

# Implementation of the QGD algorithm using AMR technology and GPU parallel computing

Ivan But<sup>1,2</sup>[0000-0001-5915-0941], Andrey Epikhin<sup>1,2</sup>[0000-0002-6249-4283], Maria Kirushina<sup>2</sup>[0009-0008-8849-5218], and Tatiana Elizarova<sup>2</sup>[0000-0001-6169-5270]

<sup>1</sup> Ivannikov Institute for System Programming of the RAS, 109004, Russia

<sup>2</sup> Keldysh Institute of Applied Mathematics of the RAS, 125047, Russia

**Abstract.** The paper presents an algorithm based on the quasi-gasdynamic approach for the solution of unsteady compressible flows over a wide range of Mach numbers. It is implemented on the AMReX open platform, which uses adaptive mesh refinement technology to facilitate parallelization of computations on GPU architectures. To validate its effectiveness, the developed solver is applied to the numerical simulation of the shock-vortex interaction problem with flow parameter values of  $M_v = 0.9$  and  $M_s = 1.5$ . Cross-validation to assess its performance is conducted with OpenFOAM-based solvers, specifically rhoCentralFoam and QGDFoam. Schlieren fields are used to evaluate oscillations of the numerical schemes and algorithms, while resolution capabilities of the algorithm are assessed by comparing density fields in five cross-sections with the reference values.

**Keywords:** Shock-vortex interactions · Compressible flow · Quasi-gas dynamic equations · OpenFOAM · AMReX.

## 1 Introduction

The increasing complexity of physical processes modelling has made parallelization of computations on GPUs and integration of adaptive mesh refinement (AMR) technology essential [1]. GPUs, capable to solve massively parallel problems, offer a significant advantage in accelerating the computation of complex aerohydrodynamic problems. At the same time, AMR has emerged as a promising method to solve problems with complex and dynamic nature of aerohydrodynamic flows. The ability to automatically adjust mesh resolution in regions of interest not only improves simulation accuracy, but also optimises computational resources by allocating higher resolution only where necessary.

The Quasi-Gasdynamic (QGD) algorithm based on the regularised equations [2] allows modelling of compressible ideal gas flows over a wide range of Mach numbers, from subsonic to supersonic. The QGD equations, which differ from the Navier-Stokes equations, include additional terms proportional to the small parameter  $\tau$ , which depends on the size of the computational cell and local sound velocity. In particular, the universality of the algorithm for all types of flows distinguishes it. This numerical algorithm was implemented in OpenFOAM

[3–6] and showed high efficiency in modelling complex unsteady flows and flows with strong discontinuities [7, 8]. While this algorithm has shown high efficiency in modelling unsteady flows and flows with strong discontinuities, it tends to be slower than the alternatives due to its numerical specificity.

To address this challenge, the use of AMR technology and the potential for GPU computations parallelization offers a solution to speed up calculations and reduce expended resources. Currently, the most suitable free platform to achieve this goal is AMReX [9, 10], which has demonstrated higher computational efficiency compared to OpenFOAM [11]. Moreover, as of 2023, AMReX is part of the Linux Foundation’s High-Performance Software Foundation [12], which means it will receive significant support and development.

The rest of the paper is structured as follows: Section 2 describes the mathematical model underlying QGD. Section 3 describes the implementation of QGD in AMReX by demonstrating the numerical algorithm and the structure of the AmrQGD solver. Section 4 presents the formulation of the shock-vortex interaction problem. Section 5 shows the computational results in AmrQGD and OpenFOAM-based cross-validation with three solvers. The performance study of the algorithms is presented in section 5.3. Section 6 contains a paper with the main conclusions.

## 2 Mathematical model

The regularized gas dynamics equations in the form of continuity, momentum, total energy and state equations are used for the implementation of the AmrQGD solver in AMReX:

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

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\mathbf{j}_m \otimes \mathbf{u}) + \nabla p = \nabla \cdot \hat{\sigma}, \quad (2)$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\mathbf{j}_m E) + \nabla \cdot \mathbf{q} = \nabla \cdot (\hat{\sigma} \mathbf{u}), \quad (3)$$

$$p = \rho RT, \quad (4)$$

where  $\rho$  is density;  $\mathbf{j}_m$  is mass flux density;  $\mathbf{u}$  is the velocity vector;  $p$  is pressure;  $E = e + |\mathbf{u}|^2/2$  is total energy,  $e$  is the specific internal gas energy;  $\hat{\sigma}$  is the viscous stress tensor,  $\mathbf{q}$  is heat flux;  $\otimes$  is the direct tensor product.

The presence of additional QGD terms  $\mathbf{w}$ ,  $\hat{\sigma}_{QGD}$ ,  $\mathbf{q}_{QGD}$  proportional to the small parameter  $\tau$  is the main difference of quasi-gasdynamic equations:

$$\mathbf{j}_m = \rho(\mathbf{u} - \mathbf{w}), \quad \mathbf{q} = \mathbf{q}_{NS} + \mathbf{q}_{QGD}, \quad \hat{\sigma} = \hat{\sigma}_{NS} + \hat{\sigma}_{QGD} \quad (5)$$

$$\mathbf{w} = \frac{\tau}{\rho} (\text{div}(\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p), \quad \mathbf{q}_{QGD} = -\tau \rho \mathbf{u}((\mathbf{u} \cdot \nabla)e + p(\mathbf{u} \cdot \nabla)/\rho)$$

$$\hat{\sigma}_{QGD} = \tau \mathbf{u} \otimes (\rho(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p) + \tau \hat{I}((\mathbf{u} \cdot \nabla)p + \gamma p \nabla \mathbf{u})$$

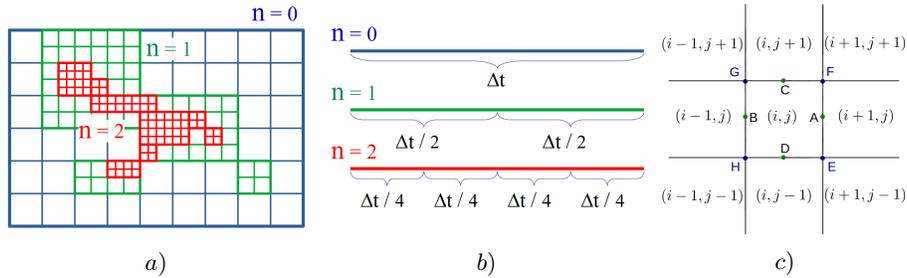
$$\tau = \alpha \frac{\Delta_h}{c} + \frac{\mu_d}{p}; \quad c = \sqrt{\gamma RT} \quad (6)$$

where  $\Delta_h$  is the local step size of the spatial mesh;  $c$  is the local speed of sound;  $\alpha$  is the tuning parameter of the algorithm;  $\mu_d$  is the dynamic viscosity coefficient;  $R$  is the specific gas constant. Viscosity coefficient  $\mu = \mu_d + \tau p S_{c_{QGD}}$ , where  $\alpha$  and  $S_{c_{QGD}}$  are the tuning parameter of the related numerical algorithm.

A detailed description of the derivation and characteristics of these equations is given in, e.g. [2, 5].

### 3 QGD implementation in AMReX

The AMReX software package allows us to use ready-made logic to refine mesh into levels with a corresponding change in time step (Fig. 1a,b), and also provides the possibility of transferring the computations to the GPU, which makes the computations much less time-consuming compared to OpenFOAM. On this basis, the implementation of the numerical algorithm based on the QGD equations in AMReX will significantly reduce the computational cost and increase the computational speed for problems in a wide range of Mach numbers.



**Fig. 1.** QGD implementation in AMReX: a) Scheme of adaptive mesh refinement by levels; b) Scheme of adaptive time step refinement at each level of the computational mesh; c) Scheme of the numerical QGD template implemented in AmrQGD.

#### 3.1 Numerical algorithm based on QGD in AMReX

Figure 1c shows the structure of the template for the QGD equations, which, unlike classical approaches, requires all surrounding cells. A two-dimensional (2D) numerical algorithm is currently implemented:

1. Discretization of  $\rho$ ,  $u_x$ ,  $v_y$ ,  $p$  is carried out by the central differences method:

$$\rho_A = \frac{\rho_{i,j} + \rho_{i+1,j}}{2}, \quad \rho_E = \frac{\rho_{i,j} + \rho_{i+1,j} + \rho_{i,j-1} + \rho_{i+1,j-1}}{4} \quad (7)$$

2. Calculation of the sound speed, the parameter  $\tau$ , the QGD part  $w$ ;
3. Calculation of mass flux density  $\mathbf{j}_m$ ;

4. The continuity equation is solved (the parameter with the hat is the variable value at the new time step):

$$\hat{\rho}_{i,j} = \rho_{i,j} - \frac{\Delta t}{\Delta x}(j_{mA} - j_{mB}) - \frac{\Delta t}{\Delta y}(j_{mC} - j_{mD}) \quad (8)$$

5. The viscosity coefficient, the Reynolds viscous stress tensor and its QGD analogue are considered;  
6. Calculation of the momentum equations:

$$\begin{aligned} \hat{u}_{i,j}^x = \rho_{i,j} u_{i,j}^x - \Delta t \left( \frac{j_{mA} u_A^x - j_{mB} u_B^x}{\Delta x} + \frac{j_{mC} u_C^x - j_{mD} u_D^x}{\Delta y} + \frac{p_A - p_B}{\Delta x} \right) \\ + \Delta t \left( \frac{\sigma_A^{xx} - \sigma_B^{xx}}{\Delta x} + \frac{\sigma_C^{yx} - \sigma_D^{yx}}{\Delta y} \right) \end{aligned} \quad (9)$$

$$\begin{aligned} \hat{u}_{i,j}^y = \rho_{i,j} u_{i,j}^y - \Delta t \left( \frac{j_{mA} u_A^y - j_{mB} u_B^y}{\Delta x} + \frac{j_{mC} u_C^y - j_{mD} u_D^y}{\Delta y} + \frac{p_C - p_D}{\Delta y} \right) \\ + \Delta t \left( \frac{\sigma_C^{yy} - \sigma_D^{yy}}{\Delta y} + \frac{\sigma_A^{xy} - \sigma_B^{xy}}{\Delta x} \right) \end{aligned} \quad (10)$$

7. Calculation of the temperature, heat transfer coefficient, specific internal energy, heat fluxes and enthalpy

$$H_A = \frac{(u_A^x)^2 + (u_A^y)^2}{2} + \gamma \frac{p_A}{\rho_A(\gamma - 1)}$$

8. Calculation of the energy balance equation:

$$\begin{aligned} \hat{E}_{i,j} = E_{i,j} - \Delta t \left( \frac{j_{mA} H_A - j_{mB} H_B}{\Delta x} + \frac{j_{mC} H_C - j_{mD} H_D}{\Delta y} + \frac{q_A - q_B}{\Delta x} \right. \\ \left. + \frac{q_C - q_D}{\Delta y} \right) + \Delta t \left( \frac{\sigma_A^{xx} u_A^x - \sigma_B^{xx} u_B^x}{\Delta x} + \frac{\sigma_C^{yy} u_C^y - \sigma_D^{yy} u_D^y}{\Delta y} \right. \\ \left. + \frac{\sigma_A^{xy} u_A^y - \sigma_B^{xy} u_B^y}{\Delta x} + \frac{\sigma_C^{yx} u_C^x - \sigma_D^{yx} u_D^x}{\Delta y} \right) \end{aligned} \quad (11)$$

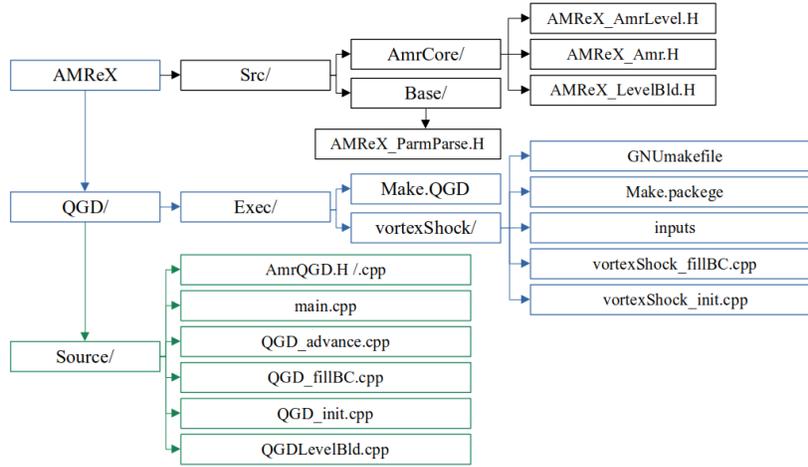
9. Pressure definition:

$$\hat{p}_{i,j} = (\gamma - 1) \left( \hat{E}_{i,j} - \hat{\rho}_{i,j} \frac{(\hat{u}_{i,j}^x)^2 + (\hat{u}_{i,j}^y)^2}{2} \right) \quad (12)$$

### 3.2 Solver structure

Figure 2 shows the structure of the developed QGD numerical algorithm in AMReX.

The **QGD/Source** folder contains the solver sources:



**Fig. 2.** Structure of the AmrQGD solver in AMReX. Black colour - AMReX kernel; Blue - case location folder including its initial and boundary conditions; Green - solver sources location.

- **AmrQGD.H** - the main solver class `AmrQGD` is declared in the file. It inherits from the `AmrLevel` class defined in the AMReX core;
- **AmrQGD.cpp** - the main solver class `AmrQGD` is implemented in this file;
- **main.cpp** - numerical algorithm;
- **QGD\_advance.cpp** - quasi-gasdynamic equations are implemented in this file;
- **QGD\_fillBC.cpp** - the file describes the boundary conditions;
- **QGD\_init.cpp** - initial conditions are set in this file;
- **QGDLevelBld.cpp** - the description of mesh refining.

The `QGD/Exec` folder contains statements of calculation tasks:

- **inputs** - the file where the calculation parameters are set;
- **vortexShock\_fillBC.cpp** - a file containing the boundary conditions;
- **vortexShock\_init.cpp** - file containing the initial conditions.

It is also worth noting that at each level it is not necessary to refine the whole mesh, but only some parts of it (Fig. 1a,b). For this purpose, the `errorEst` method is defined in the **AmrQGD.cpp** file. This method takes a reference to the `tags` instance of the `TagBoxArray` container. Each mesh cell is defined, which is marked for partitioning if it meets some condition (criterion). The remaining cells that do not meet the condition are not subject to partitioning.

Solver is available on GitHub [13]

## 4 Problem statement

To demonstrate features and advantages of the developed AmrQGD solver, it has been decided to choose a problem with complex unsteady flow, a good example

of which is a strong vortex-shock wave interaction. The complexity lies in the fact that the vortex passing through the shock is strongly deformed and actually splits into two coupled vortices, generating a large number of compression waves, which leads to numerical instability of the calculations.

Studies of the shock-vortex interaction problem were carried out for more than 70 years and include experimental [14, 15] and theoretical studies [16, 17]. In recent years, this problem was widely used to demonstrate the correctness of high-order methods [18–20]. In particular, this problem was used to evaluate algorithms based on QGD [21] equations, and our problem formulation is also consistent with this work. The reference values are taken from [22], where they were obtained by a high-order method.

The geometry of the calculation area is a rectangle of size  $2 \times 1$  m (Fig. 3a). Gas parameters are:  $\gamma = 1.4$ ,  $R = 1$ ,  $C_p = 3.5$ . At the initial moment of time the stationary shock wave  $M_s = 1.5$  is located vertically at  $x = 0.5$  m, to the left of it at the point with coordinates  $(0.25, 0.5)$  m there is a vortex with inner radius  $a = 0.075$  m and outer radius  $b = 0.175$  m,  $M_v = 0.9$ ,  $v_{max} = M_v \sqrt{\gamma}$ . Flow conditions before the shock ( $x < 0.5$ ):

$$(\rho, u_x, v_y, p, T)_{\text{left}} = (1, M_s \sqrt{\gamma}, 0, 1, p/(\rho R)) \quad (13)$$

Stationary shock conditions:

$$\frac{\rho_{\text{left}}}{\rho_{\text{right}}} = \frac{u_{\text{right}}}{u_{\text{left}}} = \frac{2 + (\gamma - 1)M_s^2}{(\gamma + 1)M_s^2}, \quad \frac{p_{\text{left}}}{p_{\text{right}}} = 1 + \frac{2\gamma}{\gamma + 1}(M_s^2 - 1), \quad v_{\text{right}} = 0 \quad (14)$$

The domain's initial conditions:

$$(\rho, u_x, v_y, p, T) = \begin{cases} (1, 1.77482, 0, 1, 1) & x < 0.5 \\ (1.862, 0.953146, 0, 2.45833, 1.32022) & x \geq 0.5 \end{cases} \quad (15)$$

The initial conditions in the vortex zone are shown in Figure 3b. The condition for the angular velocity of the vortex is calculated by the formula for vortex velocity  $v_\theta$ :

$$v_\theta(r) = \begin{cases} v_m \frac{r}{a} & r \leq a \\ v_m \frac{a}{a^2 - b^2} \left( r - \frac{b^2}{r} \right) & a < r \leq b \\ 0 & r > b \end{cases} \quad (16)$$

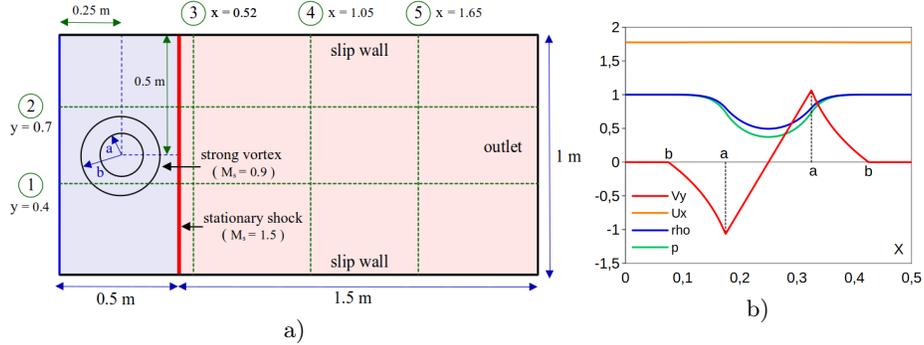
where  $r = \sqrt{(x - 0.25)^2 + (y - 0.5)^2}$  - radius from the vortex center. Then in the projection on the OX and OY axes:  $u_x(r) = u_{\text{left}} - v_\theta(r) \sin(\theta)$ ,  $v_y(r) = v_{\text{left}} + v_\theta(r) \cos(\theta)$

The temperature inside the vortex is set as:

$$T(r) = \begin{cases} T(b) - \frac{\gamma-1}{R\gamma} \frac{v_m^2}{a^2} \left( \frac{a^2 - r^2}{2} \right) - \\ - \frac{\gamma-1}{R\gamma} v_m^2 \frac{a^2}{\sqrt{a^2 - b^2}} \left( \frac{b^2 - a^2}{2} - 2b^2 \ln \frac{b}{a} - \frac{b^4}{2} \left( \frac{1}{b^2} - \frac{1}{a^2} \right) \right) & r \leq a \\ T(b) - \frac{\gamma-1}{R\gamma} v_m^2 \frac{a^2}{\sqrt{a^2 - b^2}} \left( \frac{b^2 - r^2}{2} - 2b^2 \ln \frac{b}{r} - \frac{b^4}{2} \left( \frac{1}{b^2} - \frac{1}{r^2} \right) \right) & a < r \leq b \\ 0 & r > b \end{cases} \quad (17)$$

Density and pressure inside the vortex are:

$$\rho(r) = \rho_{\text{left}} \left( \frac{T(r)}{T_{\text{left}}} \right)^{\frac{1}{\gamma-1}}, \quad p(r) = p_{\text{left}} \left( \frac{T(r)}{T_{\text{left}}} \right)^{\frac{\gamma}{\gamma-1}} \quad (18)$$



**Fig. 3.** Problem statement: a) Geometry of the computational domain; b) Initial conditions on the line  $Y = 0.5$  m in the vortex location region ( $a = 0.075$  m,  $b = 0.175$  m).

Boundary conditions:

- Smooth wall conditions are placed at the top and bottom (zero gradient for pressure and density, slip condition for velocity).
- Left (inlet): Conditions correspond to the initial parameters for  $x < 0.5$ .
- Right (outlet): Smooth boundary conditions (zero gradient).

The numerical Schlieren value fields are used to visualize the calculation  $\text{Sch} \in (0.05; 2.4)$ :  $\text{Sch} = \frac{\ln(1+\nabla\rho)}{\ln(10)}$

The density values on five lines are also compared (Fig. 3a), the reference values are taken from [22], obtained in this work by the higher order method on a mesh of  $12800 \times 6400$ .

## 5 Results and discussion

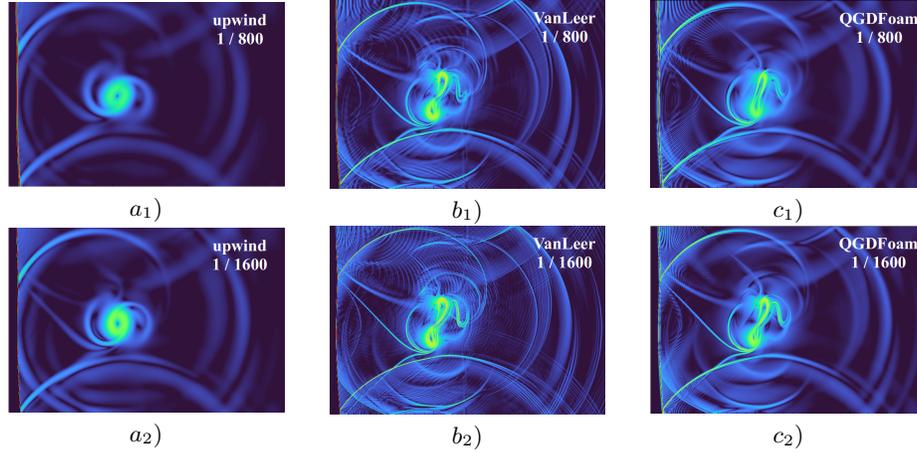
The vortex-shock wave interaction problem is numerically modelled on  $\frac{1}{400}, \frac{1}{800}, \frac{1}{1600}$  meshes along the OY axis and different Courant numbers. The results of the calculations are presented for the moment  $t_{\text{end}} = 0.7$ .

The following software packages and solvers are used for further cross-validation:

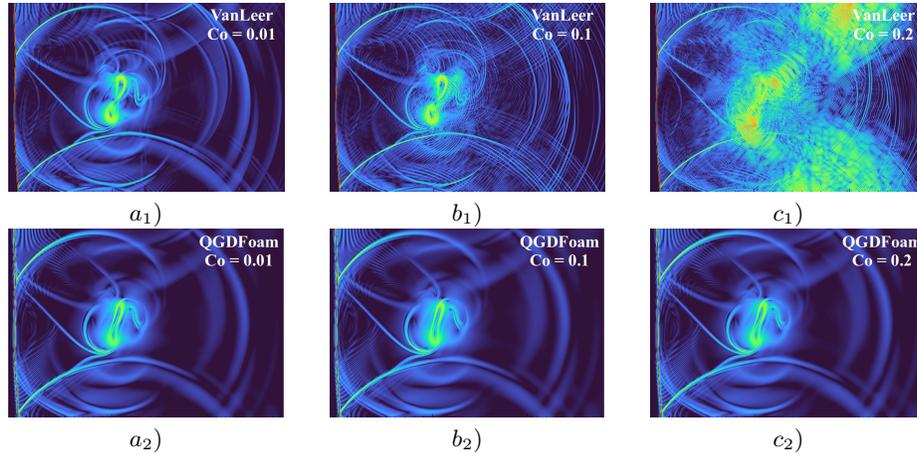
- rhoCentralFoam solver for compressible flows based on the central Kurganov-Tadmor schemes [23], implemented in OpenFOAM. The upwind [24] (1<sup>st</sup> order) and Van Leer [25] (2<sup>nd</sup> order) numerical schemes are used.
- QGDFoam [4] solver for flows over a wide range of Mach numbers, implemented in OpenFOAM and based on the QGD equations. [2]
- A new QGD-based equation solver AmrQGD implemented on AMReX software.

### 5.1 OpenFOAM (rhoCentralFoam and QGDFoam)

Figure 4 shows numerical Schlieren fields for OpenFOAM based solvers as a function of mesh cell size. Parameters of the numerical algorithm  $\alpha = 0.1$ ,  $Sc_{QGD} = 0.1$  are used for the QGDFoam, these parameters are defined in the paper [21] where this problem is considered.



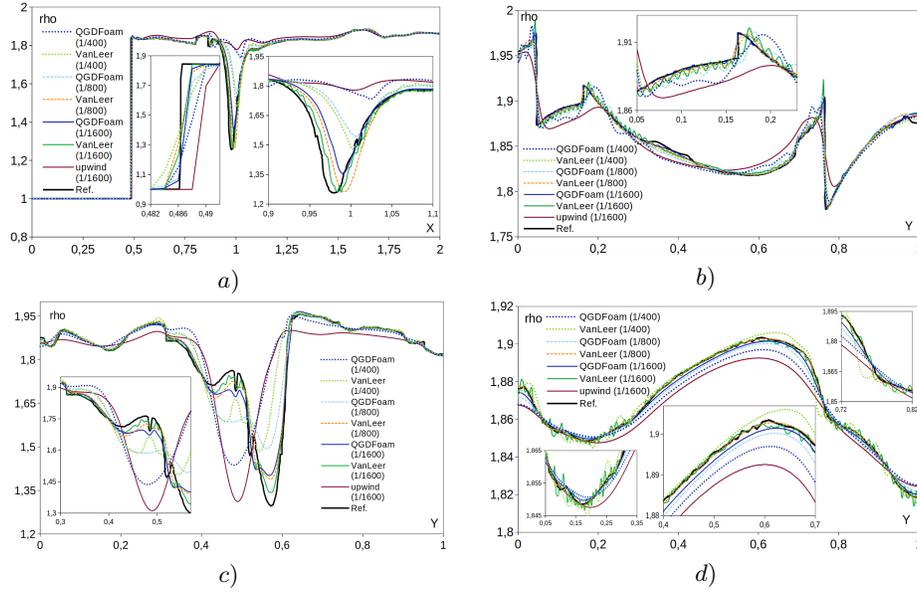
**Fig. 4.** Numerical Schlieren with  $\Delta t = 10^{-5}$ s and  $t_{end} = 0.7$ s.



**Fig. 5.** Numerical Schlieren on a 1/800 mesh.  $(a_1, b_1, c_1)$  - Van Leer;  $(a_2, b_2, c_2)$  - QGDFoam.  $(a_1, a_2)$ - $Co = 0.01$  ( $\Delta t = 0.5 \cdot 10^{-5}$ s);  $(b_1, b_2)$  -  $Co = 0.1$ ;  $(c_1, c_2)$  -  $Co = 0.2$ .

It can be seen that the upwind scheme does give an unacceptable solution even on a 1/1600 mesh, but the Van Leer and QGD schemes resolve the flow most correctly. However, when investigating different values of the time step (Fig. 5),

it is found that the Van Leer scheme is oscillatory, the QGD approach gives an acceptable solution at the Courant number  $Co = 0.2$ , while to obtain a non-oscillatory solution using the Van Leer scheme, the Courant number  $Co = 0.01$  is required.



**Fig. 6.** Comparison of the density field plots on the lines (Fig.3): a) Line 1; b) Line 3; c) Line 4; d) Line 5.  $t_{end} = 0.7s$ .

The density plots in Figure 6 show that the upwind scheme does not resolve the vortex structure, making it inapplicable in the context of the problem, while the Van Leer and QGD schemes correctly resolve the vortex structure and its features.

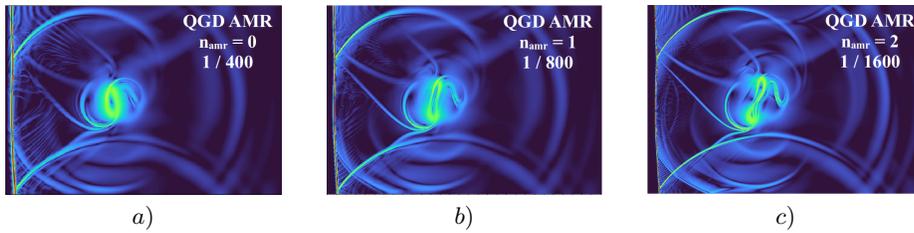
## 5.2 AmrQGD features

The QGD algorithm has two tuning parameters  $\alpha$  and  $Sc_{QGD}$ , in [21] it was noted that the best choice for this problem is  $\alpha = 0.1$  and  $Sc_{QGD} = 0.1$ , however these recommendations were given for numerical implementation based on the OpenFOAM package, so additional study is required for the newly developed solver. Figure 7 a gives the initial simulation result, which shows that the solution is free from numerical oscillations only at  $Sc_{QGD} = 0.5$ , but the solution is viscous due to the large value of  $Sc_{QGD}$ . In order to reduce oscillations, a local variation of the  $Sc_{QGD}$  number (variable varSc) is introduced, which is equal to 1 on the stationary shock when varSc = true. Figure 7 b shows the effect of the additional viscosity on the shock. It can be seen that this approach significantly reduced numerical oscillations after the vortex passes the shock, and an

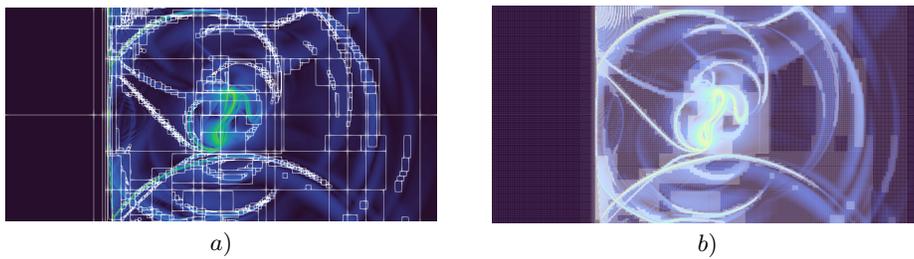
acceptable solution is obtained when  $\alpha = 0.1$ ,  $Sc_{QGD} = 0.1$ , which is similar to the numerical algorithm implemented in OpenFOAM. It is likely that the numerical implementation of the QGD algorithm in OpenFOAM automatically includes numerical limiters, resulting in less oscillations, while the numerical implementation in AMReX does not include limiters.



**Fig. 7.** Numerical Schlieren in QGD AMReX on a 1/1600 mesh with  $\alpha = 0.1$  and  $Sc_{QGD} = 0.1$ ,  $t_{end} = 0.7s$  : a) Dynamic Schmidt - **off**; b) Dynamic Schmidt - **on**,

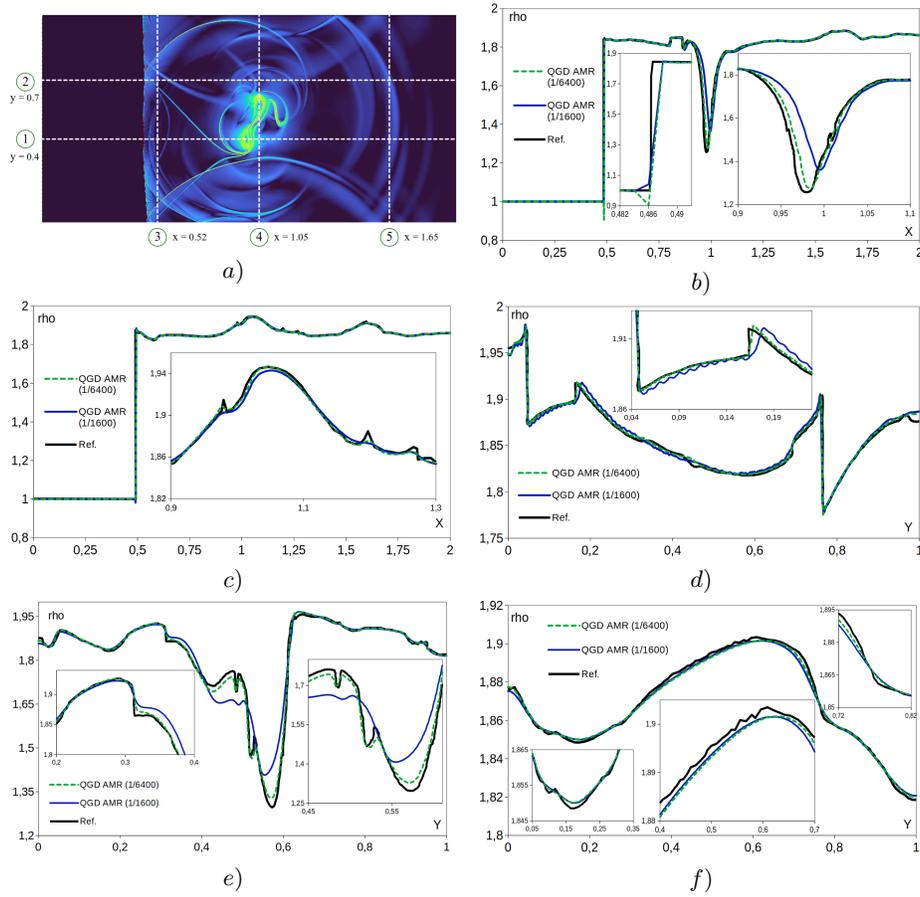


**Fig. 8.** Numerical Schlieren in QGD AMReX with  $\alpha = 0.1$ ,  $Sc_{QGD} = 0.1$ : a) 1/400; b) 1/800; c) 1/1600.



**Fig. 9.** Example of adaptive mesh refinement: a) by blocks; b) by cells.

The influence of the adaptive mesh refinement function (number of levels  $n_{amr}$ ) on the results obtained in AmrQGD is shown in figures (8-10). The adaptation is performed on the density gradient, i.e. if the gradient (the difference



**Fig. 10.** Comparison of density values on five reference lines: a) Numerical Schlieren field and five reference lines; b) Line 1; c) Line 2; d) Line 3; e) Line 3; d) Line 4; f) Line 5;  $t_{end} = 0.7s$ .

of the density values in the centers of neighboring cells divided by the distance between them) is greater than 0.0003, the mesh cell is split into 4 and the time step in each of them is reduced by half compared to the time step on the previous equation of the computational cell. Figure 8a shows the flow structure without splitting, on a mesh of 1/400. Figure 8b has a level of adaptation ( $n_{amr} = 1$ ), i.e. in a given region the mesh is equivalent to 1/800. In figure 8c there are 2 levels of adaptation ( $n_{amr} = 2$ ), corresponding to the mesh 1/1600 in the vortex resolution zone and reflected shock waves. Figure 9 shows the partitioning of the computational domain into blocks according to the density gradient criterion (fig. 9a) and the refinement of the computational mesh in these blocks (fig. 9b).

To demonstrate the capability of the developed numerical algorithm, a simulation of the flow with local mesh refinement corresponding to a mesh size of

1/6400 was performed (Fig. 10). The use of AMR technology allows the calculation for such a mesh refinement to be performed within 6 hours on 24 cores. Such a calculation on a stationary grid in the OpenFOAM package would have been much more computationally expensive.

### 5.3 Solving performance

A cluster of 12 Intel(R) Xeon(R) CPU 5160 @ 3.00GHz core nodes is used to evaluate the performance of the selected algorithms. Measurements are performed on an orthogonal mesh of 1/1600 on the OY axis with  $\Delta t = 10^{-5}$  s. and  $t_{end} = 0.01$  s. (Tab. 1). Where AmrQGD (off) means that there is no mesh adaptation, and AmrQGD (on) that two levels of adaptation are on. It can be seen that on 12 cores, AmrQGD ( $n_{amr} = 2$ ) computes 9 times faster than upwind, 15 times faster than Van Leer, 32 times faster than QGDFoam and 4.5 times faster than AmrQGD on a 1/1600 stationary mesh. As the number of cores increases, the speedup difference decreases, due to both the non-ideal parallelization of the algorithm in AmrQGD and the fact that the algorithm reaches its lower bound on the number of cells per core (which is about twenty thousand cells per CPU core).

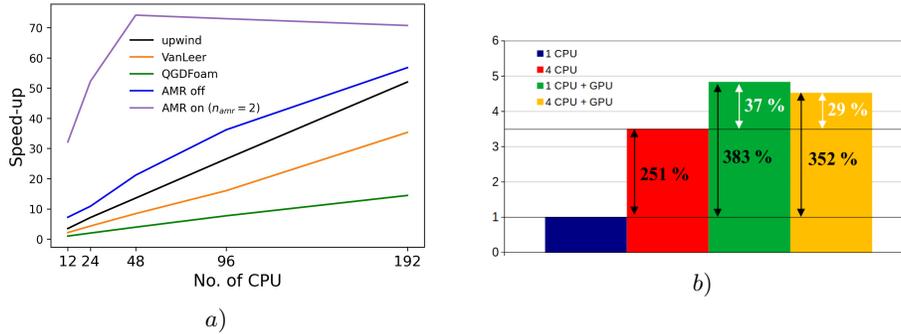
The study of GPU computing performance is conducted on the NVIDIA GeForce RTX 2060 card based on the TU106 processor. It is found that the computing time on one CPU core plus GPU is 4.8 times faster than on one CPU core and 1.4 times faster than on four CPU cores.

**Table 1.** Calculation time on the 1/1600 mesh is  $\Delta t = 10^{-5}$  s and  $t_{end} = 0.01$  s.

CPU	Cell per CPU	upwind	vanLeer	QGDFoam	AmrQGD (off)	AmrQGD (on)
12	426.7k	1317.23	2124.01	4597.02	636	143
24	213.3k	642.82	1060.39	2297.98	422	88
48	106.7k	338.25	544.53	1160.61	217	62
96	53.3k	173.05	286.46	595.18	127	63
192	26.7k	86.74	130.16	318.27	81	65

**Table 2.** Algorithms parallelization efficiency [%]. Computation mesh 1/400  $\Delta t = 10^{-5}$  s and  $t_{end} = 0.01$  s

CPU	Cell per CPU	upwind	vanLeer	QGDFoam	AmrQGD (off)	AmrQGD (on)
1	320k	-	-	-	-	-
2	160k	83	87	89	81	60
4	80k	78	81	86	79	47
8	40k	64	67	67	61	45



**Fig. 11.** Solving performance: a) Speedup of various algorithms on CPU; b) relative AMR speedup on GPU.

## 6 Conclusion

A new two-dimensional quasi-gasdynamics solver AmrQGD based on AMReX with the possibility of adaptive mesh refinement and parallelization of computations to graphics kernels is developed and described.

Cross-validation of the OpenFOAM and AMReX software packages, as well as the different upwind, Van Leer and QGD algorithms, the latter being implemented in two software packages simultaneously. It is found that the upwind algorithm does not give an acceptable solution on a reasonable grid and time step. The Van Leer algorithm gives an acceptable solution on a grid of  $1/1600$  and a time step of  $\Delta t = 10^{-5}$  s, corresponding to a Courant number of  $Co = 0.03$ . A comparable result is given by the QGD algorithm implemented in QGDFOAM and AmrQGD, with a Courant number of  $Co = 0.2$ .

When the performance of the algorithms is examined on an orthogonal grid of  $1/1600$  on the axis of the OY and with a time step  $\Delta t = 10^{-5}$  s, it is found that AmrQGD on 12 cores with two levels of adaptation is 15 times faster than Van Leer, 32 times faster than QGDFOAM and 4.5 times faster than AmrQGD on a stationary mesh. However, because AMReX is designed to handle large numbers of computational cells, the parallelization efficiency of QGD in AMReX decreases as the number of cells per core decreases. It is found that adding a GPU to the computation in AmrQGD can speed up the computation by up to 4.8 times.

It is planned to extend the presented AMReX QGD solver to three-dimensional flows, to optimize the parallelization of the numerical algorithm and to test it on a wide range of validation tasks in the future.

**Acknowledgments.** This work was supported by Moscow Center of Fundamental and Applied Mathematics, Agreement with the Ministry of Science and Higher Education of the Russian Federation, No. 075 – 15 – 2022 – 283

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Shalf John. "The future of computing beyond Moore's Law." *Philosophical Transactions of the Royal Society A* 378.2166 (2020). <https://doi.org/10.1098/rsta.2019.0061>
2. Elizarova T.G. *Quasi-gas-dynamic Equations*. Springer Berlin Heidelberg, (2009). <https://doi.org/10.1007/978-3-642-00292-2>
3. Jacobsen N.G., Fuhrman D.R., and Fredsøe J. "A wave generation toolbox for the open-source CFD library: OpenFoam®." *International Journal for numerical methods in fluids* 70.9 (2012). <https://doi.org/10.1002/flid.2726>
4. QGDSolvers, <https://github.com/unicfdlab/QGDSolver> Last accessed 19 Apr. 2024
5. Kraposhin M.V. et al. "Development of a new OpenFOAM solver using regularized gas dynamic equations." *Computers & Fluids* 166 (2018). <https://doi.org/10.1016/j.compfluid.2018.02.010>
6. Kraposhin M.V., Ryazanov D.A., and Elizarova T.G. "Numerical algorithm based on regularized equations for incompressible flow modeling and its implementation in OpenFOAM." *Computer Physics Communications* 271 (2022). <https://doi.org/10.1016/j.cpc.2021.108216>
7. Epikhin, A.S., Elizarova, T.G. Numerical simulation of underexpanded supersonic jets impingement on an inclined flat plate. *Thermophys. Aeromech.* 28, 479–486 (2021). <https://doi.org/10.1134/S0869864321040028>
8. Melnikova, Valeriia G., Andrey S. Epikhin, and Matvey V. Kraposhin. 2021. "The Eulerian–Lagrangian Approach for the Numerical Investigation of an Acoustic Field Generated by a High-Speed Gas-Droplet Flow" *Fluids* 6, no. 8: 274. <https://doi.org/10.3390/fluids6080274>
9. Zhang Weiqun et al. "AMReX: a framework for block-structured adaptive mesh refinement." *The Journal of Open Source Software* 4.37 (2019) <https://doi.org/10.21105/joss.01370>
10. Zhang Weiqun, et al. "AMReX: Block-structured adaptive mesh refinement for multiphysics applications." *The International Journal of High Performance Computing Applications* 35.6 (2021) <https://doi.org/10.1177/10943420211022811>
11. Andrey Epikhin and Ivan But. "Numerical Simulation of Supersonic Jet Noise Using Open Source Software." *International Conference on Computational Science*. Cham: Springer Nature Switzerland, (2023). [https://doi.org/10.1007/978-3-031-36030-5\\_24](https://doi.org/10.1007/978-3-031-36030-5_24)
12. TheLinuxFoundation, <https://www.linuxfoundation.org/press>. Last accessed 19 Apr. 2024
13. AmrQGDSolvers , <https://github.com/unicfdlab/AmrQGDSolvers>. Last accessed 19 Apr. 2024
14. Hollingsworth W.A. "A schlieren study of the interaction between a vortex and a shock wave in a shock tube." *British Aeronaut. Research Council Rept.* 17,985 (1955).
15. Dosanjh D.S. and Weeks T.M. "Interaction of a starting vortex as well as a vortex street with a traveling shock wave." *AIAA journal* 3.2 (1965). <https://doi.org/10.2514/3.2833>
16. Ribner H. S. "The sound generated by interaction of a single vortex with a shock wave." *University of Toronto*. (1959).
17. Dosanjh D.S. and Weeks T.M. "Sound generation by shock-vortex interaction." *AIAA Journal* 5.4 (1967). <https://doi.org/10.2514/3.4045>

18. Rault A., Chiavassa G., and Donat R. "Shock-vortex interactions at high Mach numbers." *Journal of Scientific Computing* 19 (2003) <https://doi.org/10.1023/A:1025316311633>
19. Rodionov A.V. "Simplified artificial viscosity approach for curing the shock instability." *Computers & Fluids* 219 (2021). <https://doi.org/10.1016/j.compfluid.2021.104873>
20. Kundu Abhishek and Gautam Biswas. "Analysis of multipolar vortices in the interaction of a shock with a strong moving vortex." *Computers & Fluids* 248 (2022). <https://doi.org/10.1016/j.compfluid.2022.105686>
21. Kirushina M.A., Elizarova T.G. and Epikhin A.S. "Simulation of Vortex Interaction with a Shock Wave for Testing Numerical Algorithms." *Mathematical Models and Computer Simulations* 15.2 (2023) <https://doi.org/10.1134/s2070048223020072>
22. 5th International workshop on high-order CFD methods, <https://how5.cenaero.be/> Last accessed 29 Feb 2024
23. Kurganov A., and Tadmor E. "New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations." *Journal of computational physics* 160.1 (2000). <https://doi.org/10.1006/jcph.2000.6459>
24. Swanson R.C., and Turkel E. "On central-difference and upwind schemes." *Journal of computational physics* 101.2 (1992). [https://doi.org/10.1016/0021-9991\(92\)90007-L](https://doi.org/10.1016/0021-9991(92)90007-L)
25. Van-Leer B. "Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme." *Journal of computational physics* 14.4 (1974). [https://doi.org/10.1016/0021-9991\(74\)90019-9](https://doi.org/10.1016/0021-9991(74)90019-9)