

This paper presents a numerical realization of the Navier-Stokes equations in irregular domains using the fictitious domain method with a continuation along with the lowest coefficient. To solve numerous connected issues in irregular regions, the fictitious domain method is broadly used. The advantage of the fictitious domain method is that the problem is solved not in the original complex domain, but in a few other, easier domains. Using the method, computation is done easily for a sufficiently wide class of problems with arbitrary computational domains.

The problem is solved using two methods. The primary method is based on the development of a distinct issue in variables of the stream function and the vortex velocity using the pressure uniqueness condition. The second method is to understand the expressed issue by the fictitious domain method with a continuation by lower coefficients.

Using the fictitious domain method, a computational algorithm is constructed based on the explicit finite difference schemes. The finite difference scheme is stable and has high computational accuracy and it gives the possibility to parallelize. Temperature distributions and stream functions are presented as numerical results.

A parallel algorithm has been developed using Open Multi-Processing (hereinafter OpenMP) and Message Passing Interface (hereinafter MPI) technologies. Within the parallel approach, we used OpenMP technology for parallel calculation of vorticity and stream work, and for calculating temperature we applied MPI technology. The performance analysis on our parallel code shows favorable strong and weak scalability. The test results show that the code running in the parallel approach gives the expected results by comparing our results with those obtained while running the same simulation on the central processing unit (CPU)

Keywords: Navier-Stokes equations, stream function, vorticity, numerical methods, parallel algorithm

UDC 519.6

DOI: 10.15587/1729-4061.2022.254261

DEVELOPMENT OF PARALLEL IMPLEMENTATION FOR THE NAVIER-STOKES EQUATION IN DOUBLY CONNECTED AREAS USING THE FICTITIOUS DOMAIN METHOD

Almas Temirbekov

PhD, Associate Professor*

Arshyn Altybay

Corresponding author

PhD, Senior Lecturer

Department of Computer Science**

E-mail: arshyn.altybay@gmail.com

Laura Temirbekova

PhD, Senior Lecturer

Department of Informatics and Informatization of Education

Abai Kazakh National Pedagogical University

Dostyk ave., 13, Almaty, Republic of Kazakhstan, 050010

Syrym Kasenov

PhD, Senior Lecturer*

*Department of Computational Sciences and Statistics**

**Al-Farabi Kazakh National University

Al-Farabi ave., 71, Almaty, Republic of Kazakhstan, 050040

Received date 09.02.2022

Accepted date 23.03.2022

Published date 28.04.2022

How to Cite: Temirbekov, A., Altybay, A., Temirbekova, L., Kasenov, S. (2022). Development of parallel implementation for the Navier-Stokes equation in doubly connected areas using the fictitious domain method. *Eastern-European Journal of Enterprise Technologies*, 2 (4 (116)), 38–46. doi: <https://doi.org/10.15587/1729-4061.2022.254261>

1. Introduction

The rapid growth of the performance of a modern computer and the development of parallel computing technologies contributed to the solution of important practical problems in the ecology.

One example of such tasks is the performance evaluation and forecasting of air pollution assessment indicators. Due to the complexity of mathematical models describing these processes, calculations for one domain can take a longer time. Therefore, the issue of developing efficient parallel algorithms that can significantly speed up calculations becomes relevant. High-performance computing is widely used in scientific studies. Every day, computer technologies and hydrodynamic models are being developed, which make it possible to assess and analyze various technological processes.

In this regard, the efficiency of solving scientific problems increases. Many researchers proposed their algorithms using various methods for parallel implementation of numerical solution of the Navier-Stokes equation.

For numerical modeling of applied problems of hydrodynamics, ecology with complex geometry, it is effective to use the fictitious domain method. When using the fictitious domain method, the solution of the problem is not carried out in the considered complex domain, it is solved in an easier domain. Therefore, this method has a high degree of automation when programming in the algorithmic languages.

Currently, there are several methods for the numerical solution of boundary value problems in complex geometric domains, such as the method of curvilinear grids and the method of fictitious domains. The construction of curvilinear grids for the numerical solution of problems requires the transformation

of the equation into curvilinear coordinates, which has a more complex form than the original equation. When constructing curvilinear grids, various requirements are imposed on the difference equations, which makes the construction of curvilinear grids a complex mathematical problem.

Therefore, research devoted to the construction of an efficient and easily parallelizable finite difference scheme and the application of the fictitious domain method for the numerical implementation of the Navier-Stokes equation for a doubly connected domain is relevant.

2. Literature review and problem statement

To begin with, we present a literature review on the application of the fictitious domain method. In [1], the authors started widely using the fictitious domain method applying it to the elliptic type problems and Navier-Stokes equations. They were interested in the estimates for the error functions.

In [2], the fictitious domain method is used to solve many problems in computational fluid dynamics. They also proposed a methodology that allows the direct numerical simulation of incompressible viscous fluid flow past moving rigid bodies. They concluded the methodology is particularly well suited to the direct numerical simulation of particulate flow, such as the flow of mixtures of rigid solid particles and incompressible viscous fluids, possibly non-Newtonian.

In [3], the problem of a viscous incompressible fluid flow past obstacles with a slip boundary condition according to Navier's law is considered. In this paper, they discussed a least-squares/fictitious domain method for incompressible viscous flow around obstacles with Navier slip boundary condition.

Note that in [1–3] the authors gave clear definitions and described the importance of the fictitious domain method and implemented this method for solving various problems of computational fluid dynamics. The calculation methods were based on finite element methods.

In [4], the authors study the two-phase Stokes problem with surface tension forces. This paper explores a mixed finite element formulation for solving the Stokes problem with general surface forces causing a jump in the normal trace of the stress tensor at the interface, which splits the domain into two subdomains.

In [5], a fictitious domain method with Distributed Lagrange Multipliers for simulating two-dimensional (*hereinafter 2D*) unsteady shear-thinning non-newtonian incompressible flow in a single-screw and twin-screw extruder is developed. The issues of flow motion with an arbitrary density of particles are considered in [6].

In [4–6], the authors successfully applied the fictitious domain method for numerical simulation of incompressible fluids in different areas, carried out theoretical and numerical analyses, and also used the finite element method. It is important to note that the finite element method won't always work for parallel computing.

The work [7] is devoted to the numerical approximation of the fluid structure interaction for stabilization of the fluid flow around an unstable stationary solution in a two-dimensional domain, in the presence of boundary perturbations. Here the authors to implement the fictitious domain method applied the Extended Finite Element method with stabilization terms. However, numerical schemes are not clearly described.

In [8], the energy steadiness of a one-field fictitious domain method is demonstrated and approved by numerical

tests in two and three dimensions. The distinctive feature of this strategy is that it understands for one speed field for the total fluid-structure domain; the intuitive stays decoupled until tackling the ultimate linear algebraic equations.

In [9], the authors proposed a new framework to directly simulate super-quadric (*hereinafter SQ*) particles in fluid flows based on a forcing fictitious domain method. Specifically, a simple SQ function with five parameters has been used to generate different particle shapes. The results of the numerical solution and its stability have not been studied.

In [10], the method of fictitious domains with a penalty is studied for the Stokes problem with the Dirichlet boundary condition. It also introduces some interpolation/projection operators, as well as an inf-up condition with a norm depending on ϵ . Using these preliminary data, error estimates for the finite element approximation were obtained.

The paper [11] is devoted to the application of the fictitious domain method in the numerical simulation of the transducer of impulsive oscillations. In [12], the method of fictitious domains with a distributed Lagrange multiplier is investigated for parabolic problems of the jump type with moving boundaries.

Application of the fictitious domain method to the problems with discontinuous coefficients started in [13]. In [14], the dispersed Lagrange multiplier/fictitious domain (DLM/FD) finite element method is examined for a non-specific Stokes interface issue with hop coefficients, which has a place in a sort of linearized stationary fluid-structure interaction issue.

Equations of the elliptic type with strongly varying coefficients are investigated in [15]. In these works, a special method is used for the numerical solution of these equations. For this, the estimates for the convergence rate of the numerical algorithm are proved. Based on these estimates, a computational algorithm is developed and numerical calculations are carried out to illustrate the effectiveness of the proposed method.

In [16], the fictitious domain method is successfully implemented for the elliptic equation and a theorem is proved for the rate of convergence of the iterative process developed. But the parallelization of numerical solutions is not considered.

One of the first works devoted to paralleling of the computational algorithms using the fictitious domain method started in [17]. Namely, a parallel computational algorithm is constructed for the three-dimensional Helmholtz equation. However, the parallel algorithm is not clearly and thoroughly described.

In [18], a parallel direct-forcing (DF) fictitious domain method for the simulation of particulate flows is proposed. Here, the authors developed an efficient parallel code of the fictitious domain method. The parallel approach was based on an implicit finite difference scheme and explicit finite difference scheme using MPI technology. However, the parallelization of a tridiagonal matrix system is not clearly described here.

The paper [19] describes the possibilities of using the fictitious domain method for problems of biomechanics. However, there is no justification for the convergence and stability of the difference scheme and clear description of parallelization.

The work [20] suggested parallelization of the three-dimensional Navier-Stokes equation based on an implicit difference scheme. Here, parallelization processes are not explicitly described. In [21], another parallel numerical implementation of the 2D Navier-Stokes equation using pseudo-spectral methods is proposed. In this work, the authors parallelized two-dimensional spectral codes by combining the Parallel Task-Distribution (PTD) and Parallel Fast Fourier Transform (PFFT) schemes. And it showed a significant

improvement in parallel efficiency. But the strategy used only MPI technology. In [22], a parallel computational code for the numerical integration of the Navier-Stokes equations using the fourth-order Runge-Kutta scheme is developed. The paper does not consider the problems of the Navier-Stokes equation in doubly connected areas. In [23], a hybrid implementation was developed using the MPI, OpenMP and Compute Unified Device Architecture (CUDA) technologies and showed that this implementation gives good results.

Most ongoing research is still often based on finite element methods to implement fictitious domain methods for the Navier-Stokes equations since it does not always succeed in parallelization. Little attention has been paid to parallelization using the possibilities of modern heterogeneous computer architecture.

Let us dwell on the advantages and disadvantages of the above variants of fictitious domain methods. The fictitious domain method proposed and implemented in the works allows satisfying the boundary conditions on the actual boundary using variational principles. This process in some cases may even behave worse than the solution of the original problem on a non-uniform grid consistent with a curvilinear boundary. The results of numerical calculations show convergence to the solution in the «average» due to the use of variational principles, that is, the functional containing the main equation and the boundary condition is minimized. In this paper, the authors propose an explicit finite difference for the numerical solution, which was later implemented in parallel computing.

This review allows us to claim that it is reasonable and necessary to study the parallel numerical implementation of the Navier-Stokes equation in doubly connected domains using the fictitious domain method.

3. The aim and objectives of the study

The aim of this study is to solve numerically Navier-Stokes equations in doubly connected domains by the fictitious domain method using parallel computing technologies.

To achieve this aim, the following objectives are accomplished:

- to formulate a finite difference scheme for Navier-Stokes equations in irregular domains using the fictitious domain method;
- to develop a computational algorithm to solve finite-difference problems and present numerical results;
- to develop a parallel approach for the numerical simulation of the Navier-Stokes equation in doubly connected domains.

4. Materials and methods

4.1. Mathematical model

In this section, we study the changes in the temperature and stream function in the domain as shown in Fig. 1.

Fig. 1 shows the physical domain with the boundary conditions of the problem. All variables presented in the figure are described after equations (1)–(6).

To simulate convective flows, we consider the Navier-Stokes equations in the Boussinesq approximation:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} = \frac{1}{\text{Re}} \Delta u, \tag{1}$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} = \frac{1}{\text{Re}} \Delta v - \text{Gr}\theta, \tag{2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{3}$$

$$\frac{\partial \theta}{\partial t} + u \frac{\partial \theta}{\partial x} + v \frac{\partial \theta}{\partial y} = \frac{1}{\text{RePr}} \Delta \theta, (x, y) \in D, t \in (0, T] \tag{4}$$

with initial and boundary conditions:

$$\begin{aligned} u &= u_0(x, y), \quad v = v_0(x, y), \\ \theta &= \theta_0(x, y), \quad (x, y) \in \bar{D}, \quad t = 0, a, \end{aligned} \tag{5}$$

$$\begin{aligned} u &= a_x(x, y, t), \quad v = a_y(x, y, t), \\ \theta &= \xi(x, y, t), \quad (x, y) \in \partial D, \quad t = (0, T], \end{aligned} \tag{6}$$

where u, v are the velocity components, p is the pressure, θ is the temperature, Re is Reynolds number, Gr is Grashof number, Pr is Prandtl number, $\partial D = \gamma_1 \cup \gamma_2$ is the boundary of the domain D , \bar{D} is the closed domain with boundary.

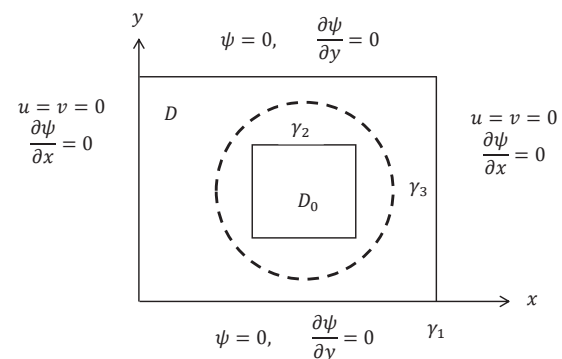


Fig. 1. Physical domain

It is convenient to solve problem (1)–(6) by eliminating pressure from the equations of motion and introducing new variables – stream function ψ and vortex of velocity ω . Integral conditions for pressure uniqueness play an essential role. In [24] for the unique solvability of the difference scheme for the Navier-Stokes equation, the pressure uniqueness condition of the following form is added to the system of equations:

$$\iint_{\bar{D}} p(x, y) dx dy = 0. \tag{7}$$

The need to formulate a condition of the form (7) is due to the fact that the difference analog of (3) is not linearly independent. The work [25] proposed another version of the pressure unambiguity condition written as:

$$\oint_{\gamma_3} \frac{\partial \Pi}{\partial x} dx + \frac{\partial \Pi}{\partial y} dy = 0, \tag{8}$$

where Π is the full pressure.

Let us introduce the stream function ψ and the vortex of velocity ω related to the velocity components by the following relations:

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}, \quad \omega = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}. \tag{9}$$

The problem (1)–(6) in variables ψ, ω can be written as follows [25]:

$$\frac{\partial \omega}{\partial t} + \frac{\partial \psi}{\partial y} \frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial y} \frac{\partial \omega}{\partial y} = \frac{1}{\text{Re}} \Delta \omega + \text{Gr} \frac{\partial \theta}{\partial x}, \quad (10)$$

$$\Delta \psi = \omega, \quad (11)$$

$$\frac{\partial \theta}{\partial t} + \frac{\partial \psi}{\partial y} \frac{\partial \theta}{\partial x} - \frac{\partial \psi}{\partial y} \frac{\partial \theta}{\partial y} = \frac{1}{\text{RePr}} \Delta \theta,$$

$$(x, y) \in D, t \in (0, T], \quad (12)$$

$$\omega = \alpha(x, y), \theta = \varphi(x, y), (x, y) \in D, t = 0, \quad (13)$$

$$\psi = \xi_1(x, y, t), \frac{\partial \psi}{\partial \bar{n}} = \eta_1(x, y, t),$$

$$(x, y) \in \gamma_1, t \in (0, T], \quad (14)$$

$$\psi = \xi_2(x, y, t) + \lambda(t), \frac{\partial \psi}{\partial \bar{n}} = \eta_2(x, y, t),$$

$$(x, y) \in \gamma_2, t \in (0, T], \quad (15)$$

$$\theta = \beta_l(x, y, t), (x, y) \in \gamma_l, l = 1, 2, t \in (0, T], \quad (16)$$

where $\alpha, \varphi, \xi_i, \eta_i, \beta_i, i = 1, 2$ are given functions, \bar{n} is the outward normal to the boundary ∂D .

Note that condition (8) can be written in the following form:

$$\oint \left[\left(\frac{\partial^2 \psi}{\partial t \partial y} + \omega \frac{\partial \psi}{\partial x} + \frac{1}{\text{Re}} \frac{\partial \omega}{\partial y} \right) dx + \left(-\frac{\partial^2 \psi}{\partial t \partial x} + \omega \frac{\partial \psi}{\partial y} - \frac{1}{\text{Re}} \frac{\partial \omega}{\partial x} - \text{Gr} \theta \right) dy \right] = 0. \quad (17)$$

To solve numerically the problems (10)–(16) we need to approximate them to finite-difference schemes, which is presented in detail in the next section.

The numerical experiment was carried out on a computer with the following technical characteristics: CPU Intel Core (TM) i7-9800X, 3.80 GHz, RAM 64 GB, NVIDIA GeForce RTX 2080 TI OC Windows 10 Pro. All calculations were made in the C++ programming language and visualization was obtained using the Tecplot program.

5. Results of the study of a parallel algorithm for the numerical solution of the Navier-Stokes equations in irregular domains

5.1. Finite difference scheme

The solution of the difference analog of the problem (10)–(17) will be sought in the form:

$$\begin{aligned} \omega^{n+1}(x, y) &= \omega_0^{n+1}(x, y) + \lambda^{n+1} \omega_1^{n+1}(x, y), \\ \psi^{n+1}(x, y) &= \psi_0^{n+1}(x, y) + \lambda^{n+1} \psi_1^{n+1}(x, y), \end{aligned} \quad (18)$$

where n is the number of iterations.

Auxiliary Problem 1:

$$\begin{aligned} \frac{\omega_0^{n+1} - \omega^n}{\tau} + \frac{\partial \psi^n}{\partial y} \frac{\partial \omega_0^{n+1}}{\partial x} - \frac{\partial \psi^n}{\partial x} \frac{\partial \omega_0^{n+1}}{\partial y} &= \frac{1}{\text{Re}} \Delta \omega_0^{n+1} + \text{Gr} \frac{\partial \theta^{n+1}}{\partial x}, \\ \Delta \psi_0^{n+1} &= -\omega_0^{n+1}, (x, y) \in D, \end{aligned} \quad (19)$$

$$\psi_0^{n+1} |_{\gamma_1} = 0, \frac{\partial \psi_0}{\partial \bar{n}} |_{\gamma_1} = 0, \psi_0^{n+1} |_{\gamma_2} = 0, \frac{\partial \psi_0}{\partial \bar{n}} |_{\gamma_2} = 0. \quad (20)$$

Auxiliary Problem 2:

$$\begin{aligned} \frac{\omega_1^{n+1} - \omega^n}{\tau} + \frac{\partial \psi^n}{\partial y} \frac{\partial \omega_1^{n+1}}{\partial x} - \frac{\partial \psi^n}{\partial x} \frac{\partial \omega_1^{n+1}}{\partial y} &= \frac{1}{\text{Re}} \Delta \omega_1^{n+1}, \\ \Delta \psi_1^{n+1} &= -\omega_1^{n+1}, (x, y) \in D, \end{aligned} \quad (21)$$

$$\psi_1^{n+1} |_{\gamma_1} = 0, \frac{\partial \psi_1}{\partial \bar{n}} |_{\gamma_1} = 0, \psi_1^{n+1} |_{\gamma_2} = 1, \frac{\partial \psi_1}{\partial \bar{n}} |_{\gamma_2} = 0. \quad (22)$$

Let us build a uniform mesh in the domain D by:

$$D_h = \left\{ (x_i, y_j) : x_i = (i-1)h_1, y_j = (j-1)h_2; \right. \\ \left. i = 1, 2, \dots, n_1, j = 1, 2, \dots, n_2 \right\},$$

where $h_1 = l_1 / (n_1 - 1)$, $h_2 = l_2 / (n_2 - 1)$. Suppose that the inner subdomain D_0 is a rectangle of the following form:

$$D_{0h} = \left\{ (x, y) : x_{k_1} \leq x \leq x_{k_2}, y_{m_1} \leq y \leq y_{m_2}; \right. \\ \left. x_i = (i-1)h_1, y_j = (j-1)h_2; \right. \\ \left. i = 1, 2, \dots, n_1, j = 1, 2, \dots, n_2 \right\},$$

Nodes of a uniform grid are constructed in the domain D . Consider the domain D_1 covering the domain D_0 , i.e. $D_0 \subset D_1$, $D_1 = \{(x, y), x_{k3} \leq x \leq x_{k4}, y_{m3} \leq y \leq y_{m4}\}$. Consider a difference analog of the condition (17) and substituting the expansion of the form (18). To determine λ^{n+1} , we obtain the following equation:

$$M \lambda^{n+1} = N, \quad (23)$$

where

$$\begin{aligned} M = & \sum_{j=m_3}^{m_4} \left[\frac{\psi_{1,x,k_3,j}^{n+1}}{\tau} + 0.5 \omega_{1,k_3+1,j}^{n+1} \psi_{y,k_3,j}^n + \right. \\ & \left. + 0.5 \omega_{1,k_3,j}^{n+1} \psi_{y,k_3+1,j}^n - \right. \\ & \left. - \frac{1}{\text{Re}} \omega_{1,x,k_3,j}^{n+1} - \beta g \theta_{k_3+\frac{1}{2},j}^{n+1} \right] h_2 + \\ & + \sum_{j=k_3}^{k_4} \left[\frac{\psi_{1,\bar{y},i,m_4}^{n+1}}{\tau} + 0.5 \omega_{1,i,m_4-1}^{n+1} \psi_{x,i,m_4}^n + \right. \\ & \left. + 0.5 \omega_{1,i,m_4}^{n+1} \psi_{x,i,m_4-1}^n + \frac{1}{\text{Re}} \omega_{1,\bar{y},i,m_4}^{n+1} \right] h_1 - \\ & - \sum_{j=m_3}^{m_4} \left[\frac{\psi_{1,x,k_4,j}^{n+1}}{\tau} + 0.5 \omega_{1,k_4+1,j}^{n+1} \psi_{y,k_4,j}^n + \right. \\ & \left. + 0.5 \omega_{1,k_4,j}^{n+1} \psi_{y,k_4+1,j}^n - \right. \\ & \left. - \frac{1}{\text{Re}} \omega_{1,x,k_4,j}^{n+1} - \beta g \theta_{k_4+\frac{1}{2},j}^{n+1} \right] h_2 + \\ & + \sum_{j=k_3}^{k_4} \left[\frac{\psi_{1,\bar{y},i,m_3}^{n+1}}{\tau} + 0.5 \omega_{1,i,m_3-1}^{n+1} \psi_{x,i,m_3}^n + \right. \\ & \left. + 0.5 \omega_{1,i,m_3}^{n+1} \psi_{x,i,m_3-1}^n + \frac{1}{\text{Re}} \omega_{1,\bar{y},i,m_3}^{n+1} \right] h_1, \end{aligned} \quad (24)$$

$$\begin{aligned}
 N = & \sum_{j=m3}^{m4} \left[\frac{\Psi_{0,x,k3,j}^{n+1} - \Psi_{x,k3,j}^n + 0.5\omega_{0,k3+1,j}^{n+1} \Psi_{y,k3,j}^n +}{+0.5\omega_{0,k3,j}^{n+1} \Psi_{x,k3+1,j}^n - \frac{1}{\text{Re}} \omega_{0,x,k3,j}^{n+1}} \right] h_2 + \\
 & + \sum_{j=k3}^{k4} \left[\frac{\Psi_{0,\bar{y},i,m4}^{n+1} - \Psi_{\bar{y},i,m4}^n + 0.5\omega_{0,i,m4-1}^{n+1} \Psi_{x,i,m4}^n +}{+0.5\omega_{0,i,m4}^{n+1} \Psi_{\bar{x},i,m4-1}^n + \frac{1}{\text{Re}} \omega_{0,\bar{y},i,m4}^{n+1}} \right] h_1 + \\
 & + \sum_{j=m3}^{m4} \left[\frac{\Psi_{0,x,k4,j}^{n+1} - \Psi_{x,k4,j}^n + 0.5\omega_{0,k4+1,j}^{n+1} \Psi_{y,k4,j}^n +}{+0.5\omega_{0,k4,j}^{n+1} \Psi_{y,k4+1,j}^n - \frac{1}{\text{Re}} \omega_{0,x,k4,j}^{n+1}} \right] h_2 + \\
 & + \sum_{j=k3}^{k4} \left[\frac{\Psi_{0,\bar{y},i,m3}^{n+1} - \Psi_{\bar{y},i,m3}^n + 0.5\omega_{0,i,m3-1}^{n+1} \Psi_{x,i,m3}^n +}{+0.5\omega_{0,i,m3}^{n+1} \Psi_{\bar{x},i,m3-1}^n + \frac{1}{\text{Re}} \omega_{0,\bar{y},i,m3}^{n+1}} \right] h_1. \quad (25)
 \end{aligned}$$

Here, g is the acceleration of gravity, x is the forward difference, (\dot{x}, \dot{y}) are the central differences, \bar{x}, \bar{y} are the backward difference and defined by the following formulas:

$$\begin{aligned}
 x = \frac{x_{i+1} - x_i}{\Delta x}, \quad \dot{x} = \frac{x_{i+1} - x_{i-1}}{2\Delta x}, \quad \dot{y} = \frac{y_{i+1} - y_{i-1}}{2\Delta y}, \\
 \bar{x} = \frac{x_i - x_{i-1}}{\Delta x}, \quad \bar{y} = \frac{y_i - y_{i-1}}{\Delta y}.
 \end{aligned}$$

By the fictitious domain method, let us consider the following problem auxiliary to (10)–(17):

$$\begin{aligned}
 \frac{\partial \omega^\varepsilon}{\partial t} + \frac{\partial \psi^\varepsilon}{\partial y} \frac{\partial \omega^\varepsilon}{\partial x} - \frac{\partial \psi^\varepsilon}{\partial x} \frac{\partial \omega^\varepsilon}{\partial y} = \\
 = \frac{1}{\text{Re}} \Delta \omega + \text{Gr} \frac{\partial \theta^\varepsilon}{\partial x} - \text{div}(k(x,y) \nabla \psi), \quad (26)
 \end{aligned}$$

$$\Delta \psi^\varepsilon = \omega^\varepsilon, \quad (27)$$

$$\frac{\partial \theta^\varepsilon}{\partial t} + \frac{\partial \psi^\varepsilon}{\partial y} \frac{\partial \theta^\varepsilon}{\partial x} - \frac{\partial \psi^\varepsilon}{\partial x} \frac{\partial \theta^\varepsilon}{\partial y} = \frac{1}{\text{RePr}} \Delta \theta^\varepsilon, \quad (28)$$

$$\psi^\varepsilon|_{\gamma_1} = 0, \quad \frac{\partial \psi^\varepsilon}{\partial \bar{n}}|_{\gamma_1} = 0, \quad (29)$$

$$\theta^\varepsilon = \beta_l(x, y, t), \quad (x, y) \in \gamma_l, \quad l = 1, 2, t \in (0, T], \quad (30)$$

where

$$k(x, y) = \begin{cases} 1, & (x, y) \in D_0; \\ 0, & (x, y) \in D/D_0, \end{cases} \quad (31)$$

for the small parameter $\varepsilon > 0$.

To solve numerically this problem in the multiply connected domain, in what follows we will work with the explicit difference scheme corresponding to the problem.

5.2. Computational algorithm to solve finite-difference problems and numerical results

Here, we present a sequential algorithm based on the stable explicit difference scheme (18)–(25). The algorithm for solving the problem (10) is shown in Algorithm 1. Here,

to get a convergence value of ψ , we use the «while» cycle and quite small ε , in our case $\varepsilon = 10^{-7}$.

Algorithm 1. Sequential algorithm for solving the equation (10):

- 1: get the input parameters $n_1, n_2, \varepsilon, L, T, m_0$...
- 2: initialize $\psi_0, \omega_0, \theta_0$ with initial values and boundary conditions
- 3: **while** ($t < T$) **do**
- 4: **for** $i \leftarrow 1$ to n_1 **do**
- 5: **for** $j \leftarrow 1$ to n_2 **do**
- 6: calculate ω_1
- 7: **end for**
- 8: **end for**
- 9: **while** $k_0 < m_0$ **do**
- 10: $k_0 = k_0 + 1$
- 11: **for** $i \leftarrow 1$ to n_1 **do**
- 12: **for** $j \leftarrow 1$ to n_2 **do**
- 13: calculate ψ_1
- 14: **end for**
- 15: **end for**
- 16: swap (ψ_0, ψ_1)
- 17: $\psi_1 - \psi_0 < \varepsilon$ break;
- 18: **end while**
- 19: Set the boundary condition for ω_1 by the Ohm's formulas
- 20: **for** $i \leftarrow 1$ to n_1 **do**
- 21: **for** $j \leftarrow 1$ to n_2 **do**
- 22: calculate θ_1
- 23: **end for**
- 24: **end for**
- 25: swap (ω_0, ω_1)
- 26: swap (θ_0, θ_1)
- 27: $t \leftarrow t + \Delta t$
- 28: **end while**

Numerical results. Here, we present numerical simulations. Algorithm 1 is used to numerically solve the test problem (10)–(16) for the Navier-Stokes equations of a viscous incompressible fluid in variables, stream function, vortex of velocity in the Boussinesq approximation.

Fig. 2–4 present the results of numerical calculations of the isolines of stream function and temperature.

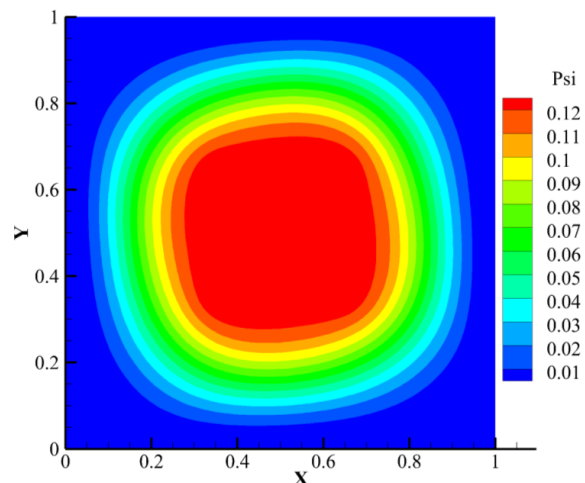


Fig. 2. Isolines of stream functions. The size of the cavity 1.0×1.0 , at the inner boundaries $\theta = 0.5$, $\text{Pr} = 5.39$, $\text{Gr} = 100$

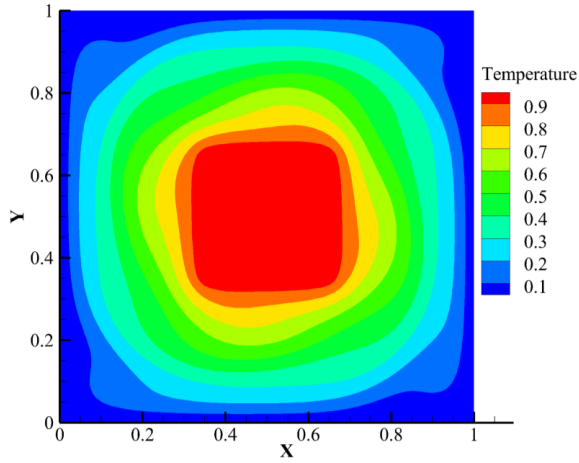


Fig. 3. Isotherms. The size of the cavity 1.0×1.0 , at the inner boundaries $\theta = 1.0$, $Pr = 5.39$, $Gr = 100$

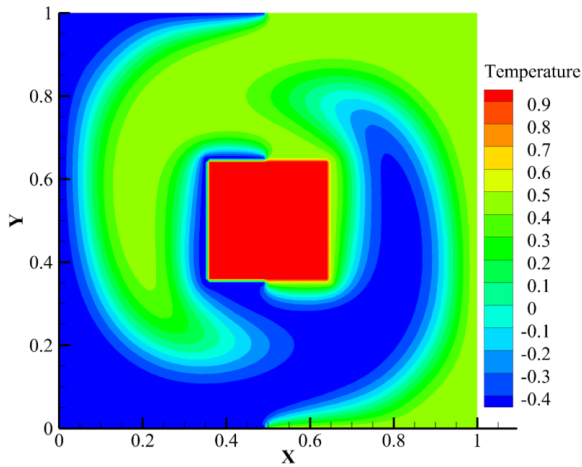


Fig. 4. Isotherms. The size of the cavity 1.0×1.0 , temperature of the right sides of the boundaries $\theta = 0.5$, left side $\theta = -0.5$, $Pr = 5.39$, $Gr = 100$

Temperature distributions and stream functions are presented as numerical results. The results are obtained for different sizes of the cavity, temperature regimes at the boundary, and values that determine the flow of dimensionless parameters, which are the Grashof Gr and Prantl Pr numbers.

5.3. Parallel approach to parallelize the sequential algorithm for numerical simulation of the Navier-Stokes equation in doubly connected domains

From the results of the calculations in the sequential algorithm, we see that the calculation of the temperature θ takes the longest time, that is, 60 % of the total calculation time since the calculation is performed on 4 separate subdomains. 15 % of the time is spent calculating the stream function ψ and calculating ψ using just two cycles. The rest of the computation time is spent on calculating the vortex velocity ω and initializing the initial conditions.

An explicit scheme for solving this problem allows using the parallelism of multi-core processors. When organizing parallel computing, we use the above data. That is, we calculate the temperature and ψ in parallel. For parallel computing, we use MPI and OpenMP technologies. We use OpenMP technology for parallel calculation of vorticity and stream function, because this technology is very convenient for calculating

cycles, and for calculating temperature we use MPI technology, that is, we use the MPI+OpenMP approach (Fig. 5).

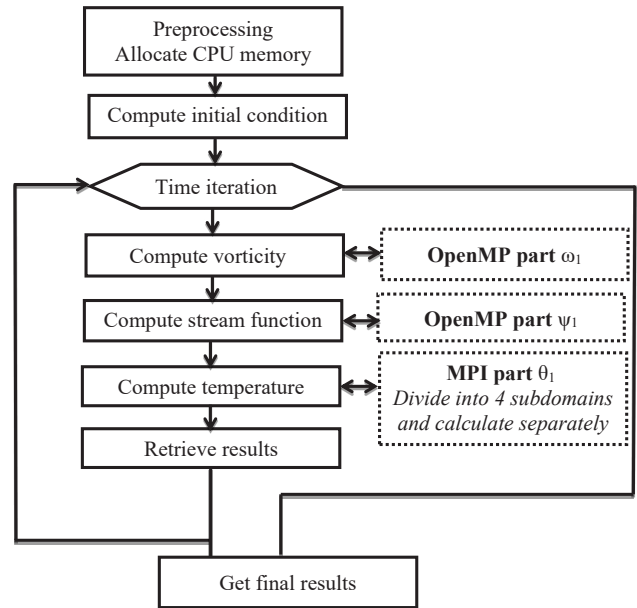


Fig. 5. MPI+OpenMP approach

Geometric decomposition of the grid region is chosen as the main approach to parallelization. In this case, we use two-dimensional decomposition (Fig. 6). After the stage of decomposition, when the data is divided into blocks to build a parallel algorithm, we proceed to the stage of establishing connections between blocks, calculations in which will be performed in parallel, planning communications.

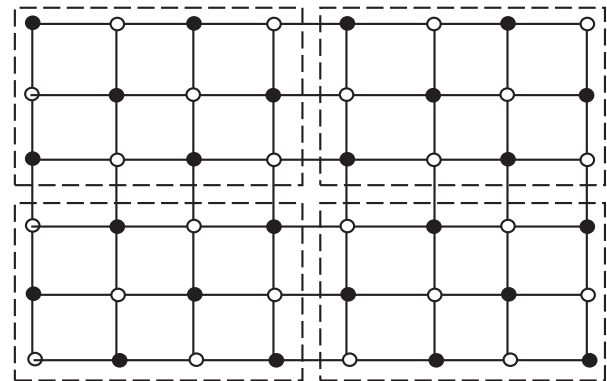


Fig. 6. Geometric data decomposition

Due to the used template of the explicit difference scheme, to calculate the next approximation at the border nodes of each subdomain, it is necessary to know the values of the grid function from the adjacent bordering processor element. For this purpose, fictitious cells are created at each computational node to store data from a neighboring computational node, and transfers of these boundary values are organized, which are necessary to ensure the homogeneity of calculations by explicit formulas.

Computational experiments. All calculations and tests were carried out on a personal computer core i9, 9th generation RAM 32GB and NVIDIA Geforce RTX 2080 Ti. The performance of a parallel algorithm is determined by calculating its speedup. In strong scaling, speedup is defined

as the ratio of the execution time of the sequential algorithm for a particular problem to the execution time of the parallel algorithm:

$$S = \frac{T_s}{T_p}$$

where S is the speedup, T_s is the computational time for running the program using one processor, T_p is the computational time running the same program with p processors. The efficiency is defined as the ratio of speedup to the number of processors.

Efficiency measures the fraction of time for which a processor is usefully used:

$$E = \frac{S}{p} = \frac{T_s}{pT_p}$$

where p is the number of processors. The test results are presented in Fig. 7, 8.

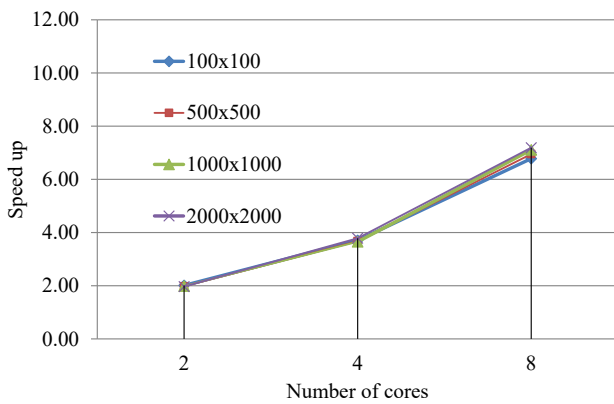


Fig. 7. Speedup

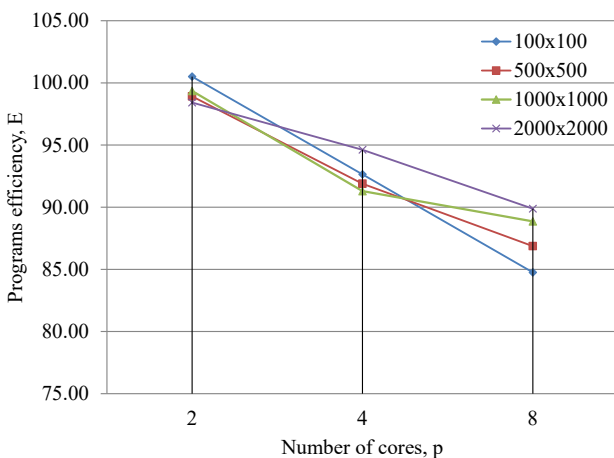


Fig. 8. Efficiency

Next, we study the weak scalability of the solver. The mesh size per core is set to 500×500 . We increase the number of nodes and measure the execution time for 100 iterations. The test results are presented in Fig. 9.

We observe in Fig. 9 that the execution time increases with the number of cores. This increase of time is due to a larger amount of information exchanged as we increase the total mesh size.

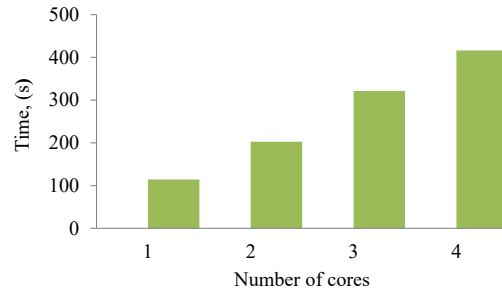


Fig. 9. Weak scaling of the solver (MPI+OpenMP implementation)

6. Discussion of the results of the study on the numerical solution of the Navier-Stokes equations in two-connected domains and parallelization

Formulation of a finite difference scheme. We have constructed a finite difference scheme for the Boussinesq approximation (the initial boundary difference problem (10)–(16) for the numerical solution of the nonlinear system of the Navier-Stokes equations in two-connected domains.

The problem is solved using two methods. The first method is based on the construction of a difference problem in variables of the stream function and the vortex velocity using the pressure uniqueness condition. This condition allows us to determine the coefficient used in (18). The solutions for the stream function and the vortex velocity are found as the sum of two simple problems. The developed algorithm converges uniformly at a certain number of iterations, and using the developed numerical algorithm, more accurate results are obtained.

The second approach to solving the stated problem is by the fictitious domain method with a continuation by lower coefficients (26)–(31). The solution of the problem is found with accuracy. This method is simple to implement and does not require satisfaction of the pressure uniqueness condition.

Numerical solution. We present the calculation algorithm, the algorithm is based on an explicit finite difference scheme and used $\epsilon = 10^{-7}$ to achieve high computational accuracy.

The above-described algorithm 1 numerically solved the tasks. The calculations used a uniform grid with dimensions 100×100 , 500×500 , $1,000 \times 1,000$, $2,000 \times 2,000$. The numerical experiment was carried out on a personal computer configured with 10th generation RAM 32 GB Intel i7 core processors, 2.40 GHz series. To visualize the obtained results, the Tecplot program was used, the results obtained are presented in Fig. 2–4. Fig. 2 shows the isolines of the stream functions, and Fig. 3, 4 show isotherms with different parameters at the boundaries.

Parallelization. To deal with parallelization we use OpenMP and MPI technologies. OpenMP is used to calculate vorticity and stream functions because they are calculated by just two-cycle, to calculate temperature function we use MPI technology. By comparing the calculation time of serial and parallel algorithms we can see that parallel algorithm gives quite high results.

On this topic, there are several methods for the numerical solution of boundary value problems in complex geometric areas, one of them is the fictitious domain method. The rest of the methods are based on curvilinear meshes, and when constructing curvilinear meshes, different requirements are imposed on the difference equations, which makes the construction

of curvilinear meshes a complex mathematical problem. When solving the problem by the fictitious domain method, there is no need to build a curvilinear grid. This makes it easier to solve the problem, since the original equation remains unchanged. One disadvantage of this work can be noted that an explicit scheme is used to solve the problem, it is necessary to take into account the Courant-Friedrichs-Lewy (CFL) condition, which affects the computational time. In the future, the authors plan to use other methods that do not require the fulfillment of the CFL condition.

Further research will be aimed at creating parallel algorithms and speeding up computations associated with solving nonlinear problems of multiphase filtering by finite element methods using the hardware-software architecture of parallel computing Compute Unified Device Architecture (CUDA).

7. Conclusions

1. An explicit finite difference scheme is constructed for the numerical solution of Navier-Stokes equations in irregular domains with a continuation by the lowest coefficient. This difference scheme for the numerical solution of the formulated problem is Courant-Friedrichs-Lewy (CFL) stable.

To illustrate the efficiency and stability of the constructed scheme, calculations were carried out on uniform grids with different dimensions.

2. Using the fictitious domain method, a computational algorithm is developed to solve auxiliary finite-difference problems. Note that this method does not require satisfaction of the pressure uniqueness condition and in this algorithm, we implement the Jacobi method because this method simplifies parallelization.

3. A parallel algorithm for the numerical solution of the Navier-Stokes equations in irregular domains using OpenMP and MPI technologies is developed. OpenMP technology is used to calculate the vorticity and stream functions, MPI technology is used to calculate the temperature function. The results of computational experiments show that the use of parallel algorithms using MPI+OpenMP for this kind of tasks gives a good acceleration.

Acknowledgments

This research is funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. AP09058430).

References

1. Glowinski, R., Pan, T.-W. (1992). Error estimates for fictitious domain/penalty/finite element methods. *Calcolo*, 29 (1-2), 125–141. doi: <https://doi.org/10.1007/bf02576766>
2. Glowinski, R., Pan, T. W., Hesla, T. I., Joseph, D. D., Périaux, J. (2001). A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow past Moving Rigid Bodies: Application to Particulate Flow. *Journal of Computational Physics*, 169 (2), 363–426. doi: <https://doi.org/10.1006/jcph.2000.6542>
3. He, Q., Glowinski, R., Wang, X.-P. (2018). A least-squares/fictitious domain method for incompressible viscous flow around obstacles with Navier slip boundary condition. *Journal of Computational Physics*, 366, 281–297. doi: <https://doi.org/10.1016/j.jcp.2018.04.013>
4. Court, S. (2019). A fictitious domain approach for a mixed finite element method solving the two-phase Stokes problem with surface tension forces. *Journal of Computational and Applied Mathematics*, 359, 30–54. doi: <https://doi.org/10.1016/j.cam.2019.03.029>
5. He, Q., Huang, J., Shi, X., Wang, X.-P., Bi, C. (2017). Numerical simulation of 2D unsteady shear-thinning non-Newtonian incompressible fluid in screw extruder with fictitious domain method. *Computers & Mathematics with Applications*, 73 (1), 109–121. doi: <https://doi.org/10.1016/j.camwa.2016.11.005>
6. Xia, Y., Yu, Z., Deng, J. (2019). A fictitious domain method for particulate flows of arbitrary density ratio. *Computers & Fluids*, 193, 104293. doi: <https://doi.org/10.1016/j.compfluid.2019.104293>
7. Fournié, M., Morrison, J. (2017). Fictitious domain for stabilization of fluid-structure interaction. *IFAC-PapersOnLine*, 50 (1), 12301–12306. doi: <https://doi.org/10.1016/j.ifacol.2017.08.2478>
8. Wang, Y., Jimack, P. K., Walkley, M. A. (2019). Energy analysis for the one-field fictitious domain method for fluid-structure interactions. *Applied Numerical Mathematics*, 140, 165–182. doi: <https://doi.org/10.1016/j.apnum.2019.02.003>
9. Wu, M., Peters, B., Rosemann, T., Kruggel-Emden, H. (2020). A forcing fictitious domain method to simulate fluid-particle interaction of particles with super-quadric shape. *Powder Technology*, 360, 264–277. doi: <https://doi.org/10.1016/j.powtec.2019.09.088>
10. Zhou, G. (2018). The fictitious domain method with H1-penalty for the Stokes problem with Dirichlet boundary condition. *Applied Numerical Mathematics*, 123, 1–21. doi: <https://doi.org/10.1016/j.apnum.2017.08.005>
11. Mottahedi, H. R., Anbarsooz, M., Passandideh-Fard, M. (2018). Application of a fictitious domain method in numerical simulation of an oscillating wave surge converter. *Renewable Energy*, 121, 133–145. doi: <https://doi.org/10.1016/j.renene.2018.01.021>
12. Wang, C., Sun, P. (2016). A Fictitious Domain Method with Distributed Lagrange Multiplier for Parabolic Problems With Moving Interfaces. *Journal of Scientific Computing*, 70 (2), 686–716. doi: <https://doi.org/10.1007/s10915-016-0262-1>
13. Sun, P., Wang, C. (2020). Distributed Lagrange multiplier/fictitious domain finite element method for Stokes/parabolic interface problems with jump coefficients. *Applied Numerical Mathematics*, 152, 199–220. doi: <https://doi.org/10.1016/j.apnum.2019.12.009>
14. Sun, P. (2019). Fictitious domain finite element method for Stokes/elliptic interface problems with jump coefficients. *Journal of Computational and Applied Mathematics*, 356, 81–97. doi: <https://doi.org/10.1016/j.cam.2019.01.030>

15. Temirbekov, A. N., Wjck, W. (2014). Numerical Implementation of the Fictitious Domain Method for Elliptic Equations. *International Journal of Electronics and Telecommunications*, 60 (3), 219–223. doi: <https://doi.org/10.2478/eletel-2014-0027>
16. Temirbekov, A. N. (2016). Numerical implementation of the method of fictitious domains for elliptic equations. *AIP Conference Proceedings*. doi: <https://doi.org/10.1063/1.4959667>
17. Heikkola, E., Rossi, T., Toivanen, J. (2003). A Parallel Fictitious Domain Method for the Three-Dimensional Helmholtz Equation. *SIAM Journal on Scientific Computing*, 24 (5), 1567–1588. doi: <https://doi.org/10.1137/s1064827500370305>
18. Yu, Z., Lin, Z., Shao, X., Wang, L.-P. (2016). A parallel fictitious domain method for the interface-resolved simulation of particle-laden flows and its application to the turbulent channel flow. *Engineering Applications of Computational Fluid Mechanics*, 10 (1), 160–170. doi: <https://doi.org/10.1080/19942060.2015.1092268>
19. Ruess, M., Varduhn, V., Rank, E., Yosibash, Z. (2012). A Parallel High-Order Fictitious Domain Approach for Biomechanical Applications. 2012 11th International Symposium on Parallel and Distributed Computing. doi: <https://doi.org/10.1109/ispdc.2012.45>
20. Üstoglu Ünal, V. (2013). Parallel numerical solution of incompressible full Navier-Stokes in 3-D. *International Journal of Numerical Methods for Heat & Fluid Flow*, 23 (4), 553–574. doi: <https://doi.org/10.1108/09615531311323737>
21. Yin, Z., Yuan, L., Tang, T. (2005). A new parallel strategy for two-dimensional incompressible flow simulations using pseudo-spectral methods. *Journal of Computational Physics*, 210 (1), 325–341. doi: <https://doi.org/10.1016/j.jcp.2005.04.010>
22. Passoni, G., Alfonsi, G., Tula, G., Cardu, U. (1999). A wavenumber parallel computational code for the numerical integration of the Navier-Stokes equations. *Parallel Computing*, 25 (5), 593–611. doi: [https://doi.org/10.1016/s0167-8191\(99\)00003-4](https://doi.org/10.1016/s0167-8191(99)00003-4)
23. Altybay, A., Ruzhansky, M., Tokmagambetov, N. (2020). A parallel hybrid implementation of the 2D acoustic wave equation. *International Journal of Nonlinear Sciences and Numerical Simulation*, 21 (7-8), 821–827. doi: <https://doi.org/10.1515/ijnsns-2019-0227>
24. Ladyzhenskaya, O. (1969). *The Mathematical Theory of Viscous Incompressible Flow*. Gordon and Breach.
25. Syrochenko, V. P. (2001). Numerical modeling of convective flows of a viscous fluid in multiply connected regions. *Proceedings of the International Conference RDAMM-2001*, 554–562.