

Article

Optimization of Turbulence Model Parameters Using the Global Search Method Combined with Machine Learning

Konstantin Barkalov ¹, Ilya Lebedev ¹, Marina Usova ¹, Daria Romanova ^{2,3}, Daniil Ryazanov ²
and Sergei Strijhak ^{2,4,*}

- ¹ Department of Mathematical Software and Supercomputing Technologies, Lobachevsky University, 603022 Nizhni Novgorod, Russia; konstantin.barkalov@itmm.unn.ru (K.B.); ilya.lebedev@itmm.unn.ru (I.L.); usova@itmm.unn.ru (M.U.)
- ² Ivannikov Institute for System Programming of the Russian Academy of Sciences, 109004 Moscow, Russia; romanovadi@ispras.ru (D.R.); ryazanov@ispras.ru (D.R.)
- ³ Faculty of Mechanics and Mathematics, Lomonosov Moscow State University, 119991 Moscow, Russia
- ⁴ Moscow Aviation Institute, Volokolamskoe Shosse 4, 125993 Moscow, Russia
- * Correspondence: s.strijhak@ispras.ru

Abstract: The paper considers the slope flow simulation and the problem of finding the optimal parameter values of this mathematical model. The slope flow is modeled using the finite volume method applied to the Reynolds-averaged Navier–Stokes equations with closure in the form of the $k-\omega$ SST turbulence model. The optimal values of the turbulence model coefficients for free surface gravity multiphase flows were found using the global search algorithm. Calibration was performed to increase the similarity of the experimental and calculated velocity profiles. The Root Mean Square Error (RMSE) of derivation between the calculated flow velocity profile and the experimental one is considered as the objective function in the optimization problem. The calibration of the turbulence model coefficients for calculating the free surface flows on test slopes using the multiphase model for interphase tracking has not been performed previously. To solve the multi-extremal optimization problem arising from the search for the minimum of the loss function for the flow velocity profile, we apply a new optimization approach using a Peano curve to reduce the dimensionality of the problem. To speed up the optimization procedure, the objective function was approximated using an artificial neural network. Thus, an interdisciplinary approach was applied which allowed the optimal values of six turbulence model parameters to be found using OpenFOAM and Globalizer software.

Keywords: global optimization; artificial neural network; function approximation; finite volume method; CFD; OpenFOAM; interFoam

MSC: 62M45; 76D05; 76F60; 90C26



Citation: Barkalov, K.; Lebedev, I.; Usova, M.; Romanova, D.; Ryazanov, D.; Strijhak, S. Optimization of Turbulence Model Parameters Using the Global Search Method Combined with Machine Learning. *Mathematics* **2022**, *10*, 2708. <https://doi.org/10.3390/math10152708>

Academic Editors: Theodore E. Simos and Charampos Tsitouras

Received: 6 July 2022

Accepted: 28 July 2022

Published: 31 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Climate change is accompanied by dangerous natural phenomena that are becoming more and more frequent. These phenomena inevitably affect human life and daily activities. Research into these issues involves mathematical modeling and analysis of geophysical data, as well as data from numerical and field experiments. The current trend in geodata analysis is associated with the use of machine learning algorithms and artificial neural networks.

Landslides, mudflows, and avalanches are classified as dangerous geological phenomena. These are slope processes associated with the separation of rocks, their movement along the slope under the influence of gravity and leading to irreversible changes in the relief. The study of the descent of landslides, mudflows, and avalanches, as well as their prediction and detection are highly relevant due to the impact of these dangerous phenomena on human life and urban infrastructure. Landslides are distinguished according to the main causes of their occurrence. These are: abrasive, erosional, anthropogenic, and

natural/anthropogenic. According to the mechanism of displacement, block landslides, shear, stretching, and liquefaction landslides are classified, respectively [1].

Data show that between January 2004 and December 2016, landslides killed about 56,000 people in as little as 5000 incidents. The spatial distribution of landslides is uneven, with Asia as the predominant geographic area [2].

A large number of landslides, avalanches, and mudflows occur in European countries (Switzerland, Austria, Italy, France, and Iceland). Similar dangerous phenomena occur in Russia, for example, in the Krasnodar Territory of the Russian Federation and in the North Caucasus, due to heavy rainfalls in these regions where mountainous terrain is present, and in connection with the development of the territory (linear and areal objects) in recent years [3]. The amount of precipitation was outstanding in the above territories of Russia (Sochi, Krasnaya Polyana) in 2021. This led to the flooding of mountain rivers and increased the risk of slope flows. Catastrophic descents of mudflows occurred, resulting in damage to road equipment and blockage of roadways [4].

Mudflows are one of the most complex exogenous geological processes, which integrate the actions of other geological processes. The mudflow is a complex heterogeneous structure consisting of liquid and solid components. The solid fraction consists of mineral particles, which are non-uniform from the granulometric viewpoint.

The most well-known landslide problem involves assessing the stability of a landslide slope for static conditions and seismic impact. Modeling the descent of the slope flow makes it possible to predict the damage caused by this phenomenon and to correctly locate the protective structures and vital objects. Such problems are solved using the finite difference method [5], the finite volume method [6], the discrete elements method [7], the cellular automata method [8], and hybrid methods. The mudflows can be modeled using a two-fluid model based on the Volume of Fluid model [9]. The methods listed above are implemented in various commercial and open source software packages MN2D [10], TITAN2D [11], and RAMMS.

Previously, the study of avalanches was carried out using computational fluid dynamics methods for Newtonian fluids and analysis of observation data, including the historical data on avalanches in Japan [12,13]. Similar work was carried out on avalanches in Switzerland, Austria, and Italy. A series of works on the study of avalanches using laboratory experiment in special trays and measuring equipment for studying avalanches in Iceland were carried out at the University of Iceland [14,15]. An experiment with the descent of a snow-water stream in Davos, Switzerland was described in [16]. Measurements were made for the flow depth for a dry snow avalanche and for a snow-water flow.

Cheng et al. [17] used Bayesian analysis to calibrate the coefficients of the Spalart–Allmaras turbulence model to correct model deficiencies and reproduce the profile of a turbulent boundary layer over a flat plate. Among similar works, the coefficients were already calibrated in the $k-\epsilon$ turbulence model when studying the process of the propagation of impurities in the process of urban development [18].

Bayesian analysis is often used to calibrate the RANS closures turbulence models to improve forecast accuracy without losing computational efficiency (as was done for $k-\epsilon$ model with Launder-Sharma damping functions, $k-\omega$ Wilcox model, Spalart–Allmaras and Baldwin–Lomax models [19,20]). De Zordo-Banliat et al. applied Bayesian analysis to compressor cascades to obtain a set of calibrated turbulence model parameters due to the inadequacy of the original model [21].

Kotaro Matsui et al. [22] proposed a new set of calibrated coefficients that improved the ability of the Spalart–Allmaras model to predict compressor cascade flow angular separation. A comprehensive review of turbulence model uncertainties is given in the paper of Xiao and Cinnella [23].

A comprehensive analysis of climatic, geological, and hydrological data was applied to model and predict the slope flows. In the scope of recent trends, huge streams of data received from satellites, from the experiments, and from mathematical modeling are

processed using machine learning, which makes it possible to develop efficient models of these processes [24,25].

Recently, several new optimization and machine learning methods have been proposed. In [26], a new method of regret analysis, called stochastic p-robust optimization, has been proposed, which allows to simultaneously take advantage of stochastic optimization and robust optimization methods to study the optimal operation of a hybrid energy system.

A learning-based method for building driving-signal aware full-body avatars was presented in [27]. The model was a conditional variational auto-encoder that could be animated with incomplete driving signals, such as human pose and facial keypoints, and produced a high-quality representation of human geometry and view-dependent appearance.

A novel multi-class wind turbine bearing fault diagnosis strategy based on the conditional variational generative adversarial network model combining multi-source signals fusion was proposed in [28]. The strategy converted multi-source 1D vibration signals into 2D signals, and the multi-source 2D signals were fused by using wavelet transform.

A complementary label adversarial network (CLARINET) was proposed to solve completely complementary unsupervised domain adaptation (CC-UDA) and partly complementary unsupervised domain adaptation (PC-UDA) problems. CLARINET maintains two deep networks simultaneously, with one focusing on classifying the complementary-label source data and the other taking care of the source-to-target distributional adaptation [29].

A challenging problem called unsupervised open set domain adaptation (UOSDA) was considered in [30]. The authors proposed a practical theoretical bound for UOSDA, which contained an effective risk estimator to evaluate the risk on data with unknown classes. In addition, a DNN-based UOSDA method under the guidance of the proposed theoretical bound was put forward. The method can accurately estimate the risk of the classifier on data with unknown classes and adequately align the distributions of data with known classes.

Methods based on convolutional neural networks (CNN) are able to extract stable spatial and spectral features [31]. A combination of satellite image and topographic data can be used to generalize the extracted features in order to identify slope flows in satellite images [32,33].

Calibration of the $k - \omega$ SST models [34–36] was mainly carried out for air flows around various profiles. For example, calibration was performed for the flow around airfoils at high Reynolds numbers in a wide range of angles of attack [37] (as it was noted by the authors, the area of applicability of the suggested modification is limited to flows around airfoils). Calibration was also carried out for the flow around small wind turbines [38,39] and around a simple flat plate [40]. No calibration of the turbulence model for fluid flows with a free surface on a slope under the action of gravity was performed. The work uses a multi-phase model to track the interface to calculate the flow. The effect of this multiphase nature should also be taken into account when using the coefficients of the turbulent model. In the present work, for the first time, the coefficients of the turbulent model are calibrated for the flow of a multiphase fluid under the action of gravity on a slope.

The slope flow model, as a rule, includes several parameters, the values of which cannot be specified in advance but can be selected based on the consistency of the numerical results with the experimental data available. Such a problem is a global optimization problem with a black box objective function because the specific type of the objective function is not known, there exists only the algorithm for calculating its values.

The complexity of the phenomena and processes under study is reflected in the complexity of the corresponding mathematical models and numerical methods for their analysis. Currently, the main (and often the only possible) tool for such an analysis is supercomputer modeling of the object behavior. Open source software is used widely for this purpose. OpenFOAM (an open platform for numerical modeling of continuum mechanics problems) is a well-known example of open source software of this kind [41].

The growing performance of modern supercomputer systems goes in parallel with the complication of mathematical models of the processes considered that makes performing even a single model calculation (a trial) a computation-costly operation. Consequently, the choice of the optimal values of the model parameters in reasonable time cannot be completed by trying all possible variants by the grid method, i.e., by searching on some regular grid in the range of variation of the parameters. The impossibility of performing a large number of search trials requires applying efficient search algorithms, which would provide an acceptable estimate of a problem solution at a relatively small number of trials using available computational resources.

It is worth noting that most existing methods for finding the optimum of time-consuming black-box functions have some drawbacks. Gradient-based algorithms cannot be used in many cases just because the derivatives of an objective function are unknown while the finite-difference approximations of these derivatives are too computationally costly. At the same time, in general, gradient-based methods only allow finding a locally optimal solution to the problem. Classical direct search methods, which do not require the derivatives, e.g., the Nelder–Mead method [42] or the Hooke–Jeeves method [43], are also local. As a rule, the application of these methods for solving global optimization problems involves several restarts from random grid nodes, which requires a large number of trials.

Deterministic methods of the Lipschitz global optimization, such as DIRECT [44], non-uniform coverages method [45,46], diagonal [47], and simplicial [48] methods guarantee (in the limit) convergence to the global solution of the problem but may require a large number of search trials. Finally, heuristic methods, e.g., differential evolution or simulated annealing, also require a very large amount of computations of the functions to obtain good estimates of the solutions in global optimization problems and at the same time lose in quality to the deterministic algorithms [49,50].

So far, the direct application of optimization algorithms to search for the minima of the time-consuming function may appear too computationally costly when a single trial takes a great deal of computation time. A well known method to overcome this problem is to construct an approximation of the objective function (also known as a response surface model or a metamodel or a surrogate model). Computing its values is a computationally inexpensive operation. The approximation is used further to find the minimum. There are many variants of constructing the approximations for multivariate functions. These are various interpolation methods utilizing polynomials, splines, radial basis functions, Kriging, as well as various regression models. Many of these algorithms are used in the development of global optimization methods.

For example, the use of the radial basis functions has been considered in detail in [51,52]. In [53–55], Kriging-based methods have been proposed. A novel approach to constructing the metamodels and trust-region methods based on these ones has been presented in [56–58].

In the present study, we used an efficient global search algorithm [59,60] for solving the Lipschitz global optimization problems combined with function approximations based on the regression models. At the first search stage, the algorithm worked with the objective function as a direct method. Then, an approximation of the objective function was constructed using the accumulated information. This approximation was used at the second stage of the search. To construct the approximations, we applied Neural Network Regression (NNR).

To study the parameters of slope flows, it is advisable to use interdisciplinary approaches from the fields of computational and experimental hydrodynamics, optimization theory, parallel computing, and neural networks. This paper uses the results of an experiment in a chute with a different angle of inclination, a mathematical model for modeling a two-phase flow, global optimization methods, and an artificial neural network. The mathematical model is based on the averaged Reynolds equations and a two-parameter turbulence model.

The turbulence model $k - \omega$ SST contains a number of coefficients calibrated for canonical flows in pipes and channels, air flow around various profiles, etc. [61–63]. Calibration of turbulence model coefficients was not carried out for calculating the free surface flows on slopes using the multiphase model for interphase tracking was performed. In this work, the coefficients of the $k - \omega$ SST turbulence model were optimized for calculating the turbulent fluid flow in the chute.

The main part of the paper has the following structure. Section 2 contains a description of the experiment. Section 3 describes the mathematical model of turbulent two-phase flows. Section 4 contains the details of numerical methods. Section 5 describes the results of numerical experiments. Section 6 contains a discussion. Section 7 concludes the paper.

2. Inclined Chute Experiment

To study slope flow parameters, experimental setups were used. The objects of study were such measurements as flow depth, flow velocity, and force of interaction with an obstacle. In our study, the velocity profile was studied.

An experimental setup at the Research Institute of Mechanics, Lomonosov Moscow State University [64], was used for this purpose. To carry out the calibration, a turbulent flow in an inclined chute was studied. Three different angles of inclination were used during the work. The chute had a simple rectangular geometry 1 m long, 12 cm wide, and 10 cm high. The section of the chute to be examined was between two points of velocity and depth measurements located at the distances of 23 cm and 82 cm from the top side of chute. The schematic diagram of the experimental setup is shown in Figure 1. Tap water was used in the experiment.

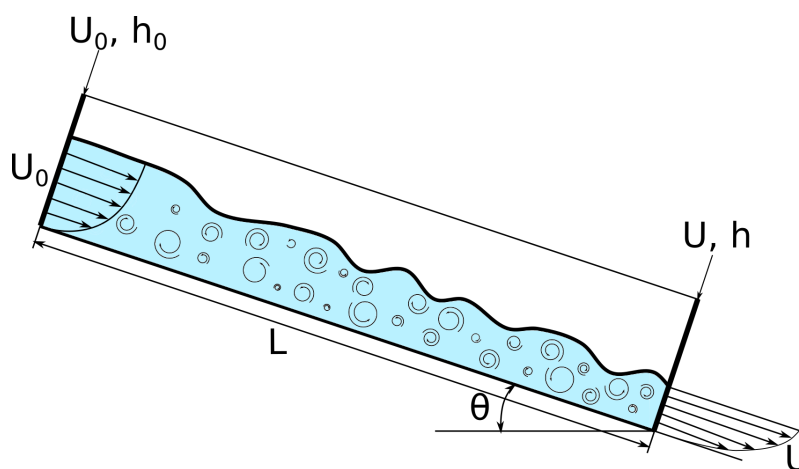


Figure 1. Schematic diagram of the experimental chute.

The experiment was carried out in a stationary mode. Stationarity was provided by a submersible pump “JEELEX Fekalnik 200” with a capacity of 200 L per minute. The measurement was carried out using a Pitot tube connected to a “KORUND-DDN-001M” pressure sensor with an error of 0.1%. Measurement points were spaced every 0.5 mm. A 10 s average was used for each measurement. The water temperature in the experiment was 20 degrees Celsius (room temperature). All temperature-related model coefficients were chosen based on this value.

Three series of experiments were performed; the initial flow profile, the initial flow depth, the slope angle were varied as shown in Table 1 [64], where u_0 is the depth-averaged velocity, h_0 is the flow depth, θ is the slope inclination angle.

Table 1. Parameters of the experiments [64].

u_0 , m/s	h_0 , mm	θ
1.63	4.20	25°
2.00	4.95	28°
1.78	3.45	33°

3. Mathematical Model

The Reynolds-averaged Navier–Stokes equations [65–67] were used to model the experiment carried out at the Research Institute of Mechanics of Lomonosov Moscow State University. The $k - \omega$ SST turbulence model [34,35] was used to obtain the values of the Reynolds stress tensor. The position of the free surface of the flow is determined using the VOF (Volume Of Fluid) method suggested by C.W. Hirt and B.D. Nichols in 1981 [9]. In this method, the volume fraction of water phase α in the cell is used to determine the free surface so that if $\alpha > 0.6$, the cell is considered to be filled with liquid, otherwise—to be filled with air.

The flow in the experimental setup is described by a system of five equations (1). These are the Reynolds-averaged Navier–Stokes equations (continuity equation and momentum conservation equation). The system also includes a transfer equation for the phase volume fraction to track the interface. The system of equations is closed by two equations of conservation of turbulent kinetic energy and a special dissipation of turbulent kinetic energy, which are used to calculate the Reynolds stresses that arise when averaging the Navier–Stokes equations.

$$\begin{cases} \nabla \cdot \bar{u} = 0, \\ \frac{\partial \alpha}{\partial t} + \nabla \cdot (\bar{u}\alpha) = 0, \\ \frac{\partial(\rho\bar{u})}{\partial t} + \nabla \cdot (\rho\bar{u}\bar{u}) = -\nabla \bar{p} + \nabla \cdot \bar{\tau} + \rho\bar{f}, \\ \frac{\partial(\rho k)}{\partial t} + \nabla \cdot (\rho\bar{u}k) = \tilde{P}_k - \beta^* \rho k \omega + \nabla \cdot ((\mu + \alpha_k \mu_t) \nabla k), \\ \frac{\partial(\rho \omega)}{\partial t} + \nabla \cdot (\rho\bar{u}\omega) = \gamma \rho \bar{s}^2 - \beta \rho \omega^2 + \nabla \cdot ((\mu + \alpha_\omega \mu_t) \nabla \omega) + \\ 2(1 - F_1) \rho \alpha_{\omega 2} \frac{1}{\omega} \nabla k \cdot \nabla \omega. \end{cases} \quad (1)$$

Here, α is the water volume fraction, $\bar{\tau} = 2\mu_{eff}\bar{s}$ is the stress tensor, $\bar{s}$ is the strain rate tensor, $\mu_{eff} = \mu + \mu_t$ is the effective viscosity, μ is a molecular viscosity, $\mu_t = \rho a_1 k / \max(a_1 \omega, b_1 \bar{s} F_2)$ is a turbulent viscosity, $\bar{f}$ is the density of the body forces, $\bar{u}$ is the mixture velocity, ρ is the mixture density, $\bar{p}$ is the pressure, ω is the specific dissipation rate of the turbulent kinetic energy, k is the turbulent kinetic energy, $F_1 = \tanh\left(\left(\min\left(\max\left(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega}\right), \frac{4\rho \alpha_{\omega 2} k}{CD_{k\omega} y^2}\right)\right)^4\right)$ is the blending function (F_1 equals zero away from the wall and the model turns into the $k - \varepsilon$ model, while inside the boundary layer F_1 equals one and the $k - \omega$ model is realized), $\bar{s}$ is the strain rate (invariant of $\bar{s}$), $CD_{k\omega} = \max\left(2\rho \alpha_{\omega 2} \frac{1}{\omega} \nabla k \cdot \nabla \omega, 10^{-10}\right)$, $F_2 = \tanh\left(\left(\max\left(\frac{2\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega}\right)\right)^2\right)$ is the second blending function, and $\tilde{P}_k = \min(\mu_t \nabla \bar{u} \cdot [\nabla \bar{u} + (\nabla \bar{u})^T], 10 \cdot \beta^* \rho k \omega)$ is the limiter on the growth of turbulence used in stagnation modes. Details can be found in the OpenFOAM package user guide [68].

The turbulence model contains four coefficients: $\alpha_k, \alpha_\omega, \beta, \gamma$. $\alpha_k, \alpha_\omega, \gamma$ are calculated using the weighted average principle: $\gamma = \gamma_1 F_1 + \gamma_2 (1 - F_1)$. By default, the coefficients of the turbulent model are set by the following values [61–63]:

$$\begin{aligned} \gamma_1 &= 5/9, \quad \beta_1 = 3/40, \quad \alpha_{k1} = 0.85, \quad \alpha_{\omega1} = 0.5, \\ \gamma_2 &= 0.44, \quad \beta_2 = 0.083, \quad \alpha_{k2} = 1, \quad \alpha_{\omega2} = 0.856, \\ \beta^* &= 0.09, \quad a_1 = 0.31, \quad b_1 = 1.0, \quad c_1 = 10.0. \end{aligned} \quad (2)$$

The details of the turbulence model description are presented in [34,36].

4. Methods

4.1. CFD Numerical Method

To implement the three-dimensional multiphase single-rate approach, the interFoam [69] solver of OpenFOAM package (v2012, created by Henry Weller in 1989, Bracknell, United Kingdom) was used.

The approximation schemes used in the work are listed below.

- The time terms were approximated with first order Euler numerical scheme;
- The convection term, the water volume fraction flux term, divergence of the stress tensor term were approximated with second order numerical scheme;
- The turbulent kinetic energy flux, the dissipation flux of the specific turbulent kinetic energy term were approximated with first order bounded numerical scheme;
- The gradient terms were calculated using Gaussian integration with linear interpolation;
- The Laplacian terms were calculated using Gaussian integration with linear interpolation with explicit non-orthogonal correction;
- All other terms in the equations were discretized using a central difference numerical scheme.

The second order linear upwind scheme used for the convection term is most efficient and accurate for Reynolds Averaged Navier–Stokes (RANS) simulations [70].

The PIMPLE algorithm [71,72], which was developed to run the equations with a large Courant number, was used to solve the system equations. The PIMPLE is a combination of PISO [73] and SIMPLE [74] algorithms.

The conjugate gradient method with preconditioner GAMG is used to solve the system of linear equations for pressure. The GaussSeidel method is used as a smoother. The values for volume fraction of water, velocity, k, ω are defined using smoothSolver and symGaussSeidel method as a smoother.

4.1.1. Definition of the Calculation Domain

The advantage of mathematical modeling is that the model allows virtual sensors to be placed at any point in the computational domain to measure the values of physical quantities. In the experiment, the physical sensors were placed at the exit from the chute. To compare the results of the experiment and the calculation, the virtual sensors were positioned in the same place.

A section of the experimental chute located between two velocity profile and flow depth measurement points was simulated. The simulations were performed for the thick 10 mm part of the chute where the influence of the side walls was small. The first measured profile was used for the input data for the computational domain. The second one was the object of comparison. The parallelepiped with a size of 590 mm in length, 10 mm in width, and 10 mm in height was used for the numerical domain. We tested the effect of grid resolution. Grid convergence was studied for various grid sizes of $290 \times 10 \times 30$, $590 \times 10 \times 60$, $1080 \times 10 \times 120$, and $2160 \times 10 \times 240$. For each run, the output velocity profile was compared with the experimental one using the loss function (5). The values of the loss functions for the last three mesh sizes varied within 0.1%. Therefore, the number

of cells was chosen to be $590 \times 10 \times 60$ for reasons of reducing computer time while maintaining accuracy.

4.1.2. Initial and Boundary Conditions

The special boundaries for numerical domain were defined: the chute bottom, the chute sides, the upper border for numerical domain, and the inlet and outlet planes.

The following boundary conditions were set:

- The solid wall with the no-slip condition was used for the chute bottom;
- The zero gradient condition was used for the chute side;
- The mixed condition with atmospheric pressure was used for the upper border of numerical domain, no inflow through the border and outflow according to zero gradient condition and fixed value condition for k and ω ;
- The fixed values were used for inlet plane for water volume fraction, velocity profile, k and ω values;
- The zero gradient condition was used for outlet plane.

The mathematical formulation of the listed initial and boundary conditions is presented in the OpenFOAM user manual [68].

The average value of Y^+ was 17 which satisfied the model of wall functions. The `nutkWallFunction` was used for boundary condition for the wall, which provides a wall constraint on the turbulent viscosity, based on the turbulent kinetic energy for both low- and high-Reynolds number turbulence models.

The initial conditions in the problem are set so that the volume is completely filled with stationary air and the liquid flows in through the inlet plane. After a while, the flow is established and measurements are taken on the outlet plane for comparison with the experimental data. The time step dt is equal to 0.001 s. The flow is considered steady after 5 s.

4.2. Global Optimization Problem Statement

The Globalizer software (Lobachevsky University) was used to solve the optimization problem.

Let us assume that the choice of some set of values of the model parameters is determined by the values of vector $y = (y_1, y_2, \dots, y_N)$ and the quality of the model corresponding to a given value of the vector of parameters is described by the function $\varphi(y)$. Let us call this function the optimization criterion: a decrease in the criterion value corresponds to a better mathematical model. Additionally, assume that some requirements must be satisfied to guarantee the applicability of the model. Meeting these requirements is usually formulated as the condition for the vector y to belong to the hyperinterval D ,

$$D = \{a_i \leq y_i \leq b_i, 1 \leq i \leq N\}. \quad (3)$$

So far, the process of choosing the optimal set of the model parameters corresponds to a global optimization problem of the kind:

$$\begin{aligned} \varphi(y^*) &= \min \{\varphi(y) : y \in D\}, \\ D &= \{y \in \mathbb{R}^N : a_i \leq y_i \leq b_i, 1 \leq i \leq N\}. \end{aligned} \quad (4)$$

When optimizing the coefficients of the turbulent model, the Root-Mean-Square Error (RMSE) of differentiation of the calculated flow velocity profile and the experimental one on the outlet plane is taken into account.

$$L_{RMSE} = \sqrt{\frac{\sum_{i=1}^N (u_{EXP}^i - u_{k-\omega SST}^i)^2}{N}} \quad (5)$$

is minimized. Here, u_{EXP}^i is the horizontal component of velocity at the control point obtained by experiment, and $u_{k-\omega SST}^i$ is the horizontal component of velocity at the control point calculated by computational fluid dynamic (CFD), N is the number of comparison points for the horizontal component of velocity over the flow depth at the outlet plane (right cross-section displayed in Figure 1). Approximately 10 comparison points are used. Their number is determined by the number of measurements performed in the experiment and varies depending on the change in the flow depth at different chute inclination angles. The points are evenly distributed in depth for the operation of measurement tools used in the experiment.

We will consider the loss function (5) as the objective function $\varphi(y)$ in the global optimization problem (4). The problems considered are characterized by the fact that the objective function $\varphi(y)$ is not defined analytically; there is only an algorithm for computing its values at the points of the domain D . In this case, one search trial corresponds to one computation according to the model and is a time-consuming operation [75,76].

Multi-extremal optimization problems have much higher computation costs for solving them as compared to other types of optimization problems since the global optimum is an integral characteristic of the problem being solved and requires investigating the whole search domain. As a result, the search for the global optimum is reduced to constructing some coverage (grid) in some range of parameters and choosing the optimal function value on this grid. The amount of computations may be reduced by constructing a non-uniform coverage of the search domain: the grid should be dense enough in the vicinity of the global optimum and less dense far away from the sought solution.

The assumption that the objective function $\varphi(y)$ satisfies the Lipschitz condition

$$|\varphi(y_1) - \varphi(y_2)| \leq L\|y_1 - y_2\|, \quad y_1, y_2 \in D, \quad 0 < L < \infty, \quad (6)$$

is a typical and is used in many global optimization methods [44,46,60,77]. The assumption of this kind is natural enough for many applied problems since the relative variations of the function characterizing the process being simulated cannot usually exceed some threshold imposed by the limited energy of variation. The question of estimating the Lipschitz constant values unknown *a priori* arising here can be resolved by introducing some additive schemes [78,79].

There are several ways to adapt efficient one-dimensional algorithms for solving multidimensional problems (see, e.g., [47,48]). In this study we apply the dimensionality reduction using Peano curve $y(x)$ continuously mapping the unit interval $[0, 1]$ onto the n -dimensional cube

$$\{y \in R^N : -2^{-1} \leq y_i \leq 2^{-1}, 1 \leq i \leq N\} = \{y(x) : 0 \leq x \leq 1\}. \quad (7)$$

Algorithms for constructing Peano-type space filling curves and the corresponding theory are considered in detail in [59,60].

By using this kind of mapping, the multivariate problem (4) could be reduced to a univariate problem

$$\varphi(y^*) = \varphi(y(x^*)) = \min \{\varphi(y(x)) : x \in [0, 1]\}. \quad (8)$$

An important property of such mapping is that if the function $\varphi(y)$ in the domain D satisfies the Lipschitz condition, then the function $\varphi(y(x))$ on the interval $[0, 1]$ will satisfy a uniform Hölder condition

$$|\varphi(y(x_1)) - \varphi(y(x_2))| \leq H|x_1 - x_2|^{1/N}, \quad (9)$$

where the Hölder constant H is linked to the Lipschitz constant L by the relation $H = 2L\sqrt[N+3]{}$ [59].

Therefore, we can consider minimization of univariate function

$$f(x) = \varphi(y(x)), \quad x \in [0, 1], \quad (10)$$

satisfying the Hölder condition.

4.3. The Global Search Algorithm

The algorithm for solving the problem (4) involves constructing a sequence of points x^k , where the values of the objective function $z^k = f(x^k) = \varphi(y(x^k))$ are calculated. Let us call the process of calculating the function value (including the construction of an image $y^k = y(x^k)$) the “trial”, and the pair (y^k, z^k) , the “trial result”. The set of pairs $\{(y^k, z^k), 0 \leq k \leq n\}$ makes up the search data collected using the method after carrying out n steps. The rules that determine the work of the *global search algorithm* are as follows.

The first two trials are performed at the boundary points of the segment $[0, 1]$, i.e., $x^0 = 0$ and $x^1 = 1$. The values $z^0 = f(x^0)$ and $z^1 = f(x^1)$ of the objective function are calculated, and the counter $k = 1$ is set. A next trial point $x^{k+1}, k \geq 1$, is chosen using the following rules.

Step 1. Renumber points of the set $X_k = \{x^0, \dots, x^k\}$ with subscripts in increasing order of coordinate values, i.e.,

$$0 = x_0 < x_1 < \dots < x_{k-1} < x_k = 1. \quad (11)$$

Note that hereinafter superscripts are used to denote the iteration number, and subscripts are used to number the points in order.

Step 2. Supposing that $z_i = f(x_i)$, $1 \leq i \leq k$, calculate values

$$\mu = \max_{1 \leq i \leq k} \frac{|z_i - z_{i-1}|}{\Delta_i}, \quad (12)$$

$$M = \begin{cases} r\mu, & \mu > 0, \\ 1, & \mu = 0, \end{cases} \quad (13)$$

where the real number $r > 1$ is the method input parameter, and $\Delta_i = (x_i - x_{i-1})^{1/N}$.

Step 3. For each interval (x_{i-1}, x_i) , $1 \leq i \leq k$, calculate a characteristic according to the following formula

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{M^2 \Delta_i} - 2 \frac{z_i + z_{i-1}}{M}, \quad 1 \leq i \leq k. \quad (14)$$

Step 4. Select the interval (x_{t-1}, x_t) corresponding to the maximum characteristic

$$R(t) = \max \{R(i) : 1 \leq i \leq k\}. \quad (15)$$

Step 5. Execute the new trial at the point $x^{k+1} \in (x_{t-1}, x_t)$, calculated using the following formula

$$x^{k+1} = \frac{x_t + x_{t-1}}{2} - \text{sign}(z_t - z_{t-1}) \frac{1}{2r} \left[\frac{|z_t - z_{t-1}|}{\mu} \right]^N. \quad (16)$$

The algorithm stops when $\Delta_t < \epsilon$, where $\epsilon > 0$ is the preset accuracy. For estimation of the global optimum, values

$$f_k^* = \min_{0 \leq i \leq k} f(x^i), \quad x_k^* = \arg \min_{0 \leq i \leq k} f(x^i), \quad (17)$$

are chosen.

A rigorous proof of this algorithm's convergence is provided in [59].

4.4. Construction of the Objective Function Approximation

4.4.1. The Use of Neural Networks

There are no universal rules for the choice of the neural network topology to solve a particular problem. However, in [80] the Kolmogorov theorem has been generalized and it was proved that any continuous function of N variables can be approximated by a three-layered artificial feedforward neural network with one hidden layer and an error backpropagation algorithm as a learning one with any degree of precision. This theorem is called the Universal Approximation Theorem or the Cybenko theorem [81].

Neural networks as approximators were implemented in many machine learning libraries. In the present study, MLPRegressor class from scikit-learn machine learning library was used to construct the objective function approximation. It implements a multi-layer perceptron (MLP), which is learned using error backpropagation without activation function in the output layer [82]. MLPs have demonstrated an ability to find approximate solutions for very complex problems.

A MLP with one hidden layer with a scalar output is shown in Figure 2.

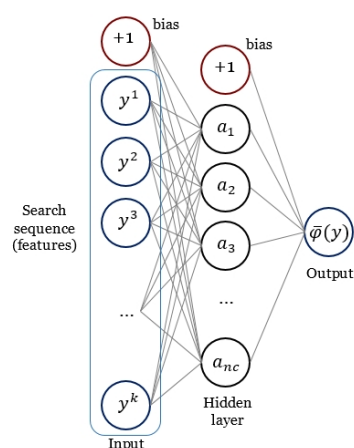


Figure 2. Three-layer perceptron with scalar output.

The left layer called the input layer consists of a set of neurons y_i , $i = \overline{1, k}$ representing the input signals (the values of variables). Each neuron in the hidden layer transforms the values from the previous layer with weighted linear summing

$$w_1 y_1 + w_2 y_2 + \dots + w_k y_k + bias, \quad (18)$$

where w_i are the weights of the neurons and $bias$ is a special weight, which does not include a factor in the form of an input value. Next, the value obtained is transformed into an output (predicted) value of the layer with a transmission function (the *activation function*). The output layer receives the values from the last hidden layer and transforms them into the output values. The network was trained by the error backpropagation method.

We used a three-layered neural network for solving the approximation problem due to the following reasons. From the theoretical point of view, such a network will be sufficient to approximate the function with a high accuracy. From the practical point of view, the use of deep neural networks here will be redundant, because the set of trial results used to build the approximation is small and will not be sufficient to train a deep network.

4.4.2. Selection of the Model Parameters

The choice of the solver, the activation function, the value of the regularization parameter, the number of neurons in the hidden layer, etc., are the variable adjustment parameters of the neural network. For example, for small sets of the multidimensional data, the “lbfgs” solver was demonstrated to be better and faster. This solver is a modification of the Broyden–Fletcher–Goldfarb–Shanno algorithm [83] and belongs to the quasi-Newton

methods. All numerical experiments were carried out using this algorithm. The sigmoidal functions (logistic or hyperbolic tangent) were used as the neuron activation functions. The number of neurons in each layer and the regularization parameter (alpha) were adjusted in the experiments and depended on a particular problem.

In the experiments conducted, we chose the following network architecture:

```
model = MLPRegressor(activation='logistic',
solver='lbfgs',
alpha=0.001,
hidden_layer_sizes=(20,),
max_iter=5000,
tol=10e-6,
random_state=10)
```

4.5. The Use of Approximations in Solving the Optimization Problem

In the present study, we applied the following method of using the objective function approximation in the optimization problems: to construct the objective function approximation using the accumulated search information, to find the minimum of the approximation, and to repeat this process, either until the computation resources are exhausted or until the convergence is achieved.

The method proposed will make sense either in the case when the amount of the search information accumulated is large enough (that allows constructing a relatively precise approximation of a multi-extremal function) or in the case when the problem is similar to a local extremum search problem. The first case corresponds to the final stage of search and can be interpreted as a method of refining the current solution. However, if the objective function is time-consuming, it is impossible to conduct a large enough number of trials. This is where we will encounter an exhaustion of computing resources.

The second case implies constructing a good approximation based on a relatively small number of trials and, in fact, includes an assumption on a weak multi-extremality of the objective function that matches well with the problem considered within the framework of the present study.

The global search algorithm using the objective function approximation can be formulated as follows. Let us assume that the available resources allow for $K_{max} = K_1 + K_2$ trials to be performed.

At the first stage, $k = K_1$ trials are performed using the core global search algorithm from Section 4.3. In the course of performing the first stage, a set of the trial results $\Omega = \{(y^k, \varphi(y^k)), 0 \leq k \leq K_1\}$ necessary to construct the objective function approximation is accumulated.

At the second stage, the algorithm works using the approximation. To compute the point y^{k+1} of the next $(k + 1)$ th trial, the following operations are performed.

Step 1. Using the set of the trial results Ω formed in the course of the algorithm execution, to construct an approximation of the objective function $\bar{\varphi}(y)$;

Step 2. Using the core global search algorithm from Section 4.3 to find the global minimum of the function $\bar{\varphi}(y)$ and to use this value as the next trial point, i.e., $y^{k+1} = \arg \min_{y \in D} \bar{\varphi}(y)$;

Step 3. If either the condition $k > K_{max}$ or the condition $\|y^k - y^{k+1}\| \leq \epsilon$ is satisfied, to stop the algorithm. Else, to perform the trial at the point y^{k+1} , to store its result in the set Ω , to increment the trial counter $k = k + 1$, and to proceed to Step 1.

The algorithm proposed here ensures the convergence to the global solution in the case if K_1 trials executed at the first stage are sufficient to construct an approximation of the objective function reflecting the main features of its behavior adequately.

5. Results

The simulations were conducted using the supercomputer of Lobachevsky University of Nizhni Novgorod (operated under the Linux CentOS 7.2 operation system). Each supercomputer node included two Intel Sandy Bridge E5-2660 2.2 GHz processors, 64 Gb RAM. The central processor unit had 8 cores. The global optimization methods considered in the present work were implemented in C++ using GCC 5.5.0 compiler and Open MPI v4.1.1. To construct the objective function approximations using a neural network, scikit-learn machine learning library from Python 3.9 was applied. To numerically solve the problem described in Section 3, the open source CFD software OpenFOAM v2012 [84] was used.

Before starting the calibration, a small study of the significance of each of the 12 coefficients of the turbulent model was carried out. As a result, it was revealed that the coefficients β^* , a_1 , $\alpha_{k1,2}$, and $\alpha_{\omega 1,2}$ make the most significant contribution to the calculation results. It was decided to calibrate these coefficients. These coefficients determine the dissipation rate of turbulent kinetic energy, the Reynolds stress, the diffusion fluxes of turbulent kinetic energy, and the specific dissipation rate. The coefficient β^* is used in the mixing functions that describe the mechanism for switching between the $k - \varepsilon$ and $k - \omega$ models. The coefficient a_1 determines the turbulent viscosity. $\alpha_{k1,2}$ characterize the diffusion flux of the kinetic energy of turbulence. $\alpha_{\omega 1,2}$ characterize the diffusion flow by the specific rate of dissipation of the kinetic energy of turbulence.

The initial values of the coefficients are

$$\beta^* = 0.09; \quad a_1 = 0.31; \quad \alpha_{k1} = 0.85; \quad \alpha_{\omega 1} = 0.5; \quad \alpha_{k2} = 1.0; \quad \alpha_{\omega 2} = 0.856. \quad (19)$$

One calculation of the objective function for given values of parameters took 15 min in average with the use of 8 MPI-processes per node.

The optimal values of parameters were adjusted for pairs, the values of the remaining parameters were fixed. First, a pair of the most important parameters β^* and a_1 was selected. To investigate the optimization problem posed, both possible approaches to solving it were applied: without the use of the objective function approximation and with the use of the approximation.

In the first experiment, the global search algorithm described in Section 4.3 was applied without the use of approximation. The parameters of the method were set as follows: $r = 3$ and $\epsilon = 10^{-3}$. In 24 h, 100 iterations of the algorithm were performed; the required accuracy was not achieved.

In the second experiment, the approach described in Section 4.5 was applied to solve the same problem. First, $K_1 = 30$ iterations of global search algorithm were performed. Afterwards, the algorithm employing the approximation with the neural network was started. A total of $K_1 + K_2 = 65$ iterations of the algorithm were performed, after that the algorithm stopped on accuracy. As a result, the best value of the objective function 0.375 was found. The total search time was reduced to 16 h, which ensured a more accurate solution for the problem in a reasonable amount of time.

The trial points and the approximating function plotted according to these points using the neural network are presented in Figure 3 (the parameters β^* and a_1 were varied). Several local minima are clearly visible. Our analysis showed good agreement between the regression model and experimental data. The R^2 score and the RMSE between the model predictions and the simulated results are equal to 0.976 and 0.098, respectively.

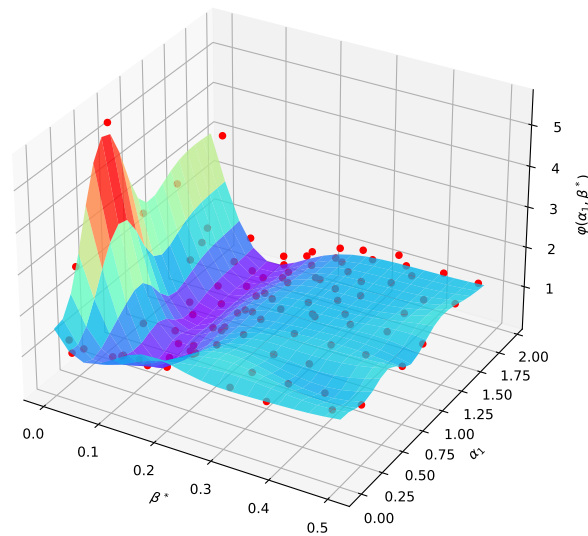


Figure 3. Objective function values (red points) and the approximation plot constructed using the neural network (parameters β^* and α_1 were varied).

The best values of parameters β^* and α_1 found were fixed, and then optimization in parameters α_{k1} , $\alpha_{\omega1}$, and α_{k2} , $\alpha_{\omega2}$ was performed. However, no significant improvement of the objective function by optimizing on these parameters was achieved: the value of 0.365 was obtained.

As a final result, the following values of the coefficients of the model were obtained:

$$\begin{aligned} \beta^* &= 0.117; \quad \alpha_1 = 1.84; \quad \alpha_{k1} = 1.999; \\ \alpha_{\omega1} &= 0.062; \quad \alpha_{k2} = 1.241; \quad \alpha_{\omega2} = 0.003. \end{aligned} \quad (20)$$

Figure 4 shows the resulting velocity profiles $\bar{u}_x$ as a function of flow depth h on the exit plane for various slope angles.

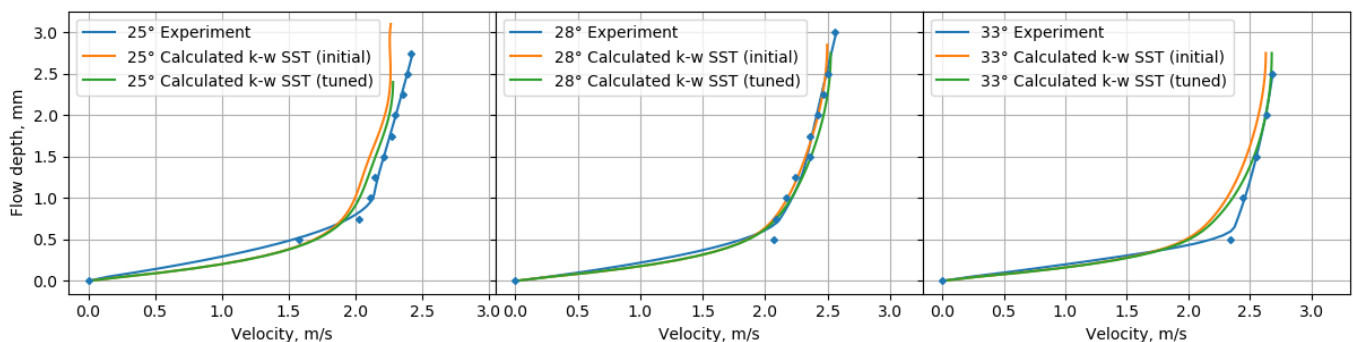


Figure 4. The comparison graphs of the experimental velocity profile and the calculated velocity profile using the standard values of the $k - \omega$ SST turbulence model coefficients and the calculated velocity profile with calibrated values of the coefficients for different slope inclination angles.

During the process of calibration, the minimization of the loss function (5) was achieved, as shown in Table 2.

Table 2. Loss function values obtained during the minimization process.

Slope Inclination Angle	Initial Loss Function Value	Minimized Loss Function Value
25°	0.165	0.155
28°	0.085	0.128
33°	0.150	0.089

Figure 4 and Table 2 show a decrease in the discrepancy between the calculated velocity profile and the experimental one for two out of three experiments. Optimization for the three experiments combined (their loss functions were summarized) was used in order to avoid the model overfitting. In one experiment, an almost perfect coincidence of the calculated velocity profile with the experimental one was achieved, which was not reproduced in other experiments. It should also be noted that the divergence of the velocity profile in the region near the chute bottom is due to the measurement error in the experiment, since it is difficult to measure the velocity with the Pitot tube used in this experiment in the immediate vicinity of the bottom.

The minimum value for the objective function was obtained in the case of the slope angle of 33 degrees. One calculation of the objective function for the given parameters took 15 min in average using 8 MPI processes on a node of the Lobachevsky University supercomputer. Total computation time was 24 h.

6. Discussion

When working on the optimization of the turbulent model coefficients, we faced a number of tasks: creating an interface for automatic interaction between the Globalizer software and the OpenFOAM package; in the process of optimization, we encountered the overfitting problem; and the study of the best loss function was carried out, etc.

Creating an interaction interface between Globalizer and OpenFOAM required the use of the pyFoam library to prepare calculation runs. Using the script of this library pyFoamPrepareCase.py, new values of the turbulent model coefficients proposed by the Globalizer software were written into the calculation cases. Next, the calculation was carried out in parallel mode and the result was evaluated using a python script that compared the obtained velocity profile with the experimental one.

A study of various loss functions has been carried out. The loss functions that estimate the absolute error of the velocity profile and the relative error were compared. The relative loss function actually penalizes the area near the bottom more, however, optimization using this loss function did not show significant improvements in the result. This behavior of the calculation is not due to the shortcomings of the numerical model, but rather due to the impossibility of taking velocity values correctly in the region very close to the bottom of the chute. It should be noted that measurements at the point closest to the bottom can have a significant error, due to the measurement method used (the Pitot tube). However, most of the profile was measured quite accurately, since for each measurement point in this stationary experiment, time averaging of 10 s was used. As a result of this study, it was decided to use the absolute loss function, since it showed the best optimization result.

We performed three experiments to avoid overfitting. It was noticed that when optimizing for one velocity profile, the algorithm perfectly calibrated the model, but for other experiments, the result was much worse. When optimizing for three experiments at once, this effect was avoided. The use of the obtained values of the coefficients for fluid flows that are close in dimensionless characteristics will allow a more accurate calculation to be made. However, when generalized to such canonical flows as air flow around different profiles, the use of the obtained values of the turbulent model coefficients is unlikely to show the best result.

This study is part of a larger research effort related to the modeling of currents on the slopes of mountains. These flows are very difficult to study, since they are turbulent multi-phase flows of a non-Newtonian fluid on slopes of complex geometry. For such flows, the use of turbulent models using standard values of coefficients does not predict the flow in the best way. It is necessary to calibrate the turbulence model coefficients or create new turbulence models. We are considering both options. Moreover, a new turbulent model is being developed based on the neural network. However, the use of neural networks requires hybrid computing clusters, which is not always possible. As a result, it is important to be able to obtain a solution of sufficient accuracy using the classical turbulent model. This problem requires the calibration of the coefficients, which was done in this work for the Newtonian medium. Optimization algorithms were developed and tested, which can later be used to calibrate the coefficients of any turbulent models. The next stage of the study is to set up an experiment with a non-Newtonian fluid and calibrate the coefficients of the turbulent model according to the developed algorithm.

We also note that when searching for the optimal model parameters, optimization was carried out first with respect to the parameters β^* and a_1 , then optimization in $\alpha_{k1}, \alpha_{\omega1}$, and $\alpha_{k2}, \alpha_{\omega2}$ was performed. This approach makes it possible to find a good solution in a reasonable amount of time. The search for a global optimum for all parameters at once would require orders of magnitude more trials and time. This effect is a key difference between global optimization and local optimization problems, in which the costs do not grow as fast.

In terms of solving a time-consuming optimization problem with a black-box objective function, it is interesting to compare the method used for solving the problem, which uses objective function approximation by a neural network, with the methods that use kriging-based approximations. Methods of this class work well in problems with a small number of local extrema. However, in essentially multi-extremal problems the computational costs (number of objective function evaluations required for solving the problem) increase significantly.

7. Conclusions

In this work, a two-phase flow in a chute was simulated using the interFoam solver, the URANS mathematical model, and the $k - \omega$ SST turbulence model. In the optimization process, six coefficients were investigated in pairs, which make the greatest contribution to the value of turbulent viscosity.

The results of calculating the velocity profiles were compared with experimental data obtained at the Research Institute of Mechanics, Moscow State University, at different sections depending on the angle of inclination of the chute. The search for the optimal coefficients of the turbulence model was performed by minimizing the objective function of the divergence of the velocity profile in the chute.

The search for the global minimum of the objective function was performed using the global search algorithm implemented in the Globalizer software [85]. In addition, a fully connected neural network with one hidden layer was used to approximate the values of the objective function by the values of the coefficients in the turbulence model. The MLPRegressor class from the scikit-learn library was used to build the objective function approximation.

Based on the results of the study, the following conclusions were drawn.

It is good practice to calibrate the turbulence model coefficients if necessary to improve the accuracy of the calculations. As shown in the present study, it is possible to improve the accuracy by up to 10%.

To perform optimization, at least two, or preferably three datasets should be used. Otherwise, an overfitted model may result. Using a neural network to predict the CFD calculation significantly reduced the optimization time while maintaining the quality of the resulting solution.

The developed calibration algorithm is reliable and can be applied to other models. The algorithm has such advantages as a parallel mode, it can be used to search not only for local, but also for global minima, and can optimize several parameters at once. According to the results of the study, the Globalizer software has performed quite well and will be used in further work.

The OpenFOAM software also shows good results due to code modularity and good documentation. These advantages have made it easier to write a software module for the interaction between the optimizer and OpenFOAM. OpenFOAM also has a high degree of parallelization and can be used to solve a fairly wide range of tasks.

In the present work, an interdisciplinary approach was utilized, which helped us to find the optimal values of six turbulence model parameters using the OpenFOAM open platform and the Globalizer. In the future, it is planned to continue improving the turbulent models, including the development of a turbulent model based on a neural network. The Globalizer software will be used to optimize the model parameters.

Author Contributions: Conceptualization, K.B. and D.R. (Daniil Ryazanov); methodology, D.R. (Daria Romanova) and K.B.; software, I.L., D.R. (Daniil Ryazanov), M.U. and D.R. (Daria Romanova); validation, D.R. (Daria Romanova) and I.L.; formal analysis, S.S., M.U.; investigation, S.S.; resources, I.L.; data curation, D.R. (Daria Romanova) and M.U.; writing—original draft preparation, K.B., S.S. and D.R. (Daria Romanova); writing—review and editing, D.R. (Daniil Ryazanov); visualization, I.L.; supervision, S.S.; project administration, K.B.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science and Higher Education of the Russian Federation, agreement No 075-15-2020-808.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors consider it their duty to acknowledge the contribution of Victor Gergel (14.01.1955–29.06.2021), who initiated this interdisciplinary study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pendin, V.; Fomenko, I. *Landslide Hazard Assessment and Prediction Methodology*; 2015; p. 320.
2. Froude, M.J.; Petley, D.N. Global fatal landslide occurrence from 2004 to 2016. *Nat. Hazards Earth Syst. Sci.* **2018**, *18*, 2161–2181. [CrossRef]
3. Hungr, O. *Landslide Risk Management: Proceedings of the International Conference on Landslide Risk Management, Vancouver, BC, Canada, 31 May–3 June 2005*; Balkema: Leiden, NY, USA, 2005.
4. Kharchenko, S.; Shvarev, S. Forecasting of Landslide Hazards in the Vicinity of Krasnaya Polyana Basing on Liniar Discriminatory Analysis. *Vestnik Moscow State Univ. Ser. 5 Geography*. **2020**, 22–33. Available online: https://vestnik5.geogr.msu.ru/jour/article/view/668?locale=en_US (accessed on 1 May 2022).
5. Bernander, S.; Kullingsjö, A.; Gylland, A.S.; Bengtsson, P.E.; Knutsson, S.; Pusch, R.; Olofsson, J.; Elfgren, L. Downhill progressive landslides in long natural slopes: Triggering agents and landslide phases modeled with a finite difference method. *Can. Geotech. J.* **2016**, *53*, 1565–1582. [CrossRef]
6. Gao, L.; Jian-li, D.; Chang-yu, L. The application of finite volume method to modeling landslide motion. *Adv. Earth Sci.* **2007**, *22*, 1129–1133.
7. Liu, Z.; Su, L.; Zhang, C.; Iqbal, J.; Hu, B.; Dong, Z. Investigation of the dynamic process of the Xinmo landslide using the discrete element method. *Comput. Geotech.* **2020**, *123*, 103561. [CrossRef]
8. Piegari, E.; Cataudella, V.; Di Maio, R.; Milano, L.; Nicodemi, M. A cellular automaton for the factor of safety field in landslides modeling. *Geophys. Res. Lett.* **2006**, *33*, L01403. [CrossRef]
9. Hirt, C.; Nichols, B. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **1981**, *39*, 201–225. [CrossRef]
10. Naaim, M.; Furdada, G.; Martínez, H. Calibration and application of the MN2D dynamics model to the avalanches of Las Leñas (Argentina). *Nat. Hazards Earth Syst. Sci.* **2002**, *2*, 221–226. [CrossRef]

11. Pitman, E.B.; Nichita, C.C.; Patra, A.; Bauer, A.; Sheridan, M.; Bursik, M. Computing granular avalanches and landslides. *Phys. Fluids* **2003**, *15*, 3638–3646. [\[CrossRef\]](#)
12. Oda, K.; Moriguchi, S.; Kamiishi, I.; Yashima, A.; Sawada, K.; Sato, A. Simulation of a snow avalanche model test using computational fluid dynamics. *Ann. Glaciol.* **2011**, *52*, 57–64. [\[CrossRef\]](#)
13. Yamaguchi, Y.; Takase, S.; Moriguchi, S.; Terada, K.; Oda, K.; Kamiishi, I. Three-dimensional nonstructural finite element analysis of snow avalanche using non-Newtonian fluid model. *Trans. Jpn. Soc. Comput. Eng. Sci.* **2017**, *2017*, 20170011. [\[CrossRef\]](#)
14. Agustsdottir, K.H. The Design of Slushflow Barriers: Laboratory Experiments. Doctoral Dissertation, University of Iceland, Reykjavik, Iceland, 2019.
15. Jones, R. The Design of Slushflow Barriers: CFD Simulations. Master Thesis, University of Iceland, Reykjavik, Iceland, 2019. Available online: <http://hdl.handle.net/1946/34502> (accessed on 1 May 2022).
16. Jaedicke, C.; Kern, M.; Gauer, P.; Baillifard, M.A.; Platzer, K. Chute Experiments on Slushflow Dynamics. In Proceedings of the 2006 International Snow Science Workshop, Telluride, CO, USA, 1–6 October 2006.
17. Cheung, S.H.; Oliver, T.A.; Prudencio, E.E.; Prudhomme, S.; Moser, R.D. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliab. Eng. Syst. Saf.* **2011**, *96*, 1137–1149. [\[CrossRef\]](#)
18. Guillas, S.; Glover, N.; Malki-Epshtein, L. Bayesian calibration of the constants of the turbulence model for a CFD model of street canyon flow. *Comput. Methods Appl. Mech. Eng.* **2014**, *279*, 536–553. [\[CrossRef\]](#)
19. Edeling, W.; Cinnella, P.; Dwight, R. Predictive RANS simulations via Bayesian Model-Scenario Averaging. *J. Comput. Phys.* **2014**, *275*, 65–91. [\[CrossRef\]](#)
20. Edeling, W.; Cinnella, P.; Dwight, R.; Bijl, H. Bayesian estimates of parameter variability in the $k-\epsilon$ turbulence model. *J. Comput. Phys.* **2014**, *258*, 73–94. [\[CrossRef\]](#)
21. de Zordo-Banliat, M.; Merle, X.; Dergham, G.; Cinnella, P. Bayesian model-scenario averaged predictions of compressor cascade flows under uncertain turbulence models. *Comput. Fluids* **2020**, *201*, 104473. [\[CrossRef\]](#)
22. Matsui, K.; Perez, E.; Kelly, R.; Tani, N.; Jemcov, A. Calibration of Spalart-Allmaras model for simulation of corner flow separation in linear compressor cascade. *J. Glob. Power Propuls. Soc.* **2021**, 1–16. [\[CrossRef\]](#)
23. Xiao, H.; Cinnella, P. Quantification of model uncertainty in RANS simulations: A review. *Prog. Aerosp. Sci.* **2019**, *108*, 1–31. [\[CrossRef\]](#)
24. Karpatne, A.; Ebert-Uphoff, I.; Ravela, S.; Babaie, H.A.; Kumar, V. Machine Learning for the Geosciences: Challenges and Opportunities. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 1544–1554. [\[CrossRef\]](#)
25. Ma, Z.; Mei, G.; Piccialli, F. Machine learning for landslides prevention: A survey. *Neural Comput. Appl.* **2020**, *33*, 10881–10907. [\[CrossRef\]](#)
26. Yu, D.; Wu, J.; Wang, W.; Gu, B. Optimal performance of hybrid energy system in the presence of electrical and heat storage systems under uncertainties using stochastic p-robust optimization technique. *Sustain. Cities Soc.* **2022**, *83*, 103935. [\[CrossRef\]](#)
27. Bagautdinov, T.; Wu, C.; Simon, T.; Prada, F.; Shiratori, T.; Wei, S.E.; Xu, W.; Sheikh, Y.; Saragih, J. Driving-Signal Aware Full-Body Avatars. *ACM Trans. Graph.* **2021**, *40*, 1–17. [\[CrossRef\]](#)
28. Zhang, L.; Zhang, H.; Cai, G. The Multiclass Fault Diagnosis of Wind Turbine Bearing Based on Multisource Signal Fusion and Deep Learning Generative Model. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [\[CrossRef\]](#)
29. Zhang, Y.; Liu, F.; Fang, Z.; Yuan, B.; Zhang, G.; Lu, J. Learning From a Complementary-Label Source Domain: Theory and Algorithms. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Zhong, L.; Fang, Z.; Liu, F.; Yuan, B.; Zhang, G.; Lu, J. Bridging the Theoretical Bound and Deep Algorithms for Open Set Domain Adaptation. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–15. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 645–657. [\[CrossRef\]](#)
32. Qin, S.; Guo, X.; Sun, J.; Qiao, S.; Zhang, L.; Yao, J.; Cheng, Q.; Zhang, Y. Landslide Detection from Open Satellite Imagery Using Distant Domain Transfer Learning. *Remote Sens.* **2021**, *13*, 3383. [\[CrossRef\]](#)
33. Prakash, N.; Manconi, A.; Loew, S. A new strategy to map landslides with a generalized convolutional neural network. *Sci. Rep.* **2021**, *11*, 9722. [\[CrossRef\]](#)
34. Menter, F. Zonal Two Equation $k-\omega$ Turbulence Models For Aerodynamic Flows. In Proceedings of the 23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference, Orlando, FL, USA, 6–9 July 1993. [\[CrossRef\]](#)
35. Menter, F.R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA J.* **1994**, *32*, 1598–1605. [\[CrossRef\]](#)
36. Menter, F.; Kuntz, M.; Langtry, R. Ten years of industrial experience with the SST turbulence model. *Heat Mass Transf.* **2003**, *4*, 625–632.
37. Matyushenko, A.A.; Garbaruk, A.V. Adjustment of the $k-\omega$ SST turbulence model for prediction of airfoil characteristics near stall. *J. Phys. Conf. Ser.* **2016**, *769*, 012082. [\[CrossRef\]](#)
38. Rocha, P.C.; Rocha, H.B.; Carneiro, F.M.; da Silva, M.V.; de Andrade, C.F. A case study on the calibration of the $k-\omega$ SST (shear stress transport) turbulence model for small scale wind turbines designed with cambered and symmetrical airfoils. *Energy* **2016**, *97*, 144–150. [\[CrossRef\]](#)
39. Rocha, P.; Rocha, H.; Carneiro, F.; Silva, M.; Bueno, A. $k-\omega$ SST (shear stress transport) turbulence model calibration: A case study on a small scale horizontal axis wind turbine. *Energy* **2013**, *65*, 412–418. [\[CrossRef\]](#)

40. Kalitzin, G.; Medic, G.; Xia, G. Improvements to SST turbulence model for free shear layers, turbulent separation and stagnation point anomaly. In Proceedings of the 54th AIAA Aerospace Sciences Meeting, San Diego, CA, USA, 8 January 2016. [\[CrossRef\]](#)
41. Weller, H.G.; Tabor, G.; Jasak, H.; Fureby, C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* **1998**, *12*, 620. [\[CrossRef\]](#)
42. Nelder, J.; Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **1965**, *7*, 308–313. [\[CrossRef\]](#)
43. Hooke, R.; Jeeves, T. “Direct Search” Solution of Numerical and Statistical Problems. *J. ACM* **1961**, *8*, 212–229. [\[CrossRef\]](#)
44. Jones, D.R. The DIRECT global optimization algorithm. In *Proceedings of the The Encyclopedia of Optimization*; Springer: Heidelberg, Germany, 2009; pp. 725–735.
45. Evtushenko, Y.; Malkova, V.; Stanevichyus, A.A. Parallel global optimization of functions of several variables. *Comput. Math. Math. Phys.* **2009**, *49*, 246–260. [\[CrossRef\]](#)
46. Evtushenko, Y.; Posypkin, M. A deterministic approach to global box-constrained optimization. *Optim. Lett.* **2013**, *7*, 819–829. [\[CrossRef\]](#)
47. Sergeyev, Y.; Kvasov, D. *Deterministic Global Optimization: An Introduction to the Diagonal Approach*; Springer: New York, NY, USA, 2017.
48. Paulavičius, R.; Žilinskas, J. *Simplicial Global Optimization*; Springer: New York, NY, USA, 2014.
49. Sergeyev, Y.; Kvasov, D.; Mukhametzhanov, M. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Sci. Rep.* **2018**, *8*, 435. [\[CrossRef\]](#)
50. Kvasov, D.; Mukhametzhanov, M. Metaheuristic vs. deterministic global optimization algorithms: The univariate case. *Appl. Math. Comput.* **2018**, *318*, 245–259. [\[CrossRef\]](#)
51. Gutmann, H.M. A Radial Basis Function Method for Global Optimization. *J. Glob. Optim.* **2001**, *19*, 201–227. [\[CrossRef\]](#)
52. Regis, R.; Shoemaker, C. Constrained global optimization of expensive black box functions using radial basis functions. *J. Glob. Optim.* **2005**, *31*, 153–171. [\[CrossRef\]](#)
53. Jones, D.; Schonlau, M.; Welch, W. Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Optim.* **1998**, *13*, 455–492. [\[CrossRef\]](#)
54. Ur Rehman, S.; Langelaar, M.; van Keulen, F. Efficient Kriging-based robust optimization of unconstrained problems. *J. Comput. Sci.* **2014**, *5*, 872–881. [\[CrossRef\]](#)
55. Ollar, J.; Mortished, C.; Jones, R.; Siens, J.; Toropov, V. Gradient based hyper-parameter optimisation for well conditioned kriging metamodels. *Struct. Multidiscip. Optim.* **2017**, *55*, 2029–2044. [\[CrossRef\]](#)
56. Polynkin, A.; Toropov, V. Mid-range metamodel assembly building based on linear regression for large scale optimization problems. *Struct. Multidiscip. Optim.* **2012**, *45*, 515–527. [\[CrossRef\]](#)
57. Ollar, J.; Toropov, V.; Jones, R. Sub-space approximations for MDO problems with disparate disciplinary variable dependence. *Struct. Multidiscip. Optim.* **2017**, *55*, 279–288. [\[CrossRef\]](#)
58. Gergel, V.; Barkalov, K.; Kozinov, E.; Toropov, V. Parallel multipoint approximation method for large-scale optimization problems. *Commun. Comput. Inf. Sci.* **2018**, *910*, 174–185. [\[CrossRef\]](#)
59. Strongin, R.G.; Sergeyev, Y.D. *Global Optimization with Non-Convex Constraints. Sequential and Parallel Algorithms*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2000.
60. Sergeyev, Y.D.; Strongin, R.G.; Lera, D. *Introduction to Global Optimization Exploiting Space-Filling Curves*; Springer Briefs in Optimization; Springer: New York, NY, USA, 2013.
61. Launder, B.; Spalding, D. The numerical computation of turbulent flows. *Comput. Methods Appl. Mech. Eng.* **1974**, *103*, 456–460. [\[CrossRef\]](#)
62. Tahry, S.H.E. k-epsilon equation for compressible reciprocating engine flows. *J. Energy* **1983**, *7*, 345–353. [\[CrossRef\]](#)
63. Launder, B.; Morse, A.; Rodi, W.; Spalding, D. Spalding, The prediction of free shear flows—A comparison of the performance of six turbulence models. In Proceedings of the NASA Conference on Free Shear Flows, Hampton, VA, USA, 20–21 July 1972.
64. Romanova, D.; Ivanov, O.; Trifonov, V.; Ginzburg, N.; Korovina, D.; Ginzburg, B.; Koltunov, N.; Eglit, M.; Strijhak, S. Calibration of the $k-\omega$ SST Turbulence Model for Free Surface Flows on Mountain Slopes Using an Experiment. *Fluids* **2022**, *7*, 111. [\[CrossRef\]](#)
65. Wilcox, D.C. *Turbulence Modeling for CFD*; DCW Industries: La Canada, CA, USA, 2006.
66. Hirsch, C. *Numerical Computation of Internal and External Flows. The Fundamentals of Computational Fluid Dynamics*; Elsevier Ltd.: Amsterdam, The Netherlands, 2007. [\[CrossRef\]](#)
67. Ferziger, J.; Peric, M. *Computational Methods for Fluid Dynamics*; Springer: Berlin, Germany, 2002; Volume 3. [\[CrossRef\]](#)
68. OpenFOAM: User Guide. Available online: <https://www.openfoam.com/documentation/guides/v2112/doc/index.html> (accessed on 1 May 2022).
69. Rusche, H. *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*. Doctoral Dissertation, Imperial College London, London, UK, 2003.
70. Robertson, E.; Choudhury, V.; Bhushan, S.; Walters, D. Validation of OpenFOAM numerical methods and turbulence models for incompressible bluff body flows. *Comput. Fluids* **2015**, *123*, 122–145. [\[CrossRef\]](#)
71. Holzmann, T. *Mathematics, Numerics, Derivations and OpenFOAM®*; Holzmann CFD: Loeben, Germany, 2019.
72. Yin, R. Comparison of four algorithms for solving pressure-velocity linked equations in simulating atrium fire. *Int. J. Arch. Sci.* **2003**, *4*, 24–35.

73. Issa, R. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys.* **1986**, *62*, 40–65. [[CrossRef](#)]
74. Issa, R.; Gosman, A.; Watkins, A. The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. *J. Comput. Phys.* **1986**, *62*, 66–82. [[CrossRef](#)]
75. Kalyulin, S.; Shavrina, E.; Modorskii, V.; Barkalov, K.; Gergel, V. Optimization of Drop Characteristics in a Carrier Cooled Gas Stream Using ANSYS and Globalizer Software Systems on the PNRPU High-Performance Cluster. *Commun. Comput. Inf. Sci.* **2017**, *753*, 331–345.
76. Paulavičius, R.; Sergeyev, Y.; Kvasov, D.; Žilinskas, J. Globally-biased BIRECT algorithm with local accelerators for expensive global optimization. *Expert Syst. Appl.* **2020**, *144*, 113052. [[CrossRef](#)]
77. Paulavičius, R.; Žilinskas, J.; Grothey, A. Investigation of selection strategies in branch and bound algorithm with simplicial partitions and combination of Lipschitz bounds. *Optim. Lett.* **2010**, *4*, 173–183. [[CrossRef](#)]
78. Strongin, R.; Barkalov, K.; Bevzuk, S. Global optimization method with dual Lipschitz constant estimates for problems with non-convex constraints. *Soft Comput.* **2020**, *24*, 11853–11865. [[CrossRef](#)]
79. Strongin, R.; Barkalov, K.; Bevzuk, S. Acceleration of Global Search by Implementing Dual Estimates for Lipschitz Constant. *Lect. Notes Comput. Sci.* **2020**, *11974*, 478–486. [[CrossRef](#)]
80. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [[CrossRef](#)]
81. Hassoun, M. *Fundamentals of Artificial Neural Networks*; MIT Press: Cambridge, MA, USA, 1995.
82. Hecht-Nielsen, R. Theory of the backpropagation neural network. In Proceedings of the IJCNN International Joint Conference on Neural Networks, Washington, DC, USA, 19–22 June 1989; pp. 593–605. [[CrossRef](#)]
83. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer: New York, NY, USA, 2006.
84. Moukalled, F.; Mangani, L.; Darwish, M. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*; Springer: Cham, Switzerland, 2016. [[CrossRef](#)]
85. Gergel, V.; Barkalov, K.; Sysoyev, A. A novel supercomputer software system for solving time-consuming global optimization problems. *Numer. Algebra Control Optim.* **2018**, *8*, 47–62. [[CrossRef](#)]