ISSN 2786-7102 (Online), ISSN 2786-7633 (Print)
https://doi.org/10.15407/srenergy2024.02.058

UDC 517.9: 519.6

Vladyslav Khaidurov[1,2]*, PhD (Engin.), Senior Researcher, https://orcid.org/0000-0002-4805-8880
Vadym Tatenko[1], https://orcid.org/0009-0008-4869-9689
Mykyta Lytovchenko[1], https://orcid.org/0009-0001-6671-7763
Tamara Tsiupii[3], PhD (Engin.), Associate Professor, https://orcid.org/0000-0003-2206-2897
Tetiana Zhovnovach[4], https://orcid.org/0000-0003-1037-4383
[1]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Beresteiskyi Prosp., Kyiv, 03056, Ukraine;
[2]General Energy Institute of NAS of Ukraine, 172, Antonovycha St., Kyiv, 03150, Ukraine;
[3]National University of Life and Environmental Sciences of Ukraine, 15, Heroiv Oborony St., Kyiv, 03041, Ukraine;
[4]Cherkasy branch of European University, 83, Smilyanska St., Cherkasy, 18008, Ukraine
*Corresponding author: allif0111@gmail.com

ACCELERATION OF COMPUTATIONS IN MODELLING OF PROCESSES IN COMPLEX OBJECTS AND SYSTEMS

Abstract. *The development of methods of parallelization of computing processes, which involve the decomposition of the computational domain, is an urgent task in the modeling of complex objects and systems. Complex objects and systems can contain a large number of elements and interactions. Decomposition allows you to break down a system into simpler subsystems, which simplifies the analysis and management of complexity. By dividing the calculation area of the part, it is possible to perform parallel calculations, which increases the efficiency of calculations and reduces simulation time. Domain decomposition makes it easy to scale the model to work with larger or more detailed systems. With the right choice of decomposition methods, the accuracy of the simulation can be improved, since different parts of the system may have different levels of detail and require appropriate methods of additional analysis. Decomposition allows the simulation to be distributed between different participants or devices, which is relevant for distributed systems or collaborative work on a project. In this work, mathematical models are built, which consist in the construction of iterative procedures for "stitching" several areas into a single whole. The models provide for different complexity of calculation domains, which makes it possible to perform different decomposition approaches, in particular, both overlapping and non-overlapping domain decomposition. The obtained mathematical models of subject domain decomposition can be applied to objects and systems that have different geometric complexity. Domain decomposition models that do not use overlap contain different iterative methods of "stitching" on a common boundary depending on the types of boundary conditions (a condition of the first kind is a Dirichlet condition, or a condition of the second year is a Neumann condition), and domain decomposition models with an overlap of two or more areas consist of the minimization problem for constructing the iterative condition of "stitching" areas. It should be noted that the obtained models will work effectively on all applied tasks that describe the dynamic behavior of objects and their systems, but the high degree of efficiency of one model may be lower than the corresponding the degree of effectiveness of another model, since each task is individual.*
Keywords: mathematical modelling, decomposition of the computational domain, parallelization, optimization, complex objects and systems.

1. Introduction

Parallelization of calculations and decomposition of the computational domain are closely related concepts used to efficiently perform computing tasks on multiprocessor systems [1–3]. Parallelization of calculations is a method in which calculations are broken down into smaller subtasks that can be performed independently of each other. These subtasks are then distributed among the available computing resources (processors, cores, nodes) for acceleration general performance [2, 4, 5]. Parallelization can only be done at the level of data, tasks, or instructions. Examples of parallelization include parallel loops, where iterations of the loop are performed independently, and distributed calculations, where tasks are performed on different nodes in the network [4, 6, 7]. Decomposition of the computational domain means dividing a large

computational problem into smaller subproblems that can be solved independently. This is especially useful in simulation and numerical equation solving problems, where the domain is broken down into smaller subdomains. Each part of the area is then processed separately. Examples of decomposition include the Finite Element Method (FEM), the Finite Difference Method (FDM), and other numerical methods that break down physical space into grids or grids for accurate modeling [8, 9].

The combination of such concepts allows you to quickly and efficiently use parallel resources to solve complex computing problems [10–12]. Each small task is performed in parallel, which allows you to increase the speed of the entire process.

2. Methods and materials

The method of domain decomposition is a method that is based on the division of the entire study area into smaller subareas, that is, the solution of the general problem is reduced to the solution of smaller problems (subproblems) that are interrelated. The peculiarity of the method is that each of the subproblems, obviously, makes it possible to reduce the solution of the problem to the solution of subproblems that have a lower algebraic dimension and are interconnected by some conditions on the lines of the sections of the domain. So, an iterative process is built, on one iteration, which needs to be solved in a subdomain.

Area decomposition methods are divided into methods with intersecting subareas and methods with non-overlapping subareas. The motivation for the use of domain decomposition can be the complex geometry of the original domain, which can be simplified with its help, the use of various mathematical models and approximations in the subdomain, the possibility of using direct methods in the subdomains.

2.1. Mathematical Models of Decomposition of the Subject Area of Objects Without Overlap

Recently, the method of decomposition of the region has gained great popularity in connection with the development of computing systems with parallel architecture. When the method is implemented on multiprocessor computers, iterations are organized in such a way that the solution of problems in the sub-domain is carried out in parallel, due to which a gain in computational time is achieved.

To date, domain decomposition methods for second-order elliptic equations have been most developed.

Let Ω be the domain on which the numerical solution of the Poisson equation is found. The task can be written as follows:

$$\Delta u = -f \text{ in } \Omega. \tag{1}$$

Suppose for (1) that Ω is divided into two subregions Ω_1 and Ω_2, which do not intersect with each other. The boundaries of each of the regions are G_1 and G_2, respectively. D is the boundary between the two regions of each of the regions Ω_1 and Ω_2. Then problem (1) is equivalent to two subproblems with "stitching" conditions on Γ:

$$\Delta u_1 = -f \text{ in } \Omega_1,\ \Delta u_2 = -f \text{ in } \Omega_2, \tag{2}$$

$$u_1 = u_2, \frac{\partial u_1}{\partial n} = \frac{\partial u_2}{\partial n} \text{ on } \Gamma. \tag{3}$$

where n is the normal to G. Let's give an example of a Ω-area (Fig. 1). Subproblems (2) and (3) are solved iteratively, and the unification of their solutions in subdomains Ω_1 and Ω_2 must coincide with the solution of the entire problem (1) in the entire study area Ω.

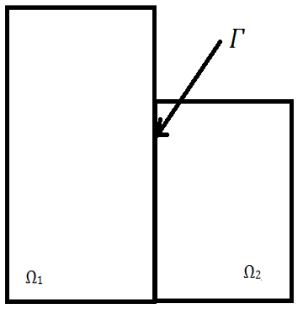

Figure 1. Example of Areas Without Overlap

2.1.1. Iterative method according to the Neumann-Dirichlet principle

The Neumann-Dirichlet iteration method is a method that belongs to the iterative methods of domain decomposition. This method has such a name because, sequentially solving the problem, we first solve the boundary value problem with the Dirichlet condition, and then with the Neumann condition. A problem of the first kind or a Dirichlet problem is called a problem of the first kind if the value of the function is given at the ends. For example: $y(a)=\alpha.$ A problem is called a problem of the second kind or a Neumann problem if the value of the derivative function is given at the ends. For example: $y'(a)=\gamma$.

To find the solution to problem (1), iteratively solving subproblems (2) and (3), we set the real value and $\theta>0$ build an iterative process with an initial approximation λ^0. Let the λ^k unknown then find an approximation u_1^{k+1} u_2^{k+1} to the solution in the subdomains by solving the following equations sequentially. So, the mathematical model of the problem looks like this:

$$\begin{cases} \begin{cases} -\Delta u_1^{k+1} = f \text{ в } \Omega_1, \\ u_1^{k+1}\big|_{\Gamma_1} = \lambda^k; \end{cases} \\ \begin{cases} -\Delta u_2^{k+1} = f \text{ в } \Omega_2, \\ \left.\dfrac{\partial u_2^{k+1}}{\partial n}\right|_{\Gamma_2} = \dfrac{\partial u_1^k}{\partial n} \end{cases} \\ \lambda^{k+1} = \lambda^k + \theta\left(u_2^{k+1}\big|_{\Gamma_2} - \lambda^k\right). \end{cases} \tag{4}$$

The last condition in (4) is the condition for the iterative "stitching" of the two regions. As we can see, in the first part of step (4) the boundary value problem with the Dirichlet condition on G_1 is solved, in the second substep in (4) the boundary value problem with the Neumann condition on G_2 is solved, so the method is called the Neumann–Dirichlet iterations. In the third substep in (4), the recalculation of the iterative approximation is reproduced. Iterations λ^{k+1} are carried out until the Condition begins to be fulfilled for neighboring values $\lambda.$, which can be written as the following relation:

$$\left|\lambda^{k+1} - \lambda^k\right| < \varepsilon, \tag{5}$$

where ε is the accuracy of the calculation for "stitching" two areas. The Neumann-Dirichlet method can also be used for a fairly large number of areas. Condition (5) is a classical condition for stopping such iterative methods of domain decomposition, but also for most optimization methods and algorithms. Now let's give a more detailed algorithm for the numerical solution of the problem by this method.

Neumann-Dirichlet Method Algorithm

1. Set a small value of $\varepsilon>0$.
2. We solve the Poisson equation in the first subdomain:

$$\begin{cases} -\Delta u_1^{k+1} = f \text{ in } \Omega_1, \\ u_1^{k+1}\big|_{\Gamma_1} = \lambda^k. \end{cases} \tag{6}$$

3. We solve the Poisson equation in the second subdomain, taking into account the value of the derivative taken from the first step at the boundary of the first domain Γ_1:

$$\begin{cases} -\Delta u_2^{k+1} = f \text{ in } \Omega_2, \\ \left.\dfrac{\partial u_2^{k+1}}{\partial n}\right|_{\Gamma_2} = \dfrac{\partial u_1^k}{\partial n}. \end{cases} \tag{7}$$

4. We make a recalculation to "stitch" the regions:

$$\lambda^{k+1} = \lambda^k + \theta\left(u_2^{k+1}\Big|_{\Gamma_2} - \lambda^k\right). \tag{8}$$

5. Check the execution of the inequality:

$$\left|\lambda^{k+1} - \lambda^k\right| < \varepsilon. \tag{9}$$

In the case of inequality, proceed to step 6, otherwise to step 2.

6. End of algorithm. Visualization of the results obtained.

The above algorithm coincides with the desired solution of the whole problem (1) in the Ω domain. Steps 2 and 3 of this algorithm can obviously be extended to a larger number of subdomains. This means that there will be more subtasks of the form (6) and (7). The stitching condition for each pair will be given in the same way, i.e. in the form (8). The stop criterion (9) of solving problem (1) with a large number of subdomains will be given as the maximum deviation at the "stitching" boundary.

2.1.2. Iterative method according to the Neumann-Neumann principle

Similarly, you can derive an algorithm that solves the same problem using other boundary conditions and the "stitching" condition. In this case, the values of the derivatives are given at the boundaries of areas that do not intersect with each other. This method is called the Neumann-Neumann method because the derivative values (Neumann condition) are given at the edges of the regions that contain a common boundary. In the case of the two subdomains, the Neumann-Neumann iteration method is an iterative solution on some domain. In the simplest case, the area to be split is a rectangle.

Now let's write down the mathematical model of the problem. It looks like this:

$$\begin{cases} \begin{cases} -\Delta u_1^{k+1} = f \text{ in } \Omega_1, \\ \left.\dfrac{\partial u_1^{k+1}}{\partial n}\right|_{\Gamma_1} = \lambda^k; \end{cases} \\ \begin{cases} -\Delta u_2^{k+1} = f \text{ in } \Omega_2, \\ \left.\dfrac{\partial u_2^{k+1}}{\partial n}\right|_{\Gamma_2} = \lambda^k; \end{cases} \\ \lambda^{k+1} = \lambda^k + \dfrac{\theta}{2}\left(u_2^{k+1}\Big|_{\Gamma_2} - u_1^{k+1}\Big|_{\Gamma_1}\right), \end{cases} \tag{10}$$

where the iterative condition of "stitching" areas is written as:

$$\lambda^{k+1} = \lambda^k + \frac{1}{2}\theta\left(u_2^{k+1}\Big|_{\Gamma_2} - u_1^{k+1}\Big|_{\Gamma_1}\right). \tag{11}$$

In this case, the calculations (11) can be carried out until the deviations between adjacent values λ^k and λ^{k+1} become less than the predetermined accuracy of the calculations:

$$\left|\lambda^{k+1} - \lambda^k\right| < \varepsilon, \tag{12}$$

where ε is the accuracy of the calculation for "stitching" two areas.

Obviously, the Neumann-Nyman method (10)–(12) can also be applied to more areas. As already mentioned, the Neumann-Neumann iteration method is easily generalized to the case of partitioning into a large number of subareas. $\Omega_i, i = \overline{1,m}$. Now let's give a more detailed algorithm for the numerical solution of the problem by this method.

Neumann-Neumann method algorithm

1. Set a small value of $\varepsilon > 0$.
2. We solve the Poisson equation in the first subdomain:

$$\begin{cases} \Delta u_1^{k+1} = -f \text{ in } \Omega_1, \\ \left.\dfrac{\partial u_1^{k+1}}{\partial n}\right|_{\Gamma_1} = \lambda^k. \end{cases} \tag{13}$$

3. We solve the Poisson equation in the second subdomain:

$$\begin{cases} \Delta u_2^{k+1} = -f \text{ in } \Omega_2, \\ \left.\dfrac{\partial u_2^{k+1}}{\partial n}\right|_{\Gamma_2} = \lambda^k. \end{cases} \tag{14}$$

4. We make a recalculation to "stitch" the regions:

$$\lambda^{k+1} = \lambda^k + \frac{1}{2}\theta\left(u_2^{k+1}\Big|_{\Gamma_2} - u_1^{k+1}\Big|_{\Gamma_1}\right). \tag{15}$$

5. Check the execution of the inequality:

$$\left|\lambda^{k+1} - \lambda^k\right| < \varepsilon. \tag{16}$$

In the case of inequality, proceed to step 6, otherwise to step 2.

6. End of algorithm. Visualization of the results obtained.

Convergence (13)–(16) is in practice faster than (10)–(12).

2.1.3. Iterative method according to the Dirichlet-Dirichlet principle

Similarly, the Dirichlet-Dirichlet method is used. Let's go back to the division of Ω into two subregions: Ω_1 and Ω2. For the numerical solution of the problem, two subproblems are formed with a certain condition for "stitching" these areas. This method got this name because boundary conditions of the first kind (Dirichlet conditions) are given at the boundary of the "stitching" of regions.

Let's write down a mathematical model of the problem.

$$\begin{cases} \begin{cases} -\Delta u_1^{k+1} = f \text{ in } \Omega_1, \\ u_1^{k+1}\Big|_{\Gamma_1} = \lambda^k; \end{cases} \\ \begin{cases} -\Delta u_2^{k+1} = f \text{ in } \Omega_2, \\ u_2^{k+1}\Big|_{\Gamma_2} = \lambda^k; \end{cases} \\ \lambda^{k+1} = \lambda^k + \dfrac{\theta}{2}\left(\left.\dfrac{\partial u_2^{k+1}}{\partial n}\right|_{\Gamma_2} - \left.\dfrac{\partial u_1^{k+1}}{\partial n}\right|_{\Gamma_1}\right). \end{cases} \tag{17}$$

The "stitching" condition will look like this:

$$\lambda^{k+1} = \lambda^k + \frac{\theta}{2}\left(\left.\frac{\partial u_2^{k+1}}{\partial n}\right|_{\Gamma_2} - \left.\frac{\partial u_1^{k+1}}{\partial n}\right|_{\Gamma_1}\right). \tag{18}$$

For the Dirichlet-Dirichlet algorithm (17)–(18), the corresponding description is given below.

Algorithm of the Dirichlet-Dirichlet method

1. Set a small value of $\varepsilon > 0$.
2. We solve the Poisson equation in the first subdomain:

$$\begin{cases} \Delta u_1^{k+1} = -f \text{ in } \Omega_1, \\ u_1^{k+1}\Big|_{\Gamma_1} = \lambda^k. \end{cases} \tag{19}$$

3. We solve the Poisson equation in the second subdomain:

$$\begin{cases} \Delta u_2^{k+1} = -f \text{ in } \Omega_2, \\ u_2^{k+1}\big|_{\Gamma_2} = \lambda^k. \end{cases} \tag{20}$$

4. We make a recalculation to "stitch" the regions:

$$\lambda^{k+1} = \lambda^k + \frac{\theta}{2}\left(\frac{\partial u_2^{k+1}}{\partial n}\bigg|_{\Gamma_2} - \frac{\partial u_1^{k+1}}{\partial n}\bigg|_{\Gamma_1} \right). \tag{21}$$

5. Check the execution of the inequality:

$$\left|\lambda^{k+1} - \lambda^k\right| < \varepsilon. \tag{22}$$

In the case of inequality, proceed to step 6, otherwise to step 2.

6. End of algorithm. Visualization of the results obtained.

It is obvious that when solving problems with more subareas than two, it is possible to apply combined methods of "stitching" these areas in the ratios (19)–(22).

2.2. Mathematical Models of Decomposition of the Subject Area of Objects with Overlap

Partially overlapping area decomposition methods are slightly different from area decomposition methods that do not overlap. Their peculiarity is that the controlling parameter in the problems discussed in the previous section was one vector. In this problem, the control parameter will be 2 vectors (one per region) that will "stitch" two subdomains into one. These vectors will define the boundaries of each of the regions, relative to which the "stitching" of the two subregions will take place. As a result of decomposition, it is obvious that the value of the deviation integral for these areas should be minimal. Since the given regions overlap, the deviation integral for the common part of the two regions must acquire a minimum value. In the methods of decomposition of areas with overlap, as well as in the methods of decomposition of areas without overlap, the peculiarity of the method is that each of the subproblems obviously allows you to reduce the solution of the original problem to the solution of subproblems that have a lower algebraic dimension. Let's describe these methods in more detail.

Let Ω be the domain on which the numerical solution of the Poisson equation is found. Suppose that it is divided into two subregions Ω_1 and Ω_2, which partially overlap (Fig. 2). *G* is the common region of the two sub-regions Ω_1 and Ω_2. The boundaries of each of the G_1 and G_2 regions, respectively. *G* is the boundary between the two regions of each of the regions Ω_1 and Ω_2.

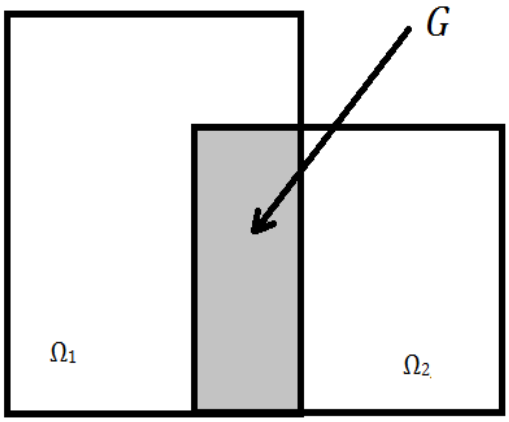

Figure 2. Example of Overlapping Areas

Newton's Method as a Basis for the Iterative Process of Decomposition of a Region with an Intersection. Functionality minimization can be carried out using any gradient method. The paper proposes Newton's method.

Newton's classical method and the conditions for its convergence.

The sequence of points, $x^{(0)}, x^{(1)}, \ldots, x^{(k)}, \ldots$, which is generated by the Newtonian method, is built based on the following considerations.

Let the function $f(x)$ be convex and twice differentiable by R^n, and the matrix is $f''(x)$ nondegenerate on R^n. Then for the point $x^{(k)}$ there is a representation:

$$f(x) - f\left(x^{(k)}\right) = f'\left(x^{(k)}\right)\left(x - x^{(k)}\right) + \frac{1}{2} f''\left(x^{(k)}\right)\left(x - x^{(k)}\right)^2 + o\left(\left(x - x^{(k)}\right)^2\right). \quad (23)$$

To determine the next point $x^{(k+1)}$ of the iterative process of Newton's method, the function $f_k(x)$, which is the quadratic part of the increment, is $f(x) - f\left(x^{(k)}\right)$ minimized in (23):

$$f_k(x) = f'\left(x^{(k)}\right), x - x^{(k)} + \frac{1}{2} f''\left(x^{(k)}\right)\left(x - x^{(k)}\right), x - x^{(k)}. \quad (24)$$

Let us show that function (24) is convex. It is easy to verify that the matrix of the second derivatives of the function coincides with the corresponding $f_k(x)$ matrix of the function at the point, $f(x)$ i.e. Since the condition is a convex function, then according to the convexity criterion the matrix is $x^{(k)}$ inherently defined. Therefore, according to this criterion, the function $f_k''(x) = f''\left(x^{(k)}\right)$ is also convex $f(x)$. $f''(x)$ $f_k(x)$

Let us now consider the problem of minimizing the convex function $f_k(x)$ on R^n. As you know, such a problem has a single minimum point, and the necessary and sufficient condition of optimality for it is as follows:

$$f_k'(x) = f'\left(x^{(k)}\right) + f''\left(x^{(k)}\right)\left(x - x^{(k)}\right) = O_n. \quad (25)$$

Having solved the system of linear equations (25) in matrix form and putting the found minimum point for $x^{(k+1)}$, we have:

$$x^{(k+1)} = x^{(k)} - \left(f''\left(x^{(k)}\right)\right)^{-1} f'\left(x^{(k)}\right). \quad (26)$$

The relation (26) and defines the iterative process of Newton's method in its classical form. If the elements of the matrix $\left(f''\left(x^{(k)}\right)\right)^{-1}$ are denoted by $\varphi_{ij}\left(x^{(k)}\right)$, $i, j = \overline{1,n}$, , then this method can be written in coordinate form:

$$I\left(\lambda_{u_1}^k, \lambda_{u_2}^k\right) = \iint_S \left(u_1\left(\lambda_{u_1}^k\right) - u_2\left(\lambda_{u_2}^k\right)\right)^2 dxdy. \quad (27)$$

$$x_i^{(k+1)} = x_i^{(k)} - \sum_{j=1}^{n} \varphi_{ij}\left(x^{(k)}\right) \frac{\partial f\left(x^{(k)}\right)}{\partial x_j}, j = \overline{1,n}. \quad (28)$$

Obviously, since function (27) is quadratic and convex over the entire domain, it has a single minimum, which is global, which can be iteratively found using (28).

Thus, the problem is reduced to the following task:

$$I_1(x,y)=\iint_G\left(u_1(x)-u_2(y)\right)^2 ds, \tag{29}$$

$$\frac{\partial I_1}{\partial x}=2\iint_G\left(u_1(x)-u_2(y)\right)u_1'(x)ds,\ \frac{\partial I_1}{\partial y}=-2\iint_G\left(u_1(x)-u_2(y)\right)u_2'(y)ds, \tag{30}$$

$$\frac{\partial^2 I_1}{\partial x^2}=2\iint_G\left(u_1''(x)-u_2(y)\left(u_1'(x)\right)^2\right)ds,\frac{\partial^2 I_1}{\partial x\partial y}=-2\iint_G u_1'(x)u_2'(y)ds, \tag{31}$$

$$\frac{\partial^2 I_1}{\partial y^2}=-2\iint_G\left(u_1(x)\left(u_2'(y)\right)^2-u_2''(y)\right)ds. \tag{32}$$

Then from (29)–(32) we can write the following:

$$\begin{pmatrix}x_{k+1}\\ y_{k+1}\end{pmatrix}=\begin{pmatrix}x_k\\ y_k\end{pmatrix}-\begin{pmatrix}\left.\frac{\partial^2 I_1}{\partial x^2}\right|_{(x_k,y_k)} & \left.\frac{\partial^2 I_1}{\partial x\partial y}\right|_{(x_k,y_k)}\\ \left.\frac{\partial^2 I_1}{\partial x\partial y}\right|_{(x_k,y_k)} & \left.\frac{\partial^2 I_1}{\partial y^2}\right|_{(x_k,y_k)}\end{pmatrix}^{-1}\begin{pmatrix}\left.\frac{\partial I_1}{\partial x}\right|_{(x_k,y_k)}\\ \left.\frac{\partial I_1}{\partial y}\right|_{(x_k,y_k)}\end{pmatrix}. \tag{33}$$

The relation (33) makes it possible to use Newton's classical method for an iterative search for parameters that minimize a convex function or functional, depending on the problem at hand.

2.3. Description of mathematical models of area decomposition

2.3.1. Dirichlet-Dirichlet iteration method

For a more accurate understanding, we present an algorithm for solving the problem with overlap for the two-dimensional case. To solve equation (1), we construct an iterative process with initial $\lambda_{u_1}^0$ approximations and. Let $\lambda_{u_2}^0$ and values on the boundaries of the areas to be stitched. They correspond to approximations $\lambda_{u_1}^k$ $\lambda_{u_1}^k$ and u_1^{k+1} to solutions in subdomains. The mathematical model of the problem is as follows: u_2^{k+1}

$$\begin{cases}\begin{cases}-\Delta u_1^{k+1}=f \text{ in } \Omega_1,\\ u_1^{k+1}\big|_{\Gamma_1}=\lambda_{u_1}^k;\end{cases}\\ \begin{cases}-\Delta u_2^{k+1}=f \text{ in } \Omega_2,\\ u_2^{k+1}\big|_{\Gamma_2}=\lambda_{u_2}^k;\end{cases}\\ \int_G\left(u_1^{k+1}-u_2^{k+1}\right)^2 ds\to min.\end{cases} \tag{34}$$

Condition (34) is a condition for the iterative "stitching" of two domains. As we can see, in the first part of the step, the boundary value problem with the Dirichlet condition on G1 is solved, in the second substep, the *boundary value problem with the Dirichlet condition on* G2 is solved. In the third substep, the iterative approximation is recalculated. λ^{k+1} Iterations are carried out until it begins to be fulfilled for neighboring λ. values A condition that can be written as the following relations:

$$\left|\lambda_{u_1}^{k+1}-\lambda_{u_1}^k\right|<\varepsilon,\left|\lambda_{u_2}^{k+1}-\lambda_{u_2}^k\right|<\varepsilon, \tag{35}$$

where ε is the accuracy of the calculation for "stitching" two areas.

Condition (35) must be true for both functions that share a boundary. Now let's give a more detailed algorithm for the numerical solution of the problem by this method.

Algorithm of the Dirichlet-Dirichlet method

1. Set a small value of $\varepsilon > 0$.
2. We solve the Poisson equation in the first subdomain:

$$\begin{cases} -\Delta u_1^{k+1} = f \text{ in } \Omega_1, \\ u_1^{k+1}\Big|_{\Gamma_1} = \lambda_{u_1}^k. \end{cases} \tag{36}$$

3. We solve the Poisson equation in the second subdomain, taking into account the value of the derivative taken from the first step on the first region *G1*:

$$\begin{cases} -\Delta u_2^{k+1} = f \text{ in } \Omega_2, \\ u_2^{k+1}\Big|_{\Gamma_2} = \lambda_{u_2}^k. \end{cases} \tag{37}$$

4. We make a recalculation to "stitch" the regions using one iteration of Newton's method for formula (3.3) from section 3.1.
5. We check the execution of irregularities:

$$\left|\lambda_{u_1}^{k+1} - \lambda_{u_1}^k\right| < \varepsilon, \left|\lambda_{u_2}^{k+1} - \lambda_{u_2}^k\right| < \varepsilon. \tag{38}$$

In case of irregularities, proceed to step 6, otherwise to step 2.

6. End of algorithm. Visualization of the results obtained.

Similarly, an algorithm is built for stitching multiple regions based on (36)–(38).

2.3.2. Neumann-Neumann iteration method

Similarly, you can derive an algorithm that solves the same problem. In the case of the two subdomains, the Neumann-Neumann iteration method is an iterative solution on some domain.

Now let's write down the mathematical model of the problem. It looks like this:

$$\begin{cases} \begin{cases} \Delta u_1^{k+1} = -f \text{ in } \Omega_1, \\ \dfrac{\partial u_1^{k+1}}{\partial n}\Big|_{\Gamma_1} = \lambda_{u_1}^k; \end{cases} \\ \begin{cases} \Delta u_2^{k+1} = -f \text{ in } \Omega_2, \\ \dfrac{\partial u_2^{k+1}}{\partial n}\Big|_{\Gamma_2} = \lambda_{u_2}^k; \end{cases} \\ \int\limits_G \left(u_1^{k+1} - u_2^{k+1}\right)^2 ds \to min. \end{cases} \tag{39}$$

In this case, the calculations can be performed until the deviations between adjacent values λ^k λ^{k+1} are less than the predetermined accuracy of the calculations. This can be written as follows:

$$\left|\lambda_{u_1}^{k+1} - \lambda_{u_1}^k\right| < \varepsilon, \left|\lambda_{u_2}^{k+1} - \lambda_{u_2}^k\right| < \varepsilon, \tag{40}$$

where ε is the accuracy of the calculation for "stitching" two areas.

Now let us give a more detailed algorithm for the numerical solution of the problem (39)–(40) by this method.

Neumann-Neumann method algorithm

1. Set a small value of $\varepsilon > 0$.
2. We solve the Poisson equation in the first subdomain:

$$\begin{cases} \Delta u_1^{k+1} = -f \text{ in } \Omega_1, \\ \left.\dfrac{\partial u_1^{k+1}}{\partial n}\right|_{\Gamma_1} = \lambda_{u_1}^k; \end{cases} \tag{41}$$

3. We solve the Poisson equation in the second subdomain:

$$\begin{cases} \Delta u_2^{k+1} = -f \text{ in } \Omega_2, \\ \left.\dfrac{\partial u_2^{k+1}}{\partial n}\right|_{\Gamma_2} = \lambda_{u_2}^k; \end{cases} \tag{42}$$

4. We perform a recalculation to "stitch" the regions using one step of the Newtonian method for the function to find $\lambda_{u_1}^{k+1}$ and $\lambda_{u_2}^{k+1}$:

$$\int_G \left(u_1^{k+1} - u_2^{k+1}\right)^2 ds \to min. \tag{43}$$

5. We check the execution of irregularities:

$$\left|\lambda_{u_1}^{k+1} - \lambda_{u_1}^k\right| < \varepsilon, \left|\lambda_{u_2}^{k+1} - \lambda_{u_2}^k\right| < \varepsilon. \tag{44}$$

In case of irregularities, proceed to step 6, otherwise to step 2.

6. End of algorithm. Visualization of the results obtained.

In algorithm (42)–(44), to minimize (43), it is possible to use not only Newton's classical method and its modifications, but also to apply the methods and algorithms of swarm intelligence in general. This kind of method does not impose a constraint on the function whose extremum is being found.

2.3.3. Neumann-Dirichlet iteration method

Using the notation and approach of the previous methods in this section, it is possible to write the Neumann-Dirichlet iteration method.

Let's write down a mathematical model of the problem.

$$\begin{cases} \begin{cases} \Delta u_1^{k+1} = -f \text{ in } \Omega_1, \\ u_1^{k+1}\big|_{\Gamma_1} = \lambda_{u_1}^k; \end{cases} \\ \begin{cases} \Delta u_2^{k+1} = -f \text{ in } \Omega_2, \\ \left.\dfrac{\partial u_2^{k+1}}{\partial n_2}\right|_{\Gamma_2} = \lambda_{u_2}^k; \end{cases} \\ \int_G \left(u_1^{k+1} - u_2^{k+1}\right)^2 ds \to min. \end{cases} \tag{45}$$

Neumann-Dirichlet Method Algorithm

1. Set a small value of $\varepsilon > 0$.
2. We solve the Poisson equation in the first subdomain:

$$\begin{cases} \Delta u_1^{k+1} = -f \text{ in } \Omega_1, \\ u_1^{k+1}\big|_{\Gamma_1} = \lambda_{u_1}^k; \end{cases} \tag{46}$$

3. We solve the Poisson equation in the second subdomain:

$$\begin{cases} \Delta u_2^{k+1} = -f \text{ in } \Omega_2, \\ \left.\dfrac{\partial u_2^{k+1}}{\partial n_2}\right|_{\Gamma_2} = \lambda_{u_2}^k; \end{cases} \tag{47}$$

4. We do a recalculation to "stitch" the regions using one step of Newton's method to minimize the function:

$$\int_G \left(u_1^{k+1} - u_2^{k+1}\right)^2 ds \to \min. \tag{48}$$

5. Check the execution of the inequality:

$$\left|\lambda_{u_1}^{k+1} - \lambda_{u_1}^k\right| < \varepsilon, \left|\lambda_{u_2}^{k+1} - \lambda_{u_2}^k\right| < \varepsilon. \tag{49}$$

In the case of inequality, proceed to step 6, otherwise to step 2.

6. End of algorithm. Visualization of the results obtained.

To minimize (48), algorithm (46)–(49) has the same approach as algorithm (42)–(44).

3. Practical results

The developed software package consists of several modules, each of which has a corresponding graphical user interface. The general structure of the software package is shown in Fig. 3.

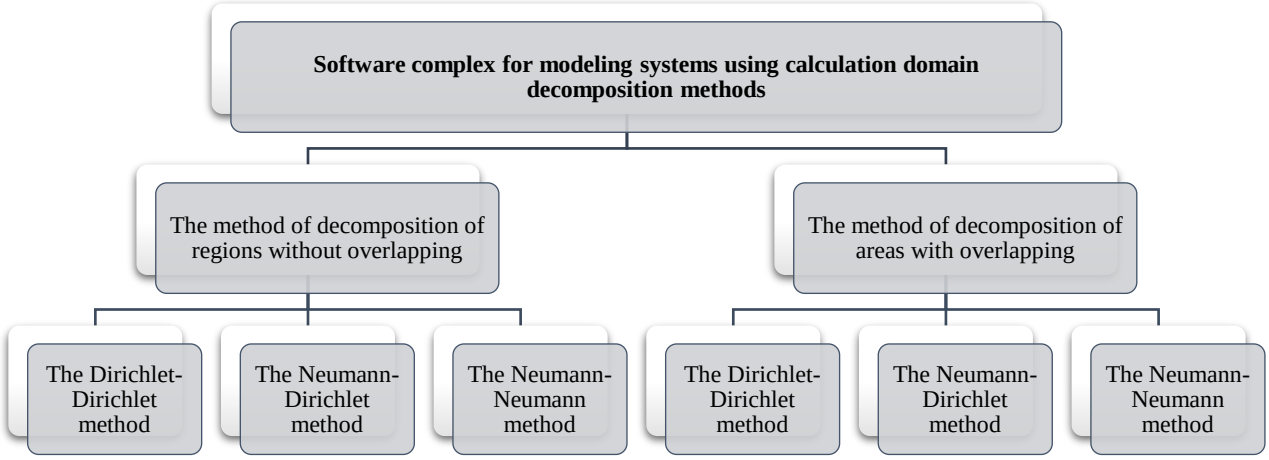

Figure 3. Structure of the software package for modeling the spread of malware

Testing of the developed software will be carried out in the following works, which will already describe the methodology for solving specific applied problems that are described by ordinary differential equations and differential equations in partial equations.

4. Discussion

The most effective methods for solving boundary value problems in domains with complex geometries usually involve simplifying the shape of the domain's geometry. Two types of iterative processes are used to solve this problem. The first type is based on the Schwarz subdomain alternation method, which is a domain decomposition method. The second type is similar to the dummy region method. The dissertation proposed developments of these approaches: the additive Schwartz method and the fictitious space method.

5. Conclusions

In this work, the obtained main mathematical models and methods of domain decomposition with intersection and without intersection of subdomains were described. The idea of area decomposition methods is that the area in which the task is considered is divided into subareas and an initial approximation is given. Next, the equations describing the given problem are solved by one of the methods in each subdomain with

special conditions on the common boundaries, which include the solutions obtained in the previous iteration. The iterative process continues until the specified accuracy is reached. Different variants of "gluing" conditions on the interfaces lead to different algorithms of the decomposition method. Mathematical models can be effectively applied during the study of complex energy objects and their systems

Software implementations in the MatLab 2022b environment were developed for the corresponding models. The obtained results are compared with the results of programs for solid areas. Approbation and comparative analysis of the mathematical models of computing methods obtained in this work will be carried out in the following scientific publications, which will contain a description of applied tasks, testing of the obtained methods for this kind of tasks, as well as a comparison of the methods with already known classical methods. These methods can also be used in parallel algorithms, since such problems can be solved in parallel on several threads.

References

1. Cuvelier, F., Gander, M. J., & Halpern, L. (2022). Fundamental Coarse Space Components for Schwarz Methods with Crosspoints. In S. C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 41–52). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_4
2. Claeys, X., Collino, F., Joly, P., & Parolin, E. (2022). Nonoverlapping Domain Decomposition Methods for Time Harmonic Wave Problems. In S. C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 53–66). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_5
3. Li, H., & Wheeler, M.F. (2022). Local Residual Minimization Smoothing for Improving Convergence Behavior of a Space-Time Domain Decomposition Method. In S. C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 103–114). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_9
4. Gu, L., Zhang, W., Liu, J., & Cai, X.-C. (2022). Decomposition and Preconditioning of Deep Convolutional Neural Networks for Training Acceleration. In S.C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 153–160). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_14
5. Genseberger, M., Fujisaki, A., Thiange, C., Eijsberg-Bak, C., Bijlsma, A., & Boderie, P. (2022). Domain Decomposition in Shallow Water Modelling of Dutch Lakes for Multiple Applications. In S.C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 281–289). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_29
6. Heinlein, A., Klawonn, A., Lanser, M., & Weber, J. (2022). Predicting the Geometric Location of Critical Edges in Adaptive GDSW Overlapping Domain Decomposition Methods Using Deep Learning. In S. C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 281–289). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_32
7. Pacheco, D. R. Q., & Steinbach, O. (2022). Space-Time Finite Element Tearing and Interconnecting Domain Decomposition Methods. In S. C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 479–486). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_51
8. Steinbach, O., & Gaulhofer, P. (2022). On Space-Time Finite Element Domain Decomposition Methods for the Heat Equation. In S. C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 547–554). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_59
9. Beuchler, S., Kinnewig, S., & Wick, T. (2022). Parallel Domain Decomposition Solvers for the Time Harmonic Maxwell Equations. In S. C. Brenner, E. Chung, A. Klawonn, F. Kwok, J. Xu, J. Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI. Lecture Notes in Computational Science and Engineering*, 145 (pp. 653–660). Springer, Cham. https://doi.org/10.1007/978-3-030-95025-5_71
10. Khaidurov, V., Tsiupii, T., & Zhovnovach, T. (2021, November 16–18). Modelling of Ultrasonic Testing and Diagnostics of Materials by Application of Inverse Problems. *ITTAP'2021: 1nd International Workshop on Information Technologies: Theoretical and Applied Problems* (pp. 1–5). ITTAP.
11. Khaidurov, V., Zaporozhets, A., & Tsiupii, T. (2021). Optimization models of industrial furnaces and methods for obtaining their numerical solution. In A. Zaporozhets, V. Artemchuk (Eds.), *Systems, Decision and Control in Energy II. Studies in Systems, Decision and Control*, 346 (pp. 121–139). Springer, Cham. https://doi.org/10.1007/978-3-030-69189-9_7

12. Khaidurov, V., Zaporozhets, A., & Tsiupii, T. (2022). Creation of High-Speed Methods for Solving Mathematical Models of Inverse Problems of Heat Power Engineering. In A. Zaporozhets (Ed.), *Systems Decision and Control in Energy III*, 399 (pp. 41–74). Springer, Cham. https://doi.org/10.1007/978-3-030-87675-3

ПРИСКОРЕННЯ ОБЧИСЛЕНЬ У МОДЕЛЮВАННІ ПРОЦЕСІВ У СКЛАДНИХ ОБ'ЄКТАХ І СИСТЕМАХ

Владислав Хайдуров[1, 2]*, канд. техн. наук, ст. досл., https://orcid.org/0000-0002-4805-8880
Вадим Татенко[1], https://orcid.org/0009-0008-4869-9689
Микита Литовченко[1], https://orcid.org/0009-0001-6671-7763
Тамара Цюпій[3], канд. фіз.-мат. наук, доцент, https://orcid.org/0000-0003-2206-2897
Тетяна Жовновач[4], https://orcid.org/0000-0003-1037-4383
[1]Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Берестейський просп., 37, Київ, 03056, Україна;
[2]Інститут загальної енергетики НАН України, вул. Антоновича, 172, Київ, 03150, Україна;
[3]Національний університет біоресурсів і природокористування України, вул. Героїв Оборони, 15, Київ, 03041, Україна;
[4]Черкаська філія ПВНЗ «Європейський університет», вул. Смілянська, 83, Черкаси, 18008, Україна
*Автор-кореспондент: allif0111@gmail.com

Анотація. *Розробка методів розпаралелювання обчислювальних процесів, які передбачають декомпозицію розрахункової галузі, є актуальним завданням при моделюванні складних об'єктів та систем. Складні об'єкти та системи можуть містити велику кількість елементів та взаємодій. Декомпозиція дозволяє розбити систему на простіші підсистеми, що спрощує аналіз та управління складністю. Шляхом поділу розрахункової області частини можна здійснювати паралельні обчислення, що підвищує ефективність розрахунків і скорочує час моделювання. Декомпозиція області дозволяє легко масштабувати модель для роботи з більшими або деталізованими системами. При правильному виборі методів декомпозиції можна покращити точність моделювання, оскільки різні частини системи можуть мати різні рівні деталізації та потребувати відповідні методи додаткового аналізу. Декомпозиція дозволяє розподілити моделювання між різними учасниками або пристроями, що є актуальним для розподілених систем або спільної роботи над проєктом. У даній роботі одержані математичні моделі, які полягають у побудові ітеративних процедур «зшиття» кількох областей у єдину цілу. Моделі передбачають різну складність розрахункових областей, що дає можливість виконувати різні підходи декомпозиції, зокрема як декомпозицію предметної області з перекриттям, так і без перекриття. Отримані математичні моделі декомпозиції предметної області можуть бути застосовані для об'єктів і систем, які мають різну геометричну складність. Моделі декомпозиції предметної області, які не використовують перекриття, містять різні ітеративні методи «зшиття» на спільній границі залежно від типів граничних умов (умова першого роду – умова Дирихле, або умова другого року – умова Неймана), а моделі декомпозиції області з перекриттям двох або більше областей полягають у задачі мінімізації для побудови ітераційної умови «зшиття» областей. Слід зазначити, що отримані моделі будуть ефективно працювати на всіх прикладних завданнях, які описують динамічну поведінку об'єктів та їх систем, але високий ступінь ефективності однієї моделі може бути нижчим за відповідний ступінь ефективності іншої моделі, оскільки кожна задача є індивідуальною.*
Ключові слова: математичне моделювання, декомпозиція розрахункової області, розпаралелювання, оптимізація, складні об'єкти і системи.

Надійшла до редколегії: 26.03.2024