Computer Physics Communications 271 (2022) 108216

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

Numerical algorithm based on regularized equations for incompressible flow modeling and its implementation in OpenFOAM [☆]

Matvey V. Kraposhin [a,*], Daniil A. Ryazanov [a], Tatiana G. Elizarova [b]

[a] Ivannikov Institute for System Programming of the RAS, Alexander Solzhenitsyn st., 25, Moscow, Russian Federation
[b] Keldysh Institute of Applied Mathematics RAS, Miusskaya sq., 4, Moscow, Russian Federation

ARTICLE INFO

Article history:
Received 23 January 2021
Received in revised form 7 October 2021
Accepted 24 October 2021
Available online 29 October 2021

Keywords:
Quasi hydrodynamic equations
Incompressible flows
OpenFOAM
QHDFoam
Finite volume method
Computational fluid dynamics

ABSTRACT

Various aspects of incompressible viscous flow modeling using regularized equations are considered: from the formulation of quasi hydrodynamic (QHD) equations for fluid motion to the demonstration of the capabilities of the associated OpenFOAM solver *QHDFoam*.

The numerical algorithm for fluid flow simulations is devised using the QHD approach. Unlike prominent operator splitting techniques (PISO, SIMPLE and their variants), the QHD approach provides an equation for pressure explicitly. Practical issues of QHD approximation and stability criteria for the numerical scheme are presented.

Several problems are used to validate the developed solver and to demonstrate its capabilities: backward facing step, skewed cavity flow, natural convection and generation of internal gravity waves attractor. Results of simulations for the last case are obtained using 3 different methods to show peculiarities of the quasi hydrodynamic equations approach in comparison with the standard PISO algorithm and finite-volume method and with the solution based on the spectral element method.

The source code of the developed OpenFOAM solver(s) can be downloaded from the GitHub repository: https://github.com/unicfdlab/QGDsolver.

1. Introduction

The following peculiarities usually come to mind when considering the modeling of incompressible viscous fluid flows: (i) the diversity of industrial and fundamental problems, involving their application and (ii) the ubiquitous usage of pressure-based numerical algorithms for their simulation.

Milovan Perić in his 2003-year report [1] claimed: “Computation of laminar flows poses nowadays no problems: the available computing resources are big enough to enable accurate solutions even in complex 3D problems.” However, if we turn our attention to complex flow phenomena with buoyancy, mesh deformation, non-inertial frame of references, surface tension forces and perhaps other effects, then the aforementioned statement must be reconsidered. This is especially critical when we deal with general-purpose codes for a first estimation of a given problem. The roughness and low accuracy of general-purpose numerical methods is usually balanced by small efforts for algorithm implementation when using finite-volume second-order platforms (such as Logos-CFD, Ansys CFX, AMReX, Star CCM and others), which makes them an attractive tool for multiphysics simulations. Therefore, the question of a reliable and accurate second-order numerical framework for the simulation of incompressible flows with complex phenomena is still open.

The essential part of all numerical algorithms for incompressible flows is the pressure field computation, since pressure field is decoupled from the density one (and hence, from the continuity equation), while the original system of equations does not contain an explicit expression for the pressure evolution. Despite the simplicity of operator-splitting algorithms and their prevalence, the question of pressure computation is still meaningful.

Various approximations of incompressible Navier-Stokes equations are known for flows with buoyancy in Boussinesq assumption, fluid-structure interaction, flows in domains with elastic boundaries, turbulent flows with adaptive-mesh approaches, low-speed two-phase flows, external aerodynamics and many others. Depending on the problem and the physical phenomena involved, the approaches for the numerical simulation of viscous incompressible flows can be classified in the following groups (non-exhaustive list):

[☆] The review of this paper was arranged by Prof. Hazel Andrew.

[*] Corresponding author.
E-mail address: m.kraposhin@ispras.ru (M.V. Kraposhin).

https://doi.org/10.1016/j.cpc.2021.108216
0010-4655/© 2021 Elsevier B.V. All rights reserved.

- meshless methods of vortex domains transport;
- operator splitting methods with pressure equation, including:
  - projection methods – Chorin projection, fractional step method;
  - single diagonal implicit Runge-Kutta (SDIRK) methods;
  - meshless methods based on Finite Pointset Method;
  - iterative predictor-corrector algorithms with implicit algebraic equation for pressure;
- marker-and-cell (MAC) method for staggered Cartesian grids.

Vortex methods are primarily intended for external aerodynamics at high *Re* numbers and/or motion of bodies in flows. These algorithms stand apart from a number of numerical algorithms for incompressible flows because the pressure field is not involved directly in the velocity calculation. Instead, the transport equation for vorticity is solved and pressure is recovered from a Cauchy–Lagrange integral analog [2]. An open-source implementation of vortex method for planar flow named VM2D is available at GitHub [3]. This code implements an algorithm of viscous vortex domain method with a modified approximation of boundary conditions [4]. A 3D version called Vortex Loops method can also be found [5]. Despite all their virtues, these methods have not been widely applied for problems outside of primary class, possibly due to the non-trivial procedure for numerical scheme derivation in complex flows.

The application of Lagrangian approaches to incompressible Navier-Stokes equations in their general form has produced several methods, which are applicable with various degrees of success to different types of flows: Smoothed Particle Hydrodynamics (SPH) [6], Particle Finite Element Method (PFEM) [7], Finite Pointset Method (FPM) [8,9] and others. While the discussion of advantages and limitations of Lagrangian methods is out of the scope of this work, we note the application of the Chorin projection method [10] for pressure-velocity coupling in many implementations of these methods.

Considering that the Chorin method is also applicable to various mesh-based approaches (finite-element method, finite-volume method, finite-difference method) and different meshes (structured, unstructured, polyhedral) it appears be concluded, that the advantage of projection-based methods arises from their wide range of application and simplicity of implementation.

The main features of the standard Chorin-Uzawa [11] projection method and its modifications for Eulerian and Lagrangian approaches are:

- the simplicity of the approach – The Poisson pressure equation is derived by splitting the original system into parts with and without a contribution from the pressure gradient;
- continuity and momentum equations are decoupled from each other and can be solved sequentially;
- a zero normal derivative of pressure as boundary condition, which introduces error in the near-wall region;
- the approximation results in two different velocity fields, the one complies with momentum balance, while the other one is divergence-free.

The method has several extensions, including:

- higher order approximations for temporal terms [12,13],
- Dirichlet boundary conditions remedying the issue with pressure approximation on solid walls [14],
- application of single diagonally Runge-Kutta methods [15,16] for temporal terms.

However, the main problems related to the issues of dual velocity fields and choice of adequate pressure boundary condition in situations with viscous flows and bulk forces (gravity, Coriolis and others) still persist.

In another approach, called Marker-And-Cell method (MAC) [17,18], the pressure equation is derived from the governing equations and solved prior to momentum balance, enforcing a continuity constraint on the velocity field at the new time layer. The method solves the previously mentioned issue with pressure boundary condition, since there is no need to specify one on solid walls. The velocity field at each time level obeys both continuity and momentum. However, two drawbacks prevent this method from mass usage:

1. non-trivial and expensive procedure for solving the pressure equation;
2. it is applicable only to Cartesian structured meshes.

A review of MAC and projection-type methods could be found for example in [19].

Algorithms based on operators splitting can be considered as an intermediate between MAC and Projection methods for pressure-velocity coupling. In all variants of this method, the momentum equation in algebraic-differential form is split into three parts – i) pressure gradient, ii) contribution to momentum balance from operators related to velocity and iii) other terms. Application of a divergence operator to the discretized momentum equation produces the Poisson equation for pressure. Depending on the way discrete momentum and pressure equations are solved, two approaches can be used:

1. coupled approach, when momentum and pressure algebraic equations are solved simultaneously within one matrix;
2. PISO (Pressure Implicit with Splitting of Operators [20,21]), SIMPLE (Semi-Implicit Method for Pressure Linked Equations [22]) and their variants, when momentum and pressure equations are solved sequentially and iteratively.

The widespread use of pressure-based algorithms such as SIMPLE or PISO is connected with the relative simplicity of the procedure for pressure equation derivation and its coupled solution with momentum balance. This (sometimes apparent) simplicity manifests in the broad range of computational codes that utilize these methods and in their frequent presence in educational materials of all kinds (from short introductory lessons to comprehensive guides) dedicated to computational fluid dynamics.

Indeed, the generality of operator splitting approaches allows building universal general-purpose programs, such as STAR-CCM+, Ansys Fluent, OpenFOAM, etc.

For example, introducing SIMPLE and PISO algorithms into the OpenFOAM library allowed to create computational programs (solvers) for different types of fluid flow problems:

- laminar, turbulent, Newtonian and non-Newtonian incompressible flows;
- low-Mach number compressible flows;
- incompressible flows with buoyancy under the Boussinesq assumption;
- incompressible two-phase flows, including turbulent fluid-particles mixtures [23].

Usually, when some operators splitting approach is employed, a base computational algorithm is common for all mentioned models, while changes affect only the right-hand side (r.h.s.) of the momentum equation. In theory, this allows using a modular framework where additional phenomena are accounted through decoupled momentum sources and transport equations (such as temperature, salinity, etc.), which are solved sequentially.

However, in practice changes to the time advancement algorithm are more substantial than just an addition or subtraction of coefficients from the momentum matrix diagonal. Depending on the problem statement (equations and boundary conditions, grid properties – regular vs. irregular, structured vs. unstructured, Cartesian vs. curvilinear, constraints on the approximation error), changes also involve a reformulation of force balance, a modification of mass flux predictor-corrector, a modification of pressure gradient approximation and other numerical tricks to meet target requirements.

Another issue of PISO and SIMPLE methods arises from using of cycles within the time advancement procedure. Usually three nested loops of iterations are used inside the main time cycle to couple pressure and velocity:

1. outer loop for SIMPLE iterations;
2. inner loop for PISO iterations;
3. deferred pressure correction loop to account for non-orthogonality of mesh.

Needless to say, introducing of more complex flow phenomena to a model such as phase transitions or mesh motion can lead to additional iteration loops and adds complexity to the algorithm.

In this work, the alternative numerical approach based on QHD or quasi-hydrodynamic equations is employed. Within that approach the pressure equation is derived directly from the continuity equation, thus allowing the construction of a general-purpose, sequential and even decoupled approximation of fluid motion equations for incompressible viscous flows. A similar approach, but applied to perfect gas dynamics equations was previously implemented in OpenFOAM for compressible flow modeling [24].

QHD equations were derived by Yu. V. Sheretov in [25,26] as a special case of liquid motion equations. The close mutual relation between QHD and the Navier-Stokes equations, the positiveness of the QHD dissipative function and the existence of a family of common solutions of both systems were shown.

Later, QHD equations were approximated and used in the numerical simulation of incompressible flows by T.G. Elizarova, e.g. [27,28]. The applicability of regularized hydrodynamic equations algorithms was shown for several flow problems. Her pioneering work allowed elaborating practical recommendations for adjusting the regularization parameter τ, which is the unique tuning coefficient of the model.

The approach is still in active development, enhancement and usage both in applied and theoretical aspects. For example, the properties of QHD system were studied in [29,30], whereas in [31,32] the approach has been extended to simulate the motion of compressible two-phase fluid with surface tension.

Regardless of the momentum equation's r.h.s., the numerical algorithm within the QHD approach includes 4 steps:

- computation of face fluxes;
- solution of pressure equation;
- computation of volumetric fluxes that satisfy the continuity equation;
- solution of momentum and other equations.

Therefore, changes in momentum equation lead to changes in the regularization terms of continuity equation leaving the overall computation algorithm unaffected.

The simplicity and generality of the QHD approach make it a promising alternative to PISO/SIMPLE algorithms for incompressible flows with complex phenomena. The difference between QHD and other approaches can be summarized as follows:

- in contrast to the projection approach, the velocity field in the QHD algorithm satisfies the approximated momentum equation;
- unlike the MAC approach, the QHD numerical algorithm is applicable to unstructured grids, but requires Neumann boundary condition for pressure;
- by comparison with the PISO/SIMPLE approach, the QHD numerical approach does not need iterations within time step. The equation for pressure is derived directly from the continuity equation, the pressure-velocity coupling procedure is independent of the momentum equation's r.h.s. This equation requires additional Neumann boundary condition on walls, which can impact slightly the solution in the boundary layer like in PISO/SIMPLE approach;
- these features lead to a simple serial algorithm without additional nested loops, therefore, such approach might be useful for developing hydrodynamic models with multiple phenomena in domains with complex geometries.

The paper is organized as follows. In Section 2, the system of regularized or quasihydrodynamic equations with buoyancy is presented. Section 3 contains the temporal-spatial approximation of QHD equations for unstructured mesh. Details of the QHD time-advancement algorithm and its implementation as the *QHDFoam* solver based on the OpenFOAM computational library program are presented in Section 4. Section 5 contains the results of solver verification and validation, as well as a comparison of QHD technique with the PISO approach and the spectral element method (program Nek5000 [33,34]). The conclusion Section summarizes distinctive features of the QHD algorithm implementation and results of its application to mentioned above flow problems.

2. Mathematical model

For many years, Navier-Stokes equations with constant density have been recommended as a reliable tool for the simulation of low-Mach number flows. Small variations of density due to heat convection or solute transport and their influence on the buoyancy force are usually treated under the Boussinesq approximation, which provides enough accuracy for both scientific researchers and engineers.

However, more than twenty years ago an extension of Navier-Stokes equations system was proposed by Yu. V. Sheretov for the numerical simulation of incompressible flow, see [25–28]. These equations were named regularized hydrodynamic or quasi-hydrodynamic (QHD) equations.

QHD equations originate from the work of B.N. Chetverushkin [35,36] devoted to derivation and application of kinetic algorithms (later they were named as quasi-gasdynamic or QGD algorithms) to rarefied gas flows. The extension of this approach to incompressible flows produced QHD equations. The equations differ from the conventional Navier-Stokes ones by additional terms containing second-order spatial derivatives multiplied by small regularization parameter τ that has the dimension of a time. These terms produce additional dissipation and provide stability to numerical algorithms. When the value of τ tends to zero, QHD equations degenerate into the Navier-Stokes ones.

QHD equations include continuity equation (1), momentum balance equations (2) and scalar s transport equation (3):

$$\nabla \cdot \left(\vec{U} - \vec{W}\right) = 0, \tag{1}$$

$$\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \left((\vec{U} - \vec{W}) \otimes \vec{U}\right) - \nabla \cdot \nu \left(\nabla \vec{U} + (\nabla \vec{U})^T\right) - \nabla \cdot \left(\vec{U} \otimes \vec{W}\right) = -\frac{1}{\rho_0}\nabla \tilde{p} + \vec{F}, \tag{2}$$

$$\frac{\partial s}{\partial t} + \nabla \cdot \left((\vec{U} - \vec{W}) s \right) - \nabla \cdot \frac{\nu}{Sc} (\nabla s) - \nabla \cdot \left(\tau \vec{U} (\vec{U} \cdot) \nabla s \right) = 0, \quad (3)$$

where $\vec{U}$ is the velocity, ν is the kinematic viscosity, ρ_0 is a reference density; $\vec{F} = \beta \vec{g} \tilde{s}$ is the body force density, $\tilde{p} = p(t, \vec{x}) - p(0, \vec{x})$ is pressure $p(t, \vec{x})$ perturbation, $Sc = \frac{\nu}{D}$ is a fluid Schmidt number defined as the ratio between kinematic viscosity ν and coefficient of mass diffusion D, s is the transported scalar quantity (temperature or solute concentration) which influences the body force $\rho_0 \vec{F}$ associated with buoyancy, $\tilde{s} = s(t, \vec{x}) - s(0, \vec{x})$ denotes the variation of s and $\beta = \frac{1}{\rho_0} \frac{\partial \rho}{\partial s}$ is the coefficient of dilatation (in case of temperature) or contraction coefficient (in case of solute) of the fluid under consideration.

Regularized addition to velocity $\vec{W}$ (or the regularizing velocity) that appears in Eqs (1)-(3) is defined as:

$$\vec{W} = \tau \left((\vec{U} \cdot \nabla) \vec{U} + \frac{1}{\rho_0} \nabla \tilde{p} - \vec{F} \right). \quad (4)$$

One of the advantages of QHD equations resides in the possibility of deriving the pressure equation directly from the continuity equation (1) together with the expression (4) for $\vec{W}$:

$$\nabla \cdot \frac{\tau}{\rho_0} \nabla \tilde{p} = \nabla \cdot \left(\vec{U} - \tau (\vec{U} \cdot \nabla) \vec{U} + \tau \vec{F} \right). \quad (5)$$

The initial–boundary value problem of fluid motion under the Boussinesq approximation is governed by equations (2)–(5), boundary conditions (BCs) at inlet (6), outlet (7) and impermeable walls (8) of a computational domain and initial conditions (ICs) for pressure p (or its perturbation $\tilde{p}$), velocity $\vec{U}$ and scalar s:

$$\vec{U} = \vec{U}_b, \quad \frac{\partial \tilde{p}}{\partial \vec{n}} = \rho_0 \vec{n} \cdot \left(-\vec{U}_b \cdot \nabla \vec{U} + \vec{F} \right), \quad s = s_b, \quad (6)$$

$$\frac{\partial \vec{U}}{\partial \vec{n}} = 0, \quad \tilde{p} = 0, \quad \frac{\partial s}{\partial \vec{n}} = 0, \quad (7)$$

$$\vec{U} = 0, \quad \frac{\partial \tilde{p}}{\partial \vec{n}} = \rho_0 \vec{n} \cdot \left(-\vec{U}_b \cdot \nabla \vec{U} + \vec{F} \right), \quad \lambda \frac{\partial s}{\partial \vec{n}} + \gamma s = \psi, \quad (8)$$

where $\vec{U}_b$ and s_b are prescribed values at inlet for velocity and scalar s, respectively. λ, γ and ψ are specified constants to switch between Neumann and Dirichlet BCs for scalar s, Neumann boundary condition for pressure is derived from a condition imposed on the regularized mass flux at the corresponding boundary: $\vec{n} \cdot \vec{W} = 0$.

When solid walls are stationary, the body force is negligible and the normal velocity gradient at inlet is zero, BCs (6) and (8) degenerate into (9) and (10):

$$\vec{U} = \vec{U}_b, \quad \frac{\partial \tilde{p}}{\partial \vec{n}} = 0, \quad s = s_b, \quad (9)$$

$$\vec{U} = 0, \quad \frac{\partial \tilde{p}}{\partial \vec{n}} = 0, \quad \lambda \frac{\partial s}{\partial \vec{n}} + \gamma s = \psi, \quad (10)$$

Values of physical constants ν, Sc (or Pr in case of thermal convection), ρ_0, β and $\vec{g}$ correspond to properties of the fluid under specified conditions. At first glance, the value of τ could be calculated using its physical interpretation as characteristic hydrodynamic time of the problem under consideration. For example, in a number of cases the hydrodynamic time scale can be assessed as:

$$\tau = \frac{\nu}{U_{ref}^2}, \quad (11)$$

where U_{ref} is some characteristic velocity. Such approximation results in a reasonable solution in terms of balance between accuracy and computational cost.

QHD system differs from the classical hydrodynamic equations by the assumption that the mass-flow rate $\vec{j} = \rho(\vec{U} - \vec{W})$ is not equal to the momentum flux $\rho \vec{U}$, but differs from it by a small quantity $\vec{W}$. The possibility of a difference between momentum flux $\rho \vec{U}$ and mass flow has been considered by a number of authors, and the constructed equation systems were called two-velocity models. QHD system can be regarded as one of the two-velocity family model. In contrast with previous variants, the QHD model is mathematically consistent and convenient for practical applications.

The physical meaning of additional velocity $\vec{W}$, the QHD equations and their mathematical equivalents are closely related. Navier-Stokes (NS) equations for hydrodynamic flows satisfy to five integral laws: conservation laws of mass, momentum, total energy, angular momentum and the law of non-diminution of entropy (second law of thermodynamics). Moreover, the last two laws are a direct consequence of the first three ones. The derivation of these laws for the QGD and the QHD systems as well as proofs of their dissipative properties are given in [26–28]. This allows one to solve only the first three equations of the system, which express conservation of mass, momentum and energy.

It appears that QHD equations with the chosen expression of additional velocity $\vec{W}$ exhibit the same mathematical properties. Existence of the entropy theorem provides the fact that τ-terms bring the additional dissipation. In the present work, a variant of QHD equations for incompressible flows is used and the equation for entropy is transferred to the equation of kinetic energy with a non-negative dissipative function.

A number of theoretical results have been proved about the connection between QHD and NS solutions. One of them states that any exact solution of the steady NS system is an exact solution of the steady QHD system. The simplest of them are the Archimedes law, the Poiseuille parabola, the flow between two vertical walls with different temperatures. Thus QHD and NS systems are closely related and do not contradict each other. Moreover, QHD system can be obtained by averaging the NS system over a small time interval of the order of τ. So this system has the physical meaning of the smoothed NS equations.

Proper mathematical features of QHD system allowed its successful use in numerical applications. The stability of the numerical algorithm is ensured by the significantly non-linear character of τ-dissipation. It plays the role of a regularizing artificial viscosity and replaces the usual limiters of various structures. The adjustment of τ, regarded as a small parameter, determines the quality of the algorithm. This dissipation reduces to zero in zones with steady solutions described by Euler equations. It is convenient to have only one adjustment parameter, in contrast with the commonly used limiters.

The time-explicit schemes based on QHD equations enable to use a central-difference approximation for all space derivatives including the convective terms, which simplifies significantly the numerical implementation.

3. Approximation

QHD equations (2)-(5) together with BCs (6)-(8) are approximated by a finite-volume method for unstructured polyhedral meshes with compact stencil and collocated storage.

The time derivatives of velocity $\vec{U}$ and scalar s in transport equations are approximated using 1st order Euler (12) or 2nd order Adams-Bashforth (13) schemes, depending on the required properties of the numerical solution:

$$\frac{\partial \vec{U}}{\partial t} \approx \frac{\vec{U}^n - \vec{U}^o}{\Delta t}, \quad \frac{\partial s}{\partial t} \approx \frac{s^n - s^o}{\Delta t}, \quad (12)$$

$$\frac{\partial \vec{U}}{\partial t} \approx \frac{1}{\Delta t}\left(\frac{3}{2}\vec{U}^n - 2\vec{U}^o + \frac{1}{2}\vec{U}^{oo}\right),$$
$$\frac{\partial s}{\partial t} \approx \frac{1}{\Delta t}\left(\frac{3}{2}s^n - 2s^o + \frac{1}{2}s^{oo}\right), \tag{13}$$

where Δt is a time step, index ^{n} refers to values at a new time level, index ^{o} refers to values of a quantity at an old (previous to new) time level and index ^{oo} refers to values of quantity at an old-old (previous to an old) time level.

Convection and diffusion terms are approximated using the Gauss approach with a linear interpolation [1,37], which produces a second order central difference scheme on Cartesian rectangular grids. The discrete analog for the Laplacian-type term $\nabla \cdot \frac{\tau}{\rho_0}\nabla \tilde{p}$ of pressure equation reads:

$$\nabla \cdot \frac{\tau}{\rho_0}\nabla \tilde{p} \approx \frac{1}{V}\frac{\tau}{\rho_0}\sum_f |\vec{S}_f|\frac{\delta \tilde{p}}{\delta \vec{n}_f}, \tag{14}$$

where $\frac{\delta \tilde{p}}{\delta \vec{n}_f}$ denotes an approximation of the surface normal derivative of $\tilde{p}$ at the center of face f, $\vec{S}_f$ is the product of the normal $\vec{n}_f$ to face f and its area $|\vec{S}_f|$, V is the volume of the cell, over which the Laplacian operator is discretized.

The surface normal derivative of a field (for example, $\tilde{p}$) is approximated at the center of face f (Fig. 1) as a finite difference between values in adjacent cells P and N with centers $\vec{x}^P$ and $\vec{x}^N$:

$$\frac{\delta \tilde{p}}{\delta \vec{n}_f} = \frac{\tilde{p}^P - \tilde{p}^N}{|\vec{x}^P - \vec{x}^N|}. \tag{15}$$

The approximation of regularization terms, or in other words, τ-terms requires to compute partial derivatives at face centers f, since gradient and divergence differential operators and their combinations are used in flux expressions. While the surface-normal component of differential operators can be approximated using linear interpolation of values in adjacent to face cells centers, the tangential component(s) needs special treatment. Several approaches are considered for the approximation of τ-terms:

1. calculation in cell centers with linear interpolation;
2. reduced method when only the surface-normal derivative is used, while tangential components are neglected [38];
3. least squares method which uses Taylor's expansion to derive discrete expressions for partial derivatives [38];
4. Gauss method applied to a fictitious control volume defined around considered face f [39]. Within this method the computational stencil includes vertices of the face and points in cells adjacent to the face (see Fig. 1). For example, the expression for the x-derivative of scalar field α on a quadrangle face reads:

$$\frac{\partial \alpha}{\partial x} \approx \frac{1}{V_f}\sum_{m=1}^{8} n_{m,x}\alpha_m, \tag{16}$$

where V_f is the volume of the fictitious cell constructed on the face f, m is an index of face in this fictitious cell, α_m is the average of α over face m, $n_{m,x}$ is the x-component of the normal to face m.

The computational experience has shown that the first approach introduces too much error in the numerical solution and produces non-physical oscillations. It was found that approach number 2 generates excessive numerical diffusion [38], while the third approach is very demanding in terms of mesh quality [38]. Thus, the last approach appears to be the most reasonable considering previous practice and successful application [40].

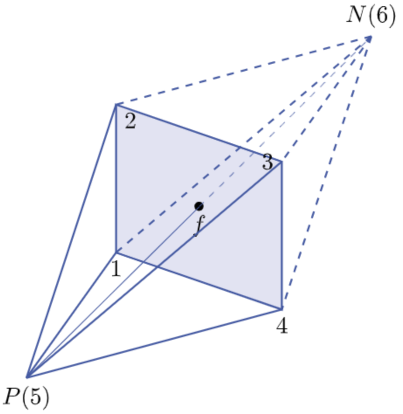


Fig. 1. Geometrical scheme of stencil for numerical computation of partial derivatives on finite volume face f: P denotes cell center with normal of f points outward, N denotes center of cell to which normal of f points inward.

QHD equations (2)–(5) are approximated using a semi-implicit approach. The discretized expressions for convective and τ-terms are derived for the previous time step ^{o}, Laplacian terms are approximated using values from the new time step ^{n}. Finally, the algebraic analogue of the original system consists of:

- algebraic equation (17) for velocity $\vec{U}$ corresponding to momentum equation (2);
- algebraic transport equation (18) for scalar s corresponding to equation (3);
- expression for regularization velocity $\vec{W}^n$ corresponding to equation (19);
- algebraic Poisson equation for pressure perturbation (20), corresponding to pressure equation (5).

$$\begin{aligned}&\frac{\delta \vec{U}}{\delta t} + \frac{1}{V}\sum_f \vec{S}_f \cdot \left(\vec{U}^o - \vec{W}^n\right)_f \otimes \vec{U}^o_f \\ &\quad - \frac{1}{V}\sum_f \nu_f \frac{\delta \vec{U}^n}{\delta \vec{n}_f}|\vec{S}_f| - \frac{1}{V}\sum_f \nu_f \vec{S}_f \cdot [\nabla \vec{U}^o]^T_f \\ &\quad - \frac{1}{V}\sum_f \vec{S}_f \cdot \left(\vec{U}^o \otimes \vec{W}^n\right)_f = -\frac{1}{\rho_0}\frac{1}{V}\sum_f \tilde{p}^n_f \vec{S}_f + \vec{F}^o,\end{aligned} \tag{17}$$

$$\begin{aligned}&\frac{\delta s}{\delta t} + \frac{1}{V}\sum_f \vec{S}_f \cdot \left(\vec{U}^o - \vec{W}^n\right)_f s^o_f - \frac{1}{V}\sum_f \frac{\nu_f}{Sc}\frac{\delta s^n}{\delta \vec{n}_f}|\vec{S}_f| - \\ &\quad - \frac{1}{V}\sum_f \vec{S}_f \cdot \left(\tau_f \vec{U}_f(\vec{U}_f \cdot [\nabla s^o]_f)\right) = 0,\end{aligned} \tag{18}$$

$$\vec{W}^n_f = \tau_f\left(\vec{U}^o_f \cdot [\nabla \vec{U}^o]_f + \frac{1}{\rho_0}[\nabla \tilde{p}^n]_f - \vec{F}^o_f\right), \tag{19}$$

$$\frac{1}{V}\sum_f \frac{\tau_f}{\rho_0}\frac{\delta \tilde{p}^n}{\delta \vec{n}_f}|\vec{S}_f| = \frac{1}{V}\sum_f \vec{S}_f \cdot \left(\vec{U}^o_f - \tau_f(\vec{U}^o_f \cdot [\nabla \vec{U}^o]_f) + \tau_f \vec{F}^o_f\right), \tag{20}$$

where $\frac{\delta}{\delta t}$ denotes an approximation of time derivatives (Euler or Adams-Bashforth, for example), $\frac{\delta}{\delta \vec{n}_f}$ denotes an approximation of surface-normal derivatives and square brackets $[\cdot]_f$ denote an approximation of the enclosed operator at the center of face f, which can be done by means of any previously mentioned method (reduced, least squares, etc.).

The method can be fully explicit if one approximate the terms with diffusion in explicit manner and it can be implemented using the standard OpenFOAM tools.

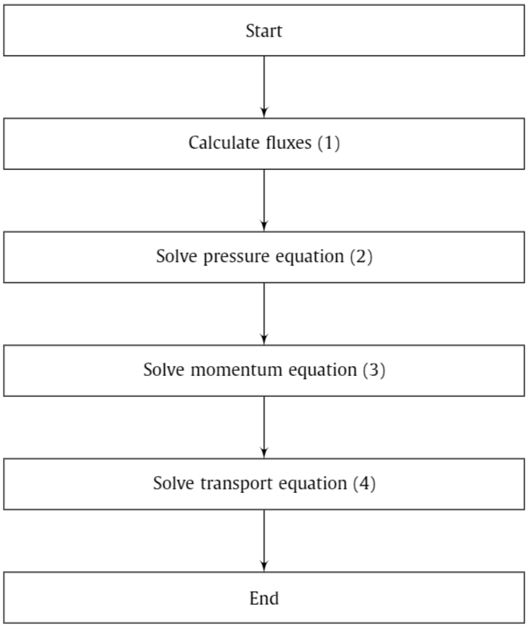

Start
Calculate fluxes (1)
Solve pressure equation (2)
Solve momentum equation (3)
Solve transport equation (4)
End

Fig. 2. QHD algorithm flowchart.

Assessment of τ value (as mean free time in case of gases or as relaxation time in case of liquids) gives range from $\sim 10^{-10}$s for dry air at atmospheric conditions to $\sim 10^{-13}$s for water at similar conditions, which makes the theoretical definition of τ impractical for real numerical simulation problems. However, the regularization parameter τ can be considered as a tuning coefficient of the numerical model, which introduces a controllable additional diffusion and damps numerical oscillations and instabilities. In this case, the value of τ can be determined by characteristic flow time $\sim \nu/(\vec{U} \cdot \vec{U})$, or, in a non-dimensional form involving Reynolds and Grashoff numbers. Some considerations on the choice of τ for compressible flows are given in [24].

For incompressible flows, determination of τ generally involves the following steps:

1. Make a coarse estimate, e.g., using expression $\tau = \frac{\nu}{\vec{U} \cdot \vec{U}}$ and relation $\tau \sim Re^{-1}$ or $\tau \sim Gr^{-1}$, etc.;
2. Make a first calculation with given time step and spatial mesh resolution and check that the solution is smooth. If not, then increase value of τ.
3. Decrease τ gradually to check the convergence of the numerical algorithm (refinement and sensitivity study).

The stability condition for QHD algorithms is obtained by analogy with that for gas dynamic algorithms (cf. [28]) and numerical practice. It was obtained empirically that the constant c_τ^{-1} cannot be larger than $1/2$. Even smaller values must be used for strongly distorted grids.

Details of the general procedure for some typical flows problems are presented in Section 5.

4. Implementation

The algebraic system of equations (17) – (20) approximating the QHD system (2) – (5) has been implemented into the OpenFOAM [41] software as an application (*QHDFoam*), a part of *QGDsolver* [42].

Most OpenFOAM applications use segregated approaches (such as PISO, SIMPLE and their variants) with a sequential solution of algebraic balance equations (momentum, pressure and others). For the numerical integration of the discretized QHD system, the similar approach is used. Each of equations (17) – (20) is solved separately (Fig. 2). If we compare the QHD algorithm (Fig. 2) with the dominating algorithm employed in OpenFOAM, namely PIMPLE (mixed PISO-SIMPLE, Fig. 3) algorithm, the first one has a number of merits:

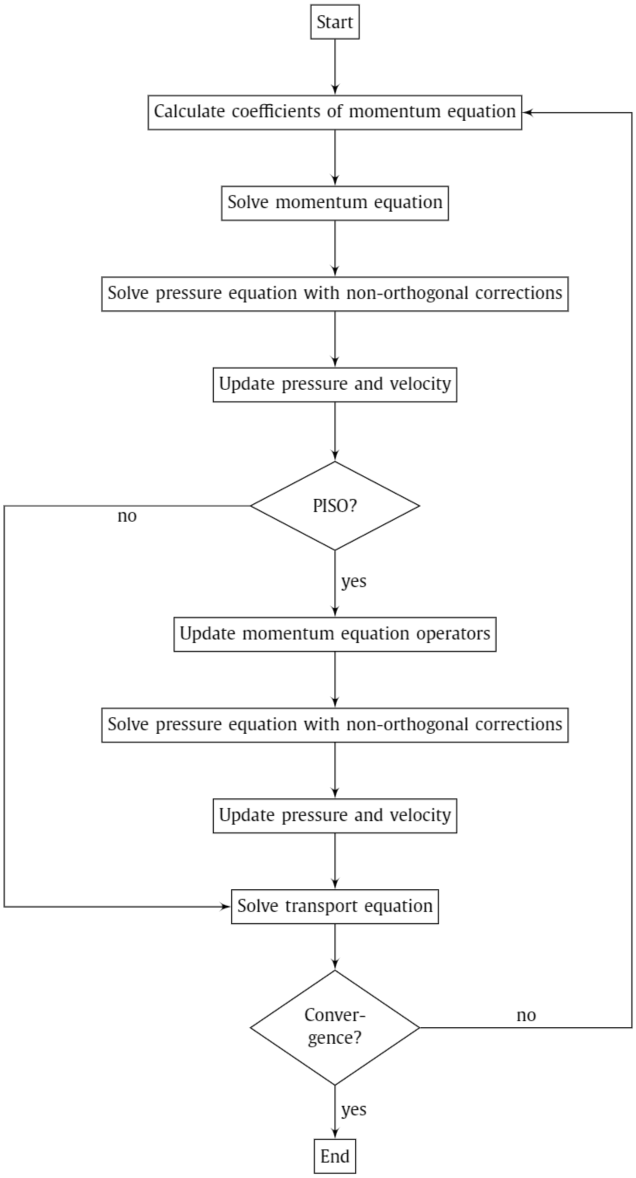

Start
Calculate coefficients of momentum equation
Solve momentum equation
Solve pressure equation with non-orthogonal corrections
Update pressure and velocity
PISO?
no
yes
Update momentum equation operators
Solve pressure equation with non-orthogonal corrections
Update pressure and velocity
Solve transport equation
Conver-gence?
no
yes
End

Fig. 3. PIMPLE algorithm flowchart.

- The QHD algorithm does not require additional pressure-velocity loops, as used in PIMPLE to couple momentum and continuity equations;
- The QHD algorithm does not use non-orthogonal corrections for skewed and non-orthogonal meshes;
- The changes in the pressure equation due to source terms in the momentum equation (buoyancy, Coriolis force, mesh motion, etc.) are devised naturally from the regularization procedure of the QHD approach, while in PISO-SIMPLE methods such changes require additional loops of coupling or a special treatment.

The above-listed advantages of the QHD algorithm allow connecting the developed hydrodynamic application to many of the standard OpenFOAM modules (such as turbulence models, mesh motion, etc.) with minimal efforts. The final algorithm for the numerical integration of QHD equations includes following steps:

1. Interpolation of hydrodynamic fields from cell centers to cell faces – pressure, velocity, temperature, viscosity, body forces, etc.
2. Explicit calculation of differential operators to be used in expressions for face fluxes (gradient, divergence and their combinations).
3. Calculation of convective, diffusive and regularization fluxes.
4. Verification of stability criteria $|\vec{U}|\frac{\Delta t}{\Delta x}$ and $\Delta t < \tau \cdot C_{\tau}^{-1}$ and adjustment of time step Δt to meet specified restrictions.
5. Advancement in time $t^n = t^o + \Delta t$.
6. The application of a user-selectable turbulence model.
7. Calculation of pressure field from equation (20).
8. Calculation of regularization velocity $\vec{W}^n$ using expression (19).
9. Calculation of velocity field from equation (17).
10. Calculation of scalar quantity field from equation (18).

As it was mentioned above, the explicit calculation of differential operators can be based on various numerical methods. Thus, the new module for the approximation of τ-terms was developed within namespace `fvsc` to mimic OpenFOAM API conventions on functions for discretization (namespaces `fvm` and `fvc`). The new namespace includes next functions:

- `fvsc::grad(...)` for the calculation of scalar and vector field gradients at face centers of a finite volume mesh;
- `fvsc::div(...)` for the calculation of vector and tensor field divergence at face centers of a finite volume mesh.

An example of algebraic momentum equation (17) assembled using OpenFOAM API and newly created functions is presented in the listing of Fig. 4. According to OpenFOAM standarts, the particular numerical method for the calculation of pressure gradient at face centers (variable `gradPf`) is set in file `fvSchemes` (see listing in Fig. 5 for example). Parameters to control QHD/QGD numerical algorithms (such as formulae for τ calculation, reference values, etc) are set in the dedicated section of fvSolution dictionary, Fig. 6.

```
gradPf = fvsc::grad(p);
Wf = tauQGDf*((Uf & gradUf) + gradPf/rhof - BdFrcf);
surfaceVectorField phiUfWf = mesh.Sf() & (Uf * Wf);
phiUf -= phiUfWf;

{
solve
    (
        fvm::ddt(U)
        +
        fvc::div(phiUf)
        -
        fvm::laplacian(muf/rhof,U)
        -
        fvc::div(muf/rhof * mesh.Sf()
        & qgdInterpolate(Foam::T(fvc::grad(U))))
        ==
        -
        fvc::grad(p)/rho
        +
        BdFrc
        +
        USu
    );
}
```

Fig. 4. Example of OpenFOAM source code for momentum equation assembling.

```

fvsc
{
    default    GaussVolPoint;
}
```

Fig. 5. Example of OpenFOAM settings file *fvSchemes* for user run-time selection of τ-terms approximation method.

```

QGD
{
    pRefCell        0;
    pRefValue       0;
    implicitDiffusion true;
    QGDCoeffs constTau;
    constTauDict
    {
        Tau 0.005;
    }
}
```

Fig. 6. Example of OpenFOAM settings file *fvSolution* for user run-time selection of QHD algorithm parameters.

5. Numerical experiments

Till now, the QHD algorithm has been extensively studied and validated using in-house programs implementing the finite volume method with Cartesian orthogonal grids and central-difference approximations. These studies have demonstrated high accuracy and convergence of the QHD numerical algorithm for many problems, such as steady and transient isothermal flows in a cavity, behind a backward-facing step, around a cylinder. Non-isothermal problems have been also considered, such as the flow in a vessel with heated walls in Boussinesq assumption, Marangoni convection, etc. Results of QHD algorithm applied to these problems are published in, e.g. [26,28,43,44].

In contrast to previous implementations, the OpenFOAM-based approximation of QHD equations allows extending substantially the area of their application. The cases selected in this section demonstrate the following features of the present QHD algorithm implementation in OpenFOAM:

- simulation of viscous flows;
- sensitivity to the Reynolds number;
- ability to resolve flows with steep velocity gradients without flux limiters;
- simulation of buoyancy-driven flows;
- support for meshes with non-orthogonal cells;
- transient complex 2D flows and convergence of the numerical solution to the reference one.

Problems presented below were used to verify the QHD numerical algorithm implementation, as well as to demonstrate its capabilities and its distinctions from the PISO algorithm (OpenFOAM) and spectral element method (Nek5000). These cases are:

- laminar flow over a backward facing step;
- laminar flow in a cavity with heated walls;
- laminar flow in a skewed cavity;
- generation of internal gravity wave attractor in a vertically stratified medium.

The last case is chosen since its simulation requires of all above-listed features. A well-known tool for the accurate simulation of such complex transient flows is the Nek5000 code, which is based on spectral elements. Several previous works [45,46] showed that Nek5000 is capable to predict results of physical experiments

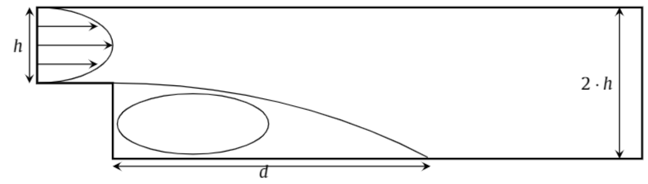


Fig. 7. Sketch of problem setup for the backward step laminar flow.

in a simple geometry with a relatively simple case setup. However, for treating complex geometrical configurations or when accounting for additional phenomena (such as particle motion), coarser approaches (e.g., Finite Volume Method) provide useful means for first estimates. In this context, the QHD algorithm must be verified for the above-mentioned features as well as it must demonstrate the convergence of its results toward reliable reference data.

5.1. Laminar flow over a backward-facing step

Verification of a numerical algorithm for viscous flows usually involves several stages of simulation: near-plate boundary layer development, Poiseuille flow in a channel, cavity flow, flows with abrupt area change, turbulent flows, etc. The problem of a laminar flow over a backward-facing step [47] can be considered as an integral case, which allows to check the modeling of several important features of a fluid flow, such as viscous slowdown near the wall, Poiseuille flow, separation and reattachment. This case was used for the assessment of many incompressible codes [48–50].

The sketch of the problem is shown in Fig. 7: a laminar flow with parabolic profile enters the domain at the upper left vertical border and leaves the channel through the right vertical border. The flow is isothermal and it is described by equations (2), (3), (4) and (5) with $\vec{F} = 0$.

Due to the presence of an abrupt area change after the inlet channel, the separation zone after the backward step evolves, forming a reversed flow region. In the provided experimental data [47], the length d of this separation zone depends on the Reynolds number $Re = \frac{hU_{in}}{\nu}$, where h is the height of the inlet channel, U_{in} is the average inlet velocity, ν is the kinematic viscosity of the medium. The ability of a numerical method to find the dependence $d = d(Re)$ shows the quality of the viscous flow solution and demonstrates the resolution accuracy of the interaction between viscous and inertial forces in the assessed simulation program. Several values of Re number were considered: 100, 200 and 400.

Boundary conditions were imposed as follows:

- inlet boundary condition with prescribed velocity profile on the upper left vertical border, fixed zero pressure gradient according to (6);
- outlet BC with zero gradient for velocity and fixed value of pressure on the right vertical boundary according to (7);
- no-slip wall boundary conditions on other boundaries of the computational domain according to (8).

Following settings of the numerical algorithm were used for simulations:

- mesh resolution with 20 intervals per step length (according to converged values for other simulation programs, e.g. [28]);
- values of $\tau = 0.005, 0.0025, 0.00125$ for $Re = 100, 200, 400$ respectively [28];
- time step $\Delta t = \frac{1}{2}\tau$;
- Euler 1st order scheme for temporal terms;
- central differences scheme for spatial terms with convective and diffusive fluxes;
- Gauss scheme for $\tau-$terms.

Table 1
Length of separation zone behind a backward facing step calculated using QHDFoam and other approaches.

	$Re = 100$	$Re = 200$	$Re = 400$
QHDFoam	5.0	8.25	11.9
QHD [28]	5.0	8.2	14.8
Sparrow E. M. and Chuck W. [62]	5.0	7.5	-
Kim J. and Moin P. [63]	5.0	8.3	12
Hackman L. P. et al. [64]	5.0	8.5	-
Armaly B. F. et al. [47]	5.0	8.5	14.2

The value of regularization parameter τ in this case is inversely proportional to the Reynolds number: $\tau \sim Re^{-1}$.

The numerical simulations have been carried out until the steady-state solution was reached. The comparison of separation zone length d calculated with *QHDFoam* versus reference data from the physical experiment [47] and versus numerical experiments of other authors is presented in Table 1. Flow streamlines colored with velocity magnitude are shown in Fig. 8. It can be seen from Table 1 that the developed numerical algorithm is capable of treating viscous flows with separation and it is sensitive to the Reynolds number of the problem. The relatively large difference at $Re = 400$ can be related with the intrinsic numerical diffusion in the present OpenFOAM implementation. However, such a behavior is typical of most algorithms [50] and only few approaches, e.g. [50] show a good agreement for Re larger than 300. All aforementioned considerations allow concluding that the implementation of the numerical method for incompressible laminar flows is correct, while its accuracy for viscous flow is comparable to that of many conventional CFD methods (such as SIMPLE, PISO, etc).

5.2. Natural convection in a square cavity

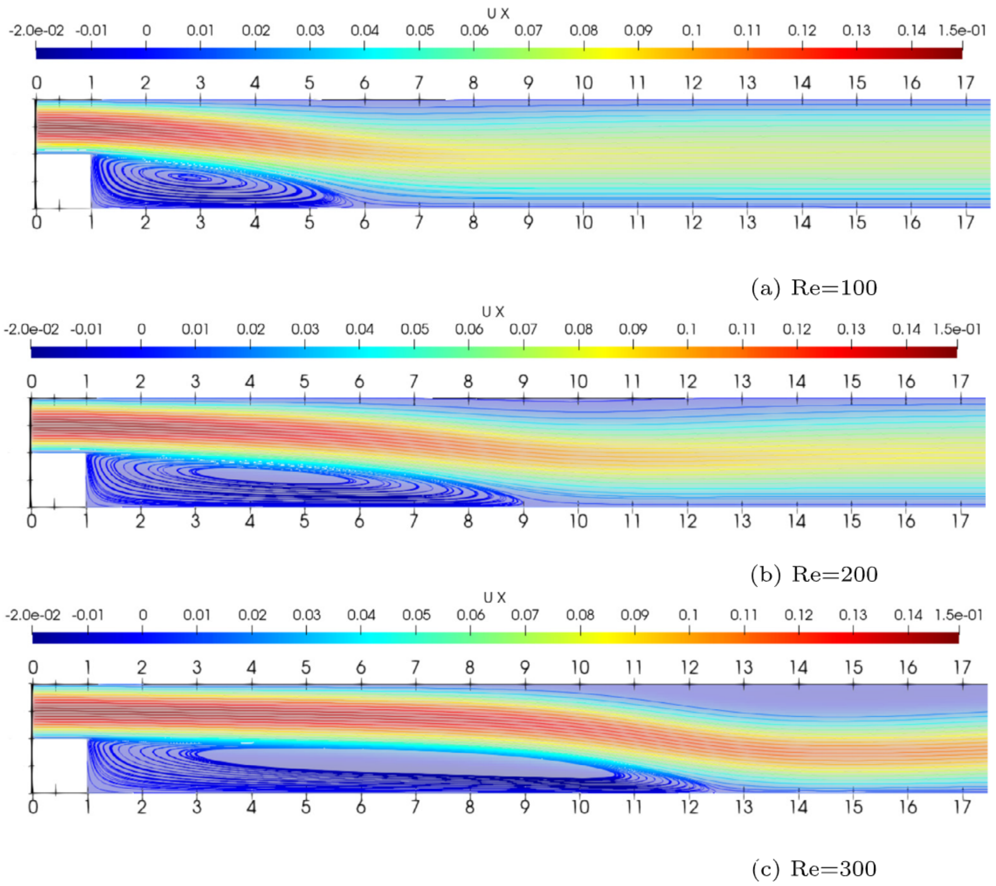
The second test case for the QHD algorithm is the laminar flow induced by natural thermal convection [51,52]. The case allows verifying correctness of buoyancy, convection and viscous terms approximations in the algorithm. The secondary objective of this numerical experiment is the demonstration of the procedure for choosing the regularization parameter τ in flows driven by natural convection.

The sketch of the case under consideration is presented on Fig. 9. The flow is described by the system (2), (3), (4) and (5) with temperature T as transported scalar s and Pr number instead of Sc.

The two-dimensional liquid domain is enclosed by square impermeable boundaries with height L: constant temperature T_0 is specified at the left vertical boundary, constant temperature T_1 is specified at the right vertical boundary and adiabatic conditions are imposed on horizontal boundaries. The positive value of $\Delta T = T_0 - T_1 > 0$ creates a stationary vortex with upward motion near the left boundary and downward motion near the right boundary. The gravity vector g points vertically downward.

The Grashof number of the problem $Gr = \beta g \Delta T L^3/\nu^2 = 10^4$, indicates a stable laminar regime with a steady-state solution. According to general recommendations [28] the value of τ in convection driven flows should be proportional to Gr^{-1}. After several iterations, the final value of τ was chosen to be 10^{-4}. All other settings of the numerical algorithm (temporal and spatial derivatives approximation schemes, time step restriction $\Delta t = \frac{1}{2}\tau$) are identical to those of the previous problem.

The resulting streamlines are shown in the Fig. 10. Quantitative results of the simulation (maximum horizontal and vertical velocities) for different spatial grids are presented in Tables 2 and 3. The comparison of calculated values at different grid resolutions



(a) Re=100

(b) Re=200

(c) Re=300

Fig. 8. Streamlines for a backward facing step flow colored with the horizontal velocity component.

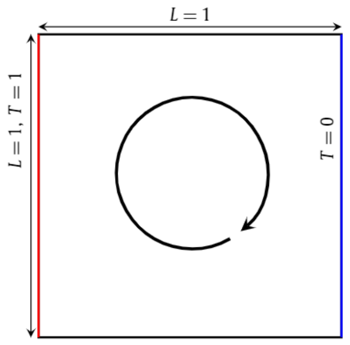


Fig. 9. Sketch of problem setup for the natural convection in a heated cavity.

shows spatial convergence, as well as perfect match to reference solutions.

5.3. Laminar flow in a skewed cavity

To study the behavior of the *QHDFoam* solver with a highly distorted computational mesh, the case of laminar flow in the deformed cavity is considered [53,54]. The comparison of the results with reference data [54] and with the simulation results from an OpenFOAM implementation of PIMPLE (mixed PISO/SIMPLE) allows highlighting the main features of the QHD numerical algorithm as well as demonstrating the general procedure for adjusting the regularization parameter τ.

The sketch of the computational domain is presented in Fig. 11. The flow is isothermal and it is described by equations (2), (3), (4) and (5) with $\vec{F} = 0$.

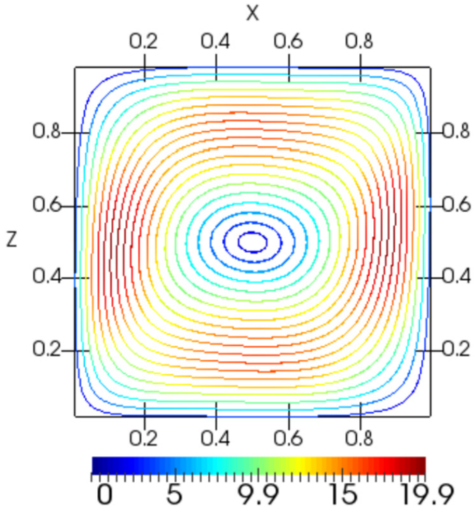


Fig. 10. Flow streamlines in a heated cavity colored with velocity magnitude.

Table 2
Maximum horizontal velocity for the natural convection a in cavity.

Mesh	U_x [28]	U_x [52]	U_x *QHDFoam*
20 × 20	15.938	16.144	16.040
40 × 40	16.005	16.262	16.410
80 × 80	16.070	16.219	16.225

A uniform tangential velocity $\vec{U}_b$ is specified at the horizontal upper border, while the condition of impermeable no-slip wall is

Table 3
Maximum vertical velocity for the natural convection in cavity.

Mesh	U_y [28]	U_y [52]	U_y *QHDFoam*
20 × 20	19.513	19.363	19.670
40 × 40	19.663	19.602	19.910
80 × 80	19.663	19.648	19.757

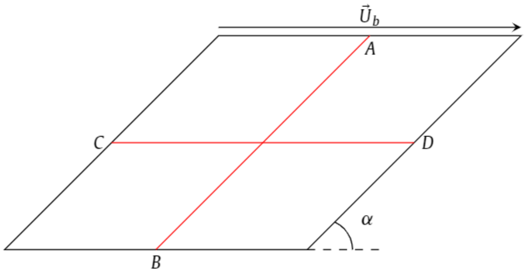


Fig. 11. Sketch of problem setup for laminar flow in a skewed cavity.

imposed on other borders. All BCs are described by (8). Reynolds number of the problem is calculated as $Re = \frac{|\vec{U}_b|L}{\nu}$, where $L = 1$ – length of upper horizontal border and $|\vec{U}_b| = 1$.

The following parameters are varied to study the properties of the QHDFoam algorithm:

- mesh density (20×20, 40×40, 80×80 and 512×512 computational points per width×height)
- time step Δt;
- value of regularization parameter τ;
- Reynolds number $Re = \{100, 1000\}$;
- cavity skewness angle $\alpha = \{45°, 30°, 15°\}$.

Other settings of the numerical algorithm are identical to the used for previous cases. The simulations are carried out till the steady-state solution is reached – in terms of computational time it corresponds to several circulation cycles of a liquid particle in the domain. After that, the x-component of velocity field on the line A-B and the y-component of velocity field on line C-D are extracted from the computational flow field and are compared with results, obtained at different grid resolutions, with different settings or with different methods ([54,53] and standard OpenFOAM solver *pimpleFoam*).

As a first step, the cavity with the largest angle $\alpha = 45°$ (which corresponds to the smallest mesh distortion) is simulated.

Before choosing the value of τ three considerations should be taken into account:

1. the larger is τ, the more diffusion is introduced into the numerical scheme and consequently, the more oscillations (including non-physical) are filtered out;
2. for the explicit scheme, the upper limit of τ is determined by the time step Δt: $\tau|\vec{U}|^2 \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$;
3. the lower limit of τ is determined by computational resources, since the time step Δt is restricted as $\Delta t \leq C_\tau^{-1}\tau$, where $C_\tau \geq 2$.

Therefore, the optimal value of τ should provide numerical solution which is:

- free of spurious non-physical oscillations;
- close enough to the actual (reference) solution with some predefined tolerance;
- stable during the time evolution;
- consuming reasonable computational resources.

Table 4
Influence of τ on the solution for the laminar $Re = 1000$ incompressible flow in the skewed cavity with angle $\alpha = 45°$.

τ	L_1	$L_0/max(U_y(\tau = 0.004))$
0.005	0.00022	0.0028
0.01	0.00133	0.0177
0.02	0.00361	0.0515
0.04	0.00829	0.1221
0.08	0.01750	0.2840
0.16	0.02795	0.5169

The first estimate of τ can be assessed as $\tau \approx \nu/|\vec{U}_b|^2$ resulting in $\tau \approx 0.01$ for $Re = 100$ and $\tau \approx 0.001$ for $Re = 1000$.

The case with $Re = 1000$ is more prone to oscillations (since physical viscosity is smaller), therefore we start with $\tau = 0.001$. Then we must determine value of C_τ constant. For 2D cases with orthogonal grids $C_\tau = 2$, but for meshes with angle $\alpha = 45°$, the simulation diverges with $C_\tau = 2$ and a stable regime can be obtained $C_\tau \geq 4$. Now, keeping time step $\Delta t = 10^{-3}$ constant and increasing value of τ, it is possible to study how the latter one influences the solution. For example, in this study we took values of $\tau =$ 0.004, 0.005, 0.01, 0.02, 0.04, 0.08 and 0.16. For the last regime $\tau = 0.16$, the stability criterion $\tau|\vec{U}|^2 \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$ was violated, therefore Δt was reduced by a factor of 2.

It can be seen from Fig. 12, that for τ in range from 0.004 to 0.01, the difference in the numerical solution is negligible, while values of τ larger than 0.01 produce a very significant dissipation. With τ less than 0.004, the time step should drop below 0.001, which increases computational efforts to levels that are not justified by the increase of accuracy. The choice of τ can be justified by comparing metrics of the solution. For example, the following quantities are compared in Table 4:

- approximation of L_1 norm between solution with $\tau = 0.004$ and solutions with other values of τ;
- approximation of L_0 norm, calculated as the difference between maximum values of U_y at different values of τ, normalized by U_y: $|max(U_y(\tau_i)) - max(U_y(\tau = 0.004))|/max(U_y(\tau = 0.004))$.

It can be seen from Table 4 that the difference between the solutions with $\tau = 0.01$ and $\tau = 0.004$ is less than 2%, which means that a further reduction of $\tau < 0.004$ is not necessary if the accuracy of a few percent is acceptable.

As it is seen from Fig. 13, the results of QHD simulations with $\tau = 0.005$, the results of PISO simulations and reference velocity fields [54] for geometry for $\alpha = 45°$ perfectly coincide with one another for grids finer than 40×40. It is worthwhile to note, that PISO algorithm needs correction loops and flux limiters for higher Re number, while the QHD algorithm does not require these techniques. Here results are presented only for $Re = 100$, but simulations for $Re = 1000$ show a similar behavior.

Results of QHD and PISO simulations and their comparison with reference data [54] for $\alpha = 30°$ are shown in Figs. 14 and 15. While QHD and PISO methods converge toward the reference solution in a different way, they eventually fall almost on the same curve for grid 80×80. The only difference in QHD settings between $\alpha = 45°$ and $\alpha = 30°$ is the value of the regularization parameter – a larger non-orthogonality of the mesh requires larger values of τ, namely $\tau = 0.007$ is needed for a stable simulation with $\Delta t = 10^{-3}$ and $\alpha = 30°$.

In accordance with previous considerations about the choice of τ, C_τ and Δt, an increase of mesh non-orthogonality leads to an increase of C_τ. In other words, mesh distortions can be treated in the current QHD algorithm implementation:

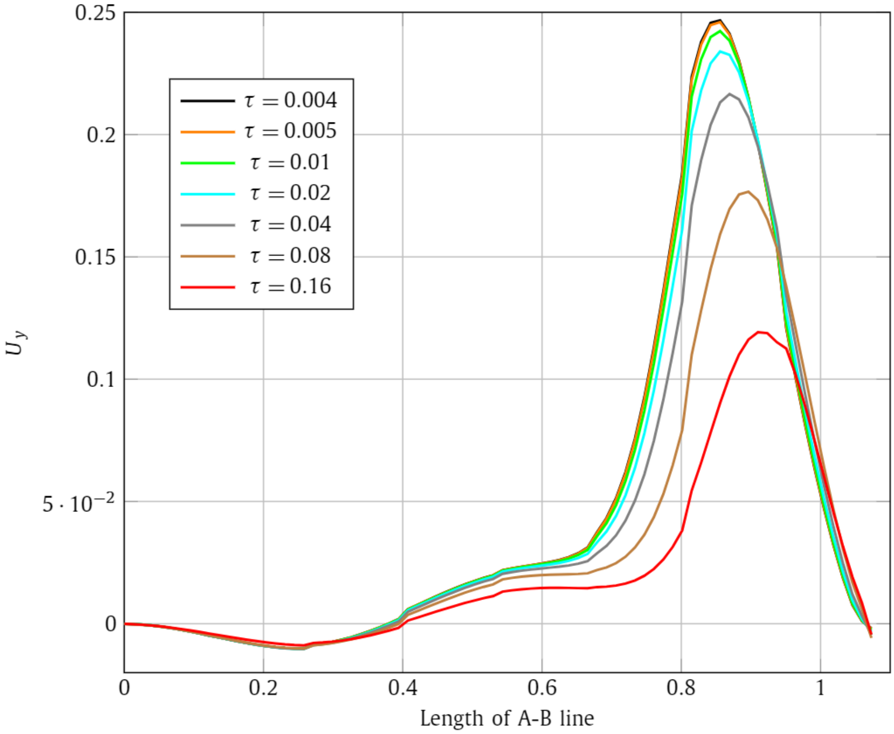


Fig. 12. Influence of regularization parameter τ on the numerical solution (vertical component of velocity on line A-B) for the laminar incompressible flow with $Re = 1000$ in the skewed cavity with angle $\alpha = 45°$.

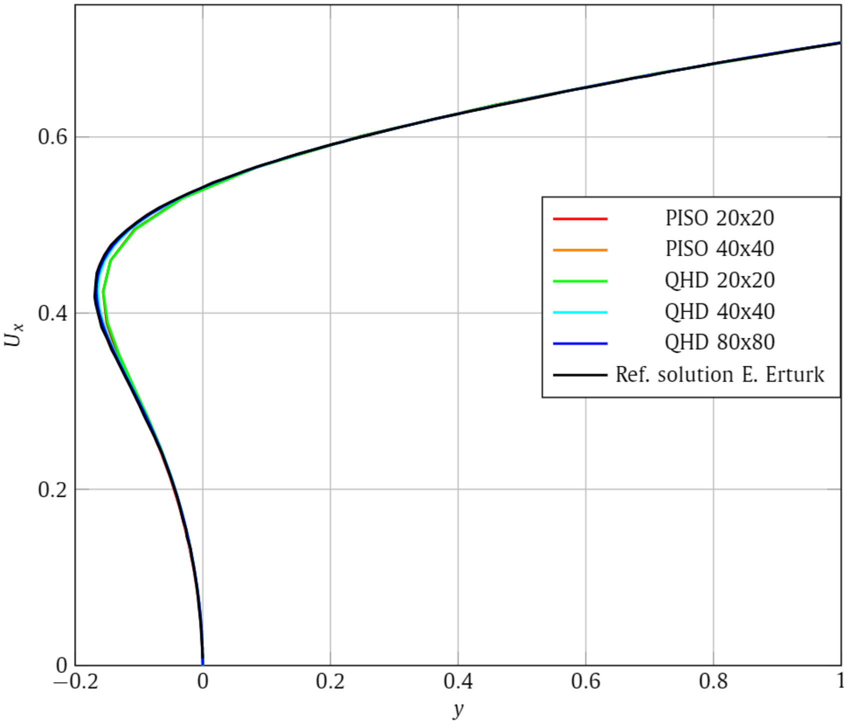


Fig. 13. Grid convergence of QHD and PISO algorithms for the flow in skewed cavity: Re=100, $\alpha = 45°$, values of U_x vs y along AB line.

1. by increasing the numerical diffusion for a given time step;
2. by decreasing time step maintaining the accuracy.

The expected degradation of accuracy due to increase of τ can be assessed by analyzing the solution for different values of the regularization parameter. For example, it was concluded from the previous analysis for $\alpha = 45°$, that increasing τ from 0.005 to 0.007 does not lead to an error larger than 2% (see Table 4).

The above conclusion can be further demonstrated on the case with a larger non-orthogonality, namely $\alpha = 15°$. Results of simulations using the QHD approach for different meshes and values of τ and Δt, their comparison with PISO method and reference data [54] are presented in Figs. 16, 17, 18. Convergence of the

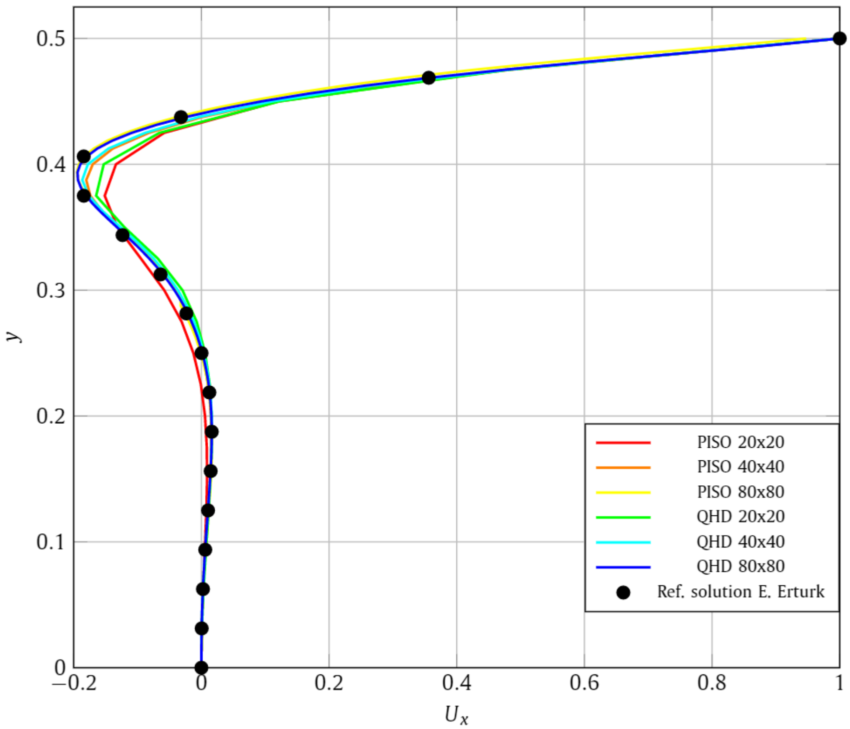


Fig. 14. Grid convergence of QHD and PISO algorithms for the flow in skewed cavity: Re=1000, $\alpha = 30°$, values of U_x vs y along AB line.

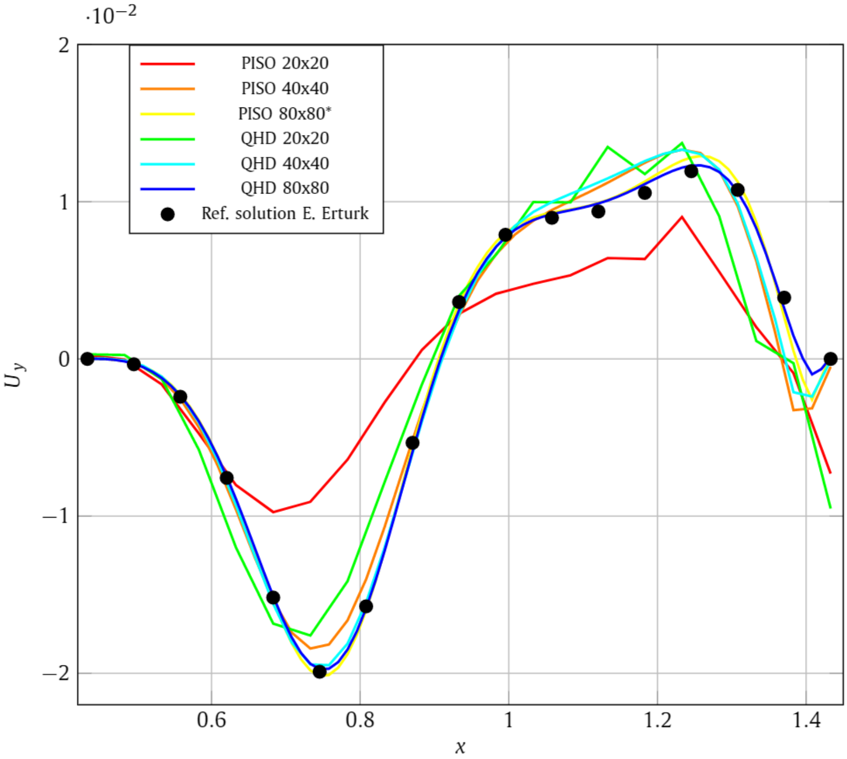


Fig. 15. Grid convergence of QHD and PISO algorithms for the flow in skewed cavity: Re=1000, $\alpha = 30°$, values of U_y vs x along CD.

QHD algorithm at $\alpha = 15°$ is achieved only for $C_\tau = 24$, which yields $\tau = 0.024$ if $\Delta t = 10^{-3}$. The introduction of diffusion with coefficient $\tau = 0.024$ produces a relatively large error, which reduces with decreasing τ – see Fig. 18. A quantitative analysis of mesh convergence for $\tau = 0.005$ is presented in Fig. 19. The calculated order of the *QHDFoam* numerical algorithm on the highly distorted mesh appears to be between the 1^{st} and the 2^{nd}. The accurate solution is obtained on the finest grid 512 × 512 and is verified versus reference data [54] – see Table 5. Except two points ($x \approx 0.607962913$ and $x \approx 0.857962913$), where the absolute value of U_y is close to 0, the maximum relative difference between reference data [54] and QHD algorithm does not exceed 4%, which indicates that τ-convergence is nearly reached.

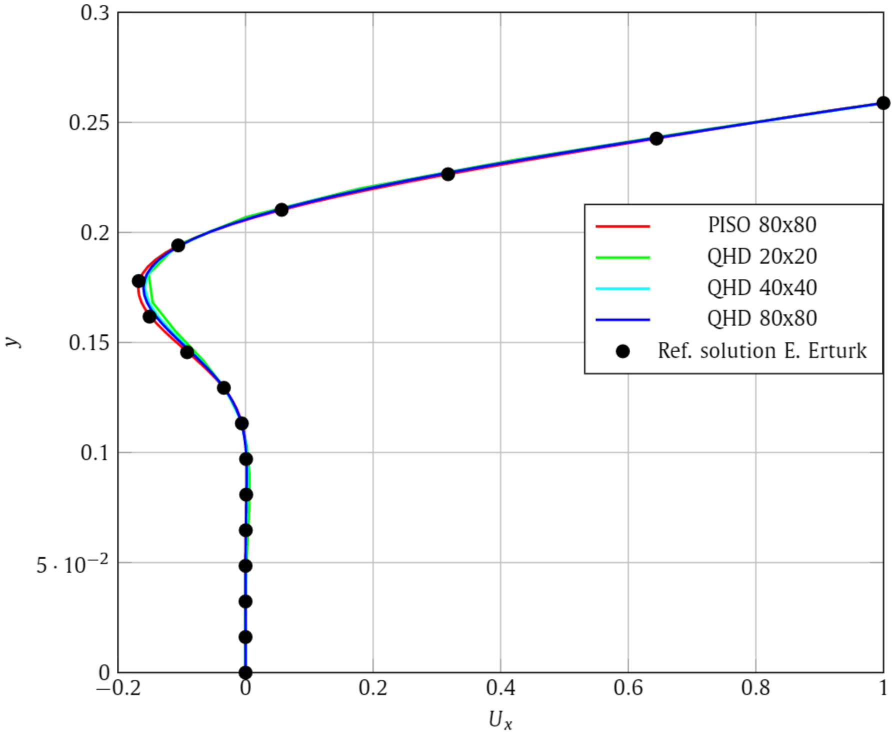


Fig. 16. Grid convergence of QHD and PISO algorithms for the flow in skewed cavity: Re=1000, $\alpha = 15°$, values of U_x vs y along AB.

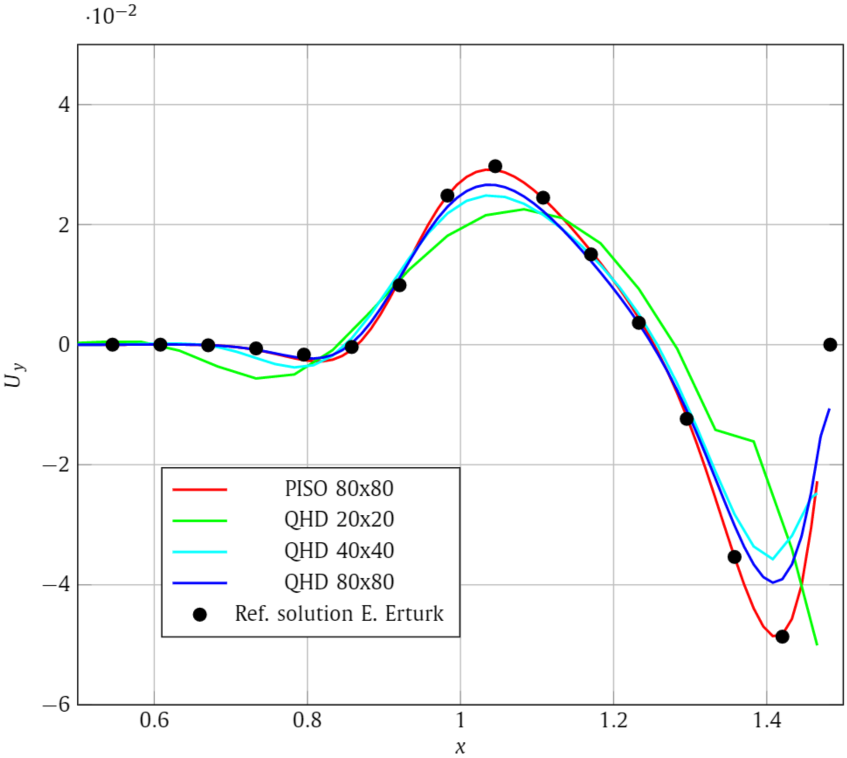


Fig. 17. Grid convergence of QHD and PISO algorithms for the flow in skewed cavity: Re=1000, $\alpha = 15°$, values of U_y vs x along CD.

Finally, the following values of proportionality coefficient C_τ between τ and Δt can be suggested for stable simulations on distorted meshes:

- $\alpha = 90° - C_\tau \geq 2$;
- $\alpha = 45° - C_\tau \geq 4$;
- $\alpha = 30° - C_\tau \geq 7$;
- $\alpha = 15° - C_\tau \geq 24$,

where angle $\alpha = 90°$ corresponds to fully Cartesian mesh (all edges are aligned along X, Y or Z axes).

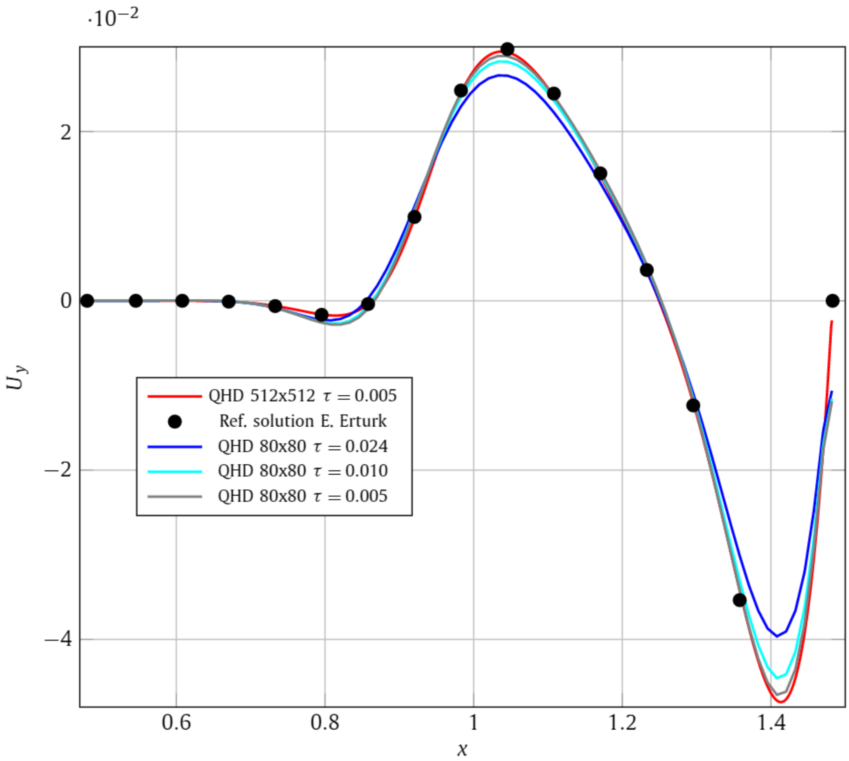


Fig. 18. Influence of regularization parameter τ on the numerical solution (vertical component of velocity on line C-D) for the laminar $Re = 1000$ incompressible flow in a skewed cavity with angle $\alpha = 15°$.

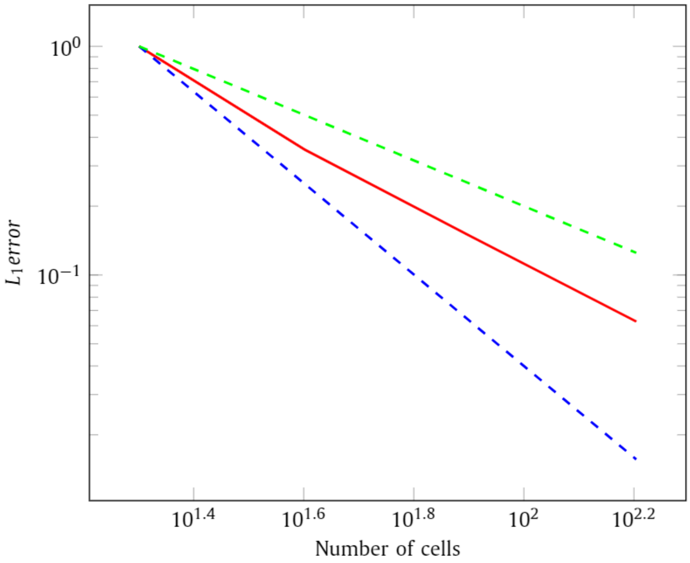


Fig. 19. QHDFoam mesh convergence for the laminar flow in a skewed cavity, Re=1000, $\alpha = 15°$: estimation of L_1 error vs. number of cells (red line), first order of approximation (green line), second order of approximation (blue line). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

5.4. *Generation of internal gravity waves attractor*

The generation of internal gravity waves attractor is a complex phenomenon that occurs after multiple reflections of internal waves from the walls of a reservoir (see Fig. 20). The internal waves are induced by periodic disturbances on one of the external boundaries of a vessel. Since the reason for attractor generation is a combination of several factors (inclination angle of tilted border, stratification gradient, frequency and amplitude of perturbation), then the successful solution of the wave attractor requires an accurate simulation of incompressible viscous flows with buoyancy on curvilinear meshes. Experimental observations of the phenomenon and conditions for its genesis are presented in [55].

The flow is described by equations (5), (2) and (3) with $\vec{F} = \beta\vec{g}\tilde{s}$. The QHD algorithm is used to study the evolution of an internal gravity waves attractor. Two regimes are considered: 1) monochromatic, with a single period of harmonic perturbation and 2) biharmonic, with a perturbation acting as the sum of two harmonic waves.

Both regimes are calculated using the geometry shown in Fig. 20 with the following boundary and initial conditions:

Table 5
Absolute and relative differences between results obtained with *QHDFoam* and reference solution by E. Erturk, B. Dursun [54] for the problem of laminar flow in a skewed cavity.

x	ΔU_y	$\Delta U_y/U_y$
0.482962913	0	0
0.545462913	-1.2049×10^{-08}	-0.0078
0.607962913	-3.7708×10^{-07}	-2.5139
0.670462913	4.7865×10^{-07}	-0.0047
0.732962913	1.2069×10^{-05}	-0.0192
0.795462913	4.4453×10^{-05}	-0.02682
0.857962913	2.7624×10^{-05}	-0.0722
0.920462913	-1.4765×10^{-04}	-0.0149
0.982962913	-2.8405×10^{-04}	-0.011
1.04546291	-3.7823×10^{-04}	-0.0095
1.10796291	-3.4155×10^{-04}	-0.0154
1.17046291	-2.6265×10^{-04}	-0.0174
1.23296291	-1.3046×10^{-04}	-0.0358
1.29546291	3.0913×10^{-04}	-0.025
1.35796291	1.1181×10^{-03}	-0.0316
1.42046291	1.5845×10^{-03}	-0.0325
1.48292737	0	0

- On the right tilted border, the top and the bottom horizontal borders a no-slip boundary condition is imposed on velocity and a fixed normal gradient is applied to pressure and salinity (with $\lambda=1$, $\gamma=0$ and $\psi=\vec{n}\cdot\frac{\partial s(0,\vec{x})}{\partial t}$), according to (8);
- On the left vertical border, the switchable and continuity preserving inlet/outlet condition is imposed with a velocity given by a space-periodical and time-periodical function $\vec{U}_b(t,\vec{x})$:
  1. $\vec{U}(t,\vec{x})=\vec{U}_b(t,\vec{x})$;
  2. $\frac{\partial\tilde{p}}{\partial\vec{n}}=\rho_0\vec{n}\cdot\left(-\vec{U}_b\cdot\nabla\vec{U}+\vec{F}\right)$;
  3. $\frac{\partial s}{\partial\vec{n}}=0$;
- Initially the medium in the domain is at rest: the velocity is zero, the salinity $s(0,\vec{x})$ varies linearly from the bottom border to the top border according to a given constant gradient $\frac{\partial s}{\partial z}$, the pressure field is aligned to compensate the gravity force (mechanical equilibrium).

The numerical scheme settings for each case are taken from the verification test considered in [56] and is set as follows:

- the Adams-Bashforth numerical scheme for time derivatives;
- the Gauss scheme with volume-to point interpolation for τ-terms;
- the Gauss scheme with linear interpolation for convective and diffusive fluxes.

The vertical red line "AB" in Fig. 20 indicates the segment where the solution is sampled and compared with reference data.

Monochromatic regime of internal gravity wave attractor. Parameters of the regime are set according to Table 6.

The horizontal velocity function is prescribed on the left vertical boundary:

$$\vec{U}_b(t,x,y,z)=\vec{i}A\cdot\cos\left(\frac{\pi\cdot z}{H}\right)\cdot\omega_0\cdot\sin(\omega_0 t),\qquad(21)$$

where $\vec{i}=(1,0,0)$.

The fluid flow in this problem is the result of a nonlinear interaction between buoyancy force, inertial force and the pulsation of vertical wall in the horizontal direction. Therefore, simple assessments of τ used in previous examples cannot be applied here. Instead, τ is related to buoyancy period T_b and to the Reynolds number Re_H estimated for the maximum velocity of the vertical wall.

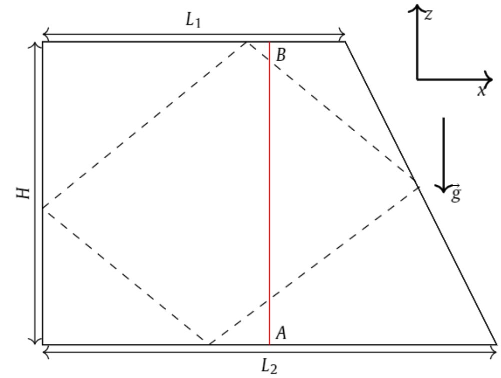


Fig. 20. Sketch of problem setup for the internal gravity waves attractor problem.

The buoyancy period T_b, corresponding to given stratification parameters is evaluated as:

$$T_b=\sqrt{-g_z\beta\frac{\partial s}{\partial z}}=1\text{ s}.\qquad(22)$$

The Reynolds number Re_H is estimated using perturbation velocity as:

$$Re_H=\frac{HA\omega_0}{\nu}=62.3.\qquad(23)$$

The maximum value of τ is estimated using T_b and Re_H as:

$$\tau=\frac{T_b}{Re_H}\approx 0.02s.\qquad(24)$$

The upper limit of time step Δt^{max} can be estimated from the criterion of stability (Fourier criterion), imposed by an explicit approximation of diffusive τ-terms:

$$\tau|\vec{U}_{max}|^2\frac{\Delta t^{max}}{\Delta x^2}=0.02(10A\omega_0)^2\frac{\Delta t^{max}}{H^2/300^2}\approx 0.027\Delta t^{max}\le\frac{1}{2}.\qquad(25)$$

Thus even for the very accurate mesh with 300 points per height of computational domain and in the worst case when the maximum velocity is equal to 10 perturbation velocities, the upper limit of time step is much larger than the maximum value of τ.

According to observations that were made for the skewed cavity case, a reduction of the maximum of τ by 4-10 times leads to errors of about 5%-10%. Thus, for the simulation we used the following values of $\tau=0.02$ s, 0.01 s, 0.005 s.

The non-orthogonality of the geometry is between 90° and 45°. Therefore C_τ must be between 2 and 4 with corresponding to a maximum time step Δt between 0.00125 s and 0.0025 s.

The objective of this investigation is the comparison of numerical simulations results obtained with different approaches. The comparative study allows scrutinizing the qualitative features and convergence properties of the QHD algorithm as well as demonstrating how numerical scheme settings (spatial resolution and value of τ) affect the solution. The approaches under consideration are:

- QHD algorithm implementation as *QHDFoam* OpenFOAM solver;
- SEM implementation as Nek5000 program;
- PISO algorithm implementation as *salinityBoussinesqPimpleFoam* OpenFOAM solver [57].

The custom implementation [57] of PISO for Navier-Stokes equations with buoyancy is used because the standard OpenFOAM

Table 6
Dimensional parameters of the monochromatic internal wave attractor problem.

ρ_0 kg/m^3	β –	g_z m/s^2	$\frac{\partial s}{\partial z}$ m^{-1}	ν m^2/s	Sc –	A m	ω rad/s	H m	L m
1000	0.04	-10	-2.5	10^{-6}	1000	0.025×10^{-2}	0.623	0.4	0.6

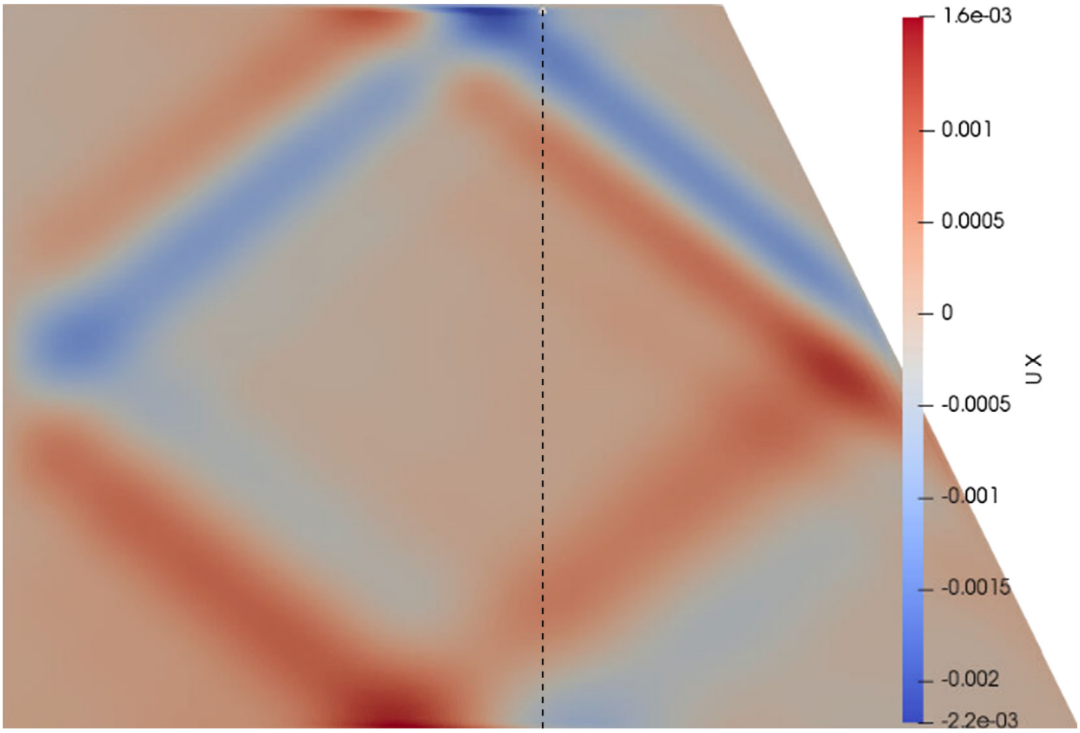


Fig. 21. Visualization of the horizontal component of velocity for internal gravity waves attractor.

solvers are too diffusive and are unable to reproduce the generation of internal gravity wave attractors.

The SEM [34] solution is selected as the reference, since it is thoroughly compared with experiments and studied in several works: in 2D [58] and later in 3D [59]. Furthermore, the high order spatial approximation employed in Nek5000 suggests that numerical results obtained in similar conditions are very close to the actual solution.

Similar comparative studies, but without usage of PISO algorithm, were conducted previously in the work [56]. The difference between SEM and FVM with QHD algorithm, measured by comparing the most energetic POD modes [60] was less than 5%.

After 70-100 periods of wave maker oscillation at the tank, the phenomenon of internal waves focusing occurs, see Fig. 21.

Results of QHD and PISO simulations along the line “AB” (Fig. 20) are compared with those of a Nek5000 simulation after 70-100 periods of external boundary perturbations (corresponding to $\approx 700 - 1000$s). The horizontal component of instant velocity is presented in Fig. 21. A comparison of Nek5000 and *QHDFoam* results is presented in Figs. 22, 23, 25, 27, while results for the PISO algorithm are presented in Figs. 24, 26, 28.

It can be clearly seen that the PISO algorithm qualitatively evidences the effect of wave focusing, but quantitatively it does not converge to the SEM solution. In contrast, the QHD solution is characterized by τ – and mesh convergence.

Biharmonic regime of internal gravity wave attractor. The theory and simulation of an internal gravity wave attractor have a particular interest when the perturbation occurs in a range of frequencies, as it is the case for undulation of ocean surface. As a reduced variant of a perturbation with a continuous spectrum, the regime with 2 harmonics is considered here.

The parameters of the regime are set according to Table 7, similar to [24]. The buoyancy period T_b is estimated as 1s, and $Re_H \approx 133$. The horizontal velocity is prescribed on the left vertical boundary as

$$\vec{U}_b(t,x,y,z) = \vec{i} A \cdot \cos\left(\frac{\pi \cdot z}{H}\right) \cdot (\omega_1 \cdot \sin(\omega_1 t) + \omega_2 \cdot \sin(\omega_2 t)), \tag{26}$$

where $\vec{i} = (1,0,0)$.

In a previous work [56] the value of τ for the same conditions, but for a monochromatic perturbation is estimated as $\approx 0.0015 - 0.003$ s. If we use the procedure from the previous example, we arrive at $\tau = 0.0025$ s, which is very close to the first estimate. We set $\Delta t = \frac{1}{2}\tau$.

Boundary and initial conditions as well as numerical scheme settings are identical to those in the monochromatic case.

The results of calculations with different values of τ are shown in (Fig. 30). The comparison of computed velocity fields with a ray tracing technique (Fig. 29) developed to predict internal waves paths also shows a nice coincidence.

The general computational cost of the method based on QHD is not so large – see Table 8. It exceeds SEM in a few times and is less than in PISO/SIMPLE by 30%.

The implemented QHD algorithm employs no any corrector loops, neither for pressure-velocity coupling, nor for skewness or non-orthogonality of the mesh. The *QHDFoam* solver is proposed as a first estimate tool for problems with complex geometry and physics, when using of high-fidelity algorithms and their imple-

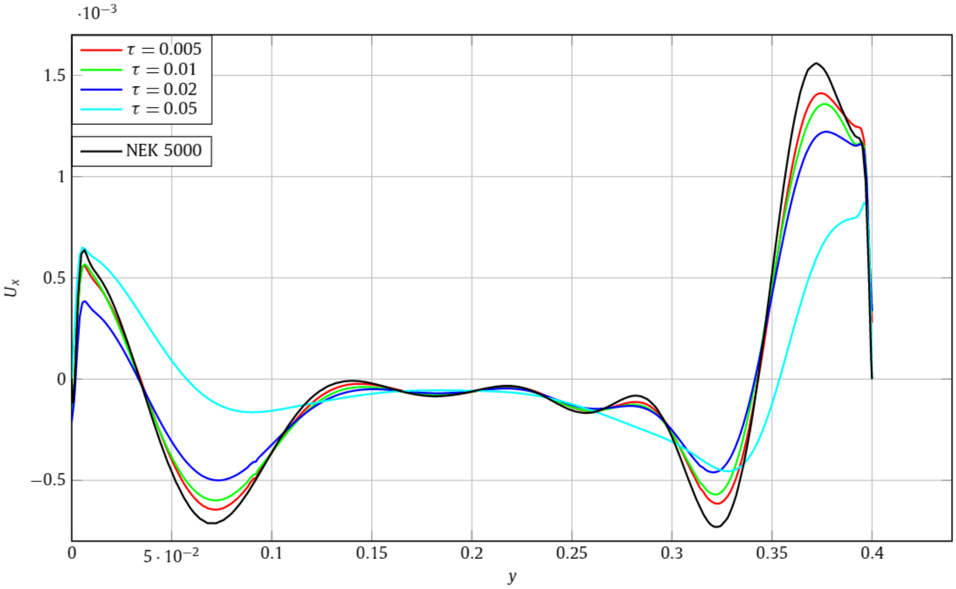


Fig. 22. Influence of regularization parameter τ on the numerical solution (horizontal component of velocity on line A-B) for the monochromatic internal gravity waves attractor, mesh 225×300, time t=200 s.

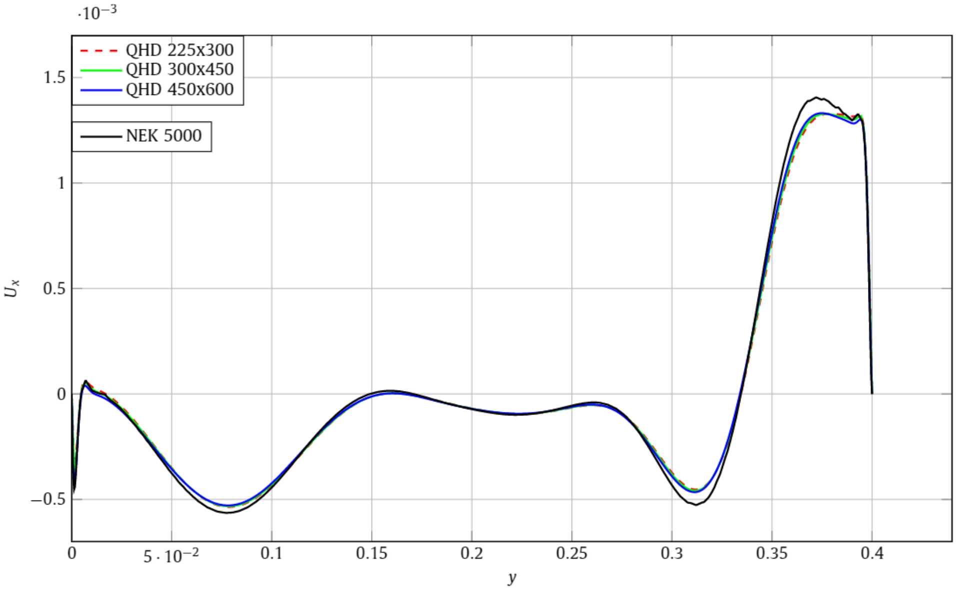


Fig. 23. Mesh convergence of the QHD numerical ($\tau = 0.005$ s) solution (horizontal component of velocity on line A-B) for the monochromatic internal gravity waves attractor, time t=160 s.

Table 7
Dimensional parameters of biharmonic internal wave attractor problem.

ρ_0 kg/m^3	β –	g_z m/s^2	$\frac{\partial s}{\partial z}$ m^{-1}	ν m^2/s	Sc –	A m	ω_1 rad/s	ω_2 rad/s	H m	L m
1000	1	−10	−0.1	10^{-6}	1000	0.02×10^{-2}	0.58	0.665	0.2	0.3

Table 8
Performance comparison of PIMPLE, SEM and QHD approaches implementations on Intel(R) Xeon(R) CPU X5670 2.93 GHz, 100 s of model time, with $5 \cdot 10^{-3}$ time step, gcc -O3.

Approach	Execution time (s)	Number of elements	PIMPLE Corrections	Non-orthogonal corrections
Nek5000	1037	1296 spectral elements	0	0
PISO	7630	67 500 finite volumes	3	1
QHD	4479	67 500 finite volumes	0	0

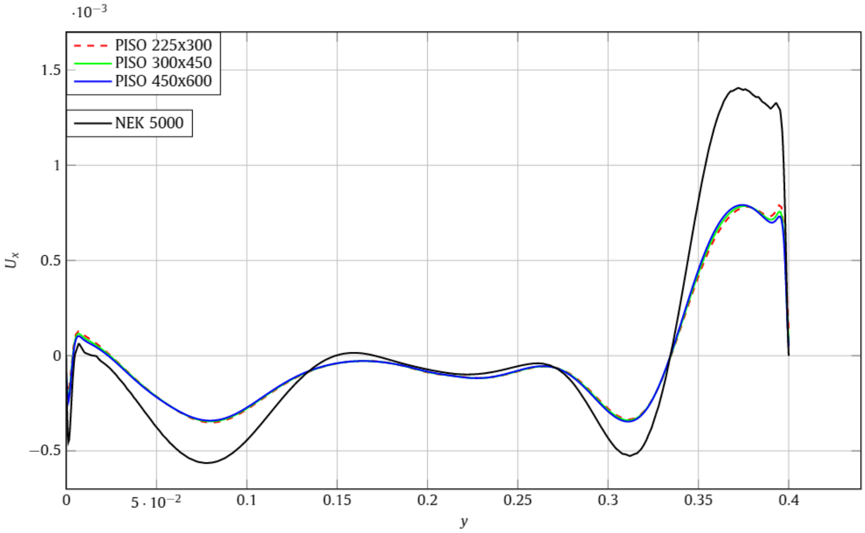


Fig. 24. Mesh convergence of the PISO numerical solution (horizontal component of velocity on line A-B) for the monochromatic internal gravity waves attractor, time t=160 s.

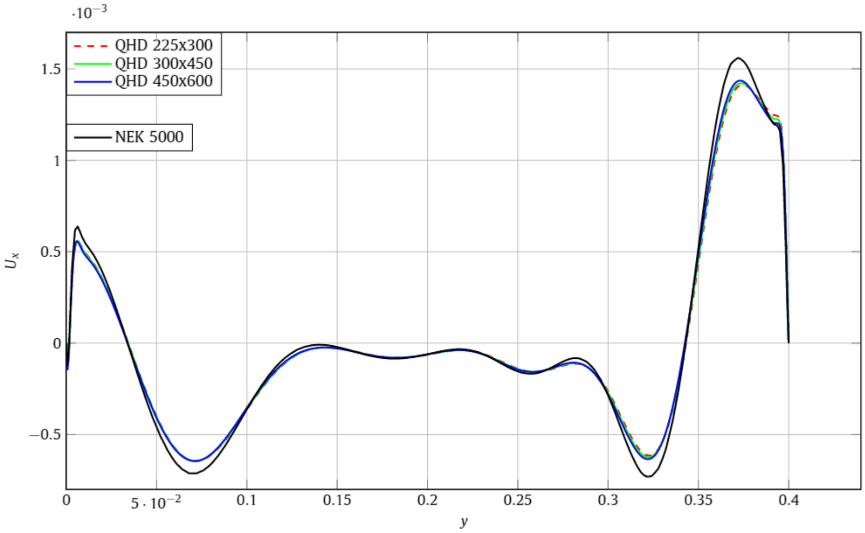


Fig. 25. Comparison of the numerical solution (horizontal component of velocity on line A-B) obtained with QHD algorithm ($\tau = 0.005$) on different meshes for the monochromatic internal gravity waves attractor, time t=200 s.

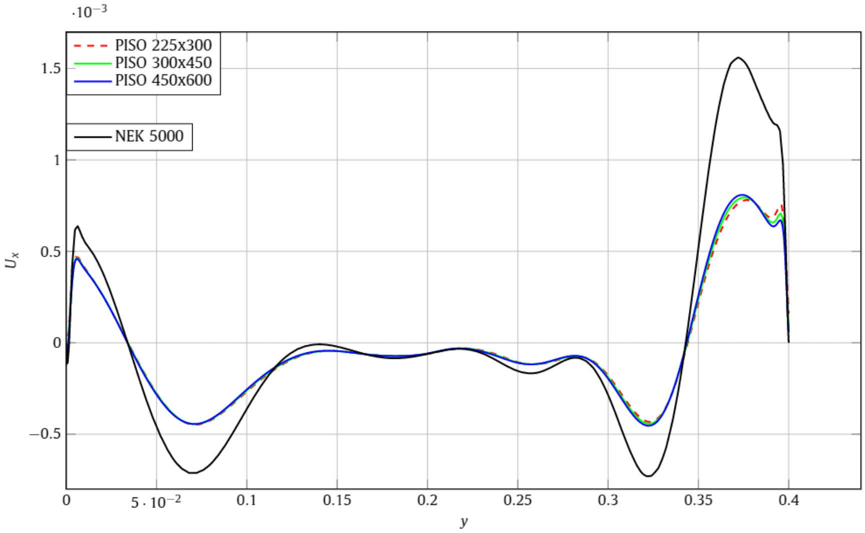


Fig. 26. Comparison of the numerical solution (horizontal component of velocity on line A-B) obtained with PISO algorithm on different meshes for the monochromatic internal gravity waves attractor, t=200 s.

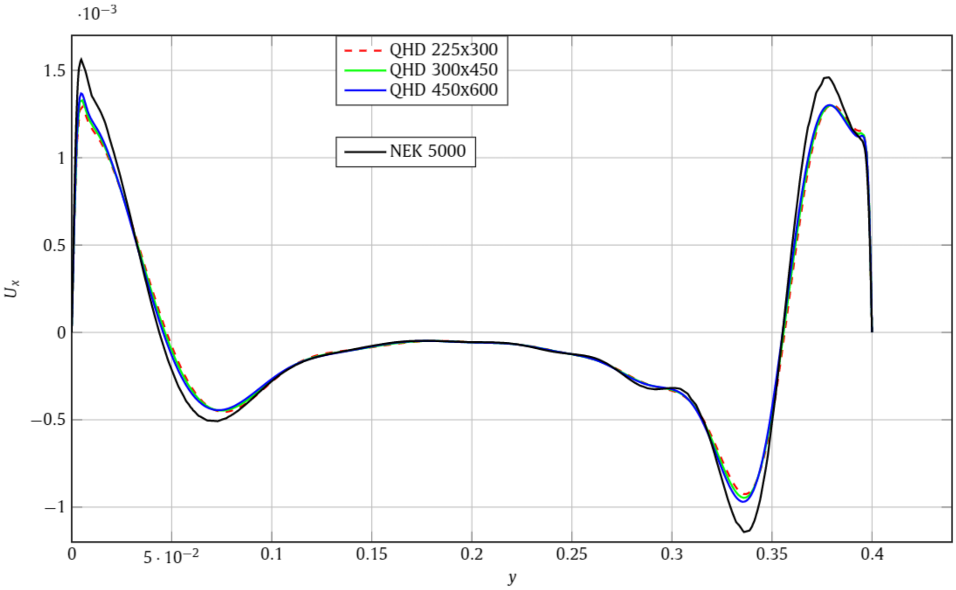


Fig. 27. Comparison of the numerical solutions (horizontal component of velocity on line A-B) obtained with QHD algorithm ($\tau = 0.005$) on different meshes for the monochromatic internal gravity waves attractor, t=300 s.

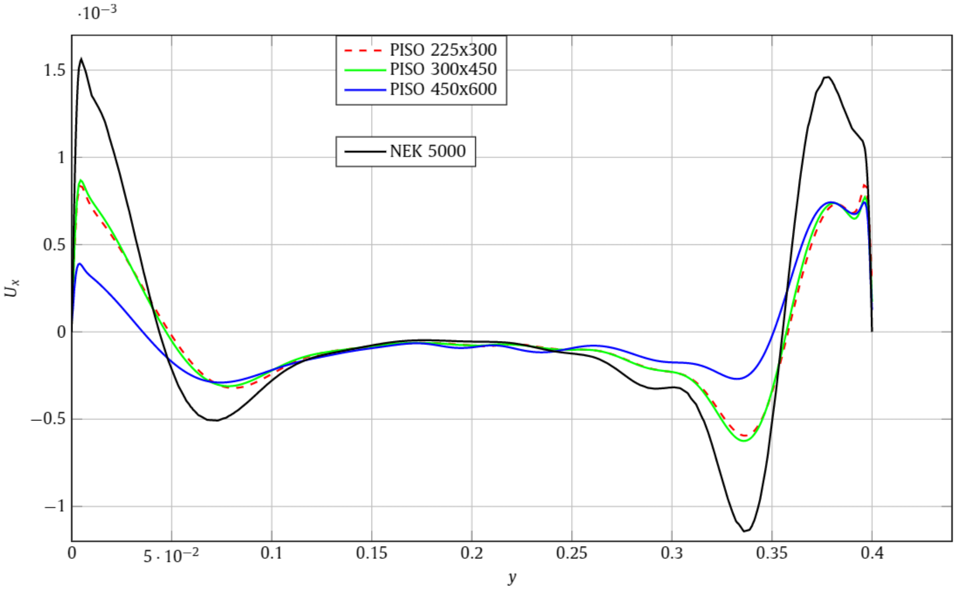


Fig. 28. Comparison of the numerical solutions (horizontal component of velocity on line A-B) obtained with PISO algorithm on different meshes for the monochromatic internal gravity waves attractor, time=300 s.

mentations (such as Nek5000) demands formidable efforts or is even impossible due to limitations imposed by structured grids or numerical methods.

Further applications of the QHD algorithm, implemented as *QHDFoam* solver, involve, in particular, simulations of 3D vessels with curvilinear boundaries and complex perturbation laws for medical applications, see [61].

6. Conclusions

A numerical algorithm based on the QHD approach and the Finite Volume Method for unstructured polyhedral mesh is proposed for the simulation of incompressible viscous flows with buoyancy under the Boussinesq assumption. The approach involves an approximation of quasi- (or regularized) hydrodynamic equations using a compact stencil, explicit spatial schemes and an implicit correction for diffusion terms. The discretized equations are solved sequentially: the pressure (continuity) equation, the momentum equation and the transport equation(s) for passive scalar(s).

Features and merits of the QHD algorithm have been analyzed by comparisons with several widespread approaches for pressure-velocity coupling in numerical models of incompressible fluid flows:

- unlike projection methods (such as Chorin-Uzawa), a single velocity field obeying the momentum equation is used;
- in contrast to the MAC approach, unstructured polyhedral meshes are used;
- there is no necessity for several embedded correction cycles, similar to employed in PISO/SIMPLE approaches.

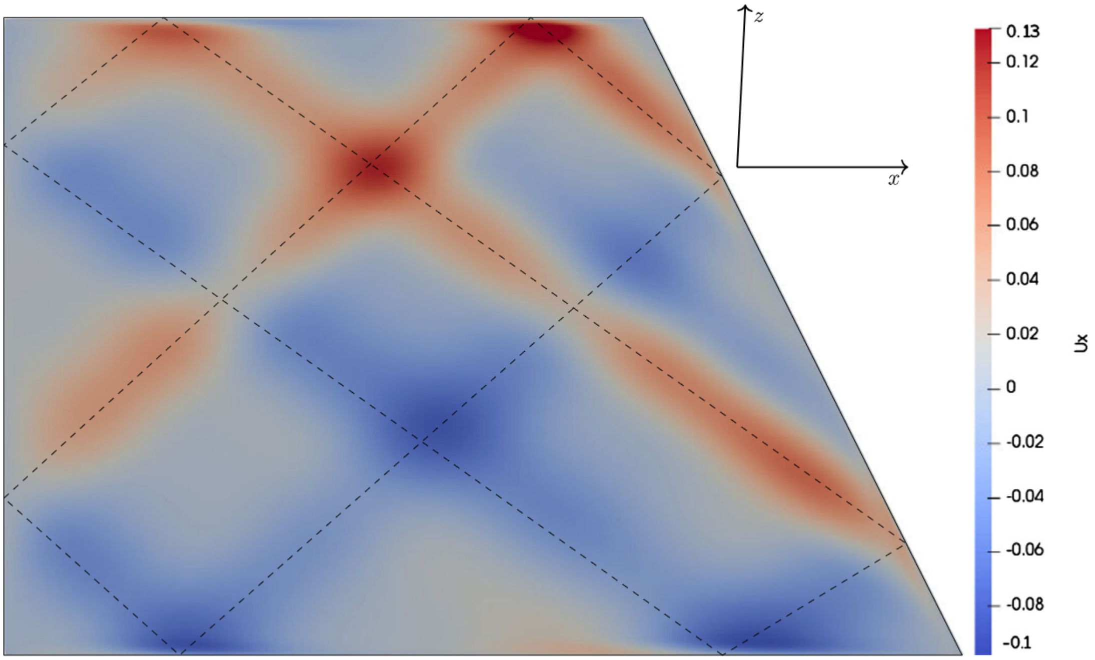


Fig. 29. Visualization of biharmonic internal gravity waves attractor velocity field and its comparison with ray tracing prediction.

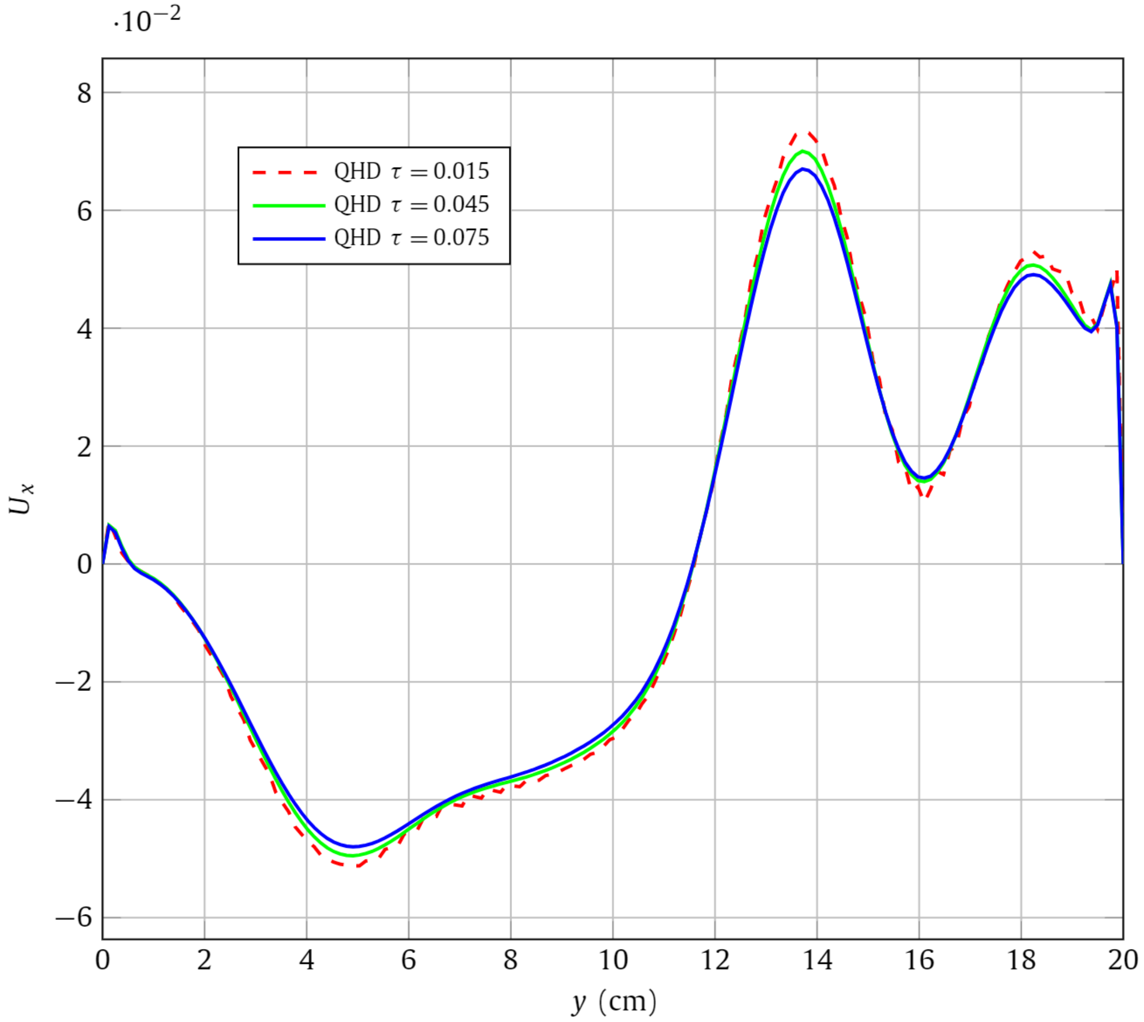


Fig. 30. Influence of regularization parameter τ on the numerical solution (horizontal component of velocity on line A-B) for the biharmonic internal gravity waves attractor, mesh 240×160, time t=260 s.

The described approach has been implemented as an OpenFOAM application – solver *QHDFoam*. The source code is published on GitHub platform: https://github.com/unicfdlab/QGDsolver.

The implementation of the QHD algorithm has been verified on several flow problems:

- laminar flow over a backward-facing step;
- natural convection induced laminar flow;
- laminar flow in the skewed cavity;
- monochromatic and biharmonic internal gravity wave attractor.

All simulations have shown a monotonic convergence of the QHD algorithm to reference solution. A comparison of the QHD algorithm implementation with the PISO implementation on the problem of monochromatic internal gravity wave attractor has showed that both algorithms reproduce the flow qualitatively properly. However, the PISO approach does not converge to the reference data obtained with the SEM program Nek5000 while the QHD algorithm does. The convergence process of the QHD algorithm is controlled by tuning a single parameter τ. The study of natural and forced convection flows in different conditions allows to propose a general procedure for choosing a τ value.

The developed solver (as well as the QHD approach) can be recommended as the tool for first estimates of complex transient incompressible flows in domains with curvilinear geometry, where high-order methods (3^{rd} or more) require too much efforts, whereas FVM with PISO/SIMPLE lacks accuracy.

In the present work, a basic case is described: the flow of a viscose incompressible fluid with buoyancy forces. Other problems were regarded (including within OpenFOAM) such as multi-phase flow with surface tension, multi-component flows (accounting for reactive mixtures, dispersed aerosols, etc.), moving boundaries, compressible flows (subsonic, transonic, supersonic, hypersonic), etc. [42].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The study was supported by the Russian Science Foundation, Project No. 19-11-00169.

The results of the work were obtained using computing resources of the federal collective usage center “Complex for simulation and data processing for mega-science facilities at NRC Kurchatov Institute”, http://ckp.nrcki.ru.

The authors are grateful to J.-C. Lengrand for fruitful discussions and remarks on the present report.

References

[1] M. Peric, Numerical methods for computing turbulent flows, Technical Report VKI LS 2004-06, 2003, http://www.staff.city.ac.uk/~ra600/Modelling/Peric-CFDlecture.pdf.

[2] G.Y. Dynnikova, Fluid Dyn. 35 (1) (2000) 24–32, https://doi.org/10.1007/bf02698782.

[3] I.K. Marchevsky, https://github.com/vortexmethods/vm2d, 2019, GitHub.

[4] K.S. Kuzmina, I.K. Marchevskii, Fluid Dyn. 54 (7) (2019) 991–1001, https://doi.org/10.1134/s0015462819070103.

[5] S. Dergachev, I. Marchevsky, G. Shcheglov, Aerosp. Sci. Technol. 94 (2019) 105374, https://doi.org/10.1016/j.ast.2019.105374.

[6] E. Napoli, M.D. Marchis, E. Vitanza, Comput. Fluids 106 (2015) 185–195, https://doi.org/10.1016/j.compfluid.2014.09.045.

[7] F.D. Pin, S. Idelsohn, E. Oñate, R. Aubry, Comput. Fluids 36 (1) (2007) 27–38, https://doi.org/10.1016/j.compfluid.2005.06.008.

[8] F.R. Saucedo-Zendejo, E.O. Reséndiz-Flores, Comput. Methods Appl. Mech. Eng. 324 (2017) 619–639, https://doi.org/10.1016/j.cma.2017.06.027.

[9] J. Kuhnert, General smoothed particle hydrodynamics, Ph.D. thesis, Technische Universität Kaiserslautern, 1999.

[10] A.J. Chorin, Bull. Am. Math. Soc. 73 (6) (1967) 928–931, https://projecteuclid.org:443/euclid.bams/1183529112.

[11] A. Prohl, Projection and Quasi-Compressibility Methods for Solving the Incompressible Navier-Stokes Equations, Vieweg+Teubner Verlag, 1997.

[12] J. van Kan, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 870–891, https://doi.org/10.1137/0907059.

[13] J. Guermond, P. Minev, J. Shen, Comput. Methods Appl. Mech. Eng. 195 (44–47) (2006) 6011–6045, https://doi.org/10.1016/j.cma.2005.10.010.

[14] L.J.P. Timmermans, P.D. Minev, F.N.V.D. Vosse, Int. J. Numer. Methods Fluids 22 (7) (1996) 673–688, https://doi.org/10.1002/(sici)1097-0363(19960415)22:7<673::aid-fld373>3.0.co;2-o.

[15] E. Komen, E. Frederix, T. Coppen, V. D'Alessandro, J. Kuerten, Comput. Phys. Commun. 253 (2020) 107145, https://doi.org/10.1016/j.cpc.2020.107145.

[16] V. D'Alessandro, S. Montelpare, R. Ricci, in: CFD Techniques and Thermo-Mechanics Applications, Springer International Publishing, 2018, pp. 65–81.

[17] F.H. Harlow, J.E. Welch, Phys. Fluids 8 (12) (1965) 2182, https://doi.org/10.1063/1.1761178.

[18] D. Kwak, C.C. Kiris, Computation of Viscous Incompressible Flows, Springer, Netherlands, 2011.

[19] C. Févrière, J. Laminie, P. Poullet, P. Angot, J. Comput. Appl. Math. 226 (2) (2009) 228–245, https://doi.org/10.1016/j.cam.2008.08.014.

[20] R. Issa, J. Comput. Phys. 62 (1) (1986) 40–65, https://doi.org/10.1016/0021-9991(86)90099-9.

[21] R. Issa, A. Gosman, A. Watkins, J. Comput. Phys. 62 (1) (1986) 66–82, https://doi.org/10.1016/0021-9991(86)90100-2.

[22] S. Patankar, D. Spalding, Int. J. Heat Mass Transf. 15 (10) (1972) 1787–1806, https://doi.org/10.1016/0017-9310(72)90054-3.

[23] M. Riella, R. Kahraman, G. Tabor, Comput. Fluids 192 (2019) 104275, https://doi.org/10.1016/j.compfluid.2019.104275.

[24] M.V. Kraposhin, E.V. Smirnova, T.G. Elizarova, M.A. Istomina, Comput. Fluids 166 (2018) 163–175, https://doi.org/10.1016/j.compfluid.2018.02.010.

[25] Y. Sheretov, Mat. Model. 6 (10) (1994) 35–45.

[26] Y.V. Sheretov, Continuum Dynamics under Spatiotemporal Averaging, SPC Regular and Chaotic Dynamics, Moscow-Izhevsk, 2009 (in Russian).

[27] T.G. Elizarova, Y.V. Sheretov, Comput. Math. Math. Phys. 41 (2) (2001) 219–234.

[28] T.G. Elizarova, Quasi-Gas Dynamic Equations, Springer, 2009.

[29] A.A. Zlotnik, Math. Notes 83 (5–6) (2008) 610–623, https://doi.org/10.1134/s0001434608050040.

[30] A.A. Zlotnik, Comput. Math. Math. Phys. 50 (2) (2010) 310–321, https://doi.org/10.1134/s0965542510020120.

[31] V. Balashov, A. Zlotnik, J. Comput. Dyn. 7 (2) (2020) 291–312, https://doi.org/10.3934/jcd.2020012.

[32] V. Balashov, A. Zlotnik, Math. Model. Anal. 25 (1) (2020) 110–129, https://doi.org/10.3846/mma.2020.10577.

[33] N. Offermans, A. Peplinski, O. Marin, P. Schlatter, Fluids 197 (2020) 104352, https://doi.org/10.1016/j.compfluid.2019.104352.

[34] P.F. Fischer, J.W. Lottes, S.G. Kerkemeier, Nek5000 Web page, https://nek5000.mcs.anl.gov/, 2020.

[35] T.G. Elizarova, B.N. Chetverushkin, Dokl. Akad. Nauk SSSR 279 (1) (1984) 80–83.

[36] T.G. Elizarova, B.N. Chetverushkin, Dokl. Akad. Nauk SSSR 25 (5) (1985) 164–169.

[37] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamics, Springer, 1996.

[38] M.V. Kraposhin, D.A. Ryazanov, E.V. Smirnova, T.G. Elizarova, M.A. Istomina, in: 2017 Ivannikov ISPRAS Open Conference (ISPRAS), IEEE Xplore, 2017.

[39] M.A. Istomina, E.V. Shilnikov, Keldysh Inst. Prepr. 86 (2019) 1–22, https://doi.org/10.20948/prepr-2019-86 (in Russian).

[40] E.V. Shilnikov, T.G. Elizarova, High Temp. Mat. Proc. Int. Q. High-Technol. Plasma Proc. 22 (2–3) (2018) 99–113, https://doi.org/10.1615/hightempmatproc.2018024713.

[41] ESI Group, OpenFOAM+, https://openfoam.com/, 2020. (Accessed 18 August 2020).

[42] UniCFD Lab, QGDsolver – OpenFOAM framework for simulation of fluid flows using regularized equations approach, https://github.com/unicfdlab/QGDsolver, 2020. (Accessed 18 August 2020).

[43] D. Gurov, T. Elizarova, Y. Sheretov, Math. Model. 8 (7) (1996) 33–44.

[44] T. Elizarova, I. Kalachinskaya, A. Kluchnikova, Y. Sheretov, Comput. Math. Math. Phys. 38 (10) (1998) 1662–1671.

[45] I.N. Sibgatullin, E.V. Ermanyuk, J. Appl. Mech. Tech. Phys. 60 (2) (2019) 284–302, https://doi.org/10.1134/s002189441902010x.

[46] I. Sibgatullin, X. Xu, A. Tretyakov, E. Ermanyuk, in: Central European Symposium on Thermophysics 2019 (CEST), AIP Publishing, 2019.

[47] B.F. Armaly, F. Durst, J.C.F. Pereira, B. Schönung, J. Fluid Mech. 127 (1983) 473, https://doi.org/10.1017/s0022112083002839.

[48] J. Yin, M.V. Fischels, R.G. Rajagopalan, in: AIAA Aviation 2019 Forum, American Institute of Aeronautics and Astronautics, 2019.

[49] A. Lestari, Development of unsteady algorithms for pressure-based unstructured solver for two-dimensional incompressible flows, https://lib.dr.iastate.edu/etd/10611/, 2009.

[50] W. Sutikno, Flux Corrected Method: An Accurate Approach to Fluid Flow Modeling, Ph.D. thesis, Iowa State University, 1997.

[51] D. De Vahl, I. Jones, Int. J. Numer. Methods Fluids 3 (3) (1983) 227–248, https://doi.org/10.1002/fld.1650030304.

[52] P. Vabishevich, M.M. Makarov, A.G. Chudanov, V.V. Churbanov, Numerical modelling of convective flows in Function of stream, velocity vorticity, temperature, http://library.keldish.ru/preprint.asp?id=1993-28, 1993.

[53] J. Hines, A comparative study of the simple and fractional step time integration methods for transient incompressible flows, Ph.D. thesis, University of Waterloo, 2008.

[54] E. Erturk, B. Dursun, Z. Angew. Math. Mech. 87 (5) (2007) 377–392, https://doi.org/10.1002/zamm.200610322.

[55] C. Brouzet, T. Dauxois, E. Ermanyuk, S. Joubaud, I. Sibgatullin, in: EGU General Assembly Conference Abstracts, 2015, p. 14401.

[56] M. Kraposhin, D. Ryazanov, T. Elizarova, I. Sibgatullin, M. Kalugin, V. Velikhov, E. Ryabinkin, in: 2018 Ivannikov Ispras Open Conference (ISPRAS), IEEE, 2018.

[57] UniCFD Lab, salinityBoussinesqPimpleFoam – transient solver for buoyant, turbulent flow of incompressible fluids with salinity transport and support for static and moving meshes, https://github.com/unicfdlab/salinityBoussinesqPimpleFoam, 2020. (Accessed 18 August 2020).

[58] C. Brouzet, E.V. Ermanyuk, S. Joubaud, I. Sibgatullin, T. Dauxois, Europhys. Lett. 113 (4) (2016) 44001, https://doi.org/10.1209/0295-5075/113/44001.

[59] I. Sibgatullin, E. Ermanyuk, K. Vatutin, D. Ryazanov, X. Xu, in: The XXII Workshop of the Council of Nonlinear Dynamics of the Russian Academy of Sciences, vol. 47(1), 2019, pp. 112–115.

[60] M. Kalugin, V. Korchagova, M. Kraposhin, I. Marchevsky, V. Moreva, in: Herald of the Bauman Moscow State Technical University, in: Series Natural Sciences, vol. 78, Jun. 2018.

[61] T.V. Stenina, T.G. Elizarova, M.V. Kraposhin, Keldysh Inst. Prepr. 66 (2020) 1–30, https://doi.org/10.20948/prepr-2020-66.

[62] E.M. Sparrow, W. Chuck, Numer. Heat Transf. 12 (1) (1987) 19–40, https://doi.org/10.1080/10407788708913572.

[63] J. Kim, P. Moin, J. Comput. Phys. 59 (2) (1985) 308–323, https://doi.org/10.1016/0021-9991(85)90148-2.

[64] L.P. Hackman, G.D. Raithby, A.B. Strong, Int. J. Numer. Methods Fluids 4 (8) (1984) 711–724, https://doi.org/10.1002/fld.1650040802.