To Solving Problems of Mathematical Modeling of Non-stationary Processes on High-performance Computers

Oleksandr Duchenko [1,†], Oleksandr Khimich [1,†], Alla Nesterenko[1,†] and Oleksandr Popov [1,†]

[1] *Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine, Academic Glushkov Avenue, 40, Kyiv, 03187, Ukraine*

Abstract

Mathematical modeling is currently the main tool of studying objects, processes and phenomena of various natures. The resolution of applied problems often requires studying of mathematical models of different processes. In particular, mathematical modeling is essential for simulation of non-stationary processes when analyzing the strength of complex structures and heat transfer processes. Non-stationary problems of strength analysis of complex structures or their components arise when addressing issues such as extending the lifespan of structures, determining their reliability, and identifying the properties and behavior of complex structures subjected to combined loads (force, thermal). Nowadays, these problems are among the most resource-intensive problems in mathematical modeling. This is due to increasing demand on the quality of design solutions, the necessity of performing calculations for complex and unique structures, and the use of new structural materials etc. Improving the quality and efficiency of mathematical modeling is only possible through the use of fundamentally new and detailed models and by shifting from simulating individual elements to studying the object as a whole. The use of detailed computer models leads to a significant increase in the size of discrete models and the corresponding computational problems, which exceed the capabilities of modern personal computers and workstations. Therefore, in the mathematical modeling of non-stationary processes of various natures it is important to develop new methods and algorithms for parallel computing on high-performance heterogeneous computers, in particular those of hybrid architecture. To effectively utilize such high-performance systems, it is required to develop new-generation computer algorithms and software. The development of algorithmic and software tools is a key direction in creating high-performance computational resources. This paper proposes a methodology for solving computational problems (Cauchy problems) of mathematical modeling of non-stationary processes on high-performance computers with parallel organization of calculation, using the example of mathematical modeling in continuum mechanics.

Keywords

Mathematical modeling, non-stationary processes, high-performance computing, Cauchy problems for systems of ordinary differential equations.

1. Introduction

Nowadays, mathematical modeling is one of the main tools for researching objects, processes, and phenomena of various natures. The need to study mathematical models of various processes often arises when solving applied problems. Thus, the application of such mathematical models is essential for the research, assessment and diagnostics of the stress-strain state of various structures such as buildings, welded and other constructions. Non-stationary problems of strength analysis of complex structures or their components arise when addressing issues such as extending the lifespan of structures, determining their reliability, and identifying the properties and behavior of complex structures subjected to combined loads (force, thermal). In addition, such models are used in the researching of thermal processes that occur

Workshop "Intelligent information technologies" UkrProg-IIT`2025 co-located with 15th International Scientific and Practical Programming Conference UkrPROG'2025, May 13-14, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

duchenkostud@gmail.com (O. Duchenko); khimich505@gmail.com (O. Khimich); alla.nest1958@gmail.com (A. Nesterenko); alex50popov@gmail.com (O. Popov)

0009-0009-4157-1743 (O. Duchenko); 0000-0002-8103-4223 (O. Khimich); 0000-0001-6174-1812 (A. Nesterenko); 0000-0002-1217-2534 (O. Popov)

during welding of complex structures, various manufacturing processes, as well as in nanostructures to optimize the cooling of electronic systems, etc.

Solving computational problems related to the modeling of non-stationary processes in strength analysis of complex structures and heat transfer processes requires significant computational resources. This is due to increase in demand for the quality of design solutions, the need to perform calculations for complex and unique structures, the use of new construction materials, etc. Ensuring the reliability of computer simulation results is another resource-intensive factor, which depends on the accuracy of the mathematical models used, as well as accuracy of the data provided, the size of the computational problems etc. Improvement in quality and efficiency of mathematical modeling is possible only through the use of new detailed models and shift from modeling separate stages of processes to modeling the entire process as a whole. The use of detailed computer models leads to a significant increase in the size of discrete (finite-dimensional) models and the corresponding computational problems.

Considering these problems in such a setting leads to computational problems involving huge volumes of data, for example, matrices with orders exceeding 10^7, and increasing the adequacy of computer modeling is usually accompanied by an exponential increase in research costs.

Modern personal computers and workstations lack the resources necessary to implement these tasks. Nowadays, the increase in computing performance is achieved through parallelization of calculations, based on the use of heterogeneous computers with, in particular hybrid architecture.

In computers with hybrid architecture, both MIMD and SIMD models of parallel computation are used, and multi-core processors are complemented by co-processor accelerators. Nvidia and AMD, as leading companies in high-performance hardware, have proposed the use of graphics processing units (GPUs, graphics cards) as such accelerators.

The use of heterogeneous computing systems is one of the promising directions in the development of high-performance computing. A number of such hybrid computers of varying performance levels have been created, ranging from personal hybrid supercomputers for local use to high-performance clusters. It is necessary to develop next-generation computer algorithms and software to effectively utilize such high-performance systems. The development of algorithmic and software tools is a priority in building high-performance computational resources.

This article proposes a methodology for solving computational problems (Cauchy problems) of mathematical modeling of non-stationary processes on high-performance computers with parallel organization of computation, using the example of mathematical modeling in the continuum mechanics.

2. Mathematical models of non-stationary processes

Mathematically, the problem of calculating the stress-strain state (a dynamic problem in the theory of elasticity), using the principle of virtual displacements, can be formulated as a variational problem [1]: find a vector-function of displacements $u \in U_0$ that, for any vector-function $v \in U_0$ (any admissible displacement), satisfies the integral identity

$$a(u,v) + b(u'',v) + c(u',v) = l(f,v), \qquad u(t_0) = u(0), \;\; u'(t_0) = u^{(1)}; \tag{1}$$

where $U_0([t_0,t_k],\Omega)$, $\Omega \subset R^3$ is an infinite-dimensional functional space of possible displacements, and the symmetric bilinear functionals $a(u,v)$, $b(u'',v)$, and $c(u',v)$ are proportional to the potential and kinetic energies, and damping, respectively. The linear functional $l(f,v)$ is proportional to the work done by the applied (external) forces under load. Here, u' denotes the first derivative of the vector-function $u(t,s)$ and u'' the second derivative.

The heat transfer process is modeled by the following initial-boundary value problem in the domain $\Omega \subset R^3$ with boundaries Γ and $t \in [t_0,t_k]$

$$c\frac{\partial T}{\partial t} = \frac{\partial}{\partial x_1}\left(\lambda\frac{\partial T}{\partial x_1}\right) + \frac{\partial}{\partial x_2}\left(\lambda\frac{\partial T}{\partial x_2}\right) + \frac{\partial}{\partial x_3}\left(\lambda\frac{\partial T}{\partial x_3}\right) + w, \;\; \mathrm{x} = (x_1,x_2,x_3) \in \Omega, \tag{2}$$

$$l(T) = g(t,x), \qquad x \in \Gamma, \;\; T(t_0) = T_0, \qquad t \in [t_0,t_k],$$

where $l(T)$ is the boundary condition operator, $c(T)$ is the specific heat capacity, $\lambda(T)$ is the thermal conductivity of the material, $w(t)$ is the volumetric power of the heat source.

The main approach to solving such problems are numerical methods. To do this, the original problem in an infinite-dimensional space is replaced with its finite-dimensional (discrete) analogue. Most often, projection methods are used for this purpose, for example, the finite element method or finite difference method [2, 3]. As a result of discretization, a Cauchy problem for a system of ordinary differential equations (ODEs) is obtained.

For a non-stationary strength analysis problem using the finite element method, a second-order system of ODEs is obtained:

$$Bx''(t) + Cx'(t) + Ax(t) = b(t), \quad x(t_0) = x^{(0)}, \quad x'(t_0) = x^{(1)}. \tag{3}$$

For the heat transfer problem, using the finite element method or the finite difference method, we obtain the following matrix Cauchy problem for a first-order system of ODEs.

$$Cx'(t) + Ax(t) = b(t), \qquad x(t_0) = x^{(0)}, \quad x'(t_0) = x^{(1)}. \tag{4}$$

Here, $x(t)$ is the desired n-dimensional solution vector of the system of ODEs (e.g., displacement values for (3) or temperature values for (4)), $b(t)$ is an n-dimensional right-hand side vector (e.g., loads or heat source power), and $A(t)$, $C(t)$, $B(t)$are $n \times n$ matrices with sparse structure, where n is the dimension of the systems of ODEs (3) or (4).

Discrete problems (3), (4) exhibit a number of features, in particular:

- high order of matrix dimensions in the discrete problems (up to tens of millions);
- the matrices have a sparse structure (e.g., banded, envelope, skyline, etc.);
- the elements of the matrices and vectors are computed with errors, caused by initial data inaccuracies, discretization errors, and computational errors when evaluating these elements on a computer.

3. Solving Cauchy problems for systems of ODEs

For the numerical solution of Cauchy problems for systems of ODEs, depending on their properties, a variety of time integration methods exist. Direct integration methods are used, particularly Runge–Kutta methods of various orders for first-order systems, and the Wilson-θ method of for second-order systems. Additionally, for second-order systems, some software tools (see, for example, [2]) utilize the Fourier method based on expanding the desired functions in terms of the structure's natural vibration modes.

Runge–Kutta methods are also used to solve second-order systems of ODEs by introducing auxiliary functions $y = x'$. This transforms the system into a first-order system of ODEs of twice the size:

$$y = x',$$

$$y' = -B^{-1}\big(Ax + Cy - b(t)\big).$$

Let's consider solving the Cauchy problem for a first-order system of ODEs using the Runge–Kutta method:

$$\frac{dv}{dt} = f(t, v) \qquad v(t_0) = v_0 \tag{5}$$

where $v(t) = (v_1, v_2, \dots, v_n)^T$ is the desired vector-function solution.

Among the Runge–Kutta methods, the fourth-order Runge–Kutta method is most commonly used, as it provides the required accuracy with relatively low computational complexity. This method is implemented over the time interval $[t_i, t_i + h_i]$ using the formulas:

$$v^{(i+1)} = v^{(i)} + (k_1 + 2k_2 + 2k_3 + k_4)/6 \tag{6}$$

where

$$\begin{aligned} k_1 &= h_i f\left(t_i, v^{(i)}\right), \\ k_2 &= h_i f\left(t_i + h_i/2, v^{(i)} + 0.5k_1\right), \\ k_3 &= h_i f\left(t_i + h_i/2, v^{(i)} + 0.5k_2\right), \\ k_4 &= h_i f\left(t_i + h_i, v^{(i)} + k_3\right). \end{aligned} \tag{7}$$

i = 0,1,2,... the upper index is the point number, lower index is the component of the vector.

Thus, the majority of arithmetic operations are performed to compute the right-hand side of the vector function $f(t, v)$.

The Wilson-θ method is used for the numerical solution of second-order systems of ODEs with initial conditions (3). To apply it, we rewrite the problem (3) as:

$$BU'' + CU' + AU = b(t) \tag{8}$$

with initial conditions

$$U'(0) = U'_0, \qquad U(0) = U_0. \tag{9}$$

Here, U is the vector of the values of the function u at the nodes of the finite element mesh.

In the Wilson-θ method, the time interval T is divided into n equal subintervals $h = T/n$. At each time interval $[t_i, t_i + \theta h_i]$, i = 0,1,..,n-1, $\theta \geq 1$ a linear change in acceleration is assumed (the second derivative is a linear function of time).

It is assumed that at the initial moment of time $t_0 = 0$, the vectors U_0, U'_0, U''_0 are known. Before starting integration over time, the integration step h is determined, required constants, the effective stiffness matrix $A_e = A + a_0 B + a_1 C$ and its decomposition $A_e = LDL^T$ are computed.

Then the integration of the system (8)–(9) over time at point $t_{i+1} = t_i + \theta h$ is performed according to the scheme:

- computation of the right-hand side of the system of linear algebraic equations $A_e u_{e,i} = b_{e,i}$ using

$$b_{e,i} = b_1 + \theta(b_{i+1} - b_i) + B(a_0 U_i + a_2 U'_i + 2U''_i) + C(a_1 U_i + 2U'_i + a_3 U''_i)$$

- using the LDL^T decomposition of matrix A_e, computation of u_e as a solution of the system $A_e u_{e,i} = b_{e,i}$;

- computation of the approximate solution U_{i+1}, and its first and second derivatives using following formulas:

$$U''_{i+1} = a_4\left(u_{e,i} - U_i\right) + a_5 U'_i + a_5 U''_i;$$

$$U'_{i+1} = U'_i + a_7(U''_{i+1} + U''_i);$$

$$U_{i+1} = U_i + \theta U'_i + a_8(U''_{i+1} + 2U''_i).$$

Thus, this method consists of several sub-tasks requiring a significant number of arithmetic operations: forming the matrix A_e, its LDL^T decomposition, and repetitive solving of N systems of equations of the form $A_e u_{e,i} = b_{e,i}$.

Another method used to solve the initial value problem (3) is the Fourier method. This method is based on decomposition by the eigenvectors of the algebraic eigenvalue problem (AEVP):

$$Az = \lambda Bz \tag{10}$$

An approximate solution to problem (3) is represented as a linear combination of several eigenvectors corresponding to the smallest eigenvalues of problem (10). That is, if λ_k, z_k $\left(z_j^T B z_k = \delta_{jk},\ j,k = 1,2,\dots,n\right)$ is a solution of AEVP (10), then assuming $x(t) = \sum_{k=1}^{n} y_k(t) z_k$ in (3), we obtain (under certain assumptions regarding the damping matrix C [2]) a system that decomposes into equations independent with respect to $y_i(t)$:

$$y''_k(t) + 2\xi_k\omega_k y'_k(t) + \omega_k^2 y_k(t) = P_k(t),\ t > t_0,$$
$$y_k(t_0) = y_k^{(0)},\ y'_k(t_0) = y_k^{(1)},$$

where $0 < \xi_k < 1,\ \omega_k^2 = \lambda_k,\ P_k(t) = z_k^T b(y),\ y_k^{(0)} = \left(x^{(0)}\right)^T Bz_k,\ y_k^{(1)} = \left(x^{(1)}\right)^T Bz_k,\ k = 1,2,\dots,n_0.$

A significant contribution to the solution of problem (3) is usually provided by only about 20–30 terms corresponding to the smallest eigenvalues. Thus, we obtain a system of n_0 second-order ordinary differential equations that are mutually independent. The solution to this system can often be found analytically, or it can be easily integrated using, for example, the fourth-order Runge–Kutta method. Since such a problem exhibits inherent parallelism, it can be efficiently implemented on a parallel computer.

The most computationally intensive part of this approach is solving the partial AEVP, which is best done using subspace iteration methods [4].

It is worth noting that both the Wilson θ-method and the Fourier method allow for multiple solutions of the Cauchy problem for second-order system of ODEs (3) with different right-hand sides (i.e., different loads) $b(t)$.

4. Features of parallel algorithms for Cauchy problems for systems of ODEs

For effective implementation of mathematical modeling, it is advisable to use high-performance computing tools — both hardware and corresponding software. Modern high-performance computing systems with parallel processing architectures — ranging from personal computers with a single multi-core processor and multiple graphic accelerators to supercomputers with a massive number of processing units of various architectures — utilize different models of parallel computation, dynamic computing environment, network technologies, etc. [5].

As the capabilities of modern computers continue to grow, approaches to designing parallel algorithms are also evolving. These algorithms must take into account both the properties of the problem and the architectural features of the computing resources, including memory structure of the processing units, interconnections between them, synchronization of computations and data exchanges, and other relevant factors.

Depending on the method used to solve Cauchy problems for systems of ODEs, as mentioned above, the computational load varies across different types of operations. In most algorithms, these are linear algebra operations: matrix-vector and matrix-matrix operations, matrix decompositions, etc.

For example, in the 4th-order Runge–Kutta method, the most computationally intensive operation in terms of resource consumption and runtime is the multiplication of a sparse matrix by a vector. In the Wilson-θ method, it is the LDL^T decomposition of a symmetric positive-definite matrix and the repeated solution of the corresponding system of linear algebraic equations. In the Fourier method — solving the AEVP.

As previously mentioned, the matrices involved in real-world computational problems often have very large orders and sparse structures, such as banded or envelope etc [6]. These factors influence the selection of methods, algorithms, and tools for solving the Cauchy problems for systems of ODEs. Therefore, in order to choose the optimal parallel solution algorithm and reduce the required computational resources (runtime, memory etc.), it is necessary to identify the data structure (both of the original matrix and, if necessary, of the decomposition matrices). To enhance efficiency, the matrix structure needs to be optimized if required. To recognize sparse data structures, this article recommends the use of neural networks and machine learning. Structural regularization algorithms [6] allow transformation of arbitrary matrix structures into one of the regular sparse forms (block-banded, block-skyline, block-diagonal with framing, etc.). The solution algorithm is determined based on the resulting data structure.

As mentioned above, solving Cauchy problems for systems of ODEs involves processing of extremely large data volumes. It is possible to efficiently execute large volumes of homogeneous operations by using block versions of methods and algorithms. Such approach consolidates a significant number of operations into matrix-matrix or matrix-vector operations with dense blocks, which are obtained from structural regularization of the data. To implement these operations, it is recommended to use software tools provided by hardware developers, which are optimized for performing block matrix operations on the corresponding processing units [7, 8].

Important stages of development a parallel algorithm for hybrid systems include selection of an efficient data representation method and consideration of the specific features of architecture of hybrid computers. The increasing number of processors in parallel computers and the creation of new hybrid heterogeneous architectures significantly increase communication overhead and reduce process efficiency. Therefore, taking into account the computer architecture is a necessary condition in designing a parallel algorithm.

The data is distributed among processing units based on the choice of the parallel solution algorithm. When designing algorithms for sparse matrix problems, it is crucial to choose the appropriate storage and representation methods for non-zero elements. These methods are determined by the sparse matrix's structure and the requirement of the solution algorithm. Data distribution and storage schemes are used to ensure compact data representation, fast access and processing of large datasets, and minimization of data exchange between processing units.

The amount of calculations for each processing units in use should be approximately the same — this will ensure uniform computational load (balancing) of the processing units. Additionally, subtask distribution among processing units should aim to minimize the number of information dependencies (communication interactions) among subtasks.

Thus, to efficiently solve Cauchy problems for systems of ODEs arising in the mathematical modeling of non-stationary processes, it is recommended to use intelligent software tools [4, 9].

To summarize, the design of parallel algorithms for solving Cauchy problems for systems of ODEs intended for implementation on computers of hybrid architecture, requires to follow the steps below:

- recognize sparse data structures of the matrices of the given problem and, if necessary, perform their structural regularization;
- identify the architecture that is most efficient for solving the given problem;
- divide the problem into subtasks, identify information dependencies between them, and select the appropriate parallelization environment;
- take into account the memory structure of the processing units to ensure high algorithm performance;
- distribute data and computations across processing units to ensure balanced load on the computer's computing elements.

5. Conclusions

The paper proposes the basic principles for efficient mathematical modeling of non-stationary processes of various natures in a variable computational environment on modern high-performance hybrid computer systems. During development of algorithmic and software tools for solving computational problems (Cauchy problems for systems of ODEs) in the modeling of non-stationary processes it is necessary to:

- consider the architecture and technical features of hardware and the specifics of the software tools developed by hardware manufacturers;
- consider the mathematical properties of the problem being solved that ensure the reliability of computational results;
- provide automatic selection of an efficient variable computational environment (a number and types of processor devices, their interconnections, synchronization of computations and data exchanges, memory types, etc.);

- utilize elements of artificial intelligence (in particular, artificial neural networks, machine learning, etc.);
- utilize multi-precision arithmetic and variable-precision arithmetic;
- utilize block versions of corresponding algorithms which involve the allocation of subtasks with large volumes of homogeneous arithmetic operations.

It is recommended to use these principles to develop or modify algorithmic and software tools for resolution of a wide range of problems of computer modeling of processes, phenomena and objects from various subject areas on the latest high-performance computer systems with parallel organization of calculations. In the future, it is advisable to develop algorithms using the NVIDIA-library NCCL (NVIDIA Collective Communication Library) for collective communications, which easily integrates into modern parallel programs. This library is designed for high efficient organization of data exchange between graphic processing units, taking into account the topology of the computing system.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] S.P. Timoshenko, J. Goodier, Theory of elasticity, M., Nauka, 1975.

[2] A. S. Gorodetsky, I. D. Evzerov, Computer models of structures, FACT, Kyiv, 2007, 394 p.

[3] K. J. Bathe, E. L. Wilson, Numerical Methods in Finite Element Analysis, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1976, 528 p.

[4] A.N. Khimich, I.N. Molchanov, A V. Popov, T.V. Chistyakova, M.F. Yakovlev, Parallel algorithms for the solving of computational mathematics problems, Naukova Dumka, Kyiv, 2008.

[5] Top500 – The list. URL: http://www.top500.org/

[6] A. George, J. W-H Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1981.

[7] Intel® Math Kernel Library (Intel® MKL). URL: https://software.intel.com/en-us/intel-mkl

[8] CUBLAS Library documentation. URL: https://docs.nvidia.com/cuda/cublas/index.html

[9] O. M. Khimich, T. V. Chistjakova, V. A. Sidoruk, P. S. Ershov, Adaptive computer technologies for solving problems of computational and applied mathematics, in: Cybernetics and Systems Analysis, 2021, volume 57 (6), pp. 990–997. https://doi.org/10.1007/s10559-021-00424-z.