01.04.02 Computational Modeling in Technology and Finance

General Locations Southern Federal University

All Locations Rostov-on-Don, Russia

Duration 2 years

Earliest Start Date March 2023

Application Deadline June 2023

All Languages English

Study Type Campus

Pace Full-time

The best way to enroll is to participate in the portfolio competition. This way is highly recommended. You are requested to present

1. **Motivation letter.**

Motivational letter should disclose the reasons for choosing this master's program and the goals that the applicant sets for himself/herself when studying for this master's program. The recommended size of a cover letter is 1-2 pages.

Criteria of evaluation for a motivational letter:

- **Topics**: 《Why do I want to study at Southern Federal University?》;
- expectation of the goals and expected learning outcomes in the magistracy, plans for scientific activities carried out during training, CV (brief description of life and professional skills);
- justification of the choice of the Southern Federal University as a place of study;
- substantiation of the choice of a master's program and its connection with current or future professional (research) activities, a link to an open repository with previous project or projects.

2. Essay. Essay requirements: No less than 2 pages of the A4 format. Margins: left - 30 mm, right, top, bottom - 20 mm. Font - Times New Roman, 14 pt., line space 1.5.

TOPIC: 《The role of mathematical modeling and information technology in science and technology》.

3. Documents confirming scientific and professional achievements of the applicant (certificates, diplomas, letters of recommendation).

4. Language proficiency certificate: IELTS 5.0-6.0, TOEFL (iBT) 57 – 86 (online version); TOEFL (PBT) 487 – 566 (paper version); Cambridge Tests PET Preliminary.

If you do not score the required number of points (40), you still have an opportunity to enter the university by taking entrance examinations in August, 2023.

Program Description

Overview

Mathematical modelling is an integral part of modern society and science. It finds applications in designing new materials and devices, simulate natural phenomena, forecasting the weather and so on. The importance of mathematical modeling will only increase in future.

The program goal is to prepare specialists in the field of science-based and high-tech production, research, and education

Master's program "Mathematical Modelling and Information Technologies" offers a student an extensive knowledge in such areas as: mathematical modeling in nanomechanics and biomechanics; modelling of biological and water systems; modern computer technologies and data analysis; mathematical theory of pattern recognition and machine learning.

Courses, offered in master's program cover a wide range of areas. Student will gain the skills, which he/she can apply in future career in science or industry. You will gain the fundamental skills of mathematical modeling and using software in qualitative research, which they can apply in a future to real-world problems.

Career prospects

The graduates of our master's program get a Master of Science degree in applied mathematics and computer science. They are equipped with up-to-date research methods and tools, which help them solving R&D problems in IT companies and industry individually or as a part of an international scientific group.

Typical employment opportunities are

- Research institutions, universities
- IT-companies
- R&D departments of industrial companies.

Master's graduation work may be a good starting point for Ph.D. studies. After obtaining an MSc, it is possible to continue studies and apply for admission to a four-year Ph.D. program.

Admission requirements for Master's program

Students must be comfortable with undergraduate-level mathematics: Mathematical analysis, Linear algebra, Probability, and statistics. They also should have Programming experience and acceptable English language qualification. Applicants for the program should have at least a Bachelor's degree in Mathematics, Computer Science, Physics or Engineering.

Facilities

The Institute of Mathematics, Mechanics and Computer Sciences of Southern Federal University has the material and technical base that provides for all kinds of disciplinary and interdisciplinary training, educational laboratories with modern computers and modern licensed software.

Teaching methods

The program consists of a combination of lectures, practical sessions, project work, and seminar discussions. Student performance is assessed through examinations, coursework, and projects.

A particular emphasis is placed on the problem formulation, the analytical and numerical techniques for its solution and the computation of practically oriented results

Master's program offers a student an extensive knowledge in principal areas of applied mathematics:

• Mathematical modeling of natural and industrial objects, systems and processes. Special attention is given to innovative sections of mechanics and biomechanics related to modeling and optimization of parts and devices on the base of new materials.

• Modern numerical methods, techniques and software tools including numerical methods for initial and boundary-value problems for PDE (partial differential equations) based on Galerkin-type approach, FEM (the finite element method), FDM (the finite difference method), etc., as well as the use of software (computer programs) such as MATLAB, Maple, ANSYS, COMSOL, etc.

• Advanced computer technologies and data analysis with a focus on parallel and distributed programming and statistical data processing.

Supervised independent work of students includes elements of research work in the field of mathematical modeling.

Basic modules

"Modern Computer Technologies in Modelling"

To study this academic discipline, knowledge, and skills formed by previous disciplines are required: Programming, Numerical Methods, Differential Equations.

Discipline goals

The study of modern computer technologies and numerical methods including programming skills for solving wide range of mathematical modeling problems.

Knowledge. Basic theoretical knowledge on the principles of constructing and implementing algorithms for solving mathematical modeling problems, methods for parallelizing algorithms, numerical methods for solving systems of algebraic and differential equations, results on the properties of initial-boundary value problems, computational experiment, and visualization of scientific research results.

Skills. Possibility of records of algorithms in algorithmic languages, apparatus for solving problems of mathematical modeling, continuum physics, etc. Discipline supplements and expands knowledge of mathematical physics equations, ordinary differential equations, programming, numerical methods. Programming using OpenMP and MPI technologies for parallelizing algorithms. a computational experiment, solving initial-boundary value problems for partial differential equations, programming algorithms using Matlab environments, visualizing research results using Matlab tools.

The knowledge, skills acquired during the study of this discipline can be used to solve professional problems in research, scientific-production and design activities, in particular, in the preparation of graduation qualification work.

The objectives of mastering the discipline:

• Gaining knowledge of the principles of parallelizing algorithms for solving mathematical modeling problems.

• The formation of basic knowledge in students in the field of data visualization using computer technology.

• Mastering modern numerical methods for solving mathematical modeling problems.

Tasks:

• Study of programming technologies for multiprocessor technology OpenMP and MPI.

• Acquisition of skills in developing methods for solving mathematical modeling problems using multiprocessor technology.

• Formation of skills for the implementation of modern numerical methods.

• Using the tools of the Matlab package to visualize the results of the study of mathematical models.

5. Requirements for the results of mastering the discipline

As a result of mastering the discipline, the student must possess the following knowledge, abilities, and skills:

Knowledge:

• basic concepts of programming technologies OpenMP and MPI;

• techniques for constructing algorithms for solving mathematical modeling problems for parallel computers;

• methods of analysis and visualization of research results of mathematical models;

• basic Matlab system commands.

Skills:

• develop and implement algorithms for solving mathematical modeling problems;

• write programs using the programming technologies OpenMP and MPI in C ++;

• process data using the Matlab package;

• visualize the results of mathematical modeling using the Matlab package.

• Numerical experiment with mathematical models;

• data visualization using the Matlab package;

• explanation of the results of the study of mathematical models.

The process of studying the discipline is aimed at the formation of the following competencies:

Discipline content

Section 1. Principles of developing methods and algorithms for solving mathematical modeling problems on modern computers. The finite-difference, projection and meshless methods and parallel algorithms of their realization. OpenMP and MPI technologies for parallelizing algorithms in the C ++ programming language.

Section 2. The principles of a computational experiment and verification of methods and algorithms. The main methods of analyzing the results of a computational experiment. Key features of the Matlab package. Visualization and animation tools in Matlab. Ways of interfacing various computational approaches and tools.

Questions to be taken for independent study.

• *Commands and functions of the OpenMP and MPI libraries. Principles of constructing of algorithms for parallel computers with common and shared memories.*

• *Commands and functions of the libraries of the Matlab package. Solving different problems of mathematical modeling using Matlab.*

• *Examples of the use of modern computing technologies for the study of mathematical models.*

"Advanced problems of Mathematical physics"

To study this academic discipline (module), knowledge, skills and abilities are required that are formed by the previous disciplines in the undergraduate program "Differential Equations", "Equations of Mathematical Physics", "Functional Analysis", "Complex and Vector Analysis". The knowledge and skills gained in the course of studying this discipline (module) can be used to solve professional problems in research, scientific-production and project activities, in particular, when performing the final qualifying work (master's thesis).

Learning outcomes

developing students' competencies related to scientific research in the field of mathematical physics and its applications.

Tasks:

- based on the skills and abilities obtained in the bachelor's degree, master the methods of modern mathematical physics and be able to apply them to solve research problems;
- the use of modern information technologies in the scientific research of the problem, including the creation of computer programs and / or the use of ready-made software.

Content

Section 1. Transition from discrete models of coupled oscillators to continuous models of a continuous medium

Topic 1.1. Conservative systems with one degree of freedom.

Conservative systems with one degree of freedom. Phase space. Building a phase portrait using the potential energy graph.

Topic 1.2. Wave equations and systems of coupled oscillators.

Derivation of an equation describing a system of coupled oscillators. Transition from a discrete model to a continuous one. Wave equation ..

Topic 1.3. Finding exact and approximate solutions of nonlinear equations.

Soliton equations/

Section 2. Applications of functional analysis to study the properties of solutions to linear and nonlinear equations of mathematical physics.

Topic 2.1. Strongly continuous semigroups. Contracting semigroups.

Determination of the exponent of a linear operator in the finite-dimensional case. Infinite-dimensional case. Restricted operators. Contraction semigroups. Unlimited operators.

Topic 2.2. Applications of functional analysis to partial differential equations.

Lyapunov-Schmidt method. Center manifold theorem.

Topic 2.3. Special functions and their app cations

Cylindrical functions. Spherical functions.l

Section 3. Modern problems of biomechanics

Topic 3.1. Soliton equations in models of natural science

Korteweg-de Vries equation. Sine-Gordon equation.

Topic 3.2. Reaction-diffusion equations

Reaction-diffusion equations. Methods for the study of stability. Turing instability. ...

Topic 3.3. Nerve impulse propagation equations

Rayleigh equations, Fitzhugh-Nagumo equation and its special cases.

"Modern numerical methods in mathematical modeling"

To study this academic discipline, knowledge and skills formed by the discipline “Mathematical Analysis and Normed Spaces”, “Functional Analysis”, “Ordinary Differential Equations”, “Partial Differential Equations”, “Linear Algebra”, “Calculus” and “Informational Technologies” studied in the undergraduate program are required.

The knowledge and skills gained during the study of this discipline can be used to solve professional problems in research, scientific-production and design activities in the development of mathematical models, algorithms and programs in preparation for final qualification work.

Discipline goal

Course “Modern numerical methods in mathematical modeling” deals with advanced concepts and techniques of mathematical modeling in applied mathematics, which are based on differential equations (ODE and PDE). So, the course is dedicated to the overview of modern numerical methods for computer-based solutions of initial and boundary-value problems for ODE and PDE. The subject of the course is focused on the ideas of different numerical methods, analyses of their distinctive features and comparison of their effectiveness for applications. While the major emphasis is on the study of the finite element method (FEM) because of its well-structured computer-oriented scheme, universality, and adaptability for 2D and 3D problems in domains with complicated geometry, the course also draws attention to the finite difference method (FDM) and some popular Galerkin-based techniques: Bubnov method; method of least squares; collocation methods (single and multiple collocation points, collocation in cells), etc.

Requirements for the results of mastering the discipline

As a result of mastering the discipline, the student should have the following knowledge and skills:

Knowledge of:

On successful completion of the course, students are expected to know:

- different approaches to the discretization procedure for the typical initial and boundary-value problems for ODE and PDE
- numerical method for given regular initial and boundary-value problems for ODE and PDE
- regular differential and discrete problems and analyze their correlations
- the influence of calculations errors (both absolute and relative)

Skills

On successful completion of the course, students are expected to be able to:

- describe and compare different approaches to the discretization procedure for the typical initial and boundary-value problems for ODE and PDE
- choose and implement a suitable numerical method for given regular initial and boundary-value problems for ODE and PDE
- ascertain basic properties of regular differential and discrete problems and analyze their correlations

- define and analyze the influence of calculations errors (both absolute and relative)

Discipline content

Section 1. Introduction.

Topic 1.1. Mathematical modeling and computational experiment based on BVP for ODE and PDE

Role of mathematical modeling and computational experiment in natural science and engineering. ODE and PDE as mathematical models. Fundamental PDEs, its classification and basic initial and boundary-value problems. Basic formulas for differential operators, their characteristics and transformations. Operator representation for problems of mathematical physics.

Topic 1.2. Remarks on the theoretical foundations and a reminder of some theoretical facts

Exact solution of PDE. Green operator and its representation as series or integral. Calculation of values of exact solution in fixed points. Numerical solution of PDE vs. its classical exact solution. Dirichlet and Neumann problems for Poisson equation in regular domains. Numerical (MATLAB, Maple) and analytical (Fourier technique and separation of variables) approaches.

Section 2. Modern numerical methods: ideas and approaches

Topic 2.1. Projection approach

Galerkin-type approximation for operator equation (projection approach). Relations between residual of equation and error of solution approximation. Inner, boundary and overall residuals. Numerical methods classification.

Topic 2.2. Method overview, analysis, and comparison

Analysis of weighted residual methods (Bubnov method; method of least squares; collocation methods (single and multiple collocation points, collocation in cells); finite difference method). Testing a set of weighted residual methods for second order ODE on segment (1D two-point boundary-value problem)

FEM in 1D – basic ideas and terminology. Variation and projection approach. FEM as a weighted residual method.

Elliptic problems: classic, generalized, and weak solutions for boundary-value problems. 3D, 2D and 1D examples. Essential and natural boundary conditions. Testing a set of weighted residual methods for second order PDE in rectangle (2D boundary-value problem).

FEM in 2D: domain triangulation; linear triangle finite element; shape functions; bilinear quadrangle finite element. Stiffness matrix and load vector (global and local level). Stiffness matrix for inner and border elements.

Example: Dirichlet problem for Poisson equation in square. Assembling procedure. Essential boundary conditions processing. Nodes numeration problem. Testing a second order PDE in rectangle (2D boundary-value problem) by FEM.

High-order finite elements. Lagrangian and Hermitian finite elements.

FEM for parabolic and hyperbolic problems

Section 3. Modern numerical methods: implementations

Topic 3.1. Solving Dirichlet problem for Poisson equation in square analytically.

Topic 3.2. Solving 1D two-point boundary-value problem by one of the projection methods.

Topic 3.3. Solving 2D Dirichlet problem for Poisson equation in square by one of the projection methods.

Topic 3.4. Solving 2D Dirichlet problem for Poisson equation in square by FEM.

Self-study questions

- Interpolation in h-version Finite Element Spaces.
- The p-version of the Finite Element Method.
- Spectral Methods.
- Adaptive Wavelet Techniques in Numerical Simulation.
- Mixed Finite Element Methods.
- Meshfree Methods.
- Boundary Element Methods: Foundation and Error Analysis.
- Coupling of Boundary Element Methods and Finite Element Methods.
- Arbitrary Lagrangian–Eulerian Methods.
- Finite Volume Methods: Foundation and Analysis.
- Multiscale Finite Element Method for heterogeneous problems: ideas and implementation.
- Discontinuous Galerkin-type methods: ideas and implementation.

"Web Applications for Data Visualization"

To study this academic discipline, knowledge and skills formed by the discipline "Fundamentals of Programming" studied in the undergraduate program are required.

The knowledge and skills gained during the study of this discipline can be used to solve professional problems in research, scientific-production and design activities, in particular, in the development of algorithms and programs and making of scientific reports in preparation for final qualification work.

Discipline goal

The study of modern web technologies and tools for scientific data visualization

Tasks:

- study of modern standards of JavaScript and TypeScript languages;
- acquisition of skills in rich web application development using Angular framework;
- acquisition of skills to analyze, visualize and build reports on scientific data using modern front-end technologies;
- acquisition of skills needed for deploying and maintaining serverless web applica-tions.

Requirements for the results of mastering the discipline

As a result of mastering the discipline, the student should have the following knowledge and skills:

Knowledge of:

- main features of ES6 specification;
- main features of TypeScript language;
- principles of SPA building;
- Angular framework components and patterns;

- data visualization principles and tools;
- D3.js features and possibilities;
- Front-end applications bundling principles;

Skills:

- writing modern JavaScript (ES6) and TypeScript based applications;
- using Angular and Angular CLI for building single page applications;
- using D3.js for data visualization;
- deploying serverless applications to GitHub Pages and maintain code quality in VCS.

Discipline content

Section 1. Modern JavaScript dialects

Topic 1.1. ECMAScript 6 Standard

Basic JavaScript syntax. JavaScript Standards. Constants. Scoping. Arrow Functions. Ex-tended Parameter Handling. Template Literals. Extended Literals. Enhanced Regular Expression. Regular Expression Sticky Matching. Enhanced Object Properties. Destructuring Assignment. Modules. Classes. Symbol Type. Iterators. Generators. Map/Set & WeakMap/WeakSet. Typed Arrays. Promises. Meta-Programming. Internationalization and Localization.

Topic 1.2. Type Script and Bundling

Node.js. NPM. Package Management Tools. Bundling. TypeScript language reference. Differences between TypeScript and JavaScript.

Section 2. Single Page Applications with Angular JS

Topic 2.1. SPA concepts

Types of Web applications. SPA. Data Transfer Techniques. AJAX. Web Sockets, Web Workers. Modern SPA frameworks overview.

Topic 2.2. Angular

Angular concepts. Angular CLI. Angular project structure. Angular Components and Templates. NgModules. Reactive programming with RxJS. Routing and navigation. Forms and validations. Animations. Services and Dependency Injection.

Topic 2.3. Building SPA

Developing and deploying web applications. Isomorphic Applications. Serverless applica-tions.

Section 3. Data Visualization with D3.js

Topic 3.1. Data Visualization Principles

D3.js library reference. D3.js framework bindings. Tables. Charts. Plots. SVG. Graph Visualization. Animations. 3D graphics in browser. Building D3 plugins.

Topic 3.2. Data Driven Research

Hypothesis checking. Basic mathematical statistics instruments for data visualization. Basic data mining for web applications.

Self-study questions

Yarn, Rollup. Canvas Rendering. Performance Measurement for web applications.

"Mathematical models in biology"

To study this academic discipline, knowledge and skills, formed by previous disciplines: Differential equations, Equations of mathematical physics, Functional analysis.

Knowledge. The main theoretical results on the properties of initial-boundary value prob-lems, computational experiment, stability studies and bifurcations of models of bio-logical flows.

Skills. apparatus for solving problems of hydrodynamics, mathematical biology, physics of continuous media, etc. Discipline complements and expands knowledge of the equations of mathematical physics, ordinary differential equations.Computational experiment, solving initial-boundary value problems for parabolic equations, programming algorithms using the Maple and MATLAB environments.

Knowledge and skills obtained during the study of this discipline, can be used to solve professional problems in research, research and production and project activities, in particular, in the preparation of the release qualification work.

Goals and objectives of studying the discipline

Understanding the basic theoretical results on models of mathematical biology, methods for constructing and investigating the continuum and difference models, applying the apparatus of the theory of dynamical systems and the computational experiment, preparation for studies relat-ed to modeling.

Objectives: mastering the concepts necessary for constructing and researching models based on the use of models with continuous and discrete time; study of specific systems describing the dynamics of population systems.

Requirements for the results of the discipline

As a result of mastering the discipline, the trainee must possess the following knowledge and skills:

Knowledge:

The main theoretical results on the properties of initial-boundary value problems, specific for mathematical biology; oriented on the ecological applications of the numerical experiment, methods for studying the stability and bifurcations of models of biological flows.

Skills:

One needs to formulate the initial-boundary value problem, organize a computational experiment, conduct a study of the stability of solutions, and analyze bifurcations of models of biological flows. to use computational packages, to carry out analytical and numerical analysis of differential equations and related mathematical models. Construction mathematical models of biological flows. Application of the hardware of dynamic systems to study scenarios for the development of biological and ecological systems, apply computational packages for modeling.

Content of the discipline

Section 1. Basic concepts of biofluid dynamics.

Introduction to modeling of biological fluxes.

Basic equations for biological media.

Section 2. Methods and Techniques

Analysis of steady and periodic solutions of differential equations.

Numerical modeling elements.

Questions to be taken for independent study.

Models of population systems, application of computational experiment for forecasting dynamics.

"Parallel and distributed programming"

To study this academic discipline, knowledge, skills and abilities are required, formed by the disciplines studied in the undergraduate course "Fundamentals of Programming", "C ++ Programming Language", "Operating Systems".

The knowledge and skills gained while studying this discipline can be used to solve professional problems in research, scientific production and design activities, in particular, in the development of algorithms and programs as part of the preparation of the final qualification work.

Course aims and objectives

Study of modern methods and technologies of parallel and multi-threaded programming.

Tasks:

- study of the basic means of parallel programming: MPI libraries and OpenMP program-ming interface;
- acquaintance with the basic techniques for parallelizing computational tasks;
- acquisition of skills in creating, running and debugging parallel programs using specialized tools;
- acquisition of user and programming skills for working with a computing cluster.

Intended learning outcomes

As a result of mastering the discipline, the student must have the following knowledge, skills and abilities:

Knowledge:

- main paradigms of interaction of computing processes;
- principles of functioning of parallel programs;
- the basic principles of the technology for the development of parallel programs based on message transfer;
- the main features of the MPI library related to message forwarding and collective interaction of parallel application processes;
- additional capabilities of the MPI library related to the definition of new user data types and their packing, working with process groups and communicators, using virtual topologies;
- principles of technology for the development of parallel programs based on the use of shared memory;
- OpenMP software interface and Parallel LINQ technology;

- basic parallel algorithms related to numerical integration (grid algorithms), processing matrices and systems of linear equations (matrix algorithms), modeling the interaction of sets of particles (point algorithms).

Abilities:

- use modern software implementations of the MPI library (in particular, the MPICH soft-ware package) for the development of parallel programs;
- use various tools of the MPI library to solve typical tasks related to parallel programming;
- use various tools of the OpenMP interface to solve typical tasks related to parallel programming;
- work with a computing cluster;
- develop educational tasks for parallel programming.

Skills:

- in the use of standard tools for the development and launch of parallel programs.

Course content

Section 1. Introduction to the parallel calculations. Developing of distributed ap-plications by means of the MPI technology

Topic 1.1. Overview multiprocessor systems programming tools. MPI Technology: a general description

Basic architectures of multiprocessor systems: shared memory systems, distributed memory systems. The methodology for developing a parallel program (Y. Foster). Examples of implementations of parallel programs to calculate a definite integral: branching (fork / wait), mul-tithreading based on POSIX Pthread and Thread object, OpenMP library, distributed computa-tions based on the PVM and MPI interfaces, High Performance Fortran language (HPF). The his-tory of development and the versions of the MPI interface (Message Passing Interface). The basic MPI notions: a process, a communicator, a message sender and a message receiver, MPI runtime. The MPICH system and the electronic book of educational training tasks Programming Taskbook for MPI.

Topic 1.2. Point-to-point communication. Collective communication

Six basic functions of MPI. Timing functions. Blocking and non-blocking point-to-point communications. Communication modes: the standard mode, synchronous mode, the buffered mode, the ready mode. Receiving messages and related functions including MPI_Probe and MPI_Get_count. Deadlocks during the blocking communications and how to avoid them. A non-blocking receiving messages and the Wait and Test family of functions. Receiving messages from any source (MPI_ANY_SOURCE) and receiving messages with any tag (MPI_ANE_TAG). Combined send-receive operations and their features. Collective communica-tions and their main differences with the point-to-point communications. Barrier synchronization via the MPI_Barrier function. The types of collective communications: broadcasting communica-tion MPI_Bcast, gathering data to one member MPI_Gather and MPI_Gatherv, gathering data where all members receive the result MPI_Allgather and MPI_Allgatherv, scattering data to all members MPI_Scatter and MPI_Scatterv, combined scattering / gathering data from all to all members MPI_Alltoall and MPI_Alltoallv. Collective communications with a varying count of data for each process, their features.

Topic 1.3. Global reduction operation. Derived datatypes. Packing and unpacking

The predefined reduction operations, the specific features of the MPI_MINLOC and MPI_MAXLOC operations. The types of possible reduction operations: receiving the result data in one process MPI_Reduce, receiving the result data in all processes MPI_Allreduce, combined reduction and scattering the results data to all processes MPI_Reduce_scatter, the prefix reduction MPI_Scan. Derived MPI datatypes and their characteristics: the extent (the MPI_Type_extent function) and the size (the MPI_Type_size function). Datatype constructors: replication of a datatype into contiguous locations (MPI_Type_contiguous), replication of a datatype into loca-tions that consist of equally spaced blocks (MPI_Type_vector and MPI_Type_hvector); replica-tion of a datatype into a sequence of blocks, where each block can contain a different number of copies and have a different displacement (MPI_Type_indexed and MPI_Type_hindexed); replica-tion of a datatype into a sequence of blocks, where each block consists of different datatypes (MPI_Type_struct). Derived datatypes committing and releasing. Using the derived datatypes for the efficient sending of two-dimensional data blocks. Packaging and unpackaging data via the MPI_Pack and MPI_Unpack functions, their features. Sending of packed data (the MPI_PACKED datatype). Packaging and unpackaging data via the MPI_Pack and MPI_Unpack functions, their features. Sending of packed data (the MPI_PACKED datatype). Packaging and unpackaging data via the MPI_Pack and MPI_Unpack functions, their features. Sending of packed data (the MPI_PACKED datatype).

Topic 1.4. Groups of processes and communicators. Virtual topologies

Predefined groups of processes; group constructors: create a new groups of processes on the basis of an existing ones or on the basis of the communicator, using a set operations for groups. Predefined communicators; communicator constructors: create a new communicator on the basis of existubg groups and using the MPI_Comm_split function for splitting of the existing communicator. Communicators' comparing. The use of new communicators for execution of col-lective operations within the part of existing processes. Communicators with the topology. Types of topologies: the Cartesian topology and the graph topology. Constructors for communicators with the topology. Cartesian topology with periodicity in some dimensions. Additional features of communicators with the Cartesian topology: rank-to-coordinates and coordinates-to-rank translation, the data shift for a given coordinate (non-periodic or periodic), partitioning of Carte-sian communicator into communicators with low-dimensional Cartesian topology. Additional features of communicators with the graph topology: providing the information about the neigh-bors of a certain process (adjacency information for a graph topology).

Section 2. Developing multi-threaded applications using OpenMP and Parallel LINQ technologies

Topic 2.1. OpenMP Programming Interface: a general description

OpenMP interface and its versions. Model of the OpenMP program. The compiler mode with the OpenMP support. OpenMP directives and routines (functions). Parallel parts of the OpenMP-program. Shared and local variables in the OpenMP-program. Setting the parameters of the OpenMP-program via environment variables, OpenMP functions, and clauses of the OpenMP directives. Timing functions.

Topic 2.2. The basic means of parallelization in OpenMP

Low-level parallelization via omp_get_thread_num () and omp_get_num_threads (). Par-allel sections, their features. Directives for single-thread execution. Parallelization of loops, its features. The schedule clause and schedule types for load balancing (static block distribution, dy-namic block distribution with fixed-size blocks and guided block distribution with variable-size blocks).

Topic 2.3. The synchronization means in OpenMP

The importance of synchronization for parallel programs with shared memory. Critical sections and their usage (the critical directive). The barrier synchronization (the barrier directive). The assignment synchronization (the atomic directive). An example of ineffective and effective synchronization of the implementation of the parallel algorithm for finding the maximum or min-imum value.

Topic 2.4. The Parallel LINQ technology for .NET framework

LINQ technology for .NET framework: a general description. Interface IEnumerable <T> and its usage for the implementation of an abstract sequence. LINQ queries and their features; types of queries. The Parallel LINQ interface as an implementation of the multi-threaded execu-tion of queries, and its restrictions. Types of the elements' distribution among multiple threads. Parallel execution of the Aggregate query, its features.

Section 3. Parallel algorithms in numerical methods. Development and imple-mentation of parallel programs on computer clusters. Development of educational software for parallel technologies

Topic 3.1. Overview architectures of multiprocessor computer systems

Flynn classification of computer systems. Vector-conveyor supercomputers. Symmetric multi-processor system (SMP). Systems with a massive parallel processing (MPP). Clustered systems. The effectiveness of parallel programs. Amdahl's law. Load balancing. Parallel pro-grams with the shared memory: problems arising in connection with the cache memory. Account-ing the communication environment properties (bandwidth and latency) for parallel programs with the distributed memory.

Topic 3.2. The parallel matrix algorithms

Using the MPI technology to implement a big matrices multiplication. The band matrix multiplication algorithms. The block matrix multiplication algorithms (the Fox algorithm and the Cannon algorithm), features of their implementation (using the derived datatypes, communicators with the Cartesian topology, collective communications).

Topic 3.3. Parallel methods of solving boundary value problems of mathematical physics

The Dirichlet problem for the Poisson equation, non-parallel versions of its numerical so-lution (the Gauss-Seidel method and the Gauss-Jacobi method). Parallel algorithms for solving the problem using OpenMP technology: two variants of the Gauss-Seidel method. Eliminating the mutual influence of the parallel threads: a parallel variant of the Gauss-Jacobi method; Gauss-Seidel method with the alternation of odd and even rows; wave algorithm and problems with its efficiency. Parallel algorithms for solving the problem using MPI technology: the band algorithm of the Gauss-Seidel method. Features of the band algorithm (using communicators with the Car-tesian topology and various types of collective communications).

Topic 3.4. Parallel methods of solving n-body problem

N-body problem, its mathematical model and non-parallel version of its numerical solu-tion. Parallel algorithms for solving the problem that use a shared memory without and with the load balancing, a comparison of their efficiency. Parallel algorithms for solving the problem that use messaging: the “manager-workers” model, the pulse model, the conveyor model, their soft-ware implementation.

Topic 3.5. Running programs on a Unix cluster

The Unix-systems properties. Organization of the file system; files and processes. Com-mand executing on a Unix system. Commands for working with directories, files, and processes. Using the Midnight Commander file manager on Unix-based servers. Remote work with Unix-based servers: programs for Windows-based client (the PuTTY terminal program, the FAR file manager and its using for files transfer); configuring of the PuTTY program; communication with the server and authorization. Compilation of parallel programs on Unix-based servers, in-cluding the compilation of programs using MPI. The parallel program running in several nodes of the cluster using the job scheduling system PBS (Portable Batch System).

Topic 3.6. Developing sets of learning tasks on parallel programming by means the TaskMaker of the Programming Taskbook for MPI

A constructor of learning tasks TaskMaker: overview and basic functions. Design a multi-lingual set of learning tasks. Additional functions of TaskMaker for implementing the tasks on parallel programming. Samples of learning tasks created by the TaskMaker.

Questions for self-study

MPI: persistent communication requests. OpenMP: tasks using and advanced synchronization by means of simple and nestable locks.

"Machine learning: the mathematical basis"

To study this academic discipline, knowledge and skills formed by the disciplines “Selected topics in probability and statistics”, “Computer technologies”. Basic knowledge on probability theory, optimization methods, linear algebra and programming, formed by the undergraduate disciplines, is also required.

The knowledge and skills gained during the study of this discipline are useful for “Econometrics”, “Stochastic modeling and statistical data processing”, Research seminar, as well is for study of the professional problems in research, scientific-production and design activities, in particular, in the preparation for final qualification work.

Discipline goals

To understand the main principles and algorithms of machine learning; to get familiarity with the statistical learning theory and optimization methods; to master the modern software (Python, scikit-learn, tensorflow, keras libraries), used for model construction and data analysis; to gain experience on specific projects.

Requirements for the results of mastering the discipline

Knowledge: principles and algorithms of machine learning, modern software tools (based on Python) used in machine learning, results from the theory of statistical learning and optimization methods.

Skills: building machine learning models and their software implementation in scikit-learn, tensorflow and keras libraries.

Discipline content

1. Machine learning technologies

1.1. Python basics. NumPy, Matplotlib, SciPy, Pandas, scikit-learn libraries.

1.2. Basic principles of machine learning. Supervised learning. Classification and regres-sion tasks. Empirical risk minimization. Bias-complexity tradeoff. Overfitting and underfitting. Train/test splitting. Cross-validation. Validation and Learning curves. Model selection. Regulari-zation.

1.3. Basic machine learning models and techniques. Linear regression, Logistic regres-sion, Support vector machines, Decision trees, Random forest, Gradient boosting, Neural net-works, Kernel methods, Preprocessing and scaling, Dimensionality reduction, Feature extraction.

2. Statistical learning theory

2.1. Conditional expectations

2.2. Rademacher complexity. Generalization bounds for Lipschitz losses.

2.3. Binary classification: Vapnik-Chervonenkis dimension

2.4. Convex functions, subgradients

2.5. Regularized Loss Minimization and stability

3. Optimization for machine learning

3.1. Gradient descent and its variants

3.2. Stochastic gradient descent. Learning with SGD

3.3. Convex duality

3.4. Singular value decomposition. Dimensionality Reduction

4. Deep neural networks

4.1. Generic Representation of Neural Networks. Gradient descent

4.2. Multilayer Perceptron

4.3. Convolutional Neural Networks

4.4. TensorFlow, Keras

Topics for self-learning. Online learning, reinforcement learning.

"Modern Computer Technology"

As basic input knowledge, skills and abilities acquired in the framework of university undergraduate courses in programming, differential equations and numerical methods are required. A preliminary course in Theoretical Mechanics is desirable but not strictly required.

The study of the discipline contributes to the preparation of students for solving modern problems of mathematical modeling, the implementation of modern software tools, including the use of multiprocessor technology. The discipline complements and expands knowledge of programming, numerical methods, mathematical and computer modeling. The knowledge and skills gained in the course of studying this discipline can be used to solve professional problems in research and project activities.

The purpose of studying the discipline

The main goal of studying the discipline is to develop students' competencies related to the possession and use of modern mathematical methods and software for modeling mechanical, physical, chemical and biological processes and phenomena.

Objectives:

- acquaintance with modern technologies of mathematical modeling based on particle dynamics;
- study of modern software tools and libraries of high-performance computing in the field of molecular dynamics;
- possession of the capabilities of the particle system as a graphics engine for modeling and rendering smoke, rain, fire and other phenomena that require a large number of small moving objects.

As a result of the successful mastering of this course, the student will be able:

- To understand areas of application of methods and approaches based on particle dynamics in various fields of modern natural science;
- To apply existing software libraries to simulate the behavior of systems containing a large number of material particles;

- To use a computational experiment for a comparative analysis of the effectiveness of algorithms and their improvement;
- To understand typical algorithms (including parallel ones) for studying large particle dynamics systems;
- To use the capabilities of modern graphic 3D modeling packages to visualize phenomena and processes based on particle dynamics;
- To understand mathematical foundations of molecular dynamics;
- To implement independently algorithms for calculating and visualizing multipoint systems using modern development tools;
- To set up independently the tasks of modeling a physical phenomenon, choose technologies and tools for this modeling and adapt these tools to solve a specific problem

Discipline content

Module 1. Particle systems and their visualization.

Module 2. Molecular dynamics methods

Module 3. Particle systems in computer graphics.

"Stochastic modeling and statistical data processing"

To study this academic discipline, knowledge, skills and skills, formed by previous disciplines: Probability theory and mathematical statistics, Linear and General algebra, Computer tools of intellectual activity, Computer algebra packages

Knowledge. The main theoretical laws of probability distribution, nature of random variables

Skills of using computer tools for data visualization and their elementary analysis, approaches of linearization and methods of linear algebra methods.

Knowledge and skills obtained during the study of this discipline, can be applied to solve professional problems in research, including in the framework of the graduate work of the master.

Goals and objectives of studying the discipline

The goal is to teach students to effectively deal with statistical data, perform primary statistical analysis, determine the nature of the data, motivated to choose the appropriate statistical model and use the functionality of applied packages for the statistical analysis.

Requirements for the results of the discipline

As a result of mastering the discipline, the train must possess the following knowledge, skills and abilities:

Knowledge:

An effective approaches to data preprocessing; basic distribution laws, statistical criteria; basic statistical models.

Skills:

Apply primary statistical analysis schemes to recognize the nature of the data; select the appropriate model; justify the model, research strategy; using the functionality of R-lang, MatLab, Maple packages.

Abilities:

An effective and evidence-based, including means of R packages-lang, MatLab, identification of the nature (characteristics, model kind and properties) of the data.

Content of the discipline

Section 1. Foundation of mathematical apparatus of probability theory and of mathematical statistics. Discrete probabilistic space and continuous probabilistic space. Models of distribution laws widely used in statistical research. Statistical parameter estimation. Statistical tests.

Section 2. Initial statistical treatment of data.

Descriptive statistics and empirical distributions. Regression. Principal components regression Linear classification. Application to chemometrics.

Questions to be taken for independent study.

R functionality

Neural networks application

"Modern Information Technologies and Analytical Computing Packages"

To study this academic discipline, knowledge and skills formed by the following disciplines studied in the undergraduate program are required: "Algebra and Geometry", "Differential equations", "Scientific computing software packages", "Numerical methods".

The knowledge and skills gained during the study of this discipline can be used to study the Master level disciplines "Advanced problems in Mathematical Physics" and "Modern Numerical Methods in Mathematical Modeling", and can be also used to solve professional problems in research, scientific-production and design activities, as well as in preparation for final qualification work.

Discipline goal

Main goals:

Training in the use of modern mathematical packages and applications for the formulation and solution of applied problems of computational algebra, theoretical mechanics and mathematical physics;

Mastering the foundations of algebra methods, differential equations and mathematical-mathematical equations for the analytical and numerical solution of applied problems in mechanics and physics;

Development of the ability to write algorithms and programs for solving the problems posed using the proposed solution methods in the mathematical package Maple, building analysis of the results of numerical and analytical solutions to problems.

Main tasks:

Expansion of students' skills in working with application programs;

formation of skills in the correct formulation and solution of applied problems of theoretical mechanics and mathematical physics;

Training in working with modern mathematical packages for solving applied problems of theoretical mechanics and mathematical physics;

The ability to conduct a comparative and qualitative analysis of the results of solving the tasks.

Requirements for the results of mastering the discipline

As a result of mastering the discipline, the student is expected to have the following knowledge, skills and abilities:

Knowledge:

Fundamentals of procedural programming, the basic methods of using the interface and writing programs for calculating applied problems; obtaining and analyzing the results of solving applied problems; bases of methods of algebra, differential equations and equations of mathematical physics for solving a number of important applied problems in mechanics and physics.

Skills:

Application of known methods of algebra, differential equations and equations of mathematical physics to solve specific applied problems of theoretical mechanics and mathematical physics; the development of application programs for solving the tasks posed; obtaining and analyzing analytical and numerical results.

Abilities:

Correct formulation and solution of the problems of theoretical mechanics and mathematical physics, team work in the implementation of one project, work with modern packages of applied programs; solving applied problems using algebra, differential equations and equations of mathematical physics, creating a short effective code; Optimal choice of the methodology of the solution depending on the task.

Content of the discipline

The concept of mathematical modeling. A brief introduction to Maple and its application to physics.

Demonstration of Maple's capabilities for performing algebraic transformations, plotting charts and performing computational operations. Symbolic and numerical calculations.

Application of the basic principles of theoretical mechanics for the formulation of equations, the use of the Maple system for solving physical problems.

Fundamentals of second-order differential equations with constant coefficients for describing mathematical models of actual physical processes.

Differential equation of harmonic oscillations. Harmonic oscillator, types of oscillations. The oscillation energy of a harmonic oscillator. Phase portrait.

The course is divided into the following sections:

- Basic Algebra and Solving Equations
- Calculus
- Differential Equations
- Vectors and Matrices
- Simple Harmonic Oscillator
- Damped Oscillation
- Sinusoidally Driven Oscillation
- Phase Space

Questions submitted for independent study

1. Solving a block of problems demonstrating the capabilities of the Maple mathematical package.

2. Creating and building programs for solving tasks.

3. Analysis of the results of solving problems.

“Numerical methods of linear algebra”

To study this academic discipline, knowledge and skills formed by the following disciplines studied in the undergraduate program are required: “Algebra and Geometry”, “Differential equations”, “Scientific computing software packages”, “Numerical methods”.

The knowledge and skills gained during the study of this discipline can be used to study the Master level disciplines “Advanced problems in Mathematical Physics” and “Modern Numerical Methods in Mathematical Modeling”, and can be also used to solve professional problems in re-search, scientific-production and design activities, as well as in preparation for final qualification work.

Discipline goal

Study of modern numerical methods for solving the problems of computational algebra for the large and extra-large linear systems of equations with sparse matrices and the use of mod-ern mathematical software for programming numerical solution methods for sparse linear systems.

Main goals:

- Study of structures and sparse storage formats for the matrices that arise from the discretization of partial differential equations,
- Review of direct and iterative solution methods for sparse linear systems of large size,
- Study of projection methods for sparse linear systems,
- Detailed study of the Krylov subspace methods based on the Arnoldi orthogonalization and Lanczos biorthogonalization,
- Study of preconditioning techniques and different types of preconditioners,
- use modern mathematical software (Maple, Matlab) for programming numerical solution methods for sparse linear systems.

Requirements for the results of mastering the discipline

As a result of mastering the discipline, the student is expected to have the following knowledge, skills and abilities:

Knowledge:

- main sparse storage formats for large sparse matrices;
- direct solution methods for large sparse linear systems;
- main iterative solution methods for linear systems;
- projection solution methods for linear systems;
- Krylov subspace solution methods for large sparse linear systems;
- Preconditioning methods.

Skills:

- apply sparse matrix technology to investigate modern numerical problems of large size;
- use of numerical algorithms to solve large sparse linear systems;
- writing programs in modern mathematical software packages to work with sparse matrices.

Abilities:

- demonstrate ability to use technology of sparse matrices for solving discretized problems of mathematical physics;
- apply a suitable numerical solution method for a given sparse linear system and justify its suitability both theoretically and practically;
- employ preconditioning techniques to precondition a given sparse linear system by select-ing appropriate type of preconditioner;
- implement direct and iterative algorithms for solving sparse linear systems in the form of program code;
- use modern mathematical software (Maple, Matlab) for programming numerical solution methods for sparse linear systems.

Content of the discipline

Module 1. Background in sparse linear systems.

Topic 1.1. Main concepts of linear algebra and matrix theory

Background in Linear Algebra. Basic definitions. Types and structures of square matrices.

Topic 1.2. Matrix norms. Basic matrix factorizations

Vector and matrix norms. Range and kernel. Orthonormal vectors. Gram-Schmidt process. Eigenvalues and their multiplicities. Basic matrix factorizations and canonical forms: QR, di-agonal form, Jordan form, Schur form. Basic matrix factorizations: SVD, LU, Cholessky. Properties of normal, Hermittian matrices and positive definite matrices.

Topic 1.3. Perturbation analysis and condition number.

Existence of Solution. Perturbation analysis and condition number. Errors and costs.

Topic 1.4. Discretization of partial differential equations.

Discretization of partial differential equations (PDEs). Finite differences. 1D Poisson's equation. 2D Poisson's equation. Overview of Finite element method. Assembly process in FEM.

Module 2. Direct, iterative and projection methods for sparse linear systems.

Topic 2.1. Structures and graph representations of sparse matrices

Graph representation of a matrix, matrix and its adjacency graph. Permutations and Reordering. Basic concepts and some algorithms. Storage schemes for sparse matrices (Coordinate format, Compressed Sparse Row format CSR, Compressed Sparse Column format CSC, Modified Com-pressed Sparse Row format with separate storage of diagonal elements MSR, Modified Com-pressed Sparse Column format with separate storage of diagonal elements MSC, Diagonal format for matrices with diagonal structure DIAG, Ellpack-Itpack format). Algorithms for basic matrix operations for sparse formats: matrix-by-vector product, solving triangular systems for matrices in sparse formats.

Topic 2.2. Review of direct solution methods for sparse linear systems.

Comparison of direct and iterative methods. Overview of direct solution methods. Direct sparse methods (Gaussian elimination with partial pivoting).

Topic 2.3. Basic iterative methods for linear systems with sparse matrices.

Iterative methods: general idea and convergence criterion. Basic iterative methods: Jacobi, Gauss-Seidel, Successive Over Relaxation (SOR), Symmetric Successive Over Relaxation (SSOR). Properties of diagonally dominant matrices, location of matrix eigenvalues. Convergence criteria for iterative methods.

Topic 2.4. Projection methods.

Projection methods: general formulation of a projection method. One-dimensional projection methods: Steepest Descent method (SDM), Minimal Residual Iteration method (MRIM), Residu-al Norm Steepest Descent method (RNSD).

Module 3. Krylov subspace methods for sparse linear systems and preconditioning tech-niques.

Topic 3.1. Krylov subspace methods based on Arnoldi orthogonalization

Krylov subspace methods. Definition of Krylov suspace. General formulation of a Krylov sub-space method. The process of Arnoldi orthogonalization to form a basis for Krylov subspace. Ar-noldi relation and its properties. Methods based on Arnoldi process: Full Orthogonalization method (FOM). Methods based on Arnoldi process: Generalized Minimal Residual method (GMRES). Givens rotations in GRMRES. Calculation of residual in FOM and GMRES. Residu-al polynomials.

Topic 3.2. Krylov subspace methods based on Lanczos orthogonalization

Lanczos orthogonalization for symmetric systems. Lanczos methods for symmetric systems: classic and direct. Derivation of Direct Lanczos method. Derivation of Conjugate Gradient meth-od (CG) for systems with symmetric positive definite matrices. Generalization of CG for systems with Hermitian and nonsymmetric matrices: Conjugate Residual (CR), Generalized Conjugate Residual (GCR).

Topic 3.3. Krylov subspace methods based on Lanczos biorthogonalization

Lanczos biorthogonalization for nonsymmetric systems. Classic Lanczos method for nonsymmet-ric systems. Derivation of Biconjugate Gradient method (BiCG). Overview: Efficient and optimal methods.

Topic 3.4. Preconditioning techniques. Preconditioned iterative algorithms

Basic ideas of preconditioning technique. Examples of preconditioners: Jacobi, Gauss-Seidel, SOR and SSOR preconditioners, incomplete LU-factorization preconditioners.

Self-study questions.

Preconditioned Krylov Subspace methods. Preconditioned Conjugate Gradient method (PCG), Split Preconditioned Conjugate Gradient method (Split PCG). Preconditioned Generalized Minimal Residual method, algorithms of GRMES with left and right preconditioning.

"Modern problems of applied mathematics and informatics"

To study this academic discipline (module), the following knowledge, skills and abilities are required, formed by the previous disciplines: linear and general algebra, mathematical analysis, courses of differential equations, equations of mathematical physics, numerical methods, functional analysis and continuum mechanics (preferably, but not necessarily)

Knowledge: statement of problems of mathematical physics, types of equations of mathematical physics, methods of discretization of equations of mathematical physics, methods for solving systems of linear and nonlinear problems and dynamic problems, fundamentals of the finite element method (desirable but not required);

Skills: formulate continual and computational problems of mathematical physics, apply methods of applied mathematics for the numerical solution of basic problems of mathematical physics; setting and solving basic problems of mathematical physics, using basic numerical methods, skills in working in computer algebra packages.

The list of subsequent academic disciplines for which the knowledge, skills and abilities formed by this academic discipline are required: all subsequent disciplines in mathematical modeling, in which it is assumed the numerical solution of problems in mathematical physics, continuous models, research work, research seminar, practice and work on a master's thesis.

Goals and objectives of studying the discipline

The objectives of mastering the discipline "Modern problems of applied mathematics and informatics" are to get acquainted with the methods of constructing and analyzing complex mathematical models for related physical and mechanical problems; in the presentation of the basic numerical methods together with algorithms and software implementations for the study of these types of mathematical models; in acquaintance with the modern science-intensive software complex ANSYS, and in the use of modern software tools and computer technology for solving various physical and mechanical problems.

The objectives of the course are reduced to acquaintance of undergraduates with a unified end-to-end methodology for setting mathematical problems, their research, transformations based on generalized formulations, the use of discretization procedures, numerical methods and software solutions for the analysis of relevant related physical and mechanical problems focused on real practical applications.

Requirements for the results of mastering the discipline

As a result of mastering the discipline, the student must have the following knowledge, skills and abilities:

Knowledge: methods of analysis and numerical solution of modern problems of mathematical physics, features of finite element modeling and implementation of modern computing systems.

Skills and abilities: to implement the main types of finite element analysis for complex static and dynamic problems of mathematical physics; plan computational experiments, implement methods and algorithms for finite element modeling in ANSYS; to program and carry out computational experiments for solving various problems of mathematical physics.

Content of the discipline

Module 1. Modeling related physical and mechanical problems

1. General Provisions. Vector and tensor fields. Basic operations with tensors. Related physical and mechanical problems. Statement of heat conduction problems. Homogeneous, heterogeneous, isotropic and anisotropic media. Linear and nonlinear formulations of heat conduction problems. Questions of the uniqueness of the statement of the heat conduction problem. Generalized formulation of heat conduction problems. Physical meaning of generalized statements, variational statements.

2. General scheme of the finite element method. FEM as the Bubnov-Galerkin method. Features of finite element bases. Computational advantages of FEM. Ensemble FE objects. Consideration of the main boundary conditions. One-dimensional FE: linear, quadratic FE. Isoparametric FE. h- and p-convergence. Quadrangular FE for 2D problems. Bilinear FE. Sirendip type quadrangular FEs. Triangular linear and quadratic Lagrangian FE for two-dimensional problems. Consistency problems of FE partitions. FE for 3D problems. FE with node

reduction and vertex collapse. Technique for calculating FE matrices for isoparametric elements. Numerical integration in FEM.

3. Static and dynamic formulations of problems in the theory of elasticity. Three types of dynamic tasks. The main features of non-stationary problems of hyperbolic type on the example of the problem for a rod. Generalized formulations of problems in the theory of elasticity. Modeling problems of electroelasticity. Classical and generalized formulations of electroelasticity problems. Semi-discrete approximations in problems of electroelasticity. Review of the main features of the problems of electroelasticity. Static and dynamic problems using finite-dimensional approximations. Three types of dynamic problems with semi-discrete approximations. FE approximation of dynamic problems. The main types of matrix problems of the FEM. Features of FE analysis of various types of dynamic problems. Modal analysis. Properties of natural frequencies and modes in FEM discretized problems of elasticity theory. Methods for solving generalized eigenvalue problems of FE modal analysis. Direct time-step integration schemes for the equations of motion of the FEM. Central difference methods, Newmark, Crank-Nicholson and their computational characteristics.

4. Features of modeling finite element problems. Issues of accuracy of finite element calculations, selection of mesh and elements. The principles of organizing modern computing systems on the example of ANSYS, ABAQUS, MSC, COMSOL packages. Problems of generating mesh partitions. Algorithmic features of FEM. Data storage in solid and finite element models. Review of methods for solving problems with large sparse matrices. Superelements. Features of postprocessor processing of results.

5. Features of the equations of heat and mass transfer and propagation of impurities. Peclet number. Problems with large Peclet numbers. Features of boundary value problems in hydroaerodynamics. Navier-Stokes equation, Rayleigh number and turbulence modeling problems. Fundamental solutions to problems of steady-state oscillations. Reciprocity theorem. Boundary integral equations (BIE) for problems of steady-state oscillations. The boundary element method (BEM) in problems of steady-state vibrations. Comparison of BEM with FEM. Fundamental solutions to non-stationary problems. BEM and FEM for non-stationary tasks. Schemes using the Laplace transform. Conjugation of finite element and boundary element methods.

Module 2. Workshop on solving physical and mechanical problems in modern finite element complexes

1. Lab. slave. 1 "Calculation of the plane stress state of an isotropic plate". Solution in interactive and command mode ANSYS and in FlexPDE.

2. Lab. slave. 2 "Solution of the problem of heat conduction in a flat region". Solution in command mode ANSYS and in FlexPDE.

3. Lab. slave. 3 "Calculation of natural and steady vibrations". Solution in command mode ANSYS and in FlexPDE.

4. Lab. slave. 4 "The solution of the static problem of the theory of elasticity and the stationary problem of heat conduction in the three-dimensional region." Solution in command mode ANSYS and in FlexPDE.

5. Lab. slave. 5 "Static deformation of a piezoelectric transducer".

The research seminar and Master's thesis

The Research seminar teaches students to work with contemporary scientific literature, to adapt general methods to a concrete problem, and to present the results of the study in the style adopted in the academic environment.

Module gives an introduction to autonomous research work by carrying out workshops from different fields of applied mathematics focused in mechanics and biomechanics

On successful completion of the module, the students are expected to be able to:

- prepare and give scientific presentations by using modern computer tools;
- present the report on the obtained results orally in an understandable manner, observing general rules for scientific reporting;
- demonstrate knowledge regarding the area of study and related professions from the discussion in the research seminars on mathematical modelling and IT;
- demonstrate capacity for research and abilities to search, process and analyze the information from a variety of sources;
- show within class attending activity and through extended project work preparation and presentation the ability to work autonomously

The module works in a seminar form. Each student receives a project work topic and presents the problem as well as the work plan at the beginning. Typically, the topics cover modeling problems from different scientific and engineering fields are focusing on mechanics and biomechanics, along with numerical solutions. Solution methods for the project work problems are discussed during the module. As a conclusion, the participants present their project works. The project work typically is an introduction to the diploma work topic of the student. Suitable also for postgraduate studies as joint seminar sessions or conference presentations.

Featured partners

Southern Scientific Center of Russian Academy of Science

Group of companies "Softline"

Regional Scientific and Educational Mathematical Center

Academics

Konstantin Nadolin

Course Director