Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Емблема кафедри (за наявності)

Department of Computing Technics

ALGORITHMS AND METHODS OF CALCULATIONS

Syllabus

Requisites of the Course

Cycle of Higher Education	***First cycle of higher education (Bachelor's degree)***
Field of Study	*12 Information Technologies*
Speciality	*123 Computer Engineering*
Education Program	*Computer Systems and Networks*
Type of Course	*Normative*
Mode of Studies	*full-time*
Year of studies, semester	*2 year (spring semester)*
ECTS workload	*4 credits (ECTS). 36 hours of lectures, 18 hours of laboratory work and 66 hours of self-study.*
Testing and assessment	*3 semester – Final test*
Course Schedule	*1 class per week by the timetable http://rozklad.kpi.ua/*
Language of Instruction	*English*
Course Instructors	Lecturer: *Doctor of Technical Sciences, Professor, Mykhailo Anatoliyovych Novotarskyii* Teacher of laboratory work: *assistant, Artem Mikolaevich Ponomarenko*
Access to the course	

Outline of the Course

1. Course description, goals, objectives, and learning outcomes

The academic discipline "Algorithms and methods of calculations" is designed to provide basic mathematical training of students in the field of methods for solving scientific and engineering problems and programming, taking into account the peculiarities of the algorithmic implementation for numerical methods.

The subject "Algorithms and methods of calculations" has the code PO9 in the list of components of the educational program and belongs to the cycle of professional training.

Within the first part of the course the concept of algorithm, basic properties and types of algorithms are considered, three main types of universal algorithmic models are studied: recursive functions, Turing machine and normal Markov algorithms.

The second part of the course studies methods of the table function interpolation, methods for solving nonlinear equations, basic properties of matrices, vectors and determinants, methods for solving systems of linear algebraic equations, approximate calculation of integrals, numerical solution of Cauchy problem for ordinary differential equations and numerical methods for solving boundary value problems for differential equations with partial derivatives.

The purpose of the "Algorithms and methods of calculations" course is study of modern methods and technologies for developing and evaluating algorithms, fundamental training of students in choosing and using algorithm methods. The calculation methods should be resistant to errors, effective for calculating mathematical problems, analysis of approximate solutions, creating highly efficient algorithms and programs that take into account the peculiarities of the implementation of calculations, and promoting the development of logical and analytical thinking of students.

The subject of the academic discipline consists of:

- methods of algorithm analysis;

- methods for determining the computability and solvability of functions;
- Numerical methods for solving mathematical problems.

According to the requirements of the educational program, students after mastering the credit module must demonstrate the following learning outcomes:

Knowledge: basic calculation methods and corresponding effective algorithms for solving mathematical problems on a PC; calculation of approximate values and interpolation of functions, solution of systems of linear algebraic equations and nonlinear equations, numerical integration and differentiation, solution of differential equations with choice of method. Use of splines in interpolation of tables of values of functions and calculation of integrals. Analysis and processing of experimental results and a posteriori estimates of errors.

Ability to choose and justify the use in practice of certain calculation methods that are resistant to errors and most effective in their practical implementation on a PC.

Experience: the student must know the basic principles of developing algorithms and software for solving problems on a PC. He should have skills to explore computational algorithms, identify their advantages and disadvantages, choose the optimal algorithms for solving problems, data processing and develop programs for solving problems; perform analysis and processing of the results of problem solving, use optimization methods, a posteriori estimates of errors in solving problems by numerical methods.

2. Prerequisites and post-requisites of the course (the place of the course in the scheme of studies in accordance with curriculum)

The material of the discipline is related to the materials studied in the courses "Higher Mathematics: Differential Calculus, Linear Algebra", "Programming" and "Discrete Mathematics".

The knowledge and practical skills acquired in this discipline can be applied in the study of the following courses: "Computer Modeling", "Fundamentals of Software Engineering", "System Programming", "Distributed Computing Technology", "Network and Information Technology" and other.

3. Content of the course

The "Algorithms and methods of calculations" academic discipline includes the study of the following topics.

Section 1. Fundamentals of algorithm theory

Topic 1.1. Introduction. Main methods and tasks, connection with other disciplines.

The main sections and issues to be explored. Contents of individual and laboratory works. Features of laboratory work and individual tasks, requirements for them.

Topic 1.2. The concept of algorithm, properties and methods of algorithm setting.

Topic 1.3. Measures of algorithm complexity. Problem classes P and NP. Fundamentals of algorithm analysis.

Topic 1.4. Error. Classification, sources and rules for calculating errors. Significant cypher, rules for counting numbers according to Bradys, the general formula for error.

Topic 1.5. Formalization of the concept for algorithm. Universal models of algorithms. Recursive functions. Primitively recursive functions. Superposition operator. Primitive recursion operator. Partially recursive functions. Minimization operator.

Topic 1.6. Turing machine. Functions, which are calculated by Turing.

Topic 1.7. Normal Markov algorithms. Equivalence of different universal algorithmic models. Markov substitutions. Normal algorithms that use words. Computed Markov function. The principle of Markov normalization. Coincidence of the class of all normally computed functions with the class of all functions computed by Turing.

Section 2. Fundamentals of numerical methods

Topic 2.1. Interpolation and the problem of interpolation. Generalized polynomials. Interpolation by algebraic polynomials. Lagrange interpolation polynomial. Lagrange interpolation polynomial for equidistant nodes. Inverse interpolation. Newton interpolation polynomial. Spline interpolation. Trigonometric interpolation.

Topic 2.2. Methods of numerical differentiation and integration. General procedure for numerical differentiation. A direct way to calculate a definite integral. Methods for approximate calculation of integrals. Newton-Cotes methods: rectangle method, trapezoidal method, parabola formula (Simpson's formula). Methods of spline integration.

Topic 2.3. Methods for solving nonlinear equations. Problem statement, stages of approximate solution of nonlinear equations. The method of half division. Proportional parts method (chord method). Newton method (tangent method). Modified Newton method. Combined method, iteration method, modified iteration method.

Topic 2.4. Solving systems of linear algebraic equations. Types of systems of linear algebraic equations. Compatible and incompatible systems of algebraic equations. The set of solutions of a algebraic equation system. The extended matrix of the system of linear algebraic equations. Rank of the matrix (system). Base minor. Kronecker-Capelli theorem. Systems of linear algebraic equations with a square matrix. Homogeneous systems of linear algebraic equations. Fundamental system of solutions of a system of homogeneous linear algebraic equations. General solution of an inhomogeneous system of linear equations. Gaussian exclusion method. Gauss-Jordan method. Square root method. Running method.

Topic 2.5. Iterative methods for solving systems of algebraic equations. Accurate and iterative methods. General scheme of construction of iterative methods. Construction of the iterative process. Simple iteration method in coordinate form. Modification of the iteration method. A sufficient condition for the convergence of the iteration process in coordinate form. Jacobi method in coordinate form. Gauss-Seidel method in coordinate form. Method of relaxation in coordinate form. Canonical form of recording one-step iterative methods. Generalized solution of a system of linear equations. Stationary and non-stationary one-step methods. Simple iteration method (Jacobi method). Gauss-Seidel method.

Topic 2.6. Solving systems of nonlinear algebraic equations. Determining the solution of systems of nonlinear equations. Simple iteration method. Convergence of the iterative process. Checking the convergence condition. Seidel method. Seidel convergence condition. Newton method. Simplified Taylor series for an arbitrary function. Representation of a function by Taylor series. Formation of a matrix of equations. Checking the achievement of accuracy by the iterative process. Algorithm for solving a system of nonlinear equations by Newton method.

Topic 2.7. Numerical solution of differential equations. Partial solution of a differential equation. General solution of the differential equation. Cauchy problem (problem with initial data), boundary value problem or problem with boundary conditions. One-step methods. Euler method, refined Euler method. Runge-Kutta method. Multi-step methods. Explicit Adams method (Adams-Bashfort method). Implicit Adams method (Adams-Multon method). Comparison of Adams and Runge-Kutta methods. Prognosis and correction methods (predictor-corrector methods). Prediction and correction method based on the Adams method of the fourth order. Milne method. Hamming method.

Topic 2.8. Numerical methods for solving the boundary value problem for the ordinary differential equation (ODE). Finite-difference method for solving a boundary value problem with boundary conditions of the first kind. Finite-difference method for solving a boundary value problem with boundary conditions of the third kind. Approximation of the differential equation by finite differences in general, approximation of boundary conditions. Running method for solving systems with a three-diagonal matrix.

Topic 2.9. Solving partial differential equations. Classification of differential equations with partial derivatives, types of equations (elliptical, parabolic, hyperbolic, mixed). The concept of the method of finite differences. Applying the finite difference method to solving parabolic equations. Finite-difference approximation of equations of hyperbolic type. Finite-difference approximation of the third boundary value problem for the Poisson equation in a rectangle. Solving the Dirichlet problem by the grid method (finite difference method).

Topic 2.10. Iterative asynchronous methods. Method of asynchronous iterations. Method of asynchronous iterations with fixed points.

4. Training materials and resources

Basic:

1. Markov A.A., Nagorny N.M. The Theory of Algorithms. – Springer. – 1988. – 393 p.
2. Klein S.T. Basic Concepts in Algorithms. – World Scientific Publishing. –2021. – 365 p.
3. Steffensen J.F. Interpolation: Second Edition. – Dover Publications. –2006. –272 p
4. Kelley C.T. Solving Nonlinear Equations with Newton's Method. – Society for Industrial and Applied Mathematics. –1987. –118 p.
5. McMullen Ch. Systems of Equations: Substitution, Simultaneous, Cramer's Rule: Algebra Practice Workbook with Answer. –Zishka Publishing. –2015. – 367 p.
6. Pollard H., Tenenbaum M. Ordinary Differential Equations. – Dover Publications. –1985. –832 p.
7. Benton D.J. Numerical Calculus: Differentiation and Integration. – Independently published . –2018. – 109 p
8. Evans L.C. Partial Differential Equations: Second Edition. – American Mathematical Society. –2003. –749 p.

Additional:

1. Björck A. Dahlquist G. Numerical Methods. – Dover Publications. –1985. –592 p.
2. Solomon J. Numerical Algorithms: for Computer Vision,Machine Learning and Graphics. – A K Peters/CRC Press. –2015. – 400 p.
3. Kiusalaas J. Numerical Methods in Engineering with Python 3 . – Cambridge University Press. – 2013. – 432 p.
4. Bayen A., Siauw T., Kong Q. Python Programming and Numerical Methods: A Guide for Engineers and Scientists. –Academic Press. –2020. –480 p.
5. Canale R., Chapra S. Numerical Methods For Engineers. –McGraw-Hill Education. –2020. –1008 p.

Educational content

5. Methodology

Profile for topics

Names of sections and topics	Number of academic hours				
	Total	including			
		Lectures	Practice	Laboratory work	Self-study
SECTION 1. THEORY OF ALGORITHMS	0				
Topic 1.1. Introduction. Methods and tasks of the course, connection with other disciplines. The main sections and issues to be explored. Contents of independent and laboratory works. Features of laboratory work and individual tasks, requirements.	6	1		2	4
Topic 1.2. The concept of algorithm, properties and methods for algorithm setting.	7	1		1	5
Topic 1.3. Measures of complexity of algorithms. Task classes P and NP	9	2		2	5

Names of sections and topics	Number of academic hours				
	Total	including			
		Lectures	Practice	Laboratory work	Self-study
Topic 1.4. Error. Classification, sources and rules for calculating errors. Significant cypher, rules for counting numbers according to Bradys, the general formula for error**.**	9	2		2	5
Test work 1	3				3
Topic 1.5. Formalization of the concept of algorithm. Universal models of algorithms. Recursive functions. Primitively recursive functions. Superposition operator. Primitive recursion operator. Partially recursive functions. Minimization operator.	5	2			3
Topic 1.6. Turing machine. Functions calculated by Turing machine.	5	2			2
Topic 1.7. Normal Markov algorithms. Equivalence of different universal algorithmic models. Markov substitutions. Normal algorithms, which use words. Computed Markov function. The principle of Markov normalization. Coincidence of the class of all normally computed functions with the class of all functions computed by Turing.	5	2			3
SECTION 2. CALCULATION METHODS	0	0		0	0
Topic 2.1. Interpolation and the problem of interpolation. Generalized polynomials. Interpolation by algebraic polynomials. Lagrange interpolation polynomial. Lagrange interpolation polynomial for equidistant nodes. Inverse interpolation. Newton interpolation polynomial. Spline interpolation. Trigonometric interpolation.	12	4		2	5
Topic 2.2. Methods of numerical differentiation and integration. General procedure for numerical differentiation. A direct way to calculate a definite integral. Methods of approximate calculation of integrals. Newton-Cotes methods: rectangle method, trapezoidal method, parabola formula (Simpson's formula). Methods of spline integration**.**	7	2		2	3
Topic 2.3. Methods for solving nonlinear equations. Problem statement, stages of approximate solution of nonlinear equations. The method of half division. Proportional parts method (chord method). Newton method (tangent method). Modified Newton method. Combined method, iteration method, modified iteration method.	7	2		2	3
Topic 2.4. Solving systems of linear algebraic equations. Types of systems of linear algebraic equations. Compatible and incompatible systems of algebraic equations. The set of solutions of a system of algebraic equations. The extended matrix of the system of linear algebraic equations. Rank of the matrix (system). Base minor. Kronecker-Capelli theorem. Systems of linear	7	2		2	3

Names of sections and topics	Number of academic hours				
	Total	including			
		Lectures	Practice	Laboratory work	Self-study
algebraic equations with a square matrix. Homogeneous systems of linear algebraic equations. Fundamental system of solutions of a system of homogeneous linear algebraic equations. General solution of an inhomogeneous system of linear equations. Gaussian exclusion method. Gauss-Jordan method. Square root method. Running method.					
Topic 2.5. Iterative methods for solving systems of algebraic equations. Accurate and iterative methods. General scheme of construction of iterative methods. Construction of the iterative process. Simple iteration method in coordinate form. Modification of the iteration method. A sufficient condition for the convergence of the iteration process in coordinate form. Jacobi method in coordinate form. Gauss-Seidel method in coordinate form. Method of relaxation in coordinate form. Canonical form of recording one-step iterative methods. Generalized solution of a system of linear equations. Stationary and non-stationary one-step methods. Simple iteration method (Jacobi method). Gauss-Seidel method.	7	2		2	3
Topic 2.6. Solving systems of nonlinear algebraic equations. Determining the solution of systems of nonlinear equations. Simple iteration method. Convergence of the iterative process. Checking the convergence condition. Seidel method. Seidel convergence condition. Newton method. Simplified Taylor series for an arbitrary function. Representation of a function by the Taylor series. Formation of a matrix of equations. Checking the achievement of accuracy by the iterative process. Algorithm for solving a system of nonlinear equations by Newton method.	6	2			2
Topic 2.7. Numerical solution of differential equations. Partial solution of a differential equation. General solution of the differential equation. Cauchy problem (problem with initial data), boundary value problem or problem with boundary conditions. One-step methods. Euler method, refined Euler method. Runge-Kutta method. Multi-step methods. Explicit Adams method (Adams-Bashfort method). Implicit Adams method (Adams-Multon method). Comparison of Adams and Runge-Kutta methods. Prognosis and correction methods (predictor-corrector methods). Prediction and correction method based on the Adams method of the fourth order. Milne method. Hamming method.	4	2			2
Topic 2.8. Numerical methods for solving boundary value problems for ordinary differential equations (ODE). Finite-difference method for solving a boundary value problem with boundary conditions of the first kind. Finite-difference method for solving a boundary value problem with boundary conditions of the third kind. Approximation of the differential equation by finite	4	2			2

Names of sections and topics	Number of academic hours				
	Total	including			
		Lectures	Practice	Laboratory work	Self-study
differences in general, approximation of boundary conditions. Running method for solving systems with a three-diagonal matrix.					
Topic 2.9. Solving partial differential equations. Classification of differential equations with partial derivatives, types of equations (elliptical, parabolic, hyperbolic, mixed). The concept of the method of finite differences. Apply the finite difference method to solve parabolic equations. Finite-difference approximation of equations of hyperbolic type. Finite-difference approximation of the third boundary value problem for the Poisson equation in a rectangle. Solving the Dirichlet problem by the grid method (finite difference method).	6	4			2
Topic 2.10. Iterative asynchronous methods. Method of asynchronous iterations. Method of asynchronous iterations with fixed points.	6	2			4
Test work 2	4				4
Final test	**6**				**6**
Total in the semester:	**120**	**36**		**18**	**66**

Profile for lectures

№ з/п	The title of the lecture topic and a list of key issues
1	**Introduction. Methods and tasks of the course, connection with other disciplines.** The main sections and issues to be explored. Contents of independent and laboratory works. Features of laboratory work and individual tasks, requirements. **The concept of algorithm, properties and methods of setting algorithms.** Stages of problem solving, algorithm concept (algorithm definition, numerical algorithms, logical algorithms, sequential algorithms, parallel algorithms). General properties of algorithms (discreteness, mass, determinism, elementary steps, efficiency). Ways to set algorithms (verbal, graphical, pseudocode, software). Image of the algorithm in the form of a block diagram (rules of graphic design of the block diagram, linear algorithm, branching algorithm cyclic algorithm).
2	**Measures of complexity of algorithms. Problem classes P and NP.** Fundamentals of algorithm analysis. Computational complexity. Asymptotic notation. Comparison of functions. Principles of selection of classes of function growth. Algorithms of polynomial complexity (class P). Asymptotic complecsity $O(n),O(n^2)$. Algorithms of nondeterministic polynomial complexity (class of NP problems). Typical tasks. The salesman problem, the coloring of the graph, the problem of the sum of the elements of subsets, the problem of the truth of KNF-forms.
3	**Error. Classification, sources and rules for calculating errors.** Mathematical model error, input data error, method error, rounding error. Absolute and relative error, the limit of absolute and relative error. Calculation with strict errors. Significant figure, calculation rules without strict consideration of errors. Rules for counting numbers according to Bradys, the general formula for error.

№ з/п	The title of the lecture topic and a list of key issues
4	**Interpolation and the problem of interpolation. Generalized polynomials.** Interpolation by algebraic polynomials. Lagrange interpolation polynomial. Lagrange interpolation polynomial for equidistant nodes. Lagrange interpolation polynomial for non-equidistant nodes. Lagrange interpolation polynomial error for equidistant nodes. Inverse interpolation**.**
5	**Interpolation and the problem of interpolation (continued). Newton interpolation polynomial.** Divvided differences, expressions for divvied differences due to the values of functions. Theorem on the existence of a Newton polynomial for arbitrarily given nodes. Newton polynomial advantage, Newton interpolation polynomial error. Finite differences. Newton interpolation polynomial for equidistant nodes, interpolation polynomial error for equidistant nodes. Spline interpolation. Trigonometric interpolation. Search for coefficients of a trigonometric polynomial.
6	**Methods of numerical differentiation and integration. Basic rules of analytical differentiation.** General procedure for applying numerical differentiation. Application of the numerical differentiation formula based on Newton interpolation polynomial with non-equidistant nodes. Estimates of errors of numerical differentiation. Numerical differentiation with a Newton polynomial with equidistant nodes. Abbreviated formulas for approximate differentiation. A direct way to calculate a definite integral. Methods of approximate calculation of integrals. Newton-Cotes methods: rectangle method, trapezoidal method, parabola formula (Simpson formula). Methods of spline integration.
7	**Methods for solving nonlinear equations.** Problem statement, stages of approximate solution of nonlinear equations. The method of half division. Proportional parts method (chord method). Newton method (tangent method). Modified Newton method. Combined method, iteration method, modified iteration method.
8	**Solving systems of linear algebraic equations.** Types of systems of linear algebraic equations. Compatible and incompatible systems of algebraic equations. The set of solutions of a system of algebraic equations. The extended matrix of the system of linear algebraic equations. Rank of the matrix (system). Base minor. Kronecker-Capelli theorem. Systems of linear algebraic equations with a square matrix. Homogeneous systems of linear algebraic equations. Fundamental system of solutions of a system of homogeneous linear algebraic equations. General solution of an inhomogeneous system of linear equations. Gaussian exclusion method. Gauss-Jordan method. Square root method. Running method.
9	**Iterative methods for solving systems of algebraic equations.** Accurate and iterative methods. General scheme of construction of iterative methods. Construction of the iterative process. Simple iteration method in coordinate form. Modification of the iteration method. A sufficient condition for the convergence of the iteration process in coordinate form. Jacobi method in coordinate form. Gauss-Seidel method in coordinate form. Method of relaxation in coordinate form. Canonical form of recording one-step iterative methods. Generalized solution of a system of linear equations. Stationary and non-stationary one-step methods. Simple iteration method (Jacobi method). Gauss-Seidel method.
10	**Solving systems of nonlinear algebraic equations.** Determining the solution of systems of nonlinear equations. Simple iteration method. Convergence of the iterative process. Jacobi matrix. Checking the convergence condition. Seidel method. Seidel convergence condition. Checking the achievement of accuracy by the iterative process. Newton method. Simplified Taylor series for an arbitrary function. Representation of a function near Taylor. Formation of a matrix of equations. Checking the achievement of accuracy by the iterative process. Algorithm for solving a system of nonlinear equations by Newton method. Conditions of convergence of Newton's method. Verification of convergence conditions for a system of nonlinear equations.
11	**Numerical solution of differential equations.** Partial solution of a differential equation. General solution of the differential equation. Cauchy problem (problem with initial data), boundary value problem or problem with boundary conditions. One-step methods. Euler method, refined Euler method. Runge-Kutta method. Multi-step methods. Explicit Adams method (Adams-Bashfort

№ з/п	The title of the lecture topic and a list of key issues
	method). Implicit Adams method (Adams-Multon method). Comparison of Adams and Runge-Kutta methods. Prognosis and correction methods (predictor-corrector methods). Prediction and correction method based on the Adams method of the fourth order. Milne's method. Hamming method.
12	**Numerical methods for solving the boundary value problem for the ordinary differential equation (ODE**). Finite-difference method for solving a boundary value problem with boundary conditions of the first kind. Finite-difference method for solving a boundary value problem with boundary conditions of the third kind. Approximation of the differential equation by finite differences in general, approximation of boundary conditions. Running method for solving systems with a three-diagonal matrix.
13	**Solving partial differential equations.** Classification of differential equations with partial derivatives, types of equations (elliptical, parabolic, hyperbolic, mixed). The concept of the method of finite differences. Apply the finite difference method to solve parabolic equations. The first boundary value problem for an equation of hyperbolic type. The second boundary value problem for an equation of hyperbolic type. The third boundary value problem for an equation of hyperbolic type. Finite-difference approximation of equations of hyperbolic type. The first boundary value problem for an equation of elliptic type. The third boundary value problem for an equation of elliptic type. Finite-difference approximation of the third boundary value problem for the Poisson equation in a rectangle.
14	**Solving partial differential equations (continued).** Solving the Dirichlet problem by the grid method (finite difference method). Liebman process. Computing template. Preparation of templates**.**
15	**Iterative asynchronous methods. Basic concepts.** Method of chaotic iterations. Method of asynchronous iterations. Method of asynchronous iterations with fixed points
16	**Formalization of the concept of algorithm. Universal models of algorithms.** Computed function. Solvability (solvability). The relationship between computability and solvability. Recursive functions. Primitively recursive functions. Superposition operator. Primitive recursion operator. Partially recursive functions. Minimization operator.
17	**Turing machine.** Functions calculated by Turing machine. Scope of the Turing machine. Mathematical model of the Turing machine. Tape. Reading head. Internal memory of the car. Control device. Turing machine program. The operation of the Turing machine. Examples of Turing machines operating in the alphabet {a, b}. An example of a Turing machine running in the alphabet {a, b, c}. Description of the class of functions calculated by Turing. Examples of functions calculated by Turing.
18	**Normal Markov algorithms. Equivalence of different universal algorithmic models.** The emergence of the theory of normal algorithms. Markov substitutions. Normal algorithms and their application to words. Normally calculated functions. An example of an algorithm for a computational function in the extended alphabet. The principle of Markov normalization. Coincidence of the class of all normally computed functions with the class of all functions computed by Turing.

Self-study

The main task of the cycle of laboratory classes is to provide students with the necessary practical skills to develop algorithms and software for solving computational problems on a PC; research of algorithms in terms of computational complexity, mastering the basic methods and algorithms of interpolation of functions, development of algorithms for solving nonlinear equations and systems of linear algebraic equations.

Profile for laboratory work

№	The title of laboratory work	Number of acad. hours
1	Laboratory work № 1. The concept of algorithm. Setting algorithms in the form of block diagrams. - Section 1. Topic 1.1, - Section 1. Topic 1.2.	 1 1
2	Laboratory work № 2. Computational complexity of sorting algorithms. - Section 1. Topic 1.3, - Section 1. Topic 1.4.	 2 2
3	Laboratory work № 3. Interpolation of functions. Interpolation polynomials - Section 2. Topic 2.1, - Section 2. Topic 2.2.	 2 2
4	Laboratory work № 4. Solving nonlinear equations on a computer. - Section 2. Topic 2.3, - Section 2. Topic 2.4.	 2 2
5	Laboratory work № 5. Solving systems of linear algebraic equations. - Section 2. Topic 2.5, - Section 2. Topic 2.6.	 2 2

Profile for lectures

№	The title of the lecture topic and a list of key issues (List of teaching tools, references and tasks for self-study)
1	**Introduction. Methods and tasks of the course, connection with other disciplines.** **Tasks for self-study.** Structuring algorithm schemes. Requirements for the graphic image of the algorithm, step-by-step specification of the schematic image of the algorithm. The system of notation and rules for recording algorithms. (Lecture 1 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* DSTU. Schemes of algorithms, programs, data and systems. Symbols and rules of execution DSTU. Schemes of algorithms and programs. Rules of execution. DSTU. Schemes of algorithms and programs. Symbols are graphic
2	**Measures of complexity of algorithms. Problem classes P and NP.** **Tasks for self-study.** Determining the computational complexity of the Hoare accelerated sorting algorithm. (Lecture 2 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* T. Cormen, C. Laiserson, R. Rivest: Algorithms: construction and analysis. Ed. Williams, 2013. - 1328 p.
3	**Error. Classification, sources and rules for calculating errors.** **Tasks for self-study.** Sources of errors and their classification. Rounding numbers on a computer. Algorithms of calculations. Errors in calculating the values of functions. Inverse tasks of the theory of errors. (Lecture 3 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab)

№	The title of the lecture topic and a list of key issues (List of teaching tools, references and tasks for self-study)
	References Taylor J. Introduction to Error Analysis. – Univ Science Books; Second 2nd Edition. –1982. – 270 p.
4	**Interpolation and the problem of interpolation. Generalized polynomials. Interpolation by algebraic polynomials.** **Tasks for self-study.** Aitken interpolation polynomial. Finite high order differences. Interpolation formulas of Gauss, Sterling, Bessel, features of application. (Lecture 4 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Steffensen J.F. Interpolation: Second Edition. – Dover Publications. –2006. –272 p
5	**Interpolation and the problem of interpolation (continued). Newton's interpolation polynomial.** **Tasks for self-study.** Spline interpolation, construction of algorithms, application feature. (Lecture 5 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Fraser D.C. Newton's Interpolation Formulas. – Dover Publications. –2006. –272 p
6	**Methods of numerical differentiation and integration.** **Tasks for self-study.** Quadrature formulas for calculating integrals, estimating coefficients, estimating integration errors. Estimation of stability of quadrature formulas to errors of calculation function values. Analysis of error behavior in the interpolation interval. Quadrature formulas (Gaussian formulas). (Lecture 6 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Benton D.J. Numerical Calculus: Differentiation and Integration. – Independently published . –2018. – 109 p
7	**Methods for solving nonlinear equations.** **Tasks for self-study.** Comparison of methods for solving nonlinear equations by the number of iterations. (Lecture 7 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Solving Nonlinear Equations with Newton's Method. – Society for Industrial and Applied Mathematics. –1987. –118 p.
8	**Solving systems of linear algebraic equations.** **Tasks for self-study.** Determination of vectors and matrix norms. Analysis of SLAE solution methods. Estimation of complexity of algorithms and speed of convergence. Running method in SLAE solution, determination of method stability. Analysis of the fastest descent method in solving SLAE. (Lecture 8 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab)

№	The title of the lecture topic and a list of key issues (List of teaching tools, references and tasks for self-study)
	References McMullen Ch. Systems of Equations: Substitution, Simultaneous, Cramer's Rule: Algebra Practice Workbook with Answer. –Zishka Publishing. –2015. – 367 p.
9	**Iterative methods for solving systems of algebraic equations.** **Tasks for self-study.** Solving systems of linear algebraic equations by the method of choosing the principal element, estimating the rate of convergence of the iterative process of solving SLAE. (Lecture 9 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Greebbaum A. Iterative Methods for Solving Linear Systems (Frontiers in Applied Mathematics, Series Number 17) – Society for Industrial and Applied Mathematics –1987 –234 p.
10	**Solving systems of nonlinear algebraic equations.** **Tasks for self-study.** Separation of equation roots, algorithm construction and program testing. Solving systems of nonlinear equations by Newton method, estimating the error of the value of the obtained root, the order of the iterative process. (Lecture 10 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Gilberto E. Urroz, Solution of non-linear equations // at the website https://www.coursehero.com/file/7598891/NonLinearEquationsMatlab/
11	**Numerical solution of differential equations.** **Tasks for self-study.** Influence of step size on Cauchy solution, local and general error. Using the Taylor series in the numerical solution of the Cauchy problem, choosing the sampling step. Methods for improving the accuracy of solving the Cauchy problem using the Lagrange polynomial. Forecast and correction method. Peculiarities of solving the Cauchy problem by the Adams and Milne method. (Lecture 11 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Pollard H., Tenenbaum M. Ordinary Differential Equations. – Dover Publications. –1985. –832 p.
12	**Numerical methods for solving the boundary value problem for the ordinary differential equation (ODE).** **Tasks for self-study.** Stability of difference schemes for solving boundary value problems. Investigation of the method of finite-difference solution of a boundary value problem for stability. (Lecture 12 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References*

№	The title of the lecture topic and a list of key issues (List of teaching tools, references and tasks for self-study)
	Russel R.D., Mattheij R.M.M., Asher U.M. Numerical Solution of Boundary Value Problems for Ordinary Differential Equations (Classics in Applied Mathematics). – Society for Industrial and Applied Mathematics.–1987. –621 p.
13	**Solving partial differential equations.** **Tasks for self-study.** Formulate and solve a boundary value problem in a two-dimensional quadratic domain with boundary conditions of the first kind for an elliptic equation. (Lecture 13 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Roessel H., Leonard I.E., Hillen T. Partial Differential Equations: Theory and Completely Solved Problems.–Wiley .–2012.– 696 p.
14	**Solving partial differential equations (continued).** **Tasks for self-study.** Construct a computational template for an elliptical domain. (Lecture 14 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Pinder G.F. Numerical Methods for Solving Partial Differential Equations: A Comprehensive Introduction for Scientists and Engineers.– Wiley.–2018 .–304 p
15	**Iterative asynchronous methods.** **Tasks for self-study.** To form a difference scheme of calculations by the method of asynchronous iterations on a cross-shaped template. (Lecture 15 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Baudet G.M. Asynchronous iterative methods for multiprocessors.– Carnegie-Mellon University. Dept. of Computer Science .– 1976.– 30 p.
16	Formalization of the concept of algorithm. Universal models of algorithms. Tasks for self-study. Partially computed functions, Gödel function system, Post machine. (Lecture 16 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) References Friedlander G. Algorithm Universe Model 2nd Edition.– Independently published.–2019 .–261 p.
17	Turing machine. Tasks for self-study. Write a program for performing arithmetic operations on a Turing machine. (Lecture 17 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) References

№	The title of the lecture topic and a list of key issues (List of teaching tools, references and tasks for self-study)
	Herken R. The Universal Turing Machine: A Half-Century Survey .– Oxford University Press.– 1992 .–676 p.
18	**Normal Markov algorithms. Equivalence of different universal algorithmic models.** **Tasks for self-study.** Write a Markov algorithm to perform the operation of multiplying two numbers in the decimal number system. (Lecture 18 and textbook in the Google class https://classroom.google.com/c/NDA3ODA0NzU2NjMx?cjc=4o4iqab) *References* Markov A.A., Nagorny N.M. The Theory of Algorithms. – Springer. – 1988. – 393 p.

Policy and Assessment

6. Course policy

During classes in the "Algorithms and methods of calculations" academic discipline students must follow certain disciplinary rules:

• it is forbidden to be late for classes;

• at the entrance of the teacher, as a sign of greeting, persons studying at “KPI Igor Sikorsky” must stand up;

• no extraneous conversations or other noise that interferes with the classes;

• leaving the classroom during the lesson is allowed only with the permission of the teacher.

• it is not allowed to use mobile phones and other technical means without the permission of the teacher.

Laboratory works are handed over personally with preliminary check of the theoretical knowledge, which is necessary for performance of laboratory work. Verification of practical results includes verification of code and performance of test tasks.

During the training, the teacher has the right to receive up to 5 incentive points for early performance of laboratory work, for creative approach in performing an individual task or for active participation in the discussion of issues related to the topic of the lecture or practical lesson.

For the performance and delivery of laboratory work after the specified deadline, for a significant number of missed classes, or for violation of the rules of conduct in the classroom, the teacher may assign up to 5 penalty points.

When conducting control measures and performing laboratory work, graduate students must follow the rules of academic integrity. If a significant percentage of write-offs or plagiarism is detected, the teacher may refuse to accept the work and demand a fair implementation of the curriculum.

7. Monitoring and grading policy

At the first class the students are acquainted with the grading policy which is based on Regulations on the system of assessment of learning outcomes https://document.kpi.ua/files/2020_1-273.pdf

Types of control in the "Algorithms and methods of calculations" discipline include:

-*Laboratory works:* laboratory works

-*Current control:* testing by closed tests.

-*Final test:* closed test or interview with a teacher.

- Bonuses.

Conditions of admission to semester control: semester rating more than 40 points.

At the first class the students are acquainted with the grading policy which is based on Regulations on the system of assessment of learning outcomes https://document.kpi.ua/files/2020_1-273.pdf The student's rating in the course consists of laboratory work scores (R1), current test scores (R2), final test scores (R4) and bonus scores (R4)

Rs=R1+R2+R3+R4=100 scores

The maximum average weight score is equal to:

R1=5 labs x 10 points = 50 points

R2=5 current testpoints x 5 points = 25 points

R3=Final test=25 points

R4=from 0 to 10 points for outstanding achievements in learning and completing additional tasks

According to the university regulations on the monitoring of students' academic progress (https://kpi.ua/document_control) there are two assessment weeks, usually during 7th/8th and 14th/15th week of the semester, when students take the Progress and Module tests respectively, to check their progress against the criteria of the course assessment policy.

The students who finally score the required number of points (≥60) can:

- get their final grade according to the rating score;
- perform personal test or to pass an interview with the teacher in order to increase the grade.

Students whose final performance score is below 60 points but more than 30 are required to complete personal test or to pass an interview with the teacher. If the final grade is lower than the grade, which the student gets for his semester activity, a strict requirement is applied - the student's previous rating is canceled and he receives a grade based on the results of the Fail/Pass test. Students whose score is below 30 are not allowed to take the Fail/ Pass Test.

The final performance score or the results of the Fail/ Pass Test are adopted by university grading system as follows:

Score	***Grade***
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactory
64-60	Sufficient
Below 60	Fail
Course requirements are not met	Not Graded

8. Additional information about the course

Teaching the academic discipline "Algorithms and methods of calculations" for the field of "Computer Engineering" has its own specifics, which is due to the fact that the scope of methods of machine data analysis is constantly expanding. Widespread informatization leads to the accumulation of huge amounts of data in research, production, transport, and health care. Forecasting, control, and decision-making tasks often require machine data analysis and machine learning, as previously such tasks were either not posed at all or solved by precise methods that required large computing power due to the complexity of the respective algorithms.

Syllabus of the course

Is designed by teacher Doc.Sci,professor Mykhailo Novotaerskyi

Adopted by Department of Computing Technics (protocol № 10 , date 25.05.22)

Approved by the Faculty Board of Methodology (protocol № 10, date 09.06.22)