THE MINISTRY OF EDUCATION AND SCIENCE OF DPR

DONETSK NATIONAL TECHNICAL UNIVERSITY

WORKBOOK

LABORATORY WORKS ON
“MATHEMATICAL METHODS AND MODELS ”

DONETSK – DonNTU – 2017

THE MINISTRY OF EDUCATION AND SCIENCE OF DPR

DONETSK NATIONAL TECHNICAL UNIVERSITY

WORKBOOK
LABORATORY WORKS ON
"MATHEMATICAL METHODS AND MODELS "

Approved at the meeting of the
Department "Industrial Enterprise
And City Power Supply"
Minute №3/02 of 27.02.2017.

Approved at the meeting of
The Academic Editorial Board of
DonNTU
Minute № 3 of 03.04.2017 .

DONETSK – DonNTU - 2017

УДК 681.332(07)

Workbook for laboratory works on “Mathematical Methods and Models” (for the students of the speciality 7.090603 Energy Consumption Electrical Systems”) /Staff: S.G. Dzhura, S.V. Shlepnyov, V.V. Yakimishina.

The theoretical data on the computational mathematics methods applied to engineering tasks solving in electrical and power engineering is presented. The task and methodical recommendations to do 16 laboratory works to train students on the methods of computer solving of linear and non-linear equations and their systems, differential equations, methods of numerical integration, function approximation, extreme value search are given.

Compliers: S.G. Dzhura, Associate Professor.
S.V. Shlepnyov, Associate Professor
V.V. Yakimishina, Assistant Professor

Reviewer: V. M. Pavlysh, Professor

Inspector: V.I. Chursinov, Associate Professor

INTRODUCTION

One of the main current trends of the science and technical advance is the development of methods and means of the information science and computing.

The application of the mathematical methods of engineering task computer solving raises the efficiency of design, parameter computation, research, analysis and synthesis of different technical systems, including those of power supply. As to the mathematics, many energy and electrical engineering tasks add up to solving of algebraic, transcendental and differential equations and their systems, matrix, vector and set operations, table function approximation, functional minimization etc. These tasks can't always be solved analytically and require numerical method application.

This workbook has tasks and methodical recommendations to the laboratory works necessary to gain the skills of algorithmization, programming, and computer tasks solving with the help of the computational mathematics methods.

The tasks provided in the workbook can be replaced by the analogous ones from the parallel disciplines or coordinated with the SRW topics. This as well as the programming language is to be approved by a teacher.

Laboratory Work 1

COMPUTING OF POLYNOMIAL VALUE ACCORDING TO THE HORNER'S METHOD

Purpose of the work: to learn to compute the polynomial values in the most economical way, to gain the programming skills with the application of the user's functions and sub-programmes.

1.1 Theoretical Data

There is the necessity to compute the functions which look like a polynomial when automated control systems are analyzed and synthesized as well as in the electric circuit theory:

$$P_n(x)=a_0\,x^n+a_1\,x^{n-1}+\ldots+a_{n-1}x+a_n=\sum_{i=0}^{n} a_i x^{n-i}\ , \qquad (1.1)$$

Where n – polynomial degree;

$\vec{A}=(a_0,\ a_1,\ \ldots,\ a_n)$- coefficient vector,

x – independent variable .

The polynomial (1.1) can be converted into:

$$P_n(x)=(\ldots(((\,a_0x+a_1)x+a_2)x+a_3)x+\ldots+a_n)\ . \qquad (1.2)$$

The computing algorithm $P_n(x)$, formed on the basis of the phrase (1.2) is called the Horner's method.

According to this method the polynomial of the i-order is stated through the polynomial of the (i-1)-order according to the formula

$$P_i=P_{i-1}x+a_i. \tag{1.3}$$

Taking $P_0=a_0$ and doing the operation (1.3) n times under i=$1,2,...,n$, the necessary value is obtained.

The Horner's method is proved to be the most economical algorithm for the general form polynomials as to the number of operations (n additions and n multiplications).

1.2 Tasks

Compute the value of the variable z under x which changes from -1 to + 1 with the step 0.1. The phrases for z computing are given in the table 1.1. In these phrases the functions $f_1(x)$, $f_2(x)$ *i* $f_3(x)$ are polynomials which differ from each other by the coefficient order and value

For odd variants:

$$f_1(x)=1.07x^5-12x^4-2.8x^3+6.3x^2+3.7x+4,$$

$$f_2(x)=10.1x^7+37x^5-15x^4+8.2x+5.4,$$

$$f_3(x)=-23x^3+13.6x^2+0.5x-1.2.$$

For mating variants:

$$f_1(x)=8.16x^4+14x^3+0.9x^2+3.8x-2,$$

$$f_2(x)=19.7x^6+11.4x^4+2.3x^3-1.8x+0.9,$$

$$f_3(x)=21.6x^5-17.4x^4+8.7x^3+11x.$$

Make up the sub-programme (procedure) to work in the polynomial coefficient value, and the user's function to compute these polynomials.

Table 1.1 – Output data for the laboratory work №1

Variant	Phrase to compute the variable
1	2
1,2	$\dfrac{f_2(x)}{2(f_1(x)+f_3(x))}$
3,4	$\sin f_1^2(x)-f_2(x)+f_3(2x)$
5,6	$\sqrt{f_1^2(x)+f_2^2(x)}-3f_3(x)$
7,8	$\dfrac{(f_1(\sin x)+f_3(\cos x))^2}{f_2(x)}$
9,10	$\sqrt{\lvert f_1(x)-f_2(x)\rvert}+f_3(\frac{x}{2})$
11,12	$\dfrac{f_1^2(3x)-f_2(\frac{x}{3})}{3f_3(x)}$
13,14	$f_2(\sin x)-f_1(x)+\sin(f_3(\frac{x}{2}))$
15,16	$\dfrac{f_1(x)-2\sqrt{\lvert f_2(x)\rvert}}{5f_3(\frac{x}{5})}$
17,18	$(f_1^3(x)-f_2(\frac{x}{2}))f_3(5x)$

Continuation of the table 1.1

1	2
19,20	$\dfrac{f_1(e^x)-f_2(e^x)}{2(f_2(x)+f_3(x))}$
21,22	$\dfrac{f_2(x)}{\sqrt{f_1^2(\sin x)+f_3^2(\cos x)}}$
23,24	$\left\lvert f_1(\sin^2 x)f_3(e^x)f_2(2x)\right\rvert$
25,26	$e^{f_1(x)}-\dfrac{f_2(\frac{x}{2})}{f_3(2x)+1}$

1.3 Methodical Recommendations

If the programme is made up in the Pascal language, the coefficient mass for the functions $f_1(x)$, $f_2(x)$ *i* $f_3(x)$ in the programme and the coefficient mass to compute the value of the polynomial $P_n(x)$ in the user's function have to be of the same type described in the main module. For example:

```
const    nmax=6;
type     vector=array[1..nmax] of real,
```

where nmax – polynomial maximum order

It's better to mark the above-listed masses by different identifiers.

Laboratory Work 2

PRIMITIVE OPERATIONS WITH MATRICES

Purpose of the work: to learn to compute the matrix sum, difference, and scalar product, to transpose them and define the matrix norm.

2.1 Theoretical data

Matrices and vectors are often used in electrical computation (a vector is a matrix-line or a matrix-column). Examine the primitive operations with matrices: sum, difference, multiplication, transposition, some norm computing. The more complicated operation will be examined later.

2.1.1 Sum and difference of two matrices

The sum of two matrices of the same size

$$A+B=[a_{i,j}]+[b_{i,j}] \quad (i=1,2,\ldots,m;\ j=1,2,\ldots,n)$$

Is the matrix $C=[C_{i,j}]$ of the same size the elements of which are equal to the sums of the corresponding elements of the matrices A and B :

$$C_{ij}=a_{ij}+b_{ij}. \tag{2.1}$$

The difference of the matrices is computed similar to the sum, but the elements of the matrix which is deducted have the opposite sign, that is the elements of the matrix $C=B-A$ are computed according to the formula

$$C_{ij}=a_{ij}-b_{ij} \tag{2.2}$$

$$(i=1,2,\ldots,m;\quad j=1,2,\ldots,n).$$

2.1.2 The matrix scalar products and their exponentiation

The scalar product of the matrix A of the size $m\times k$ to the matrix B of the size $k\times n$ is the matrix C of the size $m\times n$ the elements of which are computed according to the formula:

$$C_{ij}=a_{i1}b_{1j}+a_{i2}b_{2j}+\ldots+a_{ik}b_{kj}=\sum_{l=1}^{k}a_{il}b_{jl} \tag{2.3}$$

It should be mentioned that the matrix C=$A\times B$ is defined only when the number of the columns of the matrix A is equal to the number of the lines in the matrix B.

For the matrix scalar product the commutative law is not applied, that is $AB\neq BA$.

The multiplication of the matrix A with the size $m\times k$ by the vector-column $\vec{B}$ which consists of k elements, and the vector-line A which consists of k elements by the matrix B with the size $k\times n$. $m\times k$ is the special case of matrix multiplication.

In the first case the result will be the vector –column with the elements

$$C_i = a_{i1}b_1 + a_{i2}b_2 + \ldots + a_{ik}b_k = \sum_{j=1}^{k} a_{ij}b_j \tag{2.4}$$

(i=1, 2,..., m) , and in the second case – the vector- line with the elements

$$C_j = a_1 b_{1j} + a_2 b_{2j} + \ldots + a_k b_{kj} = \sum_{i=1}^{k} a_i b_{ij} \ . \tag{2.5}$$

According to the term the matrix scalar product only the square matrix can be exponentiated into the integral positive rate:

$$A^k = \underbrace{((AA)\cdot A)\ldots\cdot A)}_{k\text{- factors}} \tag{2.6}$$

2.1.3 Matrix Transposition

If to replace the lines of the matrix A with the size $m \times n$ by the corresponding columns, we will get the matrix A^{T} with the size n×m which is called the transposed one relative to the matrix *A*.

Thus,

$$a_{ij}^{T} = a_{ji}$$

$$(i=1,2,\ldots,n;\ j=1,2,\ldots,m).$$

2.1.4 Matrix Norms

The norm of the matrix $A=[a_{ij}]$ is the real number $||A||$ which meets the following requirements:

- $||A|| \geq 0$ (and $||A=0||$ only when $A=[0]$),
- $||\alpha A||=|\alpha|*||A||$, where α - the real number (and $||-A||=||A||$),
- $||A||+||B|| \geq ||A+B||$,
- $||AB|| \leq ||A||*||B||$,
- $||A-B|| \geq |||B||-||A|||$.

The following three norms are considered the easiest to compute :

$$\|A\|_1 = \max_i \sum_j |a_{ij}| - \tag{2.7}$$

The maximum sum of the matrix element modules in the lines;

$$\|A\|_2 = \max_j \sum_i |a_{ij}| - \qquad (2.8)$$

The maximum sum of the matrix element modules in the columns;

$$\|A\|_3 = \sqrt{\sum_{ij} \left(|a_{ij}|\right)^2} - \qquad (2.9)$$

Square root of the sum of the squares of the modules of all matrix elements.

2.2 Tasks

Do the operations with matrices according to the phrases given in the table 2.1. The repeated actions should be presented as separate procedures:

$$A = \begin{pmatrix} 2 & 3.1 \\ 4.5 & 10.7 \\ 7 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 7.5 & 11 & 1.7 \\ 5 & 4 & 2 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 \\ 7 & 8 \\ 5 & 6 \end{pmatrix}, \quad D = \begin{pmatrix} 7.4 & 5 \\ 9 & 8 \end{pmatrix}.$$

Table 2.1

Number of the variant	Tasks
1	2
	Check the correlations::
1	$\|A+C\|_1 \le \|A\|_1 + \|C\|_1$
2	$\|A*B\|_1 \le \|A\|_1 * \|B\|_1$
3	$\|A-C\|_1 \ge \| \|C\|_1 - \|A\|_1 \|$
4	$\|A+C\|_2 \le \|A\|_2 + \|C\|_2$
5	$\|A*B\|_2 \le \|A\|_2 * \|B\|_2$
6	$\|A-C\|_2 \ge \| \|C\|_2 - \|A\|_2 \|$
7	$\|A+C\|_3 \le \|A\|_3 + \|C\|_3$
8	$\|A*B\|_3 \le \|A\|_3 * \|B\|_3$

The continuation of the table 2.1

1	2
9	$\|A-C\|_3 \geq \| \|C\|_3-\|A\|_3 \|$
10	Compute: $K=A+C+B^T$
11	$K=B(A+C)+D$
12	$K=(A-C)*(D*B)$
13	$K=B^T-A-C$
14	$K=B*A*B$
15	$L=\|A+C\|_1+\|B\|_1$
16	$L=\|A\|_1+\|B\|_2+\|C\|_1$
17	$K=C*D+A+C$
18	$K=B*C*D$
19	$\|A\|_1 \quad \|A^T\|_1 \quad \|B\|_1 \quad \|B^T\|_1$
20	$K=D*C^T*A$
21	$K=D-B*C-B*A$
22	$K=A^T+C^T+B$
23	$K=(A-C)*B$
24	$K=C*D-A-C$

Laboratory Work 3

SOLUTION OF THE SYSTEMS OF LINEAR EQUATIONS WITH REAL COEFFICIENTS

Purpose of the work: to learn to compute the roots of the systems of linear equations with real coefficients.

3.1 Theoretical Data

The solution of the systems of linear equations is used in electrical engineering and related to it disciplines when the static modes of the branched electric circuits are computed.

The system of the linear equations n with the unknown n looks like:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n. \end{cases} \quad (3.1)$$

It can be put down in the matrix form:

$$A * \overrightarrow{X} = \overrightarrow{B}, \quad (3.2)$$

Where $A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}$ - coefficient square matrix ;

$\vec{B} = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{pmatrix}$ - absolute term vector;

$\overrightarrow{X} = \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{pmatrix}$ - root required vector..

The ways of solution of the systems of linear equations are divided into two groups:

- accurate methods (coefficient matrix inversion, Cramer's law, Gaussian method etc.);
- iteration methods (Newton's, Seidel's, simple iterations etc.).

If the matrix A is not special, that is its determinant is not equal to zero, the system has the single solution:

$$\overrightarrow{X} = A^{-1} * \overrightarrow{B}, \tag{3.3}$$

Where A^{-1} - matrix inverse to the matrix A.

Computing of the roots according to the formula (3.3) is called the coefficient matrix inversion method.

According to the Cramer's law the roots are computed according to the formulae :

$$x_1 = \frac{\Delta_1}{\Delta}, x_2 = \frac{\Delta_2}{\Delta}, \ldots\ldots\ldots, x_n = \frac{\Delta_n}{\Delta},$$

Where Δ - determinant of the matrix A;

Δ_i - determinant of the matrices obtained from the matrix A by the replacement of its *i*-column by the vector of the free terms *B*.

Both of above-mentioned methods are used only when the systems of the equations of not high order are solved by hand. If $n>3$ these methods are too labour-consuming and not economical.

As to the accurate methods, that of Gaussian is the most widely spread.

It can be divided into two stages:

- the forward trace, that is gradual reduction of roots from 1st to *n* and transformation of the coefficient matrix to the rectangular;

- return trace that is gradual reduction of roots from 1st to *n* out of the transformed equation system..

The reduction of the *k* root (k=1, 2,..., n-1) out of the *i*-equation ($i=k+1$, $k+2,..., n$) is done by the replacement of all the coefficients of the *i*-equation by the difference between former coefficients of this equation and the corresponding coefficients of the *i*-equation multiplied by the measurement factor:

$$p = \frac{a_{ik}}{a_{kk}}. \tag{3.4}$$

As a result the coefficients of the *i*-equation have the following values:

$$a_{ik}=0, \tag{3.5}$$

$$a_{ij}=a_{ij}-p\,a_{kj} \quad (j=k+1,k+2,\ldots,n), \tag{3.6}$$

$$b_i=b_i-pb_k. \tag{3.7}$$

The mark "=" in the formulae (3.6) and (3.7) is used as the symbol of the assignment operation, with the former values of the coefficients a_{ij} and b_i used in the right-hand part , and the new ones – in the left-hand part.

When the minimal error roots are reduced the rounding off under the coefficient re-computation can be obtained by the replacement of the equations in such a way that the modules of the coefficients a_{kk} with the reduced roots x_k are maximum possible. This stage of the Gaussian method is called the choice of the main element. .

According to above-mentioned the scheme of the algorithm of the forward trace can be as presented in the fig. 3.1.

As a result of the forward trace the system of the equations (3.1) is transformed into:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ \qquad a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \qquad\qquad\qquad a_{nn}x_n = b_n. \end{cases} \tag{3.8}$$

The coefficients a_{ij} and b_i of the system (3.8) don't coincide with the corresponding coefficients of the output system (3.1).

The roots of the transformed system can be computed according to the formulae:

$$x_n = \frac{b_n}{a_{nn}}, \tag{3.9}$$

$$x_i = \frac{b_i - \sum\limits_{j=i+1}^{n} a_{ij}x_j}{a_{ii}} (i = n-1, n-2, \ldots, 1).$$

(3.10)

The return trace scheme can be as presented in the fig. 3.2.

The forward trace algorithm can be reduced if the square matrix of the coefficients A_{nxn} and the vector-column of the free terms $\vec{B}$ are combined into the single matrix $AR_{nx(n+1)}$, which is called the coefficient augmented matrix, and in which the elements of the vector $\vec{B}$ make up the (n+1)-column:

$$ar_{ij} = \begin{cases} a_{ij} & npu & j \le n, \\ b_i & npu & j = n+1. \end{cases} \tag{3.11}$$

In this case the blocks 9 and 13 will disappear from the scheme, and the parameter j (number of the column) of the blocks 7 and 14 will change not to *n*, but to *n+ 1*.The variables b_n i b_i of the blocks 1 and 6 (fig. 3.2) should be changed into the variables $a_{n,n+1}$ i $a_{i,n+1}$ correspondingly.

3 Tasks

.

Compute current and tension of the electric circuit branches given in the fig. 3.3 with the help of the Kirchhoff laws. The scheme parameters are given in the table 3.1. Test the results.

3.3 methodical Recommendations

If the Pascal programming language is used: it is more convenient to have the equation system root determination in the form of the programme (procedure) with the formal parameters *n, A, B* and *X*, with the massive *X* being described as the parameter-variable (var. being the key-word). If the augmented coefficient matrix is used, n, A and X. are the procedure formal parameters.

To describe the massive types in the part of constants the massive maximum permissible sizes should be identified. .

For the Gaussian method programme to be universal there should not be the process of the output data input and the result output in it.

They main programme module should have the output data input, forming of the actual parameters for the Gaussian method sub-programme, this sub-programme call, the result output, the solution verification.

To verify the values of the functions can be computed and displayed.

$$f_i = a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n - b_i \qquad (3.12)$$

$$(i=1, 2, \ldots, n).$$

If the solution is correct these values are near-zero

The matrix output and the result test can be arranged as separate sub-programmes.

Table 3.1 Scheme Parametres

Number of the Variant	B	B	B	B	B	B	Ом	Ом	Ом	Ом	Ом	Ом
1-6	130	500	120	240	170	380	21	14	13	16	9	20

7-12	360	190	210	130	450	170	8	9	16	13	21	12
13-18	120	220	340	80	510	160	5	18	12	14	7	28
19-24	280	540	310	160	90	360	12	6	24	10	14	18
25-31	340	110	280	210	130	260	27	30	4	6	22	11

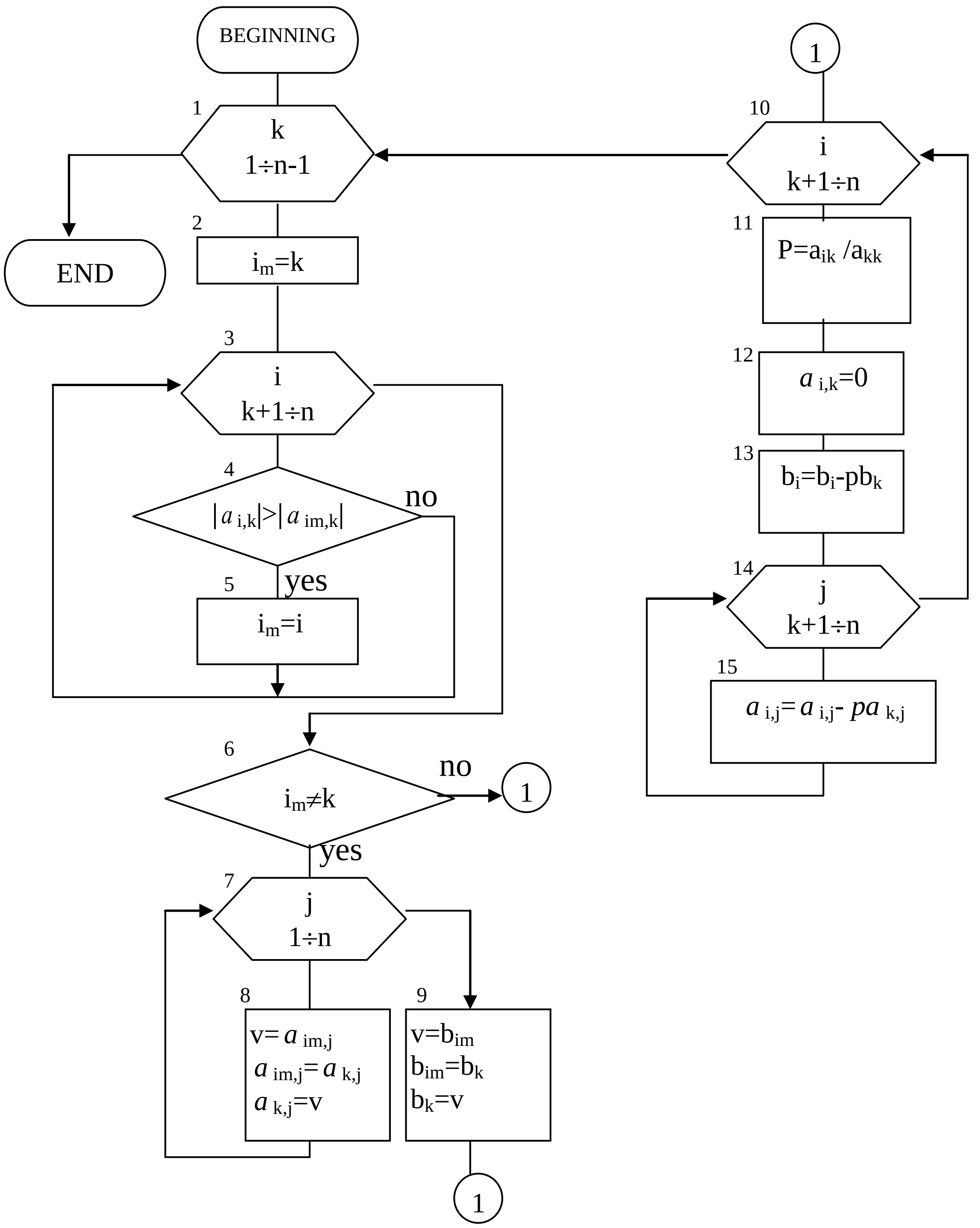


Figure 3.1 – Gaussian Method Forward Trace

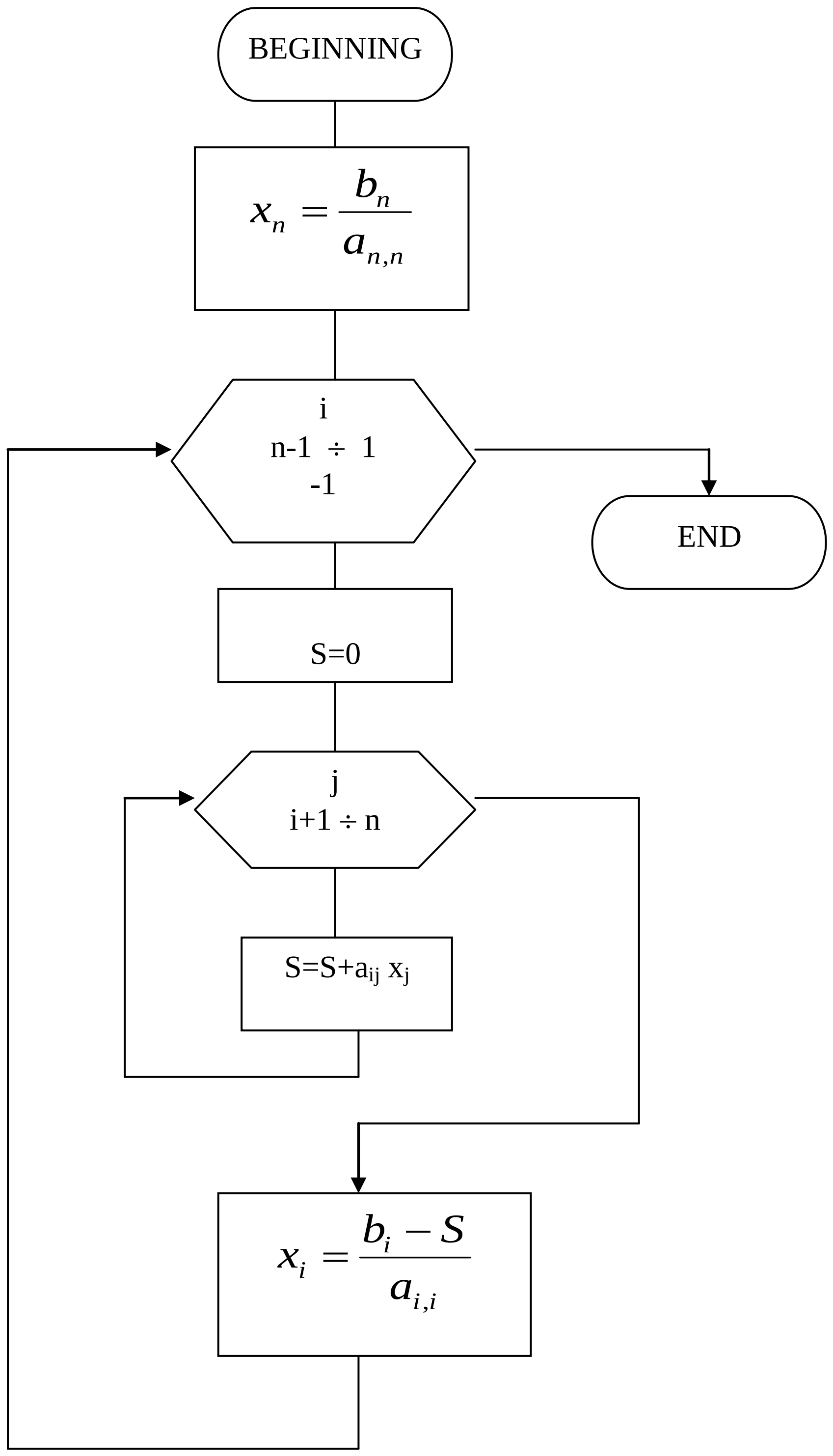


Figure 3.2 – Gaussian Method Return Trace

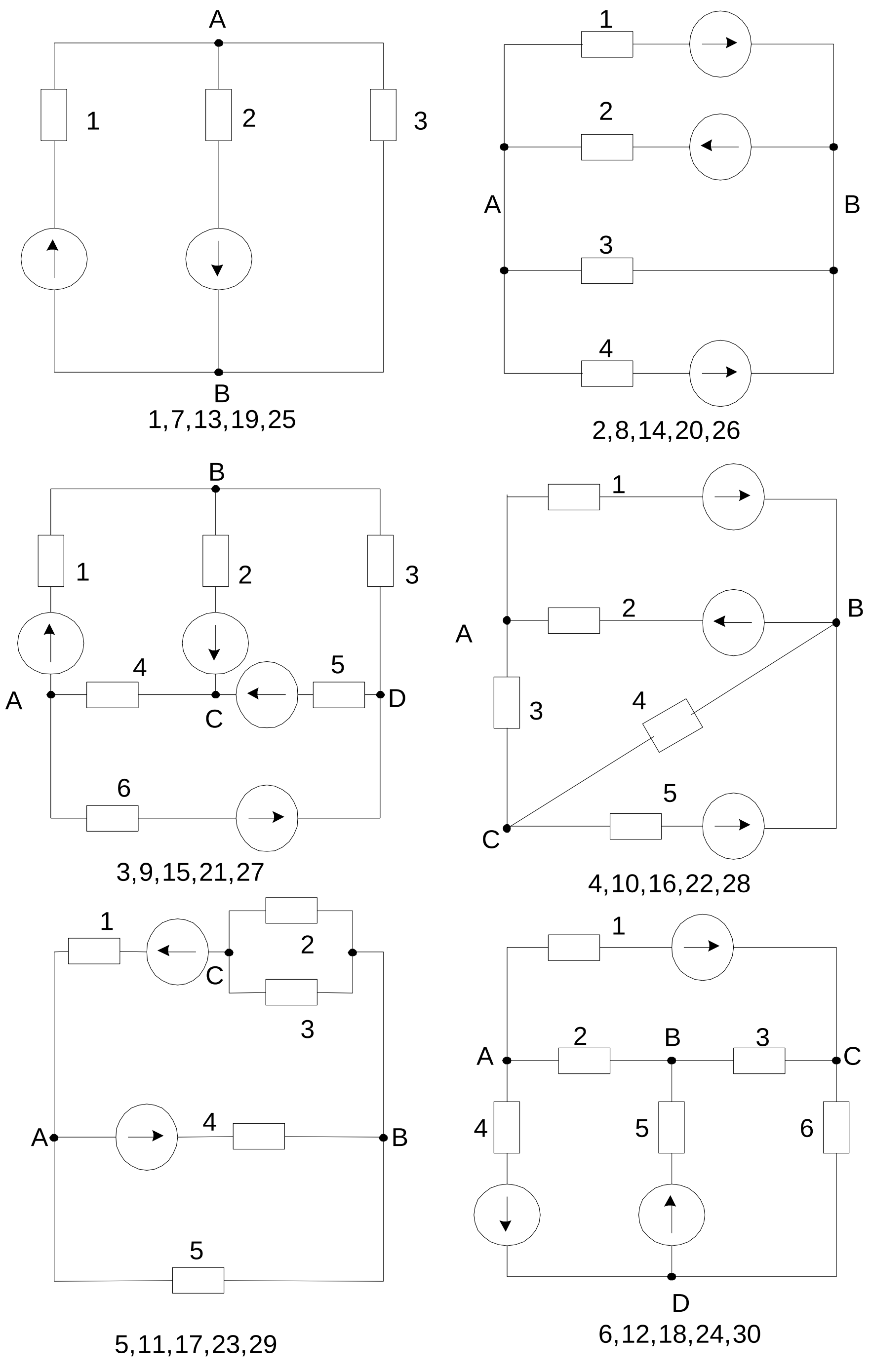


Figure 3.3 – Schemes for the Task 3.2

Laboratory Work 4

SOLUTION OF THE LINEAR EQUATION SYSTEMS WITH THE COMPLEX COEFFICIENTS

Purpose of the work: to learn to compute static modes in the branched electric circuits.

4.1 Theoretical Data

If there are resistor, dc sources, resistance coils, capacitors and ac sources in the electric circuit, to compute current and tension in constant modes the equation systems with the complex coefficients are solved. If the element static characteristic nonlinearity is not taken into account, the algebraic equation linear system is obtained. To solve it all methods mentioned in the previous laboratory work, including the Gaussian method, are applied.

The specificity of the solution is in fact that we operate with the complex numbers rather than with real ones. Such algorithmic languages as FORTRAN and PL-1 have the complex type data and operate with it as easily as with the real type arithmetic data.

When the Pascal language is used the programmer has to make up the sub-programme to do the operations with the complex numbers. It can be done much easier due to the possibility to create the types identified by the user, and the presence of the formal and actual parameter device in the sub-programmes. For example, in the Pascal programme description part the data complex type can be defined as the record which consists of two parts: real (re) and imaginary

```
type complex = record
      re, im: real
      end;
```

Then a number of sub-programmes and functions to work with the complex numbers are made up. For example, the sub-programme of multiplication of two complex numbers $X=X_{re}+jX_{im}$ i $Y=Y_{re}+jY_{im}$ can be as follows:

```
procedure MultC(x,y:complex; var z:complex);
begin
   z.re:=x.re*y.re-x.im*y.im;
   z.im:=x.re*y.im+x.im*y.re;
end;
```

If such a programme is available to compute the value of the variable

$$W=(3,6-j8)(5+j2,1)$$

It is enough to record the operator sequence in any programming module for which the MultC procedure is available:

```
with a do
  begin re:= 3.6; im:= -8 end;
with b do
  begin re:= 5; im:= 2.1 end;
MultC (a,b,w);
```

The variables a,b,w are preliminary described as the complex type data:

```
var a,b,w:complex.
```

Table 4.1- Scheme Parametres

Number of the Variant	Гн	мкФ	В	В	Ом	Ом	Ом
1-6	0,05	170	360	200	18	10	27
7-12	0,08	150	400	250	30	13	20
13-18	0,07	270	200	320	18	21	32
19-24	0,09	130	220	440	28	20	16
25-31	0,06	180	320	240	25	16	34

4.2 Task

Compute constant current of the electric circuit branches given in the fig. 4.1, making up the equation system with the Kirchhoff laws. The scheme parameters are given in the table 4.1. Verify the results.

4.3 Methodical Recommendations

To do the work, use the recommendations and algorithm schemes given in the previous work. Replace the complex coefficient operations by the call of the corresponding sub-programmes (procedures) or functions.

The following procedures and functions should be added:

- input and output of the values of complex variables or their arrays,
- multiplication, division, adding of two complex numbers,
- complex number sign change,
- complex number module computing function.

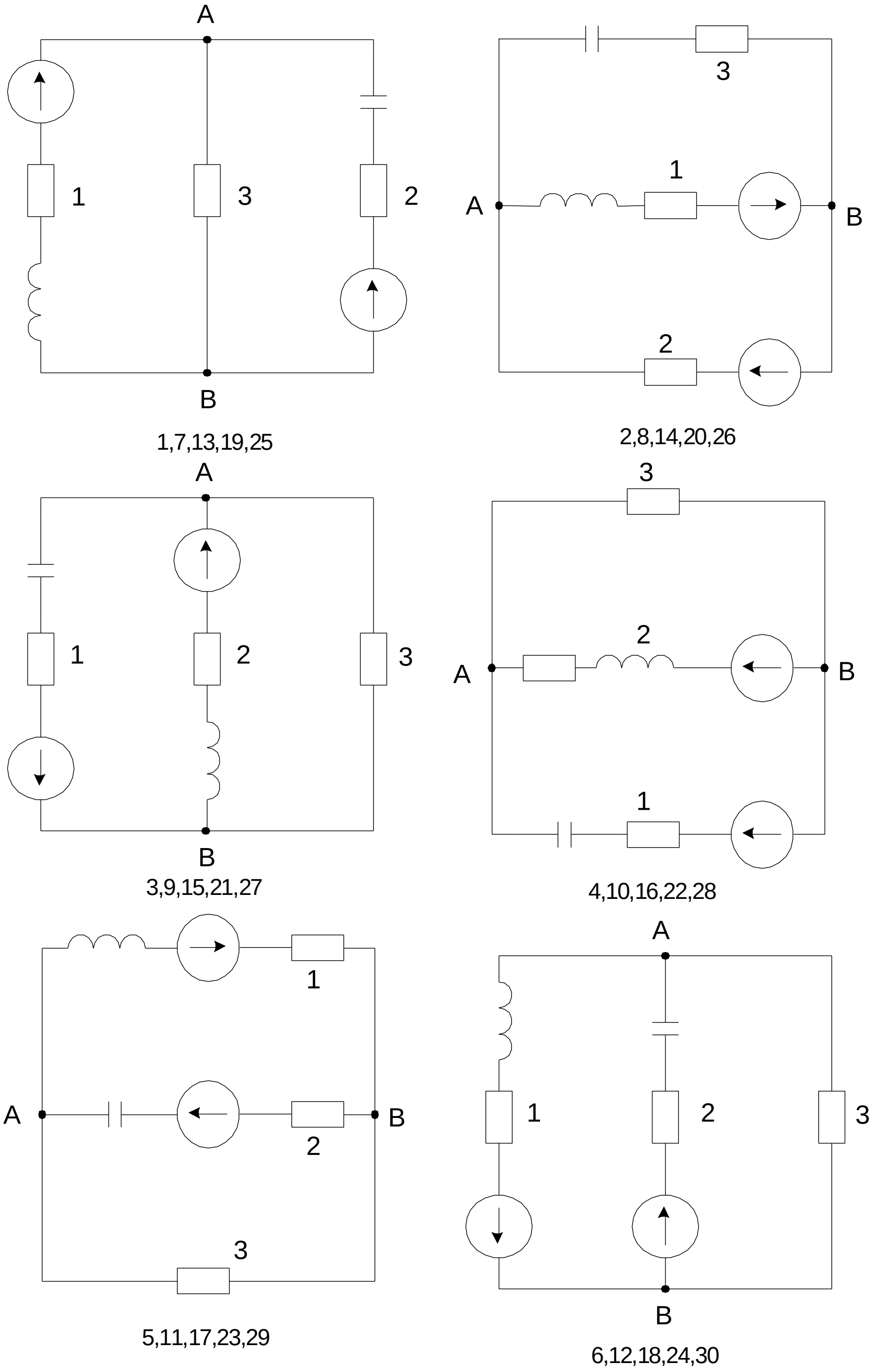


Рисунок 4.1 – Scheme variants

Laboratory Work 5

MATRIX INVERSION

Purpose of the work: to learn to compute the inverse matrix with respect to the given one

5.1 Theoretical Data

The matrix inversion is widely used when the branched electric circuits are computed in the form of matrix by different methods.

The matrix

$$X_{nxn}=A^{-1}_{nxn}, \quad (5.1)$$

Is called the inverse with respect to the output square matrix which being multiplied by the output one gives the identity diagonal matrix E_{nxn}:

$$A_{nxn}*A^{-1}_{nxn}=E_{nxn}, \quad (5.2)$$

Or in the expanded form

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} * \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}. \quad (5.3)$$

For the small size matrices ($n \le 3$) the inversion is made by hand with the application of the formula

$$A^{-1} = \frac{\tilde{A}^T}{\Delta}, \quad (5.4)$$

Where A - the union matrix (the matrix composed of the algebraic complements);

Δ - determinant .

With n>3 the computations according to the formula (5.4) become very cumbersome.

As it is vivid from (5.3), the elements of the KC column of the inverse matrix X can be determined by the solution of the system of linear equations n with the n unknowns.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} * \begin{pmatrix} x_1, KC \\ x_2, KC \\ \dots \\ x_n, KC \end{pmatrix} = \begin{pmatrix} e_1, KC \\ e_2, KC \\ \dots \\ e_n, KC \end{pmatrix}, \quad (5.5)$$

Where $e_{i,KC} = \begin{cases} 0 & npu \quad i \ne KC, \\ 1 & npu \quad i = KC, \end{cases}$

(5.6)

KC=1, 2, ..., n.

Thus, to determine all the elements of the inverse matrix *n* equation systems should be solved.

This approach is used in machine computations. The equation systems can be solved by any of the known methods, by the Gaussian one, for example.

If there is the equation system solution sub-programme, the matrix inversion algorithm can be represented by the scheme (fig. 5.1).

If there is no equation system solution sub-programme, the forward trace of the Gaussian method is done once with the expanded matrix *A* composed of the output matrix *AI* and the identity square matrix *E* joined to it on the left:

$$A = \begin{pmatrix} ai_{11} & ai_{12} & \dots & ai_{1n} & 1 & 0 & \dots & 0 \\ ai_{21} & ai_{22} & \dots & ai_{2n} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ ai_{n1} & ai_{n2} & \dots & ai_{nn} & 0 & 0 & \dots & 1 \end{pmatrix} \tag{5.7}$$

or

$$a_{ij} = \begin{cases} ai_{ij} & при \quad j \le n, \\ 1 & при \quad j = n + i, \\ 0 & в \quad інших\ випадках, \end{cases} \tag{5.8}$$

$$i=1, 2, ..., n,$$
$$j=1, 2, ..., 2n.$$

In the forward trace scheme as compared to the algorithm of the fig. 5.1, there are no blocks 9 and 13, and in the blocks 7 and 14 the final value of the variable j is equal to *2n*.

The return trace is done *n* times (with *КС=1, 2,..., n*). Along with it the elements of the vector of the roots x_n, x_j і x_i in the scheme of the fig. 3.2 are replaced by the elements of the inverse matrix $x_{n,KC}$ $x_{j.KC}$ і $x_{i,KC}$, and variables b_n і b_i – by the variables $a_{n,\ n+KC}$ і $a_{i,n+KC}$ correspondingly.

5.2 Task

Do the inversion of free square matrices of the second, third, and fourth order with the result verification.

5.3 Methodical Recommendations

Input, output, matrix inversion, and the result verification should be arranged as separate procedure blocks.

To verify the solution correctness, display the output matrix scalar product and the obtained inverse matrix. If the identity diagonal matrix (see the equation (5.3)) is obtained, the solution is correct.

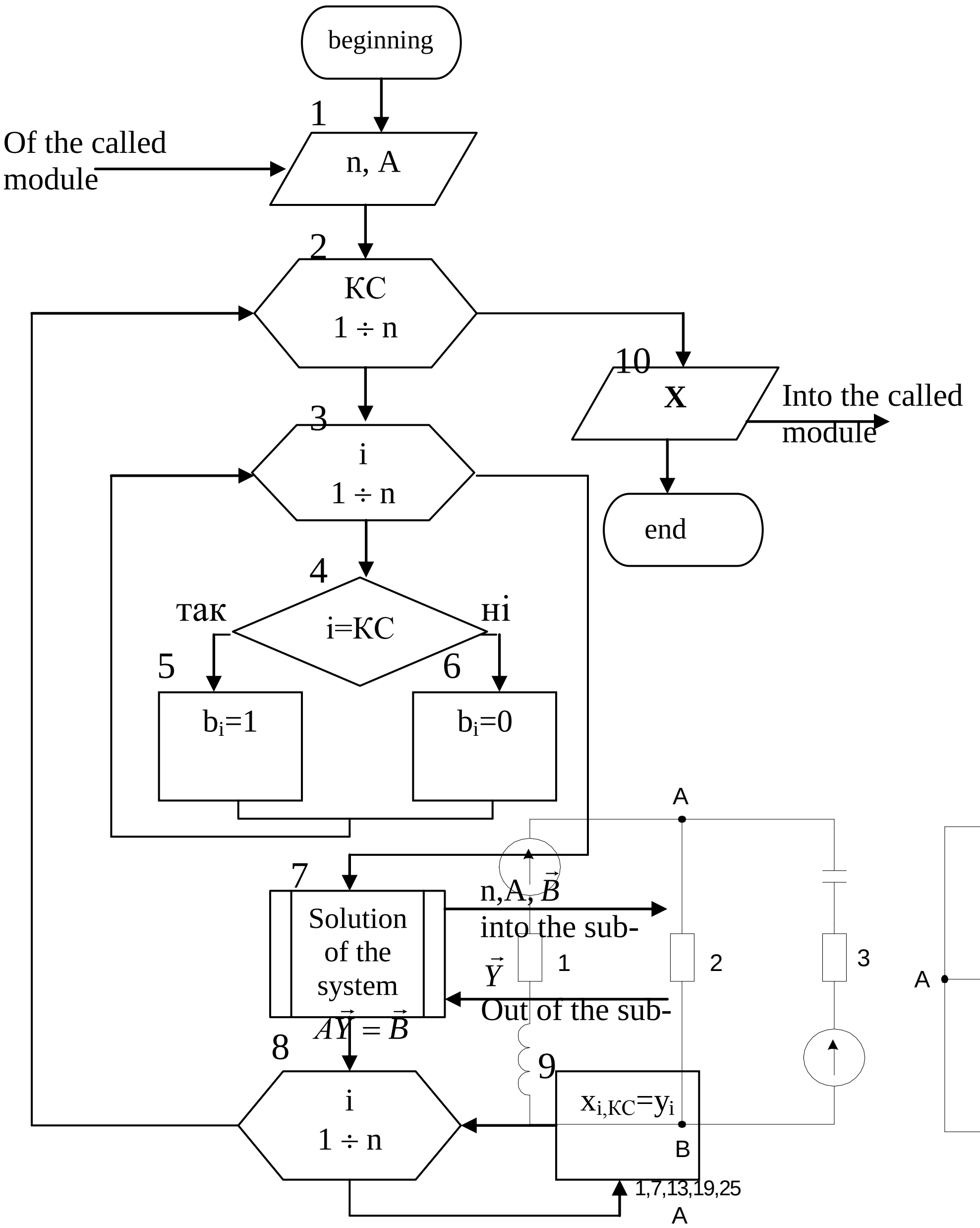
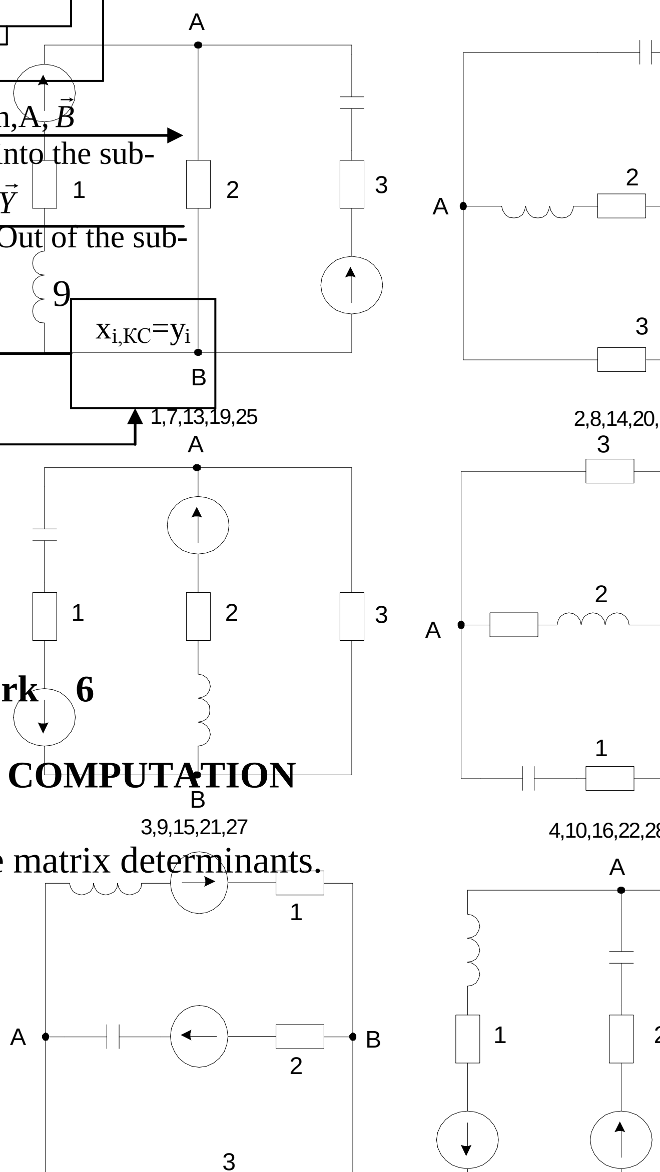


Fig. 5.1 – Matrix Inversion



Laboratory Work 6

MATRIX DETERMINANT COMPUTATION

Purpose of work: to learn to compute the matrix determinants.

6.1 Theoretical Data

To compute the determinant the output matrix is transformed into the triangular with the help of the Gaussian method forward trace, and its diagonal element product is computed.

If the disarray is used in the forward trace scheme, the fact that one replacement of the kind changes the determinant sign into the opposite one is taken into account.

6.2 Task

Solve the equation system given in the table 6.1 by the Kramer method. Verify the results.

Table 6.1- Output data

Number of the Variant	Equation System
1	2
1,2	$3.14\,x_1-2.2x_2+1.17x_3=1.27$ $-2.12x_1+1.32x_2-2.45x_3=2.13$ $1.17x_1-2.45x_2+1.18x_3=3.14$
3,4	$2.45x_1+1.75x_2-3.24x_3=1.23$ $1.75x_1-1.16x_2+2.18x_3=3.43$ $-3.24x_1+2.18x_2-1.85x_3=-0.16$
5,6	$1.65x_1-2.27x_2+0.18x_3=2.25$ $-2.27x_1+1.73x_2-0.46x_3=0.93$ $0.18\,x_1-0.46x_2+2.16x_3=1.33$
7,8	$3.23x_1+1.62x_2+0.65x_3=1.28$ $1.62x_1-2.33x_2-1.43x_3=0.87$ $0.65x_1-1.43x_2+2.18x_3=-2.87$
9,10	$0.93x_1+1.42x_2-2.55x_3=2.48$ $1.42x_1-2.87x_2+2.36x_3=-0.75$ $-2.55x_1+2.36x_2-1.44x_3=1.83$

11,12	$1.42\,x_1 - 2.15x_2 + 1.07x_3 = 2.48$ $-2.15x_1 + 0.76x_2 - 2.18x_3 = 1.15$ $1.07x_1 - 2.18x_2 + 1.23x_3 = 0.88$

Continuation of the table 6.1

1	2
13,14	$2.23x_1 - 0.71x_2 + 0.63x_3 = 1.28$ $-0.71x_1 + 1.45x_2 - 1.34x_3 = 0.64$ $0.63\,x_1 - 1.34x_2 + 0.77x_3 = -0.87$
15,16	$1.63x_1 + 1.27x_2 - 0.84x_3 = 1.51$ $1.27x_1 + 0.65x_2 + 1.27x_3 = -0.63$ $-0.84x_1 + 1.27x_2 - 1.21x_3 = 2.15$
17,18	$0.78x_1 + 1.08x_2 - 1.35x_3 = 0.57$ $1.08x_1 - 1.28x_2 + 0.37x_3 = 1.27$ $-1.35x_1 + 0.37x_2 + 2.86x_3 = 0.47$
19,20	$0.83x_1 + 2.18x_2 - 1.73x_3 = 0.28$ $2.18\,x_1 - 1.41x_2 + 1.03x_3 = -1.18$ $-1.73x_1 + 1.03x_2 + 2.27x_3 = 0.72$
21,22	$2.74x_1 - 1.18x_2 + 1.23x_3 = 0.16$ $-1.18x_1 + 1.71x_2 - 0.52x_3 = 1.81$ $1.23x_1 - 0.52x_2 + 0.62x_3 = -1.25$
23,24	$1.35x_1 - 0.72x_2 + 1.81x_3 = 0.88$ $-0.72x_1 + 1.45x_2 - 2.18x_3 = 1.72$ $1.38x_1 - 2.18x_2 + 0.93x_3 = -0.72$

Laboratory Work 7

MATRIX INVERSION

Purpose of work: to revise the function construction graphs, and learn to locate the roots of transcendental equations.

7.1 Theoretical Data

The equation

$$f(x)=0 \tag{7.1}$$

is called transcendental if it has trigonometrical or other special functions (exponent, logarithmic etc.) of the variable x .

The transcendental equations have both indeterminative and infinite number of solutions.

If the equations do not have analytical solution, they are solved by the iteration methods. The root location, that is determination of the interval of the root existence and its initial estimate, is the first stage of such task solution.

To locate the transcendental equation real roots, it is often enough to construct the graph of the function $f(x)$ or to transform the output equation $f(x)=0$ into $\varphi_1(x)=\varphi_2(x)$, construct the graphs of two functions $\varphi_1(x)$ i $\varphi_2(x)$ and determine the sphere of the point of their intersection. The required interval of the existence of the root [*a*,*b*] should meet the requirement of

$$f(a)*f(b)<0. \tag{7.2}$$

7.2 Task

Locate the first positive root of the transcendental equation given in the table 7.1 using the graphic method.

7.3 Methodical Recommendations

The left search limit $x_н$ equals to zero (if $f(0)$ exists) or to the nearest to it positive value, for example 0.01 or 0.1. The right search limit is free and depends on the function $f(x)$. It should not be more than 5 for the table equations. As to the initial value of the search step Δx, no more than 20 function values are computed at the search interval. Display them.

If the change of the sign of the function $f(x)$ does not take place, reduce Δx and (or) increase $x_к$, and repeat the computation.

To measure the graph determine (with the programme or visually) the function minimal and maximal value, the graph limits on the basis of this data, and construct the graph.

Use the graph not only to locate the root, but also to pick out the most appropriate method of its refinement.

The function f*(x)* is determined as the user's function

Table 7.1 – tasks for the laboratory work №7

№	Equations	Solution method
1	2	3
1	$2\sin(x+\pi/3)-0.5x^2+1=0$	Tangent
2	$\cos(x+0.3)-x^2=0$	Tangent
3	$tg^3x-x+1=0$	Tangent
4	$2arctg\,x-x+3=0$	Tangent
5	$(x+3)\cos x-1=0$	Tangent
6	$tg(0.58x+0.1)-x^2=0$	Tangent
7	$\ln x-\frac{7}{2x+6}=0$	Chord
8	$\frac{1}{tg1.05x}-x^2=0$	Bisection
9	$2\ln x-x/2+1=0$	Bisection
10	$\ln x-1/x^2=0$	Bisection
11	$4.3\sin 4x-3.5x=0$	Chord
12	$2^x-2^{(x-2)}-1=0$	Bisection
13	$\cos(15.6x)+0.5=0$	Bisection
14	$0.5^x+1-(x-2)^2=0$	Tangent
15	$3^{(x-1)}-2-x=0$	Chord
16	$x^2\cos 2x+1=0$	Chord

17	x^2-$2^{(x-1)}$=0	Tangent
18	*5sinx-x=0*	Bisection
19	*arctg(x-1)+2x=0*	Chord
20	$(x-2)^2$-2^x=0	Tangent

Continuation of the table 7.1

1	2	3
22	$2e^x$-5x-2=0	Tangent
23	*cos(x+0.5)*-x^3=0	Tangent
21	x^2-20sin x=0	Chord
24	*2arctg x-1/2*x^3=0	Chord
25	e^{-x}+x^2-2=0	Chord

Laboratory Work 8

LOCATION OF ROOTS OF ALGEBRAIC EQUATIONS

Purpose of work: to learn to locate roots of algebraic equations..

8.1 Theoretical Data

Algebraic equations of *n*-order

$$a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + a_n = 0 \qquad (8.1)$$

Have n roots

.

When the location of roots of algebraic equations takes place their following characteristics should be taken into account:

1) n roots of the algebraic equations of n-order can be real or complex:;

2) if all the coefficients a_i are real, all complex roots make up complex conjugate pairs;;

3) the number of the positive real roots equals or is less than the number of the sign change of the sequence of the coefficients a_i of the polynomial $f(x)$;

4) the number of the negative real roots equals or is less than the number of the sign change of the sequence of the coefficients of the polynomial $f(-x)$;

5) if $f(x)$ possesses the value of different signs at the ends of the segment[a,b], that is $f(a)*f(b)<0$, there is at least one root in the middle of this segment. It is integrated if the derivative $f'(x)$ keeps the constant sign in the middle of the [a,b]. постійний знак;

6) the top $R_в$ and bottom $R_н$ limits of the positive R^+ and negative R^- of the real roots can be computed by the Lagrange theorem :

$$R_в^+ = 1 + \sqrt[k]{B/a_0}, \tag{8.2}$$

where k – number of the first of the negative coefficients of the equation (8.1) with $a_0>0$;

B – the biggest of the absolute values of the negative coefficients :

$$R^+_H=1/R_1, \quad R^-_H=-R_2, \quad R^-_B= -1/R_3,, \tag{8.3}$$

where R_1, R_2, R_3- variables computed according to the formula (8.2) for the corresponding auxiliary equations:

$$\begin{cases} f_1(x)=x^n f(1/x)=0, \\ f_2(x)=f(-x)=0, \\ f_3(x)=x^n f(-1/x)=0. \end{cases} \tag{8.4}$$

If R^-_H and R^+_B, are known every root can be computed by the algorithm of the fig. 8.1.

8.2 Task

Locate each real root of the algebraic equation $f(x)=0$ for $f(x)$ given in the table 8.1. Construct the graph of the function $f(x)$ on the interval [R^-_H, R^+_B].

8.3 Methodical Recommendations ї

Save the programme of the root location and use it as the first part of the programme of the equation solution to refine roots in the laboratory works.

Table 8.1 – tasks for the laboratory work №8

№	$f(x)$	Solution Method
1	$4.2x^3-31.92x^2+74.3x-51.87$	Bisection
2	$3.6x^3-172.8x^2+5.184x-237.32$	Chord
3	$5.8x^3-47.56x^2+121.2x-97.02$	Bisection
4	$6.1x^3-90.28x^2+388.2x-506.2$	Chord
5	$3.6x^3-39.96x^2+12.17x+426.4$	Bisection
6	$2.7x^3-37.26x^2+16.71x-202.7$	Chord

7	$1.3x^3$-$5.98x^2$-$1.09x$+13.76	Simple iteration
8	$4.5x^3$-$26.1x^2$+$176.6x$-112.4	Simple iteration
9	$5.1x^3$-$62.22x^2$+$142.7x$+109.2	Simple iteration
10	$1.6x^3$-$3.04x^2$-$29.18x$+8.98	Simple iteration

Continuation of the table 8.1

1	2	3
11	$-2.3x^3+0.23x^2+17.05x+13.48$	Simple iteration
12	$1.6x^3-14.24x^2+38.13x-29.02$	Simple iteration
13	$5.3x^3-36.04x^2+12.25x+28.05$	Simple iteration
14	$-2.6x^3+4.68x^2+14.38x+3.822$	Bisection
15	$-1.5x^3-14.25x^2-37.98x-22.03$	Simple iteration
16	$3.4x^3-46.58x^2+127.3x-60.34$	Simple iteration
17	$2.8x^3-25.76x^2+6.18x+107.4$	Bisection
18	$-1.4x^3-10.78x^2-22.54x-11.85$	Simple iteration
19	$3.1x^3-62.6x^2+414.7x-898.9$	Simple iteration
20	$1.6x^3-12.48x^2+25.04x-8.12$	Bisection
21	$5.4x^3-54x^2+140.6x-73.8$	Simple iteration
22	$2.7x^3-17.6x^2-45.4x+123$	Bisection
23	$-1.8x^3-5.58x^2+1.5x+119$	Bisection
24	$-2.5x^3+8.25x^2+61.9x-117$	Simple iteration

Laboratory Work 9

Refinement of the roots of transcendental and algebraic equations

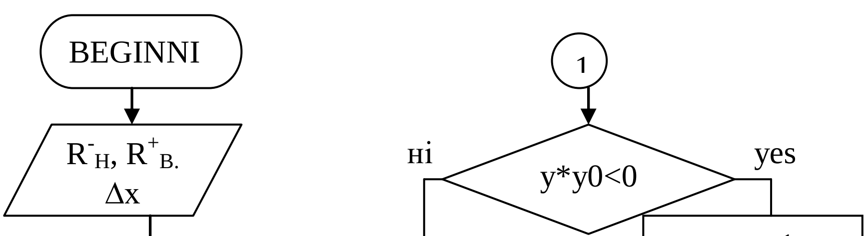
Purpose of work: to learn to solve transcendental and algebraic equations..

9.1 Theoretical Information

Computational solution of the equation

$$f(x)=0 \tag{9.1}$$

has two stages: root location and refinement of their initial approximation by the iteration methods.



The bisection, chord, tangent, and simple iterations methods are the most accepted ones to refine roots.

9.1.1 Bisection Method

The Bisection Method or the method of half-division consists in repeated halving of the section which has a root:

$$x = \frac{a+b}{2} \;,$$

Where a i b – left and limit of the root, that is

$$b>a, \tag{9.2}$$

$$f(a)*f(b)<0. \tag{9.3}$$

To do every subsequent division the half of the section at the ends of which the function has the opposite sign is chosen. Along with it the interval of the root existence converges at the expense of change of one of its limits: left ($a = x$) or right ($b = x$).

The iteration division process finished when the condition

$$b\text{-}a \leq \varepsilon, \tag{9.4}$$

is done

where ε - given accuracy of the root computing. Sometimes the (9.4) is required to be done simultaneously with:

$$|f(x)| \leq \varepsilon. \tag{9.5}$$

The Bisection Method is a simple and reliable method to search simple roots of the equation f$(x) = 0$. It runs together with any continuous function $f(x)$, including those which are not differentiated. The speed of coincidence is not high. To obtain accuracy

$$N \approx log_2((b\text{-}a)/\,\varepsilon) \tag{9.6}$$

Iterations are spent. This means that to get every 3 correct decimal digits about 10 iterations are done.

If there are some roots on the section [*a*,*b*] the process coincides to one of them. The Method is not used to search the multiple root of the paired order.

9.1.2 Chord Method

The Chord Method or the method of proportional parts consists in subsequent subdivision of the section [*a*, *b*] which has a root into the parts proportional to the function values at the end of the section:

$$\frac{f(a)}{f(b)} = \frac{a - x}{b - x}, \tag{9.7}$$

So

$$x = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}. \tag{9.8}$$

Geometrically it is equivalent to the replacement of the graph of the function $f(x)$ by the chord that passes through the points $(a, f(a))$ and $(b, f(b))$.

To finish the iteration process the condition

$$|x_{i+1} - x_i| \leq \varepsilon, \tag{9.9}$$

Is used instead of the condition (9.4)

Where x_{i+1}, x_i – the last computed and previous approximation of the root In other cases this method is analogous to the Bisection Method but provides faster coincidence.

9.1.3 Tangent Method

The Tangent Method or the Method of Newton consists in subsequent approximation of the function $f(x)$ by the tangents to curve in the point of the previous approximation $(x_i, f(x_i))$, which cross the entire abscissa in the point of the next approximation x_{i+1} determined by the formula

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \tag{9.10}$$

The sequence (9.9) keeps to the real value of the root of the equation $f(x)=0$, if the initial approximation of the root belongs to the interval $[a, b]$ $(f(a)*f(b)<0)$, on which the derivatives $f'(x)$ and $f''(x)$ save their sign and the condition

$$f(x_0)*f''(x_0)>0. \tag{9.11}$$

is met.

The iterations are stopped when the conditions (9.9) and (or) (9.5) are met.

The Newton's Method is efficient if the good initial approximation to the root is known and the graph of the function has large slope at the root environs. In preferable cases the number of correct decimal signs in the next approximation is doubled, that is the process coincides very quickly.

The necessity to compute not only the function value but also the derivative value in every point is the method's drawback.

9.1.4 Simple Iteration Method

The Simple Iteration method consists in replacement of the initial equation $f(x)=0$ by the equivalent to it equation

$$x=\varphi(x) \tag{9.12}$$

and computed sequences

$$x_{i+1}=\varphi(x_i) \tag{9.13}$$

(i=1, 2, 3,...), which coincide to the accurate solution $i \to \infty$.

Iterations are stopped if the condition

$$|x_{i+1}-x_i|\le\varepsilon. \qquad (9.14)$$

is met.

$$|\varphi`(x)|<1. \qquad (9.15)$$

is a sufficient and necessary condition of the method coincidence.

The coincidence speed is increased when $|\varphi`(x)|$ is reduced.

9.2 Task

Compute the first positive root of the transcendental equation from the table 7/1 and all real roots of the algebraic equation from the table 8.1 given in the tables by he methods with the accuracy of $10^{-4}, 10^{-5}$ i 10^{-6}.

9.3 Methodical Recommendations

Determine the intervals $[a, b]$ for each root or the initial approximations of the roots x_0 using the results of the works 7 and 8.

Before the tangent or simple iteration methods are applied, check their coincidence.

Display the computation results of each operation.

To verify the solution, display not only progressive approximation of the roots but also the value of the function *f(x)* in these points.

Evaluate the speed of coincidence of different methods.

Laboratory Work 10

SOLUTION OF THE NON-LINEAR EQUATION SYSTEMS

Purpose of work: to learn to solve non-linear equation systems by the iteration methods.

10.1 Theoretical Data

The system of n equations with the unknown n is of the form :

$$\begin{cases} f_1(x_1,x_2,\ldots,x_n)=0, \\ f_1(x_1,x_2,\ldots,x_n)=0, \\ \ldots\ldots\ldots\ldots\ldots\ldots.. \\ f_n(x_1,x_2,\ldots,x_n)=0. \end{cases} \qquad (10.1)$$

Iteration methods are used to solve the non-linear equation systems.

Some of them are as follows:

10.1.1 Simple Iteration Method

To apply the method the initial eqaution system is transformed into:

$$\begin{cases} x_1 = \varphi_1(x_1, x_2, \ldots, x_n), \\ x_2 = \varphi_2(x_1, x_2, \ldots, x_n), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n = \varphi_n(x_1, x_2, \ldots, x_n). \end{cases} \qquad (10.2)$$

If the initial approximations of the roots are

$$\vec{X}^{[0]} = (x_1^{[0]}, x_2^{[0]}, \ldots, x_n^{[0]}), \qquad (10.3)$$

To make them more accurate the following formulae are used :

$$\begin{cases} x_1^{[k]} = \varphi_1(x_1^{[k-1]}, x_2^{[k-1]}, \ldots, x_n^{[k-1]}), \\ x_2^{[k]} = \varphi_2(x_1^{[k-1]}, x_2^{[k-1]}, \ldots, x_n^{[k-1]}), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n^{[k]} = \varphi_n(x_1^{[k-1]}, x_2^{[k-1]}, \ldots x_n^{[k-1]}), \end{cases} \qquad (10.4)$$

Where k=1,2,3,…,- the iteration number

The iterations are stopped if the condition

$$\max_i \left| x_i^{[k]} - x_i^{[k-1]} \right| \le \varepsilon, \qquad (10.5)$$

Is met

де ε - the admissible error of the results

The sufficient conditions of the coincidence of the iteration process are in the form:

$$\sum_{j=1}^{n} \frac{\partial \varphi_j(x_1^{[0]}, x_2^{[0]}, \ldots, x_n^{[0]})}{\partial x_i} < 1, \qquad (10.6)$$

or

$$\sum_{j=1}^{n} \frac{\partial \varphi_i(x_1^{[0]}, x_2^{[0]}, \ldots, x_n^{[0]})}{\partial x_j} < 1.$$

They are done for all the values *i (i =1,2,... ,n).*

10.1.2 Seidel Method

The Seidel Method differs from the simple iteraion method only by the formulae of the root refinement:

$$\begin{cases} x_1^{[k]} = \varphi_1(x_1^{[k-1]}, x_2^{[k-1]}, \dots, x_n^{[k-1]}), \\ x_2^{[k]} = \varphi_2(x_1^{[k]}, x_2^{[k-1]}, \dots, x_n^{[k-1]}), \\ x_n^{[k]} = \varphi_n(x_1^{[k]}, x_2^{[k]}, \dots, x_n^{[k-1]}). \end{cases} \tag{10.7}$$

In the majority of cases it provides fasteк coincidence of the iteration process.

10.1.3 Newton’s Method

The Newton’s Method derives from the Tangent Method for one equation.

The vector of increase of the roots $\Delta\vec{X}$ at every step of the iteration process is defined by solution of the system *n* of linear equations with *n* unknowns:

$$W^{[k-1]} \cdot \Delta\vec{X} = -\vec{F}(\vec{X}^{[k-1]}), \tag{10.8}$$

where $W = \dfrac{\partial \vec{F}}{\partial \vec{X}} = \begin{pmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \dots & \dfrac{\partial f_1}{\partial x_n} \\ \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \dots & \dfrac{\partial f_2}{\partial x_n} \\ \dfrac{\partial f_n}{\partial x_1} & \dfrac{\partial f_n}{\partial x_2} & \dots & \dfrac{\partial f_n}{\partial x_n} \end{pmatrix}$ (10.9)

Jacobian matrix;

$\vec{F}(\vec{X})$ - vector of the right parts of the output system of equations (10.1).

Root refinement is made according to the formula:

$$\vec{X}^{[k]} = \vec{X}^{[k-1]} + \Delta\vec{X}. \tag{10.10}$$

Iterations are stopped when the condition(10.5) is met. To have more strict verification the condition

$$\max_i \left| f_i(x_1^{[k]}, x_2^{[k]}, \ldots, x_n^{[k]}) \right| \le \varepsilon . \qquad (10.11)$$

Can be verified together with the condition (10.5).

10.2 Task

Solve the non-linear equation system with the initial approximation taken from the table by the given method.

10.3 Methodical Recommendations

1. Mark the variables in the equation output systems by one name with different indices.
2. Check wehter the coincidence conditions are met with the given initial approximations.
3. To solve the non-linear equation systems by the Newton's method make up the sub-programmes to compute the Jacobian matrix, solve the linear and non-linear equation systems. Enter the initial approximations, address to the sub-programme of solution of the non-linear equation system, display the computing results in the main module.
4. Make up the sub-programme to compute $\varphi_1..\varphi_n$. If the system is solved by the Seidel method and that of simple iterations.

Table 10.1 – Tasks for the Laboratory Work №10

Number of the variant	Equation system	Method	Initial Approximations
1	2	3	4
1	$2x + tg\, xy = 0$ $(y^2 - 7{,}5)^2 - 15x = 0$	Simple iterations	$x_0=3$ $y_0=0$
2	$tg\, x - \cos 1{,}5y = 0$ $2y^3 - x^2 - 4x - 3 = 0$	Seidel	$x_0=0$ $y_0=1$
3	$10x^2 + 9y^2 - 1 = 0$ $\sin(3{,}2x + 0{,}3y) + 3x = 0$	Newton's	$x_0=0$ $y_0=0{,}5$
4	$\cos y + 2x = 0$ $0{,}24x + 3{,}5y + x^2y = 0$	Seidel	$x_0=0$ $y_0=0$
5	$\sin(x+0{,}4) + 3{,}5y - 1{,}5 = 0$ $\cos(y+0{,}2) + 0{,}5x = 0$	Simple Iterations	$x_0=-1{,}3$ $y_0=0{,}5$
6	$\sin(3{,}3x - 0{,}4y) + 4x = 0$ $8x^2 + 25y^2 - 1 = 0$	Newton's	$x_0=0$ $y_0=0{,}5$
7	$0{,}16x + 2{,}1y + x^2y = 0$ $\cos y + x = 0$	Seidel	$x_0=-1$ $y_0=0$
8	$2{,}1y^3 - x^2 - 4x - 3 = 0$ $tg\, 2x - \cos 2y = 0$	The same	$x_0=0$ $y_0=1$

Continuation of the table 10.1

1	2	3	4

9	$(y^2 - 7{,}5)-15x=0$ $tg\ xy+2x=0$	Simple iterations	$x_0=3$ $y_0=0$
11	$tg\ xy+6x=0$ $-120x+(y^2 - 20)^2=0$	The same	$x_0=3$ $y_0= - 0{,}5$
12	$0{,}9x+cos(y+1{,}6)=0$ $0{,}1-2y+sin(x+1{,}8)=0$	Simple iteations	$x_0=0{,}5$ $y_0=0{,}4$
13	$cos(y+0{,}6)+0{,}6x=0$ $sin(x+0{,}8)+2y-1=0$	The same	$x_0= - 0{,}8$ $y_0=0{,}5$
14	$tg\ 4x-cos\ 3y=0$ $2{,}3y^3-x^2-4x-3=0$	Seidel	$x_0=0$ $y_0=1$
15	$2{,}2y^3-x^2-4x-3=0$ $tg\ 3x-cos\ 2{,}5y=0$	The same	$x_0=0$ $y_0=1$
16	$5x+tg\ xy=0$ $(y^2-1{,}5)^2-7{,}5x=0$	Newton's	$x_0=0{,}6$ $y_0=-2$
17	$0{,}5y-0{,}5+sin(x+1{,}2)=0$ $0{,}7x+cos(y+0{,}8)=0$	The same	$x_0=-1$ $y_0=0$
18	$sin(x+2{,}1)-3y+0{,}4=0$ $cos(y+1{,}8)+1{,}2x=0$	Simple iterations	$x_0=0{,}4$ $y_0=0{,}5$
19	$4{,}9y+0{,}32x+x^2y=0$ $cosy+3x=0$	Thy same	$x_0=0$ $y_0=0$
20	$(y^2-5)^2-20x^{-}=0$ $tg\ xy+4x=0$	Newton's	$x_0=0{,}3$ $y_0=-2{,}8$
21	$sin(4x-0{,}5y)+5x=0$ $7x^2+30y^2-1=0$	The same	$x_0=0$ $y_0=0{,}5$
22	$tg\ 6x-cos\ 4y=0$ $2{,}5y^3-x^2-4x-3=0$	Simple iterations	$x_0=0$ $y_0=1$
23	$6x+tg\ xy=0$ $(y2-2)^2-12x=0$	Newton's	$x_0=0{,}5$ $y_0=-2$
24	$sin(3{,}1x+0{,}2y)+2x=0$ $12x^2+5y^2-1=0$	The same	$x_0=0$ $y_0=0{,}5$

Continuation of the table 10.1

1	2	3	4

25	$Cosy+5x=0$ $0{,}48x+6{,}7y+x^2y=0$	Seidel	$x_0=0$ $y_0=0$
26	$tg\ 5x-cos\ 3{,}5y=0$ $2{,}4y^3-x^2-3-4x=0$	The same	$x_0=0$ $y_0=1$
27	$14x^2+3y^2-1=0$ $sin(3x+0{,}1y)+x=0$	Newton's	$x_0=0$ $y_0=0{,}5$
28	$0{,}6x+7{,}5y+x^2y=0$ $cosy+6x=0$	Simple Iterations	$x_0=0$ $y_0=0$
29	$sin(x+1{,}6)-1=0$ $cos(y+1{,}2)+0{,}8x=0$	The same	$x_0=0{,}5$ $y_0=0{,}8$
30	$4x^2+35y^2-1=0$ $sin(4{,}2x-0{,}6y)+6x=0$	Newton's	$x_0=0$ $y_0=0{,}5$

Laboratory work 11

NUMERICAL SOULTION OF LINEAR DIFFERENTIAL EQUATIONS

Purpose of work: to learn to solve ordinary linera differentiaal equations with the initial conditions and their systems by the numerical methods.

11.1 Theoretical Data

Solution of differential equations makes up the basis of mathematical simulation of different devices, processes, systems.

Solution of differential equations is applied in electrical engineering and disciplines derived of it when the transiant processes are computed.

The ordinary differential equation of the n-type is of the form:

$$F(x,y,y',y'',\dots,y^{(n)})=0, \qquad (11.1)$$

Where x - independent variable;

$y(x)$ – unknown function of theindependent variable,

$y'(x)=\dfrac{dy}{dx},\ y''(x)=\dfrac{d_2y}{dx},\ y^{(n)}(x)=\dfrac{d_ny}{dx^n}$ - its derivatives .

To determine the private separate solution (11.1) *n* of initial conditions must be known:

$$y(x_0) = y_0, y'(x_0) = y'_0, \ldots, y^{(n-1)}(x_0) = y^{(n-1)}{}_0, \qquad (11.2)$$

The numerical solution of a differential equation lies in determination of the table of the values $y_i(x_i)(i=0,1,2,\ldots,k)$ on some interval $[x_0, x_k]$.

The difference between two adjacent table values of the argument is called the intergration step.

$$h = x_{i+1} - x_i . \qquad (11.3)$$

The methods Runge-Cutta is one of the most widely-spread method of solving of differential equations.

The Runge-Cutta methods are co-ordinated with the expansion of the function $y(x)$ into the Tailor series in the circumference of the point x_i up to the members which have h^p :

$$y_{i+1} \approx y_i + hy'_i + \frac{h^2}{2!} y''_i + \ldots + \frac{h^p}{p!} y^{(p)}. \qquad (11.4)$$

The index of the order p with h in the last member which was summed up means the method's order in the Tailor series.

The first order method of Runge-Cutta is called Euler method, that of the second order – modified Euler method or Euler –Koshee. The methods of higher orders don't have special names.

To apply the Runge-Cutta methods the output differential equation (11.1) is transformed into n system of the first order differential equation system in the normal Koshee form:

$$\begin{cases} y'_1 = f_1(x, y_1, y_2, \ldots, y_n), \\ y'_2 = f_2(x, y_1, y_2, \ldots, y_n), \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ y'_n = f_n(x, y_1, y_2, \ldots, y_n), \end{cases} \qquad (11.5)$$

$$y_1(x_0) = y_{10}, y_2(x_0) = y_{20}, \ldots, y_n(x_0) = y_{n0}.. \qquad (11.6)$$

The additional varibles y_1, y_2,…, y_n and their initial conditions are by all means related to the unknown function y and its derivatives.

According to the Euler method, one step of the differential equation system (11.5) with the initial conditions (11.6) solution is done according to the formula:

$$y_i(x+h) = y_i(x) + hf_i(x, y_1, y_2, \ldots, y_n), \qquad (11.7)$$
$$i = 1, 2, \ldots, n$$

The method of Euler- Koshee requires computing of the vector of derivatives (the right partsof the differntial equations) $\vec{F}(x, \vec{y})$ in two points:

$$\begin{cases} K1i=fi\,(x,y1,y2,\ \dots,\ y_n), \\ K2i=fi\,(x+h,y1+hK11,y2+h12,\ \dots,\ y_n+hK_{1n}); \end{cases} \quad (11.8)$$

$$y_i(x+h)=y_i(x)+\frac{h}{2}(K_{1i}+K_{2i}). \quad (11.9)$$

With the fourth-order Runge-Cutta method the vector of derivatives at every step of numerical intergration is computed four times:

$$\begin{cases} \vec{K}_1=\vec{F}(x,\vec{y}(x)), \\ \vec{K}_1=\vec{F}(x+\frac{h}{2},\vec{y}(x)+\frac{h}{2}\vec{K}_1), \\ \vec{K}_3=\vec{F}(x+\frac{h}{2},\vec{y}(x)+\frac{h}{2}\vec{K}_2), \\ \vec{K}_4=\vec{F}(x+h,\vec{y}(x)+h\vec{K}_3); \end{cases} \quad (11.10\ )$$

$$\vec{y}(x+h)=\vec{y}(x)+\frac{h}{6}(\vec{K}_1+2\vec{K}_2+2\vec{K}_3+\vec{K}_4). \quad (11.11)$$

Computing according to the above-mentioned formulae goes on until the end of the interval [x_0, x_k] is reached.

The error of the Runge-Cutta method is identified by the expression:

$$\varepsilon \approx K\cdot h^p. \quad (11.12)$$

The range of the coefficient K depends on the system under solution.

11.2 Task

Solve the system of differential equations with the initial conditions taken from the table 11.1 in the given interval with the given step by the method of Euler-Koshee (od variants). Compare the results.

11.3 Methodical Recommendations

1. Mark the dependent variables of the output equation system by one name with different indices (for example $y=y_1$, $z=y_2$).

2.Make separate sub-programmes to compute the vector of the derivatives $\vec{F}$ with the given values x *and* $\vec{y}$ and one step of the numerical intergration of the differential equation system by the given method.

3. Enter the output data (x_0, x_k, h, n,*initial conditions*) and iteration cycle with the independent variable x, in the main module; call the given method sub-programme in the middle of it and display the results (as a table or graph).

4. To control the programme first solve the second order differential equation test system for which the analytical solution is known. Compare the results of the numerical and analytical solutions.

Table 11.1 – Tasks for the laboratory work №11

№	Differential equations	Parametres	Interval	Step	Initial conditions
1	2	3	4	5	6
1,2	$\begin{cases} x' = \cos(x + ay) + b \\ y' = \dfrac{a}{t + bx^2} + t + 1 \end{cases}$	$a=2{,}5$ $b=3{,}0$	$t_н=0$ $t_к=0{,}3$	$ht=0{,}02$	$x(0)=1$ $y(0)=0{,}05$
3,4	$\begin{cases} x' = \sin(ax)^2 + t + y \\ y' = t - x - by^2 + 1 \end{cases}$	$a=2{,}0$ $b=3{,}5$	$t_н=0$ $t_к=0{,}3$	$ht=0{,}02$	$x(0)=1$ $y(0)=0{,}5$
5,6	$\begin{cases} x' = \sqrt{t^2 + ax^2} + y \\ y' = \cos(by) + x \end{cases}$	$a=2{,}5$ $b=3{,}5$	$t_н=0$ $t_к=0{,}15$	$ht=0{,}01$	$x(0)=0{,}5$ $y(0)=1$
7,8	$\begin{cases} x' = e^{-(x^2+y^2)} + at \\ y' = bx^2 + y \end{cases}$	$a=2{,}0$ $b=4{,}5$	$t_н=0$ $t_к=0{,}28$	$h_t=0{,}02$	$x(0)=0{,}5$ $y(0)=1$
9, 10	$\begin{cases} x' = \ln(bt + \sqrt{a^2t^2 + y^2}) \\ y' = \sqrt{a^2t^2 + x^2} \end{cases}$	$a=3{,}0$ $b=2{,}5$	$t_н=0$ $t_к=0{,}18$	$h_t=0{,}01$	$x(0)=1$ $y(0)=0{,}5$
11,12	$\begin{cases} y' = -yz - \dfrac{\sin x}{x} \\ z' = -z^2 - \dfrac{\alpha x}{1 + x^2} \end{cases}$	$\alpha = 2{,}5 + + \dfrac{\beta}{40}$ $\beta = 25$	$x_н=0$ $x_к=1$	$h_x=0{,}1$	$y(0)=0$ $z(0)=-0{,}4$
13,14	$\begin{cases} y' = z - (\alpha y - \beta z)y \\ z' = e^y - (y + \alpha z)y \end{cases}$	$\beta = 0{,}25n$ $\alpha = \beta + 2$ $n=4$	$x_н=0$ $x_к=1{,}2$	$h_x=0{,}1$	$y(0)=1$ $z(0)=0$

Continuation of the table 11.1

1	2	3	4	5	6
15,16	$\begin{cases} y' = \cos(y + az) + c \\ z' = \dfrac{a}{x + cy^2} + 1 + x \end{cases}$	a=2,0 $c=4{,}5$	$x_н=0$ $x_к=0{,}3$	$h_x=0{,}02$	$y(0)=1$ $z(0)=0{,}05$
17,18	$\begin{cases} y' = z + \sin(\alpha y^2) + x \\ z' = x + y - \beta z^2 + 1 \end{cases}$	$\alpha = 2{,}5$ $\beta = 3{,}0$	$x_н=0$ $x_к=0{,}3$	$h_x=0{,}02$	$y(0)=1$ $z(0)=0{,}5$
19,20	$\begin{cases} y' = z + \sqrt{x^2 + ky^2} \\ z' = y + \cos(nz) \end{cases}$	$k=2$ $n=4$	$x_н=0$ $x_к=0{,}28$	$h_x=0{,}02$	$y(0)=0{,}5$ $z(0)=1$

21,22	$\begin{cases} y' = cx + e^{-(y^2+z^2)} \\ z' = dy^2 + z \end{cases}$	*c=2* *d=4,5*	$x_н=0$ $x_к=0,18$	$h_x=0,01$	*y(0)=1* *z(0)=0,5*
23,24	$\begin{cases} y' = ln(\sqrt{k^2x^2+z^2}+cx) \\ z' = \sqrt{(kx)^2+y^2} \end{cases}$	k=3 *c=2,5*	$x_н=0$ $x_к=0,3$	$h_x=0,02$	*Y(0)=1* *z(0)=0,5*

Laboratory Work 12

IINTERPOLATION

Purpose of work: to learn to determine the value of the functions given in the form of a table under any value of the arguments with the help of interpolation of functions by step polynomials.

12.1 Theoretical Data

Many functional dependences in science and engineering are given not in an analytical way, but in the form of tables and graphs.

Computers enter the information on these functions in the form of arrays. For example:

$$y_i = f(x_i), \tag{12.1}$$

$$i=0,1,\ldots,n\,. \tag{12.2}$$

Interpolation consists in finding approximate value of the non-linear function *y* in the points which differ from the nodes ($x \neq x_i$).This task can be done if the function *F(x)*, which iterpolates and gets the values at some interval[x_j, x_{j+k}] is found. These values coincide with the values of the table function(12.1) in the node points:

$$F(x_j)=y_j,\ldots,F(x_{j+1})=y_{j+1},\ldots,F(x_{j+k})=y_{j+k}.\ . \tag{12.3}$$

The point x_j is called the interpolation initial node.

The algebraic polynomial

$$P_k(x)=a_0x^k+a_1x^{k-1}+\ldots+a_k\,, \tag{12.4}$$

$$k \leq n\,. \tag{12.5}$$

Is often used as the interpolating function.

If $k=n$ the polynomial (12.4) is a global interpolant as in this case its values coincide with the values of the output function in all nodes ($j=0, j+k=n$).

If the table function is given in equispaced nodes, that is:

$$x_{i+1}–x_i=h=const, \tag{12.6}$$

the value $y(x)$ can be defined acording to the first interpolation Newton's formula:

$$\begin{aligned} y(x) \approx P_k(x) = y_j + q\Delta y_j + \frac{q(q-1)}{2!}\Delta^2 y_j + \\ \ldots + \frac{q(q-1)\ldots(q-k+1)}{k!}\Delta^k y_j, \end{aligned} \tag{12.7}$$

Where $q = \dfrac{x - x_j}{h}$;

(12.8)

$\Delta y_j, \Delta^2 y_j, \ldots, \Delta^k y_j$ - forward differences of the corresponding orders at the initial node.

If the table function nodes are placed irregularly $(x_{i+1}$-$x_i=var)$, the values $y(x)$ can be determinde by theinterpolation Lagrange formula:

$$y(x) \approx L_k(x) =$$

$$= \sum_{m=j}^{j+k} y_m \frac{(x-x_j)(x-x_{j+1})\ldots(x-x_{m-1})(x-x_{m+1})\ldots(x-x_{j+k})}{(x_m-x_j)(x_m-x_{j+1})\ldots(x_m-x_{m-1})(x_m-x_{m+1})\ldots(x_m-x_{j+k})}. \tag{12.9}$$

The formulae (12.8), (12.9) can be used to find $y(x)$ on theinterval $[x_j, x_{j+k}]$, but the highest accuracy is observed near the initial node of interpolation x_j :

$$x \in \lfloor x_j, x_{j+1} \rfloor.$$

So, befor we use the interpolation formulae the number of theinterpolation initial node should be determined. The choice condition can be formulated as follows:

$$j = \begin{cases} 0 \ \ with\ x < x_0, \\ n\text{-}k \ npu\ x > x_{n-k}, \\ \text{and with } \ x_i \le x < x_{i+1},\ i=1,2,\ldots,n\text{-}k\ . \end{cases}$$

The linear or quadratic interpolation is normally used in technical calculations. In this case, the formulae (12.7) and (12.9) are of the form

When $k = 1$:

$$y(x) \approx P_1(x) = y_j + q(y_{j+1} - y_j), \qquad (12.10)$$

$$y(x) \approx L_1(x) = y_j \frac{(x - \mathrm{x}_{j+1})}{(x_j - \mathrm{x}_{j+1})} + y_{j+1} \frac{(x - \mathrm{x}_j)}{(x_{j+1} - \mathrm{x}_j)}; \qquad (12.11)$$

when $k = 2$:

$$y(x) \approx P_2(x) = y_j + q(y_{j+1} - y_j) + \frac{q(q-1)}{2}(y_{j+2} - 2y_{j+1} + y_j), \qquad (12.12)$$

$$y(x) \approx L_2(x) = y_j \frac{(x - x_{j+1})(x - x_{j+2})}{(x_j - x_{j+1})(x_j - x_{j+2})} + y_{j+1} \frac{(x - x_j)(x - x_{j+2})}{(x_{j+1} - x_j)(x_{j+1} - x_{j+2})} +$$

$$+ y_{j+2} \frac{(x - x_j)(x - x_{j+1})}{(x_{j+2} - x_j)(x_{j+2} - x_{j+1})}. \qquad (12.13)$$

The formulae (12.10) and (12.11) are the equations of the line which passes through the points (x_j, y_j) andi (x_{j+1}, y_{j+1}), and (12.12) i (12.13) – the equations of the quadratic parabola which passes through the points (x_j, y_j), (x_{j+1}, y_{j+1}), (x_{j+2}, y_{j+2}).

12.2 Task

Compute the approximate values of the table functions given in the table 12.1 for the arguments which are changed according to the following laws:

Odd variants:

$$x=10\,\sin t,\ \ t=0.5,\ 0.6,\ldots,1.5;$$

even variants :

$$x=10\,\cos t,\ \ t=0.1,\ 0.2,\ldots,1.$$

The quadratic interpolation of Newton or the linear interpolation of Lagrange are used depending on the node placement. Varify the solution with the help of graphs.

12.3 Methodical Recommendations

1. Mark arrays of the argument table values $\vec{X} = (x_0, x_1, \ldots, x_n)$ function $\vec{Y} = (y_0, y_1, \ldots, y_n)$ and value x in the programme by diferent identifiers, for example,

$$\vec{X} \to XT, \quad \vec{Y} \to YT, \quad x \to X, \quad y \to Y.$$

2. Verify the condition $x=x_j$ after searching the interpolation initial node number. To meet the condition don't apply the interpolation formula, determine the values from the table : $y=y_j$.

Table 12.1- Task for the Laboratory Work №12

Number of the variant	Table Functions									
1	2									
1,2	x_i	-1	1	3	5	7	9	11	13	15
	z_i	8,71	109,8	124,4	122,5	112,1	96,6	80,2	6,3	57,9
3,4	x_i	2	3,2	4,4	6,2	7,8	9,5	10,9	11,5	12,7
	w_i	19,9	22	30	42,1	65	99,5	120	126,8	133,4
5,6	x_i	-3,5	-1,5	0,5	2,5	4,5	6,5	8,5	10,5	12,5
	h_i	0,45	-3,09	-4,01	-3,9	-3	-1,62	-0,18	0,99	1,72
7,8	x_i	1,25	2,59	4,4	6,54	8,5	11,5	13,5	14,9	15
	P_i	3,0	5,0	7,0	8,5	9,3	9,9	10,6	11,2	11,64
9,10	x_i	-2	0	2	4	6	8	10	12	14
	f_i	7,84	7,13	6,31	5,29	4,03	2,5	0,87	-0,68	-0,79
11,12	x_i	-1,5	1	2,7	5,5	6,5	8,3	9,6	11,2	12,75
	U_i	2,45	1,12	-1	-2,1	-2,3	- 1,9	-1	2	3,5
13,14	x_i	0,67	1,5	2,5	3,5	5	6,5	10	12,4	14
	S_i	110	118,7	124,5	125,2	122,5	115,1	88,3	70	61,2
15,16	x_i	0,5	2,5	4,5	6,5	8,5	10,5	12,5	14,5	16,5
	p_i	23,7	20,1	27,8	45,3	79,2	115,4	132,9	141,1	147
17,18	x_i	-2,77	-0,5	1	2	3,5	7	10	11,5	12,5
	z_i	-1,5	-3,65	-4,03	-4,0	-3,54	-1,58	0,73	1,4	1,83
19,20	x_i	0,5	2,0	3,5	5,0	6,5	8,5	9,5	11,0	12,5
	W_i	1,23	0,92	0,78	0,68	0,6	0,53	0,49	0,47	0,45
21,22	x_i	-1	1	3	5	7	9	11	13	15
	h_i	1,02	2,57	5,51	7,52	8,69	9,38	9,79	10,35	1,64

Continuation of the table 12.1

1	2									
23,24	x_i	-3	0,5	1,5	2,5	4,3	6,2	7,7	9,0	11
	f_i	9,4	7,52	6,75	5,8	3,6	0,53	-1,5	-2,94	-4,4
25,26	x_i	-4	-2	0	2	4	6	8	10	12
	U_i	3,1	2,66	1,74	0,35	-1,26	-2,28	-2,07	-0,54	2,53
27,28	x_i	0	0,4	1,5	3,0	4,6	7	9,2	11,5	13
	S_i	1,47	1,26	0,99	0,82	0,7	0,57	0,5	0,46	0,44

3. To do graphic varification display as a graph the table function and its iterpolated values in different forms or different colours. For example, display the

function in the form of the “grid” (sections with the end coordinates $(x_i,0)$, (x_i,y_i), $i=0,1,..,n$), and the interpolated values in the form of the points (x,y) or the function in the form of the broken curve which consists of the sections with the coordinates (x_{i-1},y_{i-1}) (x_i,y_i), and the interpolated values in the form of the grid.

Laboratory Work 13

APPROXIMATION DONE BY THE LEAST-SQUARES METHOD

Purpose of work**:** to learn to describe table functions by analytical expressions.

13.1 Theoretical Data

Approximation (Lain : *approximare* – to approach) – approximate expression of some values through other, simpler ones.

Table function approximation

$$y_i = f(x_i) , \tag{13.1}$$

$$i=1,2,...,n . \tag{13.2}$$

made by the least-squares method consists in determining some analytical *F(x)*,function parametres that provide the functional minimization.

$$\Phi = \sum_{i=1}^{n}(F(x_i) - y_i)^2 . \tag{13.3}$$

If the stepped polynomial

$$F(x) = P_k(x) = C_0 + C_1 x + ... + C_k x^k = \sum_{j=0}^{k} C_j x^j , \tag{13.4}$$

is chosen to be the approximating function, the task consists in determining the vector of the coefficients $\vec{C} = (C_0 ,C_1 ,...,C_k)$ by solving the system oflinear equations of $(k+1)$ order.

$$\begin{cases} \dfrac{\partial \Phi}{\partial C_0} = 0, \\ \dfrac{\partial \Phi}{\partial C_1} = 0, \\ \dfrac{\partial \Phi}{\partial C_k} = 0, \end{cases} \tag{13.5}$$

Where $$\Phi = \sum_{i=1}^{n}(\sum_{j=1}^{k} C_j x_i^j - y_i)^2 ,$$

(13.6)

$$\frac{\partial \Phi}{\partial C_m} = 2\sum_{i=1}^{n}(\sum_{j=1}^{k} C_j x_i^j - y_i) x_i^m ,$$

(13.7)

$$m=0,1,\ldots,k.. \qquad (13.8)$$

After the transformations the system (13.5) is of the form :

$$\begin{cases} C_0 \cdot n + C_1 \sum_{i=1}^{n} x_i + \ldots + C_k \sum_{i=1}^{n} x_i^k = \sum_{i=1}^{n} y_i \ , \\ C_0 \cdot \sum_{i=1}^{n} x_i + C_1 \sum_{i=1}^{n} x^2{}_i + \ldots + C_k \sum_{i=1}^{n} x_i^{k+1} = \sum_{i=1}^{n} x_i y_i \ , \\ C_0 \cdot \sum_{i=1}^{n} x_i^k + C_1 \sum_{i=1}^{n} x_i^{k+1} + \ldots + C_k \sum_{i=1}^{n} x_i^{2k} = \sum_{i=1}^{n} x_i^k y_i \ . \end{cases} \qquad (13.9)$$

The system (13.9) shows that the the elements of the matrix of the coefficients A and the vectors of the free terms $\vec{B}$ can be described by the formulae:

$$\begin{cases} a_{mj} = \sum_{i=1}^{n} x_i^{m+j} , \\ b_m = \sum_{i=1}^{n} x_i^m y_i , \\ m = 0,1,\ldots,k \quad , \\ j = 0,1,\ldots,k \end{cases} \qquad (13.10)$$

When the coefficients are determined, the system (13.9) can be solved by any of the known methods, for example Gaussian.

Approximation done by the least-squares method is often used to smooth the table functions obtained as a result of the experiment as well as to reduce the amount of information on the table functions when the requirements on computation accuracy are not strict..

13.2 Task

Approximate the table function given in the table of the work 12 by the step polynomial of the *k* order by the leat-squares method. For odd variants *k*=3, for even ones - *k*=2. Illustrate the results by graphs. Do the programme twice with different number of the table points (*n=9* i *n=5*). Evaluate theinfluence of the number oa points on the approximation accuracy.

13.3 Methodical Recommendations

When standard programmes are applied to solve the equation system (13.9) the indices of the standard equation system (which don't have coefficients with zero indices) are coordinated with those of the system (13.9). The coordination consists in change of the formulae (13.10).

Laboratory Work 14

NUMERICAL INTEGRATION

Purpose of work: to learn to compute integrals from the functions given in the table and analytical ways.

14.1 Theoretical Data

Tasks to compute integrals appear practically in all spheres of applied mathematics.

The basics of the numerivcal integration methods consists in the fact that the interval [*a, b*] is split into sections on which the curve described by the sub-integral fucntion *f(x),* is replaced by some other curve for which theintegral computing is done by rather simple formulae, and then all squares are summed.

When interpolating polymomials replace the sub-integral function the so called quadrature formula is obtained. The quadrature formulae for th einterpolation equidistance nodes are called the formulae of Newton- Cotes.

Depending on the degree of the interpolating polinomial we distinguish the methods of rectangles, trapezium and quadratic trapezium, or Symson method.

The main formulae and indices which charaterise these methods when the integration interval is divided into equal sections are given in the table 14.1, where the following symbols are accepted:

n – the number of the layout sections,

$$h = x_i - x_{i-1} = \frac{b-a}{n} = const \text{ - integration step}$$

$$y_i = f(x_i), i = 0,1,2,..n, \qquad (14.1)$$

$$x_0 = a, x_n = b, y_0 = f(a), y_n = f(b).$$

The method mistake is detrmined by the value of the integral of the interpolation polynomial remainder term. In the formulae to evaluate the mistake M_i the maximum value of the *i* derivative $\frac{d^i f(x)}{dx^i}$ on the interval [*a, b*].

Table 14.1 Task for the Laboratory Work №14

Name of the method	Interpolating polynomial degree	$z \approx \int_a^b f(x)dx$	Mistake
Rectangle	0	$h \cdot (y_0 + y_1 + \ldots + y_{n-1})$	$\frac{M_1 h}{2}(b-a)$
Trapezium	1	$h \cdot (\frac{y_0 + y_n}{2} + y_1 + y_2 + \ldots + y_{n-1})$	$\frac{M_2 h^2}{12}(b-a)$
Simpson	2	$h \cdot (y_0 + y_n + 4(y_1 + y_3 + \ldots + y_{n-1}) +$ $+ 2(y_2 + y_4 + .. + y_{n-2}))$	$\frac{M_4 h^4}{180}(b-a)$

When Simson method is applied the number of the layout sections is even (*n=2k*) and all the sections are identical. Under the irregular integration interval layout the rectangle and trapezium methods are applied. For them the numerical integration formulae are of the form:

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} y_i (x_{i+1} - x_i) - \tag{14.2}$$

According to the rectangle method,

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \frac{(y_{i+1} + y_i)}{2}(x_{i+1} - x_i) - \tag{14.3}$$

According to the trapezium method

To provide the required integration accurcay the algorithms with the step automated choiceare used. The integral value computing is done by one of the examined methods with the initial step h, and then these computations are repaetaed with the half-step $\frac{h}{2}$. If it appears that

$$\left| z(h) - z(\frac{h}{2}) \right| \le \varepsilon, \tag{14.4}$$

Where ε - the integration permissible mistake

The computing is stopped, if not the further step split is done.

The integral approximate value obtained by this method can be specified by the extrapolation transition to the limit proposed by Richardson:

$$z \approx z(\frac{h}{2}) + \frac{z(\frac{h}{2}) - z(h)}{2^k - 1}, \tag{14.5}$$

Where $k = \begin{cases} 1 \text{ – for the rectanle method ,} \\ 2 \text{ – for the trapezium method,} \\ 3 \text{ –or the Sipson method .} \end{cases}$

14.2 Task

14.2.1 Table Function Integration

Compute the integral from the table function given in the table 14.2. Odd variants use the trapezium method under the interval irregular layout of integration and the method of Simpson under the regular one. The even varinats use the rectangle method and that of trapezium in analogous cases.

14.2.2 Integration of the Functions Given Analitically

Compute theintended integral

$$z = \int_a^b f(x)dx$$

For the function given in the table 14.2 on the intended interval [*a*,*b*] with the intended accuracy ε ,using the step automated choice method specified in the table

14.3 Methodical Recommendations

To have the result visual verification build the sub-integration function graph and place it on the line parallel to the abscissa axis on the level of

$$y_{cp} = \frac{\int_a^b f(x)dx}{b-a}.$$

If the solution is correct the square limited by the sub-integral curve and lines $x=a$, $x=b$ i $y=0$, equals to the square of the rectangle limited by the line sections $y=y_{cp}$, $x=a$, $x=b$, $y=0$.

Tabel 14.2 Output Data for the Laboratory Work №14

Variants	Method	ε	*f(x)*	*a*	*b*	Parametres
1	2	3	4	5	6	7
1	Rectangle	10^{-5}	$\frac{x}{1+\sin x \cdot \cos\varphi}$	0		π φ =0,182
2	Sipmson	10^{-3}				
3	Trapezium	10^{-4}	$\frac{\cos^2 x}{c^2\sin^2 x + d^2\cos^2 x}$	0	$\frac{\pi}{2}$	*c=0,953*

4	Rectangle	10^{-2}				$d=2,295$
5	Trapezium	10^{-6}	$\frac{1}{c^2+d^2x}$	0	1	$c=3,18$ $d=-1,37$
6	Simpson	10^{-3}				
7	Rectangle	10^{-2}	$cos^m x \cdot cos\, mx$	0	$\frac{\pi}{2}$	$m=3$
8	Trapezium	10^{-4}				
9	The same	10^{-5}	$\frac{x^5}{c^3-x^3}$	0	1	$c=1,21$
10	Simpson	10^{-3}				
11	Trapezium	10^{-2}	$\frac{1}{c\cdot\sin x+d\cdot\cos x}$	0	$\frac{\pi}{2}$	$c=8,53$ $d=0,524$
12	Rectangle	10^{-4}				
13	Simpson	10^{-5}	$\frac{x^2}{c^3+x^3}$	0	1	$c=0,732$
14	Trapezium	10^{-3}				

Continuation of the table 14.2

1	2	3	4	5	6	7
15	Rectangle	10^{-4}	$\frac{1}{c^2-2\cdot c\cdot d\cdot cos\, x+d^2}$	0	π	$c=3,76$ $d=8,39$
16	Trapezium	10^{-2}				
17	Simpson	10^{-6}	$\frac{x\cdot e^{cx}}{(1+cx)^2}$	0	1	$c=4,18$
18	Trapezium	10^{-3}				
19	The same	10^{-2}	$\frac{x}{c^2+x^2}$	0	1	$c=0,874$
20	Rectangle	10^{-4}				
21	Simpson	10^{-5}	$\frac{ln(x+r)}{x^3}$	1	2	$r=\sqrt{x^2+c^2}$ $c=2$
22	Trapezium	10^{-3}				
23	Rectangle	10^{-2}	$sin\, ln\, x$	1	5	
24	Simpson	10^{-4}				
25	The same	10^{-5}	$\frac{1}{1-c\, sin\, x}$	0	$\frac{\pi}{2}$	$c=0,5$
26	Trapezium	10^{-3}				

Laboratory Work 15

HARMONIC ANALYSIS AND SYNTHESIS OF THE PERIODICAL FUNCTIONS

Purpose of work: to learn to determine the periodical function harmonic composition.

15.1 Theoretical Data

The time function *f(t)* is called periodical if for it the condition

$$f(t) = f(t + m \cdot T), \quad m = 1,2,3\ldots, \tag{15.1}$$

Where T - period
Is equitable.

The periodic function harmonic analysis consists in determination of the coefficients a_k , b_k of the Fourier series:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k \cos wkt + b_k \sin wkt), \tag{15.2}$$

Where $w = \frac{2\pi}{T}$ – the first harmonic circular frequency;

k – the harmonic serial number

Being limited by some final number of harmonics m in the formula (15.2) the approximating harmonic polynomial $Q_m(t)$ is obtained *:*

$$f(t) \approx \mathrm{Qm}(t) = \frac{a_0}{2} + \sum_{k=1}^{m}(a_k \cos wkt + b_k \sin wkt). \tag{15.4}$$

The Fourier coefficients are determined by the expressions

$$a_k = \frac{2}{T}\int_0^T f(t) \cdot \cos wkt, \tag{15.5}$$

$$b_k = \frac{2}{T}\int_0^T f(t) \cdot \cos wkt.$$

Using the rectangle method for the numerical integration of the formulae (15.5) when the integration interval [$0,T$] is split into n equal sections we get::

$$\begin{cases} a_0 \approx \frac{2}{n}\sum_{i=0}^{n-1} y_i, \\ a_k \approx \frac{2}{n}\sum_{i=0}^{n-1} y_i \cos\frac{2\pi\, ki}{n}, \\ b_k \approx \frac{2}{n}\sum_{i=0}^{n-1} y_i \sin\frac{2\pi\, ki}{n}, \end{cases} \tag{15.6}$$

$$k=1,2,\ldots,m,$$

$$y_i = f(t_i), \tag{15.7}$$

$$t_i = i\Delta t, \tag{15.8}$$

$$\Delta t = \frac{T}{n}. \tag{15.9}$$

Under

$$n=2k \tag{15.10}$$

the function $Q_m(t)$ becomes the trigonometric interpolator.

Getting of the periodical function by summing up its harmonic components according to the formula (15.4).is called the harmonic synthesis

15.2 Task

Compute the coefficients of the interpolating trigonometric polynomial that approximate the table function given in the points

$$t_i = \frac{2\pi}{n} i \ , \quad i = 0,1,2,\ldots,n$$

When $n = 20$.

Build the interpolating function graph and plot the output tabel function in the form of the grid.

15.3 Methodical Recommendations

For vizualisation the integrity of the points on the approximation function graph is to be 5-10 times as much of the integrity of the points which split the period into the sections for numerical integration.

Observe the influence of the number of the harmonics m under the given number of the layout sections on the approximation accuracy..

Таблиця 15.1 – Вихідні дані до лабораторної роботи №15

Variant number	Function table values
1	2
1	1.00,1.803, 3.085,4.776,6.434,7.347,7.027,5.652,3.897,2.381, 1.347, 7.422, 0.419, 0.256, 0.176, 0.142, 0.136, 0.155, 0.209, 0.324, 0.554
2	7.38, 6.76, 5.22, 3.47, 2.07, 1.16, 0.64, 0.36, 0.23, 0.16, 0.13, 0.13, 0.16, 0.23, 0.37, 0.64, 1.16, 2.08, 3.48, 5.22, 6.76
3	-1.24, -1.17, -1.08, -0.96, -0.84, -0.79, -0.8, -0.9, -1.1, -1.21, -1.02, -1.28, -1.32, -1.34, -1.36, -1.37, -1.37, -1.36, -1.35, -1.33, -1.30

4	-3.0, -3.58, -4.12, -4.56, -4.86, -4.99, -4.94, -4.73, -4.36, -3.86, -3.30, -2.7, -2.13, -1.64, -1.26, -1.05, -1.00, -1.13, -1.43, -1.87, -2.43
5	1.0,1.05, 90.6, 520.4, 1714.7, 2915.0, 2439.2, 1020.6,230.7, 32.17, 3.29, 0.3, 0.03, 0.004, 0.001, 0.0003,0.0006, 0.002, 0.01, 0.09, 0.9
6	2980.1, 2089.3, 742.4, 146.6, 18.6, 1.8, 0.16, 0.02, 0.003, 0.001, 0.001,0.001,0.002,0.003, 0.018, 0.9, 1.22, 18.6, 146.6, 742.5, 2089.7
7	1.0, 1.34, 1.75, 2.18, 2.53, 2.71, 2.65, 2.37, 1.97, 1.54, 1.16, 0.86, 0.64, 0.5, 0.42, 0.37, 0.36, 0.39, 0.45, 0.56, 0.74
8	2.71, 2.6, 2.28, 1.86, 1.44, 1.07, 0.8, 0.46, 0.42, 0.4, 0.37, 0.37, 0.4, 0.48, 0.6, 0.8, 1.07, 1.44, 1.86, 2.28, 2.6
9	-1.32,-1.28,-1.26,-1.24, -1.25, -1.25, -1.25, -1.26, -1.27, -1.29, -1.29, -1.33, -1.34, -1.37, -1.37, -1.37, -1.37, -1.36, -1.36, -1.35, -1.34

Continuation of the table 15.1

1	2
10	-4.0, -4.2, -4.5, -4.7, -4.9, -5.0, -4.9, -4.8, -4.6, -4.4, -4.1, -3.8, -3.5, -3.1, -3.0, -3.0, -3.0, -3.1, -3.2, -3.4, -3.7
11	1.0, 2.4, 5.4, 10.4, 16.3, 19.9, 18.6, 13.4, 7.7, 3.6, 1.6, 0.64, 0.27, 0.13, 0.07, 0.05, 0.05, 0.06, 0.09, 0.18, 0.4
12	20.0, 17.5, 11.9, 6.4, 2.9, 1.2, 0.5, 0.2, 0.1, 0.06, 0.05, 0.05, 0.06, 0.1, 0.5, 1.0, 1.2, 2.9, 6.4, 11.9, 17.5
13	-1.1, -0.8, -0.3, 0.3, 0.7, 0.8, 0.7, 0.5, 0.04, -0.6, -0.9, 1.1, -1.27, -1.32, -1.35,-1.37, -1.37, -1.36, -1.34, -1.3, -1.2
14	-2.0, -2.8, -3.7, -4.3, -4.7, -4.9, -4.9, -4.5, -4.1, -3.3, -2.4, -1.5, -0.6, -0.04, 0.6, 0.02, 0.99, 0.79, 0.34, 0.3, -1.1
15	1.1, 3.2, 9.5, 22.8, 41.4, 53.9, 49.4, 31.9, 15.2, 5.7, 1.8, 0.55, 0.17, 0.06, 0.03, 0.02, 0.01, 0.02, 0.04, 0.1, 0.3
16	-0.78, -1.22, -1.34, -1.39, -1.42, -1.43, -1.42, -1.41, -1.37, -1.3, -1.1, -0.1, 1.1, 1.2, 1.33, 1.36, 1.37, 1.35, 1.3, 1.17, 0.65
17	54.5, 45.7, 27.2, 12.1, 4.3, 1.3, 0.4, 0.13, 0.05, 0.03, 0.02, 0.02, 0.03, 0.05, 0.13, 0.41, 1.3, 4.3, 12.1, 21.2, 45.7
18	-0.78, 0.18, 0.89, 1.13, 1.21, 1.24, 1.23, 1.18, 1.04,0.63, -0.38, -1.01, -1.22, -1.3, -1.35, -1.36, -1.37, -1.36, -1.33, -1.27, -1.1
19	-1.0, -2.1, 3.2, -4.1, -4.7, -4.9, -4.8, -4.4, -3.7, -2.7, -1.6, -0.4, 0.7, 1.7, 2.4, 2.9, 3.0, 2.7, 2.1, 1.2, 0.2
20	1.0 , 4.36, 16.7, 49.8, 105. 0, 146. 3, 130. 9, 75.9, 30.0,8.75, 2.1, 0.47, 0.11, 0.03, 0.01, 0.007, 0.006, 0.009, 0.02, 0.05, 0.2
21	148.4, 118.8, 62.6, 25.5, 6.21, 1.45, 0.33, 0.08, 0.02,0.01, 0.007,0.007, 0.01, 0.02, 0.08, 0.32, 1.45, 6.2, 22.6, 62.2, 119.0
22	0.0, 0.97, 1.23, 1.32, 1.36, 1.37, 1.36, 1.34, 1.28, 1.130.64, -0.64, -1.13, -1.28, -1.34, -1.37, -1.36, -1.32, -1.23, -0.9, -0.2

23	-0.0001, -1.47, -2.8, -3.9, -4.65, -4.98, -4.87, -4.33, -3.4, -2.16, -0.74, 0.74, 2.17, 3.14, 4.33, 4.87, 4.98, 4.65, 3.9, 2.8, 1.4
24	1.0,5.8, 29.3, 108.9, 266.4, 396.7, 347.1, 180.5, 59.2, 13.5, 2.4, 0.4, 0.07, 0.01, 0.005, 0.003, 0.002, 0.004, 0.009, 0.03, 0.1
25	403.4, 309.0, 142.2, 42.1, 8.9, 1.56, 0.26, 0.05, 0.01, 0.0044, 0.0026, 0.0026, 0.0044,0.01, 0.05, 0.263, 1.56, 8.95, 42.1, 142.2, 309.9
26	0.78, 1.22, 1.34, 1.39, 1.42, 1.43, 1.42, 1.41, 1.37, 1.3, 1.1, 0.1, -1.1, -1.2, -1.33, -1.36, -1.37, -1.35, -1.3, 1.17, -0.65
27	1.0, -0.77, -2.3, -3.6, -4.6, -4.9, -4.8, -4.1, -3.1, -1.6, 0.1, 1.9, 3.6, 5.1, 6.2, 6.84, 6.98, 6.58, 5.69, 4.4 , 2.7
28	1.0 , 7.8, 51.5, 238.1, 675.9, 1075.4, 620.1, 429.3, 110.8, 20.8, 2.83, 0.35, 0.04, 0.01, 0.002, 0.001, 0.001, 0.001,0.004, 0.02, 0.12

Continuation of the table 15.1

1	2
29	1.10, 1.32, 1.40, 1.43, 1.45, 1.46, 1.46, 1.44, 1.42, 1.37, 1.25, 0.76, -0.8, -1.22, -1.33, -1.36, -1.37, -1.35, -1.29, -1.1, -0.1
30	2.0 , -0.06, -1.9, -3.4, -4.9, -4.8, 4.0, -2.7, -1.1, 0.95, 3.0, 5.0, 6.7, 8.1, 8.8, 8.9, 8.5, 7.47, 5.94, 4.06

Laboratory Work 16

SEARCH OF THE FUNCTION EXPTREME VALUES BY THE GOLDEN SECTION METHOD

Purpose of work: to learn to determine maximum and minimum values of the function on the given interval

16.1 Theoretical Information

The search of the extremum of function of one variable is not only of self-contained interest . It is an important element of the minimization of functions of several variables when different optimization tasks ary solved.

The below given method allows to find the point of the extremum of the function *f(x)* on the interval [*a,b]*. *The section* [*a,b*] should have one maximum or minimum of the function under investigation.

The division of the segment into two parts in such a way that the ratio of the length of all segment to the length of the biggest part equals to the ratio of the biggest part to the smallest one is called the golden section of the segment.

It’s not difficult to prove that the golden section of the segment [*a, b*] is fulfilled by two points placed symmtrically:

$$\begin{cases} x_1 = b - \tau(b-a), \\ x_2 = a + \tau(b-a), \end{cases} \qquad (16.1)$$

Where $\tau = \frac{\sqrt{5}-1}{2} \approx 0{,}6180339, \quad x_1 < x_2.$

(16.2)

Besides, the point x_1 in its turn makes the goldens ection of the segment [a, x_2], and the point x_2–of the segment [$x1$, b].

According to the above-mentioned the search of the minimum function value on the given interval [a, b] can be done as follows:

- divide the segment [a, b] by the points x_1 *and* i x_2 according to the golden section rule ;

-compute the minimized function , $f(x)$ value *in the points* x_1 i x_2;

- with $f(x_1)>f(x_2)$ change the left limit of the interval з $a=x_1$, if not –the right one $b=x_2$;

- repeat the process form the very beginning taking into account that one of the points of the golden section is already known;

- go on with the interations until the interval of uncertainty [a, b] is less than the given mistake ε;

- the minimum point is clarified by halving of the segment [a, b] when the iteration is finished:

$$x_{min} = \frac{a+b}{2}.$$

The function maximum is found in the analogous way.

16.2 Task

Find the minimum or maximum value of the function on the interval [a, b] with the accuracy ε.The output data is given in the table 16.1. Build the function grapgh and find the extremum point on it.

Table 16.1 – Task for the Laboratory Work №16

№	*f(x)*	*a*	*b*	ε	Extremum type
1	2	3	4	5	6
1	$e^{2\cos x} + 2\sin x$	4	5	10^{-5}	Minimum
2	$arctg(2{,}8\sin x - 1{,}3)$	0	2	10^{-4}	Maximum
3	$x^2 + 4e^{-0{,}25x}$	-1	2	10^{-3}	Minimum
4	$2{,}5\sin(x^2 - 0{,}5) - 0{,}67x$	1	2	10^{-5}	Maximum

5	$x^4+1{,}2arctg\,6x$	-2	0	10^{-4}	Minimum
6	x^2e^{-2x}	0	2	10^{-3}	Maximum
7	$arctg(2{,}8\sin x-1{,}3)$	-2	0	10^{-4}	Minimum
8	$\sin 3x\cdot e^{\sin 2x}$	0	1	10^{-5}	Maximum
9	$8e^{\frac{x}{2}}+x^2$	-2	0	10^{-3}	Minimum

Continuation of the table 16.1

1	2	3	4	5	6
10	$2\sin x+e^{2\cos x}$	-1	1	10^{-3}	Maximum
11	$-1{,}5arctg10x+x^4$	0	2	10^{-3}	Minimum
12	$e^{5\sin x}$	$\pi/4$	$3\pi/4$	12^{-6}	Maximum
13	$1{,}8^x+e^{\lvert x-0{,}8\rvert}$	1	3	10^{-4}	Minimum
14	$\sin(x-\cos x)$	1	2	10^{-5}	Maximum
15	$e^{\sin 2x}\cdot\sin 3x$	-1	0	10^{-4}	Minimum
16	$\ln 0{,}85x-0{,}96x+2$	0,1	2	10^{-3}	Maximum
17	$tg8x-27{,}3x$	0,1	0,18	10^{-6}	Minimum
18	$6{,}3\sin(5{,}2x)-2{,}8x$	-0,1	0,6	10^{-5}	Maximum
19	$2{,}5\sin(x^2-0{,}5)-0{,}67x$	-1	1	10^{-4}	Minimum
20	$3{,}4\cos(\frac{x}{2}-0{,}28)-0{,}1x^4$	-1	2	10^{-3}	Maximum
21	$\cos 2x\cdot\ln\frac{x}{2}-\frac{x}{4}$	1,6	3	10^{-4}	Minimum
22	$19{,}2x-2+tg(-3x)$	0,2	0,5	10^{-6}	Maximum
23	$0{,}2x^2+2{,}1\sin(0{,}2x-0{,}18)$	-5	0	10^{-3}	Minimum
24	$\ln\frac{x}{2}\cdot\cos 2x-\frac{x}{4}$	0,5	1,6	10^{-4}	Maximum
25	$5{,}8x+\ln(0{,}2x)^{-1}-1{,}5$	0,1	1	10^{-5}	Minimum

REFERENCES

1. Маликов В.Т., Кветный Р.Н. Вычислительные методы и применение ЭВМ. – К.: Вища школа, 1999.- 212 с.
2. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран, Паскаль.- Томск, 1991.- 272 с.
3. Боглаев Ю.П. Вычислительная математика и программирование. – М.: Высшая школа. 1990.- 544 с.
4. Копченова Н.В., Марон И.А. Вычислительная математика в примерах и задачах. – М.: Высшая школа. 1972.- 366 с.
5. Калиткин Н.Н. Численные методы. – М.: Наука, 1978.- 512 с.

CONTENTS

Introduction

Laboratory Work №1	Computing of polynomial value according to the Horner's method	3
Laboratory Work №2	Primitive operations with matricies	6
Laboratory Work №3	Solution of the systems of linear equations with real coefficients	10
Laboratory Work №4	Solution of the linear equation systems with the complex coefficients	18
Laboratory Work №5	Matrix inversion	20
Laboratory Work №6	Matrix determinant computation	23
Laboratory Work №7	Location of the roots of transcendental equations	25
Laboratory Work №8	Location of the roots of algebraic equations	28
Laboratory Work №9	Refinement of the roots of transcendental and algebraic equations	30
Laboratory Work №10	Solution of non-linear equation systems	34
Laboratory Work №11	Numerical Solution of linear differential equations	39
Laboratory Work №12	Interpolation	43
Laboratory Work №13	Approximation done by the least-squares method	47
Laboratory Work №14	Numerical integration	49
Laboratory Work №15	Harmonic analysis and synthesis of the periodical functions	53
Laboratory Work №16	Search of the function extreme values by the golden section method	57

Workbook for laboratory works on “Mathematical Methods and Models” for the students of the speciality 7.090603

Compliers: S.G. Dzhura, Associate Professor.
S.V. Shlepnyov, Associate Professor
V.V. Yakimishina, Assistant Professor