ISSN 2070-0482, Mathematical Models and Computer Simulations, 2023, Vol. 15, Suppl. 1, pp. S131–S142. © Pleiades Publishing, Ltd., 2023.
Russian Text © The Author(s), 2023, published in Matematicheskoe Modelirovanie, 2023, Vol. 35, No. 8, pp. 79–96.

Numerical Modeling of a Melt Flow by the Czochralski Method in the OpenFOAM Package Using a Quasi-Hydrodynamic Algorithm

M. A. Kirushina[a],*, T. G. Elizarova[a], and A. S. Epikhin[b]

[a] Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Moscow, Russia

[b] Ivannikov Institute for System Programming, Russian Academy of Sciences, Moscow, Russia

* e-mail: m_ist@mail.ru

Received May 15, 2023; revised June 19, 2023; accepted July 3, 2023

Abstract—The formulation of the problem, the numerical solution method, and description of the test calculation case for modeling a melt flow in the problem of crystal growth by the Czochralski (Chochralsky) method are presented. The mathematical model is based on regularized or quasi-hydrodynamic (QHD) equations to describe a viscous, incompressible heat-conducting fluid. The numerical algorithm is implemented on the OpenFOAM package within the mulesQHDFoam solver. The proposed approach allows calculations of an unsteady three-dimensional flow in a crucible with the given geometry for high Reynolds and Grashof numbers. As an example, the results for the flow in a simplified configuration are given. An asymmetric transient flow is obtained.

Keywords: OpenFOAM package, quasi-hydrodynamic algorithm, viscous incompressible fluid, Czochralski method

DOI: 10.1134/S2070048223070074

INTRODUCTION

The Czochralski (Chochralsky) method is one of the most common methods for growing crystals from a melt: a crystal seed is placed on the free surface of the melt, which is pulled upward as the crystal grows. In this case, the grown ingot obtains the same crystal structure as the original seed crystal. To grow crystals by the Czochralski method, crucibles—containers that are resistant to the melt and the parameters of the installation—are used. To ensure uniform distribution of temperature and impurities throughout the volume of the melt, the seed crystal with the single crystal growing on it and the crucible with the melt are rotated in opposite directions. To increase the yield of the crystalline substance, the melt is continuously heated. The classic crucible has a rounded shape, in which a heat-insulating layer is placed on top of the melt.

The shape of the resulting crystal is close to cylindrical with the distortions determined by the thermal conditions of growth, the pulling rate, the crystal structure, and the crystallographic orientation of the grown sample. For the subsequent use of crystals in the production of microelectronic devices, the grown crystal must be as homogeneous as possible. Considering the high cost of the starting materials and the installation itself, the high temperatures at which the crystallization process occurs, the impossibility of visualizing the flow or measuring the parameters of the melt, as well as the toxicity of the entire process as a whole, there is obvious interest in the mathematical modeling of the melt flow using the Czochralski method.

In the real industrial installations of the Czochralski method, the melt flow is a three-dimensional unsteady motion that can be described in terms of the equations of a viscous incompressible fluid in a gravity field in the Boussinesq approximation [1, 2]. The numerical description of this process is widely covered in the literature, a detailed analysis of which is not possible in this study. Due to the complexity of the process and the insufficient power of computational tools, the authors of the early works used fairly stringent simplifying assumptions: the flow was assumed to be axisymmetric and steady; see, for example, [3, 4], the bibliography for them, and the review [5]. Subsequently, gravitational convection was modeled for parameters of installations close to reality, including for nonaxisymmetric unsteady regimes close to turbulent regimes, for example, [6–9]. However, due to the complexity of the analysis of the results depending on the Grashof and Reynolds numbers, studies were also carried out in simplified formulations, for example, modeling three-dimensional stationary and unsteady flows in the gravity field without taking into account the rotation [10]. Basically, problems were solved within individual research codes.

In general, to describe the melt flow in the Czochralski method, a numerical algorithm is required that allows calculations at large Reynolds and Grashof numbers, is not focused on the symmetry or stationarity of the flow, and describes crucible shape used in the installations. Due to the complexity of the calculation, to increase the efficiency of the algorithm, it is required to implement it on a multiprocessor computer system.

This paper provides a concise description of a method that allows the numerical modeling of an unsteady melt flow by the Czochralski method in a full three-dimensional setting, taking into account the rotation and real parameters of the melt. The algorithm is based on regularized or quasi-hydrodynamic (QHD) equations to describe the flow of a viscous incompressible fluid. The numerical implementation was performed using the mulesQHDFoam solver included in the OpenFOAM open-source software package. The latter allows the use of arbitrary unstructured spatial grids to describe the complex configuration of the crucible and multiprocessor computing systems to speed up calculations.

This article has the following structure: the first section provides a brief description of the QHD equations for problems of gravitational convection in the Boussinesq approximation. The second section presents the formulation of a typical problem of the Czochralski method with the setting of the corresponding parameters in the OpenFOAM open-source complex. The third section describes the implementation of the calculation in the OpenFOAM package, for which the corresponding case (the computational template) is freely available. The results of the test calculation, confirming the unsteady nature of the flow, are given in the fourth section. In conclusion, the calculation results are analyzed and some conclusions are drawn.

1. QHD SYSTEM OF EQUATIONS FOR DESCRIBING THE FLOW OF A VISCOUS INCOMPRESSIBLE HEAT-CONDUCTING FLUID AND A NUMERICAL ALGORITHM

The Navier–Stokes equations for describing the flow of a viscous incompressible heat-conducting fluid include the incompressibility, momentum, and temperature equations. When modeling convection flows in a gravity field, a simplified version of the system proposed in the works of A. Oberbeck (1879) and J. Boussinesq (1903) for the Navier–Stokes equations is widely used. This approximation assumes that the density depends only on temperature in the form $\rho = \rho_0(1 - \beta(T - T_0))$, where $(T - T_0)$ is the temperature deviation from its value T_0, $\rho_0 = \mathrm{const} > 0$ is the constant density value, and $\beta > 0$ is the thermal expansion coefficient. In this case, the change in density with temperature is taken into account only in the term with the external force, which determines the thermal convection in the liquid. In addition to temperature, the model can include the dependence of density on the impurity concentration.

The system of Navier–Stokes equations for thermal convection in the Boussinesq approximation has the following form (see, for example, [1, 2]):

$$\operatorname{div}\mathbf{u} = 0, \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \operatorname{div}(\mathbf{u} \otimes \mathbf{u}) + \frac{1}{\rho_0}\nabla p = \frac{1}{\rho_0}\operatorname{div}\Pi_{NS} - \beta\mathbf{g}(T - T_0), \tag{2}$$

$$\frac{\partial T}{\partial t} + \operatorname{div}(\mathbf{u}T) = x\Delta T. \tag{3}$$

Here, $\mathbf{u} = \mathbf{u}(\mathbf{x},t)$ is the hydrodynamic velocity, $p = p(\mathbf{x},t)$ is the pressure measured from the hydrostatic pressure, $(\mathbf{u} \otimes \mathbf{u})$ is a tensor of the second rank obtained as a result of the direct product of two vectors, $\Pi_{NS} = \mu[(\nabla \otimes \mathbf{u}) + (\nabla \otimes \mathbf{u})^T]$ is the Navier–Stokes viscous stress tensor, μ is the dynamic viscosity coefficient, and x is the thermal diffusivity coefficient. In the system of equations, it is convenient to use ν, the kinematic viscosity coefficient, which is related to the dynamic viscosity coefficient μ as $\mu = \nu\rho_0$. Values ν and x are usually constant.

Calculating the divergence from Eq. (2) and taking into account the validity Eq. (1), we obtain the Poisson equation for pressure

$$\frac{1}{\rho_0}\Delta p = -\operatorname{div}\left[(\mathbf{u} \cdot \nabla)\mathbf{u}\right] - \operatorname{div}\left(\beta\mathbf{g}(T - T_0)\right). \tag{4}$$

The method of obtaining the Poisson equation as a consequence of system (1) and (2) leads to the need to include an additional stage in the numerical algorithm. The numerical solution is corrected so that for the calculated velocities, Eq. (1), which is an expression of the law of conservation of mass in the problem under consideration, must be satisfied with sufficient accuracy.

The boundary conditions for system (1)–(3) do not include the boundary condition for pressure, which is necessary to solve the Poisson equation. The pressure boundary conditions required in the numerical algorithm are written based on various assumptions. When writing difference equations on spaced spatial grids, it is possible to eliminate the use of boundary conditions for pressure. However, these mathematical problems complicate the procedure for the numerical solution of the original system of equations, making it multistage.

There are a number of ways to numerically solve the system of equations (1)–(3) (for example, [2]). In this study, for the numerical modeling of a hydrodynamic problem, the QHD method is used, based on the regularization of the system of Navier–Stokes equations (1)–(3). The QHD system, or regularized, system of equations was introduced by Yu.V. Sheretov in 1993. Let us present the form of the QHD system of equations in the Oberbeck–Boussinesq approximation according to, for example, [11–14]:

$$\operatorname{div}(\mathbf{u}-\mathbf{w}) = 0, \tag{5}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \operatorname{div}\big((\mathbf{u}-\mathbf{w})\otimes\mathbf{u}\big) + \frac{1}{\rho_0}\boldsymbol{\nabla} p = \frac{1}{\rho_0}\operatorname{div}\Pi - \beta\mathbf{g}(T-T_0), \tag{6}$$

$$\frac{\partial T}{\partial t} + \operatorname{div}\big((\mathbf{u}-\mathbf{w})T\big) = x\Delta T. \tag{7}$$

The velocity correction vector **w** and viscous stress tensor S are calculated as

$$\mathbf{w} = \tau\left[(\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u} + \frac{1}{\rho_0}\boldsymbol{\nabla} p + \beta\mathbf{g}(T-T_0)\right], \quad \Pi = \Pi_{NS} + \rho_0\mathbf{u}\otimes\mathbf{w}. \tag{8}$$

Here τ is the regularization parameter, which has the dimension of time. The QHD system is closely related to the original system of equations (1)–(3) and, in particular, has the following properties: the regularized system has a number of exact solutions that coincide with the exact solutions of the original system; for a stationary problem, any solution that is an exact solution of the original system also turns out to be an exact solution of the regularized equations. In particular, the exact solution of the stationary problem of the flow between rotating infinite cylinders within the framework of the Navier–Stokes equations is also the exact solution of this problem for the QHD equations [11, 12]. At $\tau = 0$ the regularized system transforms into the original system of equations.

The QHD system is more complex than the original equations. However, the new system of equations makes it possible to build simpler methods for the numerical solution of hydrodynamic problems, which differ from the methods that are traditionally used for the numerical solution of the Navier–Stokes system of equations written in velocity-pressure variables. In particular, this concerns the use of central differences to approximate spatial derivatives and calculate pressure, which is an essential part of the entire computational procedure as a whole.

For the QHD system, the Poisson equation for pressure follows directly from the continuity equation. Indeed, substituting expression (8) into the continuity equation (5) for **w**, we immediately obtain the equation for pressure at constant values ρ_0 and τ:

$$\frac{1}{\rho_0}\Delta p = -\operatorname{div}\big[(\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u}\big] + \frac{1}{\tau}\operatorname{div}\mathbf{u} - \operatorname{div}\big(\beta\mathbf{g}(T-T_0)\big). \tag{9}$$

We note that Eq. (9) differs from Eq. (4) by the presence of a term with the coefficient $1/\tau$ on the right side.

The boundary conditions for pressure in this case are a direct consequence of the boundary conditions for the velocity vector **u** and velocity correction vector **w**. For example, for a solid impenetrable wall, the boundary condition can be set to zero of the velocity component normal to the boundary $\mathbf{u}_n = 0$ and $\mathbf{w}_n = 0$. From these two conditions the boundary condition for pressure directly follows in the form

$$\boldsymbol{\nabla} p = -\rho_0\beta\mathbf{g}(T-T_0). \tag{10}$$

To carry out calculations using the QHD system, it is necessary to select the regularization parameter τ. It is convenient to relate its value with the characteristic hydrodynamic time of the problem, which we denote by τ_0. The characteristic time in problems of forced convection can be determined through the coefficient of kinematic viscosity ν and characteristic velocity u_0 in the dimensional form as

$$\tau_0 = \nu/u_0^2. \tag{11}$$

Parameter τ has the order of characteristic time τ_0. If the values of τ are too high, the terms proportional to this coefficient (regularizers) become dominant, which leads to the breakdown of the numerical solution. If τ is too small, then an unacceptably small time step is required for the stability of the numerical solution. As the calculation experience shows, starting from a certain value, the decrease in parameter τ ceases to affect the accuracy of the numerical solution. The choice of the regularization parameter determines the stability of the algorithm, as well as its accuracy and complexity; therefore, its optimal value should be selected in the calculation.

As shown by the computational practice, the time integration step, which ensures the conditional stability of the difference algorithm, should not exceed the value of τ and is often chosen in the form $\Delta t = \tau/2$.

To numerically solve system (1)–(3), its regularized analog in form (5)–(7) was used. The spatial approximation is constructed using the finite volume method with approximation for all spatial derivatives using central differences. The difference algorithm uses an explicit difference scheme for all terms except terms with molecular viscosity, for which the implicit iterative method is used ([15], p. 665). This method is included as one of the basic elements in the OpenFOAM complex. The QHD algorithm does not include limiters and other flow restrictors, and the stability of the algorithm is ensured by artificial τ-dissipation, whose nonlinear structure adjusts its value depending on the gradients of the numerical solution.

The experience of using QHD equations for problems of modeling unsteady flows of a viscous incompressible fluid is quite extensive; see, for example, [11, 13, 14], where the results of calculations within the framework of the research codes are presented. In [16–18] the results of the calculations obtained using the QHD algorithm within the framework of the OpenFOAM open package are presented. In [16, 17] the details of the implementation of the algorithm on spatial unstructured grids and its advantages compared to other computational kernels included in this package are described.

Note that the heat conduction equation in the QHD system, currently implemented as solver on the OpenFOAM open platform, includes an additional term on the right side compared to (7) and has the form

$$\partial T/\partial t + \operatorname{div}(\mathbf{u}T) = \operatorname{div}(\mathbf{w}T) + \chi\Delta T + \operatorname{div}(\tau\mathbf{u}(\mathbf{u}\cdot\nabla T)). \tag{12}$$

This kind of regularization for the transfer equation was first proposed in [19] in the shallow water approximation. To simulate the flow of a two-component gas mixture [20], the regularization type (12) was successfully used to calculate the transfer of densities of individual components of the mixture.

2. PROBLEM STATEMENT

To demonstrate the operation of the algorithm, a simplified version of the crucible's geometry is used, the diagram of which is shown in Fig. 1. For the calculation, the crucible's parameters were taken [8, 9] as follows: height of the outer cylinder $H = 0.1$ m and the diameter of the cylinder $D = 2R = 0.15$ m. The crucible rotates at constant angular velocity of $\omega_1 = 5$ rpm $= 0.52$ rad/s and the crystal rotates in the opposite direction with angular velocity $\omega_2 = -10$ rpm $= 1.04$ rad/s. The crystal's diameter is taken from real technological dimensions and is 0.0508 m [9]. Problems similar in their geometry are presented in [7]. Let us immediately note that when using the OpenFOAM package and a solver based on the QHD equations, constructing a grid and calculating a variant with the required geometry is not difficult.

The computational domain in a simplified formulation is a cylinder in which the side surface connected to the bottom rotates with angular velocity ω_1. A gallium arsenide GaAs melt is placed in the crucible. On the upper surface, the region of crystal growth is indicated by a dark color. The annular region between the crucible and the crystal rotating in different directions is a free surface. At the initial moment of time, the flow is stationary. The side and bottom surfaces of the crucible are impenetrable walls. The system is in the gravity field $g = 9.81$ m/s^2.

It is assumed that the GaAs melt in the crucible at the initial moment of time has a melting point temperature equal to 1511 K and the pressure is equal to atmospheric pressure. The initial conditions of the problem are presented in Table 1.

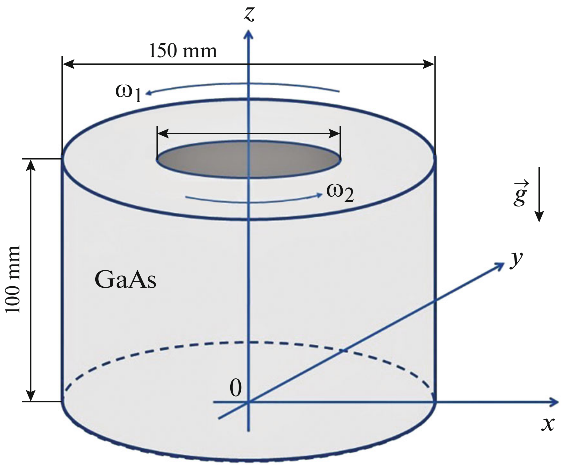

Fig. 1. Simplified diagram of the crucible used in the calculation.

At the initial moment, the melt is motionless and, following the crucible and the crystal, is drawn into a rotation due to viscosity.

The temperature of the crucible walls in the process unit is maintained in the mode required for operation through the influx of heat from outside; in a simplified setup, the walls and bottom are chosen to be adiabatic. The free surface of the melt is also assumed to be adiabatic, since under experimental conditions there is a layer of foam on the surface, which impedes the flow of heat into the atmosphere. The crystal surface is maintained cooler than the melt, at a temperature of 1500 K.

For the velocity on the free surface, the condition of nonflow with sliding is set; and on the solid walls of the crucible and crystal, the conditions of adhesion to the rotating surfaces of the crucible and crystal are set.

For pressure at the solid boundaries of the computational domain, additional boundary conditions (10) are set, which ensure the absence of the mass flow across these boundaries. On the lateral vertical solid walls, these conditions degenerate the condition for the derivative of pressure normal to the lateral wall. On the horizontal solid walls, the condition results in the form $\partial p/\partial z = -\beta\rho_0 g(T - T_0)$. Here $T_0 = 1511$ K. Atmospheric pressure is maintained on the free surface of the melt. A list of boundary conditions for pressure, velocity, and temperature is given in Table 2.

3. FEATURES OF THE CALCULATION IMPLEMENTATION IN THE OpenFOAM PACKAGE

A computational template with the problem under consideration and a description of its solution, calculated by the mulesQHDFoam solver, is freely available on https://github.com/m-ist/Chochralsky_mulesQHDFoam [21, 22].

When numerically modeling the flow of a viscous incompressible fluid with thermal convection using the mulesQHDFoam solver, implemented in the OpenFOAM open-source software package, values in dimensional form in the C system are used. The corresponding values for GaAs are given in Table 3. Note that the molar mass value is not used in QHDFoam-based solvers, although formally this value should be specified.

The boundary conditions for the hydrodynamic quantities are specified in the corresponding dictionaries of folder 0/. The boundary is divided into three regions: the crucible, the crystal, and the free surface (Table 4).

Table 1. Initial conditions of calculation

Pressure p	Velocity $\mathbf{U}$	Temperature T
10^6 Pa	(0, 0, 0)	1511 K

Table 2. Boundary conditions for the main hydrodynamic quantities

	Crucible	Crystal	Free surface
Pressure p	$\frac{\partial p}{\partial \mathbf{n}} = -\beta\rho_0\mathbf{g}(T - T_0)$	$\frac{\partial p}{\partial \mathbf{n}} = -\beta\rho_0\mathbf{g}(T - T_0)$	10^6 Pa
Velocity $\mathbf{U}$	$\omega_1 = 0.52$ rad/s	$\omega_2 = -1.04$ rad/s	$\mathbf{u}_z = 0$, $\partial\mathbf{u}_x/\partial z = \partial\mathbf{u}_y/\partial z = 0$
Temperature T	$\partial T/\partial \mathbf{n} = 0$	1500 K	$\partial T/\partial \mathbf{n} = 0$

Table 3. GaAs melt parameters in dimensional form

Molar mass M	144.6446 g/mol
Dynamic viscosity	0.00279 kg/(m s)
Kinematic viscosity	0.49×10^{-6} m^2/s
Thermal conductivity at melt temperature	0.178 W/(cm K)
Thermal expansion coefficient of melt β	1.87×10^{-4} K^{-1}
Prandtl number Pr	0.068
Density ρ_0	5720.0 kg/m^3
Heat capacity at constant pressure Cp	434.0 J/(kg K)

Table 4. Setting boundary conditions in the Chochralsky_mulesQHDFoam case

	Crucible (outerCylinder)	Crystal (disk)	Free surface (outlet)
p	qhdFlux	qhdFlux	10^6 Pa

$\mathbf{U}$	outerCylinder
	{type rotatingWallVelocity;
	origin (0 0 0);
	axis (0 0 1);
	omega 0.52;}
		Disk
		{type rotatingWallVelocity;
		origin (0 0 0);
		axis (0 0 1);
		omega –1.04;}
			Slip

T	$\partial T/\partial \mathbf{n} = 0$	1500 K	$\partial T/\partial \mathbf{n} = 0$

To carry out the test calculation, the simplest structured spatial grid was built and most of the grid cells had the shape of a rectangular parallelepiped with sides close to each other in length.

Figure 2 shows the computational grids on the surface of the crucible and crystal on the left; and the computational surface supplemented with a free surface on the right. The crystal in this implementation has a height of three grid cells.

The computational mesh was built using the blockMesh and snappyHexMesh utilities. The cylindrical crucible and crystal are specified in the geometry section with the corresponding radii. Using the topoSet utility, the corresponding faces of the free surface, crucible, and crystal are determined. The total number of computational grid cells is about 7000.

The calculation case with the problem under consideration, calculated by the mulesQHDFoam solver, can be downloaded [22] and the calculation can be performed with other parameters of the problem. In folder 0/, we can change the rotation velocity of the crucible and crystal, make the crucible isothermal, set a constant heat flow on the walls, and set the desired external relative pressure. In the constant/ thermophysicalProperties folder in the QHD section, we can vary the regularization parameter τ. In addition, in this folder we can set parameters for another melt in the crucible by setting ρ_0, Cp, μ, Pr, and β. In the system/snappyHexMesh folder, the parameters of the crucible and crystal cylinders are set, whose position and radius can be changed. The spatial grid parameters can be changed in the refinementSurfaces section. In the system/controlDict folder, a function is defined that records the velocity, pressure, and temperature fields at the control points.

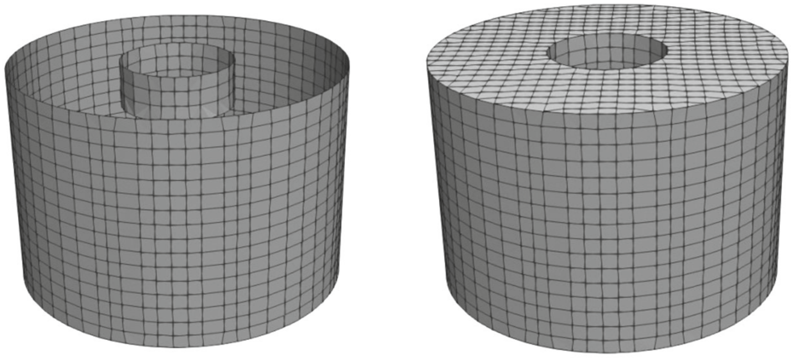

Fig. 2. Types of the computational grid on the boundaries of the computational domain.

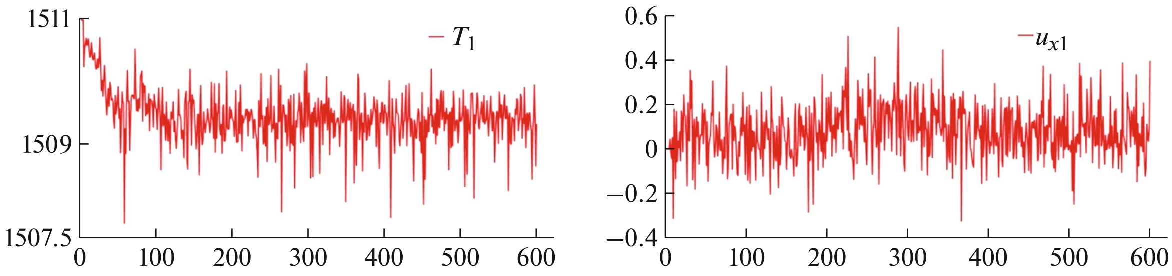

Fig. 3. Example of time dependence for temperature (left) and velocity components u_x (on right).

The pressure equation is solved using the standard conjugate gradient method with the PCG preconditioner.

We estimate the value of the regularization parameter in accordance with the data in Table 3. As the characteristic velocity, we choose the rotation velocity of the crucible to be $u_0 = \omega_1 R = 3.9 \times 10^{-2}$ m/s. Then the estimate for the regularization parameter τ takes the form $\tau_0 = \nu/u_0^2 = 3.3 \times 10^{-4}$ s.

4. CALCULATION RESULT

In the calculations below, parameter τ was chosen to be greater than its estimated value and equal to 0.001 s. The calculation was carried out with a step $\Delta t = 5 \times 10^{-4}$ s until $t = 1200$ s, which corresponds to 100 revolutions of the crucible.

An analysis of the obtained results shows that the temperature, velocity, and pressure in the melt demonstrate unsteady behavior. An example of the time dependence for the temperature and velocity components u_x at a point located near the lower surface of the growing crystal in the time interval from 0 to 600 s is shown in Fig. 3. It follows from the figure that starting from approximately 50–60 s, that corresponds to 5 revolutions of the crucible, the flow pattern becomes stably unsteady.

As an example of the flow structure, Fig. 4 shows the instantaneous trajectories of the particles in the crucible at successive moments of time with an interval of 1 s.

From the given figures, it is clear that at certain moments of time, a picture formed in the flow field resembles to toroidal vortex and demonstrates the downward descent of the cooled liquid under the crystal. It is this flow that is assumed as the stationary velocity distribution in the Czochralski problem in a number of works when modeling the problem in a stationary axisymmetric approximation; see, for example, [8, 9], as well as the estimates in [7] and bibliographies in these works. However, forced convection at high Reynolds numbers makes such a flow unstable and turns it into a chaotic one.

In order to estimate the flow field in the entire volume and the magnitude of the velocities, Fig. 5 shows the distributions of the velocity modulus for five sections along axis z for two points in time corresponding to 10 and 50 rotations of the crucible. The maximum velocity module does not exceed 0.44 m/s. From the

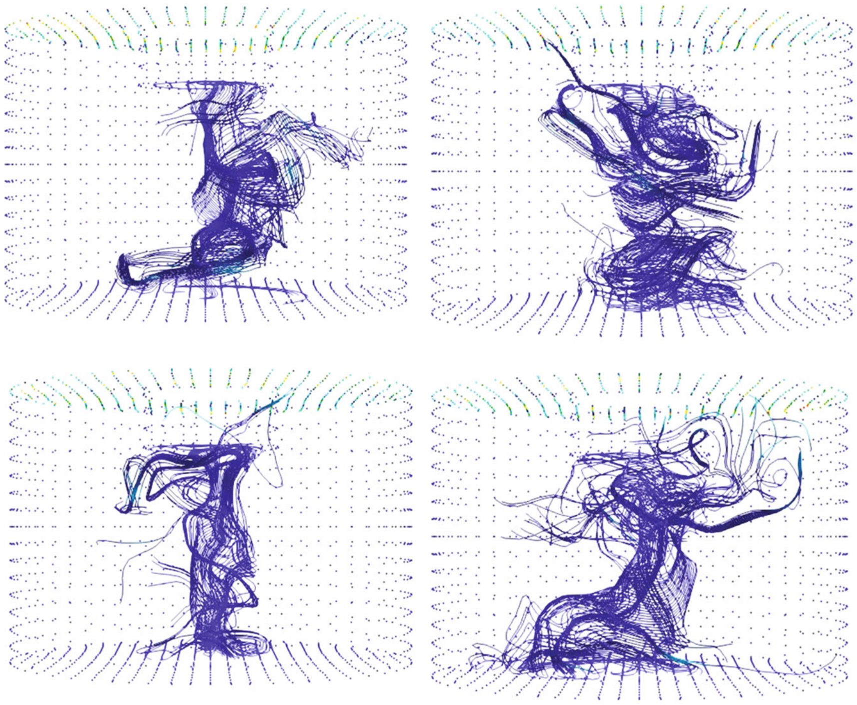

Fig. 4. Spatial picture of instantaneous particle trajectories at moments of time $t = 588$, 589, 590, and 591 s.

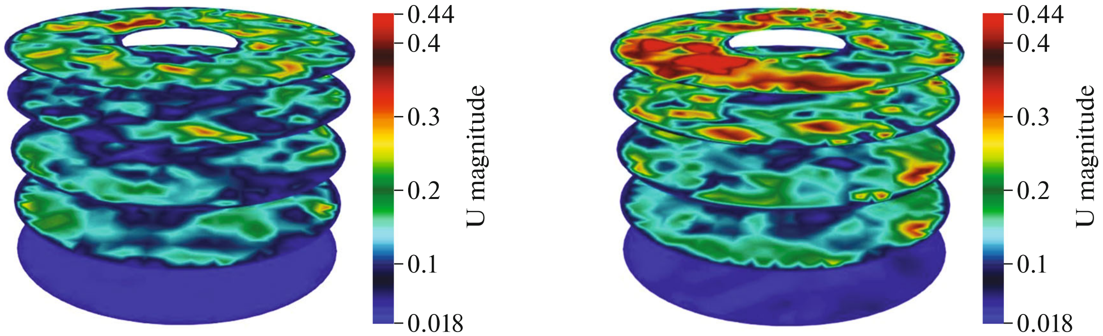

Fig. 5. Velocity modulus for 5 sections corresponding $z = 0.001$, 0.025, 0.05, 0.075, and 0.095 at times $t = 120$, 600 s, which is 10 and 50 turns of the crucible.

figures it follows that the flow is asymmetrical and unsteady both after 10 revolutions of the crucible and after 50 revolutions.

The uniformity of the grown crystal is largely determined by the distribution of velocities and temperatures in the immediate vicinity of its lower surface. To form a high-quality crystal, it is desirable to have as uniform a background of the temperature and velocities near its growing surface, which ensures the formation of a flat crystallization front.

Figure 6 shows the instantaneous particle trajectories and temperature distributions in the cross section $z = 0.076$, located in the azimuthal plane directly below the bottom surface of the crystal at the same times as in Fig. 5. A chaotic distribution of the spiral vortices with low flow velocities is observed.

The temperature distributions under the crystal at $z = 0.076$ at times of 120 and 600 s are shown in Fig. 7. The surface of the crystal is located in the plane $z = 0.08$. The figure shows that the temperature distribution is nonstationary and changes by about two degrees from the edge of the crystal to its center.

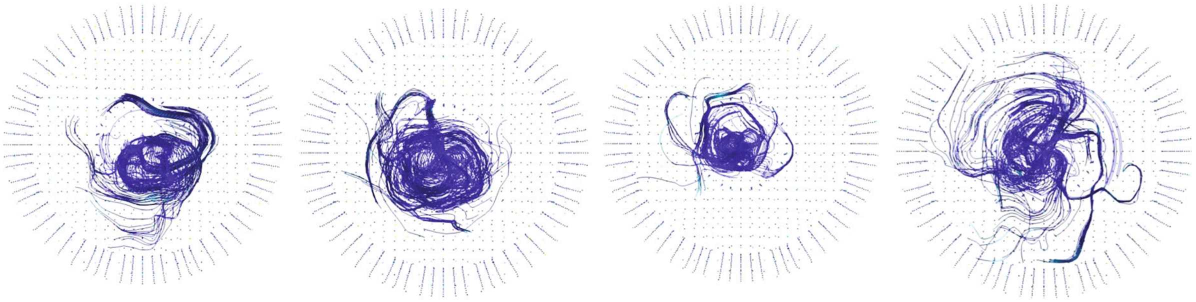

Fig. 6. Particle trajectories in the azimuthal plane at successive moments of time with an interval of 1 s.

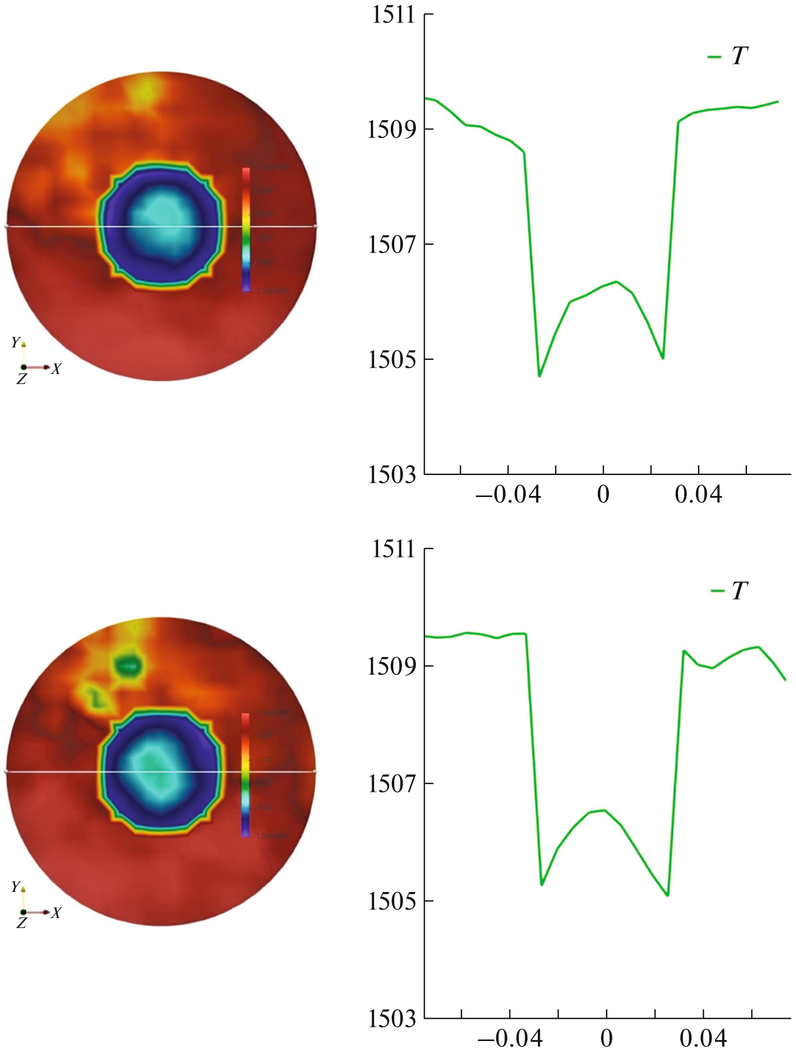

Fig. 7. Temperature distributions at moments of time $t = 120$ (top) and 600 s (bottom). On the left, in a section under the crystal ($z = 0.076$); on the right, along the x axis.

The field of the velocity modulus in the z, x plane for the instantaneous and time-averaged flow patterns is shown in Fig. 8. Averaging was performed from 600 to 1200 s in steps of $\Delta t = 1$ s. The resulting flow field shows that the instantaneous distribution is chaotic, and the averaged flow as a whole has a structured

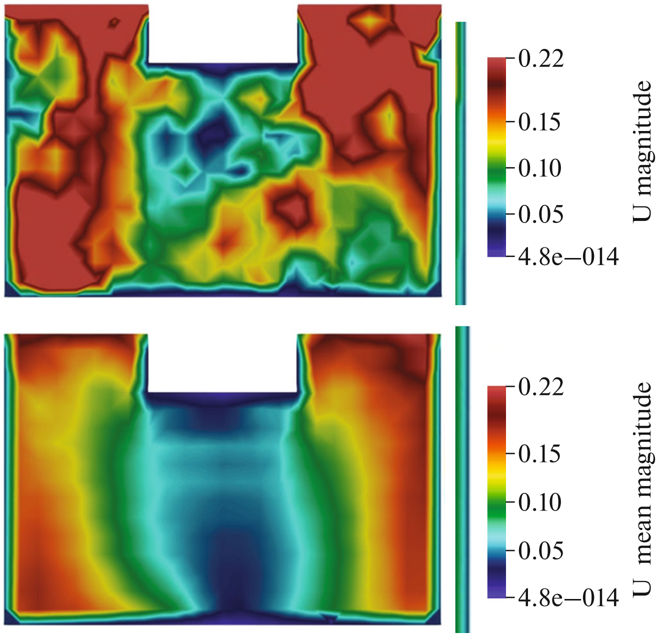

Fig. 8. Velocity module for instantaneous (top) and averaged (bottom) flow. Averaging time 600 s.

form, where some asymmetry can be related to the asymmetry of the spatial grid. In the structure of the averaged flow, a cylindrical vortex can be traced, whose axis coincides with the axis of rotation of the crucible and crystal, and its bases are located under the crystal and at the bottom of the crucible. The velocities in the vortex zone are lower than near the crucible walls.

5. CONCLUSIONS

This study provides for the first time a description of a numerical experiment on modeling a three-dimensional unsteady melt flow using the QHD algorithm. The calculation's parameters correspond to the settings actually used. In contrast to the assumptions used for this problem by many authors, it was found that the flow in the melt was asymmetrical and unsteady. The resulting type of flow is determined by the Reynolds and Grashof numbers inherent in the process. The Reynolds number, calculated from the rotation velocity of the crucible, corresponds to the developed turbulent flow regime $\mathrm{Re} = (u_0 R)/\nu \sim 6\times10^3$. The estimate for the Grashof number gives the value $\mathrm{Gr} = (\beta g \Delta T R^3)/\nu^2 \sim 3.4\times10^7$. This Grashof number corresponds to the regime of the developed thermal convection, i.e., the variant when the heat transfer due to gravitational convection prevails over the heat transfer due to thermal conductivity. The relative role of gravitational convection compared to forced convection in the heat transfer is determined by the parameter $\gamma = \mathrm{Gr}/\mathrm{Re}^2$. At $\gamma > 1$, thermal convection is predominant; in the opposite case, forced convection, caused by the rotation of the crucible and crystal, is predominant. In this example $\gamma \sim 1$. However, in the presented simulation example, all thermal processes are included. The Rayleigh number, which is often used in the analysis of flow properties, is $\mathrm{Ra} = \mathrm{Pr}\cdot\mathrm{Gr} \sim 2.3\times10^6$.

In the numerical experiments [7], a series of melt flows was simulated in formulations close to those described in this paper. A spatial grid in the (r, z, θ) geometry with the approximation of spatial derivatives not lower than the second approximation, was used. For the modes at $\mathrm{Gr} > 2\times10^6$ and $\mathrm{Re} > 2000$, unsteady periodic flows of various types with deviation from axial symmetry are observed. This fact is qualitatively consistent with the results presented above.

The experience of twenty years of using the QHD algorithm and its testing shows that the method monotonically converges when the spatial grid is refined, the method is conditionally stable with an adequate choice of the regularization parameter, and when the regularization parameter is decreased, its accuracy does not change but requires the time step to be reduced. These studies were previously carried out for test cases and are not presented here.

The detailed description of the calculation presented in this study in the OpenFOAM open software package will allow interested users to prepare a software version and carry out their own calculations of problems of this type, using the example described in this study as the basic example. Note that on the given grid, the calculation time for 100 rotations of the crucible is about 10 h on 12 cores of the K100 multiprocessor complex installed at the Keldysh Institute of Applied Mathematics, Russian Academy of Sciences.

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. L. Landau and E. Lifshitz, *Fluid Mechanics*, 2nd ed., Course of Theoretical Physics, Vol. 6 (Butterworth-Heinemann, 1987).
https://doi.org/10.1016/C2013-0-03799-1
2. V. M. Paskonov, V. I. Polezhaev, and L. A. Chudov, *Numerical Modeling of Heat and Mass Transfer Processes* (Nauka, Moscow, 1984).
3. V. S. Berdnikov, V. I. Polezhaev, and A. I. Prostomolotov, “Viscous flow in a cylindrical vessel in the presence of a rotating disk,” Fluid Dyn. **20**, 690–697 (1985).
https://doi.org/10.1007/bf01050080
4. M. P. Marchenko, A. S. Senchenkov, and I. V. Fryazinov, “Mathematical modelling of the crystal growth from solution-melt by travelling heater method,” Mat. Model. **4** (5), 67–79 (1992).
5. N. V. Nikitin, S. A. Nikitin, and V. I. Polezhaev, “Convective instabilities in hydrodynamic model of crystal growth by the Czochralsky method,” Usp. Mekh. **2** (4), 63–105 (2003).
6. O. A. Bessonov and V. I. Polezhaev, “Unsteady nonaxisymmetric flows in the hydrodynamic Czochralski model at high Prandtl numbers,” Fluid Dyn. **46**, 684–698 (2011).
https://doi.org/10.1134/s0015462811050024
7. O. A. Bessonov, “Effect of crystal and crubicle rotation on the flow stability in the Czochralski model at low Prandtl numbers,” Fluid Dyn. **51**, 469–477 (2016).
https://doi.org/10.1134/s0015462816040050
8. V. Haslavsky, E. Miroshnichenko, E. Kit, and A. Gelfgat, “On experimental and numerical prediction of instabilities in Czochralski melt flow configuration,” J. Cryst. Growth **318**, 156–161 (2013).
https://doi.org/10.1016/j.jcrysgro.2010.10.013
9. R. Faiez and Ya. Rezaei, “Rotationally-driven axisymmetric oscillatory convection in a semitransparent Czochralski melt model,” J. Cryst. Growth **457**, 72–79 (2015).
https://doi.org/10.1016/j.jcrysgro.2016.04.045
10. O. A. Bessonov and V. I. Polezhaev, “Convective interactions and flow stability in hydrodynamic model of the Czochralski method,” in *Actual Problems of Mechanics: 50 Years of the Ishlinsky Institute for Problems in Mechanics, Russian Academy of Sciences* (Nauka, Moscow, 2015), pp. 177–197.
11. Yu. V. Sheretov, *Dynamics of Continua at Spatial-Time Averaging* (Regulyarnaya i Khaoticheskaya Dinamika, Moscow, 2009).
12. Yu. V. Sheretov, *Regularized Equations of Fluid Dynamics* (Tverskoi Gosudarstvennyi Universitet, Tver, 2016).
13. T. G. Elizarova, *Quasi-Gas Dynamic Equations*, Computational Fluid and Solid Mechanics (Springer, Berlin, 2009).
https://doi.org/10.1007/978-3-642-00292-2
14. T. G. Elizarova, I. S. Kalachinskaya, A. V. Klyuchnikova, and Yu. V. Sheretov, “Computation of convective flows using the quasihydrodynamic equations,” Comput. Math. Model. **10**, 160–171 (1998).
https://doi.org/10.1007/bf02359229
15. F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics*, Fluid Mechanics and Its Applications, Vol. 113 (Springer, Cham, 2015).
https://doi.org/10.1007/978-3-319-16874-6

16. M. V. Kraposhin, D. A. Ryazanov, and T. G. Elizarova, "Numerical algorithm based on regularized equations for incompressible flow modeling and its implementation in OpenFOAM," Comput. Phys. Commun. **271**, 108216 (2022).
https://doi.org/10.1016/j.cpc.2021.108216
17. D. A. Ryazanov, "Quasi-hydrodynamic approach for simulating internal wave attractors," Math. Models Comput. Simul. **14**, 547–558 (2022).
https://doi.org/10.1134/s2070048222040093
18. T. V. Stenina, T. G. Elizarova, and M. V. Kraposhin, "Regularized equations for disk pump simulation problems in OpenFOAM implementation," Preprint No. 66 (Keldysh Institute of Applied Mathematics, Moscow, 2020).
https://doi.org/10.20948/prepr-2020-66
19. T. G. Elizarova and A. V. Ivanov, "Numerical modeling of passive scalar transport in shallow water based on the quasi-gasdynamic approach," Comput. Math. Math. Phys. **60**, 1208–1227 (2020).
https://doi.org/10.1134/s0965542520070064
20. T. G. Elizarova and E. V. Shil'nikov, "Quasi-gasdynamic model and numerical algorithm for describing mixtures of different fluids," Comput. Math. Math. Phys. **60**, 1319–1331 (2023).
https://doi.org/10.1134/S0965542523070059
21. UniCFD Web-laboratory, QGDsolver: OpenFOAM framework for simulation of fluid flows using regularized (QGD/QHD) equations. https://github.com/unicfdlab/QGDsolver.
22. Chochralsky_mulesQHDFoam. https://github.com/m-ist/Chochralsky_mulesQHDFoam.

Publisher's Note. Pleiades Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.