

# Development of a new OpenFOAM solver using regularized gas dynamic equations

Matvey V. Kraposhin<sup>a,\*</sup>, Elena V. Smirnova<sup>a</sup>, Tatiana G. Elizarova<sup>b</sup>, Maria A. Istomina<sup>b</sup>

<sup>a</sup>Ivannikov Institute for System Programming of the RAS, Alexander Solzhenitsyn st., 25., Moscow, Russian Federation

<sup>b</sup>Keldysh Institute of Applied Mathematics RAS, Miusskaya sq., 4, Moscow, Russian Federation



## ARTICLE INFO

### Article history:

Received 29 September 2017

Revised 15 January 2018

Accepted 9 February 2018

Available online 13 February 2018

### Keywords:

Quasi-gas dynamic equations

Compressible flows

OpenFOAM

Unstructured grids

RhoCentralFoam

QGDfoam

Riemann problem

Finite volume method

Computational fluid dynamics

Supersonic jets

## ABSTRACT

The paper introduces the development of a new OpenFOAM solver *QGDfoam* for the numerical simulation of viscous compressible flows within a wide range of Mach numbers in the framework of the OpenFOAM formalism. The new solver is based on the implementation of regularized, or quasi-gas dynamic (QGD) equations. The mixed finite-volume and finite-difference approximation is constructed on unstructured space grids with co-located variables storage and explicit time scheme for convection approximation. The solver has been tested for a range of 1D Riemann problems and 2D cases, comparing results with analytic solutions and OpenFOAM's implementation of the Kurganov–Tadmor scheme known as *rhoCentralFoam*.

© 2018 Elsevier Ltd. All rights reserved.

## 1. Introduction

OpenFOAM program complex is a widely known open-source finite-volume toolkit for fluid flow simulations. Gas dynamic solvers included in OpenFOAM are based on a several robust numerical algorithms written in a flux form together with their finite-volume approximation. In this paper we present a new gas dynamic solver that is constructed in a flux form and based on Quasi-Gas Dynamic (QGD), or regularized Navier–Stokes equations.

The first steps in developing QGD equation system were made more than 30 years ago basing on kinetic approaches, e.g. [1,2]. Several alternative variants of creating QGD system were developed later and the system itself was investigated and used as the basis of a new family of finite-difference algorithms for gas dynamic computations, e.g. [3–6].

The QGD equation system has the form of equations for mass, momentum and energy conservation, but it differs from the Navier–Stokes system by additional strongly non-linear dissipative terms that have a form of second order space derivatives with a small coefficient as a factor. From the mathematical point of view

the QGD system looks more complicated than the Navier–Stokes one, but due to included additional dissipation, QGD system allows to use simpler algorithms for its numerical treatment. Below for simplicity, these algorithms are named as QGD algorithms.

At the first parallel systems on transputer elements, that appeared at that time, the efficiency of parallelization of future QGD algorithms was proved [7].

Speed-up scaling was investigated for a 3D implementation of QGD algorithm on rectangular space grids using MPI implementation of the C++ and CUDA on the K100 parallel computer of the Keldysh institute of Applied Mathematics RAS [8]. The calculation times were measured for a 1000 time steps on a grid consisting of 216 blocks with  $50 \times 50 \times 50$  cells in each block. The efficiencies were normalized to 4 GPUs. For 108 GPUs, tests showed a parallel efficiency of approximately 77.5%.

Scaling tests on up to 512 GPUs were performed on the supercomputer Lomonosov in Moscow State University [8]. The increase of QGD scaling efficiency using different kinds of computer devices Nvidia Tesla, Nvidia Kepler 20, Kepler 40 and Kepler80 has been shown in [9].

Already in [10] and [11] the prospects of the new model for calculation of pulsating flows were shown. Since the first variants of the QGD system were constructed on the basis of kinetic models,

\* Corresponding author.

E-mail addresses: [m.kraposhin@ispras.ru](mailto:m.kraposhin@ispras.ru), [os-cfd@yandex.ru](mailto:os-cfd@yandex.ru) (M.V. Kraposhin).

the new approach was firstly tested using examples of numerical modeling of rarefied flows, e.g., [12,13].

Numerous examples of successful applications of the QGD-based algorithms to a variety of steady and unsteady subsonic and supersonic gas flows can be found in recent publications, for example [14] subsonic flows [15] and astrophysical applications [16,17]. In [18] an example of laminar-turbulent transition in a Taylor–Green vortex decay flow was simulated by QGD algorithm with implicit turbulence modeling. QGD implementations for shallow water problems are presented in, e.g., [19,20].

QGD algorithms were implemented in Cartesian and cylindrical coordinates for regular and irregular space-grids and two and three-dimensional flows. A number of programs have been implemented on multiprocessor computational systems with MPI standard. Nevertheless, QGD implementation in the framework of OpenFOAM facilities would significantly expand the scope of QGD equations application and would give an opportunity to a wide range of users to try this approach as well as to enhance it.

Section 2 presents a short description of regularized gas dynamic equations compared with Navier–Stokes one. Section 3 contains OpenFOAM implementation of QGD equations approximation. A comparison of the developed solver *QGDfoam* with *rhoCentralFoam* [21] is presented in Sections 4 and 5. The comparison includes results of numerical simulation of a 1D Riemann problems and several 2D problems – high Mach number jet flows and planar flows at low Reynolds ( $Re$ ) and Mach ( $Ma$ ) numbers. Evaluation of the performance and parallel efficiency of the developed solver are presented in Section 6. Conclusions and perspectives are discussed in the conclusion section.

## 2. Regularized gas dynamic equations

### 2.1 Regularized or QGD system of equations

Like Navier–Stokes equations, the QGD system describes the evolution of gas density  $\rho$ , velocity  $\vec{U}$  and pressure  $p$ , as functions of space co-ordinates and time. But in contrast with the Navier–Stokes equations, the QGD parameters are regarded as averaged, or smoothed values over some small time interval. Such smoothing of the gas dynamic parameters  $\rho$ ,  $p$  and  $\vec{U}$  leads to the appearance of additional dissipative terms in the corresponding equations with an additional dissipative coefficient, which has the dimension of a time and is denoted as  $\tau$ , e.g. [4–6,22]. It is related to the averaging time. When  $\tau$  tends to zero, the system of QGD equations, which describes the evolution of smoothed gas-dynamic parameters, reduces to Navier–Stokes equations. The dissipative nature of  $\tau$ -terms is ensured by the existence of the non-negative dissipative function for QGD equations system.

The QGD system in Cartesian coordinates writes:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \vec{j}_m = 0, \quad (1)$$

$$\frac{\partial \rho \vec{U}}{\partial t} + \nabla \cdot (\vec{j}_m \otimes \vec{U}) + \nabla p = \nabla \cdot \hat{\Pi}, \quad (2)$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\vec{j}_m h^{tot}) + \nabla \cdot \vec{q} = \nabla \cdot (\hat{\Pi} \cdot \vec{U}). \quad (3)$$

Here, for simplicity reasons, external forces and heat sources are omitted, and the gas is supposed to be ideal. The full system can be found, for example, in [5]. The total energy per unit volume  $\rho e$  and the total specific enthalpy  $h^{tot}$  are defined as  $\rho e = \rho u + \rho \frac{1}{2}(\vec{U} \cdot \vec{U})$  and  $h^{tot} = e + p/\rho$ , where  $u$  – is specific internal energy. The mass flux density  $\vec{j}_m$  is given by:

$$\vec{j}_m = \rho(\vec{U} - \vec{w}), \quad \vec{w} = \frac{\tau}{\rho}(\nabla \cdot (\rho \vec{U} \otimes \vec{U}) + \nabla p). \quad (4)$$

The viscous stress tensor  $\hat{\Pi}$  and the heat flux  $\vec{q}$  writes

$$\hat{\Pi} = \hat{\Pi}_{NS} + \tau \vec{U} \otimes \rho \left( \vec{U} \cdot \nabla \vec{U} + \frac{1}{\rho} \nabla p \right) + \tau \hat{I}(\vec{U} \cdot \nabla p + \gamma p \nabla \cdot \vec{U}), \quad (5)$$

$$\hat{\Pi}_{NS} = \mu \left( \nabla \vec{U} + (\nabla \vec{U})^T - \hat{I} \frac{2}{3} \nabla \cdot \vec{U} \right), \quad (6)$$

$$\vec{q} = \vec{q}_{NS} - \tau \vec{U} \rho \left( \vec{U} \cdot \nabla u + p \vec{U} \cdot \nabla \left( \frac{1}{\rho} \right) \right), \quad \vec{q}_{NS} = -\kappa \nabla T. \quad (7)$$

Here,  $\gamma$  is the adiabatic exponent,  $\hat{I}$  the unit tensor. The internal energy per unit mass for ideal gas is  $u = p/(\rho(\gamma - 1))$ . Pressure, density and temperature  $T$  of are linked through the perfect-gas equation-of-state (EoS):

$$p = \rho RT, \quad (8)$$

where  $R$  is the unit-mass perfect gas constant. The thermal conductivity is given

$$\kappa = \frac{\mu C_p}{Pr}, \quad (9)$$

where  $Pr$  is the Prandtl number,  $C_p$  is specific heat capacity at constant pressure. The dynamic viscosity  $\mu$  contained in expressions (5)–(9) for  $\hat{\Pi}$  and  $\vec{q}$  is defined as a function of temperature:

$$\mu = \mu(T), \quad (10)$$

Simplified forms of QGD equations for the description of transonic and incompressible flows were proposed by Sheretov and can be found in, e.g., [4–6].

The expressions for the regularization, or dissipative coefficient  $\tau$  will be specified in the next section.

The continuity equation of the QGD system (1) includes additional second order space derivatives compared with the corresponding equation in Navier–Stokes system. Therefore the QGD system must have an additional boundary condition, that can be derived from the behavior of the mass flux  $\vec{j}_m$  on the boundary of the domain. This additional boundary condition is conveniently imposed for the boundary pressure gradient. For example, in an impermeable non-moving adiabatic wall a normal mass flux must be zero,  $\vec{j}_m = 0$ . This condition satisfies by traditional boundary conditions for velocity  $\vec{U} = 0$  and adiabatic conditions for normal temperature gradient on the wall  $\nabla T = 0$ , accomplished by the boundary condition for normal pressure gradient  $\nabla p = 0$ .

### 2.2 Dissipative coefficient $\tau$ and possibilities of numerical implementation

The QGD system, compared with the Navier–Stokes one includes an additional dissipative coefficient  $\tau$ . According to the ways of construction of the QGD system, this coefficient must be small, producing a small contribution of the additional terms compared with the other ones.

The value of  $\tau$  can be determined using the kinetic derivation of the QGD system or by comparing the additional  $\tau$  terms in the continuity Eq. (1) with the classical descriptions of self-diffusion, thermodiffusion or barodiffusion effects, see [4,5]. For a perfect gas, all these approaches lead to

$$\tau = \frac{\mu}{p Sc}, \quad (11)$$

where  $Sc$  is the Schmidt number, that is of the order of 1. So  $\tau$  is close to the so-called maxwellian relaxation time  $\tau_{Max} = \mu/p$ , that is close to a mean free time for the gas particles, and  $\tau$  can be estimated as

$$\tau \sim \tau_{Max} \sim \frac{\lambda}{C_s}. \quad (12)$$

Here  $\lambda$  is the mean free path of the gas particles, and  $C_s$  is the sound velocity. A more general formula for  $\tau$  was proposed by Sheretov [4] in the form

$$\tau = \gamma \frac{\mu}{Sc C_s^2 \rho}, \tag{13}$$

Using the Laplace formula for sound velocity  $C_s^2 = \gamma p/\rho$  the expression (13) transforms in (11).

In non-dimensional form  $\tau \sim Kn$ , where  $Kn$  is the Knudsen number. For rarefied gas flows  $\tau$  can be rather large. For the numerical simulation of moderately rarefied flows and flows in microchannels, QGD equations were used with  $\tau$  given by (11) and (13).

Nevertheless, for dense gases and liquids  $\tau$  value is negligibly small, and the role of additional terms in QGD equations becomes negligible compared with that of viscous Navier–Stokes terms.

However, for computational purpose,  $\tau$  can be increased to make it act as an efficient algorithm regularizer.

The more natural variant for the numerical implementation of  $\tau$ -terms as an artificial dissipation consists in replacing the mean free path  $\lambda$  in (12) by the computational space step  $\Delta_h$  in the form

$$\tau = \alpha \frac{\Delta_h}{C_s}, \tag{14}$$

where  $\alpha = const > 0$  is a small numerical factor for tuning the computational solution. It means, for example, that the spreading of the shock waves is proportional to a space step. Despite of its simplicity, the implementation of QGD equations with  $\tau$  terms in the form (14) allowed to use the central differences approximation for all spatial derivatives without stabilization of the algorithm by any kind of limiting procedures. The Courant stability of the explicit in time central-difference QGD schemes is ensured by the  $\tau$ -terms.

The  $\tau$  value may be chosen in a more sophisticated way, depending on the problem under consideration. Several examples used in previous computations are listed below.

In problems with non-negligible variations of Knudsen numbers, the combination of (11) and (14) was used in the form

$$\tau = \frac{\mu}{p Sc} + \alpha \frac{\Delta_h}{C_s}. \tag{15}$$

For flows with high Reynolds and Mach numbers the  $\tau$  dissipation included in the QGD system can be insufficient to stabilize the solution. In this case additional dissipation can be included in Navier–Stokes viscous stress tensor to increase viscosity coefficient as

$$\mu \rightarrow \mu + Sc^{QGD} p \tau, \tag{16}$$

where  $Sc^{QGD}$  is a positive tuning coefficient. Including a dependence of the tuning coefficient from Mach  $Ma$  number in (16) as  $Sc^{QGD} = Sc^{QGD}(Ma)$  allows to vary the level of artificial dissipation. For example, it can be increased in the vicinity of shock waves and decreased in the boundary layers.

For Euler flows, where  $\mu = \kappa = 0$ ,  $\tau$  is calculated using (14), a viscosity coefficient as  $\mu = p\tau$  and a heat conductivity according with (9). Thus all dissipative coefficients are artificial.

Basic values of tuning coefficients are  $\alpha = 0.5$ ,  $Sc^{QGD} = 1$ ,  $Pr = 1$ . Numerical dissipation and diffusion could be adjusted by decreasing  $\alpha$  and  $Sc^{QGD}$  coefficients down to values at which the solution becomes unstable or begins to oscillate. The value of  $Pr$  must be kept equal to 1 in most cases.

### 3. Approximation of QGD equations with OpenFOAM

The system of QGD Eqs. (1)–(3) has been approximated with the Finite Volume Method (FVM) implemented in open-source library OpenFOAM. The choice this of approach is justified by:

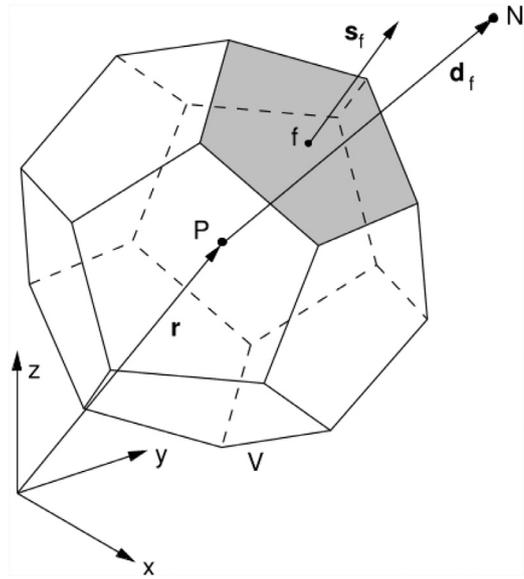


Fig. 1. Sketch of the OpenFOAM computational molecule – cell with arbitrary shape, its face and connection to neighboring cell.

- successful application of FVM on staggered irregular triangular meshes for QGD equations, see [5];
- conservativeness and boundedness properties of Finite Volume Method.

OpenFOAM's implementation of FVM uses co-located storage with compact stencil. According to this approach the computational domain is approximated as a set of non-overlapping volumes of arbitrary shape, connected to each other only through one common face – see Fig. 1, [23]. The unknown gas dynamics fields are averaged over cell volumes and the averaged values are stored at cell centers  $P$ . Balance equations describing the flow are approximated in the integral formulation using the Ostrogradsky-Gauss theorem to replace the volume integral with a surface integral and using mean value theorem to calculate the surface and volume integrals of approximated functions.

Traditionally, in the most of OpenFOAM applications the operator-splitting techniques is used (such as PISO, SIMPLE and their modifications) to solve pressure-velocity linked equations at low speeds, and Euler or multi-stage Runge–Kutta schemes with approximate Riemann solvers (Kurganov–Tadmire, HLLC, AUSM+) for high speed flows [21]. The matrix-coupled approach is used more rarely for pressure-velocity coupled simulations of incompressible fluids. The hybrid pressure-based and Kurganov–Tadmire approach was proposed for all Mach number perfect gas flows [24].

The semi-implicit approach, similar to Greenshields et al. [21] was used to approximate QGD equations. Within this approach the typical convection-diffusion Eq. (17) for some specific property  $\beta$

$$\frac{\partial \rho \beta}{\partial t} + \nabla \cdot (\vec{U} \rho \beta) - \nabla \cdot (D_\beta \nabla \beta) = S_\beta \tag{17}$$

is approximated in OpenFOAM as:

$$\frac{\rho^n \beta^n - \rho^o \beta^o}{\Delta t} = -\frac{1}{V} \sum_f \phi_f \beta_f + \frac{1}{V} \sum_f D_{\beta,f} \frac{\delta \beta}{\delta \vec{n}_f} |\vec{S}_f| + S_\beta, \tag{18}$$

where the convective flux  $\phi_f$  calculated at the face  $f$  is:

$$\phi_f = (\vec{U} \rho)_f \cdot \vec{S}_f, \tag{19}$$

and  $\delta \beta / \delta \vec{n}_f$  denotes an approximation of the surface normal derivative of  $\beta$  at the center of face  $f$ .

According to procedure [21] diffusive terms are approximated using an implicit scheme, while other terms are approximated using an explicit scheme. It has been shown that the explicit approximation of QGD equations is conditionally stable.

### 3.1. Approximation of QGD dissipative terms

In contrast to common transport equations which are similar to (17), QGD equations contain terms which needs a special approximation procedure in OpenFOAM. These terms are denoted as “QGD”-fluxes.

As an example, consider the dissipative term in the mass flux  $\vec{j}_m$  used in conservation equations:

$$\tau(\nabla \cdot (\rho \vec{U} \otimes \vec{U}) + \nabla p). \tag{20}$$

In order to apply a divergence theorem for this term, it should be calculated at the center of the face between two cells.

$$\tau_f([\nabla \cdot (\rho \vec{U} \otimes \vec{U})]_f + [\nabla p]_f), \tag{21}$$

where the subscript  $f$  denotes interpolation from a cell to a face.

The least square method is used for the evaluation of partial derivatives in the current implementation of QGD algorithm. In this approach the gradient (or other partial differential operators) at a face center is evaluated using field values from all cells connected to the given face through common vertices and edges. Values of an arbitrary property  $\beta$  in cells neighboring face  $f$  can be expressed with the gradient and value in the face center using Taylor’s expansion:

$$\beta_i = \beta_f + (\vec{x}_f - \vec{x}_i) \cdot (\nabla \beta)_f + e_i \tag{22}$$

By minimizing the weighted sum of squared errors from all cells connected to the considered face:

$$e_f^2 = \sum_i w_i^2 e_i^2, \tag{23}$$

we get the final gradient expression, evaluated with a least square method:

$$[\nabla \beta]_f \approx \sum_i w_i^2 \hat{G}^{-1} \vec{d}_i (\beta_i - \beta_f), \tag{24}$$

where  $\hat{G}$  is a  $3 \times 3$  tensor  $\hat{G} = \sum_i w_i^2 \vec{d}_i \otimes \vec{d}_i$  and weight  $w_i = 1/|\vec{d}_i|$ . The value at face  $\beta_f$  is evaluated using linear interpolation.

Final approximation expressions for the gradient are different from those derived in [5] for structured finite-difference grids, but both methods are second order in space, giving the same formal order of accuracy. The advantage of using a least square method compared with the original method is an ability to use unstructured grids with cells of arbitrary shape. However, this method has the following drawbacks: 1) it is more computationally expensive; 2) it demands an extending of a computational stencil in addition to those used in standard OpenFOAM’s FVM implementation, i.e. we need to store addressing information about the cells, that are connected to the given face by at least one vertex.

Extension of the computational stencil has led also to increase in overheads of the parallel algorithm. However, further measurements of the developed QGDfoam solver performance showed that MPI (Message Passing Interface) overheads are still reasonably small compared with other calculation costs.

### 3.2. General approximation procedure

The convective (inviscid) part of equations is approximated using an explicit scheme for mass  $\rho$ , momentum  $\rho \vec{U}$  and total energy  $\rho e$  balance equations. The viscous part of the equations is approximated as implicit corrections to the inviscid part according to the following procedure proposed in [21]

#### Mass conservation equation

$$\rho^n = \rho^o - \frac{\Delta t}{V} \sum_f \Phi_f(\rho^o)$$

$$\Phi_f(\rho^o) = \rho_f \vec{U}_f \cdot \vec{S}_f - \tau_f([\nabla \cdot (\rho \vec{U} \otimes \vec{U})]_f + [\nabla p]_f) \cdot \vec{S}_f$$

#### Momentum balance equation

$$(\rho \vec{U})^p = (\rho \vec{U})^o - \frac{\Delta t}{V} \sum_f \Phi_f((\rho \vec{U})^o)$$

$$\frac{\rho^n \vec{U}^n - \rho^o \vec{U}^o}{\Delta t} - \frac{\rho^n \vec{U}^p - \rho^o \vec{U}^o}{\Delta t} = \frac{1}{V} \sum_f \vec{S}_f \cdot \hat{\Pi}_f^{NS}$$

$$\Phi_f((\rho \vec{U})^o) = \Phi_f(\rho^o) \vec{U}_f + p_f \vec{S}_f - \vec{S}_f \cdot \hat{\Pi}_f^{QGD}$$

$$\hat{\Pi}_f^{QGD} = \tau_f \vec{U}_f \otimes (\rho_f \vec{U}_f \cdot [\nabla \vec{U}]_f + [\nabla p]_f) +$$

$$\tau_f \hat{I}(\vec{U}_f \cdot [\nabla p]_f + \gamma_f p_f [\nabla \cdot \vec{U}]_f)$$

$$\hat{\Pi}_f^{NS} = (\mu \nabla \vec{U}^n)_f + \mu_f ((\nabla \vec{U}^o)_f)^T - \frac{2}{3} \hat{I}(\nabla \cdot \vec{U}^o)_f$$

#### Energy balance equation

$$(\rho e)^p = (\rho e)^o - \frac{\Delta t}{V} \sum_f \Phi_f((\rho e)^o)$$

$$\frac{\rho^n u^n - \rho^o u^o}{\Delta t} - \frac{\rho^n u^p - \rho^o u^o}{\Delta t} = \frac{1}{V} \sum_f \left(\frac{\kappa}{C_p}\right)_f \frac{\delta u^n}{\delta \vec{n}_f} |\vec{S}_f| + \frac{1}{V} \sum_f \hat{\Pi}_f^{NS} \cdot \vec{U}_f^o$$

$$\Phi_f((\rho e)^o) = \Phi_f(\rho^o)_f h_f^{tot} - \hat{\Pi}_f^{QGD} \cdot \vec{U}_f \cdot \vec{S}_f -$$

$$\tau_f \vec{U}_f \rho_f \left( \vec{U}_f \cdot [\nabla u]_f + p_f \vec{U}_f \cdot \left[ \nabla \frac{1}{\rho} \right]_f \right) \cdot \vec{S}_f,$$

where superscript  $n$  refers to the new time layer,  $o$  – to the old time layer,  $p$  – to the predicted values,  $V$  – volume of computational cell,  $\Delta t$  – time step,  $\vec{S}_f$  – area of face between two adjacent cells, multiplied by normal vector value  $\vec{n}_f$ . If it is not mentioned explicitly, values in expressions are evaluated from previous time layer.

Within this approach, time step is limited only by CFL criterion:

$$CFL = (|\vec{U}| + C_s) \frac{\Delta t}{\Delta h} \leq CFL^{max} \tag{25}$$

### 3.3. Overall algorithm

The overall procedure of the explicit numerical solution of QGD equations involves the following steps

1. Start at initial time step  $t_0$  with initialized mesh (space discretization), all needed physical fields (velocity, density, pressure, temperature and others), parameters of numerical model (such as  $\alpha$ ,  $S_c^{QGD}$ ).
2. Check whether the current time point is less then final one. Go to the next step, if true. Otherwise – finish simulation.
3. Calculate volume and surface fields used in evaluation of fluxes:
  - compressibility field  $\psi = (\partial \rho / \partial p)_T$ ;
  - heat capacities ratio  $\gamma$ ;
  - speed of sound  $C_s = \sqrt{\gamma / \psi}$
  - interpolate from cells to faces fields of density  $\rho$ , velocity  $\vec{U}$ , pressure  $p$ , heat capacities ratio  $\gamma$ , speed of sound  $c$ , total specific enthalpy  $h^{tot} = \epsilon + \frac{\vec{U} \cdot \vec{U}}{2} + p / \rho$ , specific heat capacity at constant pressure  $C_p$ .
4. Calculate molecular viscosity and heat conductivity for flows of viscous gases.

5. Calculate QGD coefficients:
  - (a) QGD dissipative coefficient  $\tau$ ;
  - (b) QGD artificial viscosity coefficient  $\mu^{QGD}$ ;
6. Calculate gradients of pressure and velocity as well as velocity divergence at face centers using the least square method.
7. Calculate mass flux field  $\vec{j}_m$ .
8. Calculate momentum flux field  $\hat{\Pi}$ .
9. Calculate energy flux field.
10. Check CFL criterion.
11. Update time step increment  $\Delta t$ , increase time  $t^n = t^o + \Delta t$ .
12. Solve continuity equation.
13. Solve inviscid part of momentum equation.
14. Correct velocity at boundaries.
15. For viscous flow case, update velocity field by solving parabolic part of the momentum balance equation.
16. Update momentum field with new velocities and densities.
17. Solve inviscid part of energy equation.
18. Correct specific energy at boundaries.
19. For viscous flow case, update specific energy field by solving parabolic part of the energy balance equation.
20. Update energy field with new specific energy and density.
21. Correct pressure with EoS using new density and compressibility.
22. Update density at boundaries.
23. Go to next step (step 2).

#### 4. Solver validation for 1D flows

This section considers the Riemann problems discussed in, e.g. [25,26]. They reflect the characteristic features of unsteady gas flows with strong shock waves that are difficult for numerical simulation. The initial data for the Riemann problems are listed in the table according to the notations used in [26]. Specifically, the flow parameters on the left and right of the discontinuity are denoted by the indices  $L$  and  $R$ , respectively. The time at which the plots are shown is denoted by  $t_{fin}$ .

The boundary conditions are the same as the corresponding initial conditions at the ends of the computational domain. In all computations,  $\gamma = 1.4$ , except for the Noh problem (Test 3) with  $\gamma = 5/3$ . The length of the computational domain is equal to 1, from  $x = -0.5$  to  $x = +0.5$ . The discontinuity is placed at  $x = 0$ .

We compare the results obtained by *QGDFOAM* and *rhoCentralFoam* solvers. All solutions for *QGDFOAM* solver can be obtained with regularization parameter  $\alpha = 0.4$  and numerical coefficients  $Sc^{QGD} = 1$ . Particular tuning parameters allowing to improve the numerical solutions for *QGDFOAM* and *rhoCentralFoam* are mentioned in the text. *QGDFOAM* and *rhoCentralFoam* solvers use a constant Courant number  $Co$  and a variable time step.

In the OpenFOAM formulation, the problems are always solved in the dimensional form. To solve problems in the non-dimensional form using OpenFOAM, initial data should be normalized in an appropriate way, for example by scaling molecular mass  $M$  of a gas. The dimensional form for the tests presented below are constructed as follows:  $L = 1$  m,  $C_v = 1.0$  J/(kg K),  $R = \mathcal{R}/M = 0.4$  J/(kg K), where  $M = 20,785$  is the molecular mass and  $\mathcal{R}$  is the universal gas constant. The expression

$$C_p = \frac{\gamma R/M}{\gamma - 1}$$

is used to obtain  $\gamma$  values. For  $\gamma = 1.4$  we assign  $C_p = 1.4$  J/(kg K), and for  $\gamma = 5/3$  we assign  $C_p = 1$  J/(kg K).

**Test 1 – Sod problem.** The resulting flow presents all characteristic features of supersonic flows: sonic points at the boundaries of a rarefaction wave, a contact discontinuity, and a shock wave.

Fig. 2 shows the density profile for space grid step  $\Delta_h = 0.0025$  with *QGDFOAM* (Courant number  $Co = 0.4$ ) and *rhoCentral-*

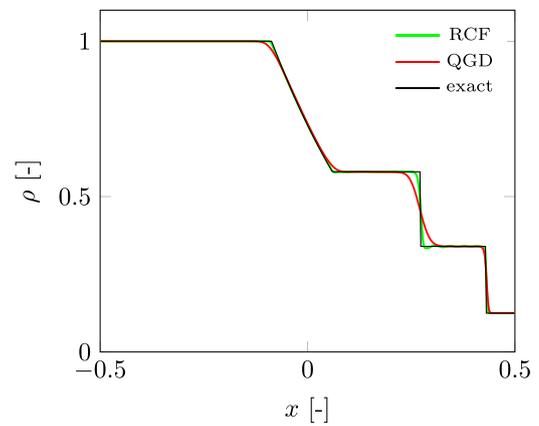


Fig. 2. Test 1. Distributions of density  $\rho$ , “RCF” – *rhoCentralFoam*, “QGD” – *QGDFOAM*, “exact” – exact solution.

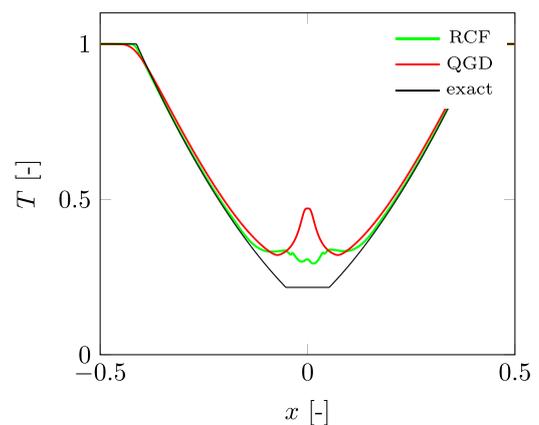


Fig. 3. Test 2. Distributions of the temperature  $T$ . “RCF” – *rhoCentralFoam*, “QGD” – *QGDFOAM*, “exact” – exact solution.

*Foam* (Courant number  $Co = 0.1$ ). The best tuning parameters for *QGDFOAM* are  $\alpha = 0.1$  and  $Sc^{QGD} = 0.001$ . For *rhoCentralFoam* the standard interpolation vanLeer scheme was used.

**Test 2 – Divergent waves.** Here the flow consists of two rarefaction waves that propagate away from the center of the domain. The difficulty in the numerical solution of this problem is that the gas density, velocity and pressure at the center (between the divergent flows) are very low, while the specific internal energy  $u = p/(\rho(\gamma - 1))$  does not tend to zero. It seems that there are no scheme in Eulerian variables that describes the behavior of the internal energy with high accuracy. The non-physical maximum of the internal energy near the point  $x = 0$  is called “an entropy tail”. Note, that all  $\tau$ -terms in this region are equal to zero, and the cause of this maximum is related to the rough approximation of the convective terms.

Fig. 3 presents the distribution of the specific internal energy  $u$  for  $\Delta_h = 0.0003125$  with *QGDFOAM* ( $Co = 0.1$ ) and *rhoCentralFoam* ( $Co = 0.025$ ). For *QGDFOAM* the minimal entropy tail was obtained with  $\alpha = 0.1$  and  $Sc^{QGD} = 1$ . This non-physical maximum can be smoothed using low artificial value of the Prandtl number  $Pr = 0.01$ . For *rhoCentralFoam* solver Minmod limiter was used.

**Test 3 – Noh problem.** The flow is formed by the collision of two hypersonic flows of a cold dense gas. As a result, two diverging “infinitely strong” shock waves are formed between which a stationary gas with a constant density and pressure remains. The estimations of the Mach number in the shock waves give  $Ma = U_L/c_c = 775$ . Fig. 4 shows the density profile for space grid step  $\Delta_h = 0.005$  with *QGDFOAM* ( $Co = 0.001$ ), and *rhoCentralFoam* ( $Co = 0.2$ ). The ar-

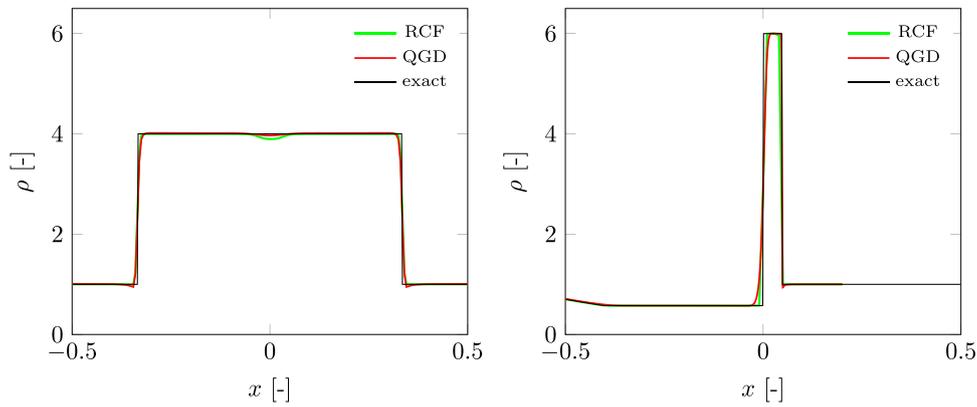


Fig. 4. Left - Test 3, right - Test 3a. Distributions of density  $\rho$ . “RCF” - *rhoCentralFoam*, “QGD” - *QGDFoam*, “exact” - exact solution.

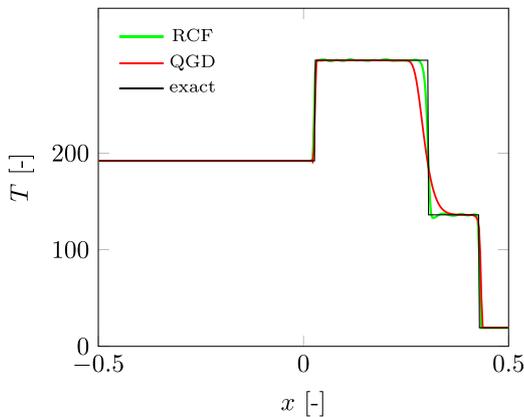


Fig. 5. Test 4. Distributions of the temperature  $T$ . “RCF” - *rhoCentralFoam*, “QGD” - *QGDFoam*, “exact” - exact solution.

tificial entropy tail formed by *rhoCentralFoam* solver near the point  $x = 0$  is seen. In QGD formulation the tail is not visible and in the corresponding region all  $\tau$ -terms are zero. The QGD solution does not depend on  $\alpha$  in the range [0.1–0.5]. For *rhoCentralFoam* we used Minmod scheme.

**Test 3a - Compression waves.** Gas compression in this flow corresponds to a pressure drop  $p_L/p_R = 10^5$ , which is complicated for numerical simulation. Fig. 4 shows the density distributions for  $\Delta_h = 0.0008$ . The Courant number  $Co = 0.01$  was used for both *QGDFoam* and *rhoCentralFoam*. In *QGDFoam*  $\alpha = 0.4$  and  $Sc^{QGD} = 1$ . In *rhoCentralFoam* the standard interpolation vanLeer scheme is used.

**Test 4 - Strong shocks.** Here the gas flow has the form of two diverging shock waves with a moving contact discontinuity between them. Fig. 5 shows the density graphs for  $\Delta_h = 0.003$  for *QGDFoam* with  $Co = 0.1$  and *rhoCentralFoam* with  $Co = 0.2$ . In *QGDFoam*, we used  $\alpha = 0.3$  and  $Sc^{QGD} = 0.5$ . In *rhoCentralFoam*, the standard interpolation vanLeer scheme is used.

**Test 5 - Stationary contact.** The flow represents a stationary contact discontinuity. Fig. 6 shows the density calculations for  $\Delta_h = 0.01$  for *QGDFoam* with  $Co = 0.1$  and *rhoCentralFoam* with  $Co = 0.2$ . When artificial viscosity and heat conductivity are switched off ( $Sc^{QGD} = 0$ ) the *QGDFoam* contact discontinuity is limited to one mesh spacing. When the whole QGD dissipation is switched off ( $\tau = 0$ ), the numerical solution is absolutely unstable. In *rhoCentralFoam*, the standard interpolation vanLeer scheme is used.

**Test 6 - Moving contact.** This problem deals with a slowly moving contact discontinuity. Fig. 7 presents density graphs for  $\Delta_h = 0.01$ . We used  $Co = 0.1$  and  $Co = 0.2$  for *QGDFoam* and *rho-*

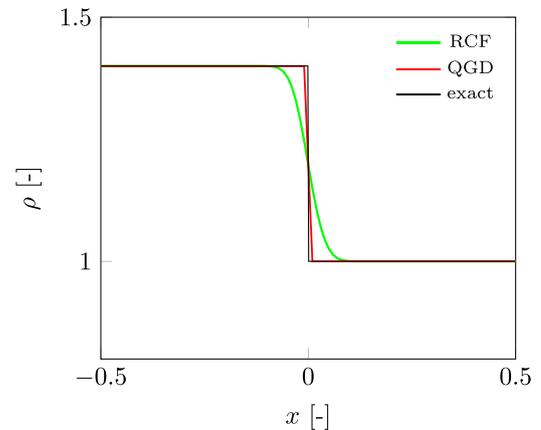


Fig. 6. Test 5. Distributions of density  $\rho$ . “RCF” - *rhoCentralFoam*, “QGD” - *QGDFoam*, “exact” - exact solution.

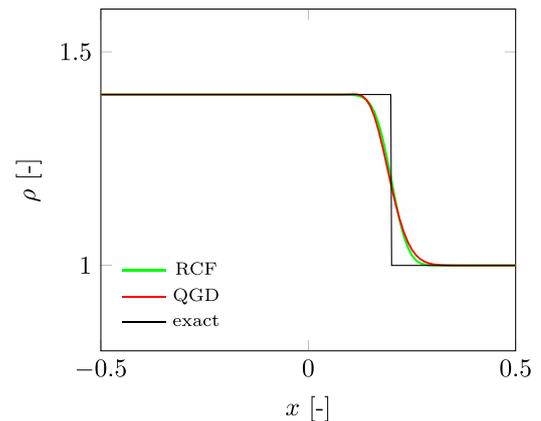


Fig. 7. Test 6. Distributions of density  $\rho$ . “RCF” - *rhoCentralFoam*, “QGD” - *QGDFoam*, “exact” - exact solution.

*CentralFoam*, respectively. In this case the decrease of the artificial viscosity and heat conductivity ( $Sc^{QGD} = 0.1$ ) in *QGDFoam* leads to a more accurate numerical solution that fully corresponds to *rhoCentralFoam* result. In *rhoCentralFoam* the standard interpolation vanLeer scheme is used.

**Test 7 - Peak problem.** Fig. 8 presents density graphs for  $\Delta_h = 0.00005$  for *QGDFoam* with  $Co = 0.05$  and *rhoCentralFoam* with  $Co = 0.3$ . The on-going process is evolving for a short time (see Table 1), that explains a small Courant number for QGD solver. For *QGDFoam*  $\alpha = 0.4$  and  $Sc^{QGD} = 1$  were used and for *rhoCentralFoam* we used upwind scheme. A comparison of  $L_1$ -norm of error,

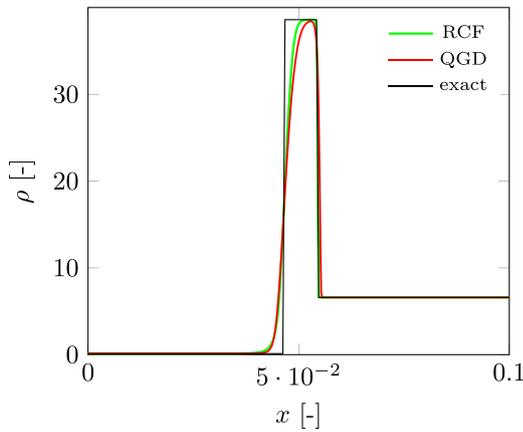


Fig. 8. Test 7. Distributions of density ρ “RCF” – rhoCentralFoam, “QGD” – QGDFoam, “exact” – exact solution.

Table 1  
Initial conditions for Riemann problems.

Test	$\rho_L$	$u_L$	$p_L$	$\rho_R$	$u_R$	$p_R$	$t_{fin}$
1	1	0.75	1	0.125	0	0.1	0.2
2	1	-2	0.4	1	2	0.4	0.15
3	1	1	$10^{-6}$	1	-1	$10^{-6}$	1
3a	1	-19.597	1000	1	-19.597	0.01	0.012
4	5.999	19.597	460.894	5.999	-6.196	46.095	0.035
5	1.4	0	1	1	0	1	2
6	1.4	0.1	1	1	0.1	1	2
7	0.126	8.904	782.928	6.591	2.265	3.154	0.0039

computed for Tests 1–7 is presented in Table 2.  $L_1$ -norm of error has been calculated on the self-similar solutions for 1D inviscid gas flow equations and a numerical approximation obtained on a uniform mesh from QGDFoam or rhoCentralFoam solvers, respectively:

$$L_1 = \frac{1}{N} \sum_{i=1}^N |a_i^{exact} - a_i^{num}|, \quad (26)$$

where  $N$  – number of computational points,  $a_i^{exact}$  – exact solution (density, velocity, etc.) at the  $i$ th computational point,  $a_i^{num}$  – numerical solution at the  $i$ th computational point calculated using rhoCentralFoam or QGDFoam solvers. For tests 1, 3, 3a, 5, 6, 7 the density  $\rho$  was used to measure  $L_1$ -norm and for tests 2 and 4 – internal energy  $u$ .

The above comparisons are the first direct comparison of QGD results with those of the widely used rhoCentralFoam numerical method in the frameworks of OpenFOAM. The comparison shows that QGD are competitive with rhoCentralFoam method, and is even better for a number of cases. For tests 1 and 2 the QGD Courant number exceeds the rhoCentralFoam  $Co$ . For tests 3a and 7 the  $Co$  values are equal. For tests 2 and 3 the entropy tail in the QGD formulation is smaller than in rhoCentralFoam one. In test 5 QGD solution coincides with the analytical solution. Nevertheless, additional tuning of the QGDFoam and rhoCentralFoam settings may bring further improvements for both methods.

### 5. Solver validation for 2D flows

To validate and verify the QGDFoam solver and to study properties of the scheme in comparison with previously developed OpenFOAM solvers other test cases are shown below, namely:

- supersonic inviscid flow over forward-facing step in a channel;
- subsonic viscous laminar flow over backward-facing step in a channel;
- supersonic underexpanded jet flow with Mach reflection;

- supersonic overexpanded jet flow with Mach reflection.

#### 5.1. Forward-facing step flow in a channel

Here we present the example of an inviscid supersonic flow in a planar channel with a ledge. A complicated configuration of shock waves serves as a known test to estimate the validity of numerical methods for solving Euler equations (see, e.g., [27]).

According to Woodward and Collela [27], the problem is solved in the following dimensionless form: the length of the channel is 3, its width is 1, the height of the ledge is 0.2, and its length is 2.4. We consider the flow of an inviscid, non-heat-conducting gas ( $\mu = k = 0$ ) with specific heat ratio  $\gamma = 1.4$  and  $Ma = 3$ . Recalculation of these parameters in dimensional form is performed as in the above Riemann problems.

The boundary conditions are given as follows: On the input boundary, the values of the gas dynamic parameters are assumed to be equal to the values of the incident flow, i.e.  $\rho = 1$ ,  $u_x = -Ma$ ,  $u_y = 0$ , and  $p = 1/\gamma$ . On the output boundary, we pose the “soft” boundary conditions  $\partial f/\partial x = 0$ , where  $f = (\rho, p, u_x, u_y)$ . On the rigid walls of the channel and ledge, we pose the “symmetry” boundary conditions:

$$\frac{\partial p}{\partial n} = 0, \quad \frac{\partial \rho}{\partial n} = 0, \quad \frac{\partial u_s}{\partial n} = 0, \quad u_n = 0,$$

where  $n$  is the normal, and  $s$  is the tangent to the corresponding boundary. As initial conditions the parameters of the incident flow are used.

The distribution of density at time  $t = 4$  for uniform space grid  $240 \times 80$  computed by QGDFoam is presented in Fig. 9 in comparison with rhoCentralFoam Upwind and rhoCentralFoam 2nd order TVD methods. For all three methods the Courant number is  $Co = 0.25$ . For QGD calculations we take numerical dissipation with coefficients  $\alpha = 0.3$  and  $Sc^{QGD} = 1$ .

All three methods clearly reproduce the formation of secondary waves reflecting from the upper wall of the channel and upper surface of the ledge. Behind the rarefaction wave, over the corner of the ledge, the gas density is at its minimum, and near the contact discontinuity, after the triple point over the ledge, the gas density is at its maximum.

The above results show that QGDFoam density distribution seems less smoothed compared with rhoCentralFoam upwind and more smoothed compared with rhoCentralFoam TVD 2nd order.

The grid convergence of QGD algorithm for  $120 \times 40$ ,  $240 \times 80$  and  $480 \times 160$  points shows the monotonic improvement of the numerical solution [5]. Our calculations for  $240 \times 80$ ,  $480 \times 160$  and  $1920 \times 640$  grid points show a similar behavior (Fig. 10).

#### 5.2. Backward-facing step flow in a channel

One of the important features of the QGD system and associated algorithm is its capability to simulate gas flows in a wide range of velocities, namely from laminar to turbulent regimes and from subsonic to hypersonic speeds. This remarkable property distinguishes QGDFoam from other solvers implemented in OpenFOAM, each of them having a limited range of applicability. For example, rhoCentralFoam fails to simulate low Mach number flows. To validate QGDFoam ability to simulate laminar subsonic flows, the problem of backward-facing step flow in a channel is considered [28]. This problem was well-studied both experimentally and numerically.

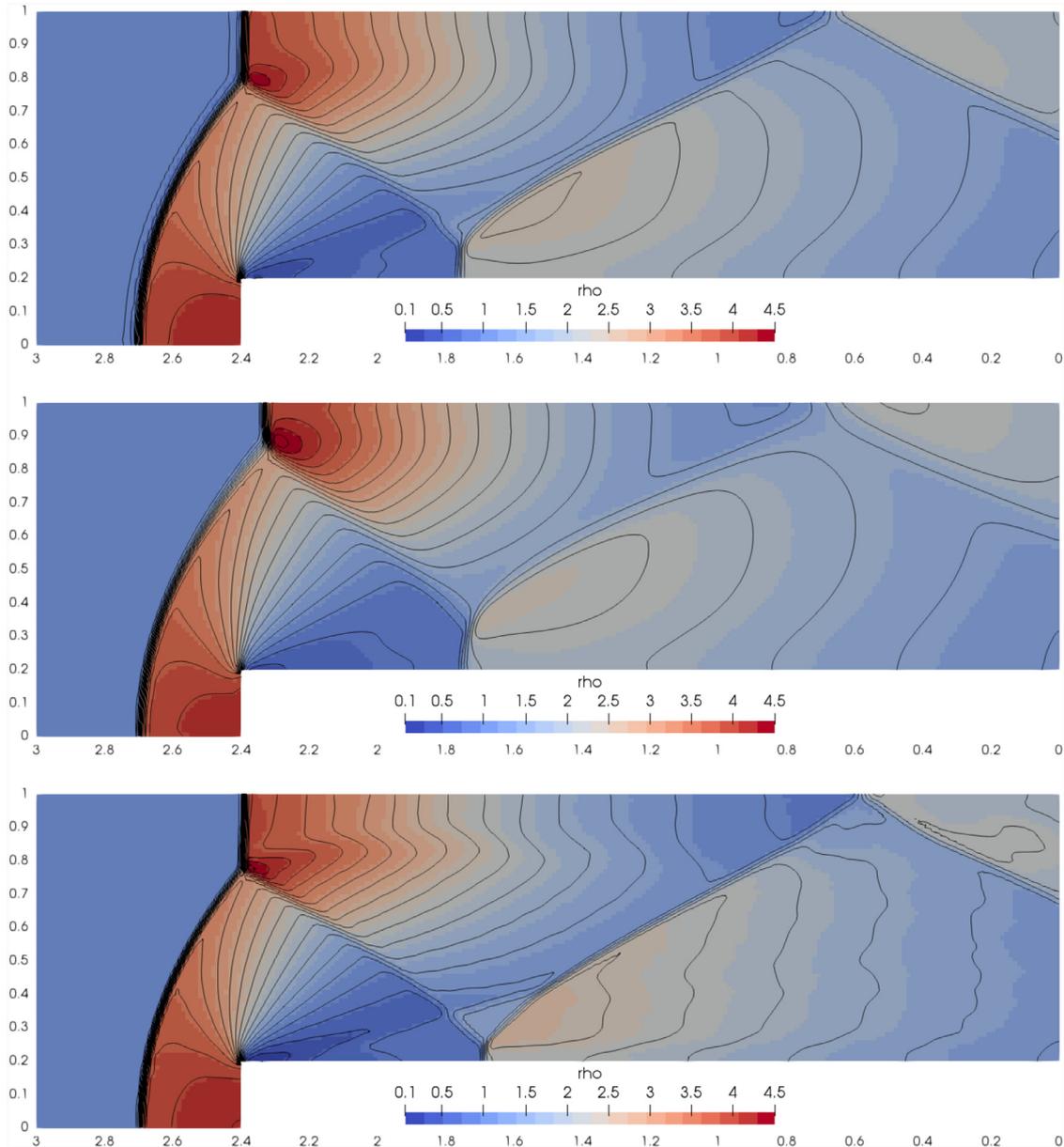
The length of the separation zone  $x_l$  behind the backward-facing step is one of the main parameters characterizing the flow. When the flow is laminar and two-dimensional, the dependence  $x_l(Re)$  is almost linear.

The computational domain consists of two straight channels with constant height. The short channel of height  $h$  expands

**Table 2**

Comparison of  $L_1$ -norm computed for Tests 1 – 7 using *rhoCentralFoam* (RCF) and *QGD* solvers.

Test no.	1	2	3	3a	4	5	6	7
RCF	0.0024	0.2726	0.0287	0.6935	1.3929	0.0103	0.0103	0.0532
QGD	0.0065	0.2909	0.0368	0.6849	3.6953	0.0021	0.0116	0.0775



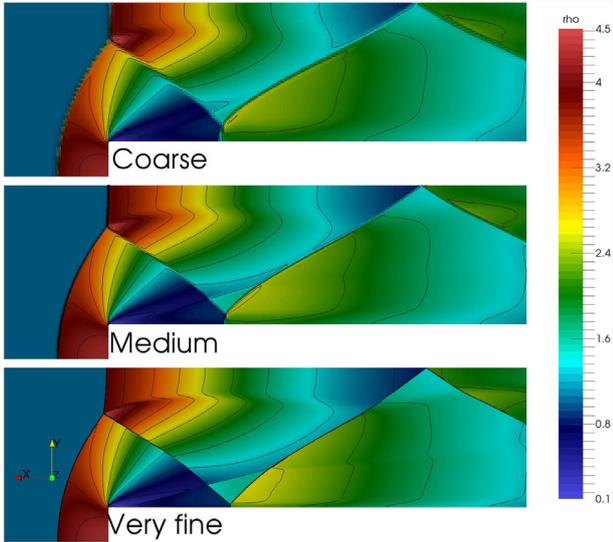
**Fig. 9.** Comparison of density contour plots for forward step case computed with QGD algorithm (top), *rhoCentralFoam* with upwind discretization (middle) and *rhoCentralFoam* with TVD 2nd order vanLeer limiter (bottom) at time  $t = 4$  s.

abruptly into the long one of height  $H$ , length  $L$ , and  $H = 2h$  (Fig. 11). The length of the short part is taken equal to  $h$  and length  $L = 26h$ .

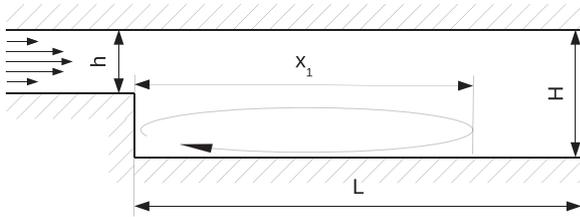
The flow in the inlet of the channel is laminar and obeys Poiseuille's law with parabolic velocity profile:  $U_x(y) = U_{max}(1 - y^2/h^2)$ , where  $y$  is measured from the center-line of the small channel. The value of the maximum velocity  $U_{max}$  is calculated from the average velocity of the flow using relation for volumetric flux  $U_{max} = 3/2U_{av}$ . The value of the average velocity  $U_{av}$  is determined by the value of far field Mach number  $U_{av} = Ma \cdot c$ .

Mach number  $Ma$  in experiment [28] was less than 0.01, which indicates a deep subsonic regime. To save computational costs, we select far field Mach number  $Ma = 0.1$ . For all calculations the average inlet velocity was kept identical.

The value of Reynolds number is varied by changing the viscosity of the fluid:  $Re_h = \rho U_{av} h / \mu$ . Note, that in [28]  $H$  was taken as a characteristic length. According to experimental data [28], the flow for  $Re_h$  up to 400 is two-dimensional and steady. The length of the separation zone  $x_1$  is deduced from the location where  $U_x$  changes its sign.



**Fig. 10.** Comparison of density distribution plots for forward step computed using QGD algorithm at 3 different grid distributions: a) coarse 240 × 80, b) medium 480 × 160 and c) very fine 1920 × 640.



**Fig. 11.** Sketch of the computational domain for the backward-facing step in a channel problem.

**Table 3**  
Comparison of calculated and experimentally measured length of separation zone  $x_1$  normalized by the step height  $h$ .

$Re$	50	100	200	300	400
Experiment [28]	3.0	5	8.5	11.3	14.2
<i>simpleFoam</i>	2.8	4.8	8.1	10.2	11.5
<i>QGDfoam</i>	2.9	4.9	8.2	10.2	11.6

All calculations are performed up to establishment of the steady state solution, which corresponds to the physical time needed for the flow to pass 5 or more times through the domain. Boundary conditions are set as follows:

- Inflow – parabolic profile for horizontal component of velocity  $U_x$ ,  $U_y = 0$ ,  $\partial p / \partial x = 0$  and  $T = const$ , that ensures the specified far-field Mach number;
- Outflow – fixed pressure and zero normal derivative for velocity components and temperature (subsonic outlet).
- Rigid walls – zero velocity components  $U_x = U_y = 0$  and zero normal gradient for pressure and temperature (no slip impermeable and adiabatic wall).

Calculations are performed using artificial QGD dissipation with coefficients  $\alpha = 0.5$  and  $Sc^{QGD} = 0$ . Uniform spatial grids with 10, 20 and 40 cells per height  $h$  were used. For the last two grids the difference in separation zone length  $x_1$  was negligible, which proves the grid convergence of the numerical solution.

The *rhoCentralFoam* fails in computing this problem, and the QGD results are compared with calculations made by *simpleFoam* standard OpenFOAM solver, see Table 3. The *simpleFoam* implements numerical FVM approximation to incompressible steady-

state Navier–Stokes equations for laminar and turbulent flows using SIMPLE algorithm [23]. In *simpleFoam* central difference schemes are used to approximate both convective and diffusive terms. From the Table 3 it is seen that for Reynolds numbers up to 300, the results for both methods are in a good agreement with the experiment. For  $Re_h \geq 400$ , the difference increases. For QGD calculations this difference is connected with the compressibility effects, due to with  $Ma = 0.1$  in the computations and  $Ma < 0.001$  in the experiments, see [5].

Flow visualization for  $Re_h = 200$  and 400 is shown on Fig. 12.

### 5.3. Underexpanded Ladenburg nozzle jet simulation

For the first jet flow benchmark, the Landenburg supersonic nozzle experiment has been selected. This case was successfully used for verification of *rhoCentralFoam* [21], making it a good tool for quantitative analysis of the developed *QGDfoam* solver. Within this experiment, Mach reflection and the formation of a Mach disk are studied. The turbulent effects can be neglected and the flow in the first shock cell behind the nozzle exit can be considered as inviscid.

Fig. 13 presents a sketch of the computational domain and the definition of boundaries. In the considered experiment, the pressure ratio between the inlet and the ambient conditions is relatively small ( $\approx 4$ ). Thus the gas jet does not expand in radial direction farther than  $3/2R$ , and the size of the first expansion cell does not exceed  $3R$  in axial direction. This characteristic lengths are used to determine the size of the computational domain as  $3R \times 6R$ .

According to the experimental setup [29] the following boundary conditions were imposed for both *rhoCentralFoam* and *QGDfoam* solvers:

- Inlet – fixed velocity in axial direction 315.6 m/s, static pressure 2.72 bar and temperature 247.1 K.
- Rigid walls – zero velocity (no-slip condition), zero gradient for pressure and temperature (impermeable adiabatic wall).
- Outlet – mixed outlet condition depending on exit local Mach number:
  - transonic and supersonic flow - zero gradient for all fields;
  - subsonic flows - fixed pressure (1 bar) and zero gradient for velocity and temperature.

The jet is assumed to be an axisymmetric flow of perfect gas with molar mass  $M = 28.96$  g/mole, constant pressure specific heat  $C_p = 1004.5$  J/(kg K), and Sutherland’s law for viscosity:

$$\mu(T) = A_s \frac{T^{1.5}}{T + T_s},$$

where  $A_s = 1.45810^{-6} \frac{kg}{m \cdot s \cdot K^{0.5}}$ ,  $T_s = 110.4$  K.

The calculations were carried out for three increasing mesh resolutions: 10, 20 and 40 cells per characteristic length (radius) with uniform distribution in radial and axial directions. According to qualitative observations on *rhoCentralFoam* and *QGDfoam* numerical diffusion from the previous section, we used the following settings for simulations:

- *rhoCentralFoam* with 2nd order TVD scheme and Minmod limiter;
- *rhoCentralFoam* with 1st order upwind scheme;
- *QGDfoam* with  $\alpha = 0.3$  and  $Sc^{QGD} = 1$  for all three mesh resolutions;
- *QGDfoam* with  $\alpha = 0.3$  and  $Sc^{QGD} = 0.15$  for fine mesh resolution.

Comparing the axial position of the Mach reflection and its radial height, we can compare the numerical diffusion of the different schemes (see Table 4).

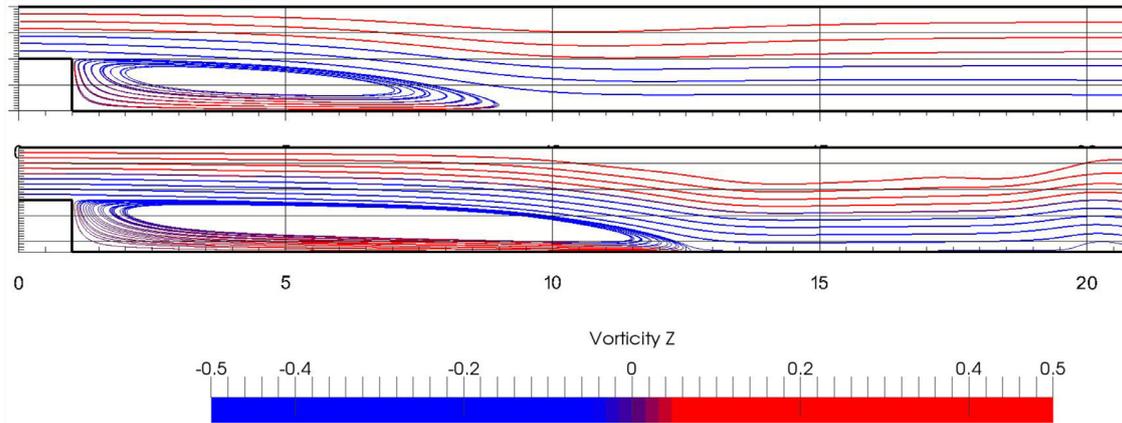


Fig. 12. Streamlines colored with vorticity for the backward-facing step in a channel at  $Re_h = 200$  (top) and  $Re_h = 400$  (bottom).

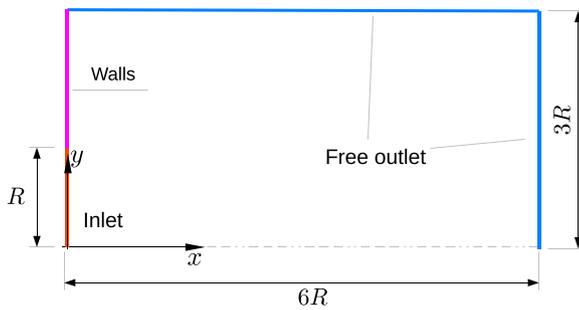


Fig. 13. Sketch of the computational domain for the simulation of Ladenburg supersonic jet flow.

Table 4

Comparison of triple point position calculated with different methods on the fine grid. The position is normalized by nozzle diameter 5.0 mm.

Solver/settings	First expansion cell length	Height of Mach disc cell length
Experiment	2.66	0.34
<i>rhoCentralFoam</i> 2nd order TVD Minmod	2.76	0.35
<i>rhoCentralFoam</i> 1st order upwind	2.94	0 (no Mach reflection)
<i>QGDFoam</i> $\alpha = 0.3$ , $Sc^{QGD} = 1$	2.93	0.225
<i>QGDFoam</i> $\alpha = 0.3$ , $Sc^{QGD} = 0.15$	2.76	0.34

For the first two mesh resolutions, numerical diffusion is large enough to suppress the formation of Mach disk, resulting in over-predicting of the length of the first expansion region. For finer mesh resolution, both *rhoCentralFoam* with 2nd order TVD scheme and *QGDFoam* resolve Mach reflection, whereas *rhoCentralFoam* with upwind discretization is still too diffusive.

The algorithm implemented in *QGDFoam* with  $Sc^{QGD} = 1.0$  is more diffusive than Kurganov–Tadmor scheme with 2nd order TVD approximation of convective fluxes, but far less diffusive than HLL scheme (Kurganov–Tadmor (KT) scheme with pure upwind). The *QGDFoam* results can be improved by adjusting artificial viscosity with  $Sc^{QGD}$  coefficient. For example, QGD algorithm produces a solution similar to KT with TVD by setting  $Sc^{QGD} = 0.15$  Fig. 14.

#### 5.4. Overexpanded supersonic jet simulation

Here the QGD implementation is tested to reproduce the shock cell structure in axisymmetric supersonic jet flows at over-expanded conditions. Reference data is a renowned NASA Langley experiment, conducted by Seiner [30,31] and widely used for CFD

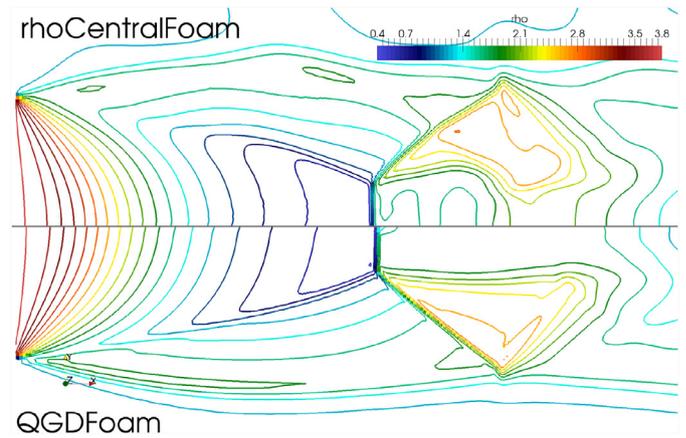


Fig. 14. Comparison of density contours calculated with *QGDFoam*  $\alpha = 0.3$ ,  $Sc = 0.15$  and *rhoCentralFoam* 2nd order TVD scheme with Minmod limiter.

code verification. In this experiment several types of nozzles were tested in a wide range of operation conditions, determined by parameter  $\beta_j$ , corresponding to ideal isentropic jet expansion:

$$\beta_j = \sqrt{M_j^2 - 1} \quad (27)$$

A nozzle with exit Mach number 2.0 and regime with  $\beta_j = 1.1$  was selected for verification. This regime corresponds to the following parameters at the nozzle exit section:

- Exit Mach number  $Ma = 2.0$ .
- Static pressure  $p = 46,632$  Pa.
- Static temperature  $T = 163.5$  K.
- Gas velocity  $U = 512.3$  m/s.

According to Kudimov et al. [32], a supersonic jet under these conditions can be divided into three regions:

1. Starting region with an almost inviscid behavior, encompassing approximately first 2 shock cells.
2. Transitional region with a momentum exchange between surrounding quiescent medium and supersonic jet.
3. Turbulent region with a fully developed turbulent flow at trans- and subsonic velocities.

Developed turbulent flows are outside of the scope of this work. Only the starting part of transitional regions are selected for comparison with experimental data ( $\approx 6$  radius from the nozzle exit). The computational model and the associated boundaries are similar to those used in previous section, Fig. 13. The flowing gas

is assumed to be viscous and to obey the perfect-gas equation-of-state: molar mass 28.96 g/mole, adiabatic heat capacities ratio  $\gamma = 1.4$ , isobaric specific heat capacity  $C_p = 1005$  J/(kg K) and constant dynamic viscosity  $\mu = 1.8 \times 10^{-5}$  Pa s. According to experiment, standard ambient conditions are used: pressure 101,325 Pa and temperature 292.15 K. Since OpenFOAM is essentially a 3D code, axial symmetry is modeled as wedge of small angle  $2^\circ$  with one cell in tangential direction. The computational domain is discretized as follows:

- the region of size  $6R \times R$ , which is located behind the nozzle exit and is expected to encompass the major part of the jet, is meshed with a uniform grid.
- the rest of the computational domain (external region) is meshed with a nonuniform grid cells that increase with the distance from the nozzle exit plane.

The following meshes are used:

- coarse mesh – 10 cells per radius, total mesh size is 7580 elements;
- medium mesh – 20 cells per radius, total mesh size is 30,320 elements;
- very fine mesh – 80 cells per radius, total mesh size is 365,120 elements;
- very very fine mesh – 120 cells per radius, total mesh size is 821,520 elements.

The boundary conditions are:

- Inlet: fixed values for pressure, temperature and velocity, according to nozzle exit parameters listed above.
- Rigid walls: zero value for all components of velocity (no-slip condition), zero normal gradient for pressure (impermeable wall) and zero normal gradient for temperature (adiabatic wall).
- Outlet: mixed boundary condition depending on local Mach number:
  - for subsonic Mach number fixed value of ambient pressure is imposed and zero gradient for velocity and temperature is set,
  - or trans- and supersonic velocities, zero normal gradient for all quantities is assigned.

According to previous studies, the parameters for the *QGDfoam* solver are set as  $\alpha = 0.3$  and  $Sc^{QGD} = 0.25$ . Time step is adjusted dynamically to keep  $CFL = 0.4$ .

On a coarse grid, the solution is stationary. Reducing the grid step leads to the appearance of oscillations in the mixing layer beginning with the 3rd shock cell. Figs. 15 and 16 are obtained after averaging the flow over the time interval  $3 \times 10^{-3}$  s, which corresponds to 10 flow passages through the computational domain. The instantaneous temperature and density fields are given for the smallest grid step “very very fine mesh” in Fig. 17.

Mesh convergence of the center-line time-averaged pressure distribution is compared with the experiment in Fig. 15. Here it is clearly seen that the QGD solution converges to experimental data (first two shock cells). Moreover, QGD algorithm resolves accurately the 3rd shock cell which is located in the beginning of the transitional region, where Kelvin–Helmholtz instability waves start to emerge, Fig. 17. Further discrepancies between calculation and experiment can be explained by both assumption of flow axisymmetry or insufficient space discretization.

The variant with the finest mesh is also simulated using *rhoCentralFoam* solver completed by  $k-\omega$  SST Reynolds Averaged Navier Stokes model [33]. Comparison of *QGDfoam* and *rhoCentralFoam* results shows a superiority of the QGD algorithm. Although *rhoCentralFoam* with RANS model qualitatively resolves shock cell structure in the inviscid region, it overpredicts the lengths of compression-expansion zones, Fig. 16.

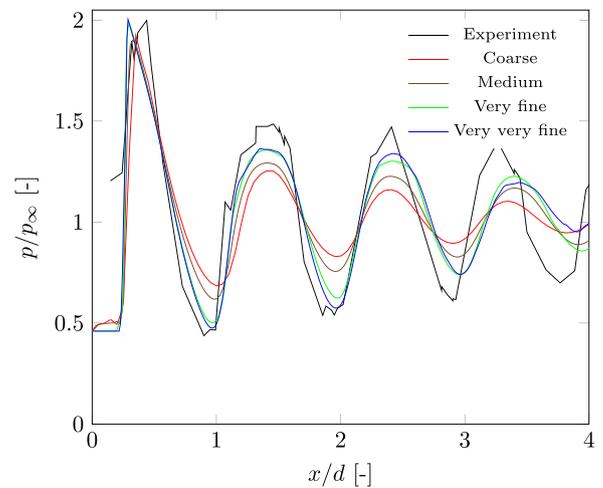


Fig. 15. Pressure distribution on the centerline of the nozzle computed with QGD equations for an under-expanded jet at different mesh resolution.

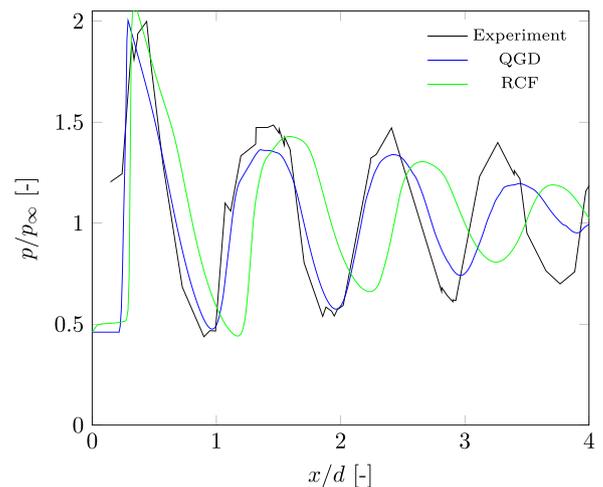


Fig. 16. Comparison of pressure distribution on the centerline of the nozzle computed with *QGDfoam* and *rhoCentralFoam* solvers on the finest mesh.

## 6. Solver performance

The performance of *QGDfoam* solver was analyzed using following cases:

1. serial calculation for the 1D Sod problem, Section 4;
2. serial calculation of 2D forward-facing step problem, Section 5.1;
3. parallel calculations of 2D forward-facing step problem, Section 5.1.

All calculations have been performed using a HP BL2x220 HPC cluster with Intel(R) Xeon(R) CPU X5670 @ 2.93 GHz 12-core processor nodes. For 1D simulations, we used a mesh with 8000 cells and constant time step  $\Delta t = 10^{-5}$  s. Calculations were performed for 25,000 time steps until the time  $t = 0.25$  s was reached.

For 2D simulations the finest quadrilateral mesh from Section 5.1 was used with approximately 1 million computational points. The time step was fixed at value  $\Delta t = 10^{-4}$  s, calculations were performed for 2500 time steps until the time  $t = 0.25$  s was reached.

For 1D simulations our implementation of *QGDfoam* is about 20% faster than *rhoCentralFoam*, showing 168 s versus 202 s of CPU time.

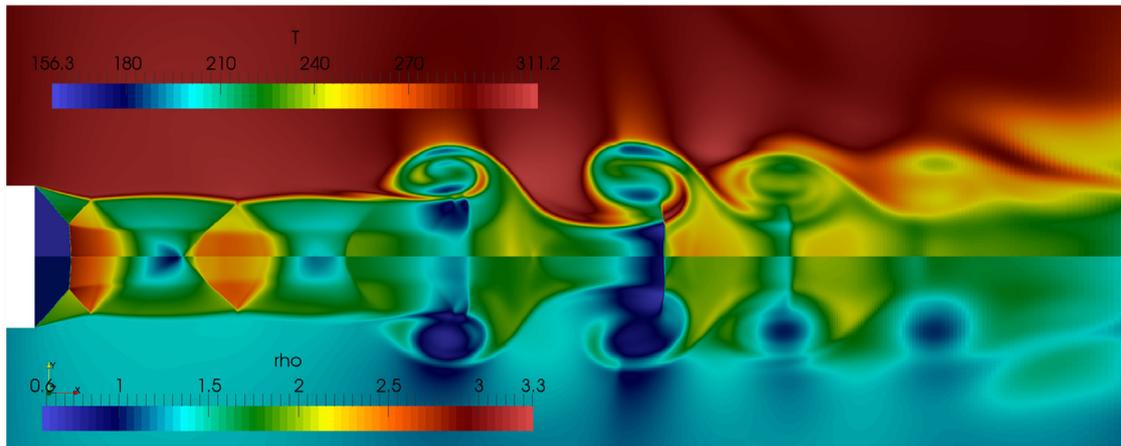


Fig. 17. Flow structure of supersonic jet at overexpanded conditions, computed with *QGDFoam* using finest mesh and  $\alpha = 0.3$ ,  $Sc^{QGD} = 0.25$ : temperature distribution (top) and density distribution (bottom).

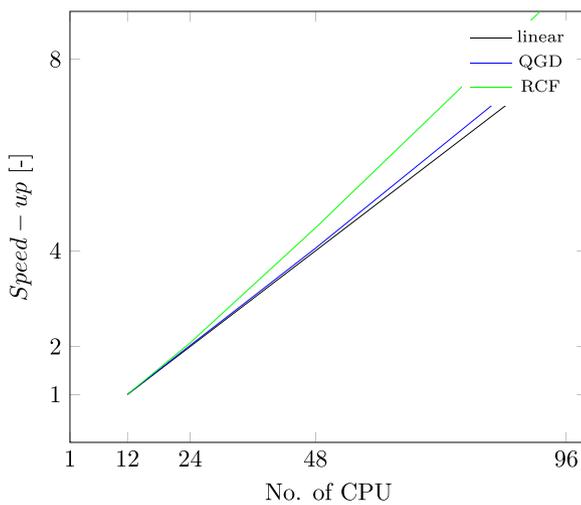


Fig. 18. Parallel efficiency of *QGDFoam* and *rhoCentralFoam* solvers.

For 2D serial simulations of forward-facing step *QGDFoam* solver is about 28% slower than *rhoCentralFoam*, showing **6941 s** versus **5032 s** of CPU time. The reason for such increase of computational cost is due to LSM approximation of QGD-terms at mesh faces which requires the calculation of spatial partial derivatives using values from adjacent cells. Optimization of expressions for approximation of QGD-terms should increase significantly the serial performance of *QGDFoam* solver. For example, efficient and fast algorithm for tetrahedral meshes can be found in [34,35] which can be extended to hexahedral meshes.

Parallel performance of *QGDFoam* together with *rhoCentralFoam* was evaluated for 96, 48, 24 and 12 cores. Results are presented in Fig. 18. It can be seen that speedup is near ideal line and is almost linear even for 96 CPUs ( $\approx 11$  thousands cells per node). Small deviations from linear line could be caused by operating system performance issues. Often it is more important to determine the lower threshold of parallel efficiency in terms of minimum number of grid cells per computational processor. To find this value, the forward-facing step case has been investigated with coarse mesh resolution (16128 cells), physical simulation time 4 s and time step  $0.25 \times 10^{-3}$  s, see Table 5.

According to above measurements of CPU time, computational costs of OpenFOAM implementation of QGD-algorithm are the next:

Table 5  
Parallel efficiency of *QGDFoam* solver.

No	No. of CPU	CPU time, s	Efficiency, %	No. of cells
1	1	585	–	16,128
2	2	327	89	8064
3	4	170	86	4032
4	6	126	77	2688
5	8	108	67	2016

- For 1D simulations –  $\approx 0.8 \times 10^{-6}$  s per cell per time step.
- For 2D simulations –  $\approx 2.2 \times 10^{-6}$  s –  $8 \times 10^{-6}$  s per cell per time step.

## 7. Conclusions

The new OpenFOAM solver for compressible perfect gas flow simulations in a wide Mach number range using regularized gas dynamics equations has been developed. Due to the nature of quasi-gas dynamics (QGD) equations, it is possible to avoid a number of techniques for numerical flux limiting widely used to suppress non-physical oscillations in the numerical solution of Euler equations. However, introduction of new second order terms in QGD equations compared with Navier–Stokes equations needs enhancement of the standard OpenFOAM's computational stencil for the calculation of partial derivatives at cell's faces. This issue was resolved by utilizing the least-square method and the finite-difference method applied to the unstructured grid. The simplicity of approximations used in the developed solver makes it a good tool for problems where massive parallel computations on unstructured grids are needed.

The developed solver has been tested for a number of Riemann 1D problems (Sod's problem, Noh test and others) and a few 2D cases – Mach 3 forward step, Ladenburg supersonic jet flow with Mach reflection, NASA Langley supersonic overexpanded jet flow and subsonic laminar flow over a backward-facing step. The testing procedure has shown that whereas QGD algorithm is more diffusive than Godunov-type methods with 2nd order TVD schemes with limiters, it is far less diffusive compared with pure upwind schemes as HLL. It was shown that the QGD algorithm allows to successfully resolve sonic and supersonic flows.

The diffusion of the QGD algorithm is determined essentially by introducing additional terms and can be tuned with parameters  $\alpha^{QGD}$  and  $Sc^{QGD}$  allowing direct control over numerical solution. Since our approximation of QGD equations has been designed to use unstructured grids with arbitrary cell shape, *QGDFoam* solver

can be potentially used for flows with complex geometries with-out diminishing the accuracy of approximation.

Scaling tests for up to 96 cores showed good scalability of QGDFoam solver (OpenFOAM implementation of QGD algorithm).

In future research we plan to test the solver for more complex flows, including at the incompressible limit, with 3D geometry and use of different OpenFOAM libraries (thermophysical properties, turbulence models, mesh motion and others). The recent version of the solver can be found at GIT repository <https://github.com/unicfdlab/QGDSolver>.

## Acknowledgments

This study was supported by the Program of Fundamental Research of the Presidium of the Russian Academy of Sciences No. 26 and by the RFBR according to research project No. 16-01-00048a.

The authors are grateful to J.-C. Lengrand for fruitful discussions and remarks to the paper presentation.

## References

- [1] Elizarova TG, Chetverushkin BN. On a computational algorithm for calculating gas-dynamic flows. *Dokl Akad Nauk SSSR* 1984;279(1):80–3.
- [2] Elizarova TG, Chetverushkin BN. Kinetic algorithms for calculating gas-dynamic flows. *Dokl Akad Nauk SSSR* 1985;25(5):164–9.
- [3] Chetverushkin BN. Kinetic schemes and quasi-gas dynamic system of equations. CIMNE Barcelona, Spain; 2009. ISBN 9788496736467.
- [4] Sheretov YV. Continuum dynamics under spatiotemporal averaging. *SPC Regular and Chaotic Dynamics*. (Moscow-Izhevsk, 2009, in Russian); 2009.
- [5] Elizarova TG. Quasi-gas dynamic equations. Springer; 2009. ISBN 978364200295.
- [6] Elizarova TG, Sheretov YV. Theoretical and numerical analysis of quasi-gasdynamic and quasi-hydrodynamic equations. *J Comput Math Math Phys* 2001;41(2):219–34.
- [7] Elizarova T, Dujsekulov A, Aspnas M. An implementation of gasdynamic problems on multiprocessor systems. *Reports Comput Sci Math, Abo Aademi, Ser-A* 1991(123).
- [8] Davydov AA, Shilnikov EV. Numerical simulation of the low compressible viscous gas flows on gpu-based hybrid supercomputers. *Adv Parallel Comput* 2014;25:315–23. *Parallel Computing: Accelerating Computational Science and Engineering (CSE)* <http://doi.org/10.3233/978-1-61499-381-0-315> doi: 10.3233/978-1-61499-381-0-315.
- [9] Davydov AA, Shilnikov EV. Parallel program complex express-3d for 3d flows simulation on hybrid computer systems. In: *Proceedings of the tenth international conference on advanced engineering computing and applications in sciences (ADVCOMP 2016)*. IARIA XPS Press, Wilmington, USA; 2016.
- [10] Antonov A, Elizarova T, Pavlov A, Chetverushkin B. Mathematical modeling of oscillating regimes during the flow of a body with a spike. *Math Modelirovanie* 1989;1(1):13–23.
- [11] Antonov A, Elizarova T, Chetverushkin B, Sheretov Y. Numerical modeling of pulsating regimes accompanying supersonic flow around a hollow cylinder. *J Comput Math Math Phys* 1990;30(2):139–44.
- [12] Elizarova T, Chetverushkin B, Sheretov Y. Quasigasdynamic equations and computer simulation of viscous gas flows. In: *Lecture notes in physics, N 414. Proceedings of 13th international conference on numerical methods in fluid dynamics*. Roma, Springer - Verlag; 1992.
- [13] Elizarova T, Graur I, Lengrand J, Chpoun A. Rarefied gas flow simulation based on quasi gas dynamic equations. *AIAA J* 1995;33(12):2316–24.
- [14] Graur I, Elizarova T, Ramos A, Tejada G, Fernandez J, Montero S. A study of shock waves in expanding flows on the basis of spectroscopic experiments and quasi-gasdynamic equations. *J Fluid Mech* 2004;504:239–70.
- [15] Elizarova T, Khokhlov A, Sheretov Y. Quasi-gasdynamic numerical algorithm for gas flow simulations. *Intern J for Numer Meth in Fluids* 2008;56(8):1209–15.
- [16] Zlotnik A, Elizarova T, Istomina M. Hydrodynamical aspects of the formation of spiral vortical structures in rotating gaseous disks. *Astron Rep* 2018;62(1):9–18.
- [17] Popov M, Elizarova T. Smoothed mhd equations for numerical simulations of ideal quasi-neutral gas dynamic flows. *Comput Phys Commun* 2015;1:348–61.
- [18] Shirokov IA, Elizarova TG. Simulation of laminar–turbulent transition in compressible Taylor–Green flow basing on quasi-gas dynamic equations. *J of Turbulence* 2014;15:707.
- [19] Elizarova T, Bulatov O. Regularized shallow water equations and a new method of simulation of the open channel flows. *Comput Fluids* 2011;46:206–11.
- [20] Elizarova T, Saburin D. Application of the regularized shallow water equations for numerical simulations of seiche level oscillations in the sea of azov. *Math Models Comput Simul* 2017;9(4):423–36.
- [21] Greenshields CJ, Weller HG, Gasparini L, Reese JM. Implementation of semi-discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high-speed viscous flows. *Int J Numer Methods Fluids* 2010;63(1):1–21.
- [22] Elizarova T. Time averaging as an approximate technique for constructing quasi-gasdynamic and quasi-hydrodynamic equations. *Comput Math Math Phys* 2011;51(11):1973–82.
- [23] Jasak H. Dynamic mesh handling in OpenFOAM. 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. American Institute of Aeronautics and Astronautics; 2009. URL <https://doi.org/10.2514/6.2009-341> doi: 10.2514/6.2009-341.
- [24] Kraposhin M, Bovtrikova A, Strijhak S. Adaptation of Kurganov–Tadmor numerical scheme for applying in combination with the PISO method in numerical simulation of flows in a wide range of mach numbers. *Procedia Comput Sci* 2015;66:43–52.
- [25] Elizarova TG, Shilnikov EV. Capabilities of a quasi-gas-dynamic algorithm as applied to inviscid gas flow simulation. *Comput Math and Math Phys* 2009;49:532–48.
- [26] Liska R, Wendroff B. Comparison of several difference schemes on 1d and 2d test problems for the euler equations. *SIAM J Sci Comput* 2003;25:995–1017.
- [27] Woodward P, Collela P. The numerical simulation of two dimensional fluid flow with strong shock. *Comput Phys* 1984;54:115–73.
- [28] Armaly BF, Durst F, Pereira JCF, Schönung B. Experimental and theoretical investigation of backward-facing step flow. *J Fluid Mech* 1983;127(-1):473. URL <https://doi.org/10.1017/s0022112083002839> doi: 10.1017/s0022112083002839.
- [29] Ladenburg R, van Voorhis C, Winkler J. Interferometric studies of faster than sound phenomena. Part II. Analysis of supersonic air jets. *Physical Review* 1949;76:662–77.
- [30] Norum T, Seiner J. Experiments of shock associated noise on supersonic jets. *AIAA 12th fluid and plasma dynamics conference*. Williamsburg, Virginia; 1979.
- [31] Norum T, Seiner J. Measurements of mean static pressure and far-field acoustics of shock-containing supersonic jets. Technical Report. Langley Research Center; 1982. NASA Technical Memorandum 84521.
- [32] Kudimov N, Safronov A, Tretyakova O. *Prikladnyye zadachi gasodinamiki i teploobmena v energeticheskikh ustanovkakh raketnoi tekhniki (in Russian)*. M.: MAI; 2014.
- [33] Menter FR., Kuntz M., Langtry R. Ten years of industrial experience with the SST turbulence model. *Begell, turbulence, heat and mass transfer 4: Proceedings of the fourth international symposium on turbulence, heat and mass transfer, Antalya, Turkey, Publisher: 2003 Begell House, Inc; 2003*.
- [34] Mitchell C. Improved reconstruction schemes for the Navier–Stokes equations on unstructured meshes. 32nd aerospace sciences meeting and exhibit. American Institute of Aeronautics and Astronautics; 1994. URL <https://doi.org/10.2514/6.1994-642> doi: 10.2514/6.1994-642.
- [35] Frink N. Assessment of an unstructured-grid method for predicting 3-d turbulent viscous flows. 34th aerospace sciences meeting and exhibit. American Institute of Aeronautics and Astronautics; 1996. URL <https://doi.org/10.2514/6.1996-292> doi: 10.2514/6.1996-292.