Numerical methods in the training of aerospace engineers

Elena Kuznetsova[1,*], *Victoria* Kindinova[1], and *Oleg* Krinetsky[1]

[1]Moscow Aviation Institute (National Research University), 4, Volokolamskoe shosse, Moscow, 125993, Russia

Abstract. The paper analyzes the experience of teaching numerical methods in the preparation of engineering students at the Moscow Aviation Institute (National Research University). The necessity of linking the numerical methods studied in the basic course with the topic of a particular specialty is substantiated. The disadvantages of using Ms Excel spreadsheets in the implementation of numerical methods are explained. The necessity of teaching software implementation of numerical methods on a computer is shown. The new American Strategic Computer Initiative is analyzed, as well as the current state of computing power in Russia. The need for the development and implementation of specialized courses in engineering areas for teaching complex methods of numerical analysis is shown. Such learning is recommended to be carried out within the framework of profile physical tasks. It is proposed to acquaint future engineers with the problem of parallelizing numerical algorithms for solving them on a superecomputer. The essence of the engineering approach to the parallelization of numerical algorithms is presented.

1 Introduction

Numerical methods are a powerful tool for solving complex scientific and technical problems using computers. For decades, the Department of Computational Mathematics and Programming has been providing teaching of the discipline Numerical Methods at engineering faculties (institutes) Moscow Aviation Institute (National Research University). The basic course of the discipline includes basic numerical methods for solving linear and nonlinear algebra problems, numerical methods for approximating functions, numerical integration and numerical differentiation, numerical methods for solving ordinary differential equations, as well as numerical methods for solving the Cauchy problem and the boundary value problem for partial differential equations. The discipline is studied in the third year of bachelor's degree and specialist degree of engineering faculties. Teaching includes a lecture course (a two-hour lecture per week) and practical classes in the same volume for one semester. Lectures are based on knowledge from the sections of higher mathematics obtained in junior courses. At the lectures, students study the mathematical formulation of problems, the numerical methods used, the issues of numerical error, convergence, as well as the application of numerical methods for calculating physical systems in a critical state (at points of bifurcation, poor conditionality, transition to another phase state). In practical classes, students get acquainted with examples of problem solving, perform training exercises. As students study the discipline, they perform calculation and graphic work on an individual task, the assessment for

[*] e-mail: oekrin@mail.ru

which is decisive when they put down the final examination grade for the discipline. In the course of training, methodological and teaching aids on the discipline are used, developed by the staff of the Department of Computational Mathematics and Programming [1, 2], as well as textbooks of Russian and foreign specialists recommended by the scientific and technical community [3, 4].

The purpose of research is to analyze the experience of teaching numerical methods in the preparation of engineering students at the Moscow Aviation Institute. Consider the shortcomings and possible prospects for the development of the discipline. As a criterion for the effectiveness of teaching, we will take the competitiveness of graduates of engineering faculties at the international level.

2 Analysis of teaching a basic course of numerical methods

Currently, there is an understanding that it is necessary to link the numerical methods studied in the basic course with the topics of a particular institute. In other words, before describing numerical methods in a mathematical formulation, it is first necessary to present a physical formulation of an engineering problem in which these methods are used. Such an approach, on the one hand, contributes to the development of the student's creative potential, develops in him the ability to find objective cause-and-effect relationships in the surrounding world, the ability to prognostic activity. On the other hand, it increases his motivation, facilitates the process of transition to practical activity in the acquired specialty. This approach was used at the department before [5, 6], but was undeservedly forgotten. It seems appropriate to begin the study of numerical methods with a description of the physical (engineering) tasks inherent in a specific specialty. Let us consider as an example the presentation of numerical methods for students of the Institute No. 2 "Aviation, rocket engines and power plants". When studying the topic of solving nonlinear equations, it will be useful to describe in advance the physical problem of calculating the steady-state surface temperature of an aircraft. And only after that to present iterative methods for solving algebraic and transcendental equations and systems: the method of simple iteration, the secant method, the Newton method. When studying the topic of solving systems of nonlinear equations, it will be useful to first consider a simplified scheme of the main hydraulic subsystem of a rocket engine. And only after that to present iterative methods of numerical solution of systems of nonlinear equations: the method of simple iteration, the Seidel method, the Newton method.

In addition, it will be useful to supplement the training course of a particular institute with a mathematical apparatus that is most adequate for solving the practical problems of this institute. So, for example, for students of Institute No. 2, it is advisable to devote more time to studying the theory of difference schemes that allow solving problems of strength and thermal conductivity, to numerical methods of hydrodynamics. For students of Institute No. 3 "Control Systems, Informatics and Power Engineering", it is desirable to pay more attention to the study of numerical optimization methods, statistical numerical methods, statistical methods for processing test results. For students of the Institute No. 1 "Aviation Engineering", the most interesting are statistical methods such as the Monte Carlo method, which allow you to effectively solve complex integro-differential systems that cannot be solved by classical analytical methods. At present, in this regard, it is practiced by students to write essays on this topic and to defend them in the framework of seminars. However, abstracts are clearly not enough.

Initially, solving seminar tasks and performing settlement and graphic work involved performing calculations manually. In recent years, it has become fashionable to perform these calculations using MS Excel spreadsheets. Many students know the technique of working with the universal MS Excel application from the school course, others master it on their own. The use of MS Excel is welcome, because, on the one hand, it reduces routine computational work and increases the accuracy of calculations, on the other hand, it expands the tools that students master. In addition, students can use the graphical capabilities of MS Office to build flat graphs and spatial surfaces. In MS Excel, there are simple options for varying parameters, facilitating the possibility of conducting numerical experiments. But it is necessary to focus students' attention on the fact that the use of MS Excel has serious drawbacks, the most important of which are the following.

- The logic of the program loses visibility. Formulas expressed in terms of cell addresses, with a large number of them, are difficult to perceive, since cell addresses themselves do not carry any semantic load. This fact makes it much more difficult to check the correctness of the program and understand the formulas out of context.
- Programs in tabular processors are highly dependent on their dimensionality and are rigidly tied to the grid. When adding and deleting rows or columns, it is often necessary to make many changes to various parts of the table. Adding and removing dimensions is so difficult and inconvenient that when developing a tabular program, the author is actually forced to choose the required dimensions in advance and plan far ahead, which contradicts the idea of using a tabular processor as a means of rapid development and modeling.
- It is difficult to differentiate access to data. A user who has access to the table may accidentally or intentionally make changes to it that may disrupt the operation of the program. It is difficult to find out what changes, by whom and when were made to the program.
- The pace of development of tabular programs is significantly reduced due to the fact that the developer has to work at the cell level, which significantly increases the risk of errors. It is difficult to control the input: it is easy to enter data in the wrong format and crash the calculations.
- It is difficult to maintain data integrity. The inability to control corrections increases the risk of errors arising from the inability to track, test and isolate changes.
- Joint development of applications by a team of developers is difficult, it is impossible to cooperate with external systems, there are no notifications, etc.

The reason for these shortcomings is the low level of abstraction of the table processor, when the program developer is forced to operate with such low-level objects as a cell and a formula fixed on a grid. It is necessary to explain to students that it is a mistake to consider spreadsheets as the only means of implementing numerical methods on a computer. In practice, they will inevitably face the need for software implementation of numerical methods using algorithmic languages on modern computers.

It seems appropriate to introduce students to the implementation of these numerical methods programmatically using modern algorithmic languages such as Fortran, Pascal, Delphi, C++, C# on personal computers and workstations. The software implementation of the numerical methods under study is in most cases a non-trivial task. In previous years, the department had a practice of teaching methods of programming engineering problems [7]. Currently, within the framework of the one-semester educational process, it is not possible to introduce students to the most important issues in the field of numerical methods programming. This is regrettable. Nevertheless, in order to prepare students for the difficulties that they will inevitably face when implementing numerical methods programmatically, the following is being undertaken. Students are invited to independently prepare and debug programs for computational tasks using algorithmic languages that they studied as part of the educational process of their institute. Errors that occur during the independent implementation of numerical algorithms are considered and corrected at seminars. Students who have not received sufficient training in programming are given the opportunity to study and debug ready-made programs for typical tasks from the relevant textbooks [8, 9].

3 Current state of computer power

Currently, the computing capabilities of modern computers are rapidly developing. In May 2005, the U.S. government reviewed an analytical report of the Advisory Committee on Information Technology under the President of the United States entitled: “Computational Science: Ensuring America’s Competitiveness” [10]. The report analyzed the potential opportunities for the development of science, industry and the economy, their relationship with achievements in the field of Computational Science. Analysts have identified Computational Science as one of the most important areas providing innovative development of society in the XXI century. The report highlighted the basic components of Computational Science, including:

- models, computational algorithms and modeling programs for solving scientific, technical and humanitarian problems;
- optimization and development of hardware, software and network data processing tools that provide solutions to these computational tasks;
- computer infrastructure to support scientific research and solve technical problems.

The report identified measures aimed at maintaining US national security, including:

- creating promising areas and maintaining a high level of research in the field of computational sciences;
- creation of infrastructure for exploratory research;
- creation of a national repository of data and software products;
- creation and support of national high-performance computing centers.

The new American strategic computer initiative entailed a new stage of competitive struggle in the field of science, education and information technology, determined the direction of development of all countries without exception.

Under the influence of these global processes, the Program “Development of supercomputers and Grid technologies” [11] was adopted in Russia in 2009 in accordance with the decision of the Commission under the President of the Russian Federation on Modernization and Technological Development of the Economy. Its goal is to create a supercomputer computing industry in Russia, including a basic range of supercomputers of various capacities, high-speed communication channels, and training of specialized specialists. The implementation of this program is entrusted to the Rosatom State Corporation. In accordance with this program, supercomputers were installed in the largest Russian state universities, including Moscow (MSU), Tomsk, Saint Petersburg Polytechnic, Nizhny Novgorod, South Ural, Far Eastern and Novosibirsk to support scientific research. In November 2021, Russia entered the top twenty in the world Top 500 ranking, presenting seven powerful supercomputers, one of which is installed at Moscow State University [12]. Currently, there is a tendency to concentrate high-performance computing resources in collective use centers and the development of remote access infrastructure using telecommunications. The Moscow Supercomputer Center of the Russian Academy of Sciences (MSC RAS), the supercomputer center and the grid system in Sarov are examples of such high-performance computing centers with remote access.

The aerospace industry, in the light of the scientific and defense challenges facing it at the present stage, is one of the main consumers of high-performance computing power. Several compact supercomputers created within the framework of the Program "Development of Supercomputers and Grid technologies" are operated in the design bureau of Sukhoi Airlines. On these computers, Sukhoi engineers calculate a significant part of the medium-scale tasks that arise when creating new models of military and civil aircraft. With their help, processes related to aerodynamics, gas dynamics, heat and mass transfer, strength, and safety of aircraft operation are investigated. Large-scale computational problems (aeroelasticity of the structure, flutter, aeroacoustics, connected and conjugate heat and mass transfer, optimization problems) requiring more detailed computer models and large computing resources are solved in remote access mode on a powerful supercomputer installed at the Russian Federal Nuclear Center in Sarov [11].

The rapid growth of computing power, the need to calculate more complex models pose new challenges for universities that train engineering personnel in the aerospace industry, and force them to take a fresh look at teaching Numerical Methods.

4 Learning in complex numerical methods

New high-performance computing technology, large-scale realistic computer models require the use of much more complex methods of numerical analysis. As it became clear at the end of the last century: "the development of mathematics in a certain period of time reflects the nature of computing tools available at that moment to a much greater extent than one might assume" [13, p. 152]. Complex methods of numerical analysis include numerical methods for solving elliptic, hyperbolic and parabolic problems [14]. It makes sense to consider the methods of this class within the framework of the physical formulation of tasks corresponding to each specific field of study.

For example, when training engineers in the field of strength, it makes sense to first determine the task of calculating the strength of the structure. This can be a strength calculation under static and dynamic loading, calculation of natural frequencies and forms of vibrations, calculation of critical loads and forms of loss of stability, two-dimensional and three-dimensional calculation of volumetric, thin-walled and rod structures. Then you need to show which mathematical model corresponds to a specific physical problem. And then you need to describe the numerical methods used: the finite element method, the boundary element method, the spectral element method and their possible variants.

When training engineers in the field of hydroaerodynamics, you first need to describe a physical problem. This may be a transient mode in the boundary layer, separation and attachment of the flow, reduction of aerodynamic drag during takeoff, the interaction of a shock wave with a turbulent boundary layer in a transonic flow. Then you need to show the corresponding mathematical model. Finally, it is necessary to describe the numerical methods used: the finite difference method, the finite volume method, the finite element method, the smoothed particle method, the method using the probability distribution function, as well as the application features of each method.

Particular attention should be paid to the implementation of complex numerical methods on a computer. The implementation of each of them is carried out in three stages:

- preparatory stage of forming the geometry of the model, determining the physical properties, boundary and initial conditions;
- stage of direct calculation;
- stage of analysis and visualization of calculation results.

Both the preparatory phase and the analysis phase are very resource-intensive and time-consuming.

There are foreign and domestic commercial software packages that implement these complex numerical methods for problems in the aerospace industry. However, using these packages without understanding the methods they use can lead to erroneous decisions.

Complex numerical methods can be taught within the discipline, which is a natural continuation and development of the discipline Numerical Methods. Building such a discipline is a difficult task. However, similar specialized courses on numerical methods are already taught at some universities in Russia: at Moscow State University, at the Moscow Institute of Physics and Technology (National Research University), at the South Ural State University. It seems promising to develop and introduce such profile courses for the training of engineers at the Moscow Aviation Institute, although some elements of the study of special numerical methods can be found in the curriculum of a number of institutes of the Moscow Aviation Institute within the framework of the specialist degree and master's degree.

5 Preparation of parallel applications

At the present time, a further increase in the performance of computers only by improving the characteristics of electronic elements has reached the limit determined by physical laws. A further increase in the productivity of computers is possible only through the parallelization of information processing processes. The construction of parallel computing processes that ensure maximum performance is an important independent problem.

Porting a conventional program to a multiprocessor computing system may not provide the expected performance gain. Moreover, as a result of such a transfer, the program may run even slower. One of the authors of the article faced the problem of transferring a sequential program performing strength calculations by the finite element method to a supercomputer back in the 90s of the last century [15, 16].

It is extremely difficult to write an effective parallel program or adapt an existing serial program for parallel computing on multiprocessor systems. This requires, on the one hand, to perform a structural analysis of the algorithm, to identify its internal parallelism. Such an analysis requires a deep understanding of the essence of the task. On the other hand, when developing parallel algorithms for composing effective code, it is necessary to know the features of the computing system on which the algorithm will be implemented. In particular, knowledge of the topology of connections, the type of processor, the relative memory volumes of caches of various levels, etc. is required.

The problem is that there are not enough specialists who are equally proficient in both the subject of research in their "own" field of knowledge and knowledge in the field of high-performance computing systems. Therefore, an approach is practiced that allows the task to be divided into two independent tasks. The first task is the task of parallel computing. The task is to perform a structural analysis of the algorithm, identify the parallelism in it and make a time diagram of its execution. This task should be performed by a highly qualified engineer, a specialist in the subject area. The second task is the task of parallel programming. The task is to compile effective code and debug the program according to the time diagram developed by the engineer. This task is performed by a programmer who knows the architecture of the computing system well.

Moreover, the inverse problem can be posed. If there is a choice of calculators, then to build the most efficient parallel algorithm, you can set the task of determining the calculators that most fully implement the features of the parallel algorithm.

The approach under consideration allows us to study the construction of parallel algorithms and programming issues independently. When studying parallel computing, the issues of analyzing the effectiveness of the parallel structure of the algorithm, building models and schemes of parallel computing, as well as the analysis of achievable acceleration and efficiency for various applied computing tasks are considered. The solution of the parallel computing problem is a description of a parallel algorithm in the form of flowcharts, graphs and diagrams that are understandable to the programmer and can be used by him to write a parallel program.

In recent years, efforts have been made to expand the training of specialists in the field of problem solving on multiprocessor supercomputers. Since the solution of the problem of parallel computing is assigned to an engineer, let's consider the essence of the engineering approach [17].

To solve the problem of parallel computing, it is necessary to first study the architecture of a parallel computing system.

Currently, the vast majority of parallel computers are implemented in the MIMD architecture class (multiple instruction streams and multiple data streams) [17, 18]. Among them, the following main subclasses are distinguished:

- Vector-conveyor computers. This subclass uses a set of vector commands that perform operations with arrays of independent data in one clock cycle. Example—CRAY.
- Massively parallel with distributed memory. In this subclass, microprocessors, each having its own local memory, are connected through some communication medium.
- Symmetric multiprocessors (SMP), consist of a set of processors having divide shared memory with a single address space and operating under the control of a single operating system.
- Clusters are formed from computing modules of any of the types discussed above, united by a system of connections or by means of shared external memory.

To describe the information dependencies of the algorithm for solving the problem, an algorithm model is used in the form of an acyclic oriented graph (algorithm graph). In this model, the set of algorithm operations and the information dependencies existing between the operations describe the set of vertices and arcs of the graph. The vertices represent the operations of the algorithm, and the arcs of the graph establish a partial order of operations. The arcs of the graph are assigned weights reflecting the intensity of information exchange between the corresponding vertices—operations.

The computing system is also represented as a graph. In the graph of a computing system, vertices are a set of processors, arcs represent communication lines between processors. The bandwidth of communication lines is characterized by the weights of the arcs of the graph.

The effectiveness of implementing specific calculations on a specific system depends on how much the structure of the algorithm corresponds to the architecture of the computing system. The central problem in the planning of computing resources is the problem of mapping a parallel algorithm and its corresponding program to the architecture of a multiprocessor computing system.

The second task is to build a schedule for the parallel implementation of the algorithm, in which each operation is assigned the number of the processor used for its execution and the start time of the operation.

In order for the schedule to be realizable, it is necessary that the same processor was not assigned to different operations at the same time; by the designated time of the operation, all the required data has already been calculated. A set of parallelization methods and techniques that do not require a strict solution to the problem of mapping the graph of an algorithm to the graph of a computing system was called an engineering approach.

The algorithm's ability to parallelize is potentially related to one of two internal properties of the algorithm, which are characterized as task parallelism (message passing) and data parallelism (data parallel). When task parallelism, a computational task is divided into several relatively independent subtasks, each of which is loaded into its own processor and implemented independently, but uses common data and/or exchanges the results of its work with other subtasks. If there is a data parallelism property in the algorithm, one operation can be performed on all elements of the data array at once. In this case, different fragments of the array can be processed independently on different processors. A number of algorithms have both task parallelism and data parallelism to one degree or another. The following types of algorithms are distinguished:

- Algorithms using Data Parallelism. They are characteristic of numerical algorithms for processing large arrays represented as vectors and matrices.
- Algorithms with data distribution (Data Partitioning). Data parallelism is used, in which the data space can be divided into disjoint areas. Independent processes are associated with each area. A rare exchange between processes is required.
- Relaxation algorithms (Relaxed Algorithm). The algorithm can be represented as independent processes without synchronization of communication between them, but processors must have access to shared data.
- Algorithms with synchronization of iterations (Synchronous Iteration). Many of the standard numerical iterative parallel algorithms require synchronization at the end of each iteration. Synchronization means that permission to start the next iteration is given after all processors have completed the previous iteration.
- Self-replicating tasks (Replicated Workers). For tasks of this class, a central pool (storage) of similar computing tasks is created and maintained. This technology is typical for graph or tree research.
- Pipelined Computing. This type of computation is typical for processes that can be represented in the form of some regular structure.

With the engineering approach, the following stages of parallel applications preparation technology are recommended:

- decomposition of the task into subtasks that are implemented independently;
- definition of information interactions for the generated set of subtasks;
- scaling of subtasks, determining the number of processors;
- determining the architecture of the system, assigning subtasks to processors, scheduling.

The execution time of the parallel algorithm is determined by the maximum time value in the schedule. For a fixed graph of calculations, it is advisable to build a schedule that ensures the minimum execution time of the algorithm. To reduce the implementation time of the algorithm, various computational schemes are considered. Each scheme has its own schedule.

The considered approach is implemented in the educational process of Samara State Aerospace University named after S.P. Korolev within the framework of an independent discipline on parallel computing. Such a discipline is a natural continuation and development of the discipline of Numerical Methods. It also seems useful at the Moscow Aviation Institute at the engineering faculties to supplement the discipline of Numerical methods with parallel computing sections.

6 Conclusion

The Moscow Aviation Institute (National Research University) is one of the leading universities in Russia that train engineers for the aerospace industry. The discipline Numerical methods at engineering faculties allows students to systematize the knowledge acquired in the study of computer modeling and algorithmization, programming languages and methods, modern information technologies, as well as apply the knowledge gained to solving problems of computational mechanics. A critical analysis of the experience of teaching the discipline at engineering faculties allows us to assess the compliance of teaching results with the requirements of state policy in the field of training competitive engineering personnel for the aerospace industry. The paper suggests possible prospects for the development of the discipline. Movement in this direction will contribute to the development of students' creative abilities, love for their chosen profession, and the opportunity to fully realize their creative potential. Familiarization of students with supercomputer technologies in the educational process is an effective way to involve students in the real practice of scientific research in the aerospace industry.

References

[1] U.G. Pirumov, *Numerical methods* (MAI Publishing House, Moscow, 1998)

[2] M.A. Cherkasov, *Numerical methods. Problem solving* (MAI Publishing House, Moscow, 2007)

[3] B.P. Demidovich, I.L. Maron, *Fundamentals of computational mathematics*, 3rd ed. (Nauka, Moscow, 1966)

[4] N.V. Kopchenova, I.A. Maron, *Computational mathematics in examples and problems* (Pal, Saint Petersburg, 2008)

[5] Yu.N. Badlevsky, P.V. Biryukov, T.Ya. Kosheleva et al., *Numerical methods and algorithms for solving problems of rocket technology*, ed. by T.Ya. Koshelevoy (MAI Publishing House, Moscow, 1990)

[6] V.I. Kireev, *Numerical methods for solving problems of algebra and approximation theory* (MAI Publishing House, Moscow, 1991)

[7] V.E. Zaitsev, O.I. Korchagina, E.A. Kuzmenkova, M.V. Rotanina, Yu.A. Shebeko, *Methods of programming engineering problems in the FORTRAN language* (MAI Publishing House, Moscow, 1988)

[8] Yu.V. Vasilkov, N.N. Vasilkova, *Computer computing technologies in mathematical modeling* (Finance and Statistics, Moscow, 2004)

[9] A.M. Mudrov, *Numerical methods for PCs in Basic, Fortran and Pascal languages* (RASKO MP, Tomsk, 1991)

[10] *Computational Science: Ensuring America‘s Competitiveness* (President‘s Information Tehnology Advisory Committee, May 27, 2005)

[11] *Supercomputers for aircraft*, All-Russian weekly newspaper VPK, 19.12.2012, URL `https://vpk.name/news/81078_superkompyutery_dlya_samoletov.html`

[12] Rating_supercomputers_Russia_and_SNG_Top50,04.01.2020, URL `https://www.tadviser.ru/index.php`

[13] T. Toffoli, N. Margolus, *Machines of cellular automata* (Mir, Moscow, 1991)

[14] V.F. Formalev, D.L. Reviznikov, *Numerical methods* (FIZMATLIT, Moscow, 2006)

[15] E.V. Kuznetsova, *A vectorized algorithm for solving systems of linear algebraic equations by the frontal method in the FEM package "Composite"* (Mobile software, Tver, 1991)

[16] E.V. Kuznetsova, *Research and development of a transportable version of the fortran package for strength calculations by the finite element method*, in Candidate of Physical and Mathematical Sciences: 05.13.11, 05.07.03

[17] V.P. Gergel, V.A. Fursov, *Lectures on parallel computing* (SSAU Publishing house, Samara, 2009)

[18] J.L. Hennessy, D.A. Patterson, *Computer architecture. Quantitative approach*, 5th ed. (Elsevier, 2012)