The Integrated Computational Environment for Optimization of Complex Systems

Valery P. Il'in
Institute of Computation Mathematics and Mathematical Geophysics,
Siberian Branch of Academy of Sciences
Novosibirsk, Russia
ilin@sscc.ru

***Abstract*—The concept, general structure and main components of the integrated computing environment (ICE) for optimization of complex systems based on the methods of conditional minimization of the objective functional characterizing the initial formulation of the problem of mathematical modeling on supercomputers are considered. The inverse problem is solved based on a sequence of direct initial-boundary value problems, each of which can be a resource-intensive interdisciplinary problem described by a system of multidimensional differential and/or integral (and/or discrete) equations in multi-scale computational domains with multi-connected piecewise smooth geometric configurations of boundaries and contrast material properties of multiphase media. This includes, in particular, the actual problems of surrogate optimization. The functional core of the ICE is focused on a high-performance support of all major stages of a large-scale computational experiment, including geometric and semantic modeling, generation of computational grids, approximation of initial equations, solution of algebraic problems, optimization algorithms for solving inverse problems, post-processing and visualization of results, decision-making on the results of the study. Technical requirements for the ICE involve a flexible expansion of the models and methods used, adaptation to the evolution of computer platforms, effective re-use of external software products, coordinated participation of different teams of developers and focus on a broad demand of the end users from different industries. These qualities are designed to provide a long life cycle and a qualitative increase in productivity in the development and application of the new generation software.**

***Index Terms*—interdisciplinary direct and inverse problems, optimization methods, technological modeling stages, integrated computing environment, software tool environment.**

I. Introduction

The concept of "complex system" is sufficiently broad, and, in general, it can be defined as an object, whose modeling and optimization is a large mathematical and resource-intensive task. Examples of such a system can be, for example, an aircraft, a regional power grid, a supercomputer, oil and gas landfill, etc. Such knowledge-intensive problems are described by systems of differential and/or integral and discrete equations, whose solution in real applications is associated with taking into account multidimensional geometric configurations of computational domains with multi-connected piecewise smooth boundary surfaces and contrast material properties of multiphase media, see [1].

The work is supported by the Russian Foundation of Basic Research, grant N 16-29-15122 ofi-m.

A rapid development of postpetaflops supercomputers the promising appearance of exaflops and zettaflops computers and – in the near future (around 2035) put forward the solution of such vast problems that one could not imagine even a decade ago. At the same time, this phenomenon leads to the need of creating a new generation of programming paradigm [2], [3].

In the recent years, a trend is the transition from the conventional problem - oriented applied program packages (APP) like ANSYS or FEnics [4], [5] to the integrated computational environment (ICE), whose example are: DUNE (Distributed and Unified Numerical Environment [6], the INMOST [7] and the basic system of modeling BSM [8], focused on the support of all the major stages of a large-scale numerical experiment: geometric and functional modeling, generation of computational grids, approximation of the original equations, solving algebraic tasks, optimization approaches to the inverse problems, post-processing and analysis of results of calculation, decision-making on the results of modeling. The implementation of each stage is designed as an instrumental method-oriented environment, developed autonomously and interacting with other software subsystems through the consistent data structures (geometric and functional, grid, algebraic, etc.). High performance and mathematical efficiency of each technological step is provided by means of excessiveness of the functional substance of the corresponding subsystems which gives the opportunity to choose the optimal algorithm in a particular situation and allows to combining such contradict features as efficiency and universality. The scalable parallelism is achieved with the help of hybrid programming on the supercomputers with distributed and hierarchical shared memory by means of interprocessor message passing, multi-thread computing and vectorization of the operations by the AVX (Advanced Vector Technology) approaches.

Optimization methods for solving the problem of conditional minimization of functionals in the framework of the BSM are implemented as the subsystem KANTOROVICH, which includes a representative set of classical and modern algorithms of nonlinear programming, see [9]- [12], at each iterative step assuming the solution of the direct problem, with the directed search for certain parameters. In the recent decades, such approaches as methods of interior points, modified Lagrange multipliers, sequential quadratic programming,

978-1-7281-2986-0/19/$31.00

truth regions, as well as surrogate optimization associated with the implementation of extremely high-tech problems have been specially updated and are being actively developed. Here it is necessary to divide the problems of local and global minimization of functionals, which essentially differ in computational complexity. Another important possible aspect of this problem is the organization of interaction between computer and natural experiments, which requires the coordinated planning of calculations and measurements. Other difficulties associated with the tasks, concern the inaccuracy and incompleteness of initial data, and such issues are overcome based on variational or empirical approaches.

The concept of ICE developments is based on the following methodological principles and technical requirements [13]: a flexible expansion of the content of a implemented models and applied methods, which should provide a wide functionality and maintain an advanced level of a mathematical apparatus, with an effective mapping of high-performance algorithms onto the architecture of multiprocessor computing systems (MPS), an open code for the use, as well as for the effective re-use of external software products which present a high intellectual potential, coordinated participation of different teams of developers, orientation to a wide demand with comfortable conditions of the access for the end users of different professional background.

The considered specifications are designed to provide a long-life cycle of the ICE, as well as a high performance of its use both in the creation and development of new applications, and in their operation for the basic research or in solving practical problems from different fields of activity.

The development of the particular applied products based on the ICE, is implemented by assembling the applications of ready or semi-ready program components, in analogy to the intellectual constructor LEGO. The created package should have the technology and robustness level similar to compiler or operating system. The considered project has an extreme complexity, because of the vast number of the practical problems to be solved, the mathematical statements, the computational methods and technologies, as well as the ways and goals of computer simulation. Its successful fulfilment requires a close cooperation of the specialists of various backgrounds: academician research, education competency and professional software developers.

This study is constructed as follows. Section 2 presents in a sufficiently general form the statements of direct and inverse problems of optimization of complex systems, as well as general features of methods and technologies for their solution on supercomputers. Section 3 deals with the analysis of the architecture and main components of the software under consideration, and finally discusses some of the inevitable scientific and organizational issues and prospects for further research.

II. Mathematical formulation and methods of optimization of complex systems

The problem of optimization of a complex system is abstractly formulated as a problem of conditional minimization of some objective functional

$$\Phi_0(\vec{u}(\vec{x},\ t,\ \vec{p}_{opt})) = \min_{\vec{p}} \Phi_0(\vec{u}(\vec{x},\ t,\ \vec{p})), \tag{1}$$

characterizing the state of the system and defining in a general case, a vector function $\vec{u}$ and dependent on the spatial and temporal coordinates y, x, t , and the vector of initial parameters $\vec{p}$, to be minimized. A concrete system definition of the objective functional can be done in different ways and is the prerogative of a researcher. The search for a minimum in a simple case is unconditional, but in general it is under some linear constraints

$$p_k^{min} \leq p_k \leq p_k^{max}, \quad k = 1, ..., m_1, \tag{2}$$

and when executing a set of nonlinear conditions,

$$\begin{aligned} \Phi_l(\vec{u}(\vec{x}, t, \vec{p}))q \leq \delta_l, \quad l = 1, ..., m_2, \\ \vec{p} = \{p_k\} \in \mathcal{R}^m, \quad m = m_1 + m_2, \end{aligned} \tag{3}$$

where Φ_l present the additional functionals. For example, Φ_0 can be the total energy of the field of an electromagnetic installation with the parameters $p_1, ..., p_m$, in which under the geometric or physical conditions of the form of (2), (3) the field strength is represented by the vector $\vec{u}$.

To close the optimization formulation, it is necessary to set the equation of state of the system, which determines the dependence of the functions $\vec{u}$ on the independent variables $\vec{x}, t$ with a specific representation of the vector of the parameters $\vec{p}$ which gives us the corresponding direct formulation represented by the initial-boundary value problem in the following operator form:

$$L\vec{u} = \vec{f}(\vec{x}, t), \quad \vec{x} \in \bar{\Omega} = \Omega \bigcup \Gamma, \quad 0 < t \leq T < \infty, \tag{4}$$

$$l\vec{u} = \vec{g}(\vec{x},\ t), \quad \vec{x} \in \Gamma, \quad \vec{u}(\vec{x},\ 0) = \vec{u}^0(\vec{x}). \tag{5}$$

Here L is generally a matrix-like operator (differential and/or integral), l is the boundary condition operator, and Ω is the computational domain with a boundary Γ that can have complex configurations that are representable as

$$\begin{aligned} \bar{\Omega} = \bigcup \bar{\Omega}_j,\ \Gamma = \Gamma^e \bigcup \Gamma^i, \\ \Gamma^i = \bigcup \Gamma^i_{j,\ k} = \bigcup (\bar{\Omega}_j \bigcap \bar{\Omega}_k), \end{aligned} \tag{6}$$

where different subdomains of Ω_j and external or internal boundary segments of Γ_e and Γ_i correspond to different functional or integral properties of a simulated object.

In the above-mentioned example of an electromagnetic device, the direct problem (4)-(6) can represent a system of Maxwell's equations in the differential or in the integral form with complex multiscale spatial and dynamic dependencies, which as it is can present an extremely resource-intensive computational problem. This is most typical of real interdisciplinary problems, When it is necessary to calculate interacting fields of different nature: stress-strain states of the bodies described by the Lame systems (elastoplastic theory), turbulent, laminar and supersonic flows (the Navier-Stokes equations in hydro-gas dynamics), heat-mass transfer and filtration equations in multiphase media, etc.

In the recent decades, such a scientific direction has been developed dramatically fast in many applications, based on the so-called surrogate optimization (see [12] and the literature therein). The last mentioned approach corresponds to the situation, when the run time for computing one value of the objective function is too expensive and requires several hours or more. In such a case, the design, or planning numerical experiments is very important, as well as using special methods for approximation of the investigated functionals.

As can be seen even at first glance at formulas (1)- (6), the mathematical formulation of a high-tech problem is a complex logical structure. The initial information is divided into the two main groups: functional and geometric. The first one includes the types of equations to be solved, representations of their coefficients, boundary and initial conditions, descriptions of minimized functionals and constraints for inverse problems, etc. The geometric data must uniquely describe a computational domain, including its subdomains, boundary fragments, edges (surface intersections), and nodes that are the intersection points of the edges.

The initial information should first of all be tested for its completeness and consistency, for example, to meet the conditions of existence, uniqueness and correctness of the solutions, as well as for the presence of any features or singularities that must be taken into account when using numerical algorithms.

It is obvious that the comprehensive study of specific processes and phenomena is an important experimental study with multiple calculations and a variation of the initial data of the problem. This gives rise to various problems of geometric modeling of functional objects. In particular, the relevant areas of analysis arise in the inverse problems related to the optimization of geometric and logical configurations, metric transformation, and discrete exterior differential forms, isogeometric analysis, etc., see [10], [14].

In the last decade, methods of numerical analysis for solving modern problems of mathematical modeling are rapidly developing and largely based on the theoretical achievements of functional analysis, geometry, group theory, etc. In addition, it is now impossible to develop and explore the new algorithms without analyzing high-performance technologies, their parallelization and mapping onto the architecture of heterogeneous MPS with distributed and hierarchical shared memory.

The starting point for solving multi-dimensional problems with real data is the construction of adaptive unstructured grids, including dynamic ones that take into account a priori and a posteriori information about the features of the desired solutions. Practical requirements for the resolution and accuracy of predictive modeling determine the generation of grids with billions of nodes, and modern effective methods are based on domain decomposition and multigrid approaches using combinatorial transformations of grid graphs, which makes such problems not only high-tech, but also sufficiently knowledge-intensive.

In the discretized computational domain, the initial equations can be approximated by a large number of methods: finite differences, finite volumes, finite elements, discontinuous Galerkin methods (FDM, FVM, FEM, DGM, see [1]), etc. For different approaches, it is always important to inherit the important qualities of the initial continuous equations, including the laws of conservation of mass, energy, etc.

An important methodological aspect of this stage is the presence of naturally parallelized element-by-element technologies based on the calculation of local matrices and assembling global matrices.

Since the communication operations on modern MPS are not only slower than the arithmetic ones, but are also most energy-consuming, this has led to the tendency to develop the new algorithms with the least information exchanges, which means the transition to high-order accuracy schemes, despite their large bulkiness, structural and analytical complexity.

The algebraic problems obtained after the discretization represent a "bottle neck" of modeling, since at this stage of numerical solution the amount of required computing resources (both arithmetic operations and memory) is determined in a nonlinear way with increasing the number of degrees of freedom in a problem in question. Therefore, the quality of the scalable parallelization of algorithms is critical here, especially when solving systems of linear algebraic equations (SLAE) of high orders, i.e. tens of billions or higher.

The post-processing and visualization technologies are also extremely time-consuming, but here the situation is saved thanks to the fact that their implementation is naturally carried out on multi-core graphics accelerators GPGPU or INTEL Phi, for example. In conclusion of this section, we must note that in the methodologies of higher-level stages associated with multivariate solutions of direct problems at the stages of optimization and decision-making, the performance of approaches is still in the process of formation, and we consider them to be relevant topics for further research. Here, we should remark that, in general, the cognitive machine experiment represent the complicated computational scheme, with multiple embedded loops, in which the functional and constraints of the inverse statement, the geometric and material properties of the direct tasks, the grid and approximation approaches, and the algebraic solvers can be changed many times. Especially, in the resource consuming modeling the flexible, interactive control of a numerical optimization process is very important.

III. The general structure and components of BSM

In general, the ICE consists of the two main parts: functional and system content. The basic system of modeling is the core of the entire instrumental environment responsible for its application functionality, including the composition of the implemented models and applied methods. Structurally, the

BSM is a set of sufficiently autonomous method-oriented subsystems, each of which supports the corresponding stage of modeling and interacts with the others through consistent data structures.

The VORONOI subsystem is responsible for geometric and functional modeling, which means the formation of mathematical models of the direct and inverse problems to be solved and computational schemes to solve them. Technologically, these issues are related to the computer -aided design and engineering systems (CAD, CAE), which have a huge global market and a tendency to integrate with modeling systems, see [14]. In the BSM, the initial information on the geometry and material properties of the media is set using the IESP (the integrated engineering software platform HERBARIUM [15]). The result of the work of the VORONOI subsystem is a geometric and functional data structure (GDS and FDS), containing all the necessary data of the problem to be solved.

The next computational step is the construction of grids, which is supported by the DELAUNAY subsystem [16], which includes a library of both the original generators and those implemented in various widespread grid systems (NETGEN, GMESH, etc.). There are many kinds of the grids with different types of finite elements, with various distributions of the meshsteps h, a local refinement and multi-grid approaches included. Also, there are a lot of algorithms for the mesh generation, which are based on the frontal principles, on comformal or quasi-conformal transformations on the differential geometry and various metrics, see the review in [16]. In general, the grid computational domain consists of the grid subdomains which can have different types of the finite volumes, and grids in different subdomains can be constructed by different algorithms. The result of this stage is a mesh structure (MDS), consistent with the generally accepted formats in this area, including all the necessary data for solving a problem, but at a discrete level. The set of algorithms and technologies DELAUNAY provides the effectiveness of a variety of grids: adaptive, quasi-structured, dynamic, nested, with different types of elements, etc.

When the mesh computational domain is constructed, the next step is an approximation of the continuous equations, supported in part by the BSM subsystem CHEBESHEV [17], implementing the technologies of FDM, FVM, FEM, discontinuous Galerkin methods for different orders of accuracy, etc. Based on the results of this stage, an algebraic data structure (ADS) is formed, which contains the necessary information about the obtained discrete problems in widespread matrix formats (CSR – a compressed sparse row, for example, as well as the block CSR format for special matrix structures). Note that for large algebraic systems, the ADS is organized immediately in the form distributed on different processors in order to avoid redundant information exchanges.

The solution of algebraic problems of higher orders is one of the well developed areas of computational mathematics, whose many algorithms are implemented as part of the world widespread software libraries, see the review in [18]. Within the framework of the BSM, the development of original algorithms and integration of external "solvers" is carried out with the help of the KRYLOV library [19], which includes parallel domain decomposition methods that are implemented on hybrid programming technologies using the MPI message transmission system between the nodes of distributed memory, multithreaded computations on the cores with shared memory and vectorization of operations by using the AVX command system. Improving the performance of the solvers is also achieved by the detailed code optimization and the active use of high-speed vector-matrix functions from the Intel MKL library. A set of available algorithms and technologies allows one to effectively solve a wide class of algebraic systems: symmetric and asymmetric, real and complex, positive definite and sign-non-definite, of a special block structure, etc.

For post-processing and visualization of grid scalar or vector functions, the available graphic systems are actively used, for example, PARAVIEW. The architecture of the KANTOROVICH subsystem and decision-making tools also includes component principles of encapsulation of widespread specialized software libraries and tools. Here it should be borne in mind that the final stages are associated with large-scale computational experiments, in which parallelization can be carried out at the level of solving numerous resource-intensive direct problems, including those in cloud technologies, when the performance of supercomputers must be evaluated from the point of view of a computer network as a whole.

The main sources and corresponding parts of the ICE are the three inseparable methodological approaches: high-performance computing, big data and artificial intelligence, see [20]. In the course of turning to postpetaflops calculations on many thousands and millions of cores in the problems of scalable computing there is a transition from quantity to quality, and here one should expect for the new approaches and solutions for obtaining a high performance. In particular, programming the algorithms under such conditions already requires taking into account the probability of failure of a computer. As with increasing the computational complexity of the modeling algorithms, and in the process of evolution of computer power volumes of arithmetic operations and information exchanges, as well as the resources of the corresponding computer devices are growing in proportion. As a result, big data are not separately treated (as in purely informational applications), but in interaction with intensive computing. The mission of artificial intelligence in supercomputer modeling is dual-purpose. On the one hand, cognitive technologies are designed to essentially improve the productivity in applied programming at the expense of automating the construction of algorithms and their mapping onto the supercomputers architecture. The second aspect is to increase the level of efficiency and comfort of interfaces with different users for achieving a high technological effectiveness of their work with software products by the principle of a "black or a "grey" box.

The extensive functionality of the BSM and strong requirements for its intellectual level stipulate creating a serious system content of the ICE. The actual question is the control of the computational process in the generated APPs, for which the PYTHON language can be efficiently used, due to its advanced instrumental tools (object-orientation, structural and functional programming, supporting the modules and packages, component and cross-platform technologies, automatically analytic transformations, etc.). The especial importance has the design of mathematical knowledge base, including various practical problems, statements, models, computational technologies, and other anthology concepts (here the project ALGOWIKI [21] presents some prototype). To provide the expressible and contensive external and internal interfaces the development of "factories" for the problem-oriented languages, which in general, in the figurative expression of A. Kleppe [22], will allow the transition from paleo-informatics to neo-informatics.

And finally, the implementation of such a large system as the ICE, with a possibility of multi-version compilation, multi-language, cross-platform compatibility requires using the distributed component object management (DECOM [23]). From the recent experience gained in this direction, we will mention the published work [24].

IV. Conclusion

This paper discusses the problems and solutions to the creation of integrated software tools for supercomputer modeling of processes and phenomena associated with the implementation of direct and inverse problems in the optimization of complex systems. The concept, architecture and basic components of the ICE based on high-performance computing, big data tools and artificial intelligence are proposed. The principles of extenibility of the functional core, adaptation to the evolution of computer platforms, the effective re-use of external software products and coordinated participation of various groups of developers are designed to provide a long-life cycle of the project and its high performance in obtaining the new fundamental knowledge and creating professional technologies in various industries. The proposed basic modeling system should become an ecosystem for coordination and interaction of developers with users from various professional background, including academic research, university competencies and highly qualified developers of software products. The widespread innovation of the ICE is focused on the globalization of predictive modeling as an indispensable attribute of scientific and technological "break-through", the formation of the digital economy, smart industries, new mass professions and national security.

References

[1] V.P Il'in, "Mathematical modelling, Part I: Continuous and discrete models" (in Russian), Novosibirsk, SBRAS Publ., 2017.

[2] M. Asch, T. Moore, R. Badia, M. Beck, P. Beckman , T. Bidot, F. Bodin, F. Cappello, A. Choudhary , B. Supinski, E. Deelman, J. Dongarra, et. al. "Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry", The Internat. J. of High Performance Computing Application, vol. 32(4), pp. 434–479, 2018.

[3] X. Liao, K. Lu, C. Yang, et. al. "Moving from exascale to zettascale computing: challenges and techniques", Front. Inform. Technol. Electron. Eng., vol. 19(10), pp. 1236–1244, 2018.

[4] ANSYS [Online]. Available: http:// www.ansys.com

[5] FEniCS [Online]. Available: http://fenicsproject.org

[6] DUNE.[Online]. Available: http://www.dune-project.org

[7] INMOST: A toolkit for distributed mathematical modeling, [Online]. Available: http://www.inmost.org

[8] V.P. Il'in, "The Conception, Requirements and Structure of the Integrated Computational Environment", V. Voevodin and S. Sobolev (Eds.): RuSCDays 2018, CCIS 965, pp. 653–665, 2019.

[9] P. Giselsson, A. Rantzer (Editors), "Large-Scale and Distributed Optimization", Springer Lecture Notes in Mathematics, 2227, 2018.

[10] M. Delfour, J.-P. Zolesio, "Shape and Geometries. Metrics, Analysis, Differential Calculus, and Optimization", Philadelphia: SIAM Publ., 201.1

[11] A. Forrester, A. Sobester, A. Keane, "Engineering Design via Surrogate Modeling", A Practical Guide, JohnWiley and Sons, New York, 2008.

[12] 12. V.P. ll'in, "On the numerical solution of the direct and inverse electromagnetic problems in geoprospecting" (in Russian), Sib.J.Num.Math., vol. 6, N 4, pp. 381–394, 2003.

[13] V.P. Il'in, "Fundamental Issues of Mathematical Modeling"', Herald of the Russian Academy of Sciences, vol. 86, N 2, pp. 118–126, 2016.

[14] J. Cottrell, T. Hughes, Y. Bazilevs, "Isogeometric Analysis", Towards Integration of CAD and FEA. Singapore: John Wiley & Sons, 2009.

[15] HERBARIUM [Online]. Available: http://tflex.ru/about/publications/detail/index.php?ID=3846

[16] V.P. ll'in, "DELAUNAY: technological media for grid generation" (in Russian) Sib. J. Industr. Appl. Math. N 16, pp. 83–97, 2013.

[17] D.S. Butyugin, V.P. Il'in, "CHEBYSHEV: the principles of automatical construc-tions of algorithms for grid approximations of initial-boundary value problems" (in Russian), Proceed. Internat. Confer. PCT-2014, Chelyabinsk, SUSU Publ., pp. 42–50, 2014.

[18] J. Dongarra, "List of freely available software for linear algebra", [Online]. Available: http://netlib.org/utk/people/JackDongarra/la-sw.ht

[19] D.S. Butyugin, Y.L. Gurieva, V.P. Il'in, D.V. Perevozkin, A.V. Petukhov, "Functionality and algebraic solvers technologies in Krylov library" (in Russian), Vestnik YuUrGU. Series "Computational mathematics and informatics", vol. 2 , N 3, pp. 92–105, 2013.

[20] V.P. ll'in, "Mathematical Modeling and the Philosophy of Science", Herald of the Russian Academy of Sciences, vol. 88, N 1, 81–88, 2018.

[21] ALGOWIKI [Online]. Available: https://algowiki-project.org

[22] A. Kleppe, "Software Language Enginneering: Creating Domain-Specific Language Using Metamodels", New York: Addison-Wesley, 2008.

[23] Maloney J. Distributed COM Application Development Using Visual C++. N.Y.: Prentice Hall, 1999.

[24] Feoktistov A., Kostromin R., Sidorov I.A., Gorsky S.A. Development of Distributed Subject-Oriented Applications for Cloud Computing through the Integration of Conceptual and Modular Programming. In Proceed. of the 41st International Conference on Information and Communication Technology, Electronics and Microelectronics, IEEE, pp. 256–251, 2018.