COMPUTATIONAL PROBLEMS OF ELECTRICAL ENGINEERING
Vol. 14, No. 1, 2024

# APPLICATION OF PARALLEL COMPUTING TECHNOLOGY FOR MODELLING COMPLEX DYNAMIC OBJECTS

**Petro Stakhiv[1], Bohdan Melnyk[2], Oksana Hoholyuk[1], Stepan Trokhanyak[2]**
[1] Lviv Polytechnic National University, Lviv, Ukraine, [2]Ivan Franko National University of Lviv, Ukraine
*petro.h.stakhiv@lpnu.ua, bohdan.melnyk@lnu.edu.ua, Oksana.p.hoholyuk@lpnu.ua, stepan.trokhanyak@lnu.edu.ua*
https://doi.org/10.23939/jcpee2024/01/030

**Abstract:** The paper is devoted to the development of approaches to the application of parallel algorithms in modelling complex dynamic objects. An overview of the existing principles of computer modelling based on parallel computing procedures is given. It is proposed to describe complex dynamic objects in the form of macromodels. An algorithm for parallelising computations when constructing a nonlinear macromodel of a dynamic object with a separate linear part is described. An iterative algorithm for constructing a macromodel that describes heterogeneous dynamic characteristics of an object is formulated.

**Key words**: dynamic object, parallel algorithms, computer modelling, macromodel, diacoptic approach.

## 1. Introduction

A significant number of processes taking place in various spheres of modern life should be considered as complex dynamic objects. They are complex because, firstly, the structure of internal relationships that determine the essence of the process is complex, and, secondly, the process is influenced by a number of external factors, the nature of which may vary, as well as the degree of their impact. An object is dynamic because processes are not constant in time, but change according to a certain law. Quite often, this law of change is not obvious. Therefore, it is necessary to analyse the activities of the object and conduct numerous experiments to determine possible scenarios of the object's behaviour. Obviously, all this involves the construction of a specific model of the object, on which one can experiment.

Since the mid-1980s, computer modelling has become the main direction in modelling complex objects. The basis for this has been block- and equation-oriented languages of sequential modelling [1, 2], parallel computing systems based on SIMD and MIMD technologies [3, 4], languages and libraries of parallel programming [5, 6], parallel methods and algorithms, and parallel modelling technology [7].

Computer modelling involves the creation of a hardware-software complex that implements mathematical methods and algorithms for constructing and simulating the relevant models. Regarding complex dynamic objects, this complex must ensure parallel computing. In this case, it is necessary to simultaneously solve two problems: scientific and theoretical, software and engineering [8].

The first of them consists in assessing the convergence of calculations, dynamically evaluating the computational parameters of various submodels, and algorithmically providing a high degree of parallelism. The other problem involves the optimal distribution of variables between parallel procedures, ensuring data exchange between individual computational processes, and synchronising parallel computational processes. Only the joint solution of these problems, while finding appropriate compromises between them, can ensure the creation of a high-performance (in terms of convergence and time parameters of computing procedures) software and hardware complex for the analysis of complex dynamic objects.

The hardware and software complex is implemented within the concept of the parallel modelling environment (PME) for a complex dynamic object [9]. It assumes that the PME consists of three interconnected parts: hardware resources (PME-Hardware), system software (System Software) and modelling software (Modeling and Simulation Software).The modelling software in turn has the following components: a dialogue subsystem (to ensure interaction with the user), parallel programming languages (for the software implementation of parallel algorithmic procedures), a programming language compiler (for compiling and activating the programme), libraries of service functions and methods (to provide additional services and implement standard algorithms). In particular, the libraries should contain methods that implement parallel algorithms.

However, in spite of the existence of standard approaches to implementing parallel procedures, there is still a large field of activity for developing our own parallel algorithms that allow us to more accurately model certain objects. In this article, we will consider a complex dynamic object that is described using a certain functional model. To build it, we will propose a procedure based on a parallel algorithm.

## 2. Macromodel of a dynamic object

It is quite difficult to describe a complex dynamic object using a model that takes into account all the structural features of the object. Therefore, such an object is described using a functional model or macromodel. In this description, the object is presented as a "black box" with an unknown internal structure. And the functional features of the object characterise the object's reactions (output variables) caused by the action of external factors (input variables). The advantages of such models are described in particular in [10]. It is also important that they most fully reflect the dynamic properties of the object. For certain types of objects, appropriate methods for constructing macromodels have been proposed and original software has been developed [11].

In general, a macromodel of a dynamic object can be represented as follows:

$$y(t) = f(x(t), p(t), v(t)), \quad (1)$$

where $y(t)$ is a vector of output parameters, $v(t)$ is a vector of input parameters, $x(t)$ is a vector of some formal parameters, $p(t)$ is a vector of the model parameters, $f(\cdot)$ is some function defining the law of output parameters variation. In this formulation, all variables and parameters are functions of continuous time $t$.

Very often, a macromodel of a dynamic object is written in the form of discrete state equations with a separate linear part

$$\begin{cases} x_{i+1} = Fx_i + Gv_i + \Phi(x_i, v_i) \\ y_{i+1} = Cx_{i+1} + Dv_{i+1} \end{cases} \quad (2)$$

where $x_i$ is a vector of the state variables, $y_i$ is a vector of output variables, $v_i$ is a vector of input variables defined at $i$-th time moment ($i = 1,2,3,\ldots$); $F$, $G$, $C$, $D$ are matrices of the corresponding size with components describing parameters of the macromodel, $\Phi$ is some nonlinear vector-function.

To construct the linear part of macromodel (2), there is a well-known algorithm from the system theory, the Ho-Kalman algorithm [10]. This part further performs the role of an initial approximation for the construction of the entire macromodel (2). The nonlinear part is found by optimization. In the case of the bilinear form of macromodel (2), optimization procedures are quite well developed [12]. For other forms of nonlinearity for the function $\Phi(\cdot)$, it is necessary to develop our own optimisation algorithms, which, in particular, involve the use of parallel computing.

## 3. Using parallel procedures in the step-by-step construction of a mathematical macromodel

The construction of complex objects is always associated with the need to perform a large number of sequential computationally intensive calculations. This results in, firstly, high time costs, and, secondly, a sharp increase in the probability of calculation errors. Therefore, it is logical to search for possible options to replace one complex sequential computational procedure with a number of simpler, independent procedures that would allow obtaining the result within a few stages of computation.

One of the parallelisation methods is partitioning by output variables, when independent submodels are built at the first stage. Each of them produces a narrowed (at best, to one variable) vector of output variables. At the same time, all other output variables of the model are considered to be input variables at this stage. In this case, each of the submodels can be constructed completely independently, which allows these submodels to be built in parallel. At the next stage, the overall macromodel is obtained through arithmetic reformulation.

Such a procedure does not require significant computational resources. And in the case of using equations in the form of state variables, it is reduced to solving a linear system of equations and transforming mathematical expressions.

For example, we will consider the construction of a macromodel in the general form (2) [13].

Let us assume that the macro model has $k > 1$ output variables. That is, the vector of output variables has $k$ components:

$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(k)} \end{pmatrix}. \quad (3)$$

For each j-th component of the output variable vector ($j = \overline{1, k}$), a submodel is constructed assuming that all other components of the output variable vector of the general macromodel are input for this submodel, given by

$$\begin{cases} x_{i+1}^{(j)} = F^{(j)}x_i^{(j)} + D^{(j)}v_i + \sum_{\substack{z=1 \\ z\neq j}}^{k} A^{(j,z)}y_i^{(z)} + \\ +\Phi^{(j)}\left(v_i, y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(j-1)}, y_i^{(j+1)}, \ldots, y_i^{(k)}\right) \\ y_{i+1}^{(j)} = C^{(j)}x_{i+1}^{(j)} + D^{(j)}v_{i+1} + \sum_{\substack{z=1 \\ z\neq j}}^{k} B^{(j,z)}y_{i+1}^{(z)} \end{cases} \quad (4)$$

where the upper index indicates the submodel number, and the matrices $A^{(j,z)}$ i $B^{(j,z)}$ define mutual relations between parts of the vector of output variables of the general macromodel, $\Phi^{(j)}$ are corresponding nonlinear vector functions of a particular submodel.

At the stage of constructing submodels the matrices $A^{(j,z)}$ i $B^{(j,z)}$ are parts of the matrices $G^{(j)}$ i $D^{(j)}$, respectively (because elements of vectors $y^{(z)}$, where $z = \overline{1,k}$, $z \neq j$, are considered to be input variables).

Based on the second equations of the submodels of type (4), a system of equations is created:

$$\begin{cases} y_{i+1}^{(1)} = C^{(1)}x_{i+1}^{(1)} + D^{(1)}v_{i+1} + \sum_{z=2}^{k} B^{(1,z)}y_{i+1}^{(z)} \\ y_{i+1}^{(2)} = C^{(2)}x_{i+1}^{(2)} + D^{(2)}v_{i+1} + \sum_{\substack{z=1 \\ z\neq 2}}^{k} B^{(2,z)}y_{i+1}^{(z)} \\ \vdots \\ y_{i+1}^{(k)} = C^{(k)}x_{i+1}^{(j)} + D^{(k)}v_{i+1} + \sum_{z=1}^{k-1} B^{(k,z)}y_{i+1}^{(z)} \end{cases} \quad (5)$$

or:

$$y_i = \tilde{C}x_i + \tilde{D}v_i + \tilde{B}y_i, \quad (6)$$

where

$$x_i = \begin{pmatrix} x_i^{(1)} \\ x_i^{(2)} \\ \vdots \\ x_i^{(k)} \end{pmatrix}, \quad (7)$$

$$\tilde{C} = \begin{pmatrix} C^{(1)} & & & 0 \\ & C^{(2)} & & \\ & & \ddots & \\ 0 & & & C^{(k)} \end{pmatrix}, \quad (8)$$

$$\tilde{D} = \begin{pmatrix} D^{(1)} \\ D^{(2)} \\ \vdots \\ D^{(k)} \end{pmatrix}, \quad (9)$$

$$\tilde{B} = \begin{pmatrix} 0 & B^{(1,2)} & B^{(1,3)} & \cdots & B^{(1,k)} \\ B^{(2,1)} & 0 & B^{(2,3)} & \cdots & B^{(2,k)} \\ B^{(3,1)} & B^{(3,2)} & \ddots & & \vdots \\ \vdots & \vdots & & 0 & B^{(k-1,k)} \\ B^{(k,1)} & B^{(k,2)} & \cdots & B^{(k,k-1)} & 0 \end{pmatrix} ()$$

The solution to the system of equations (6) will be as follows:

$$y_i = \left(1 - \tilde{B}\right)^{-1}\left(\tilde{C}x_i + \tilde{D}v_i\right) \quad (11)$$

If we substitute (11) into the first equations of type (4) and perform elementary transformations, we obtain a general macromodel in the form of (2), which can be used to model the corresponding complex object.

Thus, the construction of a general macromodel of a complex object can be divided into two distinct sequential stages. At the first stage, submodels are constructed, and at the second stage, a general macromodel is created on their basis. If at the second stage the computational cost is not significant, then at the first stage, when applying sequential computational procedures, It is significantly higher. Therefore, the use of parallelisation through independent construction of submodels of type (4) is a rather effective approach.

## Application of parallel algorithms for the construction of a general macromodel by the diacoptic method

Often, the dynamics of a complex object is not homogeneous, when different sections of its dynamic characteristics need to be described by different ratios. This leads to the problem of creating a general macromodel that would describe all such sections. In this case, the final form of the general macromodel should be constructed in stages. At first, sub- macromodels are created, each of which corresponds to a specific section of the object's dynamic characteristics, on the basis of which a general macromodel is then created. The final version of the macromodel requires matching the parameters that are common to different submacromodels. This is achieved by applying appropriate optimisation procedures. In general, this approach is called diacoptic [10].

The iterative algorithm for creating a general macro model using the diacoptic approach is as follows (Figure).

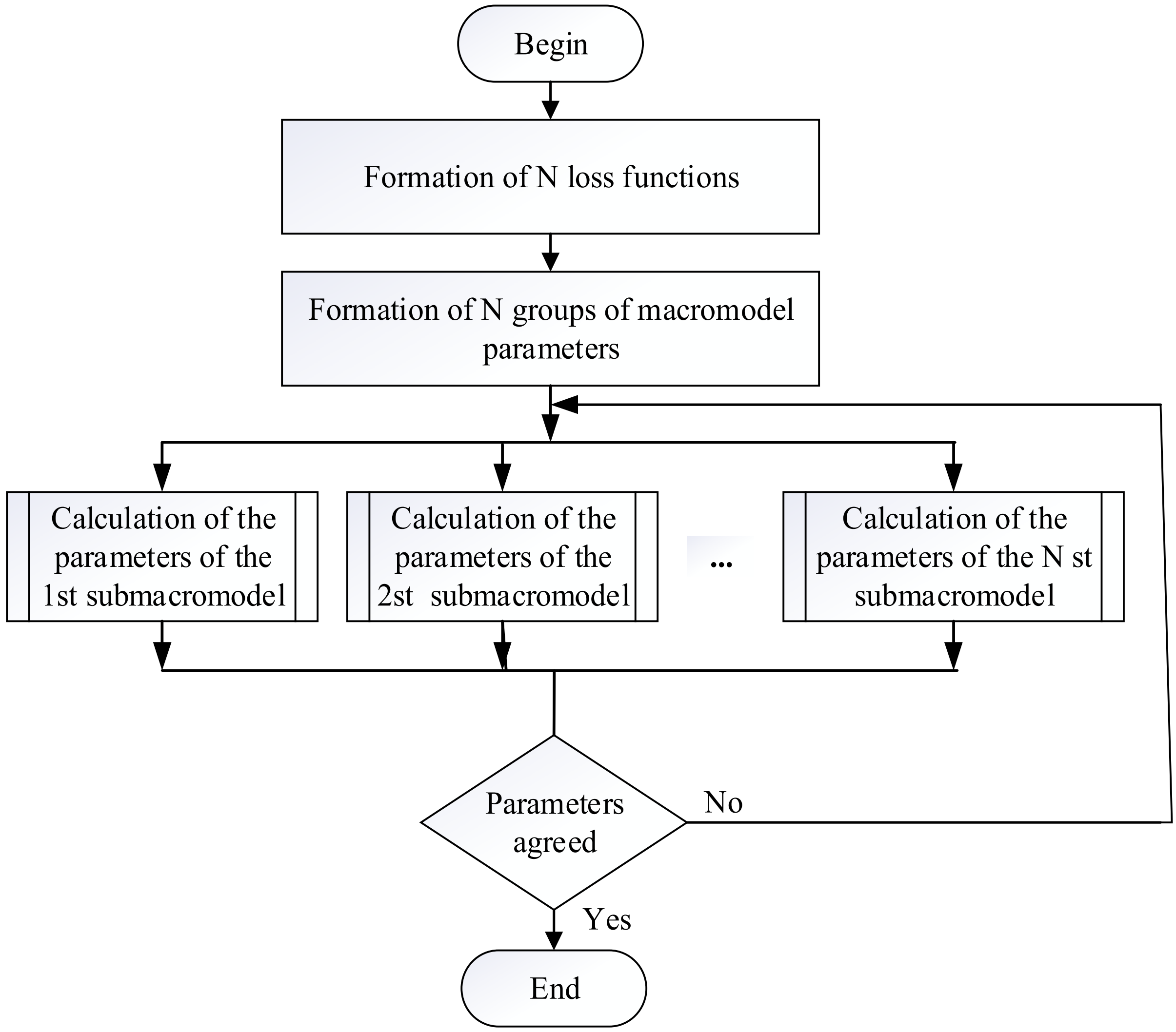
The following main steps can be highlighted in this algorithm [13].

1. For each section of the dynamic characteristic of a complex object, the objective function for the corresponding optimisation problem is constructed.

2. From the parameters of the general macromodel of the object, several groups are formed according to the number of constructed goal functions. When forming a group, priority is given to those parameters that are most relevant to the relevant objective function. The common parameters of different groups in subsequent iterations are subject to be matched.

3. Having solved the corresponding optimisation problem, the values of the parameters of the corresponding submacromodels are found. In this case, the parameters agreed upon at a certain iteration are considered constant. It is the solution of the set of optimisation problems that is subject to solution. It is important to note that in many cases, when solving such optimisation problems, it is advisable to use adaptive algorithms for finding the global minimum [14].

4. If the parameters of all submacromodels are not agreed upon, then their values are considered the next approximation for solving the corresponding optimisation problems and return to step 3. If the parameters are fully agreed upon, the process proceeds to the construction of the general macromodel of the object.




*Algorithm for creating a general macromodel.*

## 4. Conclusions

Computer modelling of complex dynamic objects requires intensive use of parallel computing procedures. In doing so, it is necessary to solve a number of problems, namely, to create an appropriate computer system architecture, as well as software based on the use of parallel algorithms.

Obviously, the development of parallel algorithms also depends on the modelling approach. If a complex model is built from a set of several simple submodels, then the procedure for building the model should be divided into several independent procedures executed in parallel.

If the model is to describe different stages of a dynamic object's functioning, then parallel algorithms should organically fit into iterative procedures aimed at matching model parameters for different sections of the object's functional characteristics.

## References

[1] H. A. Johnson, "Using Advanced Continuous Simulation Language (ACSL) to Simulate, Solve, and Fit Mathematical Models in Nutrition", in: Novotny, J. A., Green, M. H., Boston, R. C. (eds) *Mathematical Modeling in Nutrition and the Health Sciences. Advances in Experimental Medicine and Biology*, vol. 537. Springer, Boston, MA, 2003. https://doi.org/10.1007/978-1-4419-9019-8_24

[2] D. K. Chaturvedi, *Modeling and simulation of system using Matlab and Simulink*. CRS Pres Taylor & Francis Group, 2010.

[3] D. M. Harris, S.L. Harris, *Digital Design and Computer Architecture* (Second Edition), Morgan Kaufmann, 2013.

[4] S. K. Meesala, P. M. Khilar, and A. K. Shrivastava, "Multiple Instruction Multiple Data (MIMD) Implementation on Clusters of Terminals", *International*

*Journal of Science and Research (IJSR)*, Vol. 5, Issue 1, 2016. DOI: 10.13140/RG.2.1.4826.9846

[5] W. E. Nagel, D. H. Kroner, and M. M. Resch. *High Performance Computing in Science and Engineering '12.* Springer - Verlag Berlin Heidelberg, 2013.

[6] C. Hughe and, T. Hughes, *Parallel and distributed programming using C++*, NJ: Prentice Hall, 2004.

[7] G. D'Angelo and M. Marzolla, "New trends in parallel and distributed simulation: From many-cores to Cloud Computing", *Simulation Modelling Practice and Theory*, vol. 49, pp. 320–335, 2014. https://doi.org/10.1016/j.simpat.2014.06.007.

[8] P. G. Stakhiv and D. I. Krupskiy, "Modeling of complex electronic systems with adaptive parallel algorithms", *Actual problems of theoretical electrical engineering: science and didactics,* Lviv, Krym-Aluszta, pp. 131–136, 1999 (in Ukrainian).

[9] V. A. Sviatnyy, "Parallel modeling of complex dynamic systems," Modeling-2006, Kyiv, pp. 83–90, 2006 (in Ukrainian).

[10] P. G. Stakhiv, Yu. Ya. Kozak, and O. P. Hoholyuk, *Discrete modeling in electrical engineering and related fields*. Lviv: Lviv Polytechnik, 2014 (in Ukrainian).

[11] B. Mel'nyk and S. Trokhahiak, "Constrction of Shematitic Models of Multipoles for CAD System," *CPEE'02,* Zakopane, pp. 63–66, 2002.

[12] P. Stakhiv, B. Melnik, and V. Dzhala, "Design of Discrete Makromodels for Nonlinear Dynamic Object with Multiple Input," *Engineering Simulation*, Amsterdam. vol. 14, pp. 493–498, 1997.

[13] P. Stachiw, Yu. Kozak, and B. Melnyk, "Application of parallel algorithms in the synthesis of dynamic multipole macromodels", *IC-SPETO*. Gliwice-Niedica. Vol. II, pp. 263–266, 2003.

[14] P. Stakhiv, O. Hoholyuk, O. Hamola, B. Melnyk, V. Maday and N. Melnyk, "Application of the Rastrygin's Method in Modeling of Complex Electrical Systems", *2023 24th International Conference on Computational Problems of Electrical Engineering (CPEE)*, Grybów, Poland, 2023, pp. 1–5. DOI: 10.1109/CPEE59623.2023.10285315.

## ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ ДЛЯ МОДЕЛЮВАННЯ СКЛАДНИХ ДИНАМІЧНИХ ОБ'ЄКТІВ

Петро Стахів, Богдан Мельник, Оксана Гоголюк, Степан Троханяк

У статті розроблено підходи до застосування паралельних алгоритмів під час моделювання складних динамічних об'єктів. Виконано огляд засад комп'ютерного моделювання, які спираються на використання процедур паралельних обчислень. Запропоновано описувати складні динамічні об'єкти у вигляді макромоделей. Описано алгоритм розпаралелення обчислень під час побудови нелінійної макромоделі динамічного об'єкта з виділеною лінійною частиною. Сформульовано ітераційний алгоритм побудови макромоделі, яка описує неоднорідну динамічну характеристику об'єкта.

**Petro Stakhiv** – Ph.D., D.Sc., Professor, graduated from Lviv State University, Department of Physics, Ukraine in 1970, and received his M.Sc. degree in Radio Physics and Electronics. In 1975 he received his Ph.D. degree in theoretical electrical engineering. The theme of his Ph.D. work was "Synthesis of linear electric circuits (method of state variables)".

In 1992 he received his D. Sc. degree in the same specialty after defending his doctor thesis on «Analysis of dynamic regimes in electric and electronic circuits with multiterminal elements». From 1973 to 1996 he worked as Assistant Professor, as Asociate Professor, Professor and Head of Department of Theoretical Electro and Radio Engineering at Lviv State University. In 1996 became a Head of Department of Theoretical and General Electrical Engineering. His scientific interests are mainly concerned with mathematical modelling and simulation of dynamic processes in electrical engineering systems, numerical methods, optimization techniques, system theory, and parallel programming.

**Bohdan Melnyk** – is a PhD in Engineering, Associate Professor of the Department of Information Systems in Management at Ivan Franko National University of Lviv. He graduated from the Faculty of Physics of Ivan Franko National University of Lviv with a degree in Radiophysics and Electronics.

PhD thesis: Identification of parameters of discrete macromodels of analogue components of electronic and electrical circuits. Author of more than 150 scientific papers. Doctoral student of the Western Ukrainian National University.

Research interests: mathematical modelling, information technology, computer systems.

**Oksana Hoholyuk** – D.Sc, Ass. Professor, graduated from Department of Radiohysics and Electronics of Ivan Franko State University of Lviv, Ukraine. She received her Ph.D. and D.Sc degrees in the field of theoretical electrical engineering. Since 2004 she has been working at the Department of Theoretical.

Electrical Engineering at Lviv Polytechnic National University, Ukraine

Her scientific interests are focused on mathematical modelling of electric circuits and systems.

**Stepan Trokhaniak** – PhD, Associate Professor of the Department of Information Systems in Management at Ivan Franko National University of Lviv. He graduated from the Faculty of Computer Science and Information Technology of Lviv Polytech nic National University with a degree in Computer Systems of Information Processing and Management. Dissertation on the topic: Development of mathematical models of multipole circuits adapted to the input language of modern electronic circuit analysis programmes. Author of more than 50 scientific papers.

Research interests: mathematical modelling, computer technology

*Received: 16.05. 2024 Accepted: 16.05. 2024*

ORCID ID: 0000-0002-5263-0747 (P. Stakhiv)
ORCID ID: 0000-0001-6399-6317 (B. Melnyk)
ORCID ID: 0000-0003-2146-4667 (O. Hoholyuk)
ORCID ID: 0000-0002-3163-9267 (S. Troknanyak)