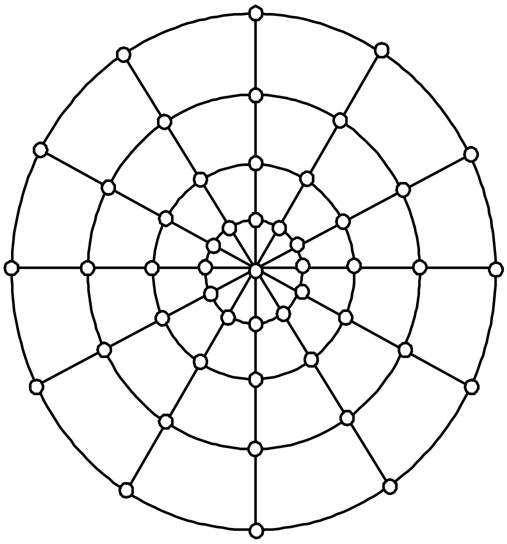
ROMAN KVYETNYY

Basics of Modelling and Computational Methods



Vinnytsia VNTU 2007

Міністерство освіти і науки України
Вінницький національний технічний університет

Р.Н. Квєтний

ОСНОВИ МОДЕЛЮВАННЯ ТА ОБЧИСЛЮВАЛЬНИХ МЕТОДІВ

Затверджено Вченою радою Вінницького національного технічного університету як навчальний посібник для студентів спеціальності 7.091401 – «Системи управління і автоматики».

Протокол N _ від "__" _______ 20__ р.

Вінниця ВНТУ 2007

УДК 681.3 (07)
К 32

Рецензенти:

Г.С. Фінін, доктор фізико-математичних наук
В.М. Лисогор, доктор технічних наук, професор
В. С. Осадчук, доктор технічних наук, професор
М.Л. Осадчук, кандидат філологічних наук, професор

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України

Квєтний Р.Н.
К 32 **Основи моделювання та обчислювальних методів.** Навчальний посібник. – Вінниця: ВНТУ, 2007. - 150 с.

Посібник присвячений розгляданню основних підходів до побудови математичних моделей складних систем та типових задач обчислювальної математики. Крім традиційних обчислювальних проблем (алгебраїчні системи, диференціальні рівняння, обробка даних) містить нові підходи такі як: фрактальні та інтервальні методи, моделювання на базі нечіткої логіки.

Призначений для широкого кола студентів та науковців з автоматики, управління, електроніки, інформатики.

Написаний англійською мовою.

УДК 681.3 (07)

Basics of Modelling and Computational Methods

Contents 3

Preface 6

Chapter 1. Modelling and Computations 7
1.1 Introduction... 7
1.2 Algorithms 8
1.3 Mathematical Modelling 12
1.3.1 Determinate Models 14
1.3.2 Stochastic Models 18
1.3.3 Fuzzy-Logic Models.. 30
1.4 Errors 34
1.5 Remarks 36
Exercises 38

Chapter 2. Problems of Linear Algebra 39
2.1 Introduction... 39
2.2 Systems of Linear Equations 39
2.2.1 Classic Methods 40
2.2.1.1 Gauss Method... 40
2.2.1.2 Gauss-Jordan Method (Exception) 41
2.2.1.3 Modified Gauss Method 41
2.2.1.4 Direct Method for Tridiagonal Systems...................................... 43
2.2.2 Iterative Methods.. 45
2.3 Remarks 47
Exercises 48

Chapter 3. Equations and Systems.. 50
3.1 Introduction... 50
3.2 Nonlinear Equations 50
3.2.1 Method of Half-Note Division (Bisection) 52
3.2.2 Method of Vicious Position (Chords) 52
3.2.3 Newton Method (Tangents) 54
3.2.4 Method of Secants.. 55
3.2.5 Method of Simple Iteration... 56
3.2.6 Problem of Complex Roots 57
3.3 Nonlinear Systems.. 58
3.4 Remarks 63
Exercises 63

Chapter 4. Ordinary Differential Equations 65
4.1 Introduction 65
4.2 Solution of the Caushe Problem 65
4.2.1 One-step Methods 66
4.2.2 Multi-step Methods 71
4.2.3 "Rigid" Problems 74
4.3 Solution of the Boundary-Value Problem 75
4.3.1 Method of “Shooting” 75
4.3.2 Difference Methods 76
4.4 Remarks 78
Exercises 79

Chapter 5. Differential Equations in Partial Derivatives (Mathematical Physics Problems) 81
5.1 Introduction 81
5.2 Difference Methods 83
5.3 Solution of Problems in Mathematical Physics 87
5.3.1 Elliptic Equations 87
5.3.2 Hyperbolical Equations 88
5.3.3 Parabolic Equations 90
5.4 Remarks 92
Exercises 92

Chapter 6. Experimental Data Processing 93
6.1 Introduction 93
6.2 Interpolation 93
6.2.1 Difference Methods 94
6.2.2 Lagrange Interpolation 100
6.2.3 Spline Interpolation 102
6.2.4 Tridimensional Space Interpolation 104
6.2.5 Interpolation by Selfsimilar Transformations……………………108
6.3 Approximation 114
6.4 Statistical Data Processing 117
6.5 Numerical Integration 122
6.5.1 Newton - Cotes Formulas 123
6.5.2 Tchebyshev’s Formula 125
6.5.3 Gauss Formula 127
6.5.4 Algorithm of Numerical Methods 130
6.5.5 Monte - Carlo Method 132
6.6 Remarks 135
Exercises 135

Chapter 7. Interval Calculations 137
7.1 Introduction.. 137
7.2 Basics of Interval Operations... 138
7.3 Applications to Calculations... 139
7.4 Applications to Modelling of Complex System 141
7.5 Remarks 145
Exercises.. 145

Bibliography 148

Concise English-Russian-Ukrainian-Polish Dictionary of Terms 149

Preface

The purpose of this book is an exposition of methods and problems of computational mathematics and basics of computer modelling. The author attempted to generalize experience of his long-term teaching of courses on computational methods and mathematical modelling to the students of specialities related to automation, control and information-measuring technique.

This book is based on the works of L. Collatz, V. Krylov, A. Samarsky, B. Demidovitch, I. Maron, J. Forsythe, R. Moor etc. The author has already published such textbooks as “Computational methods and applications of computers” (in Russian: in 1989, “High school” publication, Kiev, co-author V. Malikov) and “Methods of computerized calculations” (in Ukrainian: 2001, “VSTU” publication, Vinnytsia). In the nineties the author, together with V. Dubovoy, published a series of textbooks on the use of computerized systems. Besides the traditional sections of computational mathematics, this book contains a wide range of the author’s methods of probabilistic and interval analysis, fractal and selfsimilar processing and algorithms, formulas of multidimensional interpolation from theses, monographs and in-process approbations. The sample programs of calculations are not contained in the book, and support is done on methods and algorithms, that do it more independent on the time and tastes of the programmers. The absence of a lot of conclusions and theorems simplifies the perception of the book and at the same time the book is oriented on students, engineers and scientists designing applied problems.

The author expresses gratitude to the colleagues and students who helped him in the process and registration of results of this book I. Bogach, R. Boyko, A.Tchikalova, O. Skidan and to Ilona Kvyetna and Valery Doroshenko who edited the English text.

Chapter 1. Modelling and Computations

1.1 Introduction

Mathematical modelling is one of the main ways of scientific and technical researches. Objects replacement by their mathematical models allows to limit costs and term of researches, and also to take into account boundary situations and situations that are hard to realize. A mathematical model is a projection of objective reality under a certain point of view, as it is described in mathematical language. The mathematical apparatus chosen to describe the model can be different and concerns the researcher’s purpose of design, convenience, traditions and tastes. The purpose of design determines what descriptions of object are taken into account during construction of the model and what features are unimportant in this consideration.

The methods of modelling are widely used in different fields of human activity, especially in the fields of planning and management, where processes of acceptance of effective decisions are based on the received information.

Obtaining, transformation, presentation and use of information are the purposes of object modelling where the objects cooperate with each other and with the external environment.

A model is always built with a certain goal which has influence upon the choice of properties of the objective phenomena to be taken into account as substantial. A model is a projection of the objective reality under a certain point of view. Sometimes, depending on aims, it is possible to get different, even contradictive, projections of the objective reality. It is characteristic as a rule for complex systems in which every projection selects substantial data for a certain goal from a great number of unimportant ones.

The theory of modelling is a field of science that studies methods of research of properties of objects (originals) on basis of their substitution by other objects (models).

We will consider one of the most universal types of modelling - a mathematical one, which puts a system of mathematical equations in accordance with the designed physical process. Resolving the equations allows to get an answer to the question about the existence of an object without creation of a physical model that often leads to huge expenses of time and money.

The given mathematical model requires a decision in order to get an obvious analytical or numeral kind of the required object’s descriptions. As most applied tasks can not be decided by traditional mathematical methods, then modern mathematical modelling is inseparably related to the methods of computer calculations. These calculation methods take place first of all in the computer design because of the number of tasks (for example, nonlinear equations higher than the third order, systems of nonlinear equations, improper or numerical sets of integrals etc) that in general do not have an analytical decision; in other tasks (for example, approximation, interpolation, statistical treatment) decisions exist only in the subsections of the applied mathematics.

Many phenomena and processes of different nature are described by similar correlations, for example electro-acoustic analogy, electro-, magneto- and hydrodynamics. Therefore, for the analysis (decision, calculation) of mathematical models it is necessary to have developed mathematical skills covering all types of model tasks of the applied mathematics. As it applies to the use of computer, the basic stage of calculation of mathematical models is their realization, i.e. development of structure of the algorithm, presented as a flow-chart, flowgraph or realization with the use of principles of structured programming.

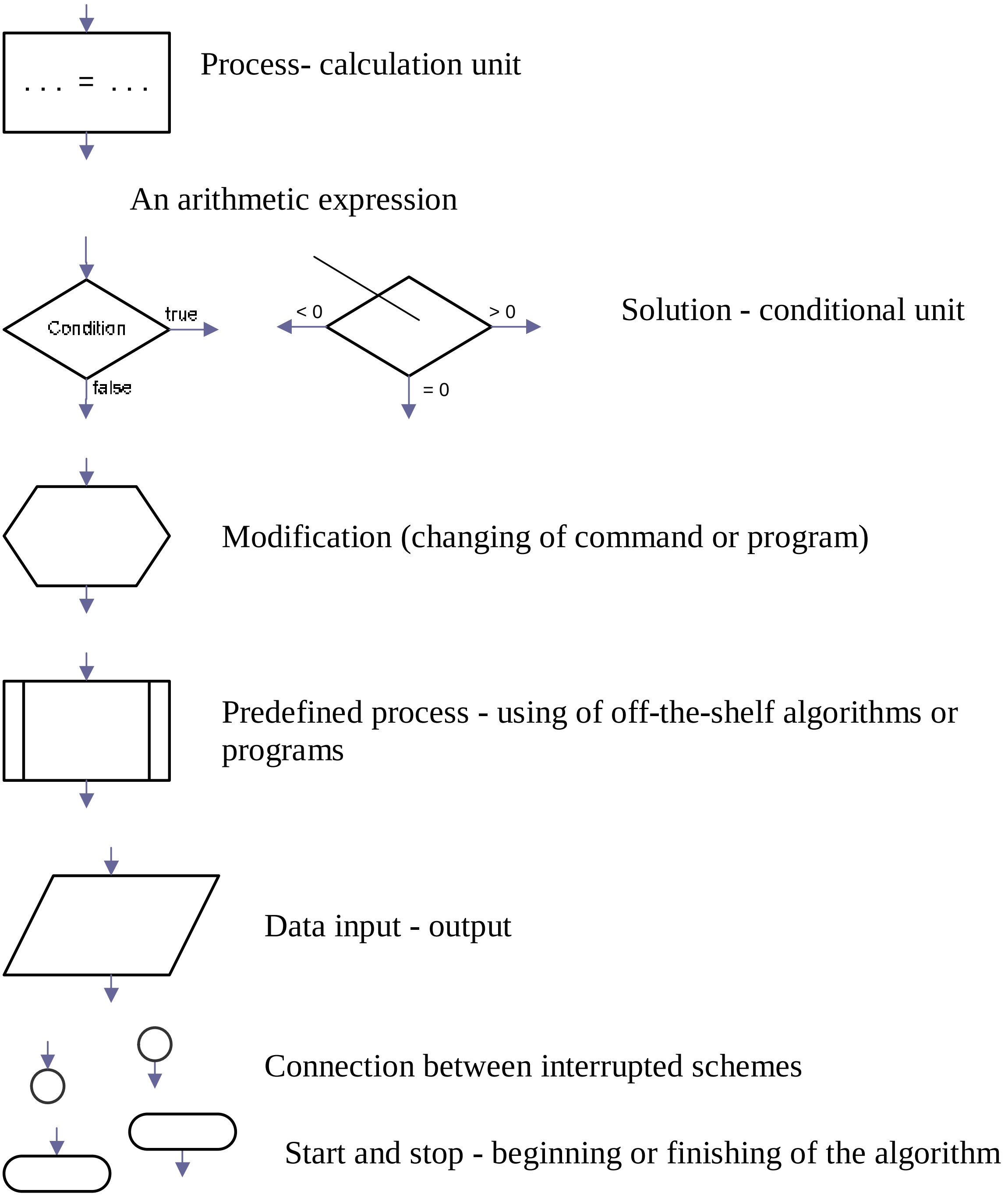
1.2. Algorithms

The term «algorithm» («algorism») was introduced by an Uzbek mathematician Al-Khwarizmi, who developed the rules of arithmetic actions above numbers in decimal notation in the 9th century.

Algorithm is a rule. These rules are formulated in a certain language and determine the process of transformation of possible basic data to the certain results after. An algorithm is characterized by: determinedness (definiteness) - uniqueness of result of the process at the basic data set; discreteness - dismemberment of algorithmic process to separate elementary acts, possibility of implementation of which by a man or machine does not cause doubting; mass character of the basic data - it is possible for an algorithm to choose from some great amount of information (potentially endless); clearness for a performer.

It is possible to select blocks that are an aggregate of elementary operations executing certain function in the structure of algorithms. Model blocks are in the flow diagram of algorithm: process, decision, modification, predefined process, input-output, connector, start-stop. Their conditional denotations are presented in Figure 1.1.

An algorithm can have linear, branching and cyclic structures. A linear structure is characterized by absence of conditional blocks. In Figure 1.2 an example of algorithm is given for the decision of problem of content changing between two computer memory cells *R* and *P.* A necessity appears in the use of the third cell *C*.



Process- calculation unit

An arithmetic expression

Solution - conditional unit

Modification (changing of command or program)

Predefined process - using of off-the-shelf algorithms or programs

Data input - output

Connection between interrupted schemes

Start and stop - beginning or finishing of the algorithm

Figure 1.1

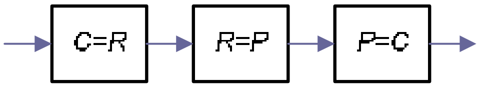


Figure 1.2

As an example of branching structure of algorithm with only one conditional block can serve an algorithm of choice with the most variable values of *N* and *M* (Figure 1.3). Revolving engineering tasks of cyclic structure (Figure 1.4) are more widespread, where *1* is preparation for the first implementation of the cycle body; *2* is the cycle which names repeatedly repetitive part of the calculable process body; *3* is preparation for the next implementation of the cycle body; *4* is implementation of verification at the end of the cycle.

There are the following types of cycles: with the set or calculated number of reiterations; iterative, in which the number of reiterations is unknown in advance; complex - with the fork in the cycle body and the embedded loops (multiple).

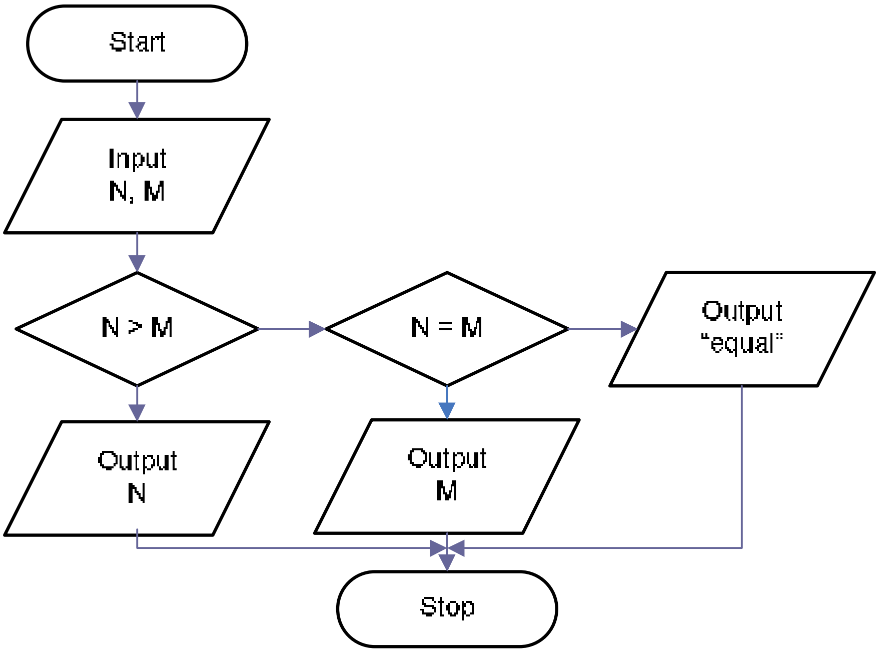


Figure 1.3

There are other ways to record algorithms - flowgraphs or graph-schemes. Boolean arithmetic operators are used for presentation of the statement chart of algorithm. Arithmetic operators provide actions related to the calculations.

We will designate operators by capital letters of the Roman alphabet with indices that indicate the number of the statement. After implementation of the operations foreseen by an arithmetic operator, the process of calculations can be continued in a unique way, regardless of the results given out by an operator. The control transfer from an arithmetic operator is designated by the statement where control is transferred to the number recorded on the right above the operator.

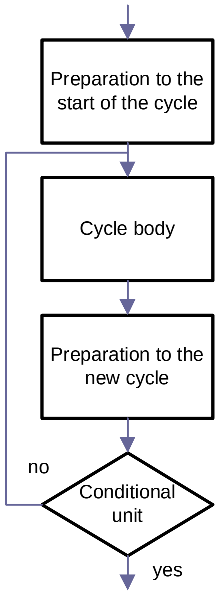

Preparation to the start of the cycle
Cycle body
Preparation to the new cycle
Conditional unit
no
yes

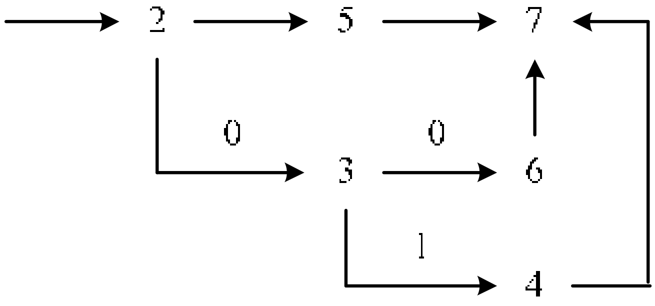
Figure 1.4

For example, a record A_P^S means that from the operator A_P control is transferred to the operator with number *S*.

Boolean operators are intended to verify implementation of the set of terms. We will mark them by letter *P* that indicates the number of the statement. After realization of the Boolean operator control is transferred to one of two operators, depending upon implementation of the condition checked up. The control transfer from a Boolean operator is designated by pointers with the statements where control is transferred to numbers. For example, $P_{K\downarrow j}^{\uparrow i}$ means that from a Boolean operator P_K control is transferred to the operator with number *i*, if the condition checked up by an operator is executed, or to the operator with number *j*, if it is not executed. For operators, both arithmetic and logical, denotation of transfer from one operator to another, directly after the following, is dropped.

The control transfer to this operator from the other is designated by the statement in which control is transferred from the number recorded on the left above the character of this operator. For example, the record ${}^{l,\,n}A_m$ means that control is transferred to the operator A_m from operators with the numbers *l* and *n*. In this case algorithm, the structure of which is presented in Figure 1.3, can be written as follows (blocks are designated by numbers):

Graph-scheme



Operator scheme

$1\ \ 2_{\downarrow 3}\ \ 5^{7}\ \ 3_{\downarrow 6}\ \ 4^{7}\ \ 6\ \ {}^{4,5}7$

Recently methods of structured programming have become very popular, where three types of structures of algorithms are uniquely used: Articulations, Choice and Iteration. Here, the basic method of creating the programs is an algorithm of the incremental working out in detail, in which without drafting a flow-chart a programmer gradually moves in the text of the program, consistently organizing and going into detailed layers, proper abstractions to the different levels, using the special universal structured programming language, for example PDL. The program thus made can be easily and simply translated into any, comfortable to the user, programming language.

Finding algorithms of decision of different classes of tasks is one of the aims of mathematics. The purpose of applied mathematics, as it applies to the use of computer, is finding decision algorithms of practical (engineering) tasks with use of computer

1.3 Mathematical Modelling

Mathematical models, being projections of the real objects, are characterized by a number of features.

Mathematical modelling can be used as means of studying the real systems by their substitution with more comfortable for experimental research systems (models), preserving the substantial features of the original.

A model is called isomorphous (identical in form), if it has complete coincidence with the real system; and it is called homomorphous, if there is accordance only between the most important components of the object and of the model.

The mathematical design includes the following stages: study of the object and drafting of its mathematical description; construction of an algorithm describing the model of the object; verification of the model’s and of the object’s adequacy; the model’s implementation.

Study of the object of design and drafting its mathematical description consists of establishment of connections between the parameters of the process, exposure of its initial conditions and formalization of the scopes of the process as a system of mathematical correlations.

Mathematical description is made on basis of physical, chemical and other laws, characterizing dynamic and static processes in the object explored, and is written in mathematical language. Most distributions in the process of construction of the determined models were obtained by means of algebraic equations, differential equations and differential equations in private derivative, matrix algebra, and stochastic design, when casual character of processes is taken into account, together with methods of probability and mathematical statistics theories. If an apriori information on the object is insufficient, the type of mathematical models is specified by methods of multidimensional statistics: regression, cross-correlation, multivariable, other analyses and also by means of planning passive or active experiments. The principles of models construction are divided into analytical ones and imitations. Analytical models allow either to get an obvious functional, depending on the sought sizes or to define the numeral decisions for the concrete initial conditions and quantitative descriptions of the model. However, with more complex objects of design the construction of an analytical model grows into a hard solving problem. Also nowadays there is a wide distribution of simulation models, when the experiments are conducted on computer, with mathematical models imitating the behaviour of the real objects. Features of the objects' functioning, the designs and types of mathematical descriptions used, determine continuous or discrete character of the model, choice of the determinate or stochastic approach to the model's construction. For example, in order to design functions of measuring the devices' transformation it is sufficient to use the determined method of description, while for the errors analysis, for estimations of the informative descriptions stochastic methods must be applied.

The method of mathematical modelling allows to eliminate the necessity of making bulky physical models, related to the financial expenses; to reduce time of descriptions determination (particularly in case of calculating mathematical models on computer and applying effective calculable methods and algorithms); to study the conduct of design object at different values of parameters; to analyze applicability of different elements; to get descriptions and indices which are difficult to obtain experimentally (cross-correlation, frequency, self-reactance sensitiveness).

We will consider the basic methods of mathematical models construction, more widespread in automation, management, information-measuring techniques.

Generally, the mathematical model of device, system, process appears as the system of functional

$$\hat{O}\,(X, Y, Z,\ t) = 0, \qquad (1.1)$$

where *X, Y* are the vectors of entrances and outputs coordinates; *Z* is the vector of external influences; *t* is the coordinate of time.

The method of presentation of *Φ* depends on the aims of design, setting of the object, volume of information and character of basic data. In future, for short we will use determination of the type of model by the following denotations:

first letter: D – determinate model, U –model in conditions of uncertainty;
second letter: A – analogue, D – digitized;
third letter: A – analytic model, S – simulation.

1.3.1 Determinate Models

The behaviour of most technical systems can be described via the so-called phases variables - physical sizes as a stream and potential. It is thus expedient to select in the design objects the large enough elements to be examined as indivisible units. The laws of the elements of the system functioning are set by the components equations relating to heterogeneous phases variables.

Community of the processes description, which is characteristic to different technical systems, allows to select a few types of elements: *R* is the element of energy dispersion; *C* and *L* are the elements of energy accumulation. We can get the equivalent chart of the technical system of any complication and the mathematical model of combination of these simplest elements and sources of the phases variables. Concrete sense of phases variables and simplest elements of the physical systems is resulted in Table 1.1.

The mathematical model as basic description of many technical objects is a system of nonlinear differential equations in general case (1.1). A similar system can be solved in whole case only via numerical methods, replacing a continuous independent variable by its discrete analogue. This operation determines the retype model on DAA.

Aggregate of phases variable values and their derivatives on the step of integration turns out as a solution of the system of *n* algebraic equations (in general case nonlinear) with *n* unknown $x_1, x_2, \dots, x_n$.

$$\begin{gathered} f_1\,(x_1,\ x_2,\ \dots\ ,\ x_n) = 0; \\ f_2\,(x_1,\ x_2,\ \dots\ ,\ x_n) = 0; \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ f_n\,(x_1,\ x_2,\ \dots\ ,\ x_n) = 0. \end{gathered}$$

Solution of such a system of equations is possible by iterative methods, among which the Newton's method is the most widespread. This method is based on Jacobi matrix,

$$W(X) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \frac{\partial f_1}{\partial x_2} \cdots \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} \frac{\partial f_2}{\partial x_2} \cdots \frac{\partial f_2}{\partial x_n} \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ \frac{\partial f_n}{\partial x_1} \frac{\partial f_n}{\partial x_2} \cdots \frac{\partial f_{n1}}{\partial x_n} \end{bmatrix}.$$

Dependence between the unknown sizes of phases variables and their derivatives used in equations (1.1) is connected with different methods of approximation.

The system of equations (1.1) is an association of components, topologies and different equations. Topologies equations set connection between the homogeneous phases variables related to the different elements of the system. Such equations in most physical systems are based on equations of equilibrium and continuity (for example, system of equations of the first and second laws of Kirchhoff).

As an example we will consider the mathematical model of a bipolar transistor.

The equivalent chart of a bipolar transistor is presented in Figure 1.5. The following denotations are accepted here: $I_{\AA}$, $C_{\AA}$, $R_{y\AA}$, I_K, C_K, R_{yK} are accordingly elements of ***p-n*** transitions of emitter-base and collector-base; $I_r = bI_E - b_I I_K$ is a source of current, reflecting passing of non-base carriers through the base and determining amplifying properties of transistor (b and b_I are normal and inversion amplification coefficients of the current); r_E, r_k and r_B is volume resistance of the regions, of accordingly emitter, collector and base.

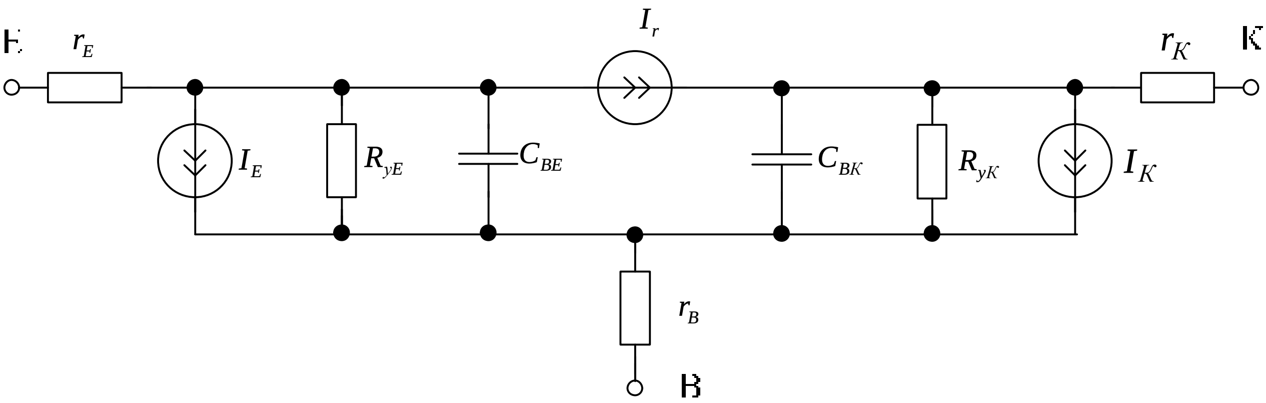


Figure 1.5.

We write components equations for every element. We will get the following system of equations:

$$\left.\begin{array}{l}
I_{rE} - U_{rE} / r_E = 0; \\
I_{RyE} - U_E / R_{yE} = 0; \\
I_E - I_{\grave{O}E} \exp(U_E / mj_{\grave{O}E} - 1) = 0; \\
I_{CE} - [C_{BE} + t/(mj_{\grave{O}E})(I_E + I_{\grave{O}E})] \overset{\&}{U}_{\tilde{N}E} = 0; \\
I_r - bI_E + b_I I_K = 0; \\
I_{CK-}[C_{BK} + t_P / mj_{\grave{O}\hat{E}} (I_{\hat{E}} + I_{\grave{O}\hat{E}})] \overset{\&}{U}_{\tilde{N}\hat{E}} = 0; \\
I_{\hat{E}} - I_{\grave{O}\hat{E}} \exp(U_{\hat{E}} / mj_{\grave{O}\hat{E}} - 1) = 0; \\
I_{Ry\hat{E}} - U_{\hat{E}} / R_{y\hat{E}} = 0; \\
I_{rK} - U_{rK} / r_K = 0; \\
I_{rB} - U_{rB} / r_B = 0;
\end{array}\right\} \qquad (1.2)$$

where $I_{\grave{O}E}$ is thermal current of transition base emitter; m is an empiric coefficient; $j_{\grave{O}E}$ is temperature potential of emitter; $\tilde{N}_{BE}$ is barrier capacity of transition base emitter; $\tilde{N}_{B\hat{E}}$ is barrier capacity of transition base collector; $j_{\grave{O}\hat{E}}$ is temperature potential of collector; $I_{\grave{O}\hat{E}}$ is thermal current of transition base collector; t_P, t - parameters, characterizing time of passing of current carriers through the regions of transistor.

Unknown variables are here

$I_{rE}, I_{RyE}, I_E, I_{CE}, I_r, I_{CK}, I_K, I_{RyK}, I_{rK}, I_{rB}, U_{rE}, U_E, U_{CE}, U_{CK}, U_K, U_{rK}, U_{rB}, U_r$.

It ensues from this list that some topologies equations are taken into account in a model: voltages $U_{\tilde{N}E}$ and U_{CK}, U_{R} and U_{RK} are eliminated because they coincided accordingly with voltages U_{E} and U_{K}.

We'll write topologies equations of the system:

$$\left.\begin{aligned}
&I_{rE}-I_{RyE}-I_{E}+I_{CE}-I_{r}=0;\\
&I_{r}+I_{CK}-I_{\hat{E}}-I_{RK}-I_{rK}=0;\\
&I_{RE}+I_{E}-I_{CE}-I_{CK}+I_{\hat{E}}+I_{Ry\hat{E}}-I_{rB}=0;\\
&U_{E}-U_{\hat{E}}-U_{r}=0;\\
&U_{rE}+U_{E}+U_{rB}-U_{BE}=0;\\
&-U_{\hat{E}}+U_{r\hat{E}}-U_{rB}+U_{B\hat{E}}=0.
\end{aligned}\right\} \qquad (1.3)$$

In the last two equations U_{BE} and $U_{B\hat{E}}$ are voltages of accordingly base emitter and base collector. Difference approximations for derivatives $\dot{U}_{\tilde{N}E}, \dot{U}_{\tilde{N}K}$ with step h join this system.

Thus, the mathematical model of a bipolar transistor is the system of equations (1.2), (1.3). The Jacobi matrix for this system is presented in Table 1.1 (zeroing elements are not marked).

The following denotations of coefficients are accepted in this matrix:

$$a_{1}=-\frac{I_{\grave{O}E}}{mj_{\grave{O}E}}\exp\left[U_{E}/(mj_{\grave{O}E})\right];$$

$$a_{2}=-\frac{t}{mj_{\grave{O}E}}U_{\tilde{N}E};$$

$$a_{3}=-\left[\tilde{N}_{BE}+\frac{t}{mj_{\grave{O}E}}(I_{E}+I_{\grave{O}E})\right];$$

$$a_{4}=-\frac{t_{P}}{mj_{\grave{O}\hat{E}}}U_{\tilde{N}\hat{E}};$$

$$a_{5}=-\left[\tilde{N}_{B\hat{E}}+\frac{t_{P}}{mj_{\grave{O}\hat{E}}}(I_{\hat{E}}+I_{\grave{O}\hat{E}})\right];$$

$$a_{6}=-\frac{I_{\grave{O}\hat{E}}}{mj_{\grave{O}\hat{E}}}\exp\left[U_{\hat{E}}/(mj_{\hat{E}})\right].$$

Table 1.1.

№ of equation	Variable																
	I_{rE}	I_{RyE}	I_E	I_{CE}	I_r	I_{CK}	I_K	$I_{Ry\hat{E}}$	I_{rK}	I_{rB}	U_{rE}	U_E	U_{BE}	U_{BK}	$U_{\hat{E}}$	U_{rK}	U_{rB}	U_r
1	1										$-\frac{1}{r_E}$							
2		1										$-\frac{1}{R_{yE}}$						
3			1									a_1						
4			a_2	1									a_3					
5			$-b$		1		b_I											
6						1	a_4							a_5				
7							1								a_6			
8								1							$-\frac{1}{R_{yK}}$			
9									1							$-\frac{1}{r_K}$		
10										1							$-\frac{1}{r_B}$	
11	-1	-1	-1	1	-1													
12					1	1	-1	-1	-1									
13		1	1	-1		-1	1	1		-1								
14												1			-1			-1
15											1	1					1	
16															-1	1	-1	
17												$-\frac{1}{h}$	1					
18														1	$-\frac{1}{h}$			

1.3.2 Stochastic Models

When describing objects of automation and information measuring technique it is possible to select the next types of stochastic (probabilistic) modelling.

Statistical modelling known as the Monte-Carlo method is applied except for the tasks of mathematical modelling for the solution of separate tasks of numerical methods, for example, for approximate calculation of integrals and solution of differential equations. The statistical models of complex processes can be realized both on ordinary computers (analogue and digital) and on specialized statistical computers supplied with designing blocks for generation and transformation of random numbers.

Analytical probabilistic modelling, as an approach to creation of models, operates not with concrete ordinary numerical sequences, but directly with their probabilistic (laws of probabilities) and spectral descriptions. Generally, the construction of analytical probabilistic models is an intricate calculable problem that does not allow to a full extent to use such their advantages, as the possibility of exact analytical task of descriptions of casual processes, absence of necessity of generation and treatment of large selections of random numbers, adjusted to operative optimization. The results of the researches directed to creation of problem-oriented systems and to application uniting numerical algorithms of solutions of more characteristic calculable procedures of analytical probabilistic design and methods of description of structures of the system packages are described in special literature.

Currently the method of statistical modelling on computer, operating with models as UDS, is the basic method of stochastic modelling. Often this type of modelling is named simulation.

The method of statistical modelling includes several stages: computer modelling of pseudo-random numerical sequences with the set correlation and law of probabilities as an imitation of entrance signals and influences on the object of simulation; modelling of transformation of the numerical sequences in the system; statistical treatment of results of modelling. We will consider these stages.

Computer modelling of pseudo-random numerical sequences with the set descriptions. When constructing the simulation model of the system there is a task of receipt by computer of pseudo-random numerical sequences with the set correlation and the law of probabilities distribution. A method of receipt of numerical sequences is known with the set statistical descriptions by sorting the initial sequences. This method is based on the fact that the coefficient of correlation of random numbers depends more on the order of their sequence, than on the size. Therefore, two pseudo-random sequences, belonging to two different distributions, if they are well-organized by identical appearance, will have approximately equal coefficients of correlation.

In accordance with the method of sorting a pseudo-random sequence $X(n)$ is generated with the set cross-correlation function, but arbitrary distribution. The sequence of integers is put to it in accordance $I(n)=n$. Then both sequences in pairs are assorted. Thus, variables $X(n)$ are disposed in ascending order, and an array $I(n)$ memorizes their previous position (places in an unregulated array $X(n)$). Thus, an integer array $I(n)$ represents correlation between the array cells $X(n)$. After organization the array $X(n)$ does not present any interest, because all information about the cross-correlation function is now contained in the array $I(n)$. A pseudo-random sequence $Y(n)$ is then generated with the set distribution and zeroing correlation and is written instead of the array $X(n)$. Then it is assorted in the multiplied order. Further arrays $I(n)$ $I(n)$ and $Y(n)$ are assorted in pairs; thus an array $I(n)$ is disposed in an increasing order. The flow diagram of an algorithm is resulted in Figure 1.6.

As a result of implementation of this algorithm we will get a pseudo-random numerical sequence, containing the up-diffused sizes on the set law and having the set cross-correlation function. It is expedient to use a sorting algorithm in those cases when for the statistical design of the system there is a small volume of statistical information not requiring the operation with numerical bulk arrays. In case of bulk arrays the time of design is substantially increased.

The known algorithm of filtration requires less expenses of machine time to get casual process with the set correlation and law of probabilities distribution. A normal stationary casual process starts as initial $X(t)$. Always there is such a nonlinear fast-response transformation $Y = W_N(X)$ which converts normal function of probability density $f_X(x)$ of process $X(t)$ in the set functions of density $f_Y(y)$. If an initial process $X(t)$ has a cross-correlation function, a regenerate process $Y(t)$ will have a cross-correlation function $R_{YY}(t)$ different from the function $R_{XX}(t)$ and related to it by some dependence $R_{YY} = j(R_{XX})$. The type of this dependence is reflected by transformation $Y = W_N(X)$. If the cross-correlation function of the regenerate process is required, it is necessary to choose the cross-correlation function of the initial process:

$$R_{XX}(t) = j^*[R_{YY}(t)],$$

where j^* is reverse function j .

Before using this method, preparatory work has to be done, consisting of a few stages:

- finding transformation function $Y = W_N(X)$ on the set function of density $f_Y(y)$;

- getting dependence $R_{YY} = j(R_{XX})$ from the found function $Y = W_N(X)$;

- solution of equation $R_{YY} = j\left(R_{XX}\right)$ in relation to R_{XX}, determination of cross-correlation function $R_{XX}\left(t\right)$ of the initial normal process $X(t)$.

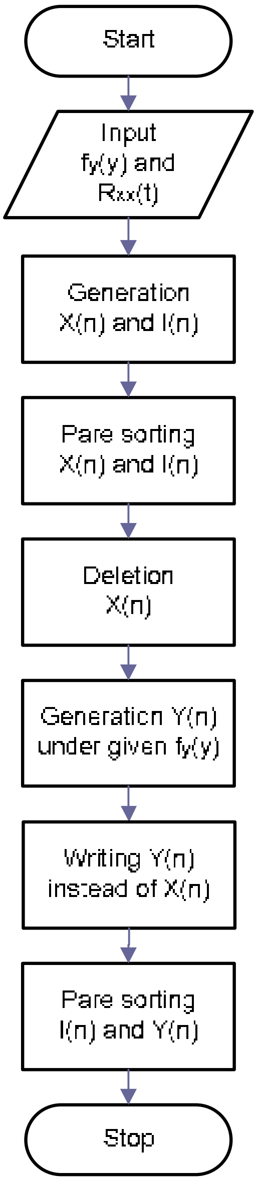


Figure 1.6

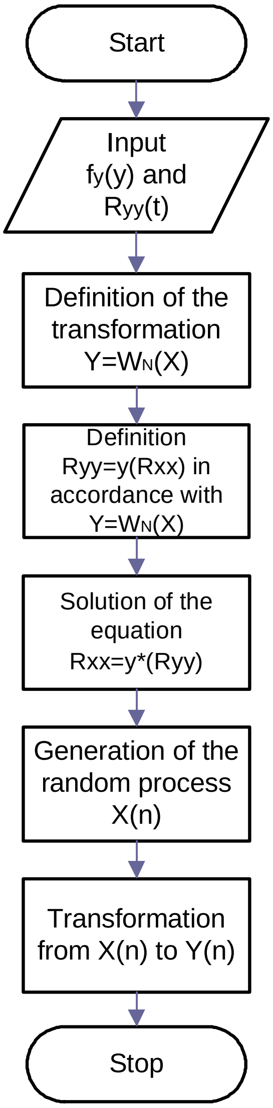


Figure 1.7

After ending the preparatory work, the design of random process with the set descriptions is taken to forming the discrete realization $X(n)$ of normal random process $X(t)$ and to transformation of this realization by the formula

$$Y(n)=W_N\{X(n)\}.$$

The flow diagram of the described algorithm is resulted in Figure 1.7.

The algorithm requires less computer time than sorting algorithm, does not require accumulation and storage in memory of large numerical arrays. The principle difficulty is that in general it is not possible to prove existence of the equation's solution of $R_{YY}(t)=j\,[R_{XX}(t)]$ relative $R_{XX}(t)$.

Applying both algorithms there is a task of generating on computer of pseudo-random numerical sequences with the set laws of distribution, zero correlation and pseudo-random numerical sequences with the set cross-correlation function and arbitrary distribution. Generation of the random numbers with the set law of probability distribution is realized in a few stages. At the beginning the sequence of pseudo-random numbers is generated on an interval [0, 1], and from it - a pseudo-random number sequence with the set law of distribution.

We will consider the algorithmic methods of random numbers generation (in practice the physical design with the use of a special prefix to computer is sometimes applied). The essence of algorithmic methods consists of generation of pseudo-random numbers that are produced by some recurrent formula, where every next $(i+1)$ value appears from previous i (or groups of previous) by application of some algorithm containing the arithmetic and logic operations.

Plenty of methods of imitation of the uniform distribution are known (take-outs, addition, truncation, interfusion methods). For all of these methods the requirements to the generated sequence of random numbers are general: the amount of operations for receipt of every pseudo-random number must be minimal; random numbers are generated as less correlated as possible, and their distribution is close to uniform, thus the type of distribution and correlation numbers degree must not change during work of the program.

In the standard mathematical and programmatic software of different types of computers there are special procedures and programs for generation of uniform distributing sequences of pseudo-random numbers.

Using a casual size generation in an interval [0, 1] *X*, it is possible to get random numbers sequence with an arbitrary set law of probability distribution. Three basic methods of forming such sequences are distinguished:

1) Direct transformation of number X_i, being realization of random variable *X*, generation on an interval [0, 1] by some function W_N, in a number y_i, which can be examined as realization of random variable *Y*, having the set distribution law;

2) screening-out numbers from the primary sequence of pseudo-random numbers generation on an interval [0, 1] so that remaining numbers are up-diffused on the set law;

3) designing of terms of the proper maximum theorems of probability theory.

Widely spread are methods of speed-up generation of random numbers. Thus, a considerable effect of increasing speed during imitation of the random numbers distributed on a normal law, as compared to the method based on the use of central limit theorem of probability theory, gives a method, including the Muller algorithm, by which a pair of independent numbers from the segment [0, 1] is transformed to the pair of independent normal random distributed variables

$$x_1 = \sqrt{-2\ln x_2}\cos 2px_1\,;\; x_2 = \sqrt{-2\ln x_1}\cos 2px_2\,.$$

We will note that this method is theoretically exact and requires the least amount of generated numbers $X(n)$.

Special methods of getting random variables with different laws of probabilities distribution are known. For example, Rayleigh distribution with one parameter equal to mean quadratic deviation of initial two-dimensional normal distribution. It leads to the following method of imitation of Rayleigh distribution:

$$h_i = s\,\sqrt{x_1^2 + x_2^2}\,,$$

where h_i is the random variable distribution by law of Rayleigh, x_1 and x_2 are random numbers having normal distribution with the zero expectation that mean quadratic deviation equal to unit.

There is also a correlation, relating the random numbers generated by law of Rayleigh, with uniform random numbers distribution on a segment [0, 1], which determines another way of generation:

$$h_i = s\,\sqrt{-2\ln x_i}\,.$$

For imitation of the Maxwell distribution law it is possible to take advantage of the random variable where Maxwell distribution can be examined as a module of three-dimensional random vectors, the projections of which on the axis of coordinates submit to normal distribution with equal mean of quadratic deviations and expectations that equal to zero. It is therefore possible to take advantage of the following formula for imitation of the Maxwell distribution law:

$$q_i = s\,\sqrt{x_1^2 + x_2^2 + x_3^2}\,,$$

where the random number is generated by the law of Maxwell; x_1, x_2, x_3 are random numbers with normal distribution and the expectation that equals to zero, and mean quadratic deviations equal to unit.

We will consider the methods of getting pseudo-random number sequences with the set cross-correlation function. An algorithm is known where the n uncorrelated random numbers $x_i, \dots, x_n$ are subject to such linear transformation after which the obtained sequences $y_i, \dots, y_n$ have the set cross-correlation matrix $K(R_{nm})$. Thus, means $y_i, \dots, y_n$ could be found from matrix equation

$$K(Y) = W_N[K(x)],$$

where W_N is a linear transformation of the vector-column $K(X)$ in $K(Y)$.

In the unfolded form we get:

$$\begin{aligned}
&y_1 = a_{11}(x_1 - m_x) + m_{y1};\\
&y_2 = a_{12}(x_1 - m_x) + a_{22}(x_2 - m_X) + m_{y2};\\
&\dots\dots\dots\dots\dots\dots\dots\dots\\
&y_n = a_{1n}(x_1 - m_x) + \dots + a_{nn}(x_n - m_X) + m_{Yn},
\end{aligned}$$

where the transformation coefficients are found from the equation

$$R_{lK} = a_{1l}a_{1k} + a_{2l}a_{2k} + \dots + a_{ll}a_{lk}$$

and $K[m_Y]$ is vector-column of the expected values y_i.

At large values of n this method of generating the correlated pseudo-random sequences becomes inconvenient for realization on computer, as memorizing the elements of matrix $K(a)$ requires a large volume of main memory $(N = n(n+1)/2)$ of cells and large volume of calculations (expenses of machine time). Due to this in a number of cases it appears more comfortable to design the correlated random processes via the method of canonical decompositions. Let a continuous random process $Y(t)$ be set by canonical decomposition

$$Y(t) = \sum_{k=1}^{\infty} V_k J_k(t), \tag{1.6}$$

where V_k is the uncorrelated random coefficient; $J_k(t)$ is a system of certain determined coordinate functions.

Digital design of random processes, set by canonical decomposition, is carried out as follows. The values of the uncorrelated casual sizes X_k are used as V_k. Infinite row (1.6) at the calculations is approximately replaced by the truncated eventual row. Using canonical decomposition, we will get relation

$$Y_n = \sum_{k=1}^{N} V_k J_k(n),$$

in which dispersions D_{Vk} of the uncorrelated casual sizes x_k and discrete coordinate functions $J_k(n)$ could be found from the following recurrent relations:

$$\begin{aligned}
&D_{Vk} = R_{YY}(1,1);\\
&J_1(n) = \frac{1}{D_{V_1}} R_{YY}(n,1);\\
&D_{Vk} = R_{YY}(k,k) - \sum_{i=1}^{k-1} D_{Vi} J_i(k),\ k = \overline{2,\ N};\\
&J_k(n) = \frac{1}{D_{Vk}}\left[R_{YY}(n,\,k) - \sum_{i=1}^{k-1} D_{Vi} J_i(n) J_i(k)\right].
\end{aligned}$$

We get from here

$$Y_n = \sum_{k=1}^{N} s_k x_k J_k(n).$$

A sequence Y_n will have normal distribution and set cross-correlation function $R_{XX}(t)$.

Methods of sorting and filtration in the combination with the considered methods of generating pseudo-random numerical sequences with the set cross-correlation functions allow to get random numbers sequences imitating entrance signals and revolting influences to the object of design.

Simulation of processes of the random signal transformation. We will consider methods of designing algorithms for modelling of processes of random transformation processes via different transformers and systems. At the design of random signal transformation by the linear dynamic systems it is efficient to use their impulsive description $g(t)$.

In order to get a digital model of the entrance casual signal transformation it is necessary to design operation packages of functions $x(t)$ and $g(t)$.

Carrying out replacement of integral of Duhamel by the sum of discrete values on the method of rectangles, we get

$$y(n) = \sum_{k=0}^{n-m} g(k)x(n-k), \tag{1.7}$$

where *m* is a discrete analogue of the transient’s duration.

There are more exact methods of integration - trapezoids, Simpson etc. In such cases the formula (1.7) assumes the following form

$$y(n) = \sum_{k=0}^{n} c(k)g(k)x(n-k),$$

where coefficients $c(k)$ are determined depending on the used method of numerical integration.

The substantial advantage of method of design of the linear dynamic systems on basis of discrete analogy of integral packages is the possibility to generalize in case of designing the linear dynamic systems with variable parameters (non-stationary systems).

A non-stationary system is described by impulsive description depending upon two variables. In this case as a random signal we mark the reaction of the non-stationary system as

$$y(t) = \int_0^t x(t\ )g(t\ ,\ t - t\ )dt$$

or in a discrete form

$$y(n) = \sum_{k=0}^{n-1} x(k)g(k, n-k)c(n-k).$$

The described method is used in the cases when modelling of the input random signal is carried out with help of the algorithms of nonlinear transformation filtration and designing of terms of central limit theorem of probability theory. The flow diagram of the design algorithm is presented in Figure 1.8. We will note that in the case of designing entrance casual signal on basis of the sorting algorithm (see Figure 1.6) it is impossible to build a simulation model of the system and to explore it expediently by matrix methods. Digital models of the closed nonlinear systems could be used as a combination of the modelling algorithms for separate linear dynamic and nonlinear static transformers (Figure 1.9).

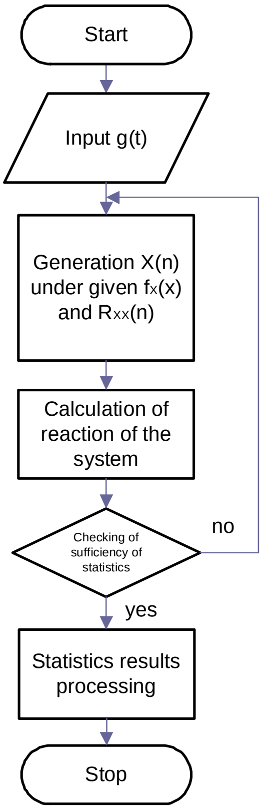

Start
Input g(t)
Generation X(n) under given fx(x) and Rxx(n)
Calculation of reaction of the system
Checking of sufficiency of statistics
no
yes
Statistics results processing
Stop

Figure 1.8

Design of the described systems is often connected with considerable difficulties, however, in a number of cases, the flow diagram of the system can be presented in the simplified form (Figure 1.9).

In this case

$$e(n) = x(n) - y(n);$$
$$e_1(n) = W_N[x(n) - y(n)],$$

where W_N is an operator of nonlinear transformer of the system.

Applying the described above algorithm for the digital designing of linear dynamic systems, we get

$$y(n) = \sum_{k=0}^{n-1} g(k) e_1(n-k).$$

Thus, the necessity of every step in resolving systems of nonlinear algebraic equations is a feature of digital models of the nonlinear closed systems, providing that the linear dynamic links of the system are designed on basis of discrete packages. The resolving of this task can be simplified, if we enter the element of delay for one period in the chain of feed-back of the system.

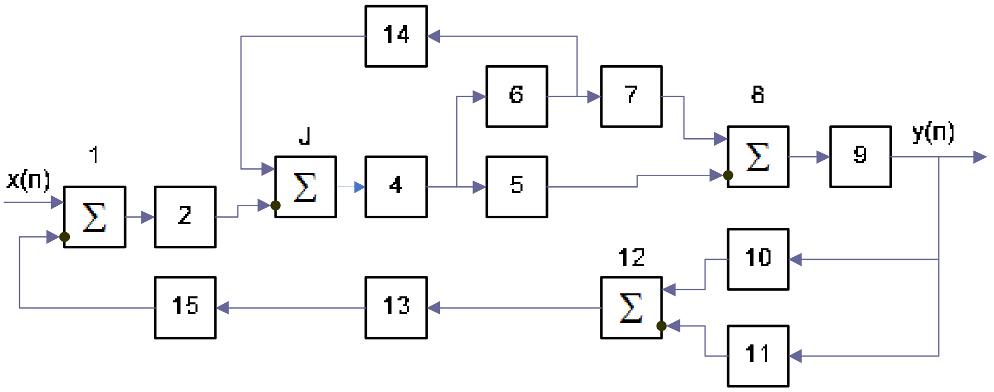


Figure 1.9

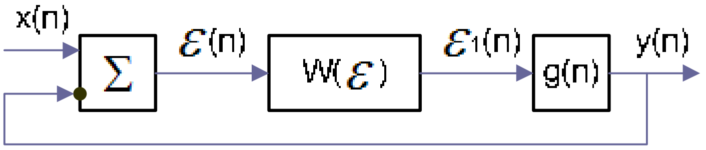


Figure 1.10

The nonlinear equation (Figure 1.8) will then be transformed into the recurrent form:

$$y(n) = \sum_{k=0}^{n-1} g(k) W_N \{x(n-k-1) - y(n-k-1)\}.$$

Introduction to the chain of feed-back from the element of delay brings in an additional error into the digital model. However, at the digitization step $t_{\Delta} \to 0$ an equivalent discrete system with an element of delay being the same as without it coincides with the initial continuous system. Therefore, by choosing a digitization step it is possible to obtain a considerably small impact of the delay error.

Statistical treatment of results is the final stage of statistical design.

Amount of realization and exactness of calculations. The amount of realization of the tasks resolved via method of statistical designing depends on the required exactness of the results to be obtained.

Let the purpose of design be calculation of probability *P* appearance of some random event *A*. As an estimation of probability *P* uses frequency of the *L/N* of event presence *A* at *N* realizations, where *L* is amount of tests at which an event is *A*. By virtue of the central limit theorem of probability theory frequency *L/N* at large enough value *N* has the normal distribution determined by the expected value *M(L/N) = P* and dispersion *D(L/N) = P (1-P) / N.*

Consequently,

$$P(L/N) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2pD(L/N)}} \exp\left\{\frac{-[(L/N)-M(L/N)]^2}{2D(L/N)}\right\} d(L/N). \qquad (1.9)$$

At large enough value *N* gets

$$P\left[\frac{(L/N)-M(L/N)}{\sqrt{D(L/N)}} < \frac{e}{\sqrt{D(L/N)}}\right] \approx \hat{O}\left[\frac{Ne}{\sqrt{D(L/N)}}\right], \qquad (1.10)$$

where $\hat{O}(z) = \frac{2}{\sqrt{p}} \int_0^x e^{-z^2} dz$; e is exactness of inequality.

Set by certain probability *P*, we will find on normal distribution the value *D(L/N)*, satisfying to equation, where $t_p = e / D(L/N)$.

We get the confidence estimation *L/N* in a form

$$P[|(L/N) - P| < e] = t_P \sqrt{P(1-P)/N} \ .$$

In a formula (1.10) with probability greater than 0.997, the size *L/N* satisfies the condition

$$|(L/N) - P| < 3\sqrt{P(1-P)/N} \ .$$

Thus, error of method of statistical modelling while calculating the probability of event *A* never exceeds sizes $e = 3\sqrt{P(1-P)/N}$ and decreases with the increasing number of tests inversely proportional to a root square from $\sqrt{N}$. Hence, it is possible to define the amount of realization *N*, necessary to get the estimation *L/N* with exactness e and truth *P*

$$N = t_p^2 P(1-P)/e^2$$

or for *P =0.997*

$$N = 9P(1-P)/e^2 .$$

It is likely possible to estimate the amount of realization necessary for estimation of the results of the modelling of random variation's mean. We will suppose forming of *N* realizations of random variation *X*, with the expectation M and dispersion s^2. We will define

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} Xi \,.$$

By virtue of central limit theorem of probability theory

$$P\left(\frac{\overline{X}-M}{s\sqrt{N}} < e\sqrt{N}/s\right) \approx \hat{O}\left(e\sqrt{N}/s\right).$$

Then

$$P\left[\left|X-M\right| < e\right] = t_p s \,/\sqrt{N}\,,$$

exactness

$$e = t_p s \,/\sqrt{N}\,.$$

At P = 0.997 formulas acquire a form accordingly e and N

$$e = 3s\,/\sqrt{N}, N = 9s^2/e^2\,.$$

Error of method of statistical modelling both at the calculation of probability of event *A* at the estimation of mean random varieties makes $e = 1/\sqrt{N}$. Diminishing of error e of close solution of task via the method of probabilistic modelling leads to the considerable increase of number of tests of N and to increase of time of calculations. For example, the increase of exactness around to an order leads to increasing the time of resolving the task one hundred times.

1.3.3 Fuzzy-Logic Models

One of the base means for modelling human-computer systems is a fuzzy-logic theory (class of models in condition of uncertainty UAA or UDA).

At development of a model the following basic concepts of fuzzy-logic model theory were used in one of the basic fuzzy logic works, offered by Zadeh:

1. Concept of universal set. The universal set of U is a complete plural that engulfs all problem areas.
2. Concept of unclear subset. Unclear subset of the F set of U concerns the function of belonging μ_F (u), where u is an element of the universal set.

3. Concept of function of belonging. The function of belonging $m_F(u)$ represents the degree of belonging of every element of universal set to an fuzzy subset of *F*. The function of belonging acquires the values from *0* to *1*.

A universal set can be both continuous and could consist of complete number of sets (or elements) $u_1, u_2, ...u_n$. In the first case, an fuzzy set appears in a form:

$$F = \int_U m_F(u)/u\,.$$

The following denotation is used in the second case:

$$F = m_F(u_1)/u_1 + m_F(u_2)/u_2 + ... + m_F(u_n)/u_n.$$

The basic operations of theory of fuzzy sets are:
1. Operation of addition of the sets:

$$\overline{F} = \sum_{i=1}^{n} (1 - m_F(u_i))/u_i,$$
$$m_{\overline{F}}(u) = 1 - m_F(u).$$

2. Operation of association of the sets:

$$F \cup G = \sum_{i=1}^{n} \{m_F(u_i) \cup m_G(u_i)\},$$
$$m_{F \cup G}(u) = m_F(u) \cup m_G(u),$$

where $\cup$ - is a sign of operation of the maximum finding.
3. Operation of crossing of the sets:

$$F \cap G = \sum_{i=1}^{n} \{m_F(u_i) \cap m_G(u_i)\},$$
$$m_{F \cap G}(u) = m_F(u) \cap m_G(u),$$

where $\cap$ - is a sign of operation of the minimum finding.

Unclear logical equations could be written down using these rules. The operations of finding the maximum and the minimum correspond to the operations of logical “and” and logical “or”. In future these operations are named as “fuzzy and”, “fuzzy or”.

Knowing about causality connection of two facts, for example “If *R*, then *G*”, that use the fuzzy sets of *R Ì U, G Ì V,* it is possible to execute the unclear inferencing of $R \to G$, $R' \to G'$, which means that if the fact of *G* is derived from the fact of *R*, then from the fact R' the fact G' is derived, where *R, G,* R', G' - are fuzzy sets.

To execute an fuzzy logical output operation, it is necessary to know the fuzzy relations between the plurals of *R Ì U* and *G Ì V*, which are set on universal sets: $W = \{w_1, w_2, ..., w_l\}$, and $V = \{v_1, v_2, ..., v_m\}$, that are covered by the matrix:

$$Y = R \times G = \sum_{i=1}^{l}\sum_{j=1}^{m}\{m_R(w_i) \cap m_G(v_i)\}.$$

At matrix with l×m size the element standing on crossing of the *i* line and *j* column is determined in this way:

$$m_y(w_i, v_j) = m_R(w_i) \cap m_G(v_j)\,.$$

For calculation of the result of logical (G’) a formula is used:

$$G' = R' o\ Y = R'\ o\ (R \times G)\,,$$

where o - is an operation of min-max composition.

Putting the formula of this operation, we get:

$$G' = \sum_{j=1}^{m} \cup_{wi \subset W}\{m_{R'}(w_i) \cap m_Y(w_i, v_i)\}.$$

We can base upon the experience of applying mathematical methods of fuzzy-logic in the tasks of medical diagnostic dependences for the complex processes of pattern recognition and for prediction processes. It is possible to apply developed scientific principles of fuzzy logic modelling:

1. Principle of linguistic variables of the system. In accordance with this principle entrances and initial variables of the model to be developed will be examined as linguistic variables with qualitative terms (with the values which adopt variables). An example of the linguistic terms is the temperature {very low, low, middle, high, very high}.

Thus, examined variables can have the concrete numerical values. Concrete evaluation tasks receive exact linguistic terms. Such terms are more natural for the specialists - experts in this knowledge field.

2. Principle of linguistics knowledge about acceptance of concrete decisions. In accordance with this principle connections between output and input parameters of the system are described in natural

language, and then they are formalized as an aggregate of unclear logical utterances of “If-Then, Else”.

The aggregate of such utterances can be examined as a set of specifications of input and output parameters. This inferencing algorithm gives a possibility to evaluate such values of the input parameters that are absent in the base.

3.Hierarchy principle of knowledge about the decision. The possibility to describe connections between the output and input parameters appears with use of the first two principles. Application of hierarchy principle allows to avoid the difficulties related to the dimension of the system (by amount of the input parameters). In accordance with this principle it is expedient to conduct classification of input parameters and to build a derivation tree.

Due to this principle it is possible to take into account the unlimited amount of input parameters which influence the decision practically. Thus, the problem of design consists of the following stages:

1. Construction of derivation tree.
2. Determination of regions of change of input parameters.
3. Estimation of qualitative (linguistic) parameters.
4. Determination of types of decisions (in case of few decisions) and regions of their change.
5. Creation of knowledge base.
6. Formalization of knowledge base as fuzzy logic utterances.
7. Receipt of the system of unclear logical equations.
8. Development of models of functions of belonging, which provide presentation of quantitative and qualitative parameters as fuzzy plurals for the different number of linguistic terms which are entered into the knowledge base.

Then, decision-making process on the basis of the obtained model is presented in Figure 1.11.

The general method of designing describes the main stages of the process; however, it doesn’t take into account some features of the technological processes with unclear entrances and initial information. From the conducted analysis of such processes the following differences become obvious: influence of management algorithm, simultaneous presence of the variety of outputs which are impossible to divide.

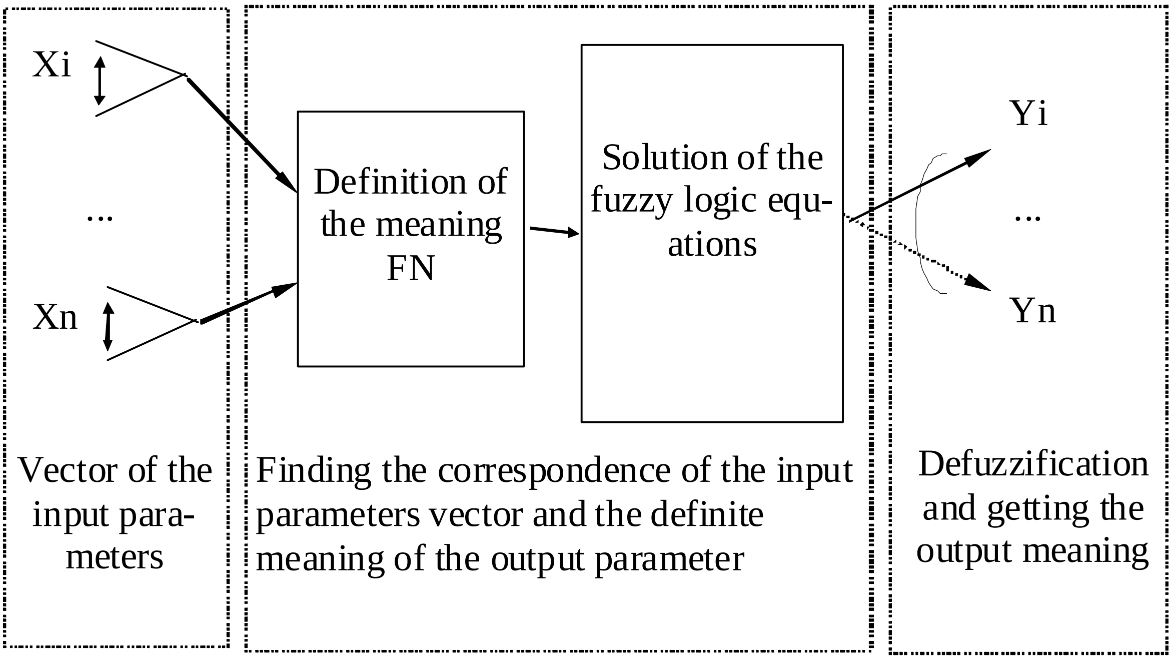

Xi
...
Xn
Definition of the meaning FN
Solution of the fuzzy logic equations
Yi
...
Yn
Vector of the input parameters
Finding the correspondence of the input parameters vector and the definite meaning of the output parameter
Defuzzification and getting the output meaning

Figure 1.11

1.4 Errors

Replacement of the original by a model is always related to some certain simplification and exclusion of the unimportant (in accordance with the accepted criteria) properties, parameters, factors. In mathematical description it determines the presence of the irremovable errors determined by methodology of designing that are called the errors of modelling method. For example, in statistical modelling these errors are related to digitizing the continuous casual sizes, limited sample size, pseudo-random character of the generated numerical sequences. Irremovable errors are always in the basic data obtained experimentally. Thus, tasks and algorithms sensitive to the change of basic data claim special attention, as there can be considerable growth of the number of errors. The computer calculations within the mathematical models require realization of an algorithm as a sequence of logic and arithmetic operations and there is presence of errors of numerical method of the problem’s decision. These errors can be divided into the following kinds:

Transaction error (error of digitization) because of the limited number of digits in computer presentation of numbers;

Error of limitation (truncating), related to the numerical method of decision, when in order to describe the function closely in place of the infinite rows only the first members are used (for example, difference description of the derivative).

Error of distribution, subsequent upon accumulation of errors appearing in the previous stages of calculation.

The indicated errors cause two types of errors: local ones are sums of the errors during the process at every step of calculations; global ones are sums of errors accumulating from the moment of start of calculations.

In the method of presentation there are distinguished: an absolute error Δ determined as a module of difference between true A and calculated a values of size,

$$\Delta = |A - a|,$$

and a relative error

$$d_a = \frac{\Delta}{|a|}.$$

Exactness of calculations is determined by the amount of numbers of the results to be trusted. The number is named “faithful” if an absolute error does not exceed half of unit of digit address which this number is in. It is obvious that all numbers preceding faithful are correct.

We will transfer the basic rules of transformation of errors in the process of calculations:

1) The absolute error of sum of eventual number of approximate numbers does not exceed the sums of absolute errors of these numbers

$$\Delta\left[\sum_{i=1}^{k} a_i\right] \le \sum_{i=1}^{k} \Delta_i \text{ .}$$

2) The relative error of sum of eventual number of approximate numbers does not exceed the maximum error of one of the elements

$$d\left[\sum_{i=1}^{k} a_i\right] \le \max_{1\le i\le k} d_i \text{ .}$$

3) the relative error of multiplication at small enough errors $(d_a \le 0,1)$ does not exceed the sum of relative errors of multiplicands

$$d\left[\prod_{i=1}^{k} a_i\right] \le \sum_{i=1}^{k} d_i \text{ ,}$$

where $\prod_{i=1}^{k} a_i = a_1 * a_2 * ... * a_k$.

During application and algorithmization it is necessary to take into account the notions of convergence and stability connected with error evaluation.

Thus, increase of exactness is achieved by change of internal parameters of algorithm (for example, by the maximally possible difference between the previous and the next approaches).

Stability of the computational algorithm is a continuous dependence on the decision on the input data.

Correctness of the method of calculation depends upon the property of indisputable existence of the problem's decision and the firmness of computational algorithm that is applied within the method's realization.

Convergence is a feature of algorithm to make the calculations with the least possible number of errors for the set class of data by way of change of its parameters. Stability of algorithm is an ability to make calculations and to get the final result with the set exactness during change of the algorithm's parameters and input data within certain margins that are called the region of stability.

In a number of cases (for example, at the design of the measuring systems and devices in conditions when unstructured data is entered) probabilistic approach to estimate the errors is used.

Error is one of basic descriptions of quality of calculable process and its estimation must accompany decision of any engineering and scientific tasks via computer methods.

1.5 Remarks

The mathematical modelling is a part of process of creation of software and hardware of automation and information-measuring technique. Calculation methods of programming on computer became the basic practical instrument of developers of automatic measuring information devices and systems. Large experience of use of calculation methods, application of numerical procedures, creation of the special software for decision of various tasks in this region allows to point out their basic types:

1. Identification of dynamic descriptions of linear links at the use of different descriptions of signals on their entrances and outputs.

2. Use of the least-squares method for identification of transmission description on data describing transitional and frequency descriptions or signals on the entrances and outputs of link arrays.

3. Research of stability of the linear dynamic systems on basis of use of different criteria. Construction of region of stability on the plane of parameters of the system.

4. Analysis of quality of the linear automatic control systems. Determination of optimum controls by way of decision of algebraic equation of

Riccati (continuous and discrete cases), which is the problem of an optimum linear controller. The decision of equation of Riccati is related to implementation of row of transformations and decision of special problems (making an initial matrix, transformations of similarity) allowing to bring matrices over to the Hessenberg and Schur type; finding own values of matrices.

5. Research of the nonlinear automatic systems on basis of close methods of decision of nonlinear differential equations. Application of methods of the harmonic linearizing and piece-linear approximation.

6. Simulation of measuring devices and systems on computer, including: generation of pseudo-random numerical sequences imitating measurands and influences; design of transformation of information parameters of signals in the explored devices; treatment of outputs of pseudo-random numerical sequences (construction of histograms, cross-correlation functions, estimation of criteria).

7. Analytical probabilistic modelling of measuring information devices and systems on basis of associate probabilistic (laws of probabilities distribution) and spectral models (spectral density of power).

8. Decision of determination problem of values distribution of measurand (flowrates, sound-wave, temperatures) in the closed region.

9. Research of automation devices by method of experiment planning.

10. Digital signal processing problems. Analysis of spectrums of different signals with the use of Fourier transformation, for example.

11. Analysis and errors estimate of measuring devices and systems on basis of methods of private derivatives, probabilistic design, interval analysis etc.

12. Decision of problems of computer-aided design of the automatic control systems. In this direction a large experience has been accumulated in creation of various software systems.

New approach to the design of models in conditions of uncertainty is interval modelling. This method is more simple compared to the stochastic methods and demands knowledge only about numerical intervals of data. The basics of interval method will be described in chapter 7 of this book.

A deeper study of the modelling theory can be found in special researches on probabilistic and statistic methods, methods based upon the fuzzy logic theory that are often used in indeterminate conditions, queuing systems simulations, models of structural transformations. Comparing the approaches it’s difficult to pick out the best possible one. Practical demands and experience require choosing the most expedient of them that could be used at the different stages of the systems modelling in automatics, information-measuring techniques and control systems development.

Exercises

1. Make examples of typical tasks of computational mathematics. Classify them according to the type of mathematical methods and physical essence. Give examples of application of these methods to solve the applied problems of automatics and control systems.
2. How do the errors of calculations emerge? Classify them.
3. What is the difference between the local and the global errors?
4. Prove all the features of arithmetic operations with errors from section1.3.
5. Give the definition of iterative methods.
6. What is the convergence of an iterative algorithm?
7. What is the firmness of an iterative algorithm?
8. What is the correctness of a computational method?
9. Name methods of generation of the random (pseudorandom) numbers.

Chapter 2. Problems of Linear Algebra

2.1 Introduction

Solution of the systems of linear equations is a widespread calculation problem of linear mathematics. Methods of solution of such problems are considered in this chapter. We suppose the readers are already acquainted with the information given below from the theory of matrices.

2.2 Systems of Linear Equations

Generally, the problem could be defined as following: to find the values $x_1, x_2, ..., x_n$ which satisfy the system of linear equations

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = c_1, \\ &a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = c_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = c_n, \end{aligned} \tag{2.1}$$

or in a matrix form $AX = C$, where

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix},$$

$$\mathrm{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}.$$

The determinant’s inequality to zero (linear independence of equations) is the necessary and sufficient condition of the decision’s existence:

$$det\, A \neq 0.$$

The methods of decision of the systems of linear equations can be divided into direct and iterative. To the lines which allow to get the exact decision, the methods of determinants of Cramer, Gauss and the special direct method for tridiagonal system could be applied. Iterative methods are based on the accepted clarification of progressive approximations to the exact decision, effective in the case, when a lot of coefficients are either equal to zero or have a higher order in the system.

2.2.1 Classic Methods

The well-known method of Cramer (determinants) in detail is considered in the standard courses of higher mathematics and can not be applied in most practical problems due to the essential complexity of the determinants calculation, providing even the tiny growth of the system’s order. That is why in this section we will consider the Gauss method, which, even if it yields to the iterative methods in certain practical problems, however is more universal, and also a special direct method, that is used in problems with tridiagonal matrices.

2.2.1.1 Gauss Method

Gauss Method (method of exception) is based on reduction of matrix of coefficients of the system (2.1) to the three-cornered form:

$$\begin{bmatrix} * & * & * & \mathbf{L} & * & * \\ 0 & * & * & \mathbf{L} & * & * \\ 0 & 0 & * & \mathbf{L} & * & * \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathbf{L} & \mathrm{M} & \mathrm{M} \\ 0 & 0 & 0 & \mathbf{L} & * & * \\ 0 & 0 & 0 & \mathbf{L} & 0 & * \end{bmatrix}$$

and consists of two stages: direct motion and reverse putting. The stage of direct motion finishes, when one of equations of the system becomes equation with one unknown. Then, carrying out the reverse putting, all the unknowns are found. This method could be easily realized on computer.

At first by division of the coefficients $a_{1,1}$ the first equation is rationed; then we multiply the equation obtained on coefficients $a_{1,i}$ and subtract from all the equations. Thus, x_1 is eliminated from all the equations, except the first. At the next stage similar procedure is used to the last $(n-1)$ equations and is repeated until the system is transformed to the three-cornered form.

On a k-step the coefficients of k-equation are rationed, and new coefficients in the next equations are concerned as

$$b_{ij} = a_{ij} - a_{ik}b_{kj,}\ i > k\,.$$

Coefficients a_{ij} change at every step.

Number of arithmetic operations in use of the method is

$$N \approx \frac{2}{3}n^3.$$

The algorithm of the method is resulted in Figure 2.1.

2.2.1.2 Gauss-Jordan Method (Exception)

This method allows to bring a matrix of coefficients to the diagonal form. In distinction to the previous method instead of $i > k$ we use $i \neq k$. In the Gauss method transformation is applied to the equations which stand only below the leading row. Equations which stand either below the leading row or above it are transformed with use of the Gauss-Jordan method.

This method facilitates the decision-making, but is accompanied with increase of calculations volume.

2.2.1.3 Modified Gauss Method

In many cases there is a necessity to solve the systems of linear equations with the varieties of matrix coefficients and a permanent column of free members. Most frequently Gauss modified method is used to solve such problems. In this method matrix equation (2.1) is transformed to the matrix of coefficients A as a multiplication of left L and right R three-cornered matrices

$$L * R = A.$$

As the diagonal elements of one of the matrices become equal to one, they can't be memorized, and it is necessary to keep both matrices in the computer memory in place of matrix of coefficients A.

In the variant of the Kraut method the following sequence of finding the elements of matrices L and R is used

$$\text{for } k = 1, 2, ..., n;$$

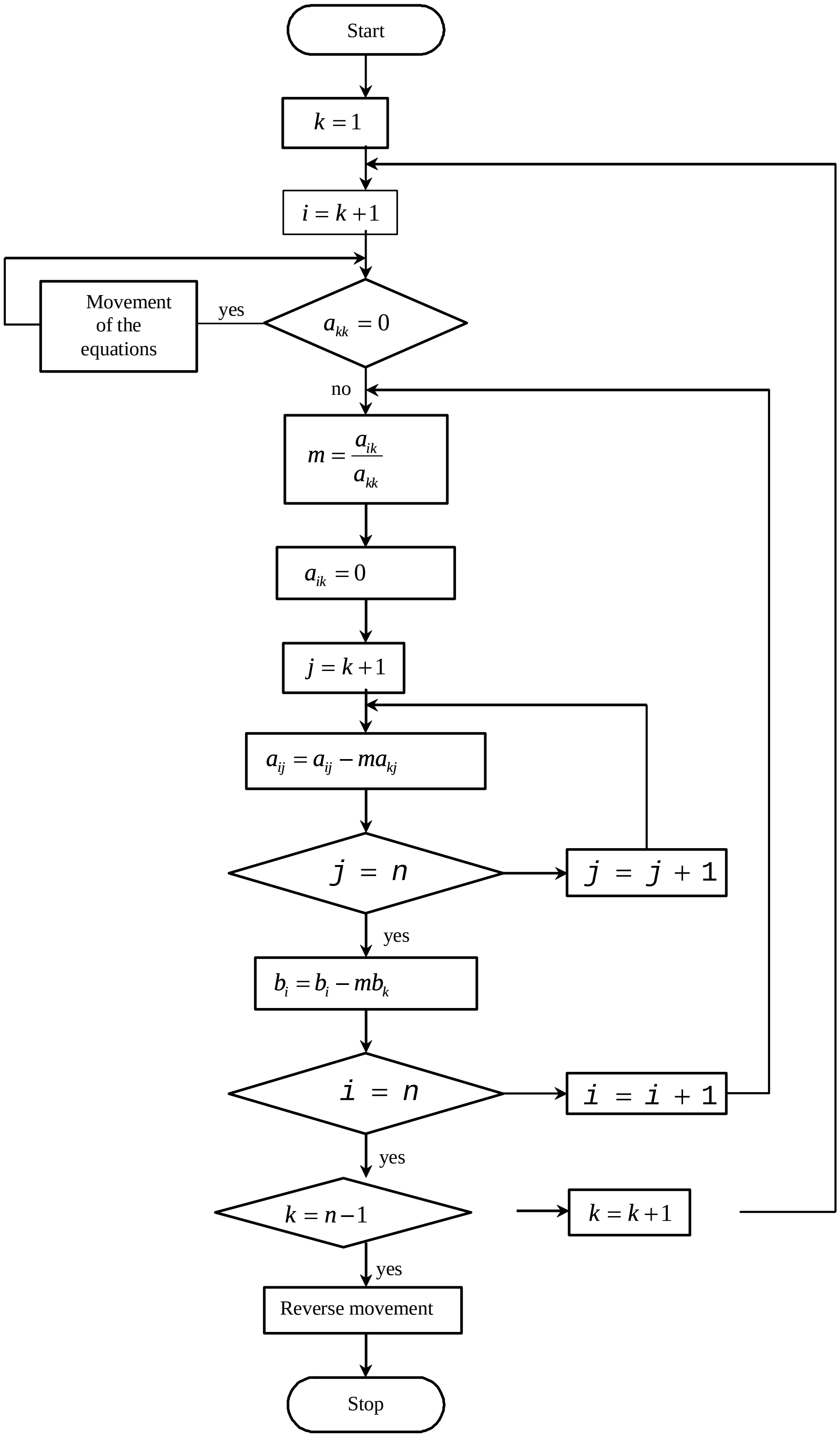

Start
$k=1$
$i=k+1$
Movement of the equations
yes
$a_{kk}=0$
no
$m=\frac{a_{ik}}{a_{kk}}$
$a_{ik}=0$
$j=k+1$
$a_{ij}=a_{ij}-ma_{kj}$
$j=n$
$j=j+1$
yes
$b_i=b_i-mb_k$
$i=n$
$i=i+1$
yes
$k=n-1$
$k=k+1$
yes
Reverse movement
Stop

Figure 2.1.

$$l_{ik} = a_{ik} - \sum_{p=1}^{k-1} l_{ip} r_{pk} \,, \qquad i = k, k+1, \ldots, n;$$

$$l_{kk} = \frac{1}{l_{kk}} \,,$$

$$r_{kj} = l_{kk}\left(a_{kj} - \sum_{p=1}^{k-1} l_{kp} r_{pj} \right), \quad j = k+1, \ldots, n;$$

$$r_{kk} = 1.$$

The system is transformed to such system, the solution of which is replaced by the solution of two systems with three-cornered matrices:

$$LY = C,$$
$$RX = Y.$$

Elements Y, X could be found from the following correlations:

$$y_1 = l_{11} c_1;$$

$$y_i = l_{ii}\left(c_i - \sum_{p=1}^{i-1} l_{ip} y_p \right) i = 2, \ldots, n;$$

$$x_i = y_i - \sum_{p=i+1}^{n} r_{ip} x_p \,, \; i = n-1, \ldots, 1.$$

Number of arithmetic operations used in this method to solve the system of linear algebraic equations is $N = 2n^2$.

2.2.1.4 Direct Method for Tridiagonal Systems

This method (named often as the method of pass) is used to solve the systems of equations with the band matrix of coefficients. We will consider its application to solve the tridiagonal system which is typical for many practical problems.

Let’s write the system in such a form:

$$\begin{array}{l}
b_0 y_0 + c_0 y_0 = \varphi_0 , \\
a_1 y_0 + b_1 y_1 + c_1 y_2 = \varphi_1 , \\
a_2 y_1 + b_2 y_2 + c_2 y_3 = \varphi_2 , \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_i y_{i-1} + b_i y_i + c_i y_{i+1} = \varphi_i , \\
a_{n-1} y_{n-2} + b_{n-1} y_{n-1} + c_{n-1} y_n = \varphi_{n-1} , \\
a_n y_{n-1} + b_n y_n = \varphi_n .
\end{array}$$

To solve this system we will use the analogue of the direct motion of Gauss method. Then the system is transformed to the following form:

$$\begin{array}{l}
y_0 - u_0 y_1 = v_0 , \\
y_1 - u_1 y_2 = v_1 , \\
\ldots\ldots\ldots\ldots\ldots\ldots \\
y_{i-1} - u_{i-1} y_1 = v_{i-1} , \\
y_i - u_i y_{i+1} = v_i , \\
y_{n-1} - u_{n-1} y_n = v_{n-1} , \\
y_n = v_n ,
\end{array}$$

where $u_0 , v_0 , u_1 , v_1 , ..., u_n , v_n$ are coefficients, that are called pass-coefficients.

Take into account, that

$$u_0 = -\frac{c_0}{b_0} ; \; v_0 = \frac{j_0}{b_0} .$$

These coefficients give possibility to find $y_n , y_{n-1} ,..., y_0$.

Eliminating from previous equations y_{i-1} by arithmetic transformations, we get the formulas for determination of the sought values:

$$u_i = -\frac{c_i}{a_i u_{i-1} + b_i} ,$$

$$v_i = \frac{\varphi_i - a_i v_{i-1}}{a_i u_{i-1} + b_i}$$

and further

$$y_n = v_n,$$
$$y_i = u_i y_{i+1} + v_i, \quad i = n-1, \dots, 1, 0.$$

2.2.2 Iterative Methods

Iterative methods are especially effective for the systems with the big order and with sparse matrices of coefficients. They are used in the systems which preliminary result in the following form:

$$\begin{aligned} x_1 &= b_{1,n}x_n + b_{1,n-1}x_{n-1} + \dots + b_{1,1}x_1 + b_{1,0}, \\ x_2 &= b_{2,n}x_n + b_{2,n-1}x_{n-1} + \dots + b_{2,1}x_1 + b_{2,0}, \\ x_n &= b_{n,n}x_n + b_{n,n-1}x_{n-1} + \dots + b_{n,1}x_1 + b_{n,0}; \end{aligned} \tag{2.2}$$

or in a matrix form:

$$X = BX + B_0,$$

where

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix}, \quad B_0 = \begin{bmatrix} b_{10} \\ b_{20} \\ \vdots \\ b_{n0} \end{bmatrix}.$$

There are a few basic variants of iterative methods. They are: Jacobi methods (simple iteration), Gauss-Seidel and successive overhead relaxation. In the basics of this method there is a systematic clarification of the variable values which are set at the beginning of the calculations.

In the Jacobi method the initial variable values are used for calculation of the new values $x_1, x_2, \dots, x_n$ in accordance with (2.2) equations. The process is finished when all the new values converge to the previous ones. In the opposite case the new values are used in place of the initial ones. This procedure repeats itself until the convergence is attained or it becomes clear that the process diverges. In this method replacement of all variable values is conducted simultaneously (simultaneous displacement).

The system of iteration equations is:

$$
\begin{aligned}
&x_1^{(m+1)} = b_{11}x_1^{(m)} + b_{12}x_2^{(m)} + b_{13}x_3^{(m)} + \ldots + b_{1n}x_n^{(m)} + b_{10},\\
&x_2^{(m+1)} = b_{21}x_1^{(m)} + b_{22}x_2^{(m)} + b_{23}x_3^{(m)} + \ldots + b_{2n}x_n^{(m)} + b_{20},\\
&\vdots\\
&x_n^{(m+1)} = b_{n1}x_1^{(m)} + b_{n2}x_2^{(m)} + b_{n3}x_3^{(m)} + \ldots + b_{nn}x_n^{(m)} + b_{n0};
\end{aligned}
$$

where, accordingly, value $x_i\ (i = 1,...,n)$ on the next iteration is with index (m+1) and the previous iteration is with index (m).

In the Gauss-Seidel method the obtained value x_1 is immediately used for the calculation of x_2. Then by the new values x_1 and x_2 we consider x_3 and so on. It allows to rate up the convergence substantially.

In the method of successive overhead relaxation all the new variable values are calculated as:

$$x_i^{(m+1)} = x_i^{(m)} + \omega(\bar{x}_i^{(m+1)} - x_i^{(m)}),$$

where $\bar{x}_i^{(m+1)}$ is the specified value $x_i^{(m)}$ due to the Gauss-Seidel method**;** ω parameter of relaxation ($1 \le w \le 2$).

At $w = 1$ this method is similar to the Gauss-Seidel method. Rate of convergence depends on ω.

One of the main prerequisites of successful application of iterative methods is convergence. For estimation of convergence the norms of matrix of coefficients $\|B\|$ are calculated from the system (2.2).

The following forms of norms are frequently used:

$$1 \text{ norm:} \max_{1\le j\le n} \sum_{i=1}^{n} |b_{ij}|,$$

$$2 \text{ norm:} \max_{1\le i\le n} \sum_{j=1}^{n} |b_{ij}|,$$

$$E\text{-norm } \textbf{(}\text{Euclid}): \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^2}\ .$$

There are a few approaches to determinate the convergence by estimation of the norms. Generally it is sufficient, that at least one of the matrix norms is less than one.

$$\|B\| < 1\,.$$

In mathematics such a condition is called “ordinary” or “strong”. In many cases convergence is provided via implementation of the so-called “weak” sign. For example, “weak” sign of sums of lines: for all the sums of lines of coefficients $(i = 1,...,n)$ the correlation is executed:

$$\sum_{j=i}^{n} b_{ij} \leq 1,$$

but there is one line p for which

$$\sum_{j=i}^{n} b_{pj} < 1.$$

Similarly, the “weak” sign is determined as the sums of the column’s coefficients.

“Weak” sign can be used in those cases, when matrix of coefficients *A* from the systems of equations (2.1) can be transformed to the form:

$$\begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix},$$

where A_1, A_3 are square matrices.

For such matrices the system of equations (2.1) disintegrates into two systems of equations which are solved consistently. In special textbooks the list of which is given at the end of this book, more detailed analysis of the properties and signs of the convergence estimation is given, but in order to perform a number of practical problems it is sufficient to use the information given above.

2.3 Remarks

The Gauss method and its modifications (Gauss-Jordan, the Crout, matrix inversion etc.) are more universal, but it is difficult to use them when the coefficients matrix is very sparse (due to a multitude of zero elements and errors that occur in the course of the multi-step calculation processes and that should be taken into account). At the same time the Cramer methods are effective only for systems with comparatively little order (less than 10-15).

The iterative methods are very simple and convenient, but only for the convergence problem solving. In case of a wide range of problems (for example, in mathematical physics) it could be reached in accordance with the correct problems formalization.

In this chapter from a variety of problems in the sphere of linear algebra only systems of linear equations were considered. The other problems (transformations of complicated matrices, problems with own meanings of the matrices) can be found in special books on the theory of matrices.

Exercises

1. Compare direct and indirect (iterative) methods.
2. Construct algorithms of Gauss and Gauss-Jordan methods.
3. What is the form of linear system that is used in the iterative methods?
4. What are the rules of checking the convergence of the iterative methods?

5. What is the difference between iterative methods of Jacobi and Gauss-Seidel? How does the difference show in the algorithms?

6. Solve on computer the following system using Kramer, Gauss and Gauss-Jordan methods

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 15; \\ x_1 - x_3 + 7x_4 = 26; \\ x_1 - 2x_2 + 3x_3 - 4x_4 = -10; \\ x_1 - x_2 + 2x_3 + 3x_5 = 20; \\ x_1 + x_3 - x_4 + 10x_5 = 50 \end{cases}$$

7. Estimate the convergence of iterative algorithm for the following system. Using the algorithm solve the system on computer

7.1

$$A = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 2 \\ -1 & -1 & -2 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 12 \\ 30 \\ 14 \\ 3 \end{bmatrix};$$

7.2

$$A = \begin{bmatrix} 2 & 0 & 7 & 1 \\ 1 & 2 & -1 & -1 \\ -1 & -2 & 1 & 2 \\ 0 & 2 & 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 11 \\ 2 \\ 0 \\ 6 \end{bmatrix};$$

7.3

$$A = \begin{bmatrix} 1 & 3 & 4 & 5 & 1 \\ 1 & 1 & 2 & -1 & 3 \\ -3 & -2 & 1 & 2 & 0 \\ 1 & 1 & 1 & 2 & 1 \\ -1 & 1 & 2 & 3 & 15 \end{bmatrix}, \qquad C = \begin{bmatrix} 38 \\ 5 \\ 4 \\ 14 \\ 19 \end{bmatrix}.$$

8. Solve each of the following tridiagonal systems via the direct method:

$$a)\ \begin{aligned} -2x_1 + x_2 &= -2 \\ x_1 - 2x_2 + x_3 &= 1 \\ x_2 - 2x_3 + x_4 &= 0 \\ x_3 - 2x_4 &= 0 \end{aligned} \qquad b)\ \begin{aligned} 5x_1 - 2x_2 &= 5 \\ -2x_1 + 5x_2 - 2x_3 &= -2 \\ -2x_2 + 5x_3 - 2x_4 &= 0 \\ -2x_3 + 2x_4 - 2x_5 &= 2 \\ -2x_4 + 5x_5 &= -5 \end{aligned} \qquad c)\ \begin{aligned} -2x_1 + x_2 &= 1 \\ x_1 - 4x_2 + x_3 &= 0 \\ x_2 - 3x_3 + x_4 &= -1 \\ x_3 - 2x_4 &= 0 \end{aligned}$$

Which of these systems can be solved using iterative methods?

Chapter 3. Equations and Systems

3.1 Introduction

In this section the methods of solution of nonlinear equations and systems of equations are considered. Many practical tasks, for example, calculations of nonlinear electric circles and systems management, decision of nonlinear differential equations, analysis of the systems firmness via estimation of their own values and so on lead to calculation of tasks of such kind.

For the simplest types of algebraic equations (not higher than the third degree) there are exact analytical formulas, for transcendent equations and any systems of equations such methods in general do not exist and due to this we should use only approximate iterative methods and algorithms. Main iterative methods and algorithms for solution of such tasks are considered below.

3.2 Nonlinear Equations

Equations in which the degrees of argument are entered only with the proper coefficients are named algebraic.

Nonlinear equations which contain trigonometric or other special functions are named transcendent.

General form of an algebraic equation:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0 = 0 . \qquad (3.1)$$

It is possible to select some important properties of algebraic equations which simplify further determination of the roots. Here and further we call some properties as theorems, as it is accepted in mathematics, but give them without proof.

1. Basic algebraic theorem. Algebraic equation of order n has n roots, which can be real or complex.

Every root is calculated the proper number of times that equals to its multiplicity. The multiplicity of root x_0 equals to k, if

$$f^{'}(x_0) = f^{''}(x_0) = ... = f^{(k-1)}(x_0) = 0 .$$

2. If all coefficients a_i of equation (3.1) are real, all complex roots form complex-conjugating pair.

3. Rule of Descartes. The number of positive real roots equals or is less than the number of changes of signs in the sequence of coefficients (that

assertion concerns just the number of negative real roots while replacing x to -x) in (3.1).

4. Theorem of Lagrange. The high bound of positive real roots could be written as

$$R = 1 + \sqrt[k]{\frac{B}{a_0}}, \ a_0 > 0,$$

where k is the number of the first negative coefficient; B is the most absolute value of the negative coefficient.

5. Theorem of Gua. If equation (3.1) has real roots and real coefficients

$$a_k^2 > a_{k-1}a_{k+1}.$$

We keep in mind that direct analytical methods exist only for algebraic equations not higher than of the third order, but for transcendent equations direct methods do not exist in general. While determining actual roots via the numeral methods two theorems should be used. The first allows to separate roots and to set as close intervals $[l, b]$ as possible, in which roots of the equation exist, and the second one is used to estimate the approach.

Theorem 1. If a continuous function $f(x)$ takes on the value of different signs at the ends of the segment $[l, b]$, where $f(l)f(b) < 0$, in the middle of this segment there is at least one root of equation $f(x) = 0$, which is $x \in (l, b)$ and in it $f(x) = 0$.

Theorem 2. Let's assume ξ is exact, and $\bar{x}$ is the root of the equation $f(x) = 0$, which is on the same segment $[\alpha, \beta]$, thus $|f'(x)| \geq m, \ a \leq x \leq b$.
Then

$$|\bar{x} - x| \leq \frac{|f(\bar{x})|}{m}.$$

There are several methods of solution of nonlinear equations, expedience of application of each of which depends on the type of equation, exactness needed etc

One should also keep in mind that determining the roots to reduce the degree of initial nonlinear equation dividing on $(x - x_i)$ (where x_i is the root that is found) should be executed very carefully; that is related to the accumulation of errors of distribution, which will be contained in the coefficients of the new equation.

3.2.1 Method of Half-Note Division (Bisection)

In this method at first the value of function is calculated in points which are located through even intervals on the axis x. When $f(x_n)$ and $f(x_{n+1})$ have opposite signs, find $x_m = \frac{x_{n+1} + x_n}{2}$, $f(x_m)$. If the sign $f(x_m)$ coincides with the sign $f(x_n)$, x_m is used at the next step in place of x_n. If $f(x_m)$ has the sign opposite to the sign $f(x_n)$, x_m is replaced by x_{n+1}. We will mark that for all the methods for the condition of finishing the iterative process it is expedient to take $|x_{n+1} - x_n| \le e$, where e is the set of errors of the root's finding.

The graphic image of the method is given in Figure 3.1, and structure of the algorithm - in Figure 3.2.

The error of solution Δ via n of iterations is in scopes

$$\Delta \le \frac{1}{2^n}|x_1 - x_0|.$$

The method has a small rate of convergence as interval, where the root diminishes no more than twice with every step.

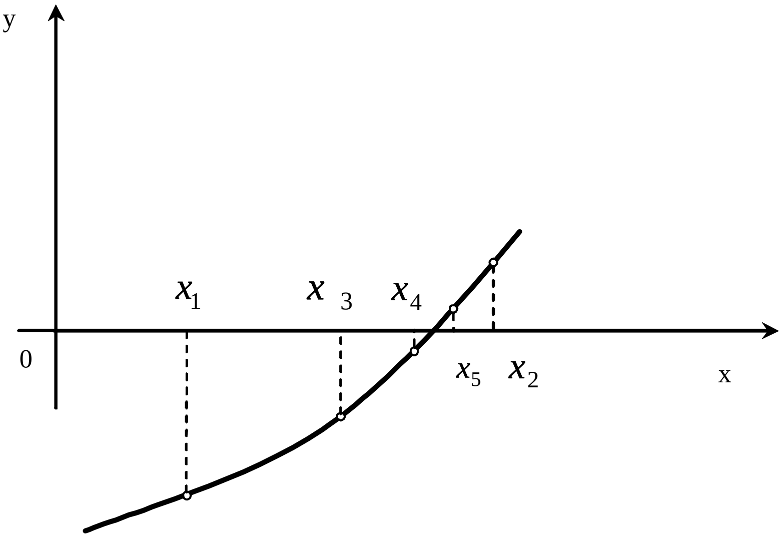
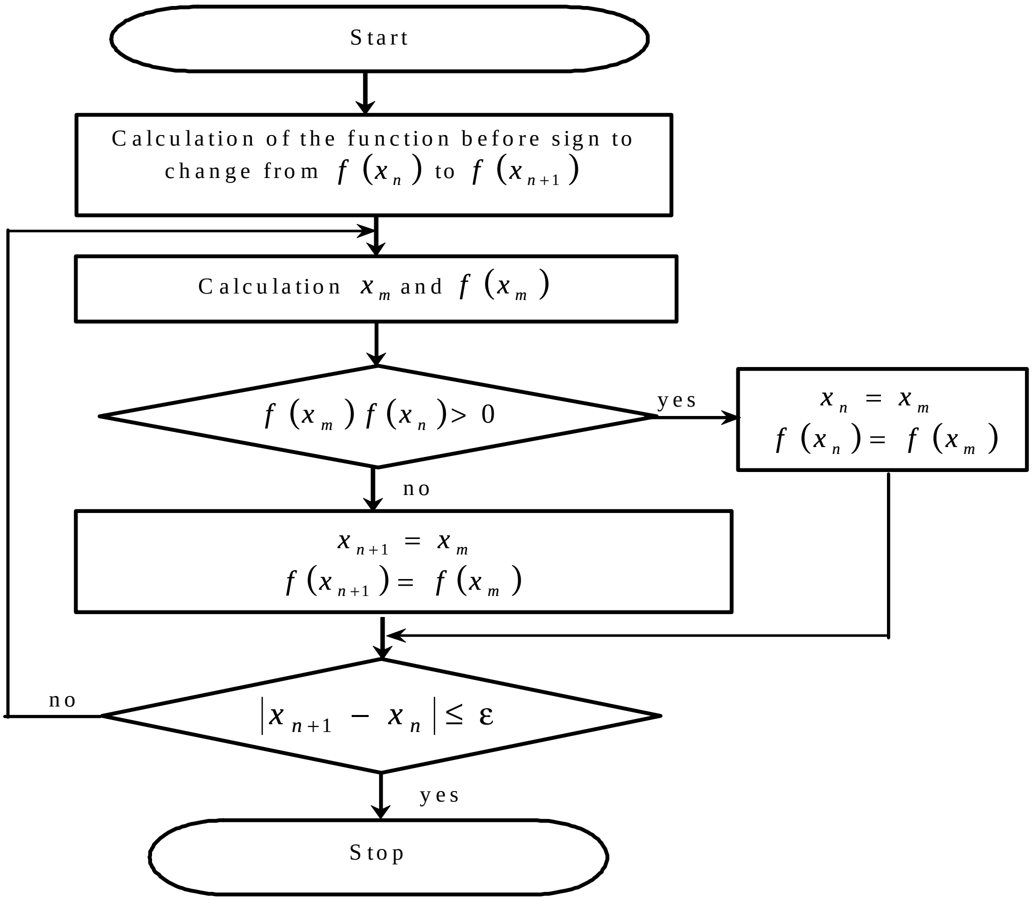


Figure 3.1

3.2.2 Method of Vicious Position (Chords)

This method is based upon linear interpolation for two meanings of function $x_{n-1,}\ x_n$ with the opposite signs. The line through these two points crosses an axis at the point

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



Start
Calculation of the function before sign to change from $f(x_n)$ to $f(x_{n+1})$
Calculation x_m and $f(x_m)$
$f(x_m)f(x_n) > 0$
yes
$x_n = x_m$
$f(x_n) = f(x_m)$
no
$x_{n+1} = x_m$
$f(x_{n+1}) = f(x_m)$
no
$|x_{n+1} - x_n| \le \varepsilon$
yes
Stop

Figure 3.2

Determine $f(x_{n+1})$ and compare its sign with the sign of $f(x_n)$. At the next step use x_{n+1} in place of the value (x_{n-1}, x_n) with which the sign coincides. If $|x_{n+1} - x_n| \le e$, the whole procedure is repeated again (Figure 3.3).

Algorithm of the method of chords is similar to the previous one, except for the procedure of estimation x_{n+1}. One should keep in mind that in this algorithm the control of error is conducted with the moving end of an interval. In the case given in Figure 3.3. progressive approximations are analysed: at the first step $|x_1 - x_2| \le e$, at the second – $|x_1 - x_3| \le e$, at the third – $|x_3 - x_4| \le e$ and etc.

The error of decision is estimated via the formula:

$$\Delta \le \frac{M_1 - m_1}{M_1}|x_{n+1} - x_n|,$$

where M_1, m_1 are accordingly, the largest and the smallest values of the module of the first derivative on the interval x_n, x_{n+1} .

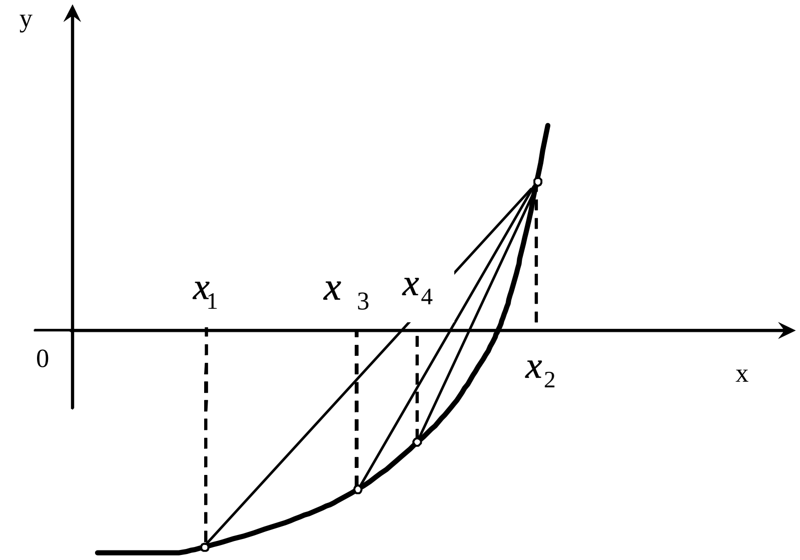


Figure 3.3

3.2.3 Newton Method (Tangents)

In the Newton method first of all the extrapolation is carried out by tangent to the curve (Figure 3.4)

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

On basis of this method decomposition of function by the Taylor row is executed:

$$f(x_n + h) = f(x_n) + hf'(x_n) + \tfrac{h^2}{2} f''(x_n) + ...,$$

Members which contain h in the second degree and higher are rejected. Taking into account that $h = x_{n+1} - x_n$, we can get the previous formula.

Convergence rate of this algorithm depends on the faithful choice of the initial point. When in the process of calculations the angle of slope of the tangent $f'(x)$ is close to zero, it becomes more difficult to use the method. We could assume that in case of very big values of $f''(x)$ (bulge of the function) or when there are multiple roots the Newton method becomes ineffective.

Therefore one should choose the initial approach from the following condition

$$\left(f'(x_0)\right)^2 > f''(x_0)f(x_0) > 0.$$

The error of the method is estimated as:

$$\Delta \le \frac{M_2}{2m_1}\left(x_{n+1} - x_n\right)^2,$$

where M_2 – the largest value of the function's module on the interval $[x_n,\ x_{n+1}]$.

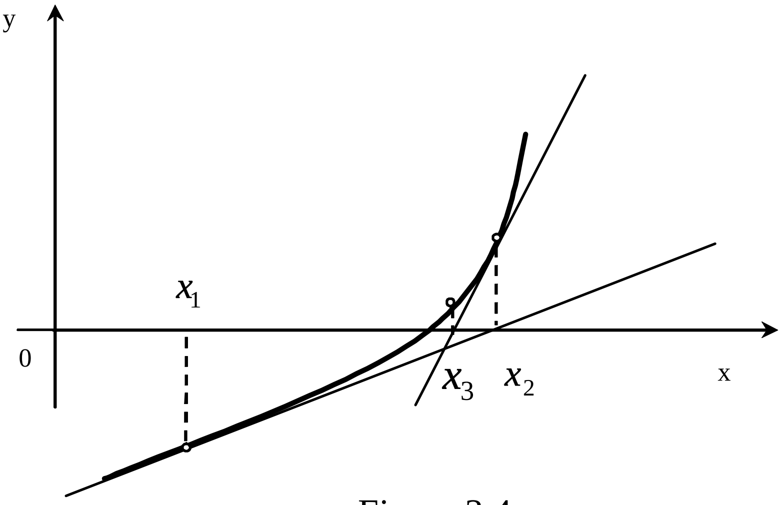


Figure 3.4

3.2.4 Method of Secants

One of the main problems while applying the Newton method is the necessity of the derivative's analytical description. If this difficultly emerges, it is possible to apply its close estimation (figure 3.5) Then, in place of the tangent method the method of secants is used

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

where $F'(x_n)$ is a close estimation of the derivative examined as both secants but not as tangent, and can be calculated using the following formula:

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

or

$$F'(x_n) = \frac{f(x_n + h) - f(x_n)}{h},$$

where h is a certain small step.

Algorithm of this method is similar to the Newton method but with another iterative formula.

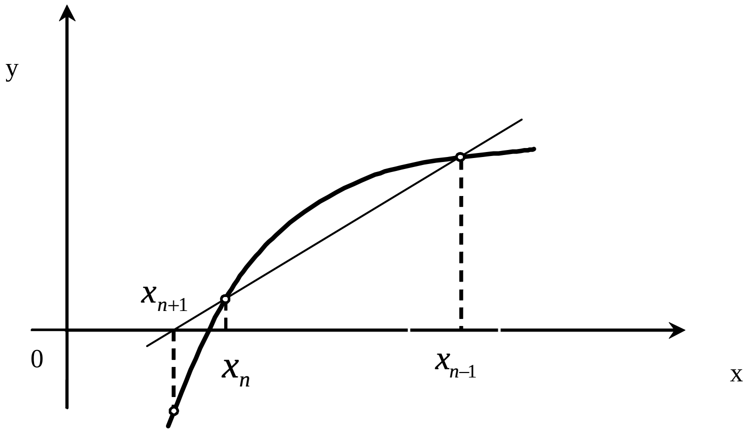


Figure 3.5

3.2.5 Method of Simple Iteration

To use this method the equation $f(x)$ = 0 should be transformed to the following form:

$$x = g(x).$$

The proper iterative formula looks like

$$x_{n+1} = g(x_n).$$

The calculations are finished when

$$|x_{n+1} - x_n| \le e.$$

To provide the convergence, the value of q (the module of the first derivate of function $g(x)$ on the segment $[x_0, x_1]$) should be less than one

$$q < 1.$$

Then there would be a convergence regardless of the choice of initial point on the interval $x \in [x_0, x_1]$.

Error of the method after n iterations is

$$\Delta \le \frac{q^n}{1-q}|x_1 - x_0|.$$

3.2.6 Problem of Complex Roots

To determine complex roots one could apply the methods similar to those used to find the real roots, but joined with a complex number (the control of convergence and errors is conducted via the module’s value). However, this method is not convenient.

There are several special methods which allow to estimate the complex roots, but via calculation by real numbers. The majority of these methods are based on transformation of the initial algebraic equation (3.1) to a variety of quadratic members

$$x^2 + px + q,$$

where p,q are the coefficients.

To perform such transformation the equation should be presented in the following form:

$$\left(x^2 + px + q\right)\cdot\left(x^{n-2} + b_{n-1}x^{n-3} + \ldots + b_3 x + b_2\right) + b_1 x + b_0 = 0, \qquad (3.2)$$

where $b_1 x + b_0$ is a linear remaining member which aims to zero, and the initial equation (3.1) is divided by a quadratic factor $x^2 + px + q$ without remainder .

In order to find coefficients $b_{n-1,} b_{n-2,} \ldots, b_{3,} b_2$ suppose $b_1 = b_0 = 0$. Then we could consider the system of equations which could be obtained from the equivalence of equations (3.1) and (3.2):

It can be performed via a special direct method for the tridiagonal systems or via an iterative method, which is presented in Figure 3.6.

$$
\begin{aligned}
&b_{n-1} = a_{n-1} - p,\\
&b_{n-2} = a_{n-2} - pb_{n-1} - q,\\
&\vdots\\
&b_{n-j} = a_{n-j} - pb_{n+1-j} - qb_{n+2-j},\\
&b_3 = a_3 - pb_4 - qb_5,\\
&b_2 = a_2 - pb_3 - qp_4,\\
&0 = a_1 - pb_2 - qb_3,\\
&0 = a_0 - qb_2.
\end{aligned}
\tag{3.3}
$$

3.3 Nonlinear Systems

Generally, the system of n nonlinear equations with n unknown is the following:

$$
\begin{aligned}
&f_1(x_1, x_2, \dots, x_n) = 0,\\
&f_2(x_1, x_2, \dots, x_n) = 0,\\
&\vdots\\
&f_n(x_1, x_2, \dots, x_n) = 0.
\end{aligned}
\tag{3.4}
$$

As nonlinear functions appear in the system, it becomes impossible to present it in a general form; neither any analytical direct method could be offered to solve such a system. The method of simple iteration based on adding the system (3.4) to the system of nonlinear equations is quite simple:

$$
\begin{aligned}
&x_1 = g_1(x_{1,} x_{2,} \dots, x_n),\\
&x_2 = g_2(x_1, x_2, \dots, x_n),\\
&\dots\dots\dots\dots\dots\dots\\
&x_n = g_n(x_1, x_2, \dots, x_n).
\end{aligned}
\quad .
$$

In a matrix form

$$X = G(X),$$

where

$$
G(x) = \begin{bmatrix} g_1(x_1, x_2, \dots, x_n),\\ g_2(x_1, x_2, \dots, x_n),\\ \dots\dots\dots\dots\dots\\ g_n(x_1, x_2, \dots, x_n) \end{bmatrix}.
$$

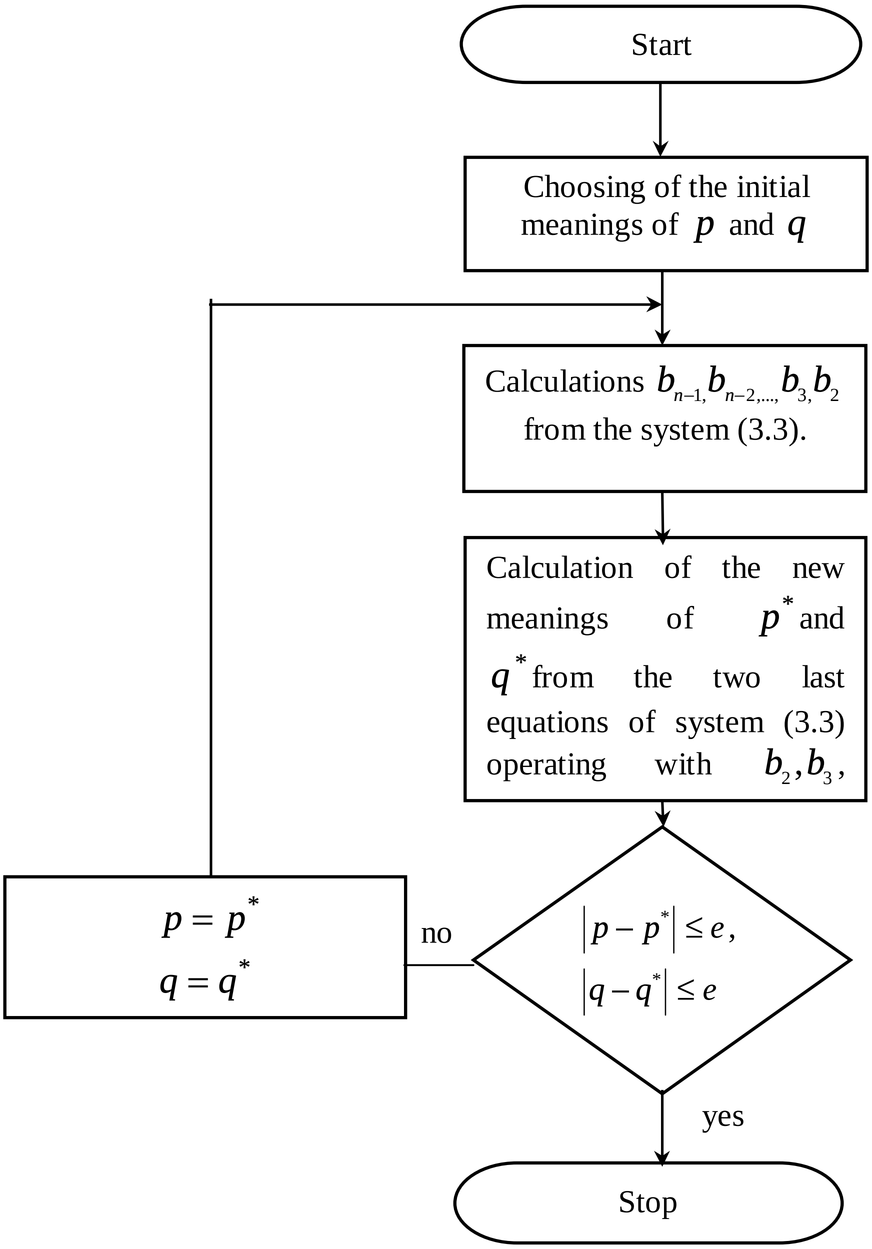

Start
Choosing of the initial meanings of p and q
Calculations $b_{n-1,}b_{n-2,...,}b_{3,}b_2$ from the system (3.3).
Calculation of the new meanings of p^* and q^* from the two last equations of system (3.3) operating with b_2, b_3,
$|p-p^*| \le e$, $|q-q^*| \le e$
no
$p = p^*$
$q = q^*$
yes
Stop

Figure 3.6.

Then there can be an applied algorithm similar to the Gauss-Seidel method for the systems of linear equations. On its basis iterative equations are used which link $(m+1)$ and m iterations.

$$
\begin{aligned}
&x_1^{(m+1)} = g_1\left(x_1^{(m)}, x_2^{(m)}, \ldots, x_n^{(m)}\right),\\
&x_2^{(m+1)} = g_2\left(x_1^{(m+1)}, x_2^{(m)}, \ldots, x_n^{(m)}\right),\\
&\vdots\\
&x_n^{(m+1)} = g_n\left(x_1^{(m+1)}, x_2^{(m+1)}, \ldots, x_n^{(m)}\right).
\end{aligned}
$$

Due to the fact that it is quite a difficult task to provide convergence using this method, and providing that the convergence interval could be extremely narrow, the choice of the initial approach is very complicated.

Generally, this method will converge, if $\left\|G'(x)\right\| < 1$, where $\left\|G'(x)\right\|$ is a norm of matrix of partial derivates that functions on variables $x_1, x_2, \ldots, x_{n.}$

$$G'(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \ldots & \frac{\partial g_1}{\partial x_n} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \ldots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}.$$

To solve the systems of nonlinear equations more frequently the Newton method is used which proved to be more reliable. It is used in the form of the Newton’s method analogue for one equation and is based upon the decomposition of all n equations in the Taylor row:

$$f_1(x_1 + \Delta x_1, \ldots, x_n + \Delta x_n) = f_1(x_1, \ldots, x_n) + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \ldots + \Delta x_n \frac{\partial f_1}{\partial x_n} + R_n;$$

……...

$$f_n(x_1 + \Delta x_1, \ldots, x_n + \Delta x_n) = f_n(x_1, \ldots, x_n) + \Delta x_1 \frac{\partial f_n}{\partial x_1} + \ldots + \Delta x_n \frac{\partial f_n}{\partial x_n} + R_n;$$

where R_n are the members of the second order or higher which will be subtracted in the course of the next transformations.

The task is transformed to the solution of the following system of linear equations:

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial x_1} & \ldots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \\ \vdots \\ -f_n \end{bmatrix}.$$

In this system the matrix of partial derivatives is called the Jacobi matrix and marked as $W(X)$W(X). An example of such matrix for the real task could be found in Table 1 (chapter 1.3.1).

Iterations of value Δx_i found for a certain $(m+1)$ step are used as the amendments to the previous approach

$$x_1^{(m+1)} = x_1^{(m)} + \Delta x_1,$$

KKKKKKK

$$x_n^{(m+1)} = x_n^{(m)} + \Delta x_n.$$

A general iterative formula in matrix presentation could be presented as:

$$X^{(m+1)} = X^{(m)} - W^{-1}\left[X^{(m)}\right]F\left[X^{(m)}\right],$$

where $F\left[X^{(m)}\right]$ is the column vector of the functions values $f_1, f_2, ..., f_n$ for approaching $X^{(m)}$; $W^{-1}\left[X^{(m)}\right]$ is the inverse Jacobi matrix.

The algorithm of Newton method is given in Figure 3.7.

Certain difficulties during using the algorithm of the Newton method emerge due to the rotation of Jacobi matrix. The rotation methods of matrices known from linear algebra are used for this purpose.

There are a lot of variants to apply the Newton method. For example, a modified Newton method

$$X^{(m+1)} = X^{(m)} - W^{-1}\left[X^{(0)}\right]F\left[X^{(m)}\right].$$

In this method one does not need to calculate the inverse Jacobi matrix at every step of the calculations; that simplifies the algorithm, but slows convergence and sensitizes the method more to the choice of the initial approach.

Newton method with a parameter τ

$$X^{(m+1)} = X^{(m)} - \tau W^{-1}\left[X^{(m)}\right]F\left[X^{(m)}\right].$$

This method is similar to the method of successive overhead relaxation for the systems of linear equations.

Various hybrid methods are also used in which the Newton method is united with the method of simple iteration.

The convergence of the Newton method is estimated by calculation of the index

$$q = \frac{M^2 LP}{2} < 1,$$

where

$$M \geq \left\| W^{-1}(X) \right\|,$$

$$L \geq \left\| W(X) \right\|,$$

$$P \geq \left\| F(X) \right\|,$$

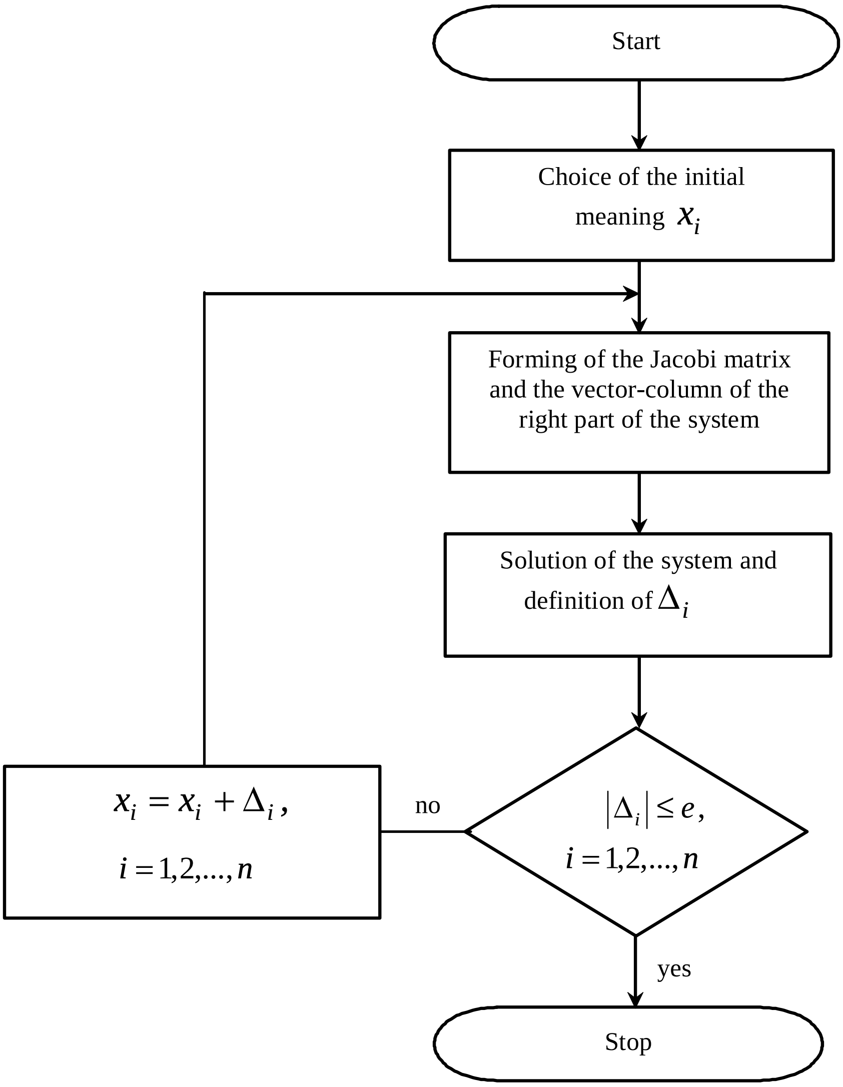

Start
Choice of the initial meaning x_i
Forming of the Jacobi matrix and the vector-column of the right part of the system
Solution of the system and definition of Δ_i
$|\Delta_i| \le e,$ $i = 1,2,...,n$
no
$x_i = x_i + \Delta_i,$ $i = 1,2,...,n$
yes
Stop

Figure 3.7

thus

$$\lim_{l\to\infty} MP\sum_{m=0}^{l} q^{2^m-1} \to 0.$$

The error on m iteration concerns the following inequality

$$\Delta \le MP\frac{q^{2^m-1}}{1-q^{2^m}}.$$

3.4 Remarks

From the variety of considered methods the bisection method is more simple and reliable, but it is very slow (as interval of the root’s position could be cut only two times with each next step). The Newton method, as well as the methods of simple interaction, has a convergence problem. But they are comparatively quick if the initial approximation choice is correct.

In practice, while solving systems of nonlinear equations, only the Newton method could be used. Both the initial approximation and the interval resolution are to be based upon the practical cases and the physical essence of the tasks. It should also be stressed that dealing with the method one needs to be very careful with complex roots and multitudes of the errors accumulated.

Exercises

1. Give the rules for the definitions of quantity and type of the roots of nonlinear equations.
2. Construct the algorithms to solve the following nonlinear system via different methods. Perform them on computer.

$$x^4 + x^2 - x + 1 = 0;$$
$$x^3 - 7x + 2 = 0;$$
$$x^5 - 10x^2 + 5 = 0;$$
$$x^3 - 2x^2 - 1 = 0;$$
$$x^5 - x^2 - 3 = 0;$$
$$x^2 - \ln x = 0;$$
$$x - 10\sin x = 0;$$
$$x^3 - 2tgx + 5 = 0;$$
$$x^2 - x + tgx = 0.$$

3. Solve nonlinear equation with complex roots

$$x^6 + 2x^5 + 3x^4 + 4x^3 - x - 1 = 0.$$

4. Find the Jacobi matrices for the following systems

$$\begin{cases} x_1 + 2x_2 + x_3^2 = 14, \\ \ln|x_1| + x_2^2 - x_3 = 1, \\ x_1 + \sin\dfrac{px_2}{4} - x_3 = 1 \end{cases}$$

and

$$\begin{cases} \textit{ñîs}\ x_1 + x_2 + \ln\dfrac{x_3}{3} = 2, \\ x_1 + x_2^2 + x_3 = 8, \\ \dfrac{arctgx_1}{p} + \dfrac{x_2}{2} - \dfrac{x_3}{12} = 1. \end{cases}$$

5. Find the inverse matrices for systems from exercise 4.
6. Construct the algorithm and program to solve systems from exercise 4 via Newton and modified Newton methods. Perform it on computer and find roots.

Chapter 4. Ordinary Differential Equations

4.1 Introduction

An ordinary differential equation has an endless amount of solutions. To obtain a concrete solution it is necessary to account for additional conditions. These conditions can differ and demand different problems. In case when additional conditions are set at one independent variable value, the Caushe problem should be considered (problem with an initial value). If the conditions are set for two or more independent variable's values, the problem becomes boundary-value problem. In the Caushe problem the additional conditions are named initial, and in a boundary-value – boundary. To solve these problems different methods and algorithms are used.

Caushe problem can be formulated in the following way.

Suppose, we have a differential equation of the first order

$$\frac{dy}{dx} = f(x,y)\,. \tag{4.1}$$

To find a function on the interval from $x = a$ *to* $x = b$ that satisfies both equation (4.1) and the initial condition $y(a) = y_0$ (it is thus always assumed that there is a unique decision for the whole interval).

A problem requiring a solution of an ordinary differential equation with additional conditions put at several independent variable values is named boundary-value problem.

We will consider a boundary-value problem on the example of ordinary differential equation of the second order:

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right)$$

with boundary terms $y\,(a) = A$, $y\,(b) = B$.

The methods of solution of equations of higher orders are similar to the equations of low orders.

4.2 Solution of the Caushe Problem

Decomposition of the function serves as a basis for a variety of methods designed to solve differential equations y in the Taylor row around initial point

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{1}{2}h^2 y''(x_0) + ...,$$

where h is a distance (step) between the initial point x_0 and point $x_1=x_0+h$.

The different amounts of members of decomposition (in multi-step methods in combination with interpolations formulas) determine the exactness of calculations for different methods. While using these methods on computer one should distinguish transaction errors due to the lack of significant numbers, used in the course of computer calculations, and the error of transaction (limitation) which is a methodical error that is related to approximation of solution by eventual rows in place of the infinite, for example, by the Taylor rows.

As a result there are two types of errors:

Local error is a sum of errors which appear in the process of calculations on a concrete step.

Global (total) error is a difference between the exact and calculated meanings, which includes the so-called error of distribution that results from the accumulation of errors at the previous stages of calculation.

Local error Δ depends on the order of method p and coefficient c

$$\Delta \leq ch^{p+1}.$$

Coefficient c concerns the derivatives and the length of the interval. During approximation of the solution by the Taylor rows it is related to the degree of members in the row which is taken into account.

The methods of solution of the Caushe problem are divided into one-step and multi-step.

In one-step methods, in order to find the next point $y = f(x)$, it is necessary to have only information about one previous step (methods of Euler and Runge-Cutt).

In multi-step methods (prognosis and correction), in order to find the next point $y = f(x)$, the information about more than one of the previous points is necessary. To get enough exact numeral data iterative procedures are often used (for example, in the methods of Milne, Adams, Hamming).

4.2.1 One-step Methods

Among simple one-step methods, which need minimum calculations, but give the possibility to obtain result with comparatively low exactness, is the Euler method.

In this method to estimate the next point $y=f(x)$ one should take into account only one linear member in the Taylor formula

$$y(x_0+h)=y(x_0)+hy'(x_0),$$

where *y¢* (x_0) could be obtained from the initial equation.
This process can be extended for the following steps

$$y_{n+1} = y_n + hf(x_n, y_n).$$

The Euler method is the method of the first order $(p = 1)$

$$\Delta \le ch^2,$$

where $c = (M_1 + M_0 M_2)/2$, M_0, M_1, M_2 – are regarded as

$$M_0 \ge |f(x,y)|,$$
$$M_1 \ge \left|\frac{\partial f(x,y)}{\partial x}\right|,$$
$$M_2 \ge \left|\frac{\partial f(x,y)}{\partial y}\right|,$$

for all $x \in [a,b]$ and $y = y(x)$.

The Euler method is often unsteady because of the error of truncation: small local errors result in the considerable increase of the global error.

This method can be improved in a number of different ways.

Among them are the Corrected Euler method and the Modified (Improved) Euler method.

Iterative formulas for these methods are the following:

$$y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_n + h, y_n + hy_n^*)\right]$$

and
$$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}; y_n + \frac{h}{2} y_n^*\right),$$

where
$$y_n^* = f(x_n, y_n).$$

Geometrical interpretations are represented in Figures 4.1, 4.2.

These are the methods of the second order. Their errors have the third degree that is achieved by improving the derivative’s approximation. The idea consists in trying to save or estimate the member of the second order in the Taylor row. However, in order to increase the exactness considerably, the additional load of computer to calculate y_n^* is required. Yet the bigger exactness can be attained by calculating the higher derivative and by maintenance of a number of Taylor row members. Such methods are called Runge-Cutt methods.

One can account for the principle on which the modified Euler method is built by using the Taylor row and keeping in it a member with h^2. Approximation of $y''(x_0)$ is processed using the finite-difference form

$$y''(x_0) = \frac{\Delta y'}{\Delta x} = \frac{y'(x_0 + h) - y'(x_0)}{h}.$$

In this way in a finite-difference form it is possible to calculate the higher derivative: value of n-derivative after the values of the previous $(n-1)$.

Runge-Cutt method gives a set of formulas to calculate the coordinates of internal points which are required to realize this idea. As there is a number of methods to find these points, the Runge-Cutt method unites the whole class of methods for the solution of differential equations.

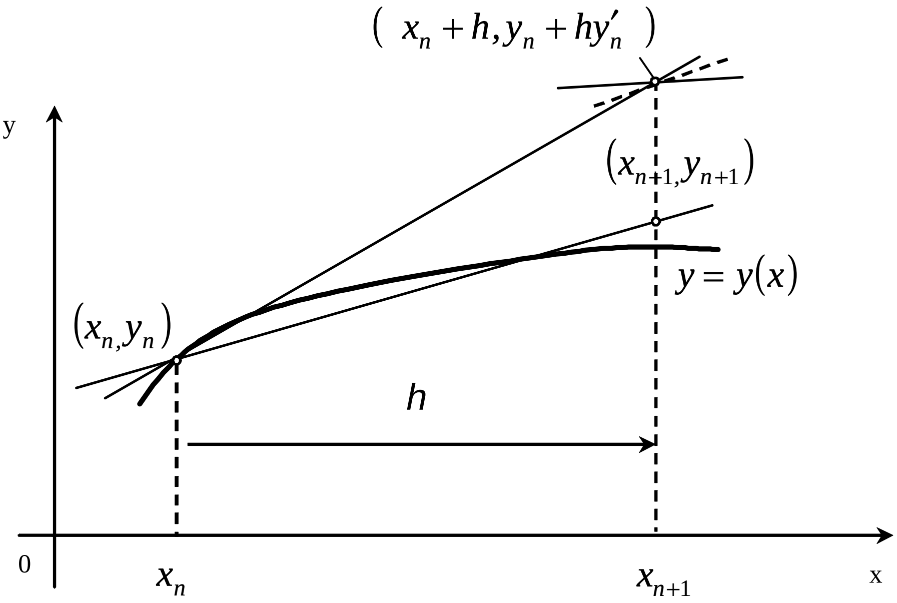


Figure 4.1

A more widespread classic method of the fourth order is given below:

$$y_{n+1} = y_n + \frac{k_0 + 2k_1 + 2k_2 + k_3}{6},$$

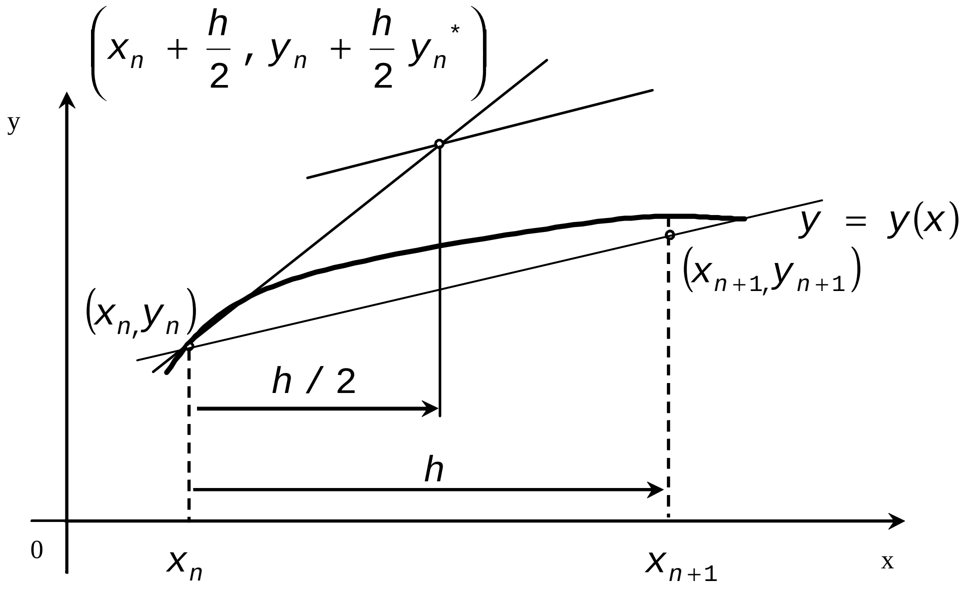


Figure 4.2

where

$$k_0 = hf(x_{n,} y_n); \; k_1 = hf\left(x_n + \frac{h}{2}; y_n + \frac{k_0}{2}\right);$$

$$k_2 = hf\left(x_n + \frac{h}{2}; y_n + \frac{k_1}{2}\right); \; k_3 = hf(x_n + h; y_n + k_2).$$

The Euler method and its modifications are called the Runge-Cutt methods of the first and of the second order. The Runge-Cutt method has higher exactness that allows considerable multiplying of the steps of solution. Its maximal size is determined by an acceptable error. Such choice is often carried out automatically and is included as a component part in the algorithm built by the Runge-Cutt method.

All of the Runge-Cutt formulas could be used to solve differential equations of higher orders and systems of differential equations. Equations of order *n* can be regarded as *n* differential equations of the first order.

As an example we will consider the solution of an ordinary differential equation of the second order:

$$\frac{d^2 y}{dx^2} = g\left(x, y\frac{dy}{dx}\right).$$

Suppose, $z = \frac{dy}{dx}$, then $\quad \frac{dz}{dx} = \frac{d^2 y}{dx^2};$

and the system is the following $\begin{cases} \dfrac{dz}{dx} = g(x,y,z), \\ \dfrac{dy}{dx} = f(x,y,z), \end{cases}$

where $f(x,y,z,) = z$.

The Caushe problem in this case contains two initial conditions:

$$y(x_0) = y_0; \quad z(x_o) = y'(x_0) = z_0.$$

The Runge-Cutt formulas in this case are:

$$y_{n+1} = y_n + k \quad \text{and} \quad z_{n+1} = z_n + l$$

where $k = \dfrac{k_0 + 2k_1 + 2k_2 + k_3}{6}$, and $l = \dfrac{l_0 + 2l_1 + 2l_2 + l_3}{6}$.

Here

$$k_0 = hz_0, \qquad l_0 = hg(x_n, y_n, z_n),$$

$$k_1 = h\left(z_0 + \frac{l_1}{2}\right), \qquad l_1 = hg\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}\right),$$

$$k_2 = h\left(z_0 + \frac{l_2}{2}\right), \qquad l_2 = hg\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}\right),$$

$$k_3 = h\left(z_0 + \frac{l_3}{2}\right), \qquad l_3 = hg(x_n + h, y_n + k_3, z_n + l_3).$$

It was previously noted that the error of truncation in the Runge-Cutt method of n order is $\Delta \le ch^{p+1}$. Calculation of the upper boundary for the coefficient *c* is an intricate problem related to the necessity of a number of additional parameters estimation. There are a few methods for the effective calculation of *c*. More practically suitable is the Richardson extrapolation method (another name is the Runge method), when the value y_n is found consistently with the step *h* and *h/2*, and the numbers obtained are put into the equation, from which *c* is determined:

$$y_n^{(h)} + ch^{p+1} = y_n^{\left(\frac{h}{2}\right)} + c\left(\frac{h}{2}\right)^{p+1},$$

that corresponds to the exact meaning of y_n.

We will get the evaluation correlation:

$$c = \frac{2^{p+1}}{2^{p+1}-1} \frac{\left[y_n^{\left(\frac{h}{2}\right)} - y_n^{(h)}\right]}{h^{p+1}}.$$

It is possible to select the general features of one-step methods:

1. In order to obtain information on a new point, it is necessary to have information only about one previous point. Let us call this property the "self-starting".
2. Basically, a one-step method is a decomposition of function in the Taylor row, members of which with degrees to p inclusive are saved. Number p is named as the order of the method. An error on a certain step has order $p+1$.
3. All one-step methods do not require any calculation of derivative. Only the function is calculated, but its values can be required in a few intermediate points.
4. The property of "self-starting" allows changing the amount of steps of calculation easily.
5. It is impossible to estimate the error without additional calculations.

4.2.2 Multi-step Methods

In these methods, in order to calculate the value of the new point, the information about a few values obtained previously is used. Two formulas are used for this purpose: prognosis and correction. An algorithm of calculation for all methods of prognosis and correction is identical and is represented in Figure 4.3. Indicated methods differ only in formulas and are not a characteristic of the "self-starting", as they require information on the previous values. Before using the method of prognosis and correction, calculate initial data by any one-step method. Often for this purpose the Runge-Cutt method is used.

The calculations are processed in the following way. At first, using the formula of the prognosis and initial variable values, find the value $y_{n+1}^{(0)}$. An index (0) means that the value, which is forecasted, is one of the sequences of values y_{n+1} at the stage of their clarification. After the value $y_{n+1}^{(0)}$ by initial differential equation (4.1.) find derivative $y_{n+1}^{(0)\prime} = f\left(x_{n+1}, y_{n+1}^{(0)}\right)$ which should be put in the formula of correction to calculate the specified value $y_{n+1}^{(j+1)}$. In turn, after $y_{n+1}^{(j+1)}$ find derivative $y_{n+1}^{(j+1)\prime} = f\left(x_{n+1}, y_{n+1}^{(j+1)}\right)$. If this value is close to the previous, it is brought into the correction formula and the iterative process continues. It should be regarded in case of closeness of values of derivative y_{n+1}. After it the process repeats itself at the next step where y_{n+2} is calculated.

Usually formulas of prognosis and correction are obtained by methods of numeral integration.

If the differential equation $y' = f(x, y)$ integrates in the interval the values from x_n to x_{n+k}, the result will be the following:

$$y(x_{n+k}) - y(x_n) = \int_{x_n}^{x_{n+k}} f(x, y)dx.$$

This integral can not be calculated directly, because *y(x)* is an unknown function. The choice of the method of integration determines the method of solution of differential equations. At the stage of prognosis it is possible to use any formula of numeral integration, if the initial value $y'(x_{n+1})$ is not included into it.

In Table 4.1 more widespread formulas of prognosis and correction are given.

For most methods of prognosis and correction an error can be estimated by such correlation:

$$\Delta \le \frac{1}{5}\left[y_n^{(0)} - y_n^{(j)}\right].$$

One should take into account that the optimal quantity of iterations equals to two for each step. Quantity of iterations, step and error of solution are connected and could be controlled by charging of the step.

Basic features of the multi-step methods:

1) Using these methods it is impossible to start solution of the problem, without information about the initial value of the function in a few previous points;

2) It is possible to get estimation of the truncation error without calculating the additional data;

3) The methods of prognosis and correction do not allow changing the step of calculations easily; for this purpose it is necessary to begin iterative process at first.

Table 4.1

Method	Prognosis formula	Correction formula
Milne	$y_{n+1} = y_{n-3} + \frac{4}{3}h*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2}),$ $\Delta \le \frac{28}{90}h^5 y^{(5)},$ where y' - fifth derivative $f(x,\ y)$	$y_{n+1} = y_{n-1} + \frac{1}{3}h*$ $*(y'_{n+1} + 4y'_n + y'_{n-1}),$ $\Delta \le \frac{1}{90}h^5 y^{(5)}$
Adams – Bashforth	$y_{n+1} = y_n + \frac{1}{24}h*$ $*(55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}),$ $\Delta \le \frac{251}{720}h^5 y^{(5)}$	$y_{n+1} = y_n + \frac{1}{24}h*$ $*(9y'_{n+1} - 19y'_n - 5y'_{n-1} + y'_{n-2}),$ $\Delta \le \frac{19}{720}h^5 y^{(5)}$
Hamming	$y^{(0)}_{n+1} = y_{n-3} + \frac{3}{4}h*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2})$ Clarification of prognosis $\bar{y}^{(0)}_{n+1} = y^{(0)}_{n+1} +$ $+\frac{112}{121}(y_n - y^{(0)}_n)*$ $*[\bar{y}^{(0)}_{n+1}]' = f(x_{n+1}, \bar{y}^{(0)}_{n+1})$	$y^{(j+1)}_{n+1} = \frac{1}{8}*(9y_n - y_{n-2}) +$ $+\frac{3}{8}h*\left([\bar{y}^{(j)}_{n+1}]' + 2y'_n - y'_{n-1}\right)$
	$y_{n+1} = y_{n-3} + \frac{4}{3}h*$ $*(2y'_n - y'_{n-1} + 2y'_{n-2}),$ $\Delta \le \frac{28}{90}h^5 y^{(5)}$	$y_{n+1} = \frac{1}{8}*$ $*[9y_n - y_{n-2} + 3h(y'_{n+1} + 2y'_n - y_{n-1})],$ $\Delta \le \frac{1}{40}h^5 y^{(5)}$

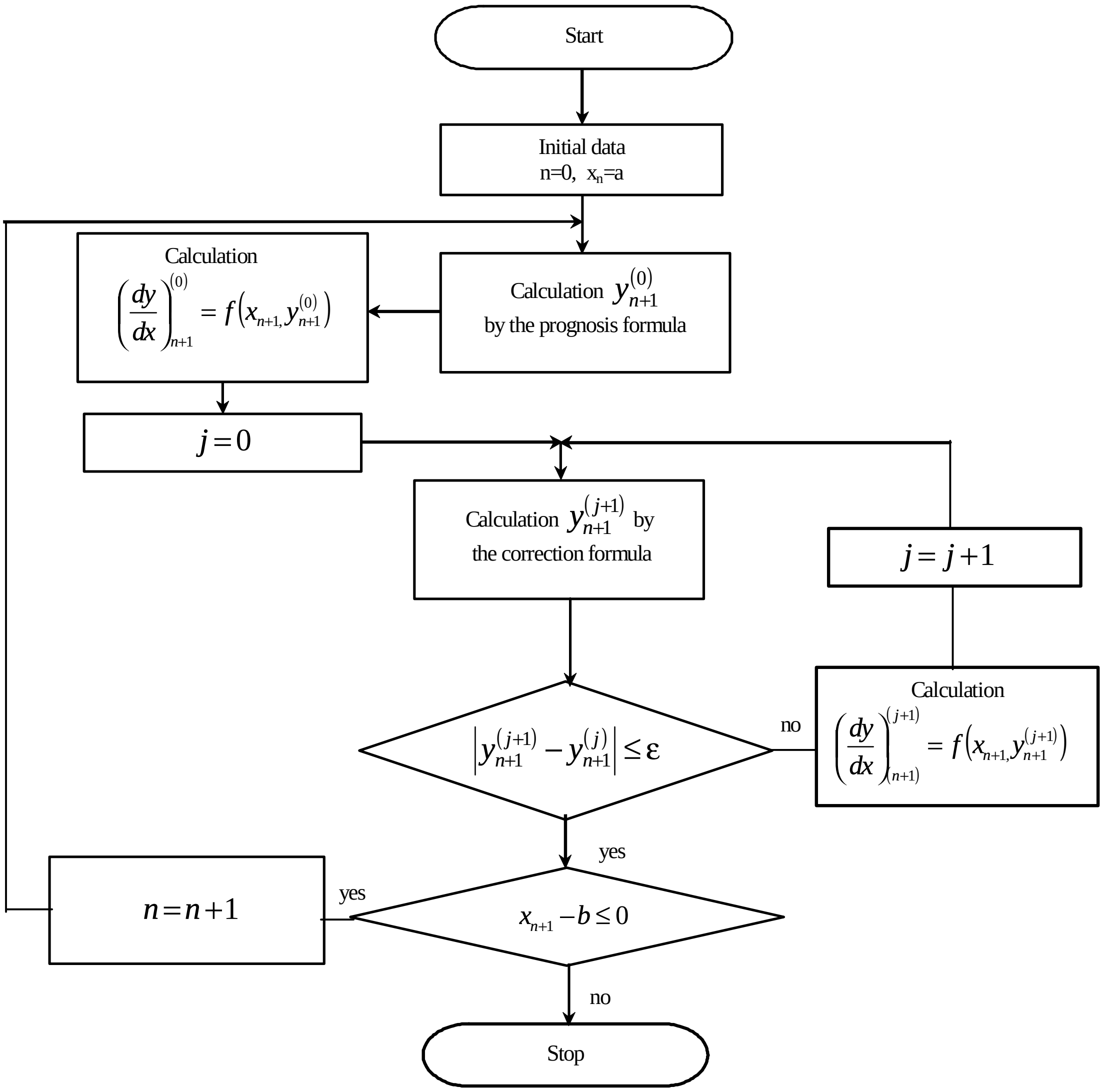

Start
Initial data
n=0, x_n=a
Calculation $\left(\frac{dy}{dx}\right)_{n+1}^{(0)} = f\left(x_{n+1}, y_{n+1}^{(0)}\right)$
Calculation $y_{n+1}^{(0)}$
by the prognosis formula
$j=0$
Calculation $y_{n+1}^{(j+1)}$ by
the correction formula
$j=j+1$
$\left|y_{n+1}^{(j+1)} - y_{n+1}^{(j)}\right| \le \varepsilon$
no
Calculation $\left(\frac{dy}{dx}\right)_{(n+1)}^{(j+1)} = f\left(x_{n+1}, y_{n+1}^{(j+1)}\right)$
yes
$n=n+1$
yes
$x_{n+1} - b \le 0$
no
Stop

Figure 4.3

4.2.3 “Rigid” Problems

There are ordinary differential equations for which it is difficult to get an appropriate solution using the methods described higher. The origin of such problems is related to the concept of time constant of differential equation as interval of time, when variable part of solution diminishes in e times. Equation of order *n* has *n* time constants; if any two of them strongly (in practice one hundred times and more) differ in size or some of them are very small in comparison with the time of solution, the problem is named “rigid” and it is practically impossible to solve it using ordinary numerical methods. In such cases the step must be small enough to make it possible to account for the components of solution which change quicker, even after their accounting becomes practically unnoticeable. But diminishing the step leads to increasing the computations time and accumulation of errors. Special methods of solving such problems are frequent in the theory of automatic control developing.

The simpler of them are called reverse Euler method, in which the solution is put in accordance with the following correlation:

$$y_{n+1} = y_n + hf(x_n, y_{n+1}).$$

In practice, equations in which coefficients of derivatives strongly (one hundred times and more) differ one from another have the “rigidity” features.

4.3 Solution of the Boundary-Value Problem

The methods of solution of the boundary-value problems are examined on the example of ordinary differential equation of the second order

$$\frac{d^2y}{dx^2} = f(x, y, \frac{dy}{dx})$$

and boundary terms $y(a) = A$, $y(b) = B.$ Methods of solution of the boundary-value problem are divided into several groups: methods that allow to reduce the solution of such problems to several Caushe problems and in which the solution could be used as one of the methods to solve the Caushe problem (method of “shooting”); difference methods; projections methods.

4.3.1 Method of “Shooting”

If ordinary differential equation of the second order is linear, it looks like:

$$y'' = f_1(x)y' + f_2(x)y + f_3(x)$$

at $y(a) = A$, $y(b) = B.$

The boundary-value problem can be brought to the Caushe problem by using an additional initial condition, instead of $y(a)=A$ we enter $y'(a)=\alpha_{1.}$

After finding the solution $y_1(x)$, it is possible to account for another initial condition $y'(a)=\alpha_2$ and to get another solution $y_2(x)$). If $y_1(b)=\beta_1$ and $y_2(b)=\beta_2$, thus $\beta_1 \neq \beta_2$ the general solution is:

$$y(x) = \frac{1}{b_1 - b_2}\left[(B - b_2)y_1(x) + (b_1 - B)y_2(x)\right].$$

This solution satisfies both initial conditions.

This method can be used only for linear equations where the principle of linear superposition of solution is correct.

4.3.2 Difference Methods

The difference methods are the powerful means of the numerical solution of ordinary differential equations and differential equations in parts derivative. According to these methods presentation of an independent argument lies in an interval $[a,\ b]$ as the discrete multitude of points knots x_i, $i=0,...,n$, $x_0=a$, $x_n=b$, which is called a net.

Most distributions were obtained by an even net with a step $x_i - x_{i-1} = h$. Thus, in place of the continuous function $f(x)$ a net function $y_i=f(x_i)$ is examined. Digitization of a function of several variables (for example, two) will be the following:

$$x_{ij} \quad i=0,...,n, \quad j=0,...,m, \quad y_{ij}=f(x_{ij}).$$

Except for a more widespread rectangular net, there exist polar, three-cornered, mowed nets etc. which are represented in Figure 4.4. Multidimensional nets find use in problems with partial derivatives.

The solution of the problem by difference methods consists of two stages:

– getting the difference approximation of differential equations and researching the algebraic equations obtained;

– solution of algebraic equations.

While getting the difference patterns charts, the common requirement should be taken into account that a difference pattern could be used to deal with basic features of the initial differential equation. Such difference patterns could be obtained using variation principles and integral correlations. The estimation of exactness of difference patterns could be added up to finding the error of approximation and firmness. A net function can be examined as a function of integer number argument

$$y(i)=y_i\,, \quad i=0, \pm 1, \pm 2, \dots\ .$$

It is possible to determine the operations which are the difference analogues of operations of differentiation and integration.

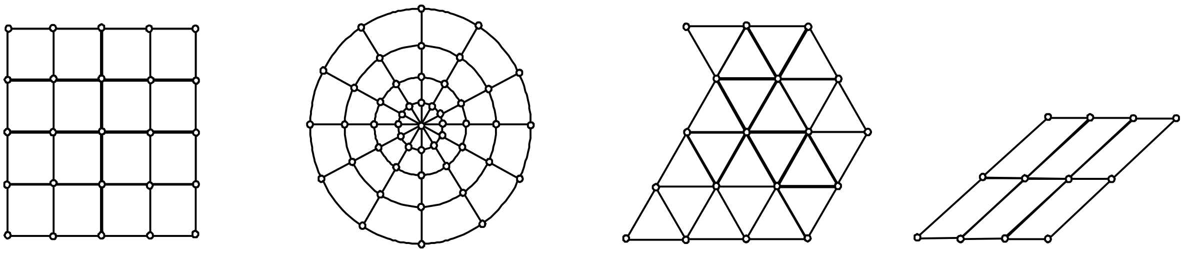


Figure 4.4

The differences of the first order are the analogues of the first derivatives:

$\Delta y_i = y_{i+1} - y_i$ - forward difference;
$\nabla y_i = y_i - y_{i-1}$ - backward difference;
$dy_i = \frac{1}{2}(\Delta y_i + \nabla y_i) = \frac{1}{2}(y_{i+1} - y_{i-1})$ - central difference.

One should take into account that $\Delta y_i = \nabla y_{i+1}$.

Therefore it is possible to get the differences of the second order

$$\Delta^2 y_i = \Delta(\Delta y_i) = \Delta(y_{i+1} - y_i) = y_{i+2} - 2y_{i+1} + y_i \ ,$$

$$\Delta\nabla y_i = \Delta(y_i - y_{i-1}) = (y_{i+1} - y_i) - (y_i - y_{i-1}) = y_{i+1} - 2y_i + y_{i-1},$$

that $\Delta^2 y_i = \Delta\nabla y_{i+1}$.

The difference of *m* order is:

$$\Delta^m y_i = \Delta(\Delta^{m-1} y_i).$$

It follows that

$$\sum_{j=k}^{i} \Delta y_i = y_{i+1} - y_k \ , \qquad \sum_{j=k}^{i} \nabla y_i = y_i - y_{k-1}.$$

On the multitude of knots of the net, which is named a template (unidimensional templates are examined in this section, and in the next section the two-dimensional ones are regarded), we will replace continuous differential operator *Ly* by the difference operator *L_hy*.

For example, difference operators for the first net derivate on three knots (*x-h, x, x+h*)

$$L_h^+ y = \frac{y(x+h) - y(x)}{h} = y_x^+ ,$$
$$L_h^- y = \frac{y(x) - y(x-h)}{h} = y_x^- ,$$
$$L_h^0 y = \frac{y(x+h) - y(x-h)}{2h} = y_x^0 ,$$

forward, backward and central difference derivatives accordingly.

Similar to the second derivative

$$L_h y = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} = \frac{y_x^+(x) - y_x^-(x)}{h} =$$
$$= \frac{y_x^-(x+h) - y_x^-(x)}{h} = y_{xx}^-(x).$$

Solving the boundary-value problem equations for all n knots the areas change on interval *[a, b]*. Using two additional boundary conditions $y_0=y(a)$ and $y_n=y(b)$, we can get the system with *n-1* algebraic equations and *n-1* unknown y_i. If initial ordinary differential equation is linear, a problem leads to solving the system of linear algebraic equations, and if nonlinear – to the nonlinear or transcendent algebraic systems.

4.4 Remarks

Comparing efficiency of one-step and multi-step methods of Caushe problem solution such features could be selected:

1. Multi-step methods require the greater volume of computer memory, as plenty of initial data are operated .

2. When using multi-step methods, there is a possibility of estimation of error on a step. This allows to select the optimal value of the step.

3. At identical exactness multi-step methods require less volume of calculations. For example, in a Runge-Cutta method with the fourth order of exactness four values of function are calculated at every step, and for provision of convergence of method of prognosis and correction of that order of exactness – two are sufficient.

4. One-step methods, unlike multi-step ones, allow at once to begin the decision of the problem (“self-starting”) and it is easy to change a step in the process of calculations.

Before the beginning of the problem’s solution it is necessary to conduct verification of the “rigidness”, and in the case of positive result to use the special methods. If the Caushe problem is very difficult, usually the method of prognosis and correction gets advantage. Beginning of solution of problem is conducted here by one-step methods. If for the calculation of value y_i two iterations cause a large truncation error, it is necessary to decrease the step. On the other hand, at very small error of truncation it is possible to multiply a step, thus promoting the fast-acting, but here all the process of solution needs to be done at first. Sometimes, in practice it is required to minimize time of preparation of problem for the solution. Then it is expedient to use Runge-Cutta methods.

A more universal method for solution of the boundary-value problem is a difference method, but in case of linear problems the “shooting” method could be used based on solution of several Caushe problems.

In conclusion, it should be noted that for the effective solution of problem it is very important to have experience, intuition and qualification for the user, both at setting the problem and in the process of choice of the method of development of algorithm and program for the computer. It is thus often convenient to use the prepared programmatic facilities which exist already (for example, in the program systems Maple, Mathematika, Matlab etc).

In this chapter we deal with numerical methods of solving the Caushe and the boundary-value problems with differential equations.

In order to solve the traditional problems within the Caushe problem it is sufficient to use one-step and multi-step methods. The problem becomes more complicated if there are some rigid features present; in this case special inexplicit methods should be used to avoid instability and divergence of computations.

As for the boundary-value problem, the difference and “the shootings” methods could be considered as the only simple approaches. Regarding the large variety of other methods available, the following ones could be applied: variation methods (Collatz), methods of integral equations (Keller) etc.

Exercises

1. Give the statement of the Caushe and boundary value problems. What is the difference?
2. Give examples of the ordinary differential equation which could be solved using the numerical methods only.
3. Could the boundary value problem of the first order be a differential equation?

4. What are the methods of estimation of the unistep and multistep methods' errors?
5. What is the feature of the "self-starting" in the unistep methods?
6. Give all the possible difference representatives of the first and second derivatives.
7. What is the "rigidness" feature of the ordinary differential equations?
8. What is the method to get the general solution of the boundary value problem from several Caushe problems solution using the "shooting" method?
9. Find the solutions of each of the following differential equations using the Runge-Cutt methods (1, 2, 4 orders) on interval $0 \le x \le 1$with step $h = 0,1$and initial condition $y(0) = 1$:

$$a)\ y' + 8y - 15 = 0;$$
$$b)\ y' - by + 9x = 0;$$
$$c)\ 2y'' + 3y' = 0;$$
$$d)\ y'' + 3y' + 6y + x - 1 = 0;$$
$$e)\ y' - 2x = 3;$$
$$f)\ y'' - y' - y + e^{x} = 0.$$

Estimate the errors of the solution.

10. Find the solutions of the boundary value problem for the equations from ex. 9 c, d, f using the "shooting" and difference methods with the boundary conditions $y(0) = y(1) = 1$ and step $h = 0,1$. Estimate the errors of the solution.

Chapter 5.Differential Equations in Partial Derivatives (Mathematical Physics Problems)

5.1 Introduction

Engineers and researchers have to deal with the solution of differential equations in partial derivatives in many fields of science and technique: in aero- and hydrodynamics, nuclear physics, telecommunications etc. Mathematical models with differential equations in partial derivatives are widely used in the theory of automatic control and in measuring techniques. Such equations contain partial derivatives in a few independent variables. Let us consider differential equation of the second order with two variables:

$$\begin{aligned} &A(x,y)\frac{\partial^2 f}{\partial x^2}+B(x,y)\frac{\partial^2 f}{\partial x\partial y}+C(x,y)\frac{\partial^2 f}{\partial y^2}+ \\ &E(x,y)\frac{\partial f}{\partial y}+D(x,y)\frac{\partial f}{\partial x}+F(x,y)=G(x,y). \end{aligned} \tag{5.1}$$

Like ordinary differential equations the unique solution of equation (5.1) can be obtained only by setting additional conditions, but as two independent variables x and y are present here, the condition must be set for some curve in plane xy. This condition can be imposed on a function f or on its derivative and depends upon the equation which determines its form and character of behaviour.

There are three types of differential equations of the second order:

– elliptic, if $B^2-4AC<0$;
– parabolic, if $B^2-4AC=0$;
– hyperbolic, if $B^2-4AC>0$.

Equations can change from one form to another depending upon values of the coefficients.

Elliptic equations describe stationary processes; thus the problem is set in the determined limits and the boundary values are set at every point of the region's boundary. Other two types of equations describe evolutional processes. In such problems it is more frequent when on one part of the boundary the boundary conditions are set, and on the other - the initial ones.

The examples of several differential equations in partial derivatives, which describe the different types of problems, are given in Table 5.1.

Table 5.1

Equation	Mathematical form	Examples of the equation problems
Laplace	$\Delta f = 0$	Stationary flows of liquids, heat fields
Poisson	$\Delta f = -R$	Heat transfer with the inner heat sources
Diffusions	$\Delta f = \frac{1}{h^2}\frac{\partial^2 f}{\partial t^2}$	Non-stationary heat conduction
Wave	$\Delta f = \frac{1}{c^2}\frac{\partial^2 f}{\partial t^2}$	Wave propagation (sound, electromagnetic etc.)
Biharmonic	$\Delta^2 f = F(x, y)$	Plates deformation

In Table 5.1 the accepted denotations of more widespread operators are used, such as:

Laplace operator $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$;

Biharmonic operator $\Delta^2 f = \frac{\partial^4 f}{\partial x^4} + 2\frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4}$.

There are two methods of solution of differential equations in partial derivatives: difference method (method of finite differences) and method of finite elements. In the modern applied mathematics both methods are considered as interpretations, which describe how to use the general theory of difference methods in the solution of differential equations in partial derivatives.

The variation calculation lies in the basics of the method of finite elements. Differential equations and boundary terms are used to define the variation problem. In the method of finite elements the physical problem is replaced by a cobbed-smooth model. This method complicates finding of the problem’s solution and demands high qualification and experience. It is unique, as every solution is used only for a concrete problem. The method of finite elements became widely spread for the solution of special problems in theoretical mechanics, hydrodynamics and in the field theory. It is complicated, requires serious preparation and knowledge in the concrete area, and its specificity is described in special textbooks. To solve the problem in automation and control systems difference methods are more frequently used.

In chapter 4.1. basic definitions and rules of difference patterns construction are considered. For differential equations of the second order in partial derivatives a two-dimensional rectangular net is more frequently used. Difference patterns which are applied to the two-dimensional square net with step *h*, presented in Figure 5.1 (index j gets an independent variable *y*, and i belongs to axis x), can be considered as a unidimensional case from chapter 4.2.2.

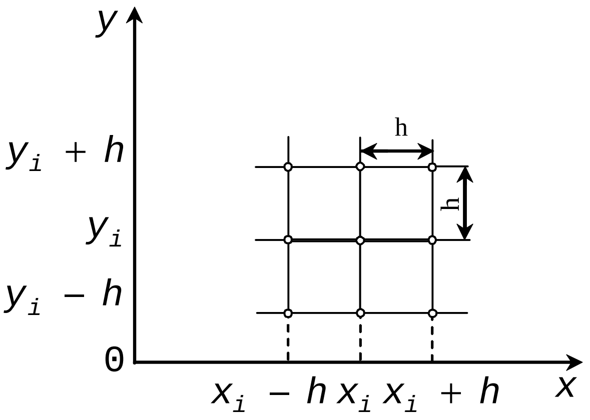
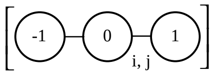


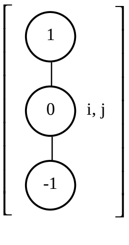
Figure 5.1

To facilitate the denotation $f(x_i + h, y_i)$ could be replaced by $f_{i+1,j}$. Using this denotation, we will get correlations to approach partials derivatives in practice. This could be illustrated by the specific calculation templates (Figure 5.2):

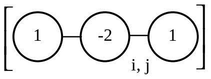
$$\frac{\partial f}{\partial x} \approx \frac{f_{i+1,j} - f_{i-1,j}}{2h} \approx \frac{1}{2h}$$



$$\frac{\partial f}{\partial y} \approx \frac{f_{i+1,j} - f_{i-1,j}}{2h} \approx \frac{1}{2h}$$



$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f_{i+1,j} - 2\,f_{i,1} + f_{i-1,j}}{h^2} \approx \frac{1}{h^2}$$



$$\frac{\partial^2 f}{\partial y^2} \approx \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h^2} \qquad \approx \frac{1}{h^2} \begin{bmatrix} 1 \\ -2 \;\; i,j \\ 1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial x \partial y} \approx \frac{f_{i+1,j+1} - f_{i-1,j+1} - f_{i+1,j-1} + f_{i-1,j-1}}{4h^2} \qquad \approx \frac{1}{4h^2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\frac{\partial^4 f}{\partial x^4} \approx \frac{1}{h^4} \qquad \begin{bmatrix} 1 & -4 & \underset{i,j}{6} & -4 & 1 \end{bmatrix}$$

$$\frac{\partial^4 f}{\partial y^4} \approx \frac{1}{h^4} \qquad \begin{bmatrix} 1 \\ -4 \\ 6 \;\; i,j \\ -4 \\ 1 \end{bmatrix}$$

Figure 5.2

From these elements the most complicated calculation templates could be built for differential equations. Adding derivatives could be performed via superposition of the calculation templates needed. Using this method the templates for Δf and $\Delta^2 f$ (Figure 5.3) are constructed.

All resulted calculation templates have errors of the second order. It is possible to make more exact calculation templates by plugging additional knots into consideration. Sometimes, in order to minimize the distribution of errors, left or right differences are used.

Often difficulties connected with the use of rectangular net emerge, as the boundary has wrong configuration which does not pass through the knots of the net. We will consider an example of such problem's solution for the calculation template of Laplace equation in the area limited by an arbitrary curve which is represented in Figure 5.4.

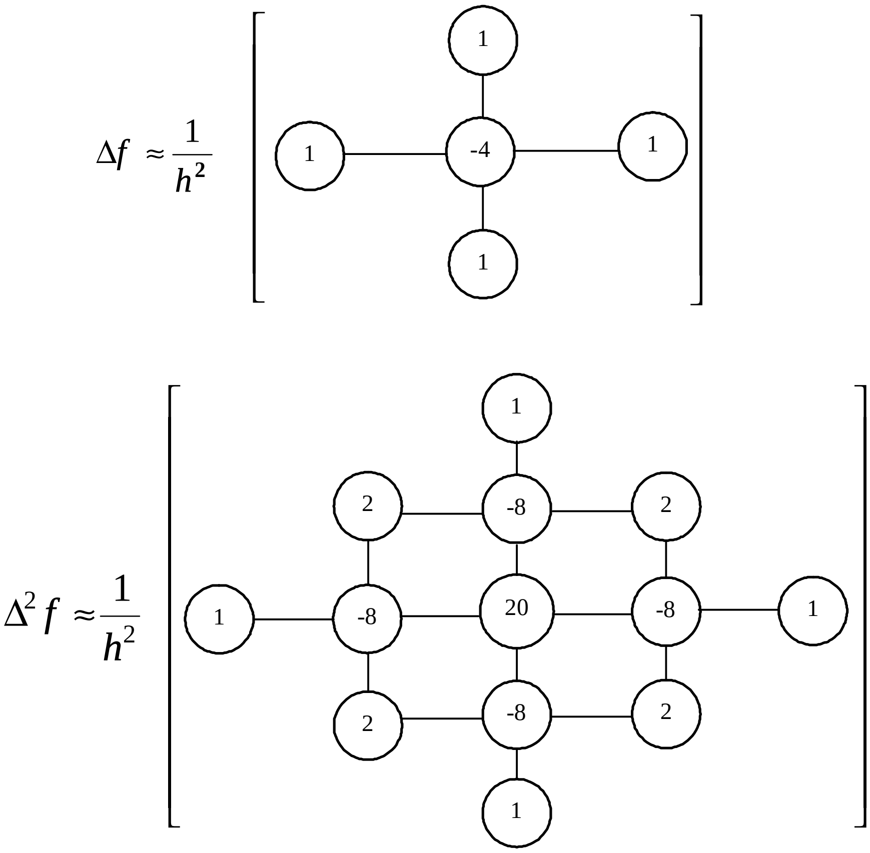

$\Delta f \approx \frac{1}{h^2}$
1
1
-4
1
1
$\Delta^2 f \approx \frac{1}{h^2}$
1
2
-8
2
1
-8
20
-8
1
2
-8
2
1

Figure 5.3

Second partial derivative for knots on the boundary of the region could be written as:

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{\dfrac{f_a - f_{i,j}}{ah} - \dfrac{f_{i,j} - f_{i-1,j}}{h}}{0{,}5(ah+h)};$$

$$\frac{\partial^2 f}{\partial y^2} \approx \frac{\dfrac{f_b - f_{i,j}}{bh} - \dfrac{f_{i,j} - f_{i,j-1}}{h}}{0{,}5(bh+h)}.$$

After superposition

$$\Delta f \approx \frac{2}{h^2}\left(\frac{f_{i-1,j}}{1+a} + \frac{f_a}{a(1+a)} + \frac{f_b}{b(1+b)} + \frac{f_{i,j-1}}{1+b} - \frac{a+b}{ab} f_{i,j}\right).$$

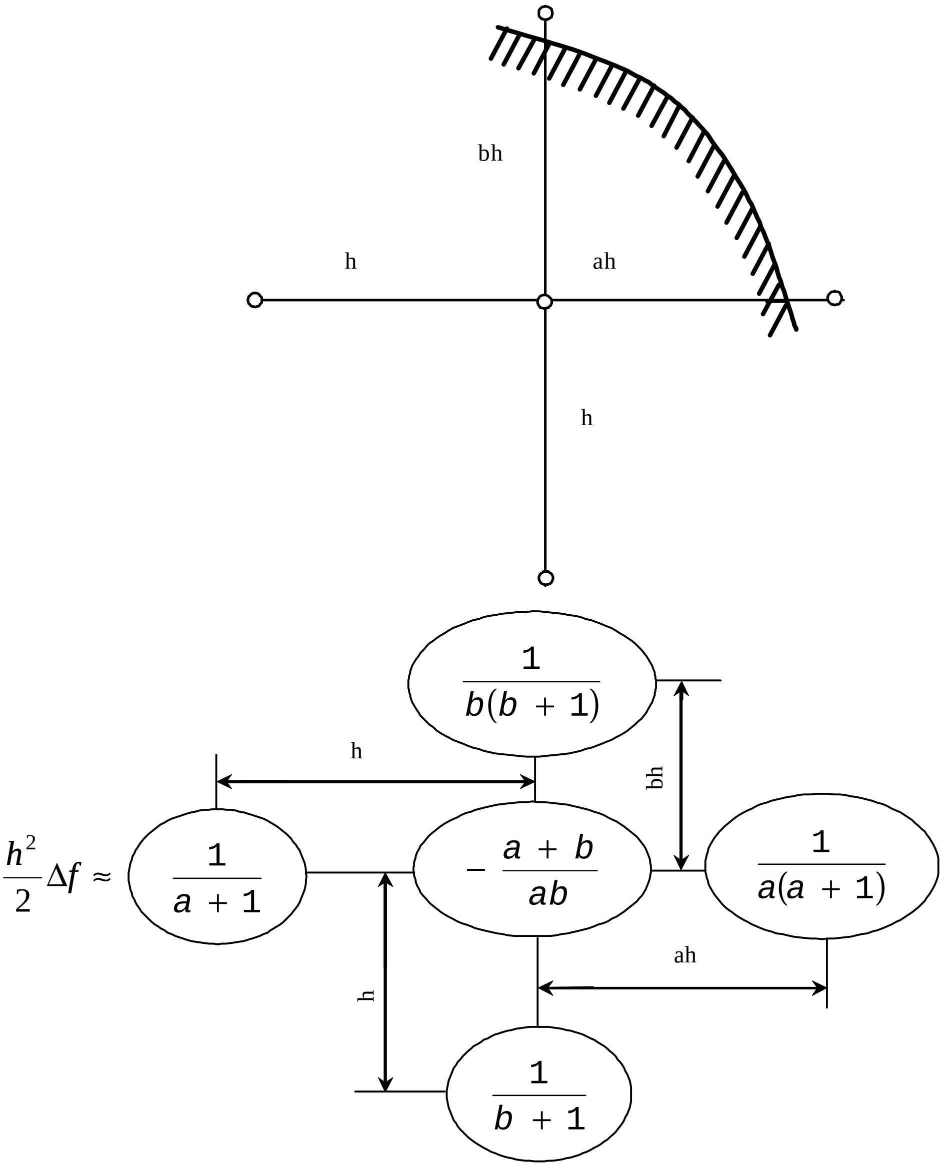


Figure 5.4

After applying a calculation template to each of n knots of the net, we will get the system of n equations, which can be linear, if initial differential equation has the proper structure. In this case the solution of the problem becomes the solution of the following system of equations:

$$\begin{bmatrix} matrix \quad of \\ coefficien\ ts \end{bmatrix} \begin{bmatrix} vector - column \quad of \quad unknown \\ meanings \quad in \quad nets \end{bmatrix} = \begin{bmatrix} vector - column \quad of \\ free \quad members \quad of \quad matrix \end{bmatrix}.$$

5.3 Solution of Problems in Mathematical Physics

Practical methods and algorithms of solution of different forms of differential equations in partial derivatives have the characteristic features and require detailed consideration on the example of the more widespread problems.

5.3.1 Elliptic Equations

Many different physical problems could be described by the elliptic equations: division of electric tensions on a plane that conducts a current; problem about the stationary flows of heat in a limited three-dimensional body etc. Often there emerges the necessity to solve such problems in the theory of automatic control. Most elliptic equations are described by Poisson equation or its special type – the Laplace equation.

We will consider the classic Dirichlet problem for the Laplace equation in a rectangular area which is formulated like this: to find the continuous function *f*(*x*,*y*), which satisfies into the rectangular area $\Omega = \{(x,y) \ / \ 0 \le x \le a, \ 0 \le y \le b\}$ of the Laplace equation:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

taking into account the boundary value:

$$\begin{aligned} x=0;&\quad f(0,y) = f_1(y), \\ x=a;&\quad f(a,y) = f_2(y), \\ y=0;&\quad f(x,0) = f_3(x), \\ y=b;&\quad f(x,b) = f_4(x). \end{aligned}$$

Let us enter in the area of solution a two-dimensional net with the step *h* on the axis *x and l* on the axis *y*. Then, using denotations from the previous chapter and approximating the Laplace equation by a difference equation, the following system of linear equations could be obtained (to simplify, suppose *l=h*):

$$\begin{gathered} f_{ij} = \frac{1}{4}(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}) \ , \\ f_{i,0} = f_3(x_i) \ , \quad f_{i,m} = f_4(x_i) \ , \quad f_{0,j} = f_1(y_i) \ , \quad f_{n,j} = f_2(y_i) \end{gathered} \qquad (5.2)$$

at *i=1, 2, ..., n-1; j=1, ..., m-1.*

This system of equations has plenty of zeroing elements and satisfies the terms of convergence at use of difference methods. Often the solution of such systems is found via the Gauss-Seidel method, which, when used to elliptic difference equations, is named the Libman method or method of successive displacements. The order of iterations can be traced, rewriting the system (5.2) in a form:

$$f_{1,1}^{(m+1)} = \frac{1}{4}\left[f_3(h) + f_1(h) + f_{2,1}^{(m)} + f_{1,2}^{(m)}\right],$$

$$f_{2,1}^{(m+1)} = \frac{1}{4}\left[f_3(2h) + f_{1,1}^{(m+1)} + f_{3,1}^{(m)} + f_{2,2}^{(m)}\right],$$

$$f_{3,1}^{(m+1)} = \frac{1}{4}\left[f_3(3h) + f_{2,1}^{(m+1)} + f_{4,1}^{(m)} + f_{3,2}^{(m)}\right],$$

$$f_{n-1,1}^{(m+1)} = \frac{1}{4}\left\{f_3[(n-1)h] + f_2(h) + f_{n-2,1}^{(m+1)} + f_{n-1,2}^{(m)}\right\},$$

$$f_{1,2}^{(m+1)} = \frac{1}{4}\left[f_1(2h) + f_{1,1}^{(m+1)} + f_{2,2}^{(m)} + f_{1,3}^{(m)}\right],$$

$$f_{2,2}^{(m+1)} = \frac{1}{4}\left[f_{2,1}^{(m+1)} + f_{1,2}^{(m+1)} + f_{3,2}^{(m)} + f_{2,3}^{(m)}\right],$$

----

where overhead indices mark the sequence number of iterations: *m* is the previous, *m+1* is the following.

Usually consider $f_{i,j}^{(0)} = 0$ for all *i* and *j*. Any elliptic equations which do not contain $\frac{\partial^2 f}{\partial x \partial y}$ are taken to the systems of iterative equations, which can be solved either by the Libman method or by other methods (Jacobi, successive overhead relaxation); as for them, the terms of convergence are executed. For elliptic equations which contain $\frac{\partial^2 f}{\partial x \partial y}$ in a general view, a problem of convergence of difference methods does not have the theoretical solution and it is necessary to examine the obtained system of equations in every separate case.

5.3.2. Hyperbolical Equations

The hyperbolical equations in partial derivatives are very often used in engineering practice. An example of such a problem is a wave equation which describes different types of vibrations: oscillation of string or membrane, distribution of sound-waves in different environments etc.

In a general form the problem can be formulated like this: to find the function f (x, t) which satisfies into the area Ω ={(x, t), *0 x ≤ and, 0 ≤ t ≤ T}* equation

$$\frac{\partial^2 f}{\partial t^2} = c^2 \frac{\partial^2 f}{\partial x^2} \qquad (c = const > 0),$$

with initial conditions

$$f(x,0) = f_0(x),$$

$$\frac{\partial f}{\partial t}\Big|_{x,0} = g(x)$$

and boundary values

$$f(0,t) = m_1(t),$$

$$f(a,t) = m_2(t).$$

As replacement of variables of t =ct brings equation over to the form:

$$\frac{\partial^2 f}{\partial t^2} = \frac{\partial^2 f}{\partial x^2},$$

in future we adopt *c=1*.

Passing to difference equation on a net with the step h on x and τ on t with central differences, we will get

$$\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\tau^2}.$$

If to enter $r = \frac{\tau}{h}$, the correlation for $f_{i,j+1}$ will be the following:

$$f_{i,j+1} = r^2(f_{i+1,j} + f_{i-1,j}) + 2(1-r^2)f_{i,j} - f_{i,j-1}. \qquad (5.3)$$

The chart of solution in accordance with equation (5.3) is named three-shares, as it links the value $f_{i,j}$ on three shares *j-1, j, j+1*. This chart is obvious and it allows to express $f_{i,j}$ through the value of f from previous shares (there are non-obvious charts based on the use of other calculation templates, but they require a lot of calculations in the course of the system's solution). To find the solution on the first share interpolations methods are usually used. For example:

$$f_{i,1} = f_{i,0} + \tau g(x_i). \tag{5.4}$$

Correlation of the net's sides concerns the size of *r*, which defines the firmness of the solution. At *r>1* the solution is unstable, at *r<1* it is stable, but exactness of it decreases with diminishing of *r*, at *r=1* the difference solution is stable and coincides with the exact one. The choice of *r=1 is* comfortable and allows to simplify the correlation (5.4)

$$f_{i,j+1} = f_{i+1,j} + f_{i-1,j} - f_{i,j-1}.$$

5.3.3 Parabolic Equations

In the example of a problem which results in parabolic equation in partial derivatives, there is a problem concerning heat-transfer on a long bar. It is described by equation of heat-transfer (or diffusions).

The problem consists of finding *f* (*x, t*), which satisfies in the area $\Omega = \{(x, t)\ 0 \le x \le a,\ 0 \le t \le T\}$ equation

$$\frac{\partial f}{\partial t} = k\frac{\partial^2 f}{\partial x^2} \quad (k = const > 0),$$

initial term

$$f(x, 0) = f_0(x)$$

and boundary values

$$f(0,t) = m_1(t) \ ,$$

$$f(a,t) = m_2(t) \ .$$

Replacement of variables *t =k t* brings equation to the form

$$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2},$$

for simplicity in future we will assume that *k=1*.

Two variants of getting the difference equation are possible on a net with a step *h* on *x* and τ on *t* (Figure 5.5).

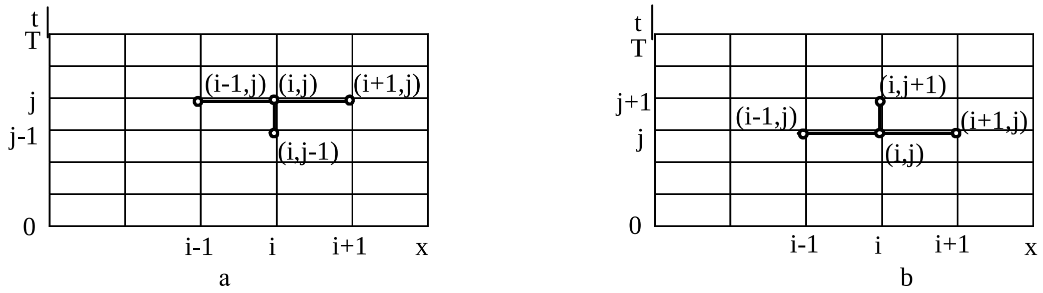


Figure 5.5

The variant with approximation on a four-nodes template (Figure 5.5, a) results in a non-obvious two shares difference chart

$$2f_{i+1,j} - (1+2r)f_{i,j} + 2f_{i-1,j} = -f_{i,j-1},$$

where $r = \dfrac{kt}{h^2}$.

This chart is complemented by the equations obtained from the boundary terms

$$f_{0,j} = m_1(t_j); \qquad f_{n,j} = m_2(t_j),$$

that brings the problem to the solution of the system of equations which have the stable solution regardless of the values of *r*.

The variant with approximation on a four-nodes template (Figure 5.5, b) results in the obvious two shares system

$$f_{i,j+1} = rf_{i+1,j} + (1-2r)f_{i,j} + 2f_{i-1,j}.$$

This chart is stable only at $r \le 0{,}5$, that results in a necessity to conduct calculations with a very small step on t, which limits the fast-acting and requires the greater charges of computer time. That is why for parabolic equations a more wide distribution received a non-obvious chart.

5.4 Remarks

The method of the problems' solution has to be chosen at the initial stage. The developers usually prefer the difference method, but in a number of cases for problems with well-developed theory (for example, problems of mechanics), it is appropriate to use the method of finite elements.

In this chapter the difference methods are considered. It is vital to be as exact as possible while determining the method of solution. In the difference methods there is an error of the second order. To estimate it, one could use ordinary differential equations in accordance with Runge method (Richardson extrapolation). In case of symmetry in the solution's area, it is possible to decrease the number of knots two or even four times (the symmetry should be on both axes of the co-ordinates). It allows saving time and decreasing the amount of calculations.

To solve the problem effectively, one should carefully choose the correct initial meanings. The speed of convergence in the course of application of the difference methods depends on it. Often, while solving the problem, a few stages should be passed: at the first stage the correct initial approach should be chosen for a rough net, and at the following stage – a more exact solution for the fine net is used .

Exercises

1. Give examples of the engineering problems leading to the differential equations in partial derivatives. Why are the problems with the partial derivatives called the problems of mathematical physics?
2. What types of equations in partial derivatives depend on their coefficients?
3. What are the stages of performing the difference method that are used to solve the partial derivatives equations?
4. Give solution of the Dirichlet problem for the Laplace equation.
5. Give solution of one type of the wave equations.
6. Give difference templates for the 1, 2, 3, 4 partial derivatives, the Laplace and biharmonic operators.
7. What are the ways to choose the correct method of solution and how could it be estimated?

Chapter 6. Experimental Data Processing

6.1 Introduction

The data processing considered in this chapter (procedures of interpolation, approximation, statistical data processing is frequently used in the problems of the computer control systems development.

6.2 Interpolation

The purpose of interpolation – which adopts the function in separate points $x_i[a,b]$ $(i = 0, 1, 2, ..., n)$ (knots of interpolation) – is to get the values:

$$F(x_0) = y_0,\ F(x_1) = y_1,\ ..., F(x_i) = y_i,\ ..., F(x_n) = y_n. \qquad (6.1)$$

These values coincide with the previously set values in the points of unknown function $y = f(x)$. It means geometrically that we need to find the curve $y = F(x)$ of certain type which crosses the system of points $M(x_i, y_i)$ (i = 0, 1, 2..., n) (Figure 6.1).

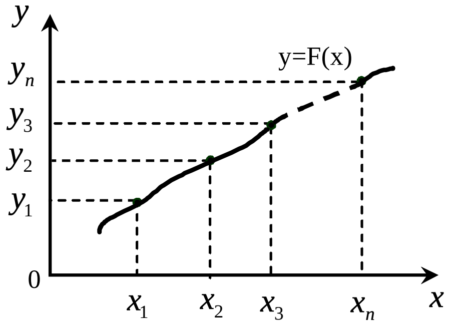


Figure 6.1

Generally, this problem has an endless number of solutions or does not have a solution at all, but it becomes one-valued when instead of the arbitrary function $F(x)$ the polynomial $P_n(x)$ of degree not higher than n, which satisfies condition (6.1) is sought, that is

$$P_n(x_0) = y_0,\ P_n(x_1) = y_1\ ,...,\ P_n(x_i) = y_i\ ,...,\ P_n(x_n) = y_n.$$

As a rule, the interpolation formula $y = F(x)$ is used for precise calculation of the values of unknown function $f(x)$ for $x \neq x_i$ $(i = 0, 1, 2, ..., n)$. Such an operation is called interpolation. One should keep in mind that interpolation is used when $x \in [x_0, x_n]$ and extrapolation - when $x \notin [x_0, x_n]$, where $x < x_0$ or $x > x_n$.

Let’s consider a few methods of interpolation.

6.2.1 Difference Methods

There are many difference methods of interpolation. The Newton method is more frequently used for the so-called "ahead" interpolation. An interpolation polynomial in this case will be:

$$P_n(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + \ldots$$
$$\ldots + C_n(x - x_0)(x - x_1)\ \ldots\ (x - x_{n-1}).$$

Coefficients C_i could be found from equations:

$$P_n(x_i) = y_i, \qquad i = 0, 1, 2, \ldots, n,$$

that allow to write the system in the following form

$$\begin{array}{l} C_0 = y_0, \\ C_0 + C_1(x_1 - x_0) = y_0, \\ C_0 + C_1(x_2 - x_0) + C_2(x_2 - x_0)(x_2 - x_1) = y_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ C_0 + C_1(x_n - x_0) + \ldots + C_n(x_n - x_0)(x_n - x_1)\ldots(x_n - x_{n-1}) = y_n. \end{array} \qquad (6.2)$$

This is a linear system of equations with a three-cornered matrix.

If we adopt step $x_{i+1} - x_i = h$ in the area where $x \in [x_0, x_n]$, we will get the difference correlations for the system’s (6.2) coefficients:

$$C_0 = y_0,$$

$$C_1 = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h},$$

Δy_0 - right first order difference in the point y_0;

$$C_2 = \frac{y_2 - 2y_1 + y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2},$$

$\Delta^2 y_0$ - right second order difference;

. . . .

$$C_j = \frac{\Delta^j y_0}{(j!)h^j},$$

$\Delta^j y_0$ - right j order difference.

Then

$$P_n(x) = y_0 + \frac{\Delta y_0}{1!h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1) + ...$$
$$... + \frac{\Delta^n y_0}{n!h^n}(x - x_0)(x - x_1)\ ...\ (x - x_n). \tag{6.3}$$

From practical point of view the following expression is used to determine the differences of higher orders:

$$\Delta^j y_i = \Delta(\Delta^{j-1} y_i) = \Delta^{j-1} y_{i+1} - \Delta^{j-1} y_i.$$

If $n = 1$ with (6.3) we get a formula for linear interpolation

$$P_1(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0),$$

and if $n = 2$ – formula of parabolic or quadratic interpolation

$$P_2(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2h^2}(x - x_0)(x - x_1).$$

Practically, in this case n is chosen so that the difference $\Delta^n y_i$ is permanent with the determined exactness. For the initial value x_0 it is possible to take any tabular value of argument x. When the amount of values of function is finite, amount n is limited and it couldn't be greater than the amount of the function's values minus one unit.

Formula (6.3) is called the first Newton interpolation formula. This correlation is not convenient for interpolation near-by the last values y. In this case, as a rule, the second Newton interpolation formula is used, which could be obtained from the left differences of the last value (x_n, y_n) ("back" interpolation). Then the interpolation polynomial looks like:

$$P_n(x) = C_0 + C_1(x - x_n) + C_2(x - x_n)(x - x_{n-1}) +$$

$$+C_3(x-x_n)(x-x_{n-1})(x-x_{n-2})+\ \ldots$$
$$\ldots\ +C_n(x-x_n)(x-x_{n-1})\ \ldots\ (x-x_1).$$

Coefficients C_j are:

$$C_0 = y_n,$$

$$C_1 = \frac{\Delta y_{n-1}}{h} = \frac{\nabla y_n}{h},$$

∇y_n - left first order difference in the point *n*

$$C_2 = \frac{\Delta^2 y_{n-2}}{2!h^2} = \frac{\nabla^2 y_n}{2h^2},$$

$\nabla^2 y_n$ - left difference of the second order

$$\ldots\ldots$$

$$C_j = \frac{\Delta^j y_{n-j}}{j!h^j} = \frac{\nabla^j y_n}{j!h^j},$$

$\nabla^j y_n$ - left difference of j order.

The final expression for the second Newton interpolation formula is:

$$P_n(x) = y_n + \frac{\Delta y_{n-1}}{1!h}(x-x_n) + \frac{\Delta y_{n-2}^2}{2!h^2}(x-x_n)(x-x_{n-1}) + \ldots$$
$$+\frac{\Delta^n y_0}{n!h^n}(x-x_n)(x-x_{n-1})\ldots(x-x_1).$$

The Newton interpolation formulas can be used for extrapolation also. If $x < x_0$, one could use the first Newton interpolation formula, thus:

$$\frac{x-x_0}{h} < 0.$$

If $x > x_n$, the second Newton interpolation formula could be used, so:

$$\frac{x-x_n}{h} > 0.$$

Thus, the first Newton interpolation formula, as a rule, is used for “ahead” interpolation and “back” extrapolation, and second - for the “back” interpolation and the “ahead” extrapolation.

In the Newton formulas left and right differences are used. The use of central differences in order to get interpolation formulas results in Gauss, Sterling and Bessel formulas.

We can consider these formulas on $(2n+1)$ equidistant knots of interpolation

$$x_{-n}, x_{-(n-1)}, \dots, x_{-1}, x_0, x_1, \dots, x_{n-1}, x_n,$$

thus

$$\Delta x_i = x_{i+1} - x_i = h \qquad (i = -n, -(n-1), \dots, n-1),$$

and for function $y = f(x)$ the values are known in these knots: $y_i = f(x_i)$.

We need to find the polynomial $P(x)$ of degree not higher than $2n$ so that:

$$P(x_i) = y_i.$$

Polynomial $P(x)$ could be found in the following way:

$$\begin{gathered} P(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + C_3(x - x_{-1}) \\ (x - x_0)(x - x_1) + \dots + C_{2n-1}(x - x_{-(n-1)})\dots \\ \dots(x - x_{-1})(x - x_0)(x - x_1)\dots(x - x_{n-1})(x - x_n). \end{gathered} \tag{6.4}$$

Similar to the Newton interpolation formulas:

$$C_0 = y_0;$$

$$C_1 = \frac{\Delta y_0}{h};$$

$$C_2 = \frac{\Delta^2 y_{-1}}{2! h^2};$$

. . . . . .

$$C_{2n-1} = \frac{\Delta^{2n-1} y_{-(n-1)}}{(2n-1)!h^{2n-1}};$$

$$C_{2n} = \frac{\Delta^{2n} y_{-n}}{(2n)!h^{2n}}.$$

These coefficients form the first Gauss interpolation formula, which contains differences (Table 6.1):

$$\Delta y_0,\ \Delta^2 y_{-1},\ \Delta^3 y_{-1},\ \Delta^4 y_{-2},\ \Delta^5 y_{-2},\ \Delta^6 y_{-2},\ \ldots.$$

Similarly, it is possible to get the second Gauss interpolation formula, which contains central differences:

$$\Delta y_{-1},\ \Delta^2 y_{-1},\ \Delta^3 y_{-2},\ \Delta^4 y_{-2},\ \Delta^5 y_{-3},\ \Delta^6 y_{-3},\ \ldots.$$

Using middle arithmetic value of the first and second Gauss interpolation formulas, we can get the Sterling formula. In general, it is appropriate to use the interpolation formulas with central differences in the middle of interval, while on its edges, as a rule, the Newton formulas should be used. Application of these formulas is given in Table 6.1.

Errors of interpolation for the Newton formulas could be estimated in the following way for the first and second formulas accordingly:

$$\Delta_n(x) = \frac{q(q-1)\ldots(q-n)}{(n+1)!}\Delta^{n+1} y_0,$$

$$\Delta_n(x) = \frac{q(q+1)\ldots(q+n)}{(n+1)!}\Delta^{n+1} y_{-n},$$

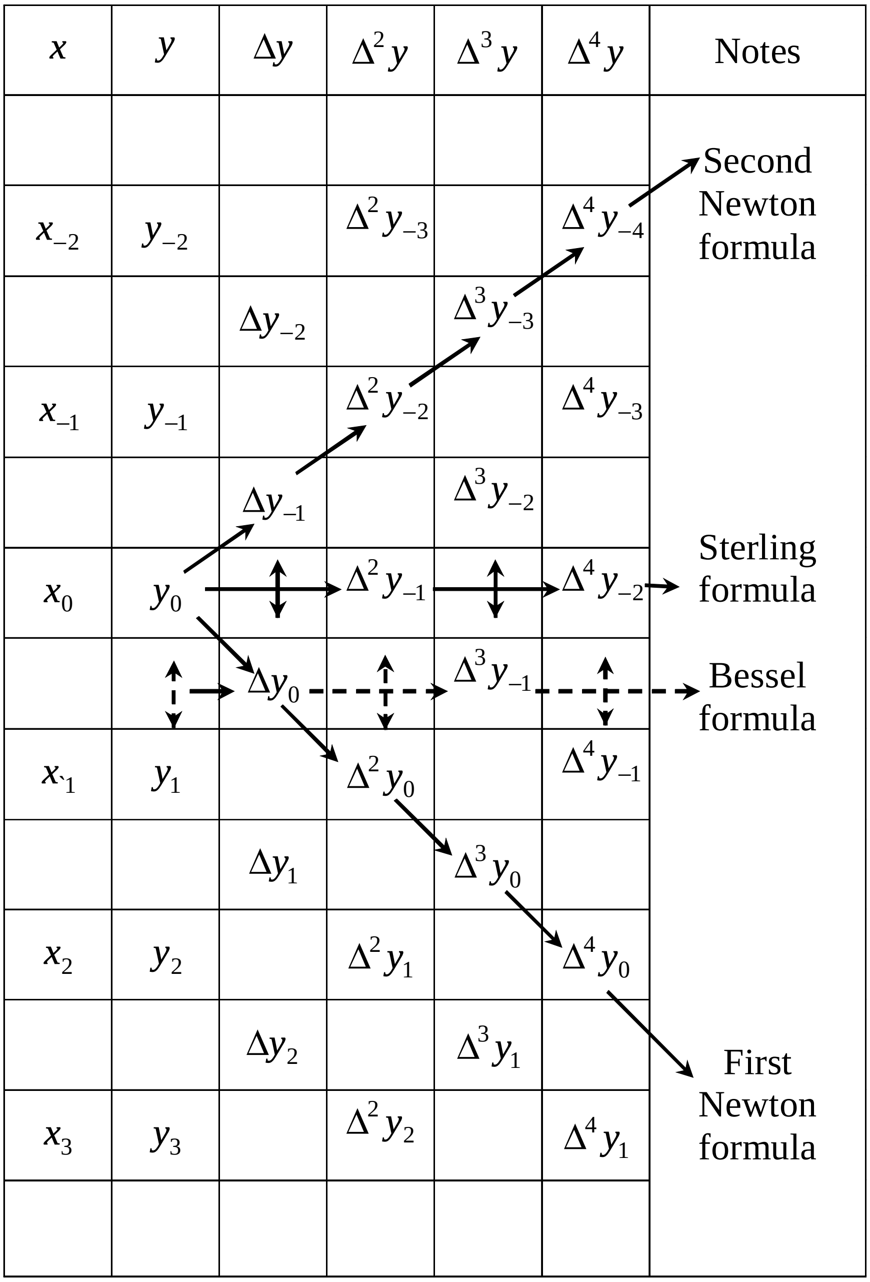
where $q = \frac{x - x_n}{h}$.

For the Sterling formula:

$$\Delta(x_n) = \frac{\Delta^{2n+1} y_{-(n-1)} + \Delta^{2n+1} y_{-n}}{2(2n+1)!} q(q^2-1)(q^2-2^2)\ \ldots\ (q^2-n^2).$$

Table 6.1

Differences interpolation methods



x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	Notes
x_{-2}	y_{-2}		$\Delta^2 y_{-3}$		$\Delta^4 y_{-4}$	Second Newton formula
		Δy_{-2}		$\Delta^3 y_{-3}$		
x_{-1}	y_{-1}		$\Delta^2 y_{-2}$		$\Delta^4 y_{-3}$	
		Δy_{-1}		$\Delta^3 y_{-2}$		
x_0	y_0		$\Delta^2 y_{-1}$		$\Delta^4 y_{-2}$	Sterling formula
		Δy_0		$\Delta^3 y_{-1}$		Bessel formula
x_1	y_1		$\Delta^2 y_0$		$\Delta^4 y_{-1}$	
		Δy_1		$\Delta^3 y_0$		
x_2	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$	
		Δy_2		$\Delta^3 y_1$		First Newton formula
x_3	y_3		$\Delta^2 y_2$		$\Delta^4 y_1$	

In case of unequidistant values of the argument it is possible to get interpolation formulas, using determination of the divided differences. For example, relation

$$[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

is called the divided difference of the first order, and relation

$$[x_i, x_{i+1}, x_{i+2}] = \frac{[x_{i+1}, x_{i+2}] \; - \; [x_i, x_{i+1}]}{x_{i+2} - x_i}$$

– the divided difference of the second order.

The divided differences of order *n* could be obtained from the recurrent relation:

$$[x_i, x_{i+2}, \; ..., \; x_{i+n}] = \frac{[x_{i+1},..., x_{i+n}] \; - \; [x_{i+1},..., x_{i+n-1}]}{x_{i+n} - x_i}.$$

It is possible to get the Newton interpolation formula for the unequidistant values of argument:

$$P(x) = y_0 + [x_0, x_1](x - x_0) + [x_0, x_1, x_2](x - x_0)(x - x_1) + \; ...$$
$$+ [x_0, x_1, ..., x_n](x - x_0)(x - x_1) \; ... \; (x - x_{n-1}).$$

6.2.2 Lagrange Interpolation

Lagrange interpolation is used generally for the arbitrarily located knots.

An interpolation polynomial for the Lagrange method could be given in a form:

$$P_n(x) = y_0 b_0(x) + y_1 b_1(x) + \; ... \; + y_n b_n(x),$$

where all $b_j(x)$ (*j=0,..., n*) are polynomials of degree *n*, the coefficients of which could be found via (*n+1*) equations:

$$P_n(x_i) = y_i,$$

as a result we will get the system of equations:

$$y_0 b_0(x_0) + y_1 b_1(x_0) + \; ... \; + y_n b_n(x_0) = y_0;$$
$$..................$$

$$y_0b_0(x_n)+y_1b_1(x_n)+\ ...\ +y_nb_n(x_n)=y_n.$$

If the value $b_j(x_i)$ is picked so, that

$$b_j(x_i)=\left|\begin{array}{l}1,\ \ \text{i = j},\\ 0,\ \ \text{i} \neq \text{j},\end{array}\right.$$

the previous system of equations is sufficient.

This condition means that any polynomial $b_j(x)$ equals to zero at each point *x*, except for x_j. That is why in general case a polynomial $b_j(x)$ is:

$$b_j(x)=C_j(x-x_0)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_n).$$

If $b_j(x_j)$=1, coefficients C_j could be found from correlation:

$$C_0=1/(x_j-x_0)...(x_j-x_{j-1})(x_j-x_{j+1})...(x_j-x_n).$$

We get the following polynomial:

$$P_n(x)=\sum_{j=0}^{n}y_j\ *$$
$$*\frac{(x-x_0)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_n)}{(x_j-x_0)(x_j-x_1)...(x_j-x_{j-1})(x_j-x_{j+1})...(x_j-x_n)}.$$

Entering denotation

$$L_j(x)=(x-x_0)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_n),$$

we get the following formula:

$$P_n(x)=\sum_{j=0}^{n}y_j\frac{L_j(x)}{L_j(x_j)}.$$

One should point out two main properties of the Lagrange polynomials:

1) $\sum_{j=0}^{n}\frac{L_j(x)}{L_j(x_j)}=1;$

2) if $P_n(x)$ linear depends on y_j, a suitable principle of superposition is the following: the interpolation polynomial of the sum of a few functions equals to the sum of interpolation polynomials of the elements.

An error at Lagrange interpolation can be calculated in this way:

$$|\Delta(x_n)| \leq \frac{M_{n+1}}{(n+1)!}(x-x_0)(x-x_1)...(x-x_n),$$

where $M_{n+1} = \max\limits_{x_0 \leq x \leq x_n} \left| f^{(n+1)}(x) \right|.$

6.2.3 Spline Interpolation

It is comparatively recently that splines started to be widely used in calculation methods. In the machine designing they have been used for quite a long time, because it were namely the French curves or flexible lines, that were transformed to make it possible to draw a curve through the multiplicity of points (x_i, y_i).

It is possible to show (using the theory of bend to the squared beam at small deformations) that a spline is a group of united cube polynomials, in the conjugacy points of which the first and second derivatives are equal. Such functions are called cube splines. To get them we need to set the coefficients which determine a polynomial on the interval between the two points.

For example, in the case presented in Figure 6.2 it is necessary to set all cube functions $q_1(x),\ q_2(x),...,q_m(x)$. In a more general case these polynomials are:

$$q_i(x) = k_{1i} + k_{2i}x + k_{3i}x^2 + k_{4i}x^3, \qquad i=1,2, ...,m$$

where k_{ji} - permanent, which is certain by the indicated conditions (*j*= *1,2,3,4*).

The first (*2m*) of the conditions demands that splines clash in certain points:

$$q_i(x_i) = y_i \qquad i=1,\ 2\ ...,\ m$$
$$q_{i+1}(x_i) = y_i \qquad i=0,\ 1,\ ...\ m-1.$$

Following *(2m-2)* condition demands that in the point of conjugacy the splines are on the levels of the first and of the second derivatives

$$q'_{i+1}(x_i) = q'_i(x_i), \qquad i=1, \dots m-1,$$
$$q''_{i+1}(x_i) = q''_i(x_i), \qquad i=1, \dots m-1.$$

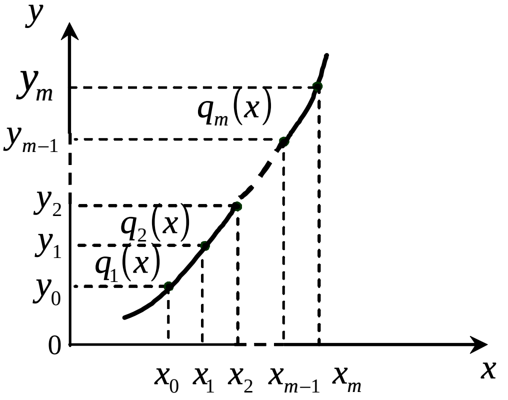


Figure 6.2

The system of algebraic equations has a solution, if the amount of equations equals to the amount of the unknowns. Two additional equations are needed. As a rule, the following additional conditions are used:

$$q''_1(x_0) = 0; \qquad q''_m(x_m) = 0.$$

The spline obtained is called a “natural cube spline”. At the coefficients of the spline the cobbed-smooth polynomial interpolation is used.

If we separately choose the type of cube polynomials, it is possible to simplify a problem considerably (to decrease the amount of equations). In case, when separate cube equations are in the following form:

$$q_i(x) = ty_i + \bar{t}y_{i-1} + \Delta x_i[(k_{i-1} - d_i)t^2\bar{t} - (k_i - d_i)t^2\bar{t}, \quad i = 1, 2, ..., m$$

where $\Delta x_i = x_i - x_{i-1}$, $t = \dfrac{x - x_{i-1}}{\Delta x_i}$, $\bar{t} = 1 - t$,

and $\Delta y_i = y_i - y_{i-1}$, $\dfrac{\Delta y_i}{\Delta x_i} = d_i$,

each of the equations $q_i(x)$ contains only two unknown coefficients. Since the first equation is recorded, with every next equation only one unknown coefficient is added. Thus, when $x = x_{i-1}$, $t = 0$, $\bar{t} = 1$, and at $x = x_i$ $t = 1$, $\bar{t} = 0$, Consequently, all conditions, except for the conditions for the second derivative, are satisfied. The second derivative is expressed for the internal points as following:

$$k_{i-1}\Delta x_{i+1} + 2k_i(\Delta x_i + \Delta x_{i+1}) + k_{i+1}\Delta x_i = 3(d_i\Delta x_{i+1} + d_{i+1}\Delta x_i),$$

and for two external points:

$$2k_0 + k_1 = 3d_1 \qquad \text{and} \qquad k_{m-1} + 2k_m = 3d_m .$$

Thus, the system of equations comes to its tridiagonal form:

$$\begin{bmatrix} 2 & 1 & 0 & 0 & \\ \Delta x_2 & 2(\Delta x_1 + \Delta x_2) & \Delta x_1 & \Delta x_2 & \\ 0 & \Delta x_3 & 2(\Delta x_2 + \Delta x_3) & 2(\Delta x_{m-1} + \Delta x_m) & \Delta x_{m-1} \\ & & \Delta x_m & 1 & 2 \end{bmatrix} *$$

$$* \begin{bmatrix} k_0 \\ k_1 \\ \cdot \\ \cdot \\ \cdot \\ k_m \end{bmatrix} = 3 \begin{bmatrix} d_1 \\ d_1\Delta x_2 + d_2\Delta x_1 \\ d_2\Delta x_3 + d_3\Delta x_2 \\ \ldots\ldots\ldots\ldots \\ d_{m-1}\Delta x_m + d_m\Delta x_{m-1} \\ d_m \end{bmatrix} .$$

The methods of solving such systems are well known.

In many cases the method of splines is more efficient and convenient, because it allows to get an analytical piecewise-polynomial function. There are splines of higher orders. It is also possible to use this method in the other fields of computational mathematics, for example, in numeral integration and while solving differential equations.

6.2.4 Tridimensional Space Interpolation

This problem is very common today in the three-dimensional designing at the delivery and processing of the real objects’ images, construction of surfaces and landscapes, in physics, astronomy, cosmonautics, medicine and other spheres.

Setting the problem. Let’s assume we have a function $f(x,\ y)$ and the row of its values is known:

$$f(x_0,\ y_0) = z_0\, ,\ f(x_1,\ y_1) = z_1\, ,\ f(x_2,\ y_2) = z_2\, ,\ \ldots\, ,\ f(x_n,\ y_n) = z_n\, .$$

We need to find a function, that traverses with function $f(x,\ y)$ in the knots of interpolation. In order to find the unique solution for the problem $F(x,\ y)$, it is supposed to be a polynomial of degree not higher than n.

To solve this problem the analogues of the interpolation methods, which exist for the function of one variable, are used.

The general formula for the Newton interpolation has the following polynomial form:

$$P_n(x,\ y) = C_0 + C_1[(x-x_0)+(y-y_0)] + C_2[(x-x_0)+(y-y_0)][(x-x_1)+(y-y_1)] + \\ + C_3[(x-x_0)+(y-y_0)][(x-x_1)+(y-y_1)][(x-x_2)+(y-y_2)] + \dots,$$

where for *i* from *0* to *n.*

Then

$$C_i = \frac{\Delta^i z_o}{i!(h+l)^i},$$

where *h* – step for *x*, *l* – step for *y*.

In a short form, in case of

$$q = \frac{x-x_o}{h},\ \ p = \frac{y-y_o}{l},$$

$$P_n(x,\ y) = z_0 + \frac{qh+pl}{h+l}\Delta z_0 + \dots + \frac{(qh+pl)[(q-1)h+(p-1)l]\dots[(q-n)h+(p-h)l]}{n!(h+l)^n}\Delta^n z_0$$

Error of the Newton method is:

$$R_n(x,\ y) = \frac{(qh+pe)[(q-1)h+(p-1)e]\dots[(q-h)h+(p-h)e]}{(n+1)!(h+1)^{h+1}}\Delta^{n+1} z_0.$$

We consider interpolation of space curves when the order of points is known. In case of a huge variety of points the problem could be much more complicated and especial two-dimensional differences are to be used.

In Lagrange method we seek for polynomial $L_n(x,y)$ of degree not higher n:

The values of coefficients b_i follow from the condition of coinciding with a function that is studied in the knots of interpolation $x_i\ y_i$:

$$f(x_i,\ y_i) = \sum_{k=0}^{n} b_k j_k(x_i,\ y_i),\ \ i = \overline{1\dots n},$$

where $j_i(x)$ are the fixed values of the function.

A polynomial will transform to the following form:

$$P_n(x,y) = b_0 j_0(x,y) + b_1 j_1(x,y) + ... + b_n j_n(x,y),$$

that $$\begin{cases} b_0 j_0(x_0,y_0) + b_1 j_1(x_0,y_0) + ... + b_n j_n(x_0,y_0) = z_0, \\ b_1 j_0(x_1,y_1) + b_1 j_1(x_1,y_1) + ... + b_n j_n(x_1,y_1) = z_1, \\ ... \\ b_n j_0(x_n,y_n) + b_1 j_1(x_n,y_n) + ... + b_n j_n(x_n,y_n) = z_n. \end{cases}$$

General Lagrange formula interpolation is analogical to the corresponding method of unidimensional interpolation:

$$P_n(x,y) = \sum_{i=0}^{n} z_j \frac{(x-x_0)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_n)}{(x_j-x_0)(x_j-x_1)...(x_j-x_{j-1})(x_j-x_{j+1})...(x_j-x_n)} \times$$
$$\times \frac{(y-y_0)(y-y_1)...(y-y_{j-1})(y-y_{j+1})...(y-y_n)}{(y_j-y_0)(y_j-y_1)...(y_j-y_{j-1})(y_j-y_{j+1})...(y_j-y_n)}.$$

Let’s assume that (x_0, y_0) ... (x_n, y_n) – $n+1$ are different points in space $[x_0 \le x \le x_n;\ y_0 \le y \le y_n)]$. There is a unique polynomial $P_n(x,y)$ of degree not higher than n, which has the following characteristics:

$f(x_i, y_i) = P_n(x_i.y_i)$ for $i = 0,1,..., n$.

Lagrange formula for the given polynomial looks like ($i = 0, 1, ..., n$):

$$P_n(x,y) = \sum_{i=0}^{n} z_j \frac{(x-x_0)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_n)}{(x_j-x_0)(x_j-x_1)...(x_j-x_{j-1})(x_j-x_{j+1})...(x_j-x_n)} \times$$
$$\times \frac{(y-y_0)(y-y_1)...(y-y_{j-1})(y-y_{j+1})...(y-y_n)}{(y_j-y_0)(y_j-y_1)...(y_j-y_{j-1})(y_j-y_{j+1})...(y_j-y_n)}.$$

In the course of the calculations, if a certain factor $(x-x_k)$ or $(y-y_k)$ meets several times, it should be counted only once.

For a case, when $x_j = x_k$ we will get:

$$P_n(x,y)=\sum_{j=0}^{n} z_j\,\frac{(x-x_0)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_{k-1})}{(x_j-x_0)(x_j-x_1)...(x_j-x_{j-1})(x_i-x_{j+1})...(x_i-x_{k-1})}\times$$
$$\times\frac{(x-x_{k+1})...(x-x_n)\cdot(y-y_0)(y-y_1)...(y-y_{j-1})}{(x_i-x_{k+1})...(x_j-x_n)\cdot(y_j-y_0)(y_j-y_1)...(y_j-y_{j-1})}$$
$$\times\frac{(y-y_{j+1})...(y-y_n)}{(y_j-y_{j+1})...(y_j-y_n)}.$$

At $y_j=y_k$:

$$P_n(x,y)=\sum_{j=0}^{n} z_j\,\frac{(x-x_0)(x-x_1)...(x-x_{j-1})(x-x_{j+1})...(x-x_n)}{(x_j-x_0)(x_j-x_1)...(x_j-x_{j-1})(x_j-x_{j+1})...(x_j-x_n)}\times$$
$$\times\frac{(y-y_0)(y-y_1)...(y-y_{j-1})(y-y_{j+1})...(y-y_{k-1})}{(y_j-y_0)(y_j-y_1)...(y_j-y_{j-1})(y_j-y_{j+1})...(y_j-y_{k-1})}$$
$$\times\frac{(y-y_{k+1})...(y-y_n)}{(y_j-y_{k+1})...(y_j-y_n)}.$$

We will enter the following denotations:

$$L_j^{(n)}(x,y)=(x-x_0)(x-x_1)...(x-x_{j-1})(x+x_{j+1})...(x-x_n)\times$$
$$(y-y_0)(y-y_1)...(y-y_{j-1})(y+y_{j+1})...(y-y_n).$$

Then general interpolation formula will be:

$$P_n(x,y)=\sum_{j=0}^{n} z_j\,\frac{L_j^{\,n}(x,\ y)}{L_j^{\,n}(x_j,y_j)}.$$

In more compact way it could be recorded as follows:

$$L_j^{(n)}(x,y)=\frac{\mathit{\Pi}_{n+1}(x)\cdot\mathit{\Pi}_{n+1}(y)}{(x-x_j)(y-y_j)},$$

where $\ddot{I}_{\ n+1}(x)=(x-x_0)...(x-x_n)$, $\ddot{I}_{\ n+1}(y)=(y-y_0)...(y-y_n)$.

The properties of Lagrange interpolation:

a) $\displaystyle\sum_{j=0}^{n}\frac{L_j(x,y)}{L_j(x_j,y_j)}=1.$

b) Actual principle of superposition:

$P_n(x,y) = P_n(z)$ - a linear function from z. Thus, the sum of Lagrange polynomial of several functions equals to the sum of polynomial of the components.

We will consider the example of the method's application. Suppose two points are set in space: $z_1(x_1, y_1)$. The expression for a linear interpolation is:

$$P_n(x,y) = z_0 \frac{(x-x_1)(y-y_1)}{(y_0-y_1)(x_0-x_1)} + z_1 \frac{(x-x_0)(y-y_0)}{(y_1-y_0)(x_1-x_0)} =$$
$$= \frac{z_0(x-x_1)(y-y_1)+z_1(x-x_0)(y-y_0)}{(x_0-x_1)(y_0-y_1)}.$$

If $x_0 = x_1$

$$P_n(x,y) = z_0 \frac{(y-y_1)}{(y_0-y_1)} + z_1 \frac{(y-y_0)}{(y_1-y_0)} = \frac{z_0(y-y_1)+z_1(y-y_0)}{(y_0-y_1)}.$$

Also $\quad P_n(x,y) = z_0 \frac{(x-x_1)}{(x_0-x_1)} + z_1 \frac{(x-x_0)}{(x_1-x_0)} = \frac{z_0(x-x_1)+z_1(x-x_0)}{(x_0-x_1)}$

If only $y_0 = y_1$.

The error of the method of interpolation is calculated using the formula:

$$R_n(x,\ y) = f(x,y) - L_n(x,y) = \frac{f^{(n+1)}(x_x, x_y)}{((n+1)!)^2} \prod\nolimits_{n+1}(x) \prod\nolimits_{n+1}(y)\ ,$$

where x_x, x_y - depend on x, y and lie within the limits of the set space.

6.2.5 Interpolations by Selfsimilar Transformations

A lot of objects possess a property of selfsimilarity or geometrical invariance to the spatial scale. If we examine these objects according to a certain scale, their similar fundamental elements will constantly appear. Quite often it is possible to see that mountain, coast, cloud, tree and other objects have a similar structure. Prevalence of selfsimilar structures in nature is really impressive. Selfsimilar are minerals and mountain breeds; locations of branches, patterns of letters, capillary system of plants; nervous, lymphatic and other systems in organisms of animals and humans; rivers, clouds, lines of seashore, mountain relieves and so on. Such objects mathematically cannot be described via simple functions.

The selfsimilar structure considers a fractal, which is a recourse model, every part of which repeats the development of the whole model.

The property of parts to be similar to the whole structure is called selfsimilarity. Selfsimilarity assumes that the printing-down, down-scaling, variations of some "standard" form allow nature to create a complex multiscale structure easily.

Hierarchical character is another important property of fractals that allows to repeat itself in different dimensions of space and time.

The classic methods of interpolation do not allow conducting interpolation effectively when operating with badly differentiated functions. Another limitation of this method is the fact that only separate points of space, not an accumulation of points could be considered as knots of interpolation. That is why for many practical problems interpolation is conducted using selfsimilar multitudes.

A selfsimilar multitude (Figure 6.3) is a multitude $X \in E_2$ (in two-dimensional space) or $X \in E_3$ (in three-dimensional space), that can be presented as an accumulation of finite amount of submultitude

$$X = \bigcup_{i=1}^{n} X_i \ . \qquad (1.2)$$

The next conditions should be followed:

1) $X_i \in E_2$ $(X_i \in E_3)$, $i = 1,...,n$;

2) multitudes x_i *and* x_j, $i,\ j = 1,...,n$ $i <> j$ in pairs do not block each other;

3) $X_i = c_i(X)$ $i = 1, ..., n$ where c_i is a transformation of similarity with the coefficient of homothety $0 < s_i < 1$.

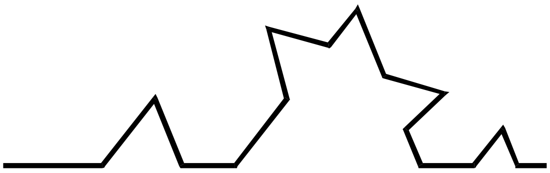


Figure 6.3

The strictly selfsimilar multitude (Figure 6.4), is a selfsimilar multitude, where all the transformations of similarity, $i = 1, ..., n$ have identical coefficients of homothety $0 < s < 1$.

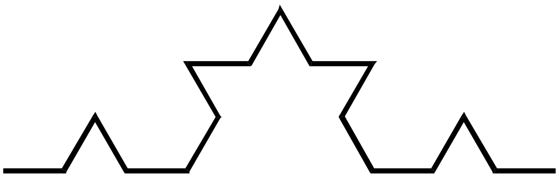


Figure 6.4

The incorporated selfsimilar multitude (Figure 6.5) is a multitude that can be represented as an accumulation of strictly selfsimilar multitudes.

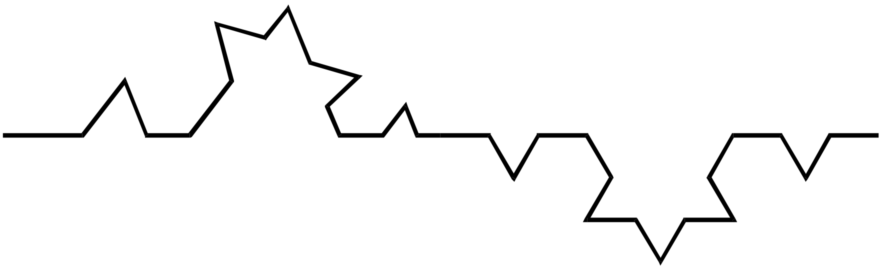


Figure 6.5

The selfaffine multitude (Figure 6.6) is a multitude (not empty), that can be represented as an accumulation of the complete amount of subset

$$X = \bigcup_{i=1}^{n} X_i .$$

The following conditions are to be followed:

1) $X_i \in E_2 \;\; i = 1,...,n$;

2) Multitudes $x_i \; and \; X_j$, $j = 1,...,n$, $i <> j$ in pairs do not block each other;

3) $X_i = w_i(X)$ where w_i are affine transformations with the coefficient of homothety $0 < s_i < 1$ and aspect $0 < q_i \le 1$ ratio in relation to one of the axis.

Accordingly distinguished is a strictly selfaffine multitude (Figure 6.7) for a case, where all affine transformations have the identical coefficient of homothety and united selfaffine multitude (Figure 6.8), which is a union of strictly selfaffine multitudes.

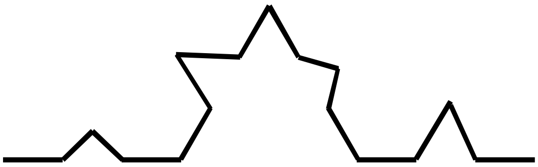


Figure 6.6

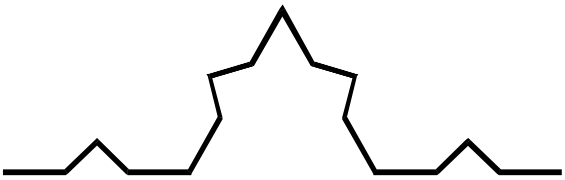


Figure 6.7

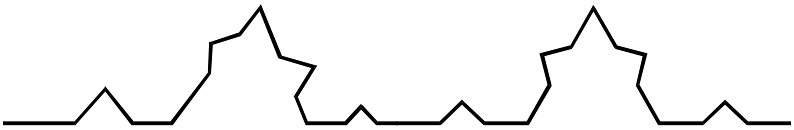


Figure 6.8

The class of selfsimilar multitudes is a part of the class of selfaffine multitudes.

Suppose $A_0 = \{a_o,...,a_n\}$, $n \geq 3$ is a certain well-organized complete multitude of three-dimensional Euclid space E_3, and there is a Hatchinson operator which satisfies the conditions:

1) sequence of multitudes

$$A_1 = W(A_0)\; A_n = W(A_{n-1})$$

is a convergence in accordance with Hausdorf metric,

2) A_0 belongs to the multitude

$$A_0 \subset W^j(A_0)\; j = 1,2,....$$

after j iterations.

The Hatchinson multitude interpolation with range j ($\overline{A}_j$) is called a continuous curve that traverses all the points of ordered multitude $A_j = W^j(A_0)$, thus a_0 is connected with a_1, a_1 is connected with a_2, . . . , a_{n-1} is connected with a_n.

Consequently, the problem of interpolation consists in finding the Hatchinson operator W that satisfies terms 1-2 and in the construction of interpolation of Hatchinson multitude $\overline{A}_j$ range j.

Let $A = \{a_o,...,a_n\}$, $n \geq 3$ is a complete ordered multitude of points of three-dimensional Euclid space E_3.

Let's enter the concept $s\,j\,t$ of relations, which will be calculated at every iteration for every element:

$$s_0 = \frac{d(a_0,a_1)}{d(a_0,a_n)}, \qquad {}_0 = (a_1 - a_0)\wedge(a_n - a_0), \qquad t_0 = 0;$$

$$s_1 = \frac{d(a_1,a_2)}{d(a_0,a_n)}, \qquad j_1 = (a_2 - a_1)\wedge(a_n - a_0), \qquad t_1 = a_1 - a_0;$$

$$s_2 = \frac{d(a_2,a_3)}{d(a_0,a_n)}, \qquad j_2 = (a_3 - a_2)\wedge(a_n - a_0), \qquad t_2 = a_2 - a_0;$$

. . . .

$$s_i = \frac{d(a_i,a_{i+1})}{d(a_0,a_n)}, \qquad j_i = (a_{i+1} - a_i)\wedge(a_n - a_0), \qquad t_i = a_i - a_0;$$

. . . . .

$$s_{n-1} = \frac{d(a_{n-1},a_n)}{d(a_0,a_n)},$$

where $(a_{i+1} - a_i) \wedge (a_n - a_0)$ is a corner between vectors $(a_{i+1} - a_i)$ and $(a_n - a_0)$. (This corner will be calculated in plane XOY and XOZ).

On the multitude of elements A the following limitations are imposed:

1) $s_i < 1,\ i = 0, ..., n-1$;

2) $\sum_{i=0}^{n-1} s_i^{\ D} = 1$;

3) $1 < D < 2$;

where D is selfsimilar dimension of multitude A.

According to the known Banah theorem about an immobile point, the error of the method of interpolation using selfsimilar multitudes could be estimated via the formula:

$$\Delta = \frac{E^{m+1}}{1-E} \cdot D;$$

where $E = \max\{s_i\}$,

$D = \max\{d(a_{i+1} - a_i)\},\ \ i = 0, ..., n-1.$

For every segment of the figure the following order of actions is set:

1. The coefficient of downscaling is calculated (Figure 6.9):

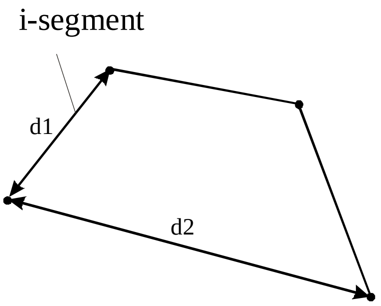
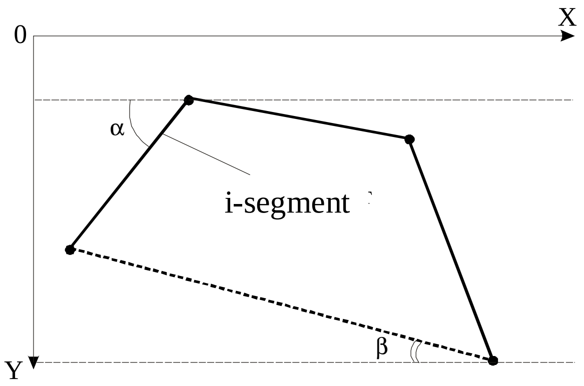


Figure 6.9

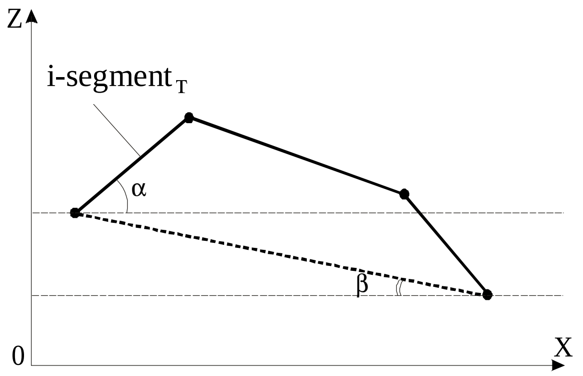
2. The corner of rotation is calculated in plane XOY (Figure 6.10):

3. The corner of rotation is calculated in plane XOZ (Figure 6.11):



$j_{xoy} = a - b$;

Figure 6.10



Ç2

$j_{xoz} = a - b$;

Figure 6.11

4. The rotation of the regenerated initial broken line is executed and its position changes (figure 6.12):

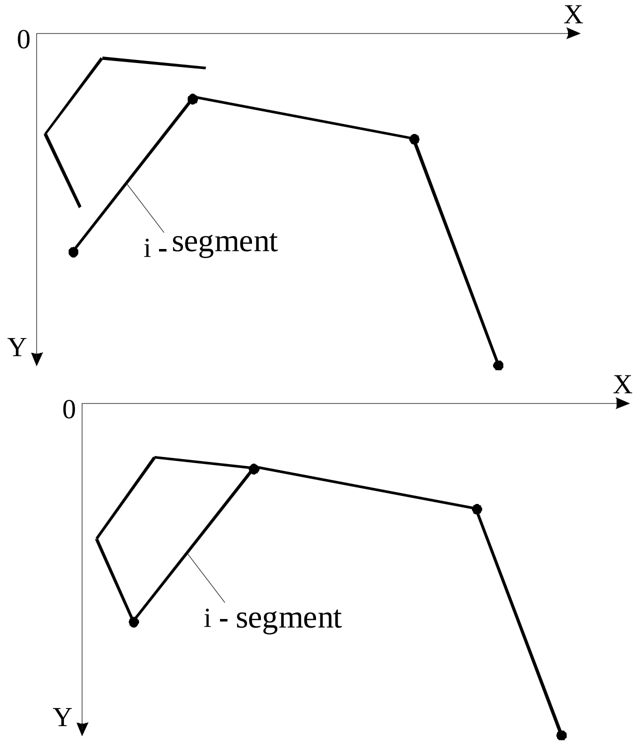


Figure 6.12

In figure 6.13 the image of trees built of selfsimilar multitudes (fractals) is given.

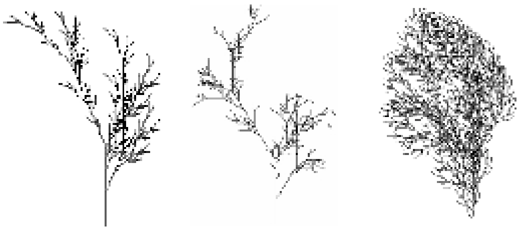


Figure 6.13

Interpolation of the functions of two variables using selfsimilar multitudes $z = f(x, y)$ could be conducted on every variable *x* and *y* consistently.

The interpolation of the function of one variable *x*:

$$f_x(x) = f(x, y_k),$$

where, $y_k = y$. Further, examining the values obtained $f_x(\bar{x}) = f(\bar{x}, y_j)$ as values of function $f(\bar{x}, y)$ of one variable *y*, by the method of interpolation using selfsimilar multitudes we find the value $f(\bar{x},\ \bar{y}) = \bar{z}$.

Example of construction of landscape of the river by selfsimilar transformations is represented in Figure 6.14

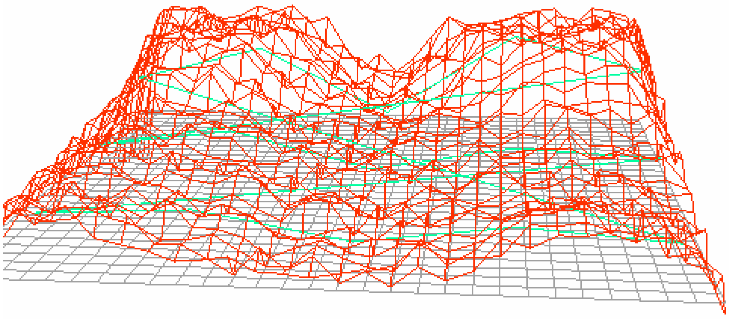


Figure 6.14

6.3 Approximation

Approximation in general is a close description by one function (approximate) of a definite type of another function (approximated) which is set in different form, for example, data file.

There are two main approaches to approximation of data information. One of them requires that a curve (possibly cobbed-smooth) passes approximately through all points which are set by the table. It is possible to do this using the methods of interpolation which were considered in the previous chapter. According to another approach the data is approximated using a simple function which is used for the whole interval of values (but not certainly) that it passes through all the stipulated points. Such approach is called an adjustment of curve, which aims to minimize the deviation from the data information. As a rule, the least-squares method is used, that brings to the minimum the sum of squares of differences between the values of function, which is set by the selected curve and the data table.

Suppose we have in the table the set $(n+1)$ point $(x_0, y_0), (x_1, y_1),\ ...,\ (x_n, y_n)$ and we need to find an approximate curve $g(x)$ in a range $x_0 \le x \le x_n$ (Figure 6.3). In this case, an error will be in every tabular point

$$e_i = g(x_i) - y_i.$$

Then the sum of squares of errors is calculated in the following way:

$$E = \sum_{i=0}^{n} \left[g(x_i) - y_i\right]^2.$$

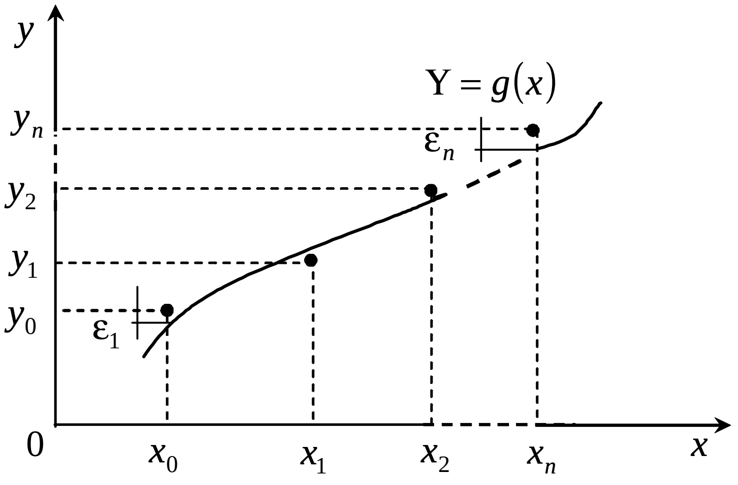


Figure 6.15

As a rule, the function $g(x)$ is elected as a linear combination of the chosen functions $g_k(x)$

$$g(x) = C_1 g_1(x) + C_2 g_2(x) + ... + C_k g_k(x).$$

The condition of minimum of E is reflected in the equation:

$$\frac{\P\ E}{\P\ C_1} = \frac{\P\ E}{\P\ C_2} = ... = \frac{\P\ E}{\P\ C_k} = 0.$$

It is known that

$$E=\sum_{i=0}^{n}\left[C_1 g_1(x_i)+C_2 g_2(x_i)+\ \dots\ +C_k g_k(x_i)-y_i\right]^2,$$

this condition is equivalent to the system of equations:

$$\frac{\P\ E}{\P\ C_1}=2\sum\left[C_1 g_1(x_1)+\dots+C_k g_k(x_i)-y_i\right]g_1(x_1)=0;$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

$$\frac{\P\ E}{\P\ C_k}=2\sum\left[C_1 g_1(x_i)+\dots+C_k g_k(x_i)-y_i\right]g_k(x_1)=0.$$

This system can be written in a matrix form:

$$\begin{bmatrix} \sum \mathrm{g}_1^2(x_i) & \sum g_1(x_i)g_2(x_i) & \dots & \sum g_1(x_i)g_k(x_i) \\ \dots & \dots & \dots & \dots \\ \sum g_1(x_i)g_k(x_i) & \dots & \dots & \sum g_k^2(x_i) \end{bmatrix} *$$

(6.5)

$$* \begin{bmatrix} C_1 \\ C_2 \\ . \\ C_k \end{bmatrix} = \begin{bmatrix} \sum g_1(x_i)y_i \\ \dots\dots \\ \sum g_k(x_i)y_i \end{bmatrix}.$$

Elements of matrix in the left part and column of vectors in the right are set by the data table. The given system of k linear equations with an unknown k can be solved. The choice of function $g(x)$ must be carried out taking into account the character of data (periodicity, property of symmetry, existence of asymptotic form). Sometimes the table is broken up into several parts and separate approximate curves are chosen for every part.

The residual finite middle quadratic error of approximation is:

$$\Delta=\sqrt{E/(n+1)}.$$

If during construction of an approximate function orthogonal polynomials are used

$$\sum g_j(x_i)g_k(x_i) = 0 \quad \text{if} \quad j \neq k,$$

the system (6.5) is simplified, and a matrix becomes diagonal. Coefficients could be obtained from the correlations

$$C_j = \sum_{i=0}^{n} g_j(x_i)y_i / \sum_{i=0}^{n} g_j^2(x_i).$$

For this reason in many standard programs of the curves adjustment orthogonal polynomials are used.

6.4 Statistical Data Processing

When processing the data in experiments there is a necessity to estimate descriptions of random values in measuring techniques (for example, estimation of the measuring error), automation (problem identification unification, optimum control), statistical radio engineering.

Estimation $\bar{x}$ of unknown mathematical expectation m_{X} of random value X demands middle arithmetic results n of the independent tests

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n},$$

and for estimation of dispersion s_x^2

$$\overline{\delta^2} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}.$$

Supposing the normal law of value distribution X it is possible to show that this value

$$T = \frac{\bar{x} - m_X}{\bar{d} / \sqrt{n}}$$

has *the t-division* of Student with $k = n - 1$ degrees of liberty. It’s possible to define a reliable interval for the real value x: after the known values of reliable probability P from Table 6.2 could be found. Thus:

$$\Delta = \varepsilon * \frac{\overline{\delta^2}}{\sqrt{n}}.$$

A casual size x is up-diffused after a normal law with a mathematical hope m_X and dispersion s_X^2. The real value x is in the interval $(m_x - \Delta, m_x + \Delta)$ with a reliable probability P. To estimate the type of law most applications have the criteria of Kolmogorov and Pirson, which act using comparison of empiric function of distribution $f_x^*(x)$, histogram obtained as a result of the experimental data processing, with hypothetical $f_x(x)$, which answers the offered hypothesis. This allows to make conclusions about their convergence or divergence at the level of significance α, which allows to calculate the probability of correctness of the given hypothesis.

In a Kolmogorov criterion coefficient l is defined as:

$$l = \left| f_X(x) - f_X^*(x) \right|_{\max} \sqrt{n},$$

which is compared to the critical value set from Table 6.3.

Table 6.2

Value for a reliable interval $-\varepsilon < t < \varepsilon$, where value t has the Student's distribution, depends on the reliable probability p and the amount of liberty degrees k

k	p=0.90	p=0.95	p=0.99
1	6.310	12.71	63.7
2	2.920	4.30	9.92
3	2.350	3.18	5.84
4	2.130	2.77	4.60
5	2.020	2.57	4.03
6	1.943	2.45	3.71
7	1.895	2.36	3.50
8	1.860	2.31	3.36
9	1.833	2.26	3.25
10	1.812	2.23	3.17
11	1.796	2.20	3.11
12	1.782	2.18	3.06
13	1.771	2.16	3.01
14	1.761	2.14	1.98
15	1.753	2.13	2.95

16	1.746	2.12	2.92
17	1.740	2.11	2.90
18	1.734	2.10	2.86
19	1.729	2.09	2.86
20	1.725	2.08	2.84
22	1.717	2.07	2.82
24	1.711	2.06	2.80
26	1.706	2.06	2.78
28	1.701	2.05	2.76
30	1.697	2.04	2.75
40	1.684	2.02	2.70
60	1.671	2.00	2.66
120	1.658	1.98	2.62
	1.645	1.96	2.58

Table 6.3

Critical values λ_0 depend on the level of significance

α	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.001	0.001
λ_0	0.828	0.895	0.974	1.073	1.224	1.358	1.520	1.627	1.950

At $\lambda < \lambda_{ђp}$ a hypothesis about convergence $f_X(x)$ and $f_X^*(x)$ is adopted. c^2 coefficient is calculated using the Pirson criterion

$$x^2 = \sum_{i=0}^{k} \frac{[f_X(x_i) - f_X^*(x_i)]^2}{f_X(x_i)},$$

where *k* is number of digits of histogram (discrete values $f_X(x_i)$).

From Table 6.4 the critical value is determined in accordance with α and amount of liberty degrees

$$r = k - l - 1,$$

where l - amount of parameters in the law of distribution (for normal l=2, for Poisson l=1 and etc).

At $\chi^2 < \chi^2_{ђp}$ the hypothesis is accepted.

Comparing the analytically obtained laws of probabilities distribution, it is convenient to measure their proximity using the value of middle quadratic error.

To estimate dependence of random values, which have stochastic connection, the coefficient of correlation is used

$$R_{xy} = \frac{1}{n-1}\frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_Y)}{\sigma_x \sigma_y}.$$

Determining the interdependence of random values in different moments of time, the coefficient of correlation is estimated by means of the following formula:

$$R_X(t) = \frac{1}{n-m-1}\frac{\sum_{i=1}^{n-m}\left[x(t_i) - m_X\right]\left[x(t_i + t) - m_X\right]}{D_X},$$

Table 6.4

Critical points of distribution

x – random value which is distributed by the χ^2 law with the liberty degrees k (the table contains values which are obtained from condition $P(x) \le \alpha$)

Liberty Degrees k	$\alpha =$ 0.01	α =0; 0.025	$\alpha =$ 0.05	$\alpha =$ 0.95	$\alpha =$ 0.975	$\alpha =$ 0.99
1	6.6	6.0	3.8	0.0039	0.00098	0.00016
2	9.2	7.4	6.0	0.103	0.051	0.020
3	11.3	9.4	7.8	0.352	0.216	0.115
4	13.3	11.1	9.5	0.711	0.484	0.297
5	15.1	12.8	11.1	1.15	0.831	0.554
6	16.8	14.4	12.6	1.64	1.24	0.872
7	18.5	16.0	14.1	2.17	1.69	1.24
8	20.1	17.5	15.5	2.73	2.18	1.65
9	21.7	19.0	16.9	3.33	2.70	2.09

10	23.2	20.5	18.3	3.94	3.25	2.56
11	24.7	21.9	19.7	4.57	3.82	3.05
12	26.2	23.3	21.0	5.23	4.40	3.57
13	27.7	24.7	22.4	5.89	5.01	4.11
14	29.1	26.1	23.7	6.57	5.63	4.66
15	30.6	27.5	25.0	7.26	6.26	5.23
16	32.0	28.8	26.3	7.96	6.91	5.81

where $x(t_i)$ is value of random value X in the moment of time t_i, and $x(t_i+\tau)$ – in the moment of time which differs from t_i on the interval τ. Thus, $x(t_i)=x_i$, $x(t_i+t)=x_j$, τ is a time domain between i and j values x, $i-j=m$.

The interval of correlation is considered as a period of time for which a cross-correlation function diminishes 95 %.

After obtaining the data regarding x arrays and y, the calculation of cross-correlation function is rather simple, but approximation of the type of cross-correlation function by typical cross-correlation functions (Table 6.5) can be performed using the least-squares method.

Table 6.5

Typical cross-correlation functions

Form	Parameters
$R_x(t)=s_X^2(1-a\|t\|)$, $t<1/a$	$a=(s_X^2-R_x(t^*))/s_X^2t^*$, $R_x(\tau^*)$ – known value of cross-correlation function
$R_x(t)=s_X^2e^{-a\|t\|}$	$a=\frac{1}{t^*}\ln\frac{s_X^2}{R_x(t^*)}$
$R_x(t)=s_X^2e^{-a^2t^2}$	$a=\frac{1}{t^*}\sqrt{\ln\frac{s_x^2}{R_x(t^*)}}$
$R_x(t)=s_x^2e^{-a\|t\|}(1+a\|t\|)$	$\alpha\approx 4{,}5/\tau_k^{max}$

$R_x(\tau)=\sigma_x^2 e^{-\alpha\|\tau\|} cos\,\beta\tau$	$a=\frac{1}{t_2^*}*\ln\frac{s_x^2\cos\frac{pt_2^*}{2t_1^*}}{R_x(t_2^*)}$, $\beta=\pi/2\tau_1^*$ at two known values of cross-correlation function $R_x(\tau)$, thus $R_x(t_1^*)=0.$

6.5 Numerical Integration

In many problems, which are related to identification, analysis, quality estimation of complex systems in automatics, there is a need to determine definite integrals.

If function $f(x)$ is an antiderivative function that is situated on the interval $[a,в]$, the definite integral from $f(x)$ can be calculated using the Newton-Leibnitz formula:

$$I=\int_a^в f(x)dx=F(в)-F(a),$$

where $F'(x)=f(x)$.

But often it’s quite difficult to calculate the integral because of complicated analytical transformations (and sometimes even impossible, especially in cases of improper integrals), as a subintegral function is set by the numerical data, for example, obtained from the experiment.

The problem of the function’s numerical integration consists of calculation of the integral’s value on basis of the subintegral function’s values. Graphically an integral is considered to be an area limited to the graph of function

$$y=f(x).$$

The most widespread methods of definite integrals’ calculation are:

- methods of Newton-Cotes, Gauss, Tchebyshev, that are based on the use of the so-called quadrature formulas replacing interpolations polynomials of $f(x)$;
- methods of Monte Carlo, based on the use of statistical models.

6.5.1 Newton-Cotes Formulas

To get Newton-Cotes formulas the integral should be given in a form:

$$\int_a^b f(x)dx = \sum_{i=0}^{n} A_i f(x_i) + \Delta, \tag{6.6}$$

where x_i - knots of interpolation; A_i - coefficients depending on the type of formula; Δ - error of quadrature formula.

Replacing in (6.6) a subintegral function by the proper interpolation Lagrange polynomial for n equidistant knots with a step $h = \dfrac{b-a}{n}$ makes it possible to get the next formula to calculate coefficients A_i at the arbitrary amount of knots.

$$A_i = \frac{b-a}{n}\frac{(-1)^{n-1}}{i!(n-1)!}\int_0^n \frac{q(q-1)...(q-n)}{(q-i)}dq, \tag{6.7}$$

where $q = \dfrac{x-a}{h}$.

Usually coefficients $H_i = \dfrac{A_i}{b-a}$ are called the Cotes coefficients.

Thus, formula (6.6) is transformed to the following form:

$$\int_a^b f(x)dx = (b-a)\sum_{i=0}^{n} H_i f(x_i) \qquad . \tag{6.8}$$

With the following characteristics:

$$\sum_{i=0}^{n} H_i = 1 \quad {}^{3} \quad H_i = H_{n-1}.$$

At $n=1$ ^{3} $n=2$ with (6.7) and (6.8) the formulas of trapezoids and Simpson could be obtained:

$$I = \frac{h}{2}\left[f(x_0) + f(x_1)\right],$$

$$I = \frac{h}{3}\left[f(x_0) + 4f(x_i) + f(x_2)\right].$$

In Table (6.6) the resulted values of coefficients for $n = 1, 2, ..., 8$ are given.

The errors of formulas of trapezoids and Simpson are estimated, accordingly, using expressions:

$$\Delta = -\frac{h^3}{12}M_2 \quad and \quad \Delta = -\frac{h^5}{90}M_4,$$

where M_2 з M_4 – are the maximal values of the second and fourth derivative $f(x),\ x \in (a, b)$.

The complex Newton-Cotes formulas are combined from the simple formulas. For example, for the formulas of trapezoids and Simpson:

$$I = \frac{h}{2}\left[f(x_0) + 2f(x_1) + 2f(x_1) + ... + f(x_n)\right],$$

$$I = \frac{h}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + ... + f(x_n)\right].$$

The errors of complex formulas are accordingly:

$$\Delta = -n\frac{h^3}{12}M_2 \quad \grave{a}nd \quad \Delta = -n\frac{h^5}{180}M_4.$$

It is possible to get the component Newton-Cotes formulas of higher orders.

In order to estimate the error practically, the methods of Runge (Richardson extrapolations) are used. This method was studied in chapter 4.

Table 6.6

Cotes Coefficients

$\acute{H}_i = H_i$ N	$\acute{H}_0$	$\acute{H}_1$	$\acute{H}_2$	$\acute{H}_3$	$\acute{H}_4$	$\acute{H}_5$	$\acute{H}_6$	$\acute{H}_7$	$\acute{H}_8$	Sum
1	1	1								2
2	1	4	1							6
3	1	3	3	1						8
4	7	32	12	32	7					90
5	19	75	50	50	75	19				288
6	41	216	27	272	27	216	41			840
7	751	3577	1223	2989	2989	1323	3577	751		17280
8	989	5888	-928	10496	-4540	10496	-928	5888	989	28350

6.5.2 Tchebyshev's Formula

Formula (6.6) can be derived to the form:

$$\int_{-1}^{1} f(t)dt = \sum_{i=1}^{n} A_i f(t_i) \qquad (6.9)$$

performing replacement of variables:

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

In the course of the Tchebyshev's formula derivation such terms are used:

- All coefficients A_i are equal;
- The quadrature formula (6.9) is exact for all polynomials up to degree n inclusively.

Thus, formula (6.9) looks like:

$$\int_{-1}^{1} f(t)dt = \frac{2}{n}\sum_{i-1}^{n} f(t_i). \tag{6.10}$$

To find t_i one should consider that formula (6.10) must be exact for the function of the form :

$$f(t) = t^k, \quad k = 1, \ ..., \ n.$$

After substituting these functions in (6.10) the following system of equations could be obtained:

$$\begin{cases} t_1 + t_2 + ... + t_n = 0; \\ t_1^2 + t_2^2 + ... + t_n^2 = \dfrac{n}{3}; \\ t_1^n + t_2^n + ... + t_n^n = \dfrac{n\left[1-(-1)^{n+1}\right]}{2(n+1)}. \end{cases} \tag{6.11}$$

System of equations (6.11) has a solution at $n < 8$ *and* $n = 9$. The imperfection of the Tchebyshev's formula consists namely in the restricted exactness. The values t_i for different n are given in table (6.7).

For an arbitrary interval (a, b) formula (6.10) could be presented as follows:

$$I = \frac{b-a}{n}\sum_{i=1}^{n} f(x_i)\,,$$

where $$x_i = \frac{a+b}{2} + \frac{b-a}{2}t_i\,.$$

Error of calculations in the Tchebyshev's method is:

$$\Delta = \int_a^b \frac{\left(x - \frac{a+b}{2}\right)^{n+1}}{(n+1)!} f^{(n+1)}(x)dx - \frac{b-a}{n(n+1)!}\sum_{i=1}^{n}\left(x_i - \frac{a+b}{2}\right)^{n+1} f^{(n+1)}(x)\,.$$

Table 6.7

Value of abscissas t_i in the Tchebyshev’s formula

n	i	t_i	n	i	t_i
2	1;2	m0.577350	6	1;6	m0.866247
3	1;3	m0.707107		2;5	m0.422519
	2	0		3;4	m0.266635
4	1;4	m0.794654	7	1;7	m0.883862
	2;3	m0.187592		2;6	m0.529657
5	1;5	m0.832498		3;5	m0.323912
	2;4	m0.374513		4	0
	3	0			

6.5.3. Gauss Formula

Gauss formula is a formula of the highest algebraic exactness. For the formula of form (6.9) the highest exactness can be achieved for the polynomials of degree $(2n-1)$, which concern $2n$ values t_i ³ $A_i (i = 1,2,...,n)$.

The problem consists of determining coefficients A_i and abscissas of points t_i.

To determine the coefficients, formula (6.9) is often used for the functions of form

$$f(t) = t^k, k = 0, 1, ..., 2n-1.$$

Obviously

$$\int_{-1}^{1} t^k dt = \begin{cases} 2/(k+1) \\ 0 \end{cases},$$

Then, the system of equations is:

$$\begin{cases} \sum_{i=1}^{n} A_i = 2; \\ \sum_{i=1}^{n} A_i t_i = 0; \\ \sum_{i=1}^{n} A_i t_i^2 = 1; \\ \sum_{i=1}^{n} A_i t_i^{2n-2} = \dfrac{2}{2n-1}; \\ \sum_{i=1}^{n} A_i t_i^{2n-1} = 0. \end{cases} \qquad (6.12)$$

This system is nonlinear, and its ordinary solution is connected with numerous difficulties in calculations. But, if we use the system for the polynomials of form

$$f(t) = t^k P_n(t), \quad k = 0,1,...,n-1,$$

where $P_n(t)$ - Legendre polynomial, then it can be derived to linear form relating coefficients A_i with the determined points t_i. As the degrees of the polynomials do not exceed in correlation $2n-1$, the system (6.12) and formula (6.9) to be executed assumes

$$\int_{-1}^{1} t^k P_n(t)dt = \sum_{i=1}^{n} A_i t_i^k P_n(t_i). \qquad (6.13)$$

As the orthogonal left part of expression (6.13) equals to 0, then:

$$\sum_{i=1}^{n} A_i t_i^k P_n(t_i) = 0,$$

that is always provided at any values A_i in points t_i which correspond to the roots of the proper Legendre polynomials.

Putting these values t_i into system (6.12), we can calculate the first n equations, and then it is possible to define coefficients A_i.

Formula (6.9), where t_i equals to zero of Legendre polynomial $P_n(t)$, and $A_i, i = 1, 2, ..., n$ are calculated from system (6.12), is called the Gauss formula.

Values t_i, A_i for different n are given in Table 6.8.

For an arbitrary interval (*a,b*) the Gauss formula is the following:

$$I = \frac{b-a}{2}\sum_{i=1}^{n} A_i f(x_i)\,,$$

where

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} t_i\,.$$

To estimate the error of the Gauss formula with n knots we could use correlation

$$\Delta \le \frac{(b-a)^{2n+1}(n!)^4 M_{2n}}{2^{2n+1}\left[(2n)!\right]^3 (2n+1)},$$

where M_{2n} - maximal value of $2n$ derivative on interval $[a,b]$.

Table 6.8

Elements of Gauss formula

n	i	t_i	A_i
1	1	0	2
2	1;2	**m**0.57735027	1
3	1;3	**m**0.77459667	$\frac{5}{9}$=0.55555556
	2	0	$\frac{8}{9}$=0.88888889
4	1;4	**m**0.86113631	0.34785484
	2;3	**m**0.33998104	0.65214516
6	1;6	**m**0.93246951	0.17132450
	2;5	**m**0.66120939	0.36076158
	3;4	**m**0.238619119	0.46791394
7	1;7	**m**0.94910791	0.12948496
	2;6	**m**0.74153119	0.27970540
	3;5	**m**0.40584515	0.38183006
	4	0	0.41795918

8	1;8	m 0.96028986	0.10122854
	2;7	m 0.79666648	0.22238104
	3;6	m 0.52553142	0.31370664
	4;5	m 0.18343464	0.36268378

6.5.4 Algorithm of Numerical Methods

The sequence in which Newton-Cotes formulas should be applied is given below.

1. Choice of the formula and finding coefficients A_i (using Table 6.6).
2. Drafting the algorithm and program, so that:

- if we set discrete values $y_i = f(x_i)$ with step h, these values are put into (6.6);
- if function $y = f(x)$ of value $y_i = f(x_i)$ is calculated, then $x_i = x_0 + ih = a + ih(a \le x \le b)$.

3. Estimation of errors.

The sequence in which the Gauss method should be applied is given below.

1. Choice of order of the method and finding coefficients A_i (using Table 6.8) and values $t_i (-1 \le t \le 1)$.
2. Laying out of interval $a \le x \le b$ *on* l subintervals (Figure 6.4).
3. Finding the values of integral I_j for every interval $(j = 1,...,l)$

$$I = \sum_{j=1}^{l} I_j \, .$$

Thus, the values on abscissas x_i on every interval j are calculated in accordance with the formulas:

$$x_i = \frac{a_{j+1} + a_j}{2} + \frac{a_{j+1} - a_j}{2} t_i \ , \qquad (6.14)$$

where

$$a_{j+1} = a_j + h; \qquad a_1 = a;$$
$$a_{l+1} = b; \qquad j = 1,\ldots,l;$$
$$h = \frac{b-a}{n}.$$

I_j is calculated as:

$$I_j = \int_{a_j}^{a_{j+1}} f(x)dx = \frac{(a_{j+1} - a_j)}{2} \sum_{i=1}^{n} A_i f(x_i). \qquad (6.15)$$

4. Estimation of errors.

The sequence in which the Tchebyshev’s method should be applied is similar to the Gauss method, but in point 1 coefficients t_i are to be taken from table 6.7, and in point 3, in order to find I_j integral, the following formula is to be used:

$$I_j = \int_{a_j}^{a_{j+1}} f(x)dx = \frac{(a_{j+1} - a_j)}{n} \sum_{i=1}^{n} f(x_i), \qquad (6.16)$$

where x_i is estimated similarly to the Gauss method in accordance with formula (6.14). Practically, the errors could be estimated using the Runge method for comparatively high number of subintegral *l*.

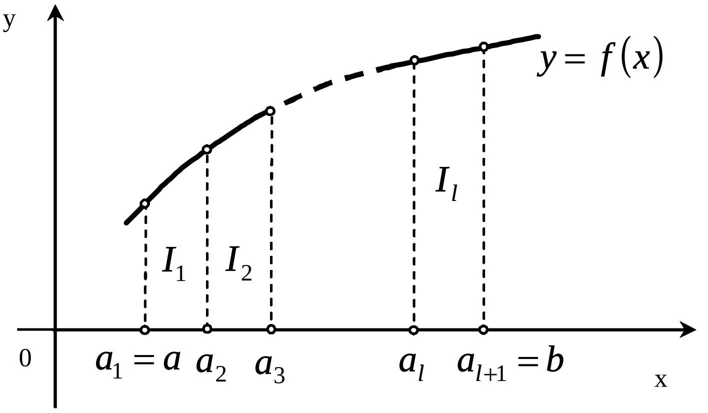


Figure 6.16

For the Gauss method

$$\Delta = C\left(\frac{b-a}{l}\right)^{2n+1},$$

and for the Tchebyshev's method

$$\Delta = C\left(\frac{b-a}{l}\right)^{n+1},$$

where coefficients *c* could be calculated via two calculations with high, but different meanings *l.*

6.5.5 Monte - Carlo Method

The method of numerical integration of Monte-Carlo is the most widespread method of statistic modelling that is used for problems solution in the applied mathematics.

Suppose, we have random values $\{x_i\} \in X$ sequence with the distribution law of probabilities $f_x(x)$. To perform functional transformation

$$y_i = \varphi(x_i),$$

the expectation of the obtained random value sequence $\{y_i\} \in Y$

$$m_Y = \int_{-\infty}^{\infty} j\,(x) f_x(x) dx$$

could be estimated using formula:

$$m_Y = \frac{1}{n}\sum_{i=1}^{n} y_i \quad . \tag{6.17}$$

Let's enter in expressions (6.17) the so-called function of the area indicator:

$$1[a,\ b,\ x] = \begin{cases} 1, & a \le x \le b; \\ 0, & x < a,\, x > b. \end{cases}$$

After that if we choose a function of a form

$$j\,(x) = \frac{f\,(x)}{f_{x(x)}},$$

the final expression will be:

$$I = m_Y = \int_a^в f\,(x)dx = \frac{1}{n}\sum_{i=1}^{n}\frac{f\,(x_i)}{f_x(x_i)}1[a,\,в,\,x_i].$$

The algorithm of the integral calculation using the Monte-Carlo method is given in Figure 6.16.

The error of Monte-Carlo method is related to the error of generation of the values probable sequence, that are computer-calculated, with definite laws. It can be estimated using the formula:

$$\Delta = \frac{1}{2\sqrt{n(1-P)}}, \tag{6.18}$$

where P is the actual probability error on interval $[-\Delta\,;+\Delta\ ]$.

The amount of tests of n does not depend on the dimension of the integral, that is why the Monte-Carlo method is applied to calculate multiple integrals, where the other methods of numerical integration are not effective because of the huge amount of calculations needed.

Let us consider the sequence in which the calculation of multiple integrals should be performed. First, we need to have m random numbers generators, where *m* equals to multiplicity of integral.

Geometrically the calculation of *m*-multiple integral

$$I = \iint\limits_{(S)}\ldots\int f\,(x_1, x_2, \ldots, x_m)dx_1dx_2\ldots dx_m\,, \tag{6.19}$$

where $y = f\,(x_1, x_2, \ldots, x_m)$ - continuous function in the limited reserved area of S - is used to determine $(m+1)$-volume of direct cylinder in space $0x_1, x_2, \ldots x_m y$, that is built on the basis of S and is limited by surface $y = f\,(x_1, x_2, \ldots, x_m)$.

To transform integral (6.19) so that the new area of integration is situated in the middle of single *m* – dimensional cube σ, we can replace the variables as follows:

$$x_i = a_i + (b_i - a_i)\xi_i\,,$$

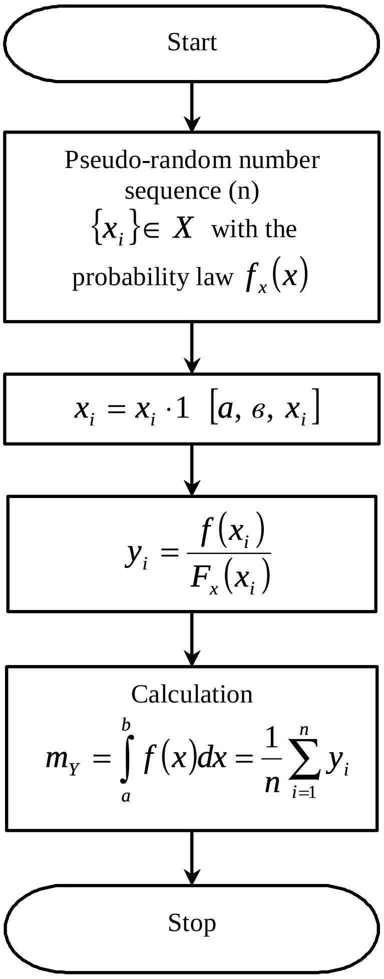
where ξ_i - the proper co-ordinates from 0 to 1; a_i, b_i - maximum values of co-ordinates, where the area of integration is located.

Then using (6.19) the integral could be calculated. So:

$$I = (a_1 - b_1)(a_2 - b_2)...(a_m - b_m)I_x,$$

where

$$I_x = \iint\limits_{(s)}...\int f(x_1, x_2, ..., x_m)dx_1 dx_2 ... dx_m.$$



Start
Pseudo-random number sequence (n)
$\{x_i\} \in X$ with the
probability law $f_x(x)$
$x_i = x_i \cdot 1\ [a,\ в,\ x_i]$
$y_i = \frac{f(x_i)}{F_x(x_i)}$
Calculation
$m_Y = \int_a^b f(x)dx = \frac{1}{n}\sum_{i=1}^{n} y_i$
Stop

Figure 6.17

If we apply *m* generators of random numbers in a range (0,1), calculation of the mean value of the function from their combinations using the multidimensional indicator of the integration area will give us the sought estimation of the integral:

$$I_x = \frac{1}{n}\sum_{i=1}^{n} f(x_{1i}, x_{2i}, ..., x_{mi})1[x_{1i}, x_{2i}, ..., x_{mi}],$$

where $1\lfloor x_{1i}, x_{2i}, ..., x_{mi} \rfloor$ equals to 1, if the point is situated in the middle of the integration area, otherwise it equals to 0.

The error of calculation of m-multiple integral using the Monte-Carlo method is estimated like single using formula (6.18).

6.6 Remarks

The issue of data processing is quite broad and includes not only the problems studied in the chapter. Statistical data processing unites a multitude of problems considering the characteristics calculation of the random processes (not only the laws of the unidimension probability and the correlation function). We should point out that many problems are connected with processing of the multidimensional data sets.

Exercises

1. What is the difference in statements of the interpolation and approximation problems?
2. What is the error of calculation $\sqrt{2}$ according to the Lagrange formula for function $\sqrt{x}$ with knots of interpolation $x_0 = 81,\ x_1 = 1,\ x_2 = 4$?
3. Give the first and second Newton interpolation formulas.
4. What is the concept of extrapolation? Is it possible to make an extrapolation basing upon the splines method?
5. Build the algorithms of Chebyshev, Gauss, Newton-Cotes numerical integration. Use these methods to calculate the following integrals:

$$\text{a) } \int_0^1 \sqrt{x} \exp x^2 \left(1 + \sin x \cos \sqrt{x}\right) dx;$$

$$\text{b) } \int_1^2 \sqrt{\sin \sqrt{x}} \cos x \left(1 + x^2\right) dx;$$

$$\text{c) } \int_2^3 \sqrt{x} e^x dx.$$

6. Using the method of the least squares, approach the points (0, -1), (1, 1), (2, 3), (3, 4), (4,5.1). Find the straight line $y = a_1 + a_2 x$.
7. Using the method of the least squares and the points

x	-3	-2	-1	0	1	2	3
y	-0.72	-0.01	0.50	0.82	0.89	0.81	0.50

find approximation of a polynomial of the form : $y = a_1 + a_2 x + a_3 x^2$.

8. Using the method of the least squares, fit the points (0, 1), (1, 0), (2, -7), (3, -26) with a polynomial of the form $y = a_1 + a_2 x + a_3 x^2 + a_4 x^3$ and show that the result is identical with the cubic polynomial in the given points.

9. For each of the following, determine the exact $y'(1)$. Then approximate $y'(1)$ using the two-point and the three-point backward-difference formulas:

$a)\ y = -5,\ \Delta x = 0.1;$ $b)\ y = x^4,\ \Delta x = 0.05;$

$c)\ y = x^2 - 3x - 7,\ \Delta x = 0.2;$ $d)\ y = x^7,\ \Delta x = 0.05;$

$e)\ y = (1-x)/(1+x^2),\ \Delta x = 0.1;$ $f)\ y = e^x,\ \Delta x = 0.1;$

$g)\ y = \sin px,\ \Delta x = 0.001;$ $h)\ y = e^{x^2},\ \Delta x = 0.1.$

10. Build an algorithm of calculations using the Monte-Carlo method for the following integral

$$I = \iint\limits_{(s)} (x^2 + y^2)\ dx\,dy,$$

where s - range of integrations define the following inequations $\frac{1}{2} \le x \le 1,\ 0 \le y \le 2x - 1.$ Are the points (0.55; 0.75); (0.25; 0.75); (0.25; 0.25); (0.99; 0.70) situated in the integration area?

11. Calculate the meaning $z = f(0.5;\ 0.03)$ for function $f(x,\ y)$ using the Newton and the Lagrange methods

x y	0.4	0.7	1.0
0	2.5	1.43	1.00
0.05	2.49	1.42	0.99
0.10	2.46	1.40	0.98

12. Give definitions of the fractal and of the selfsimilarity.

13. Build an algorithm for fractal interpolation for the initial points with two coordinates.

Chapter 7. Interval Calculations

7.1 Introduction

Arithmetic operations are always subject to a certain amount of procedures that can be performed on any computer. But due to the fact that all the numerical methods developed so far are based upon finite sequences of arithmetic operations, nowadays it is necessary to consider the issue. Because of the elementary essence of this book, we have selected the most elementary, but relatively efficient, approach to the problem that is the interval arithmetic. As its name suggests, the interval arithmetic is the arithmetic of intervals.

We are interested in this kind of arithmetic for the following reason. Suppose we have to calculate

$$x + y,$$

where x and y are known only approximately. For example, if to three decimal places, x and y are 4.102 and 1.8333, respectively, then x is definitely situated in the interval

$$4.1023 \leq x \leq 4.1033 \qquad (7.1)$$

and y - in the interval

$$1.8321 \leq y \leq 1.8333. \qquad (7.2)$$

If we add (7.1) and (7.2) termwise, which may be described as adding of the intervals, then:

$$5.9344 \leq x + y \leq 5.9366, \qquad (7.3)$$

and upper and lower bounds for the exact sum of the values x and y are obtained. Finally, if we suppose that $x + y$ is the midvalue of the interval (7.3), so that

$$x + y = 5.9355, \qquad (7.4)$$

then we can conclude, that due to the fact that 5.3720 is the midvalue, (7.4) has an error of at most +0.0011. Thus we have not only approximated the exact, but unknown, sum of $x + y$, but we have also found out that bounds for the amount could have an error.

Since such approximations, with error bounds, would be true for all arithmetic operations, not just addition, we should develop general rules of interval arithmetic. Because of the preliminary theoretic type of the discussion, it will be suitable to write an expression such as "A is a subset of B" as the usual theoretic notation "A ⊂: *B".* Also, as is expected, we allow every set to be a subset of itself, so that $A \subset B$ does not reject A's being equal to *B.*

7.2 Basics of Interval Operations

For $a < b$, the symbol *[a,b]* is often used, as in this book, to represent the interval of numbers x that satisfy $a < x < b$. In interval arithmetic, however, it will be important to allow a to be equal to b, which leads to the following definition.

Definition 7.1. For $a < b$, the interval *[a;b]* is the set of all real numbers x that satisfy $a < x < b$.

Definition 7.2. Let $I = [a;b]$, $J = [c;d]$. Then $I + J$ is the set of all real numbers $x + y$, where x Î I and y Î J.

Theorem 7.1. For two arbitrary intervals *[a;b]* and *[c;d],*

$$[a;b]+[c;d] = [a+c; b+d].$$

Proof. If x Î *[a; b]* and y Î *[c; d]*, then

$$a \le x \le b,$$
$$c \le y \le d.$$

Hence

$$a+c \pounds x+y \pounds b+d,$$

so that $x + y$ is in $[a + c; b + d]$. To complete the proof, we need only to show that any number $z \in [a + c; b+d]$ can be written in the form z = $x + y$, where $x \in [a; b]$ and y Î *[c;d]*. But this follows easily because z = $x + y$ is a continuous function which is minimal at $x = a$, $y = c$, and is maximal at $x = b$, $y = d$, and because a continuous function takes on all the intermediate values between its minimum and its maximum. Thus, the theorem is proved.

Definition 7.3. Let $I = [a;b]$, $J = [c;d]$. Then $I - J$ is the set of all real numbers $x - y$ where x Î I and $y \in J$.

Theorem 7.2. For two arbitrary intervals *[a;b]* and *[c;d],*

$$[a;b] - [c;d] = [a;b] + [-d;-c] = [a - d; b - c].$$

Proof. The proof is completely similar to that of Theorem 7.1.

Theorem 7.2 motivates the following consistent and convenient definition, which, in fact, enables one to view subtraction in the usual sense as the inverse of addition.
Definition 7.4. $-[a;b] = [-b; -a]$.
Definition 7.5. If $I = [a;b]$ and $J = [c;d]$, then $I \times J$ is the set of all numbers $x \cdot y$, where $x \in I$ and y Î J.
Theorem 7.3. For two arbitrary intervals $[a;b]$ and [c;d],

$$[a; b] \bullet [c; d] = [\min(ac, ad, bc, bd); \max[ac, ad, hc, bd].$$

Proof. The proof is similar to that of Theorem 7.1.
Definition 7.6. If 0 is not in $[c;d]$, then $[a;b] / [c;d]$ is defined as the set of all real numbers x/y where $x \in [a;b]$ and y Î $[c;d]$.
Theorem 7.4. For two intervals $[a;b]$ and $[c;d]$, where 0 is not in $[c;d]$,

$$[a;b]/[c;d]=[a;b]\times\left[\frac{1}{c};\frac{1}{d}\right].$$

Proof. The proof follows from Theorem 7.3.

7.3 Applications to Calculations

The rationale behind the very extensive applications made of interval arithmetic (see, e.g., Moore) is based on the following direct consequence of the definitions and theorems of Section 7.2. For exact values $x_1, x_2, \ldots, x_n$, suppose one wants to determine $F(x_1, x_2, \ldots, x_n)$, where F is a given rational function. Suppose, however, $x_1, x_2, \ldots, x_n$ are known only approximately; that is, for intervals $I_1, I_2, \ldots, I_n$ one knows only that x_1ÎI_1, x_2ÎI_2, …, x_nÎI_n and that by means of the interval arithmetic one calculates $F(I_1, I_2, \ldots, I_n)$. Then

$$F(x_1,x_2\ldots x_n)\,\hat{I}\,\mathrm{F}(I_1,I_2,\ldots I_n).$$

We are thus led to the following convenient three-step algorithm of the interval arithmetic application:

Step 1. In performing a set of arithmetic operations in which only rounded numbers are available, replace the numbers by intervals that contain them. For example, suppose one wants to determine x from

$$x = m^2 + 2n - c, \tag{7.5}$$

with m = 0.75, which is known to be correct to only two decimal places, with n = 0.10056, which is known to be correct to only five decimal places, and with c

= 0.00201, which is known to be correct to only five decimal places. Then, in place of

$$m^2 + 2n - c, \tag{7.6}$$

we consider

$$[0.745;0.755]^2 + 2[0.100555;\ 0.100565] - [0.002005;\ 0.002015]. \tag{7.7}$$

Step 2. Combine the result of step 1 by means of the interval arithmetic to yield a single interval *[a;b]*. Thus, for example, (7.7) combines into

$$[0.754120;\ 0.769150]. \tag{7.8}$$

Step 3. Take the midpoint of the interval *[a;b]* generated by step 2 as an approximation to the desired value. This midvalue is $\bar{x} = (a + b)/2$. By virtue of $\bar{x}$'s being the midpoint, it is in error by at most $|(b\text{-}a)/2|$. Thus, for example, the midvalue $\bar{x}$ of (7.8) is $\bar{x}$ = 0.761635 and the error $|\bar{x} - x|$ is at most 0.007515; that is,

$$|\bar{x} - x| \le \frac{0.769150 - 0.754120}{2} = 0.007515\,.$$

Two observations are important in the practical application of interval arithmetic. First, note that if one has to calculate an expression such as

$$xy + xz, \tag{7.9}$$

where $x \in I, y \in J, z \in K$, then it is convenient to rewrite it in the form

$$x(y + z). \tag{7.10}$$

For substitution of intervals into yields

$$IJ + IK, \tag{7.11}$$

whereas substitution into yields

$$I(J + K). \tag{7.12}$$

Since the error is determined by the width of the final interval, (7.11) has a width which is not greater than that of (7.12) and hence is should be considered the most desirable.

Finally, a very important rule must be followed when one is using interval arithmetic on a digital computer. The computation of intervals often requires rounding of numbers. To be sure that the resulting interval still contains the exact value that is to be approximated, one should always round the left number in an interval down and the right number - up, as it is illustrated in the following example.

Example. Suppose that in the process of performing interval calculations, a digital computer determines that the answer *x* lies in the interval *J* = [0.11127;0.21123]. But suppose that the computer can carry only four decimal places so that the end points of J have to be rounded. If one uses any of the usual rules for rounding, then the computer would consider the interval J_1 = [0.1113;0.2112]. Unfortunately, now, even though the solution *x* was in *J,* it need not be in J_1, for *J* is not contained in J_1. To be assured that *x Î* J_1, one should always round the left-hand end point of J down, and the right-hand end point up, so that *J* should be rounded to [0.1112;0.2113].

7.4 Applications to Modelling of Complex Systems

A common way to represent and analyse complex systems is to implement systems models. Starting with signal modelling one comes to the mathematical model of information parameters transformation. Construction of a model on basis of the interval analysis allows to find an effective decision to consider the uncertainty of informative parameters value, which could be formed either in time of calculating or in time of processing. It is sufficient to know an interval, in which the value of the parameter could be found. The interval analysis represents an interval as an integral object, and for all further calculations it is sufficient only to know borders of the interval.

Generally, a complex system consists of various sorts of converters. A measuring converter represents an elementary part of such a system, carrying out the transformation function (W) of set of the informative influences (X) to set the target signals (Y) in the field of various non-informative influencing factors (Z) (Figure 7.1):

$$Y = W[X, Z],$$

where $X = \{x_1, ..., x_n\}, Y = \{y_1, ..., y_n\}, Z = \{z_1, ..., z_n\}$ are sets of informative influences, target signals and non-informative influencing factors accordingly.

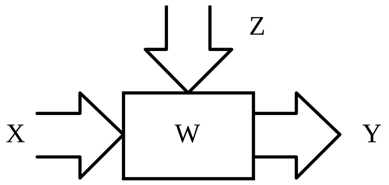


Figure 7.1

The classical approaches to modelling consider sets X, Y and Z as sets of real obtained value with a certain error. In the paper we mark them as interval numbers or just intervals. It means that if a certain parameter A of the system has value g, determined with an absolute error e, within the interval analysis we can present it as interval a:

$$A = g \pm e \Rightarrow a = [\overline{a}; \underline{a}],$$

where $\underline{a}, \overline{a}$ - bottom and top (or right and left) borders of interval a, in the standard notations:

$$\underline{a} = g - e, \overline{a} = g + e \ .$$

If all algebraic operations are carried out according to the rules of interval analysis, at the end of calculations we obtain the final result as some interval, which represents bilateral approximation of the exact result.

There are the interval models of static and dynamic linear and nonlinear converters and digital converters of complex systems. Interval models of static converters are similar to ordinary mathematical models. There are also some typical models of nonlinear static converters as “dead space”, “hysteresis”, “backlash” etc.

In dynamic models the output process could be presented in the form of a Duhamel integral:

$$y(t) = \int_0^t g(t) \cdot x(t - t)\, dt \ .$$

Let's present the initial process via the Duhamel's integral in a discrete form, that is a natural form for the discrete and analogue signals This kind of presentation is necessary to perform computer calculations:

$$y(t) = \sum_{i=0}^{n-1} g_{\Delta}(i_{\Delta} t) \cdot x(t - i_{\Delta} t)\ {}_{\Delta} t \ ,$$

where, $g_{\Delta}(i_{\Delta}t)$ is an average value of function $g(t)$ on an interval from $i\Delta t$ to $(i+1)\Delta t$, that equals to

$$g_{\Delta}(i_{\Delta}t)=\frac{1}{{}_{\Delta}t}\cdot\int_{i\Delta t}^{(i+1)\Delta t}g(t)dt;$$

where $n=\frac{t}{{}_{\Delta}t}$ is a number of intervals into which the area of integration is divided.

But the value of n becomes constant only at the end of the transient process, therefore

$$n_{\max}=Ent\left[\frac{T_n}{{}_{\Delta}t}\right]+1,$$

where T_n is the time of the transient process.

While analyzing the systems with continuous time a discretization interval should be picked out from the condition of signal reproduction on an output of the converter. Such condition for processes with limited spectrum is the Kotelnikov's theorem, which asserts, that a process can be exactly reproduced if it is presented by a series of discrete values with an interval

$${}_{\Delta}t=\frac{p}{w_c}.$$

Let's assume, that a pulse characteristic $g(t)$ of a linear dynamic converter is a determined function, which does not include interval uncertainty. In this case, if entrance process $x(t)$ is submitted as an interval function, it forms interval uncertainty of the initial process $y(t)$. It makes possible to present the linear dynamic model in the following kind:

$$Y_{IR}(t)=\sum_{i=0}^{n-1}g_{\Delta}(i_{\Delta}t)\cdot X_{IR}(t-i_{\Delta}t)\,{}_{\Delta}t\ . \qquad (7.13)$$

The formula (7.13) and some of algorithms of numerous integration make it possible to construct a model of a linear dynamic converter, the algorithm of which is presented in Figure 7.2.

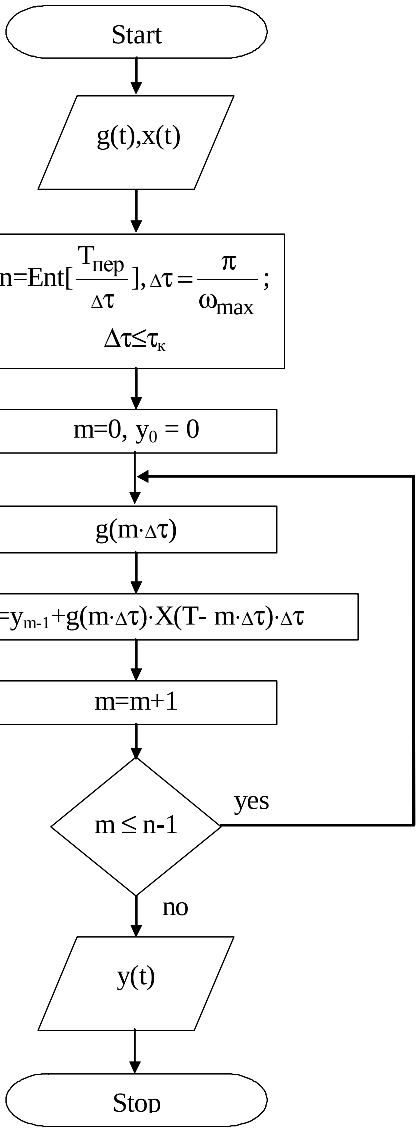


Figure 7.2.

There are two possible ways to build models of nonlinear dynamic converters. First, a certain nonlinear dynamic converter can be considered as a consecutive connection of a nonlinear static and a linear dynamic converter. When it is difficult or just impossible to perform, a nonlinear dynamic converter could be described by models of Wiener or of Hammerstein:

- Wiener's model:

$$Y(t) = W\left[\int_0^t X(t-t)\cdot g(t)\,dt\right];$$

- Hammerstein's model:

$$Y(t) = \int_0^t W\left[X(t-t)\right]\cdot g(t)\,dt \;;$$

- Mixed model:

$$Y(t) = W_1 \left\{ \int_0^t W_2 \left[X(t-t) \right] \cdot g(t)\, dt \right\}.$$

Developed classes of static and dynamic converters, modules of interval computation and methods of aggregation and transformation of such models give us an opportunity to modulate complex systems with uncertainties and ambiguities.

7.5 Remarks

The interval analysis is a more simple method of the complex systems modelling in undetermined conditions in comparison with the probability analysis and fuzzy logic. Traditionally it is used for performing calculations in applied mathematics with errors consideration. This method is preferable for analysis and modelling of complex systems, providing the estimation in terms of interval analysis is sufficient for performance of the researches and for resolving of the arising problems. In this chapter only basics of interval mathematics were described. To study the matter deeper more detailed textbooks could be recommended.

Exercises

1. Reduce each of the following to a single interval

$a)$ $[0;4]+[-2;1];$

$b)$ $[-3;-1]+[-7;11];$

$c)$ $[-2;30]+[-2;4];$

$d)$ $[1;7]-[1;9];$

$e)$ $[-3;0]-[0;3];$

$f)$ $[-8;-4]-[-6;-4];$

$g)$ $[0;2]\cdot[1;8];$

$h)$ $[-1;0]\cdot[-2;-1];$

$i)$ $[-3;5]\cdot[1;3];$

$j)$ $[-8;-7]\cdot[7;8];$

$k)$ $[-7;-4]\cdot[-4;4];$

$l)$ $[2;3]\div[3;5];$

$m)$ $[1;3]\div[-2;-1];$

$n)$ $[-2;2]\div[-2;-1];$

$o)$ $[-3;-1]\div[-2;-1];$

$p)$ $[1.243;2.687]-[6.6;6.6];$

$q)$ $[1.45;2.00]\div[3.19;3.20].$

2. Reduce each of the following to a single interval:

$a)\ \dfrac{[1;3]-4}{[2;3]};$ $\qquad b)\ 2\cdot[1;4]-3\cdot[-1;0];$

$c)\ \dfrac{2\cdot[-3;-1]-3\cdot[1;2][2;3]}{[-4;-3]+2\cdot[5;7]};$ $\qquad d)\ [2;3]^3;$

$e)\ [-2;3]^3;$ $\qquad f)\ [-3;-2]^3;$

$g)\ [2;3]^4;$ $\qquad h)\ [-2;3]^4;$

$i)\ [-3;-2]^4;$ $\qquad j)\ \dfrac{[1;3]-[-1;4]+2[-2;3]^3}{3+[-2;1]^2}.$

3. Prove the theorems 7.3 and 7.4.

4. In each of the following, the numbers have been rounded to the given number of decimal places. Use the interval arithmetic to compute each expression and give an error bound for each result:

$a)\ 71.22+64.35;$ $\qquad b)\ 71.22-83.764;$

$c)\ 71.22+64.3-83.7;$ $\qquad d)\ 71.22-(6.6)^2;$

$e)\ \dfrac{78.4-9.64}{40.0};$ $\qquad f)\ \dfrac{78.4}{9.64-40.2};$

$g)\ \dfrac{(89.5)(20.6)-(34.97)}{(2.875)+(7.6)(4.5)^2}.$

5. Solve each of the following by means of the interval arithmetic assuming that all the given numbers have been rounded to the indicated number of decimal places:

$a)\ \ y_{i+1}=y_i+2(\Delta x)x_i,\ i=0,1,2;$

$$x_0=0.33,\qquad \Delta x=0.33,$$
$$x_1=0.66,\qquad y_0=1.12,$$
$$x_2=0.99.$$

$b)\ \ y_{i+1}=\dfrac{y_i-2(\Delta x)^2}{1+(x_i)^2},\ i=0,1,2;$

$$x_0=0.0,\qquad \Delta x=0.33,$$
$$x_1=0.3,\qquad y_0=2.1,$$
$$x_2=0.7.$$

c) $y_{i+1} = \sin(x_i y_i),\ i = 0, 1, 2$;

$$x_0 = 1.42137, \qquad \Delta x = 1.00000,$$
$$x_1 = 2.42137, \qquad y_0 = 30.0,$$
$$x_2 = 3.42137.$$

6. Name and determine the modelling order of the complex systems using the interval arithmetic.

Bibliography

1. Abramowitz M., Stegun I. Handbook of Mathematical Functions. – N.Y.: National Bureau of Standards, 1964. - 825 p. / Абрамовиц М., Стиган И. Справочник по специальным функциям. – М.: Наука, 1979. – 830 с.
2. Bjork A., Dahlguist G. Metode numeryczne. – Warszawa: Panstw. wydawn. nauk., 1987. – 546 s.
3. Greenspan D. Introduction to Numerical Analysis and Applications. – Chicago: Marcham Publishing Company, 1971. – 182 p.
4. Collatz L. Funktionalanalysis und Numerishe Mathematik. – Berlin - Gottingen-Heidelberg, 1964. - 446 p. / Коллатц Л.Р. Функциональный анализ и вычислительная математика. – М.: Мир, 1969. – 447 с.
5. Демидович Б.П., Марон И.А. Основы вычислительной математики. – М.: Наука, 1970. – 664 с.
6. Дубовой В.М. Моделювання систем контролю та керування. - В.: ВНТУ, 2005. – 174 с.
7. Квєтний Р.Н. Методи комп’ютерних обчислень – В.: ВДТУ, 2001. – 146 с.
8. Квєтний Р.Н., Кострова К.Ю., Богач І.Р. Інтерполяція самоподібними множинами. – Універсум: Вінниця, 2005. – 100 с.
9. Маликов В.Т., Кветный Р.Н. Вычислительные методы и применение ЭВМ. – К.: Вища школа, 1989. – 211 с.
10. Самарский А.А. Введение в численные методы. – М.: Наука, 1987. – 286 с.
11. Скурихин В.И. и др. Математическое моделирование. – К.: Техника, 1983. – 270 с.
12. Чабан В. Чисельні методи. – Львів: Вид. НУ «Львівська політехніка», 2001. – 186 с.
13. Шокин Ю.И. Интервальный анализ. – Новосибирск «Наука», 1981. – 112 с.

Concise English-Russian-Ukrainian-Polish Dictionary of Terms

A

e **antiderivative function** r первообразная функция; u первісна функція; p funkcja pierwotna

e **approach** r приближение; u наближення; p zbliżać sie

e **approximation** r аппроксимация; u апроксимація; p aproksymacja

e **axis** r ось; u вісь; p oś

B

e **backward difference** r левая разность; u ліва різниця; p różnica wsteczna funkcji

e **boundary-value problem** r краевая задача; u крайова задача; p zagadnienie brzegowe

e **bulge of the function** r выпуклость функции; u випуклість функції; p wypukłość funkcji

C

e **central difference** r центральная разность; u центральна різниця; p różnica centralna

e **column** r столбец; u стовпець; p kolumna

e **convergence** r сходимость; u збіжність; p zbieżność

e **curve** r кривая; u крива; p krzywa

D

e **data processing** r обработка данных; u обробка даних; p przetwarzanie danych

e **derivative** r производная; u похідна; p pochodny

e **determination** r детерминированность (определенность); u детермінованість (визначеність); p wyznaczanie, określanie

e **difference** r разность; u різниця; p różnica

e **distribution error** r ошибка распространения; u похибка розповсюдження; p pomyłka dystrybucji

e **distribution law** r закон распределения; u закон розподілення; p prawo podziału

e **divergence** r расходимость; u розбіжність; p rozbieżność, dywergencja

E

e **equation** r уравнение; u рівняння; p równanie

e **error** r ошибка; u похибка; p pomyłka

e **exactness** r точность; u точність; p ścisłość, dokładność

e **exception** r исключение; u виключення; p wyjątek

e **expectation** r ожидание; u очікування; p wartość oczekiwana

F

e **faithful number** r правильное число; u вірне число; p numer wierny

e **finite difference** r конечная разность; u кінцева різниця; p różnica ograniczona

e **firmness (stability)** r устойчивость; u стійкість; p stabilność, stałość, trwałość

e **forward difference** r правая разность; u права різниця; p różnica progresywna funkcji

e **function of belonging** r функция принадлежности; u функція належності; p przynależna funkcja

e **fuzzy logic** r нечеткая логика; u нечітка логіка; p logika rozmyta

H

e **hierarchical** r иерархический; u ієрархічний ; p hierarchiczny

I

e **integer number** r целое число; ціле число; p liczba całkowita

e **interpolation** r интерполяция; u інтерполяція; p interpolacja

e **interpolation 'ahead'** r интерполяция «вперед»; u інтерполяція «вперед»; p interpolacja «naprzód»

e **interpolation 'back'** r интерполяция «назад»; u інтерполяція «назад»; p interpolacja «nazad»

K

e **knot** r узел; u вузол; p węzeł

L

e **least square method** r метод наименьших квадратов; u метод найменших квадратів; p metoda najmniejzych kwadratów

e **limitation (truncation) error** r ошибка ограничения; u похибка обмеження; p pomyłka ograniczenia

M

e **method of ‘shooting’** r метод «стрельбы»; u метод «стрільби»; p metoda «strelanie»

e **multidimensional net** r многомерная сетка; u багатовимірна сітка; p wielowymiarowy siatka

N

e **net** r сетка; u сітка; p siatka, sieć

P

e **partial derivative** r частная производная; u частинна похідна; p pochodna cząstkowa

e **plural** r множество; u множина; p liczba mnoga

e **probability** r вероятность; u вірогідність; p prawdopodobieństwo

e **probability density** r плотность вероятности; u щільність вірогідності; p gęstość prawdopodobieństwa

R

e **random process** r случайный процесс; u випадковий процес; p proces stochastyczny, proces losowy

e **rigid task** r жесткая задача; u жорстка задача; p zadanie twarde

e **root** r корень; u корінь; p pierwiastek

e **row** r ряд; u ряд; p rząd

S

e **secant** r секущая; u січна; p sieczna

e **selfsimilar plural** r самоподобное множество; u самоподібна множина; p samopodobna mnoga

e **selfstarting** r самостартование; u самостартування; p samoczynny ruch

e **sequence** r последовательность; u послідовність; p ustalać kolejność

e **set** r группа; u група; p zbiór

e **share** r слой; u шар; p lemiesz

e **simple iteration** r простая итерация; u проста ітерація; p iteracja prosta

e **simulation** r имитационное моделирование; u імітаційне моделювання; p symulacja, modelowanie

e **simultaneous displacement** r одновременная подстановка; u одночасна підстановка; p jednoczesne przemieszczenie

e **statistical processing** r статистическая обработка; u статистична обробка; p statystyczna obróbka

e **successive overhead relaxation** r последовательная верхняя релаксация; u послідовна верхня релаксація; p kolejna napowietrzna relaksacja

T

e **tangent** r касательная; u дотична; p styczna

e **template** r шаблон; u шаблон; p szablon

e **transaction (digitization) error** r ошибка преобразования (дискретизации); u похибка перетворення (дискретизації); p pomyłka transakcji

e **transformer** r преобразователь; u перетворювач; p transformator

e **tridimensional space interpolation** r трехмерная интерполяция в пространстве; u трьохвимірна інтерполяція в просторі; p interpolacja przestrzeń trójwymiarowa

U

e **uncertainty** r неопределенность; u невизначеність; p niepewność

e **unidimensional net** r одномерная сетка; u одновимірна сітка; p siatka jednowymiarowa

V

e **variable** r переменная; u змінна; p zmienna

e **vicious position** r ложное положение; u хибне положення; p zjadliwe położenie

Навчальне видання

Роман Наумович Квєтний

ОСНОВИ МОДЕЛЮВАННЯ ТА ОБЧИСЛЮВАЛЬНИХ МЕТОДІВ

Навчальний посібник

Оригінал-макет підготовлено автором
Редактор В.О. Дружиніна

Навчально-методичний відділ ВНТУ
Свідоцтво Держкомінформу України
серія ДК № 746 від 25.12.2001
21021, м. Вінниця, Хмельницьке шосе, 95, ВНТУ

Підписано до друку
Формат 29,7х42 1/4
Друк різографічний
Тираж 150 прим.

Гарнітура Times New Roman
Папір офсетний
Ум. друк. арк.

Зам. №

Віддруковано в комп'ютерному інформаційно-видавничому центрі
Вінницького національного технічного університету
Свідоцтво Держкомінформу України
серія ДК № 746 від 25.12.2001
21021, м. Вінниця, Хмельницьке шосе, 95