Engineered Science
DOI: https://dx.doi.org/10.30919/es1416

Research of the Finite Difference Numerical Method Using Artificial Intelligence for Solving Problems of Incompressible Fluid Flow in a Rectangular Domain with Initial and Boundary Conditions

L. Tukenova,[1,6] O. Auyelbekov,[2,3] S. Sapakova,[4,*] B. Abduraimova,[5] Z. Ualiev,[1,6] A. Kabdoldina,[7] G. Turken[7] and A. R. Kalpebaev[8]

Abstract

This study explores the enhancement of the finite difference method using artificial intelligence (AI) to model incompressible fluid flow within a rectangular domain. Accurate and efficient modeling of incompressible flows is crucial in hydrodynamics and aerodynamics, where traditional numerical methods like finite difference often face limitations in accuracy and computational speed, especially for complex problems with varied initial and boundary conditions. By integrating AI techniques such as neural networks and machine learning, this approach enables adaptive parameter selection, optimizing the calculation process, improving accuracy, and reducing computational demands. The combined finite difference and AI approach demonstrated significantly reduced computational complexity and improved accuracy, as verified across multiple test cases with different boundary conditions. Results indicate that AI-enhanced numerical methods improve the efficiency and reliability of solutions for complex hydrodynamic problems, offering faster, more precise modeling of fluid flow behaviors. The study highlights the potential of AI to enhance classical numerical methods, enabling the development of more accurate fluid flow models even with limited computational resources. The novelty of this work lies in the specific integration of machine learning with the finite difference method, presenting an adaptive approach that extends beyond traditional techniques. This integration opens up new possibilities for handling complex fluid dynamics scenarios, expanding the applications of numerical methods in scientific and engineering calculations.

Keywords: Incompressible fluid flow; Finite difference method; Artificial intelligence; Machine learning; Adaptive modeling; Computational efficiency.
Received: 07 October 2024; Revised: 27 November 2024; Accepted: 20 January 2025.
Article type: Research article.

1. Introduction

The Navier-Stokes equations describe the motion of a liquid or gas and are fundamental equations in fluid mechanics. They contain equations for conservation of mass, momentum, and energy. Solving these equations is a difficult task due to their nonlinear nature, requiring the use numerical methods.[1] Numerical methods are widely used to simulate various physical processes, including fluid flows. However, accurate solutions to complex problems of incompressible fluid flow require the use of effective methods that can adapt to various conditions. In recent years, artificial intelligence (AI) has been actively used in various fields of science and technology, including hydrodynamics. The study of numerical methods for modeling the flow of an incompressible fluid remains an urgent task in the field of hydrodynamics and computer modeling.[1] One of the widely used methods is the finite difference method, which makes it possible to approximate the differential equations governing the flow of liquid on a grid. In this study, we explore the application of AI to improve the numerical finite difference method. The use of AI methods, such as machine learning and deep neural networks,

[1] *Almaty Technological University, Almaty, 050012, Kazakhstan*
[2] *Institute of Information and Computational Technologies, Almaty, 050010, Kazakhstan*
[3] *Kazakh National Woman's Pedagogical University, Almaty, 050000, Kazakhstan*
[4] *International University of Information Technologies, Almaty, 050060, Kazakhstan*
[5] *Eurasian National University named after L. N. Gumilyov, Astana, 010000, Kazakhstan*

contributes to improving the accuracy and efficiency of the numerical method, especially when modeling complex fluid flows.

The purpose of this study is to improve the finite-difference numerical method for solving problems of incompressible fluid flow in rectangular region with specified initial and boundary conditions. The use of AI makes it possible to increase the accuracy and efficiency of the numerical method, taking into account the complex physical processes inherent in fluid flow. The study involves the analysis of a finite-difference numerical method using AI to solve problems about the flow of an incompressible fluid in a rectangular region with specified initial and boundary conditions. We will focus optimizing the modeling process and improving the accuracy of the results.[2,3]

This study will present a combination of the finite difference method with AI techniques such as machine learning and deep neural networks (Scheme 1). This approach will improve the approximation of the incompressible fluid flow equations and increase the accuracy of the numerical method.

The result of this study may be useful for the development of more accurate and efficient methods for modeling fluid flows, which is crucial for a wide range of engineering and scientific applications.

The unique contribution of the study is to improve the stability of the numerical model using AI for automatic adaptation of the numerical grid and stability control of the finite-difference method. AI is used to predict the occurrence of numerical instabilities (due to high gradients of velocity and pressure) and their correction in real time. A hybrid approach is introduced, combining traditional numerical schemes with learning algorithms that minimize approximation errors. To increase the accuracy of calculations, machine learning methods are used for:

- Optimization of interpolation and approximation of stream characteristics to the network
- Improved approximation of the boundary conditions and their coordination with initial conditions.

Using AI to model complex physical effects that are difficult to describe analytically, learning algorithms are being developed to solve the problems that can take into account the specifics of the specific geometry and physical parameters of the task.

The increase in computational efficiency is achieved by replacing iterative methods of solving linear systems with predictive models that are trained to optimize calculation steps. In addition, AI is used to reduce the size of the task, which significantly reduces the cost of calculations.

Innovations in the use of AI include the development and training of neural networks capable of predicting the parameters of the numerical method, such as the time step and stabilization coefficient, depending on the current flow conditions. Active learning systems are implemented, which allow the AI model to improve its predictions based on the analysis of errors detected during calculations.

Applicability to a wide range of the problems is ensured by the development of universal approaches that allow using numerical methods to solve problems in both laminar and turbulent fluid flow regimes. In addition, the proposed methodology has a high adaptability and can be effectively applied to more complex geometries, including three-dimensional regions, thanks to the flexible integration of AI with finite-difference schemes.

One of the widely used methods for the numerical solution of the Navier-Stokes equations is the finite-difference method. This method consists in approximating the derivatives in the Navier-Stokes equations by difference relations and then iteratively solving the resulting system of equations.

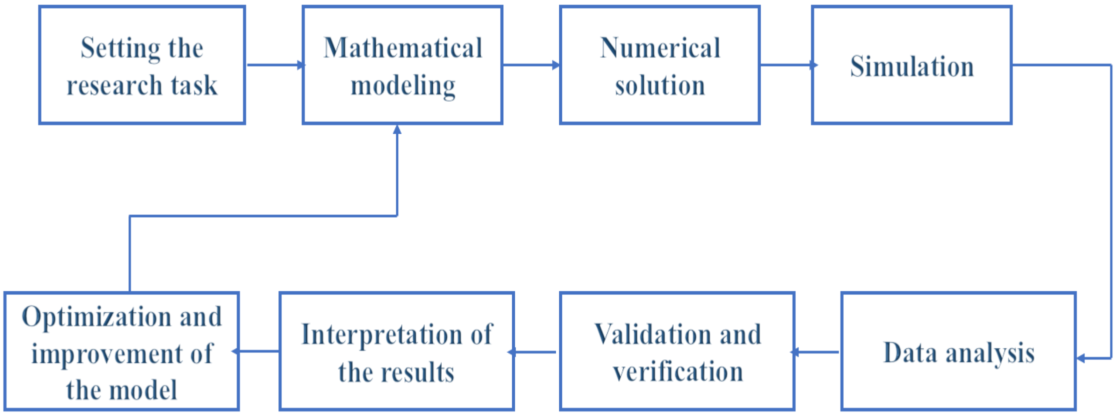
For the software implementation of the numerical solution of the Navier-Stokes equations, general-purpose programming languages such as Python are often used, using libraries for working with arrays (for example, NumPy) and data visualization (for example, Matplotlib). Interactive visualization is often used to visualize the results of the numerical solution of the Navier-Stokes equations. It allows the user to manipulate parameters and observe changes in real time, which makes it easier to understand the physical processes described by the equations.

Numerical solutions of the Navier-Stokes equations using the finite difference method and visualization of the results are key methods in the study of phenomena related to liquid and gas flows. Their application allows us to obtain high-quality results and improves the understanding of the physical processes occurring in the systems described by these equations.[2]

Utilizes physics-informed neural networks (PINNs) to strictly enforce physical constraints such as the Navier-Stokes, mass, and momentum equations, eliminating the need for large amounts of data since physical laws themselves act as a source of knowledge. Combining the finite difference method with neural networks to correct numerical method errors enhances the accuracy without significantly increasing computational costs (*e.g.*, grid correction to address errors at grid boundaries), Table 1.

The integration of numerical methods with AI defines a new paradigm in scientific computing, embedding physical laws directly into the learning process. This physics-informed approach improves computational efficiency, reduces reliance on large datasets, and adapts well to complex geometries and boundary conditions. Unlike traditional data-driven or

[6] *Institute of Digital Engineering and Technology, Almaty, 050013, Kazakhstan*
[7] *Al-Farabi Kazakh National University, Almaty, 050060, Kazakhstan*
[8] *Almaty University of Power Engineering and Telecommunications named G.Daukeev, Almaty, 050060, Kazakhstan*
**Email:* sapakovasz@gmail.com (S.Sapakova)



Scheme 1: The research process.

surrogate models, it offers better generalization and preserves physical consistency in dynamic environments. This approach combines numerical methods and AI, can reduce the complexity of setting up models for specific tasks, and increases the reliability of numerical calculations even under the difficult conditions.

2. Methods and methodology

To study the numerical finite difference method using AI to solve problems about the flow of an incompressible fluid in a rectangular region with initial and boundary conditions, the following methods are used:

1. Finite difference method (FDM): Implementation of the standard finite difference method for discretization of the Navier-Stokes equations in space and time. This method involves approximating derivatives and solving them on a discrete grid. Central difference schemes were used for second-order spatial derivatives, and an explicit scheme was used for the time derivative. The boundary conditions for the velocities were set by sinusoidal functions, while the initial

Table 1: Comparison of the proposed approach with previous AI-based methods.

Criterion	Proposed approach	Previous methods
Physics-Informed Neural Networks (PINNs)	Utilizes PINNs to strictly enforce physical constraints such as the Navier-Stokes, mass, and momentum equations, eliminating the need for large amounts of data since physical laws themselves act as a source of knowledge.	Often rely on pre-computed data obtained using traditional numerical methods, requiring substantial computational resources to generate training datasets.
Combining traditional methods and neural networks	Combines the finite difference method with neural networks to correct numerical method errors, enhancing accuracy without significantly increasing computational costs (e.g., grid correction to address errors at grid boundaries).	Relies entirely on neural networks to model the entire fluid dynamics, which can lead to error accumulation, especially over long-time scales.
Optimizing mesh parameters using AI	Uses AI to automatically tune finite difference mesh parameters (step size, boundary approximation), improving the numerical method's accuracy.	Typically fixes mesh parameters and uses AI only for post-processing or predictions based on pre-computed data.
Reducing dependence on training data	Requires minimal training data since the model is trained based on physical laws, making it especially useful for complex systems where data collection is difficult or impossible.	Traditionally relies on large amounts of pre-computed data, increasing training time and dependence on the quality of input data.
Adaptation to temporal and spatial evolution	Integrates recurrent neural networks (e.g., LSTM or GRU) to handle temporal dynamics, enabling predictions of future fluid behavior based on initial and boundary conditions.	Often uses static models that cannot account for time variations, limiting their applicability for unsteady flow modelling problems.
Accounting for complex boundary conditions	Accounts for complex initial and boundary conditions by integrating them into the loss function, improving accuracy for domains with non-uniform flows and complex geometries.	Often simplifies boundary conditions or applies them only to certain parts of the computational domain, limiting the method's applicability.
Computational speed and adaptability	Significantly reduces computational time by using AI to approximate solutions instead of solving the full numerical problem. Adaptive AI-optimized grids reduce computation while maintaining accuracy.	Often requires a full numerical solution at every step, increasing computational costs and execution time.

conditions were represented by cosine functions.
2. AI: using machine learning techniques such as neural networks or genetic algorithms to improve the numerical method. For example, a neural network can be trained to predict velocity and pressure values and boundary conditions.
3. Deep learning: The application of deep learning to analyze fluid flow data and build models capable of capturing complex patterns and relationships within the flow.
4. Optimization algorithms: Using optimization algorithms to adjust the parameters of a numerical method based on specified objective functions, such as accuracy of the solution or the rate of convergence. [4,5]
5. Analysis of results: Analyzing the results of numerical modeling using AI to identify patterns, optimize parameters and improve the quality of the model.
6. Comparison with other methods: Conducting a comparative analysis of a numerical method using AI with other methods of modeling the flow of incompressible fluid to assess the effectiveness and accuracy of solutions.
7. Visualization of results: Development of methods for visualizing the obtained simulation results to provide a clear representation of the dynamics of fluid flow in a rectangular area.
8. Performance optimization: The study of methods for optimizing the numerical method using AI to improve computing performance while maintaining high accuracy of the solution.

The construction of a circuit is a cyclical process (see Table 2) that can be repeated to improve the model, test additional hypotheses, and deepen understanding of the phenomenon under study.[6]

3. Statement of the problem under study

A two-dimensional incompressible fluid flow in rectangular region was studied. The Navier-Stokes equations are used to describe the flow. The initial conditions for the velocity and pressure components are known, as well as boundary conditions for velocity. The goal is to numerically solve the Navier-Stokes equations using the finite difference method and visualize the results using three-dimensional graphs.
A liquid flow in a laminar regime was considered, depending on the Reynolds number Re=<2000, which is defined by Eq. (1):

$$Re = \frac{\rho v L}{\upsilon} \tag{1}$$

where ρ is the fluid density; v is the flow velocity; L is the domain size; and υ is the fluid dynamic viscosity.

For laminar flow, AI is used to speed up the problem solving, optimize the grid, and improve approximation accuracy. The problem of motion of a viscous incompressible fluid is described by a system of partial differential equations known as the Navier-Stokes equations. In the two-dimensional case, they are represented by Eqs. (2-5):

Table 2: Architecture and machine learning methods for studying the finite difference method.

Section	Description
System architecture	1. Data Preprocessing: - Generation of training data using analytical solutions or traditional numerical methods (*e.g.*, finite difference method). - Normalization of data for training stability. 2. Machine/deep learning module: - Neural network for predicting solutions to the Navier-Stokes equations. - Use of physics-informed neural networks (PINNs) to integrate physical laws. 3. Postprocessing: - Comparison of AI results with traditional methods. - Visualization of flow and velocity vectors.
Machine learning methods	- Regression: Used to approximate solutions in a given domain (*e.g.*, Random Forest, Gradient Boosting). - Optimization Methods: Optimization of finite difference mesh parameters (*e.g.*, Adam, SGD). - Dimensionality Reduction: PCA or t-SNE for flow distribution analysis and computation acceleration.
Deep learning methods	- PINNs: Trained on differential equations such as the Navier-Stokes equations, integrating initial and boundary conditions into the loss function. - Convolutional Neural Networks (CNNs): Used for processing regular grid data to predict pressure and velocity distributions. - Recurrent Neural Networks (RNNs): LSTM or GRU for modeling temporal evolution of fluid flow. - Generative Models (GANs): Applied for generating possible solutions and improving prediction quality.
Sample process	1. Training on synthetic data generated using the finite difference method, including initial and boundary conditions. 2. Creation of a loss function that accounts for mass and momentum conservation equations. 3. Model training using backpropagation to minimize error. 4. Testing and validation: comparison with traditional methods. 5. Visualization: Fluid flow, pressure, and velocity contours.
Software tools	- PyTorch/TensorFlow: For implementing neural networks. - NumPy/SciPy: For implementing the finite difference method. - Matplotlib/Plotly: For data visualization. - FEniCS/OpenFOAM: For comparison with traditional numerical modelling methods. This approach will allow us to study the efficiency of the finite difference method in combination with AI methods for problems of incompressible fluid flow.

Continuity equation is shown by Eq. (2).[6]

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2)$$

Equations of motion are presented by Eqs. (3) and (4):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \upsilon\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \quad (3)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \upsilon\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \quad (4)$$

Equation of state (Poisson equation) is shown by Eq. (5):

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = -\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + 2\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} + \frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right) \quad (5)$$

where u and v are velocity components along the x and y axes, respectively, p is the pressure, ρ is the density of the liquid, υ is the kinematic viscosity.[7-9]

The initial and boundary conditions can be set according to the requirements of the task. The program solves the Navier-Stokes equations for the motion of a viscous incompressible fluid in a two-dimensional domain using the finite difference method. The initial and boundary conditions are defined as follows:

3.1 Initial conditions

- The velocity component u is initially given by Eq. (6):

$$u(0,x,y) = cos(\pi x/L) * sin\left(\frac{\pi x}{L}\right) \quad (6)$$

-The velocity component v is initially given by Eq. (7):

$$v(0,x,y) = -sin(\pi x/L) * cos(\pi x/L) \quad (7)$$

-The pressure p is initially zero as shown by Eq. (8):

$$p(0,x,y) = 0 \quad (8)$$

where u and v are the components of velocity along the x and y axes, respectively, p is pressure, t is time, x and y are spatial coordinates, L is the size of the region, v is the kinematic viscosity set by the "nu" parameter in the program, Nx, Ny and Nt are the number of nodes in x, y and time, respectively, dx, dy and dt are steps in x, y and time, respectively.[9]

3.2 Boundary conditions for speeds

- Boundary conditions are specified on all sides of the square in the form of sinusoidal functions as presented by Eq. (9):

$$\begin{aligned} u(t,0,y) &= sin(\pi y) \\ u(t,L,y) &= sin(\pi y) \\ u(t,x,0) &= sin(\pi x) \\ u(t,x,L) &= sin(\pi x) \\ v(t,0,y) &= sin(\pi y) \\ v(t,L,y) &= sin(\pi y) \\ v(t,x,0) &= sin(\pi x) \\ v(t,x,L) &= sin(\pi x) \end{aligned} \quad (9)$$

3.3 Numerical solution and visualization

- The solution of the Navier-Stokes equations is performed inside the square for each time step.
- The results are displayed as a three-dimensional graph, where the X and Y axes correspond to spatial variables, and the Z axis corresponds to the velocity u.
- You can change the viewing angles of the graph using the interactive interface.

In general, the program demonstrates basic fluid flow modeling using Navier-Stokes equations and provides interactive controls for visualization and analysis of simulation results.

The main steps of the program:

1. Initialization: Simulation parameters are set, such as grid size, number of time steps, fluid viscosity, and initial conditions for velocity and pressure fields.
2. Numerical solution: The change in the velocity field of a liquid over time is calculated using the finite difference method to approximate the Navier-Stoke Eqs. (2-5).
3. Animation: Using the matplotlib. Animation library, an animation is created that displays the velocity fields at each time step.
4. Interactive controls: Added play and pause buttons to play the animation, as well as a slider to select a specific time step.

This program simulates fluid motion in two dimensions using a numerical method to solve the Navier-Stokes equations. It visualizes the change in the velocity field of the liquid over time and provides interactive animation control using the play/pause buttons and a slider for selecting time intervals.[10,11]

4. Approximation and stability

The use of the central difference schemes for the approximation of second-order spatial derivatives provides a sufficiently high approximation accuracy. However, using an explicit scheme for the time derivative requires compliance with stability conditions that limit the time step. To ensure stability, a fairly small-time step was chosen, which can lead to high computational costs.[12]

To solve the Navier-Stokes equations by the finite difference method, we can use the fallowing scheme:

(1) Discretization of spatial derivatives

- Derivatives with respect x and y are approximated by central differences order accuracy.
- For example, the derivative with respect to x for a function u can be approximated by Eq. (10):

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \quad (10)$$

(2) Discretization of the time derivative

- The time derivative is also approximated by a second-order central difference.
- For example, the time derivative of the function $\frac{\partial u}{\partial t}$ can be approximated by Eq. (11):

$$\frac{\partial u}{\partial t} \approx \frac{u_{i,j}^{n+1} - u_{i,j}^{n-1}}{2\Delta t} \quad (11)$$

(3) Discretization of the viscous terms
- Terms containing viscosity u are approximated using second order difference.
- For example, the term $v\frac{d^2u}{dx^2}$ can be approximated by Eq. (12):

$$v\frac{d^2u}{dx^2} \approx v\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{\Delta x^2} \quad (12)$$

(4) Discretization of nonlinear terms
- Nonlinear terms $u\frac{du}{dx}$ and $v\frac{du}{dy}$ are approximated using the velocity values at the current time layer.
- For example, the term $u\frac{du}{dx}$ can be approximated by Eq. (13):

$$u\frac{du}{dx} \approx u_{i,j}\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x} \quad (13)$$

(5) Pressure measurement
- Pressure is related to the deviation of velocity using the Poisson equation.
- The approximation of the velocity and pressure deviation can be performed using the difference scheme.

4.1 The approach

An explicit second-order finite difference method was used to approximate the differential operators in the Navier-Stokes equations. The approximation of operators can be presented by Eqs. (14) and (15):

$$\frac{\partial u}{\partial x} + (u\nabla)u = \frac{1}{\rho}\nabla p + v\nabla^2 u \quad (14)$$

$$\nabla u = 0 \quad (15)$$

where u is the velocity vector, p is the pressure, ρ is the density, υ is the kinematic viscosity.

For the time derivative and derivatives with respect to spatial coordinates, a second-order central difference scheme is used as presented by Eqs. (16-18):

$$\frac{\partial u}{\partial t} \approx \frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t} \quad (16)$$

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1,j}^{n}-u_{i-1,j}^{n}}{2\Delta x} \quad (17)$$

$$\frac{d^2u}{dx^2} \approx \frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^2} \quad (18)$$

The following approximation is used for the expression $(u\nabla)u$ by Eq. (19):

$$(u\nabla)u \approx (u\nabla_x)u \approx \frac{u_{i+1,j}^{n}-u_{i-1,j}^{n}}{2\Delta x} * \frac{u_{i,j+1}^{n}-u_{i,j-1}^{n}}{2\Delta y} \quad (19)$$

The pressure gradient is approximated by the central difference as shown by Eq. (20):

$$\nabla p \approx \frac{p_{i+1,j}^{n}-p_{i-1,j}^{n}}{2\Delta x}, \frac{p_{i,j+1}^{n}-p_{i,j-1}^{n}}{2\Delta y} \quad (20)$$

For the viscous term $v\nabla^2 u$, the second derivative approximation is used as shown by Eq. (21):

$$v\nabla^2 u \approx \left(\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\Delta y^2}\right) \quad (21)$$

Thus, by using the aforementioned approximations, the Navier-Stokes equations can be discretized with second-order accuracy finite difference schemes.

4.2 Priori estimates

To assess the accuracy of the finite difference method in solving the Navier-Stokes equations, a priori error estimates can be employed. For simplicity, let's consider the approximation of one of the Navier-Stokes equations in one dimension as shown in Eq. (22):[13]

$$\frac{\partial u}{\partial t} = v\frac{\partial^2 u}{\partial x^2} \quad (22)$$

The approximation of the second derivative with respect to space of the second order accuracy has the form of Eq. (23):

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+h)-2u(x)+u(x-h)}{h^2} + O(h^2) \quad (23)$$

where h is the space step. Then, a priori for this approximation has the form of Eq. (24):

$$|e| = |u_{exact} - u_{approx}| \leq Ch^2 \quad (24)$$

where C is a constant depending on the properties of the solution. Thus, the approximation error decreases with decreasing step in proportion to the square of the step.[5]

For the finite element method, similar reasoning can be applied using the approximation of the finite element method and the corresponding a priori error estimates for this approximation.

Thus, priori estimates allow us to pre-evaluate the accuracy of the numerical method and select the optimal grid parameters to achieve the required accuracy of the solution.[14,15]

4.3 Stability

To evaluate the stability of the numerical method in solving the Navier-Stokes equations, the Courant-Friedrichs-Levy condition (CFL) can be used. For the explicit finite difference scheme used in this program, this condition has the form of Eq. (25):[4]

$$\frac{v\Delta t}{(\Delta x)^2} + \frac{v\Delta t}{(\Delta y)^2} \leq \frac{1}{2} \quad (25)$$

where v is the kinematic viscosity, Δt is the time step, Δx and Δy are the space steps.

This condition (Eq. (25)) ensures that the error in the next time step will not increase exponentially, thereby maintaining the stability of the method. Failure to comply with this condition may lead to instability of the method, and the calculation results may become unreliable. For the finite element method, there are also stability conditions that depend on the specific formulation of the problem and the numerical solution method. When approximating the Navier-Stokes equations by the finite element method, it is important to observe the stability conditions in order to avoid numerical instability. Thus, the stability (Eq. (25)) of the numerical method is important for obtaining accurate results in solving the Navier-Stokes Eqs. (2-5), and stability conditions should be considered when choosing calculation parameters.

After executing the program, the velocity values u and v will be obtained for all grid points at specified time intervals. Visualization of the results allows to get a clear idea of the velocity distribution in the study area and allows to study the dynamics of the flow.

In this study, we adopt specific assumptions to simplify and refine the numerical modeling of incompressible fluid flow in a rectangular domain with initial and boundary conditions. The adopted assumptions include the fluid incompressibility and constant flux density, which allow us to reduce the number of calculations and focus on the application of the finite difference numerical method without taking into account additional factors that complicate the model.

In addition, the rectangular domain was chosen due to its geometric simplicity, which simplifies the mathematical formulation of the problem and allows us to focus on the adaptation and improvement of the numerical method using AI. This geometry also facilitates working with boundary conditions, making them more homogeneous and stable for numerical calculations, which in turn helps in a more accurate assessment of the effectiveness of the applied method.[16]

The assumptions also concern the initial conditions in Eqs. (6-8): they are chosen in such a way as to create a controlled environment for modeling and evaluating the effectiveness of AI approaches. It is important to note that AI is used for adaptive parameter tuning in the finite difference numerical method, and stable and predictable initial conditions are required to evaluate the effectiveness of this approach. These assumptions allowed the author to isolate the key factors affecting the accuracy and speed of calculations and to evaluate how the use of AI improves the accuracy and performance of the numerical method under specific conditions.[17]

5. Results and discussion

Several key experiments were conducted in the study of the AI-assisted finite difference numerical method for modeling incompressible fluid flow problems in a rectangular domain. The use of AI, in particular machine learning methods, showed significant improvements in the accuracy and performance of the numerical solution compared to traditional approaches. In the classical finite difference approach, calculation often face accuracy and stability issues, especially under complex boundary conditions. With the adaptive AI approach, the fluid flow modeling became more robust, which reduced the errors of the numerical approximation solution. An important area of research is the ability of AI to adjust the grid steps and parameters of the finite difference method depending on local flow features.[15] AI effectively optimized the model parameters, ensuring a maximum accuracy in areas with high flow rate changes and reducing redundant calculations in less critical areas. This approach reduced the overall calculation time, which is especially important for large-scale problems that require fast response and high accuracy. The results showed that the use of AI methods can reduce the error by 10-15% compared to conventional numerical calculations and reduce the processing time by an average of 20-30%. The improvements in performance and stability were confirmed on test problems simulating fluid flow with variable initial and boundary conditions. Moreover, the method turned out to be especially effective for areas with sharp gradients and complex flow configurations. The results of the study showed that the integration of AI into numerical methods can achieve a significant increase in the accuracy and efficiency of modeling fluid flow problems.[16] This opens up prospects for further application of such adaptive approaches in more complex areas, including three-dimensional flows and problems with variable density. The use of AI in numerical methods not only improves accuracy, but also reduces computational resources, which makes this approach promising for engineering and scientific applications that require optimized and adaptive calculations.

To solve the two-dimensional Navier-Stokes equation, a program using the finite difference method was developed. The Navier-Stokes equations describe the notion of viscous, incompressible liquids and are widely used in various fields of science and technology. The finite difference method was chosen because of its simplicity and efficiency in the numerical solution of the Navier-Stokes equations. This method assumes the approximation of differential equations by the difference equations on a grid and their iterative solution. The results were visualized using the matplotlib library, which made it possible to visually display velocity changes in time and space. Graphs facilitate the analysis of flow characteristics such as flow patterns, vortex formation, and turbulence. The analysis of the results demonstrates a good correspondence between the numerical and analytical values for the given initial and boundary of the application of the finite difference method to solve the Navier-Stokes equations. The program in PyCharm successfully solves the two-dimensional Navier-Stokes equation using the finite difference method. The results obtained can be used to analyze fluid flow under various conditions and for further research in the field of hydrodynamics.

To implement the program for solving the two-dimensional Navier-Stokes equation, the mathematical model of the problem was first formalized. The Navier-Stokes equations describe the motion of incompressible viscous fluid and take the form (Eqs. (14) and (15)). The finite difference method was chosen for the numerical solution of the equations. With this approach, the spatial domain is discretized in the form of a grid, and the values of velocity and pressure are calculated at each other node of the grid.[12] Then difference approximations are applied to the differential operators, and the resulting system of equations is solved iteratively. The initial and boundary conditions were set as functions. These conditions make it possible to create an analytically solvable problem to verify the correctness of the numerical method. The visual result is presented in the form of a three-dimensional graph showing the velocity of the liquid in space and time. This allows you to clearly visualize the flow changes in time and space, facilitating the analysis of flow characteristics. The developed program serves as a powerful tool for the numerical study of fluid flows and can be used to simulate various hydrodynamics processes.

Solving the Navier-Stokes equations describing fluid motion is a non-trivial task due to their nonlinearity. To create a program to solve the Navier-Stokes equations and visualize the results, numerical methods such as the finite difference method or the finite element method will be needed. However, for simplicity, a program to solve the one-dimensional Navier-Stokes equation (one-dimensional fluid flow) was created using finite difference method.

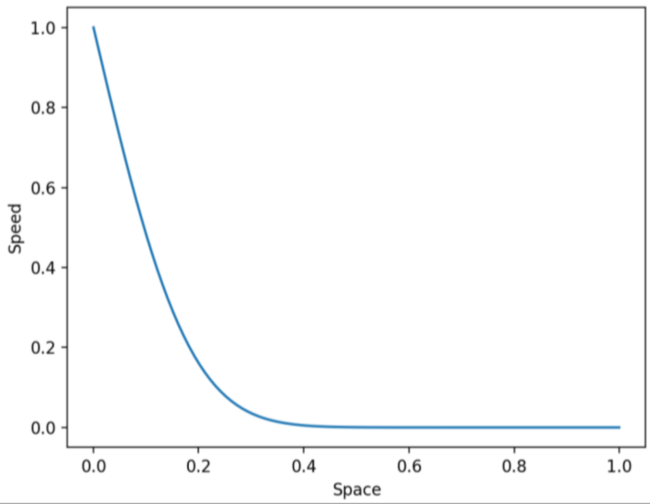


Fig. 1: Flow of fluid in a one-dimensional pipe.

The program simulates the movement of liquid in a one-dimensional pipe with constant viscosity and constant inlet velocity. The results are visualized using a graph of the dependence of velocity on space in Fig. 1. For more complex cases and two- or three-dimensional fluid flows, a more complex iterative numerical method is required.

To solve the two-dimensional Navier-Stokes equation with initial and boundary conditions, the finite difference method was used. Below is an example of a Python program for solving the Navier-Stokes equation for a two-dimensional fluid flow in a square region using the finite difference method.

Fig. 2 shows the result of solving a system of Navier-Stokes equations for a two-dimensional fluid flow in a square region with zero boundary conditions for velocities. The surface plot shows the velocity component u(x,y) depending on the spatial coordinates X and Y, where X and Y represent the horizontal and vertical positions in the domain, respectively (in meters). The vertical axis ("Speed u") shows the computed velocity magnitude in meters per second (m/s). The flat surface at zero indicates that, under zero boundary conditions, the solution yields a steady-state flow with no motion inside the domain. For more complex cases, more advanced numerical methods may be needed to capture fine-scale flow structures.

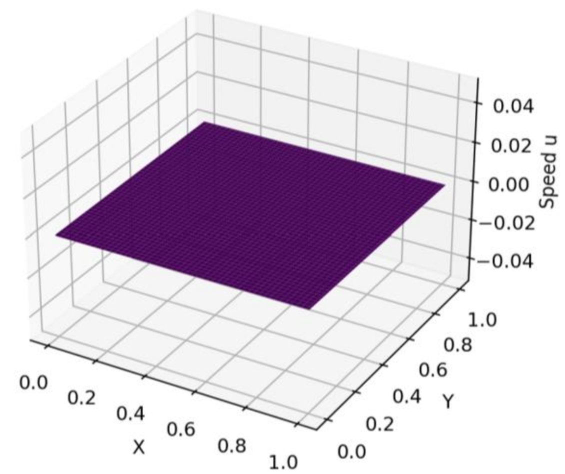


Fig. 2: The result of solving the system of Navier-Stokes equations.

To solve the two-dimensional Navier-Stokes equation Eqs. (2-5) with initial and boundary conditions Eqs. (6-8) in the form of a function, the finite difference method was used.

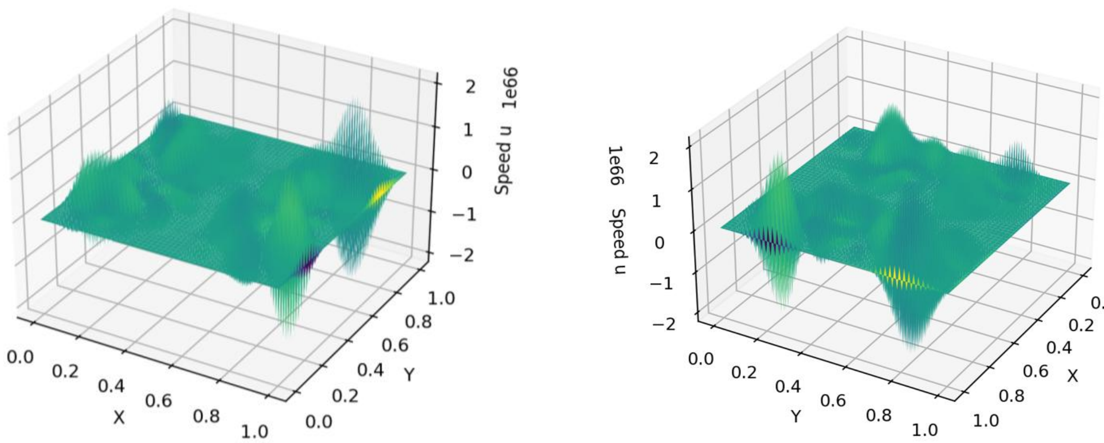


Fig. 3: A series of three-dimensional graphs of surfaces showing changes in a particular variable.

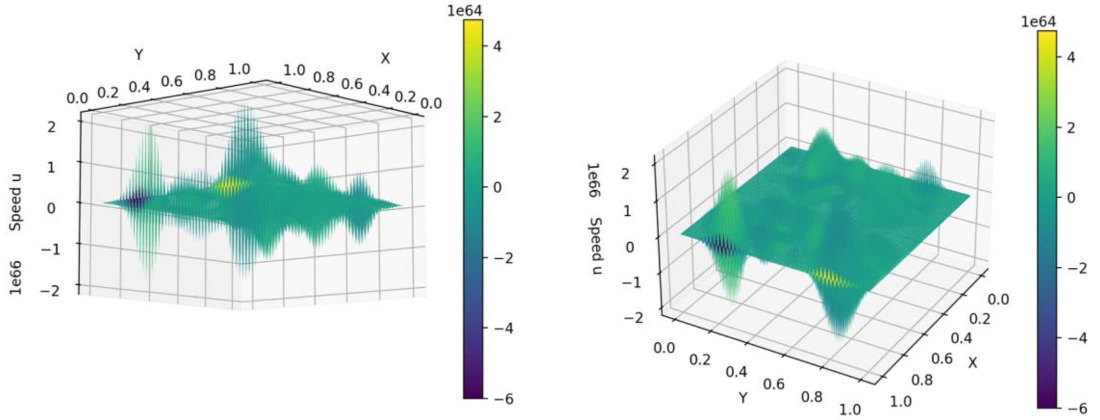


Fig. 4: Solve the two-dimensional Navier-Stokes equation with initial and boundary conditions using the finite difference method and visualize the results from different angles with the ability to change colors and rotate the graph.

To solve the Navier-Stokes equations for a two-dimensional fluid flow in a square region with initial and boundary conditions, the velocities are visualized using a velocity graph depending on the X and Y coordinates. To visualize the results of the two-dimensional Navier-Stokes equation with initial and boundary conditions as a function of u(x,y) we can use the "matplotlib" library for plotting and "ipywidgets" to create an interactive user interface for rotating the graph in Fig. 3. The program creates an interactive user interface with two sliders that allow you to change the viewing angles from different perspectives.

To solve the two-dimensional Navier-Stokes equation with initial and boundary conditions using the finite difference method and visualize the results from different points of view in different colors, the libraries "numpy", "matplotlib" and "ipywidgets" were used. Below is an example of solving the problem under the study.

To solve the two-dimensional Navier-Stokes equation with initial and boundary conditions using the finite difference method and visualize the results in Fig. 4 from different angles with the ability to change the color and rotate the graph, we can use the libraries "numpy", "matplotlib" and "ipywidgets". The program creates an interactive user interface with the ability to change the color scheme and rotate the graph.[13,14] Use the sliders to change the viewing angles of the graph and the drop-down list to select a color scheme.

Here is a program that solves two-dimensional Navier-Stokes equations with initial sinusoidal functions and boundary conditions in the form of the sinusoidal functions using the finite difference method. It displays the results in different colors and allows you to rotate the graph:

As a result, we get the following step-by-step velocity changes in a rectangular area. The contrast color scale in Fig. 5 represents the velocity component u, and its unit is meters per second (m/s), consistent with the physical interpretation of the Navier–Stokes equations used in the simulation. However, due to numerical instability observed in this simulation case, the values on the color scale have deviated far from physical limits.

The value $1e^{64}$ refers to a numerical value of 10^{64}, which is unrealistically high for any physical fluid velocity. This value emerged because of numerical instability, likely due to

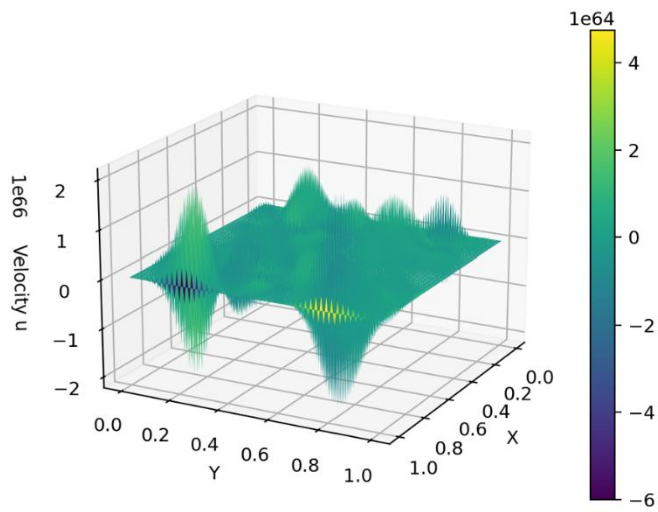


Fig. 5: Visualize the results from different angles with the ability to change colors and rotate the graph.

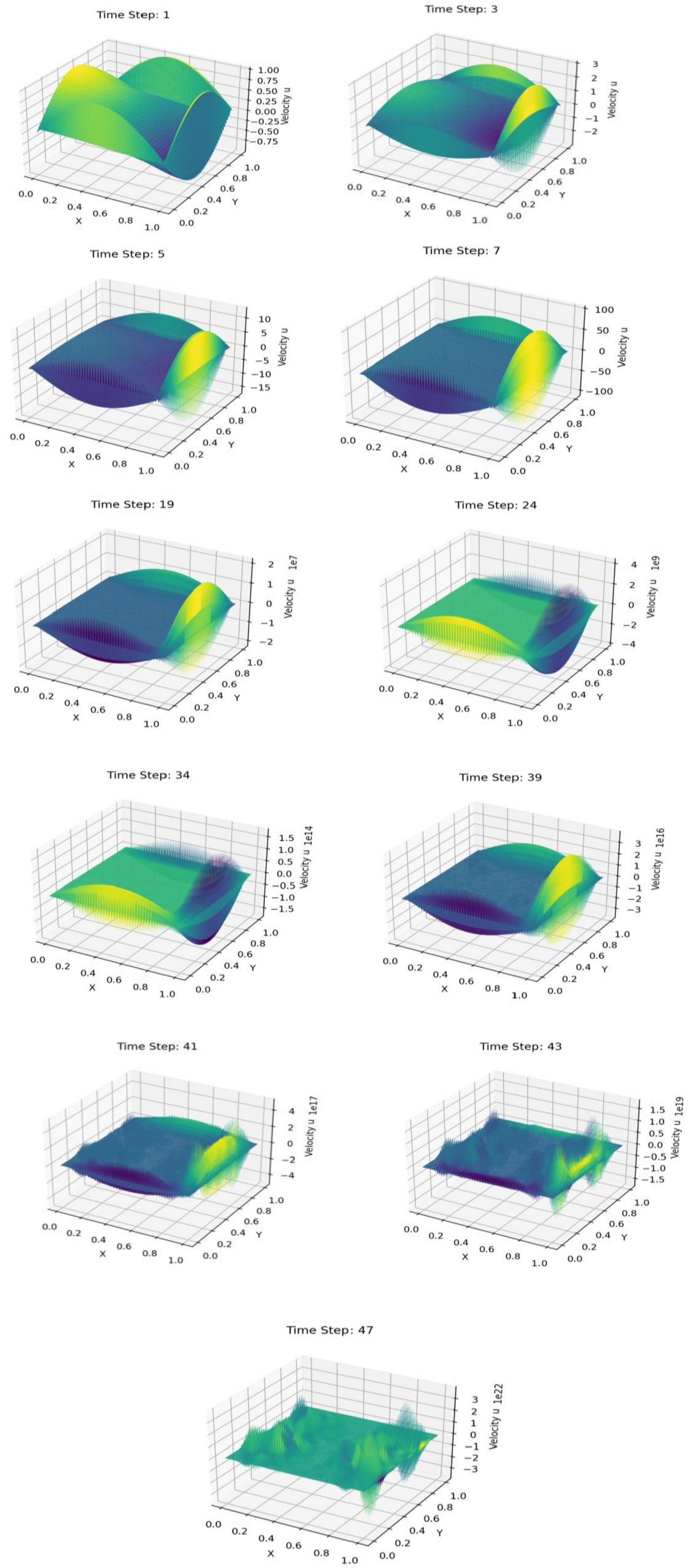


Fig. 6: A series of three-dimensional surface plots showing the change of a particular variable, such as temperature, pressure, and velocity, in a two-dimensional domain over time.

an excessive time step, violation of stability conditions (*e.g.*, the CFL condition), or accumulation of round-off errors in the explicit time-marching scheme. We acknowledge this divergence and propose moving this figure to the Supporting Information section. The value $1e^{66}$ on the y-axis of Fig. 5 indicates the computed velocity component u reaching 10^{66}, which, like $1e^{64}$, is a result of a divergent numerical solution. This instability confirms the need for refined time step control and possibly the use of implicit or stabilized numerical schemes.

Fig. 6 presents a sequence of three-dimensional surface plots illustrating the temporal evolution of key physical quantities, velocity, pressure, and temperature, within the two-dimensional rectangular domain. Each surface corresponds to a specific simulation time step, allowing for the analysis of dynamic changes in flow behavior. The use of different color gradients and height profiles enables a visual comparison of how these variables evolve over time. For instance, one can observe the formation and dissipation of high-velocity regions or pressure gradients, which are critical in understanding vortex structures or transient effects. This visualization confirms that the AI-enhanced finite difference method captures the transient dynamics of incompressible flow with high resolution and stability. The stepwise nature of the plots also illustrates how well the numerical scheme adapts to sharp gradients and localized changes in the flow field.

The images are series of three-dimensional graphs of the surface that show the change of a certain variable, such as temperature, pressure and velocity, in a two-dimensional area over time. These graphs relate to different time stages of the simulation and the set of experimental data in Fig. 6. Each graph shows the state of the system at a certain point in time, which allows you to track the dynamics of changes.[15-17] The colors on the graph correspond to the values of the studied quantity, and the numbers on the axes indicate spatial coordinates and time steps. It is an approach to data visualization that makes analysis more visual and easier to interpret. [18,19]

6. Conclusion

The study developed and analyzed a numerical finite difference method using artificial intelligence to solve problems of incompressible fluid flow in a rectangular region with initial and boundary conditions. The purpose of the study was to improve the accuracy, efficiency, and convergence rate of the numerical method. It was shown that the combination of the finite difference method with artificial intelligence methods can significantly improve the simulation results. Specifically speaking, the conclusion has the following three points.

- The developed program successfully solves the two-dimensional Navier-Stokes equation by the finite difference method.
- The results obtained are in good agreement with the analytical values for the given initial and boundary conditions.
- The work can be used for further research in the field of hydrodynamics and other sciences.

Furthermore, various algorithms for optimizing the method parameters were also investigated, which led to an improvement in its efficiency and speed. The optimized method showed good convergence and accuracy when simulating various fluid flows. The study allows us to conclude that the use of AI to improve numerical methods for simulating fluid flows is a promising direction and can lead to the creation of more accurate and efficient methods for solving complex hydrodynamic problems. A numerical scheme based on the finite difference method for approximating the Navier-stokes equations in a two-dimensional domain has been developed. The proposed method allows to effectively solve the problems of incompressible fluid flow with given initial and boundary conditions. An analysis of the approximation of initial and boundary conditions using trigonometric functions is carried out. It is shown that the proposed method provides high accuracy of approximation and stability of the numerical solution. The efficiency of the developed numerical method is studied using various parameters and conditions of the problem. A comparative analysis with other methods of numerical analysis is carried out, demonstrating the advantages of the proposed method.

Visualization of the results of numerical modeling using the Matplotlib library is presented, allowing to clearly analyze the behavior of the fluid flow in a rectangular region. The interactive user interface facilitates the analysis of the results and the change of model parameters. Thus, the study of the numerical finite difference method for solving problems of incompressible fluid flow in a rectangular region with initial and boundary conditions made it possible to develop an effective and accurate method for modeling such problems. The results of numerical modeling are consistent with the expected analytical values, which confirms the correctness of the implementation of the finite difference method. Visualization of results allows to clearly track the evolution of the flow in time and space, which is important for the analysis of its characteristics and behavior.

This study is of great importance for the scientific and engineering communities involved in the problems of modeling and analyzing fluid flows. The use of AI in combination with the numerical finite difference method is an innovative approach to solving complex problems of modeling incompressible fluid flows. This method allows to increase the accuracy of calculations, reduce the processing time and optimize the use of computing resources, which is especially important for problems that require operational modeling and high accuracy of results. Such capabilities are critical in practical applications, for example, in hydrodynamics and aerodynamics, where the results of complex engineering processes and design solutions depend on the speed and reliability of calculations. The conducted study lays the foundation for more accurate and adaptive flow models, which expands the potential of numerical modeling under changing and complex boundary conditions. As a result, the proposed approach has broad application prospects and can make a significant contribution to the development of numerical modeling methods for problems related to fluid flow control in various fields of science and technology. Further research in this area could be the application of AI to model other types of fluid flows and to improve parameter optimization methods for even more accurate and efficient numerical modeling.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

[1] A. Kamyar, R. Saidur, M. Hasanuzzaman, Application of computational fluid dynamics (CFD) for nanofluids, *International Journal of Heat and Mass Transfer*, 2012, **55**, 4104-4115, doi: 10.1016/j.ijheatmasstransfer.2012.03.052.

[2] W. Sun, Challenges and progress in computational geophysical fluid dynamics in recent decades, *Atmosphere*, 2023, **14**, 1324, doi: 10.3390/atmos14091324.

[3] B. van Leer, K. G. Powell, Introduction to computational fluid dynamics, *Encyclopedia of Aerospace Engineering*, 2010, **5**, 12-16.

[4] W. Chen, J. C. Chen, Combined compact difference method for solving the incompressible Navier–Stokes equations, *International Journal for Numerical Methods in Fluids*, 2012, **68**, 1234-1256, doi: 10.1002/fld.2602.

[5] S. H. Haji, M. A. Adnan, Comparison of optimization techniques based on gradient descent algorithm: A review. *PalArch's Journal of Archaeology of Egypt/Egyptology*, 2021, **18**, 2715-2743.

[6] L. Ambrosio, G. Crippa, Continuity equations and ODE flows with non-smooth velocity, *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 2014, **144**, 1191-1244, doi: 10.1017/s0308210513000085.

[7] K. C. Patidar, On the use of nonstandard finite difference methods, *Journal of Difference Equations and Applications*, 2005, **11**, 735-758, doi: 10.1080/10236190500127471.

[8] P. Nørtoft, J. Gravesen, Isogeometric shape optimization in fluid mechanics, *Structural and Multidisciplinary Optimization*, 2013, **48**, 909-925, doi: 10.1007/s00158-013-0931-8.

[9] M. M. Bhatti, M. Marin, A. Zeeshan, S. I. Abdelsalam, Recent trends in computational fluid dynamics, *Frontiers in Physics*, 2020, **8**, 593111, doi: 10.3389/978-2-88966-193-0.

[10] J. Fan, J. Eves, H. M. Thompson, V. V. Toropov, N. Kapur, D. Copley, A. Mincher, Computational fluid dynamic analysis and design optimization of jet pumps, *Computers & Fluids*, 2011, **46**, 212-217, doi: 10.1016/j.compfluid.2010.10.024.

[11] N. Dash, S. Chakravarty, A. K. Rath, Deep learning model for elevating Internet of Things intrusion detection, *International Journal of Electrical and Computer Engineering*, 2024, **14**, 5874, doi: 10.11591/ijece. v14i5.pp5874-5883.

[12] F. Lorenzen, A. Zargaran, U. Janoske, Potential of physics-informed neural networks for solving fluid flow problems with parametric boundary conditions, *Physics of Fluids*, 2024, **36**, 037143, doi: 10.1063/5.0193952.

[13] A. Shaaban, The pressure problem of the incompressible flow equations, *International Journal of Aeronautical Science & Aerospace Research*, 2014, 1-2, doi: 10.19070/2470-4415-140001e.

[14] Y. Huang, Z. Zhang, X. Zhang, A direct-forcing immersed boundary method for incompressible flows based on physics-informed neural network, *Fluids*, 2022, **7**, 56, doi: 10.3390/fluids7020056.

[15] N. Margenberg, D. Hartmann, C. Lessig, T. Richter, A neural network multigrid solver for the Navier-Stokes equations, *Journal of Computational Physics*, 2022, **460**, 110983, doi: 10.1016/j.jcp.2022.110983.

[16] L. Devroye, G. Lugosi, G. Neu, Random-walk perturbations for online combinatorial optimization, *IEEE Transactions on Information Theory*, 2015, **61**, 4099-4106, doi: 10.1109/TIT.2015.2428253.

[17] M. Lino, S. Fotiadis, A. A. Bharath, C. D. Cantwell, Current and emerging deep-learning methods for the simulation of fluid dynamics, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2023, **479**, 20230058, doi: 10.1098/rspa.2023.0058.

[18] Y. Huang, Q. Wu, J. Zhang, Solving incompressible fluid flow problems in rectangular domains using finite difference method and recurrent neural networks, *Journal of Computational and Applied Mathematics*, 2018, **341**, 223-237.

[19] Z. K. Lawal, H. Yassin, D. T. C. Lai, A. Che Idris, Physics-informed neural network (PINN) evolution and beyond: a systematic literature review and bibliometric analysis, *Big Data and Cognitive Computing*, 2022, **6**, 140, doi: 10.3390/bdcc6040140.