 4.0, 2.0, 7.0, 4.0, 5.0, 3.0, 6.0, 3.0, 4.0,
5.0, 8.66512879744941], dtype=object)
In [74]: d[1] = rnd.uniform (-4.26,4.26); d[1]
Out[74]:
-2.587298897242724
In [75]: m[1]= g1*L1[30]+g2*L1[29]+\
         g3*L1[28]+ d[1]; m[1]
Out[75]:
-2.08074576204602
In [76]: L1 = np.append(L1,[m[1]])
In [77]: l[1]=b+m[1]; l[1]
Out[77]:
2.48625423795398
In [78]: L=np.append(L,[l[1]])
In [79]: d[2] = rnd.uniform (-4.26,4.26); d[2]
Out[79]:
0.8071389548290568
In [80]: m[2]= g1*L1[31]+g2*L1[30]+\
         g3*L1[29]+ d[2]; m[2]
Out[80]:
0.0451337059883733
In [81]: L1 = np.append(L1,[m[2]])
In [82]: l[2]=b+m[2]; l[2]
Out[82]:
4.61213370598837
In [83]: L=np.append(L,[l[2]])
In [84]: d[3] = rnd.uniform (-4.26,4.26); d[3]
Out[84]:
3.5146713080878706
In [85]: m[3]= g1*L1[32]+g2*L1[31]+\
         g3*L1[30]+ d[3]; m[3]
Out[85]:
3.59161472387185
In [86]: L1 = np.append(L1,[m[3]])
In [87]: l[3]=b+m[3]; l[3]
Out[87]:
8.15861472387185
In [88]: L=np.append(L,[l[3]])
In [89]: d[4] = rnd.uniform (-4.26,4.26); d[4]
Out[89]:
3.332153915638866
In [90]: m[4]= g1*L1[33]+g2*L1[32]+\
         g3*L1[31]+ d[4]; m[4]
Out[90]:
0.178801469969288
In [91]: L1 = np.append(L1,[m[4]])
In [92]: l[4]=b+m[4]; l[4]
Out[92]:
4.74580146996929
In [93]: L=np.append(L,[l[4]])
In [94]: d[5] = rnd.uniform (-4.26,4.26); d[5]
Out[94]:
2.867785147080328
In [95]: m[5]= g1*L1[34]+g2*L1[33]+\
         g3*L1[32]+ d[5]; m[5]
Out[95]:
1.32898394299138
```

```
In [96]: L1 = np.append(L1,[m[5]])
In [97]: l[5]=b+m[5]; l[5]
Out[97]:
5.89598394299138
In [98]: L=np.append(L,[l[5]])
In [99]: d[6] = rnd.uniform (-4.26,4.26); d[6]
Out[99]:
1.69270588124252
In [100]: m[6]= g1*L1[35]+g2*L1[34]+\
          g3*L1[33]+ d[6]; m[6]
Out[100]:
-0.471995713797095
In [101]: L1 = np.append(L1,[m[6]])
In [102]: l[6]=b+m[6]; l[6]
Out[102]:
4.09500428620290
In [103]: L=np.append(L,[l[6]])
In [104]: d[7] = rnd.uniform (-4.26,4.26); d[7]
Out[104]:
0.10021632009046666
In [105]: m[7]= g1*L1[36]+g2*L1[35]+\
          g3*L1[34]+ d[7]; m[7]
Out[105]:
-0.248149939266434
In [106]: L1 = np.append(L1,[m[7]])
In [107]: l[7]=b+m[7]; l[7]
Out[107]:
4.31885006073357
In [108]: L=np.append(L,[l[7]])
In [109]: d[8] = rnd.uniform (-4.26,4.26); d[8]
Out[109]:
-2.232660689069062
In [110]: m[8]= g1*L1[37]+g2*L1[36]+\
          g3*L1[35]+ d[8]; m[8]
Out[110]:
-2.16719334714070
In [111]: L1 = np.append(L1,[m[8]])
In [112]: l[8]=b+m[8]; l[8]
Out[112]:
2.39980665285930
In [113]: L=np.append(L,[l[8]])
In [114]: d[9] = rnd.uniform (-4.26,4.26); d[9]
Out[114]:
1.544009122634895
In [115]: m[9]= g1*L1[38]+g2*L1[37]+\
          g3*L1[36]+ d[9]; m[9]
Out[115]:
2.73738643318719
In [116]: L1 = np.append(L1,[m[9]])
In [117]: l[9]=b+m[9]; l[9]
Out[117]:
7.30438643318719
In [118]: L=np.append(L,[l[9]])
In [119]: import matplotlib
In [120]: import matplotlib.pyplot as plt
In [121]: plt.plot(L1)
```

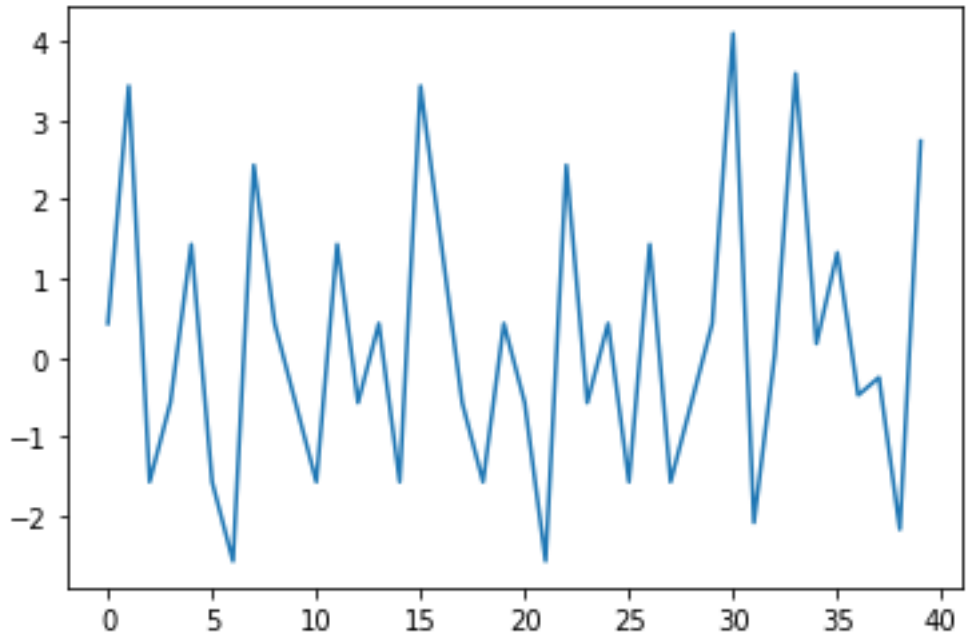


Figure 20. The graph of the time series y_t, which in the range $t \in [0,30]$ is filled with experimentally obtained values, and outside this range is filled with predicted values obtained using an autoregression model identified using experimentally obtained values

End of program 23.

A Python program to solve a problem related to autoregressive modeling of non-stationary time series y_t={ 5.,8.,3.,4.,6.,8.,10.,7.,6.,9.,12.,8.,11.,15.,12.,10., 7.,8.,12.,15., 18.,20.,17., 14.,15.,17.,16.,19.,22.,25.]}, given at *N* points $t \in [0,N-1]$ with N=30, using the ARIMA(3,0,2) model with the structure:

$$v_t = y_t - y_{t-1},\ w_t = v_t - v_{t-1},$$
$$w_t = g_1 w_{t-1} + g_2 w_{t-2} + g_3 w_{t-3} + a_t,$$
$$y_k = y_{-1} + \sum_{t=0}^{k} v_t, \qquad v_k = v_{-1} + \sum_{t=0}^{k} w_t$$

and the method of its identification in the classical form

(Program 24):

```
In [1]: import numpy as np
In [2]: L=[5.,8.,3.,4.,6.,8.,10.,7.,6.,9.,12.,8.,\
11.,15.,12.,10.,7.,8.,12.,15.,18.,20.,17.,14.,
15.,17.,16.,19.,22.,25.]
In [3]: N=30
In [4]: L1 = np.diff(L);L1
Out[4]:
array([ 3., -5.,  1.,  2.,  2.,  2., -3., -1.,  3.,  3., \
-4.,  3.,  4., -3., -2., -3.,  1.,  4.,  3.,  3.,  2., -3.,\
 -3.,  1.,  2., -1.,  3.,  3.,  3.])
In [5]: def fun (x):
    return np.sum(x)
In [6]: fun(L1)
Out[6]: 20.0
In [7]: _/(N-1)
In [8]: L11=np.diff(L1);L11
Out[8]:
array([-8.,  6.,  1.,  0.,  0., -5.,  2.,  4.,  0.,\
 -7.,  7.,  1., -7.,  1., -1.,  4.,  3., -1.,  0., -1.,\
 -5.,  0.,  4.,  1., -3.,  4.,  0.,  0.])
In [9]: fun(L11)
Out[9]: 0.0
In [10]: def fun (x):
    return np.dot(x,x)
In [11]: fun(L11)
Out[11]:
390.0
In [12]: q0=_/(N-2); q0
Out[12]:
13.928571428571429
In [13]: q0=q0.round(3);q0
Out[13]:
13.929
In [14]: L2=L11[:-1]
In [15]: L5=L11[1:]
In [16]: def fun (x,y):
    return np.dot(x,y)
In [17]: fun(L2,L5)
Out[17]:
-102.0
In [18]: q1=_/(N-3); q1
Out[18]:
-3.7777777777777777
In [19]: q1=q1.round(3);q1
Out[19]:
-3.778
In [20]: L3=L2[:-1]
In [21]: L6=L5[1:]
In [22]: fun(L3,L6)
Out[22]:
-134.0
In [23]: q2=_/(N-4);q2
Out[23]:
-5.153846153846154
In [24]: q2=q2.round(3);q2
Out[24]:
-5.154
In [25]: L4=L3[:-1]
In [26]: L7=L6[1:]
In [27]: fun(L4,L7)
Out[27]:
49.0
In [28]: q3=_/(N-5); q3
Out[28]:
1.96
In [29]: r0=q0/q0;r0
Out[29]:
 1.0
In [30]: r1=q1/q0;r1
Out[30]:
-0.29059829059829057
In [31]: r1=r1.round(3);r1
Out[31]:
-0.291
In [32]: r2=q2/q0;r2
Out[32]:
-0.3964615384615385
In [33]: r2=r2.round(3);r2
Out[33]:
-0.396
In [34]: r3=q3/q0;r3
Out[34]:
0.15076923076923077
In [35]: r3=r3.round(3);r3
Out[35]:
0.151
```

```
In [36]: L9=[r0,r1,r2,r3];L9
Out[36]:
[1.0, -0.291, -0.396, 0.151]
In [37]: import sympy
In [38]: from sympy import*
In [39]: r,r0,r1,r2,r3 = symbols('r r0 r1 r2 r3')
In [40]: M = symbols('M')
In [41]: M = Matrix([[r0,r1,r2],[r1,r0,r1],
[r2,r1,r0]]);M
Out[41]:
Matrix([
[r0, r1, r2],
[r1, r0, r1],
[r2, r1, r0]])
In [42]: g,g1,g2,g3=symbols('g g1 g2 g3')
In [43]: g = Matrix([g1,g2,g3]);g
Out[43]:
Matrix([
[g1],
[g2],
[g3]])
In [44]: M =M.subs([(r0,1),(r1,-0.291),
(r2,-0.396)]);M
Out[44]:
Matrix([
[     1, -0.291, -0.396 ],
[ -0.291,      1, -0.291],
[-0.396, -0.291,      1 ]])
In [45]: r=Matrix([r1,r2,r3]); r
Out[45]:
Matrix([
[r1],
[r2],
[r3]])
In [46]: r=r.subs([(r1,-0.291),(r2,-0.396),
(r3,0.151)]); r
Out[46]:
Matrix([
[ -0.291],
[ -0.396],
[ 0.151]])
In [47]: B=simplify(M.inv());B
Out[47]:
Matrix([
[ 1.50854880637025, 0.669522683244447, \
0.792216428146752],[0.669522683244447, \
 1.38966220164827, 0.669522683244447],
[0.792216428146752, 0.669522683244447, \
 1.50854880637025]])
In [48]: g=B*r; g
Out[48]:
Matrix([
[ -0.584494004568384],
[-0.644039407506937],
[-0.267875093393598 ]])
In [49]: g=g.evalf(3); g
Out[49]:
Matrix([
[-0.584 ],
[ -0.644],
[ -0.268]])
In [50]: g1=g[0,0]; g1
Out[50]:
-0.584
In [51]: g2=g[1,0]; g2
Out[51]:
-0.644
In [52]: g3=g[2,0]; g3
Out[52]:
-0.268
In [53]: a = symbols ('a')
In [54]: ska = symbols ('ska')
In [55]: ska = q0-g1*q1-g2*q2-g3*q3; ska
Out[55]:
7.99764599609375
In [56]: skv = symbols ('skv')
In [57]: skv = ska**(0.5); skv
Out[57]:
2.82801096109859
In [58]: skv = skv.evalf(3); skv
Out[58]:
2.83
In [59]: a11,a22 = symbols ('a11 a22')
In [60]: a11 = -2*skv; a11
Out[60]:
-5.66
In [61]: a22 = 2*skv; a22
Out[61]:
5.66
In [62]: import random as rnd
In [63]: w = symbols('w:10');w
Out[63]:
(w0, w1, w2, w3, w4, w5, w6, w7, w8, w9)
In [64]: v = symbols ('v:10');v
Out[64]:
(v0, v1, v2, v3, v4, v5, v6, v7, v8, v9)
In [65]: w=list(w);w
Out[65]:
[w0, w1, w2, w3, w4, w5, w6, w7, w8, w9]
In [66]: v=list(v);v
Out[66]:
[v0, v1, v2, v3, v4, v5, v6, v7, v8, v9]
In [67]: d = symbols ('d:10'); d
```

```
Out[67]:
(d0,d1,d2,d3,d4,d5,d6,d7,d8,d9)
In [68]: d = list (d); d
Out[68]:
[d0,d1,d2,d3,d4,d5,d6,d7,d8,d9]
In [69]: y = symbols ('y:10');y
Out[69]:
(y0, y1, y2, y3, y4, y5, y6, y7, y8, y9)
In [70]: y=list(y);y
Out[70]:
[y0, y1, y2, y3, y4, y5, y6, y7, y8, y9]
In [71]: L21=[ ]
In [72]: L22=[ ]
In [73]: L23=[ ]
In [74]: d[0] = rnd.uniform (-5.66,5.66); d[0]
Out[74]:
-3.949689744213309
In [75]: w[0]=g1*L11[27]+g2*L11[26]+\
         g3*L11[25]+ d[0]; w[0]
Out[75]:
-5.02122294733831
In [76]: L23.append(w[0]);L23
Out[76]:
[-5.02122294733831]
In [77]: v[0]=L1[28]+w[0];v[0]
Out[77]:
-2.02122294733831
In [78]: L22.append(v[0]);L22
Out[78]:
[-2.02122294733831]
In [79]: y[0]=L[29]+v[0];y[0]
Out[79]:
22.9787770526617
In [80]: L21.append(y[0]); L21
Out[80]:
[22.9787770526617]
In [81]: d[1] = rnd.uniform (-5.66,5.66); d[1]
Out[81]:
-0.35599763798745787
In [82]: w[1]=g1*L23[0]+g2*L11[27]+\
         g3*L11[26]+ d[1]; w[1]
Out[82]:
2.57876987566682
In [83]: L23.append(w[1]);L23
Out[83]:
[-5.02122294733831, 2.57876987566682]
In [84]: v[1]=L22[0]+w[1];v[1]
Out[84]:
0.557546928328509
In [85]: L22.append(v[1]);L22
Out[85]:
[-2.02122294733831, 0.557546928328509]
In [86]: y[1]=L21[0]+v[1];y[1]
Out[86]:
23.5363239809902
In [87]: L21.append(y[1]);L21
Out[87]:
[22.9787770526617, 23.5363239809902]
In [88]: d[2] = rnd.uniform (-5.66,5.66); d[2]
Out[88]:
5.4505096830373745
In [89]: w[2]=g1*L23[1]+g2*L23[0]+\
         g3*L11[27]+ d[2]; w[2]
Out[89]:
7.17717253770830
In [90]: L23.append(w[2]);L23
Out[90]:
[-5.02122294733831, 2.57876987566682,\
7.17717253770830]
In [91]: v[2]=L22[1]+w[2];v[2]
Out[91]:
7.73471946603681
In [92]: L22.append(v[2]);L22
Out[92]:
[-2.02122294733831, 0.557546928328509, \
7.73471946603681]
In [93]: y[2]=L21[1]+v[2];y[2]
Out[93]:
31.2710434470270
In [94]: L21.append(y[2]);L21
Out[94]:
[22.9787770526617, 23.5363239809902,\
31.2710434470270]
In [95]: d[3] = rnd.uniform (-5.66,5.66); d[3]
Out[95]:
1.5127310202924091
In [96]: w[3]=g1*L23[2]+g2*L23[1]+\
         g3*L23[0]+ d[3]; w[3]
Out[96]:
-2.99786690654250
In [97]: L23.append(w[3]);L23
Out[97]:
[-5.02122294733831, 2.57876987566682, \
7.17717253770830, -2.99786690654250]
In [98]: v[3]=L22[2]+w[3];v[3]
Out[98]:
4.73685255949431
In [99]: L22.append(v[3]);L22
Out[99]:
[-2.02122294733831, 0.557546928328509, \
7.73471946603681, 4.73685255949431]
In [100]: y[3]=L21[2]+v[3];y[3]
Out[100]:
36.0078960065213
```

```
In [101]: L21.append(y[3]);L21
Out[101]:
[22.9787770526617, 23.5363239809902, \
31.2710434470270, 36.0078960065213]
In [102]: d[4] = rnd.uniform (-5.66,5.66); d[4]
Out[102]:
-2.938526432470273
In [103]: w[4]=g1*L23[3]+g2*L23[2]+\
          g3*L23[1]+ d[4]; w[4]
Out[103]:
-6.49957209318491
In [104]: L23.append(w[4]);L23
Out[104]:
[-5.02122294733831,    2.57876987566682,
7.17717253770830, -2.99786690654250,
-6.49957209318491]
In [105]: v[4]=L22[3]+w[4];v[4]
Out[105]:
-1.76271953369060
In [106]: L22.append(v[4]);L22
Out[106]:
[-2.02122294733831, 0.557546928328509,
7.73471946603681, 4.73685255949431,
-1.76271953369060]
In [107]: y[4]=L21[3]+v[4];y[4]
Out[107]:
34.2451764728307
In [108]: L21.append(y[4]);L21
Out[108]:
[22.9787770526617, 23.5363239809902,
31.2710434470270, 36.0078960065213,
34.2451764728307]
In [109]: L555 = L+L21
In [110]: import matplotlib
In [111]: import matplotlib.pyplot as plt
In [112]: plt.plot(L555)
```

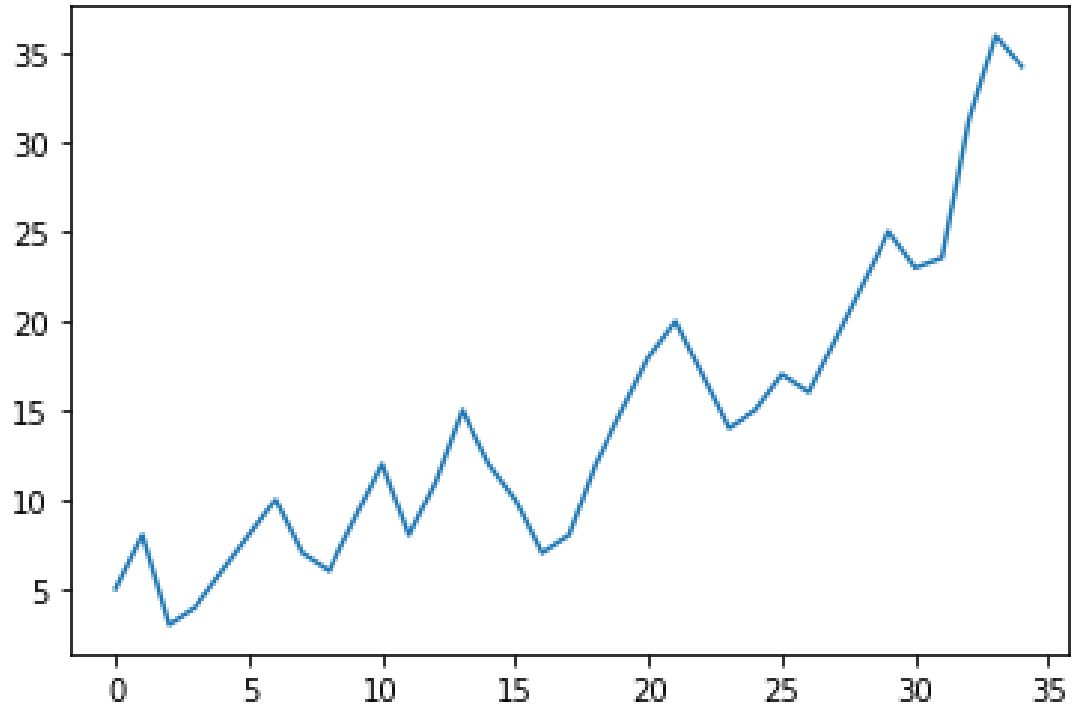


Figure 21. The graph of the time series *y_t*, which in the range *t*∈[0,30] is filled with experimentally obtained values, and outside this range is filled with forecast values obtained using the ARIMA (3,0,2) model, identified using experimental – obtained values.

End of program 24.

6.5 Tasks for self-testing

1. What is the Laplace operator?
2. What properties of the Laplace operator do you know?
3. What are the main advantages of image analysis?
4. What are the Laplace maps of the derivative of a continuous function and its integral?
5. How to determine the transfer function of the system, if the differential equation that describes the processes in this system is known?
6. Define transient and transient transient characteristics of a linear dynamic system.
7. What is a single jump and what is its schedule?
8. What is a unit impulse and what are its properties do you know?
9. How are the Laplace images of transient and impulse transient characteristics of a system related to the transfer function of this system?
10. Prove that the impulse transient characteristic of the system is a derivative of its transient characteristic.
11. Given the transfer function of a linear dynamic system, how to reproduce the differential equation that describes the processes in this system?
12. What is the inverse Laplace operator? What forms of its implementation do you know?
13. How to determine the original by its known image?
14. What is a “time series”? Give examples of stationary and non-stationary time series.
15. Define the backward and forward shift operators, the difference operator and the sum operator.
16. What is a linear filter operator?
17. What is a white noise? What are its main properties?
18. What are regression and autoregression? What form do their operators have?
19. Synthesize a time series model in the form of autoregression.
20. How are the linear filter operator and the autoregressive operator for time series related?
21. What are the moving average of a time series and the moving average operator?
22. Define autocovariance and autocorrelation of a time series. How to find their numerical values? What are their main properties do you know?
23. Why are the Yule-Walker equations needed and how are they derived?
24. How to solve the Yule-Walker equation?
25. According to the implementation of the time series, synthesize its model in the form of autoregression with the identification of this model
26. Build a time series model in the form of autoregression - moving average.
27. How can a non-stationary time series be transformed into a stationary one?
28. Build a model of a non-stationary time series in the form of autoregression - integrated moving average.

References

1. Mokin B. I. Functional analysis adapted to applied problems in the field of information technology: a textbook . B. I. Mokin, V. B. Mokin, O. B. Mokin. – Vinnytsia: VNTU, 2020 – 192 p.
2. Mokin B. I. Students textbook to learn how to solve functional analysis problems in Python, part 1 / B. I. Mokin, V. B. Mokin, O. B. Mokin. – Vinnytsia: VNTU, 2022. – 124 p.
3. Mokin B. I. Students textbook to learn how to solve problems in functional analysis in Python, part 2 / B. I. Mokin, V. B. Mokin, O. B. Mokin. – Vinnytsia: VNTU, 2023. – 144 p.
4. Box George E.P. TIME SERIES ANALYSIS. Forecasting and control. / George E. P Box, Gwilym M. Jenkins. - HOLDEN-DAY: San Francisko, Cambridge, London, Amsterdam, 1970. – 532 p.
5. Lipman. Calculus. / L. Bers. – HOLT, RINENART AND WINSTON, INC, 1969. – 488 p.
6. Bendat Julius .S. MEASUREMENT AND ANALYSIS OF RANDOM DATA./ Julius S. Bendat, Allan G. Piersol. – JOHN WILEY & SONS: New York-London-Sydney, 1967, 408 p.
7. Richard. INTRODUCTION TO MATRIX ANALYSIS. / R. Bellman. – MCGRAW-HILL BOOK COMPANY, INC: New York Toronto London, 1960. – 352 p.
8. Zgurovsky M. Z. Fundamentals of system analysis. / M. Z. Zgurovsky, N. D. Pankratova. – K.: Publishing group BHV, 2007. – 546 p.
9. Mokin B. I. Mathematical methods of identification of dynamic systems: textbook. / B. I. Mokin, V. B. Mokin, O. B. Mokin. – Vinnytsia: VNTU, 2010. – 260 p.
10. Mokin B. I. Theory of automatic control, methodology and practice of optimization: textbook./ / B. I. Mokin, V. B. Mokin, O. B. Mokin. – Vinnytsia: VNTU, 2013. – 210 p.
11. Nikolaev A. G. Functional analysis : textbook./ A. G. Nikolaev, T. V. Rvacheva, A. I. Soloviev. – Kharkiv: HAI, 2008. – 164 p.
12. Functional analysis: a textbook. / S. A. Us. – Dnipropetrovsk: National University of Mining, 2013. – 236 p. – access at : ir.nmu.org.ua/handle/123456789/3496/CD266.pdf?sequence =1
13. Polak E. COMPUTATIONAL METHODS IN OPTIMIZATION./ E. Polak. – Academic Press: New York, London, 1971. – 344 p.
14. Verlan A. F. Modeling of control systems in MATLAB environment : textbook / A. F. Verlan, I. O. Goroshko, D. E. Kontreras, V. A. Fedorchuk, V. F. Yuzvenko – K. : CCS APSU, 2002. – 68 p
15. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5108753/
16. http://scipy-lectures.org/packages/sympy.html#integration
17. https://drive.google.com/open?id=1csncEhe5s9z_bkd4tkzNXX6UDnp3jKYv
18. Python. [Electronic resource]. access at: https://www.python.org//downloads/.
19. Briggs Jason R. Python for kids (a fun introduction to programming). (English translator Oleksandra Hordiychuk). – Lviv: Old Lion Publishing House, 2019. – 400 p.
20. Dolia P.G. Introduction to Scientific Python. / P.G.Dolia.- Kharkiv: KNU named after Karazin, 2016. – 265 p.
21. Mokin B. I. On one of the approaches to the approximate calculation of Stiltjes and Lebesgue integrals in Python in problems of system analysis with discrete models / B. I. Mokin, O. B. Mokin, D. O. Shalagai. – Visnyk of Vinnytsia Politechnic Institute, 2021, №3 – P. 61-68.

Електронне навчальне видання

Мокін Борис Іванович,
Мокін Віталій Борисович,
Мокін Олександр Борисович

Functional analysis in information technologies

(Функціональний аналіз в інформаційних технологіях)

(англ. мовою)

Підручник

Рукопис підготував *Б. І. Мокін*

Редактор *М. Г. Прадівлянний*

Оригінал-макет підготовлено у *Редакційно-видавничому відділі ВНТУ*

Підписано до видання 2.04.2024 р.
Гарнітура Times New Roman, Arial Narrow.
Зам. № Р2024-073

Видавець та виготовлювач
Вінницький національний технічний університет,
редакційно-видавничий відділ.
ВНТУ, ГНК, к. 114. Хмельницьке шосе, 95,
м. Вінниця, 21021.
press.vntu.edu.ua;
Email: kivc.vntu@gmail.com
Свідоцтво суб'єкта видавничої справи