HERALD OF THE RUSSIAN ACADEMY OF SCIENCES, 2025, Vol. 95, No. 3, pp. 14–21

POINT OF VIEW

CHALLENGES OF MACHINE LEARNING AND MATHEMATICAL MODELING

© 2025
V. P. Ilyin[a,b,*]

[a] Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia

[b] Novosibirsk State Technological University, Novosibirsk, Russia

*e-mail: ilin@sscc.ru

Received October 16, 2024
Revised January 21, 2025
Accepted February 14, 2025

Abstract. The article considers the challenges and problems of machine learning that arise in supercomputer mathematical modeling of real-world processes and phenomena. Currently, such modeling has become the main tool for obtaining fundamental and applied knowledge, as well as a condition for a significant increase in labor productivity and gross domestic product. The principles of modern predictive modeling based on high-performance computing, artificial intelligence and big data processing are described. The trends in the development of high-tech mathematical and software within the framework of integrated computing environments are analyzed; the latter imply a flexible expansion of the composition of the studied models and applied algorithms, the effective use of external products, adaptation to the evolution of computer platforms focused on a long-life cycle. The methodology of machine learning based on the technological cycle is presented, which includes the formation and modification of models, the implementation of a computational experiment with the solution of direct and inverse problems, analysis of the results and decision-making on optimizing activities to achieve the goals.

Keywords: machine learning, mathematical modeling, high-performance computing, artificial intelligence, big data, science-intensive software, integrated computing environments

DOI: 10.31857/S08695873250302e6

The tasks of advanced scientific and technological development in priority areas in the era of exaflops supercomputers must inevitably rely on the mass digitalization of all spheres of human society, provided that artificial intelligence, huge volumes of data, neural networks and machine learning methodologies are actively used. A special role belongs to the means of obtaining new fundamental and applied knowledge based on science-intensive mathematical modeling of complex processes and phenomena. These studies typically include solving interdisciplinary direct and inverse problems and involve extreme volumes of computational resources with scalable parallelization of algorithms on multiprocessor computing systems (MCS). It is fundamental that in the conditions of continuously developing sciences, modeling problems are inevitably associated with dynamic development, maintenance and implementation of mathematical support and software (MSW), the high cost of which requires an economical approach to its professional operation. On the other hand, MSW users must master supercomputer technologies for obtaining knowledge and make decisions based on their analysis.

An important element of such an approach — building digital twins, or virtual realities, the study and optimization of which allow increasing labor efficiency; in fact, it is about a new productive force that ensures the growth of gross domestic product. Equally significant is the penetration of intelligent computational and

ILYIN Valery Pavlovich — Doctor of Physical and Mathematical Sciences, Chief Researcher at the Laboratory of Computational Physics, Institute of Computational Mathematics and Mathematical Geophysics SB RAS.

informational innovations into social and humanitarian spheres, not to mention national security. Figuratively speaking, computer sciences and tools play the role of a circulatory system that ensures the vital activity of human society as a single organism. This situation requires a new look at the interrelation of high-performance computing with artificial intelligence and transformations of big data — the interrelation of the three pillars of the 4th industrial revolution.

Over the past decades, outstanding achievements have been made in all these areas, requiring philosophical and methodological comprehension, bearing in mind the harmonious development of the complex of sciences about knowledge acquisition [1–7]. It should be kept in mind that, for example, emerging possibilities of achieving positive practical results based on cognitive analysis of big data through their simple statistical processing sometimes result in the conclusion about the emergence of some “post-science”, like Data Science, which supposedly does not require traditional in-depth research. To not mislead with such dubious claims, it is necessary to show that for creating artificial intelligence and neural networks, it is necessary to invest enormous effort of professionals with a high level of natural intelligence. Over the past decades, knowledge-intensive mathematical modeling has formed into a multifaceted creative process, and rapid advances in the field of artificial intelligence actively influence the technological stages of machine experimentation, which promises significant synergistic effects.

This paper is devoted to a systematic analysis of the indicated problems from the perspective of a mathematician, programmer, and specialist user in a specific applied field. The following will be considered: basic characteristics and trends of mathematical modeling as a scientific discipline and as a technology for solving practical problems; functional and systemic content of software; methodological principles of computational experiment-based machine learning.

MATHEMATICAL MODELING: SUPERCOMPUTING, INTELLECTUALIZATION, BIG DATA

The philosophical saying “being determines consciousness” directly relates to our topic in the sense that the achievable level of artificial intelligence is determined by the power of computer resources, including processing speed and memory capacity. We will discuss only traditional cluster-type architectures, including computing nodes with distributed memory and multi-core processors (CPUs with several dozen cores over shared memory and, possibly, GPGPU graphics accelerators).

Although both in our country and abroad, active development of new generations of computers is underway, including quantum and reconfigurable ones, in the next 5–10 years they will apparently not yet become mass-produced and competitive. This means that during this period, we can consider configurations with a total performance of about a petaflop and RAM of several dozen or hundreds of terabytes. The history of computer evolution shows that their performance parameters and memory volumes grow approximately proportionally. At the same time, “large tasks” (in N.N. Yanenko’s terminology [8, 9], meaning those that take quite a long time to solve — hours or tens of hours, days or several days) can now be considered as those whose formulation involves solving interdisciplinary multidimensional direct and inverse problems with a number of unknown functions around ten and which require the use of unstructured adaptive grids with the number of spatial nodes of the order of 1000^3, as well as time steps up to 10^3–10^5 or more. For such computational parameters, the computational process even on a supercomputer is a large-scale experiment with extreme volumes of data and arithmetic operations.

According to the established methodology of mathematical modeling [8–13] its technological chain, with all its diversity, includes relatively small number of stages, qualitative content of which, however, significantly changes with the development of supercomputer generations, computational algorithms and tools. We will consider a universal computer, or network of machines, with software that is integrated in the sense that it allows solving the widest possible range of problems.

The study of any object begins with the formation of its model, which can be represented by a set of differential and/or integral, as well as discrete equations and relationships, with additional constraints and optimization conditions, with large volumes of actually measured data. The latter can be approximate and even contradictory, not falling under strict mathematical concepts of existence, uniqueness, and correctness of the solution. Such complex systems have to be analyzed by hydrometeorological services when forecasting weather, climate changes, or warning of natural disasters, using data from the worldwide network of ground and space observations. Another illustration is multiphase processes in the oil and gas industry using chemical, electrophysical and other modern extraction technologies. One of the extreme problems is the consequences of a thermonuclear explosion, where there is an overlay of processes of fluid dynamics, physics of solid matter, plasma, etc.

Naturally, that strategy and tactics of modeling must provide maximum effect with minimization of computational resources, which are by no means cheap. All sciences are developing rapidly, and with them, various models grow quantitatively and qualitatively, all kinds of models, their hierarchies emerge — from simple and economical to more complex and accurate ones. Human interaction with computers requires comfortable interfaces with input languages, determining the level of communication with users of various different specialties. This stage is intended for formulating the task for the computer of that model (or their sequences from a possible set), which is necessary in accordance with the given task. Obviously, that such a level of functionality of the software subsystem and its user can be achieved only as a result of machine learning through accumulating operational experience and analyzing the results obtained. It should be noted that "smart" implementations of this stage may include qualitative analysis of the mathematical properties of models, which undoubtedly increases the value of such a product.

After specification of the mathematical problem at the continuous level, its discretization is required, that is, construction of a grid. This stage is very important from the perspective of modeling efficiency and represents a labor-intensive algorithmic problem in multidimensional tasks with real data, including complex geometric configurations of computational domains with piecewise-smooth non-simply-connected multi-scale boundaries (including moving ones) and contrasting material properties of media. In such cases, often it is necessary to consider singularities of solutions and construct adaptive unstructured grids, the question of optimization of which still remains open.

It should be noted that the most effective numerical methods for solving large problems are associated with domain decomposition, which is the main tool for algorithm parallelization, as well as with the use of multigrid approaches that provide asymptotic optimality in order of solutions (for characteristic grid steps $h \to 0$ the total number of arithmetic operations is proportional to the number of unknowns [14]). This presents to "grid generators" new interesting problems related to constructing complex data structures and intelligent operations on graphs [15].

There are many approaches to constructing grid approximations of various orders of accuracy: methods of finite differences, finite volumes, finite elements, discontinuous Galerkin algorithms and so on. Creating multifunctional software for this purpose with this goal for various types of grids and types of operators is an urgent and in-demand task (the project of the corresponding CHEBYSHEV subsystem is described in [16]). It should be noted that, although the problem of automating the construction of algorithms for grid approximations is quite old [17], widespread adoption of this approach has not yet been achieved.

Most practical problems are nonlinear and non-stationary, but after applying quasilinearization and implicit time approximations to them, it is inevitable to solve systems of linear algebraic equations (SLAEs), typically with large sparse matrices, both symmetric and non-symmetric. The main approach here is iterative preconditioned methods in Krylov spaces [18] (the corresponding subsystem is called the KRYLOV library).

Optimization methods for solving inverse problems play a key role in machine learning, as they allow finding and investigating the best scenarios of processes and phenomena [19]. The typical methodology consists of planning a series of computational experiments in which a minimized target functional is described based on previous experience, and after analyzing the obtained results, the next machine learning session is formed. Such interaction between humans and computers seems to be without alternative in many multi-criteria practical search problems, when a deterministic algorithm for their solution is fundamentally impossible to formalize. Let us note an important point: the parameters varied in this process may relate not only to the model being studied but also to the computational process itself, since in stalemate situations it may either not converge at all, or take unacceptably long.

Thus, the entire iterative cycle involves all the modeling stages considered. Methods and technologies for post-processing, visualization, and analysis of calculation results should be added to them, on the basis of which decision-making tools function [20].

MATHEMATICAL AND SOFTWARE SUPPORT AS AN ECOSYSTEM

Application software, like system software, has been rapidly developing simultaneously with computing technology for which, to everyone's surprise, Moore's law still continues to apply with certain reservations (increase in performance by 1000 times in 11 years). However, it must be acknowledged that in recent decades, the growth rate of programmer productivity has begun to significantly lag behind the pace of computer performance, that is, in a certain sense one can speak about a crisis in programming. To correct this imbalance, artificial intelligence is beginning to be actively used.

By now, an enormous amount of publicly available (Open Source) and commercial software has been accumulated worldwide in the form of libraries, special tools and problem-oriented application software packages (ASP) [10], which represent high intellectual value. Here we can mention such highly professional developments as PETSc, HYPRE, PARDISO libraries, FENIX, DEAL II software packages, specialized systems PARVIEW, MAPLE and many others. A separate global market consists of computer-aided design (CAD) systems [21], and in recent years there has been a convergence of these systems with classical ASPs. At the same time, there has been a shift from specialized to integrated software environments. Examples of major projects in this area are DUNE, OPEN FOAM, INMOST, as well as the Basic Modeling System (BMS) [12, 22–24]; all of them are primarily method oriented.

Functional content of BSM is a set of autonomous subsystems, each of which is responsible for the corresponding technological stage of modeling and is connected with others through coordinated data structures. Thanks to the model formation subsystem, functional and geometric data structures (FDS and

GDS) are created which serve as the initial ones for the stage of grid generation. Based on the resulting grid data structure (GDS) as a result of executing the stage of approximation, algebraic information arrays are built in globally accepted formats (ADS), which provide high-performance solutions for a wide class of systems of linear algebraic equations.

This architecture allows independent groups of professional developers to implement and develop various subsystems. This approach easily achieves flexible expandability of the module and algorithm composition for each computational stage, including those with the efficient reuse of external software products and adaptation to new computer platforms. The resulting integrated computational environment (ICE) represents a self-sustaining ecosystem with a long-life cycle, oriented toward successful use by a wide range of users. The provided redundant set of models and algorithms is designed to support methodologies of machine learning for both the ecosystem itself and its users.

Ensuring rich functionality and efficient use of the integrated computational environment requires the creation of diverse system content. From a mathematical point of view, this primarily means developing tools for automating the construction of algorithms, including tools for multi-version configurations of computational modules. Regarding programming languages, it is worth noting the popularity of combining styles of object-oriented approaches of C++ and rich expressive capabilities of the interpreted language Python, which includes such important intellectual components as computerization of complex analytical calculations. This opens the way to the active use of high-precision approximation methods, which are promising from a theoretical perspective due to significant reduction in required memory and energy-intensive communication costs, but are still not widely used due to labor-intensive programming.

The world is also developing specialized natural languages for computational mathematicians (there is even a slogan “programming without programming”). Although such enticing projects on “language factories” as SIDL (Scientific Interface Definition Language) and DSL (Domain Specific Languages) [25] are mentioned in Internet publications, where the urgent problem is interpreted as a transition from “paleoinformatics to neoinformatics”, no breakthrough prospects are visible in the near future, and the technologies for forming algorithmic libraries have remained unchanged for several decades. In general, language content is one of the key aspects for the level of machine learning, and the current situation can be assessed as a soft exit from the programming crisis.

Undoubtedly, one of the main qualities of an application is its performance, primarily determined by the quality of algorithm parallelization. Here, FPGAs (Field-Programmable Gate Arrays) offer enormous possibilities, which allow to design and build specialized computers with maximum performance for a given algorithm. However, such computers have one major disadvantage — commercial non-competitiveness in comparison with standard super computers of cluster type, whose computational nodes are connected by buses and exchange data using a very simple MPI library, while calculations on each of them are performed by multi-core processors with hierarchical shared memory (its different levels have varying volumes and exchange speeds) managed by a software system like OpenMP. There are also fast graphics accelerators (GPGPU), but their connection with shared memory is slow, which significantly reduces their efficiency.

The situation with computation parallelization can be considered paradoxical due to the absence of appropriate mass, or standard, systems for programming automation, with the exception of several uncommon languages or subsystems (for example, SHAPEL and DVM — Distributed Virtual Machine [26]). In fact, parallelization of algorithms on multiprocessor computing systems is a purely manual work with experimentally selected methods of acceleration of calculations, which are measured by two simple parameters:

$$S_p = T_1 / T_p, E_p = S_p / p$$

— coefficients of speedup and efficiency, where T_p is the time to solve the problem on p processors. Research on optimization of parallelization based on the concept of D-determinant can be considered promising [27]. From the perspective of machine learning, we can formulate the problem of searching for the best parallelization method by the supercomputer itself based on a series of calculations for a specific class of problems.

The speed of such inevitable routine procedures as debugging, testing, verification, and validation of the implemented code is of great importance for improving programmer productivity. The problems arising here are inevitably exacerbated when creating large software systems, which are precisely what are implied in the concept of integrated computational environment; this is primarily due to the dramatic complication of information connections, as well as internal intermodular and user interfaces. In large professional teams of developers of operating systems and compilers, these problems have long been solved using well-known component technologies COM/DCOM and CCA (Common Component Architecture) [28], but in applied programming they are still waiting for the transition to industrial thinking.

One of the main ICE components is a repository (storage), which ensures the integrity of the development and its connections with the outside world, supports the properties of multilingualism and cross-platform, as well as interaction with developers and users. GITHUB, a system of this type with a wide variety of services, has become widely used in the computer community.

ABOUT SOME PRINCIPLES OF MACHINE LEARNING AND DECISION-MAKING

All learning consists in acquiring knowledge and skills for some sphere of activity, including for making decisions based on the analysis of meaningful information. Regarding mathematical modeling, we can identify three categories of participants: inanimate object —computer, consisting of hardware and software, and two types of subjects — developers of modeling tools and end users–specialists in specific subject areas who implement supercomputer innovations. It is clear that personnel can be represented by people of different qualifications, between whom teacher–student relationships emerge.

Knowledge is active if it leads to some actions and results [29]. Its accumulation and systematization for a particular subject area involves the development of an appropriate base of active knowledge (AKB) containing all kinds of information about objects, their specifications, interrelationships and about possible actions on them. With the expansion and deepening of acquired information, the emergence of AKB is inevitable, as the volume of information becomes unmanageable for human assimilation. The structure of AKBs has not yet been finalized, an example of a project for such development for computational algebra tasks is given in [30]. A prototype of such development can be considered the ALGOWIKI system [31], created under the guidance of J. Dongarra and V.V. Voevodin.

Knowledge bases should contain all necessary information on the relevant topic. If we are talking about mathematical modeling for a specific class of problems, this includes descriptions of models, computational methods and technologies, examples of problems and their solutions (for which special archives should be created), recommendations for various applications, as well as literary sources and available software materials with documentation. In other words, an ontology should be developed that allows for text parsing, statistical data analysis and other intellectual activities. The knowledge base implies a system for collecting and assimilating huge volumes of information, for example, space, meteorological, etc., which must be integrated with operational calculations.

The most meaningful problems here are those related to optimization methods for solving inverse problems, allowing to achieve the greatest practical effect, for example, when identifying parameters of the model based on results of field measurements (oil and gas field, for instance) or optimization of operational modes of technical equipment (aircraft, ship, factory shop and so on) [19]. At the same time the problem formulation is defined as finding the minimum of some objective functional

$$\Phi_0\left(\vec{u}\left(\vec{x},t,\vec{p}_{opt}\right)\right) = \min_{\vec{p}} \Phi_0\left(\vec{u}\left(\vec{x},t,\vec{p}\right)\right)$$

$$\vec{p}$$

for solving a certain direct initial-boundary value problem, which is subject to additional linear and/or nonlinear constraints:

$$p_k^{min} \le p_k \le p_k^{max}, k = 1,\ldots,m_1,$$

$$\Phi_l\left(\vec{u}\left(\vec{x},t,\vec{p}\right)\right)q \le \delta_l, l = 1,\ldots,m_2,$$

$$\vec{p} = \{p_k\} \in \mathcal{R}^m, m = m_1 + m_2.$$

Here $\bar{p}$ is an m-dimensional vector of optimized parameters, and t, $\bar{x}$ are time and spatial coordinates. The original direct problem, or the state equation of the optimized complex system, can be formally represented in the following form:

$$L\vec{u} = \vec{f}\left(\vec{x},t\right), \vec{x} \in \bar{\Omega} = \Omega \cup \Gamma, 0 < \mathrm{t} \le \mathrm{T} < \infty$$

$$l\vec{u} = \vec{g}\left(\vec{x},t\right), \vec{x} \in \Gamma, \vec{u}\left(\vec{x},0\right) = \vec{u}^0\left(\vec{x}\right),$$

$$\bar{\Omega} = \cup\bar{\Omega}_j, \Gamma = \Gamma^e \cup \Gamma^i,$$

$$\Gamma^i = \cup\Gamma^i_{j,k} = \cup\left(\bar{\Omega}_j \cup \bar{\Omega}_k\right),$$

where L is in the general case an operator of matrix type (in differential and/or integral form), l is an operator of boundary conditions, $\bar{\Omega} \times [0, T]$ is the computational domain, which often consists of $\bar{\Omega}_k$ subdomains with different contrasting material properties, as well as with internal and external boundaries Γ^e, Γ^i , including those with multi-scale details and piecewise-smooth multiply connected surface segments. In the general case, we are dealing with interdisciplinary non-classical formulations, where even questions of existence, uniqueness and well-posedness remain open. As for the problem of constrained minimization of the functional, it can be of local or global type. In the latter case, it is required to find all existing minima.

Here is a typical example of machine learning. Let us assume that during an extended period (a year or more) we need to conduct operational optimal control of a complex system dependent on 10 parameters by solving inverse and forward problems, where each of them requires lengthy calculations (hours or tens of hours). Note that if for each parameter we consider 10 possible values, the total number of variants will be 10^{10} (the curse of dimensionality)! In this case, machine learning can be implemented as follows. First, for several weeks, using classical optimization methods, hundreds of problems are solved, the results of which are stored and statistically processed (for example, using a popular type of generalized linear regression — kriging), forming appropriate approximations in the grid parameter space. Then, based on the accumulated data, operational work of the machine-software complex begins, which quickly finds an approximation to the desired state of the system, and then with human participation, the necessary refinement of parameters occurs sequentially.

In artificial intelligence research, the concept of "Foundation Models" (or LxM — Large x Model) has emerged, defined as deep learning tools trained on a huge number of test examples and problems [32] and extensive literature, mainly unpublished, placed on publicly available internet resources like arXiv. Additionally, the term "surrogate optimization" has come into use, which implies that the search for the best solution is performed not for the real object or process, but for its model, possibly crude. Overall, the solution of complex inverse problems with multi-ravine behavior of minimized functionals requires the application of a hierarchy of models, the selection of which requires high artistry. In fact, in such cases, meta-algorithms operate for computer interaction with users who have extensive experience in solving specific classes of problems.

The development of machine learning in computational mathematics is primarily associated with the choice of the optimal or "good" algorithm for each stage of the technological chain of modeling: grid generation, approximation and discretization of the original problem, solving the resulting systems of linear and nonlinear equations and so on. As a result, the search for the best general computational process reduces to multi-level repetitive cycles with multiple numerical experiments, analysis of intermediate results and development of the final strategy for practical calculations. Often, it is necessary to make a compromise of the "perfect is the enemy of good" type, since optimization itself may be more expensive than an already known sufficiently effective approach. The penetration of machine learning into computational mathematics is quite active, and here we can note works on finite element methods and on iterative algorithms [33, 34] and literature cited therein. Regarding the solution of various applied problems, a special neuromethodology called PINN (Physics Informed Neural Networks) has emerged, focused on solving differential equations describing conservation laws, including at the continuous level, without transitioning to grid discretizations [35].

Neural network approaches evolve very rapidly. In 2017, in the paper [36] a neural network architecture called Transformer was proposed, based on the attention mechanism. Its essence resides in determining global dependencies between input and output data through pre-training by examples. Software implementations built on these principles have already significantly enhanced the toolkit for image processing, text analysis, machine translation, and more. As a convincing example of the effective use of machine learning to solve truly very complex physical-mathematical problems, let's consider the task of determining interatomic potentials that approximate models of quantum-mechanical interactions, originally described by the extremely resource-intensive Kohn–Sham theory. A sufficiently general approach to solving this problem is based on neural network potentials, the modern version of which is presented using eMTP — electron moment tensor potential, which underlies the foundation of a class of machine-learned models of interatomic interactions with the required high accuracy [37].

Another research direction is neural operators (NO) [38], which are an evolution of "physics-informed" neural networks like PINN. Unlike the latter, NO are focused on approximating inverse operators that characterize connections between functional spaces, which allows transitioning to solving entire classes of problems. It should be noted that new neural network and neural operator technologies are, in a certain sense, well-forgotten old approaches of classical methods of mathematical physics from the middle of the last century, which are based on continuous basis functions, effectively applied for small orders, but becoming dramatically more complex when increasing calculation accuracy.

Similar methodologies are applicable in other sciences (chemistry, biology, etc.) or in industries: model, computer experiment, results analysis, new knowledge, decision-making. However, this is not an end in itself; it should be followed by decisions on optimizing human activities and innovations — increasing labor productivity, quantity and quality of products, achieving social and/or humanitarian effects, etc. In fact, we are talking about a fundamental change in the ways of activity and the emergence of new mass professions. Undoubtedly, machine learning should significantly change pedagogical approaches and the education system itself — from preschool to higher education, and these relevant issues require their own research.

* * *

In modern conditions, machine learning has become an integral attribute of obtaining new knowledge. One of the problems in the field of artificial intelligence use is decision-making by humans based on the analysis of received data. Optimization of this type of activity is an urgent, but by no means new problem. As an illustration, one can cite the many years of work by G.S. Altshuller and his followers [39] on creating TRIZ — Theory of Inventive Problem Solving. Modern approaches within this direction are based on building ontologies of various subject areas, which allow creating cognitive tools for decision-making [40].

A vivid example of such a modern project is contained in the report [41] of the Center for Research on Foundation Models (CRFM, Stanford University), which describes the concept of an ecosystem focused on effective intellectual innovations in the broadest applied spheres: healthcare and biomedicine, jurisprudence and education, economics and environment, etc. In a certain sense, this approach correlates with the methodology of an integrated computational environment with a basic modeling system, which represents a knowledge-intensive functional content for high-performance solving of interdisciplinary direct and inverse problems of mathematical modeling, various aspects of which are presented in works [12, 13, 18–20, 42].

The universal nature of machine learning and the optimization of human activity planning based on it determine its global expansion, which, together with robotization, inevitably leads to philosophical understanding of the features and challenges of the digital transformation of society. It is no coincidence that publications on the moral aspects of artificial intelligence implementation are appearing [43]. Obviously, innovations directly affect both ensuring scientific and technological sovereignty, and production tasks of state scale, and ensuring national security, and the sustainable development of civilization.

FUNDING

The work was carried out with financial support from the Ministry of Education and Science of Russia (project code FSUN-2024-0003).

REFERENCES

1. *LeCun Y., Bengio Y., Hinton G.* Deep learning // Nature. 2015, Vol. 521, pp. 436–444.
2. *Weinan E.* Machine learning and computational mathematics // Commun. Comput. Phys. 2020, Vol. 28, pp. 1639–1670.
3. *Dongarra J., Grigori L., Higham N.J.* Numerical algorithms for high performance computational science // Phil. Trans. R. Soc. 2020, Vol. 378, Iss. 2166.
4. *Xu Y., Zeng T.* Sparse Deep Neural Network for Nonlinear Partial Differential // Equations. Numer. Math. Theor. Meth. Appl. 2022, Vol. 16, No. 1, pp. 58–78.
5. *Ilyin V.P.* Mathematical modeling: The philosophy of science // Collection of Popular Science Articles "Mathematics, Mechanics, and Informatics". Moscow, 2017. Pp. 8–16. (In Russ.)
6. *Ilyin V.P.* Artificial intelligence problems in mathematical modeling // Voevodin V., Sobolev S. (eds.) Russian supercomputing days 2019. CCIS – Springer. 2019, Vol. 1129, pp. 505–516.
7. *Forrester A., Sobester A., Keane A.* Engineering Design via Surrogate Modeling: A Practical Guide. Wiley, New York, 2008.
8. *Yanenko N.N., Konovalov A.N.* Some questions of the theory of modular analysis and parallel programming for problems of mathematical physics and continuum mechanics // Modern Problems of Mathematical Physics and Computational Mathematics. Moscow: Nauka, 1982. Pp. 200–217. (In Russ.)
9. *Yanenko N.N., Rychkov A.D.* Topical Problems of Applied Mathematics and Mathematical Modeling. Novosibirsk: Nauka, 1982. (In Russ.)
10. *Ershov A.P., Ilyin V.P.* Software packages: Technology for solving applied problems. Preprint No. 121. Computer Center of the Siberian Branch of the USSR Acad. Sci. Novosibirsk, 1978.
11. *Samarskii A.A., Mikhailov A.P.* Mathematical Modeling. Moscow: Fizmatgiz, 2002. (In Russ.)
12. *Ilyin V.P.* How to reorganize computer science and technology // Vestn. Ross. Akad. Nauk. 2019, No. 2, pp. 232–242. (In Russ.)
13. *Ilyin V.P.* Mathematical Modeling. Part 1. Continuous and Discrete Models. Novosibirsk: SB RAS, 2017. (In Russ.)
14. *Ilyin V.P.* Iterative Preconditioned Methods in Krylov Spaces: Trends of the XXI Century // Computational Mathematics and Mathematical Physics. 2021, Vol. 61, No. 11, pp. 1750–1775.
15. *Ilyin V.P.* Integrated Computational Environment for Grid Generation Parallel Technologies / L. Sokolinsky, M. Zymbler (eds.). CCIS. 2020, Vol. 1263, pp. 58–68.
16. *Butyugin D.S., Ilyin V.P.* CHEBYSHEV: Principles of automating the construction of algorithms in an integrated environment for grid approximations of initial-boundary value problems // Proceedings of the International Conference PAVT'2014. Chelyabinsk, 2014). Pp. 42–50. (In Russ.)
17. *Valiullin A.N., Ganzha V.G., Ilyin V.P., Shapeev V.P., Yanenko N.N.* Problem of automatic generation and investigation of difference schemes in symbolic form on a computer // DAN SSSR. 1984, Vol. 275, No. 3, p. 528–532. (In Russ.)
18. *Ilyin V.P.* Parallel intelligent computing in algebraic problems / Sokolinsky. Parallel Computational Technologies. Communications in Computer and Information Science. 2021, Vol. 1437, pp. 108-117.
19. *Ilyin V.P.* The integrated computational environment for optimization of complex systems // Proceedings of the 15th International Asian School-Seminar "Optimization Problems of Complex Systems" 2019. Pp. 65–67. https://doi.org/10.1109/opcs.2019.888015
20. *Ilyin V.P., Skopin I.N.* About performance and intellectuality of supercomputer modeling // Programming and Computer Software. 2016, Vol. 42, Iss. 1, pp. 5–16.
21. *Ushakov D.M.* Introduction to mathematical foundations of CAD. Novosibirsk, LEDAS Publ., 2008. (In Russ.)
22. *Bastian P., Blatt M., Dedner A. et al.* The Dune Framework: Basic Concepts and Recent Developments, Computers and Mathematics with Applications, 2020. DOI.org/10.1016/j.camwa.2020.06.007
23. OpenFOAM. https://www.openfoam.com/
24. INMOST: A Toolkit for Distributed Mathematical Modeling. https://www.inmost.org
25. *Kleppe A.* Software language engineering: Creating domain-specific language using metamodels. N.Y.: Addison-Wesley, 2008.
26. DVM Systems. http//www.keldush.ru/dvm

27. *Aleeva V.* Designing Parallel Programs on the Base of the Conception of Determinant // Supercomputing. RuSCDays 2018 (Communications in Computer and Information Science). 2019, Vol. 965, pp. 565–577.
28. *Allan B., Armstrong R., Wolfe A. et al.* The CCA Core specification in a Distributed Memory // SPMD Framework Concurrent Practice and Expedience. 2002, Vol. 14, pp. 323–345.
29. *Malyshkin V.E.* Active knowledge, LuNA and literacy for oncoming centuries // LNCS. 2015, Vol. 9465, pp. 292–303.
30. *Ilyin V.P.* The conception and architecture of the knowledge base of the systems of linear algebraic equations // Procced. IX Conference "Knowledge–Ontology–Theory". Novosibirsk: ISI SBRAS, 2023. Pp. 143–154. (In Russ.)
31. *Antonov A., Dongarra J., Voevodin V.* AlgoWiki Project as an Extension of the Top500 Methodology // Supercomput. Frontiers and Innovations. 2018, Vol. 5, No. 1, pp. 4–10.
32. *Yang S., Nachum O., Du Yi. et al.* Foundation Models for Decision Making: Problems, Methods, and Opportunities. ArXiv, abs/2303.04129. [cs. AI] 7 mar 2023.
33. *Grossmann T.G, Komorowska U.J., Latz J., Schönlieb C.* Can physics informed neural networks beat the finite element method? // IMA Journal of Applied Mathematics, 2024, p. hxae011.
34. *Rudikov A., Fanaskov V., Muravleva E. et al.* Neural operators meet conjugate gradients: The FCG-NO method for efficient PDE solving // Proceedings of the 41st International Conference on Machine Learning, ICML 2024, 2024.
35. *Goswami S., Bora A., Yu Y., Karniadakis G.E.* Physics-informed deep neural operator networks // Machine Learning in Modeling and Simulation: Methods and Applications, Springer, 2023, pp. 219–254.
36. *Vaswani A. et al.* Attention Is All You Need // Proceed. of 31st Conference on Neural Information Processing Systems. NIPS. 2017, pp. 1–11.
37. *Srinivasan P., Demuriya O., Grabowski B., Shapeev A.V.* Electronic moment tensor potentials include both electronic and vibrational degrees of freedom // Computational Materials. 2024, Vol. 10, Iss. 1, Id. 41.
38. *Fanaskov V.S., Oseledets I.V.* Spectral neural operators // Doklady Mathematics. 2023, Vol. 108, pp. S226–S232.
39. *Altshuller G.S.* To find idea. The introduction into theory of solving the invent tasks. Novosibirsk: Nauka, 1986. (In Russ.)
40. *Zagorulko Yu., Zagorulko G.* Architecture of extensible tools for development of intelligent decision support systems // New Trends in Software Methodologies, Tools and Techniques. Proc. of the 10th SoMeT-11. Hamido Fujita (Eds.). Amsterdam: IOS Press, 2011. Pp. 457–466.
41. *Bommasani R., Hudsa D.A., Adeli E. et al.* On the opportunities and Risks of Foundation Models. CRFM, Stanford, 2021.
doi: 10.48550/arXiv.2108.07258.
42. *Ilyin V.P.* Programming for the Near Future: Concepts and Pragmatic Considerations // Her. Russ. Acad. Sci. 2023, No. 1, pp. 92–102.
https://doi.org/10.1134/S1019331623010112 (In Russ.)
43. *Luccioni A., Bengio Yo.* On the Morality of Artificial Intelligence. arXiv:1912.11945 [cs.CY] /
doi.org/10.48550/arXiv.1912.11945