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

Ivan But$^{1,2[0000-0001-5915-0941]}$, Andrey Epikhin$^{1,2[0000-0002-6249-4283]}$, Maria Kirushina$^{2[2222-3333-4444-5555]}$, and Tatiana Elizarova$^{2[0000-0001-6169-5270]}$

[1] Ivannikov Institute for System Programming of the RAS, 109004, Russia
[2] Keldysh Institute of Applied Mathematics of the RAS, 125047, Russia

Abstract. The algorithm presented in this study uses a quasi-gasdynamic approach to address unsteady, compressible flows over a wide range of Mach numbers. This implementation is carried out within the AMReX open platform, which uses adaptive mesh refinement technology and facilitates the parallelization of computations on GPU architectures. To validate its effectiveness, the solver was applied to a specific scenario involving the shock-vortex interaction problem, with flow parameter values of $M_v = 0.9$ and $M_s = 1.5$. To assess its performance, cross-validation was performed with OpenFOAM-based solvers, specifically rhoCentralFoam and QGDFoam. Schlieren fields are used to evaluate the oscillations of the numerical schemes and algorithms, while the resolution capabilities of the algorithm are assessed by comparing the density fields in five cross-sections with reference values.

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

1 Introduction

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

Currently, the quasi-gasdynamic algorithm is implemented within the AMReX open platform as the 2D QGD solver, based on the quasi-gasdynamic equations [2]. This implementation allows the modelling of compressible ideal gas flows over a wide range of Mach numbers, from zero to infinity. The chosen algorithm has already been implemented in OpenFOAM [3,4,5,6] and has shown high efficiency in modelling unsteady flows and flows with strong discontinuities [9]. The QGD equations, which differ from the Navier-Stokes equations, include additional terms proportional to the small parameter τ, which depends on the size of the computational cell and its sound velocity. In particular, the universality of the algorithm for all types of flows distinguishes it. While this algorithm has shown high efficiency in modelling unsteady flows and flows with strong discontinuities, it tends to be slower than alternatives due to its numerical specificity.

To address this challenge, the use of AMR technology and the potential for parallelizing computations on GPUs offers a solution to speed up computations and reduce computational resources. Currently, the most suitable free platform to achieve this goal is AMReX [7,8], which has demonstrated higher computational efficiency compared to OpenFOAM [9]. Moreover, as of 2023, AMReX is part of the Linux Foundation's High-Performance Software Foundation [10], 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 then describes the implementation of QGD in AMReX by demonstrating the numerical algorithm and the structure of the solver. Section 4 presents the formulation of the problem of a composite vortex-stationary shock interaction. Section 5 shows the computational results in QGD and the OpenFOAM based cross-validation with three solvers. A performance study of the algorithms is presented in section 6. Section 7 concludes the paper with the main conclusions.

2 Mathematical model

The standard form of gas dynamics equations in the form of continuity, momentum, total energy and equation of state equations are used to implement the QGD solver:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{j_m} = 0, \tag{1}$$

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

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

$$p = \rho R T, \tag{4}$$

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

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

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

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

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

$$\tau = \alpha\frac{\Delta_h}{c} + \frac{\mu_T}{p};\ c = \sqrt{\gamma RT} \tag{6}$$

where Δ_h - the local step size of the spatial mesh; c - the local speed of sound; α - the tuning parameter of the algorithm; μ_T - dynamic viscosity ; R - specific gas constant.

$$\mu = \mu_T + \tau p Sc_{QGD} \tag{7}$$

$$k = \left(\frac{\mu_T}{Pr} + \tau\frac{pSc_{QGD}}{Pr_{QGD}}\right)\frac{\gamma}{(\gamma-1)R} \tag{8}$$

where Sc_{QGD} - the tuning parameter of the algorithm.

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

3 Implementing QGD in AMReX

The AMReX software package allows us to use ready-made logic to refine the 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.

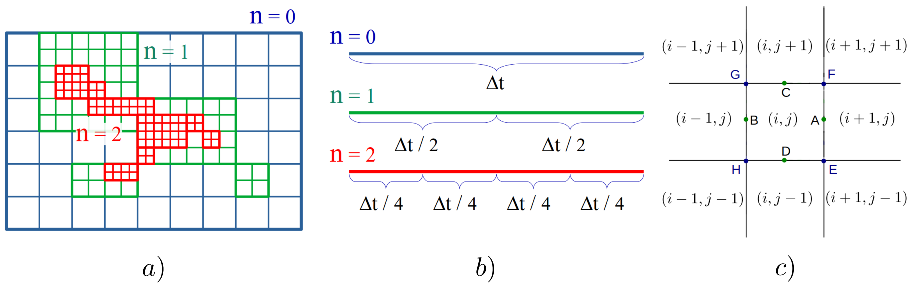


Fig. 1. a) Scheme of adaptive mesh refinement by levels; b) Scheme of adaptive time step refinement at each level of the computational mesh; c) Scheme of the numerical QGD template implemented in AMReX QGD.

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. The discretization of ρ, u_x, v_y, p is carried out by the method of central differences:
$$\rho_A = \frac{\rho_{i,j} + \rho_{i+1,j}}{2},\ \rho_E = \frac{\rho_{i,j} + \rho_{i+1,j} + \rho_{i,j-1} + \rho_{i+1,j-1}}{4} \tag{9}$$
2. Calculated the speed of sound, the parameter τ, the QGD part w
3. Calculated mass flux density $\mathbf{j_m}$
4. The continuity equation is solved (the parameter with the hat is the value of the variable at the new time step):
$$\hat{\rho}_{i,j} = \rho_{i,j} - \frac{\Delta t}{\Delta x}(j_{mA} - j_{mB}) - \frac{\Delta t}{\Delta y}(j_{mC} - j_{mD}) \tag{10}$$
5. The viscosity, the Reynolds viscous stress tensor and its QGD analogue are considered
6. Calculate the momentum equations:
$$\begin{aligned}\hat{u}^x_{i,j} = \rho_{i,j}u^x_{i,j} - \Delta t\left(\frac{j_{mA}u^x_A - j_{mB}u^x_B}{\Delta x} + \frac{j_{mC}u^x_C - j_{mD}u^x_D}{\Delta y} + \frac{p_A - p_B}{\Delta x}\right)\\ +\Delta t\left(\frac{\sigma^{xx}_A - \sigma^{xx}_B}{dx} + \frac{\sigma^{yx}_C - \sigma^{yx}_D}{\Delta y}\right)\end{aligned} \tag{11}$$
$$\begin{aligned}\hat{u}^y_{i,j} = \rho_{i,j}u^y_{i,j} - \Delta t\left(\frac{j_{mA}u^y_A - j_{mB}u^y_B}{\Delta x} + \frac{j_{mC}u^y_C - j_{mD}u^y_D}{\Delta y} + \frac{p_C - p_D}{\Delta y}\right)\\ +\Delta t\left(\frac{\sigma^{yy}_C - \sigma^{yy}_D}{\Delta y} + \frac{\sigma^{xy}_A - \sigma^{xy}_B}{\Delta x}\right)\end{aligned} \tag{12}$$
7. Calculate the temperature, heat transfer coefficient, specific internal energy, heat fluxes and enthalpy
$$H_A = \frac{(u^x_A)^2 + (u^y_A)^2}{2} + \gamma\frac{p_A}{\rho_A(\gamma - 1)}$$
8. Calculate the energy balance equation:
$$\begin{aligned}\hat{E}_{i,j} = E_{i,j} - \Delta t\left(\frac{j_{mA}H_A - j_{mB}H_B}{\Delta x} + \frac{j_{mC}H_C - j_{mD}H_D}{\Delta y} + \frac{q_A - q_B}{\Delta x}\right.\\ \left. + \frac{q_C - q_D}{\Delta y}\right) + \Delta t\left(\frac{\sigma^{xx}_A u^x_A - \sigma^{xx}_B u^x_B}{\Delta x} + \frac{\sigma^{yy}_C u^y_C - \sigma^{yy}_D u^y_D}{\Delta y}\right.\\ \left. + \frac{\sigma^{xy}_A u^y_A - \sigma^{xy}_B u^y_B}{\Delta x} + \frac{\sigma^{yx}_C u^x_C - \sigma^{yx}_D u^x_D}{\Delta y}\right)\end{aligned} \tag{13}$$
9. Pressure is being corrected:
$$\hat{p}_{i,j} = (\gamma - 1)\left(\hat{E}_{i,j} - \hat{\rho}_{i,j}\frac{(\hat{u}^x_{i,j})^2 + (\hat{u}^y_{i,j})^2}{2}\right) \tag{14}$$

3.2 Solver structure

Figure 2 shows the structure of the developed numerical algorithm based on QGD and AMReX.

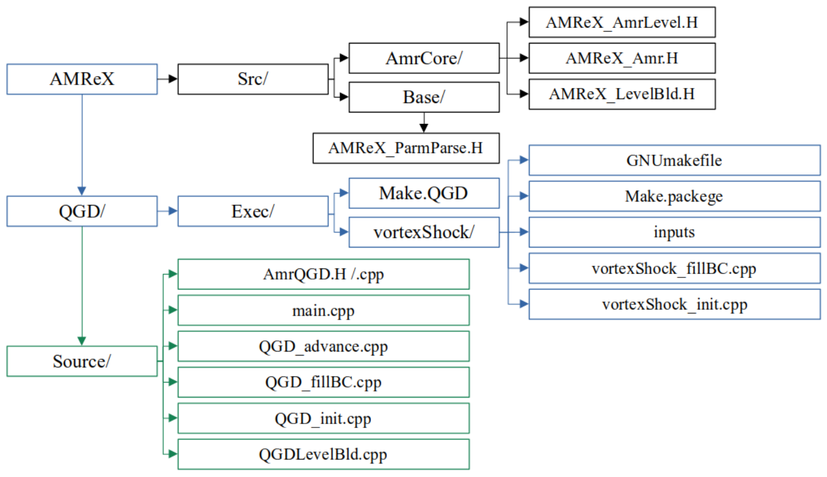


Fig. 2. Structure of the QGD solver in AMReX. Black colour - AMReX kernel; Blue - location of the problem with its initial and boundary conditions; Green - location of the solver.

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

- **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 the file
- **main.cpp** - numerical algorithm
- **QGD_advance.cpp** - quasi-gasdynamic equations are solved in the file
- **QGD_fillBC.cpp** - the file describes the boundary conditions
- **QGD_init.cpp** - initial conditions are set in the file.
- **QGDLevelBld.cpp** - description of mesh refining

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

- **inputs** - file where the calculation parameters are set
- **vortexShock_fillBC.cpp** - 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.

4 Problem statement

To demonstrate the features and advantages of the developed QGD AMReX solver, it was decided to choose a problem with complex unsteady flow, a good example of which 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 have been carried out for more than 70 years and include experimental [11,12] and theoretical studies [13,14]. In recent years, this problem has been widely used to demonstrate the correctness of high-order methods [15,16,17]. In particular, this problem has also been used to evaluate algorithms based on QGD equations [18]. The reference values are taken from [19], in where they have been obtained by a high-order method.

The geometry of the calculation area is a rectangle of size 2×1 m (Fig. 3a). Gas parameters: $\gamma = 1.4$, $R = 1$, $C_p = 3.5$. 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 ($x < 0.5$):

$$(\rho, u_x, v_y, p, T)_{\text{left}} = (1,\ M_s\sqrt{\gamma},\ 0,\ 1,\ p/(\rho R)) \tag{15}$$

Stationary shock conditions:

$$\frac{\rho_{\text{left}}}{\rho_{\text{right}}} = \frac{u_{\text{right}}}{u_{\text{left}}} = \frac{2 + (\gamma - 1)M_s^2}{(\gamma + 1)M_s^2},\ \frac{p_{\text{left}}}{p_{\text{right}}} = 1 + \frac{2\gamma}{\gamma + 1}(M_s^2 - 1),\ v_{\text{right}} = 0 \tag{16}$$

The domain's initial conditions take values:

$$(\rho, u_x, v_y, p, T) = \begin{cases} (1,\ 1.77482,\ 0,\ 1,\ 1) & x < 0.5 \\ (1.862,\ 0.953146,\ 0,\ 2.45833,\ 1.32022) & x \geq 0.5 \end{cases} \tag{17}$$

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 v_θ:

$$v_\theta(r) = \begin{cases} v_m \frac{r}{a} & r \leq a \\ v_m \frac{a}{a^2 - b^2}\left(r - \frac{b^2}{r}\right) & a < r \leq b \\ 0 & r > b \end{cases} \tag{18}$$

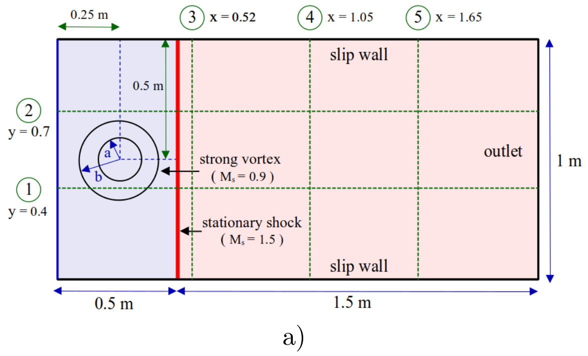
where $r = \sqrt{(x - 0.25)^2 + (y - 0.5)^2}$ - radius from the vortex centre. Then in the projection on the OX and OY axes: $u_x(r) = u_{\text{left}} - v_\theta(r)sin(\theta)$, $v_y(r) = v_{\text{left}} + v_\theta(r)cos(\theta)$

Condition on the temperature inside the vortex:

$$T(r) = \begin{cases} T(b) - \frac{\gamma-1}{R\gamma}\frac{v_m^2}{a^2}\left(\frac{a^2-r^2}{2}\right) - \\ -\frac{\gamma-1}{R\gamma}v_m^2\frac{a^2}{\sqrt{a^2-b^2}}\left(\frac{b^2-a^2}{2} - 2b^2\ln\frac{b}{a} - \frac{b^4}{2}\left(\frac{1}{b^2}-\frac{1}{a^2}\right)\right) & r \le a \\ T(b) - \frac{\gamma-1}{R\gamma}v_m^2\frac{a^2}{\sqrt{a^2-b^2}}\left(\frac{b^2-r^2}{2} - 2b^2\ln\frac{b}{r} - \frac{b^4}{2}\left(\frac{1}{b^2}-\frac{1}{r^2}\right)\right) & a < r \le b \\ 0 & r > b \end{cases} \tag{19}$$

Condition on the density and pressure inside the vortex:

$$\rho(r) = \rho_{\text{left}}\left(\frac{T(r)}{T_{\text{left}}}\right)^{\frac{1}{\gamma-1}}, \ p(r) = p_{\text{left}}\left(\frac{T(r)}{T_{\text{left}}}\right)^{\frac{\gamma}{\gamma-1}} \tag{20}$$



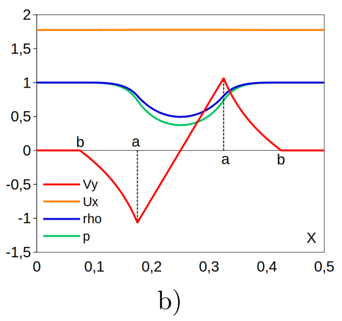


Fig. 3. a) Geometry of the computational domain; b) Initial conditions on the line $Y = 0.5$ m in the vortex location region ($a = 0.075$ m, $b = 0.175$ m).

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 in the range $x < 0.5$
- Right (outlet): Smooth boundary conditions (zero gradient)

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

The density values on five lines are also compared (Fig. 3a), the reference values are taken from [19], obtained in this work by the higher order method on a mash of 12800×6400.

5 Results

The vortex-shock wave interaction problem was numerically modelled on $\frac{1}{400}, \frac{1}{800}, \frac{1}{1600}$ meshes along the OY axis and different Courant numbers. The following software packages and solvers were used for further cross-validation:

- rhoCentralFoam solver for compressible flows based on the central Kurganov-Tadmor schemes [20], implemented in OpenFOAM. Two methods upwind [21] (1st order) and VanLeer [22] (2nd order) are used
- QGDFoam [4] solver for flows over a wide range of Mach numbers, implemented in OpenFOAM and based on the QGD equations [2]
- QGD a new QGD-based equation solver implemented on AMReX base

5.1 OpenFOAM (rhoCentralFoam and QGDFoam)

Figure 4 shows the fields of numerical Schlieren values for OpenFOAM based solvers as a function of mesh cell size. For the rhoCentralFoam solver we used upwind [21] (1st order approximation), VanLeer [22] (2nd order approximation) schemes. For QGDFoam we used the tuning parameters of the QGD algorithm $\alpha = 0.1$, $Sc_{QGD} = 0.1$, these parameters are defined in the paper [18] where this problem is considered.

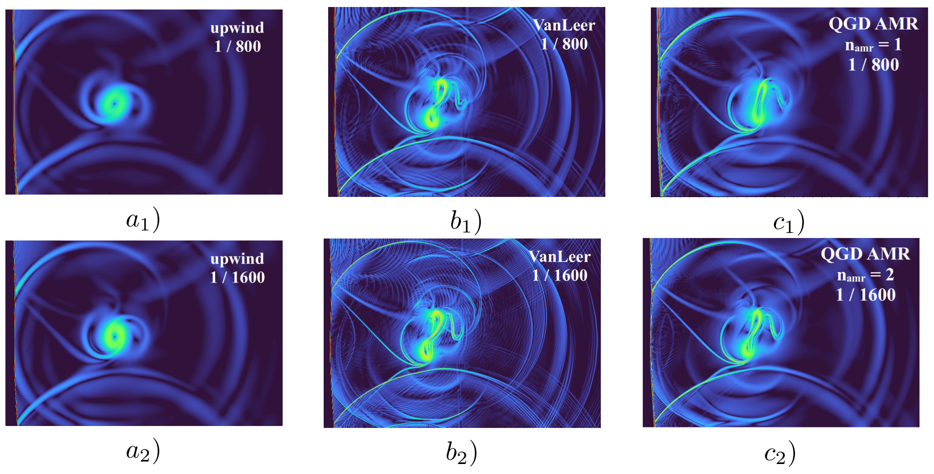


Fig. 4. Numerical Schlieren with constant $\Delta t = 10^{-5}$ c.

It can be seen that the upwind scheme does not give an acceptable solution even on a $1/1600$ mesh, and the vanLeer and QGD schemes resolve the flow most correctly. However, when investigating different values of the time step (Fig. 5), it is found that the vanLeer 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 vanLeer scheme, the Courant number $Co = 0.01$ is required.

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 vanLeer and QGD schemes correctly resolve the vortex structure and its features.

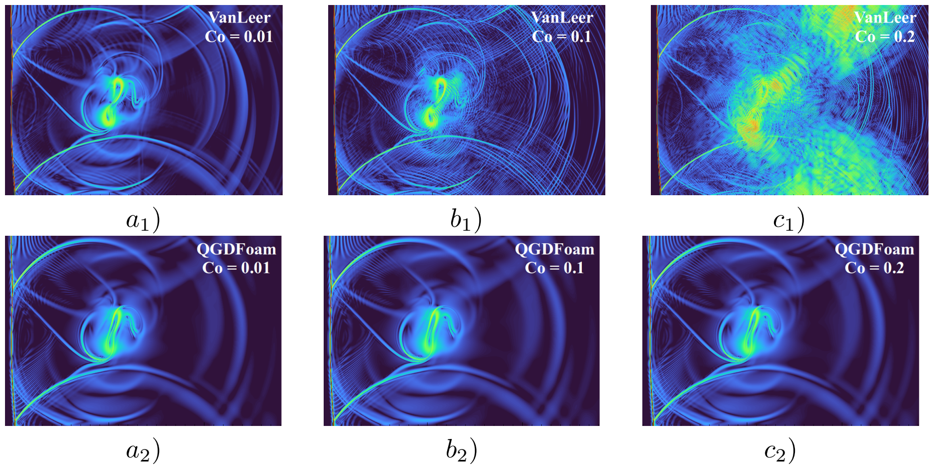


Fig. 5. Numerical Schlieren on a 1/800 mesh. (a_1, b_1, c_1) - VanLeer; (a_2, b_2, c_2) - QGDFoam. (a_1, a_2)-$Co = 0.01$ ($\Delta t = 0.5 \cdot 10^{-5}$s); (b_1, b_2) - $Co = 0.1$; (c_1, c_2) - $Co = 0.2$.

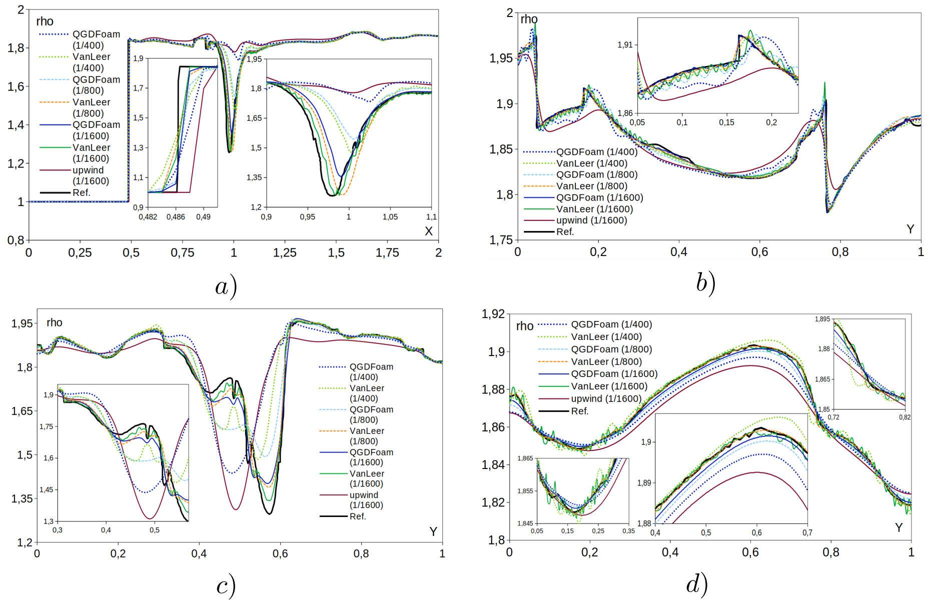


Fig. 6. Comparison of the density field plots on the lines (Fig.3. a) Line 1; b) Line 3; c) Line 4; d) Line 5.

5.2 QGD AMReX

The QGD algorithm has two tuning parameters α and Sc_{QGD} in the paper [18] it is noted that the best choice for this problem is $\alpha = 0.1$ and $Sc_{QGD} = 0.1$, however these recommendations are given for numerical implementation based on the OpenFOAM package, so additional study is required for the newly developed solver. Figure 7 a shows 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 the oscillations, a local variation of the Sc_{QGD} number (variable varSc) is introduced, which is equal to 1 on the stationary compaction shock when variable varSc = true. Figure 7 b shows the effect of the additional viscosity on the shock. It can be seen that this approach has significantly reduced the numerical oscillations after the vortex passes the compaction 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 more likely that the numerical implementation of the QGD algorithm includes the use of limiters, which leads to less oscillations, whereas in AMReX the numerical implementation does not include limiters.

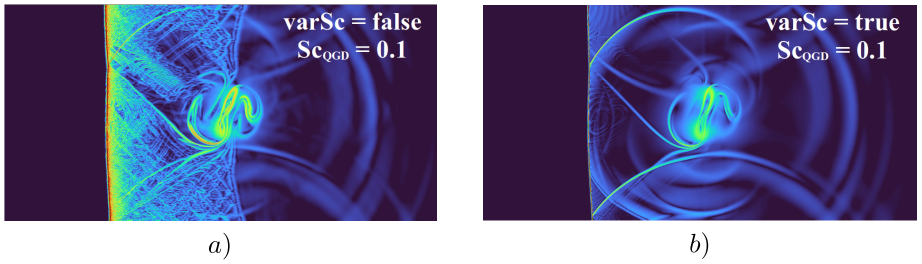


Fig. 7. Numerical Schlieren in QGD AMReX on a 1/1600 mesh with $\alpha = 0.1$ and $Sc_{QGD} = 0.1$. a) Dynamic Schmidt - **off**; b) Dynamic Schmidt - **on**,

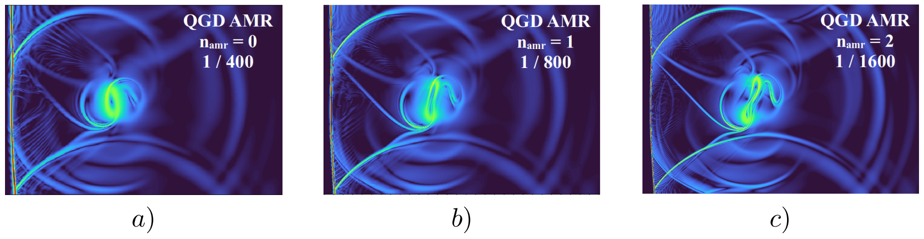


Fig. 8. Numerical Schlieren in QGD AMReX with $\alpha = 0.1$, $Sc_{QGD} = 0.1$: (a) 1/400; (b) 1/800; (c) 1/1600.

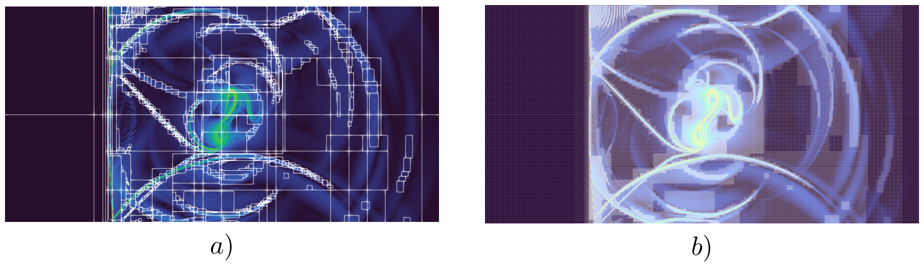


Fig. 9. Example of adaptive mesh refinement: a) by blocks; b) by cells.

Figure 8 shows the influence of the adaptive mesh refinement function (number of levels n_{amr}) on the results obtained. The adaptation is performed on the density gradient, i.e. if the gradient (the difference 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 mesh cell. Figure 8a shows the flow structure without splitting, on a mesh of size 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 compaction 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).

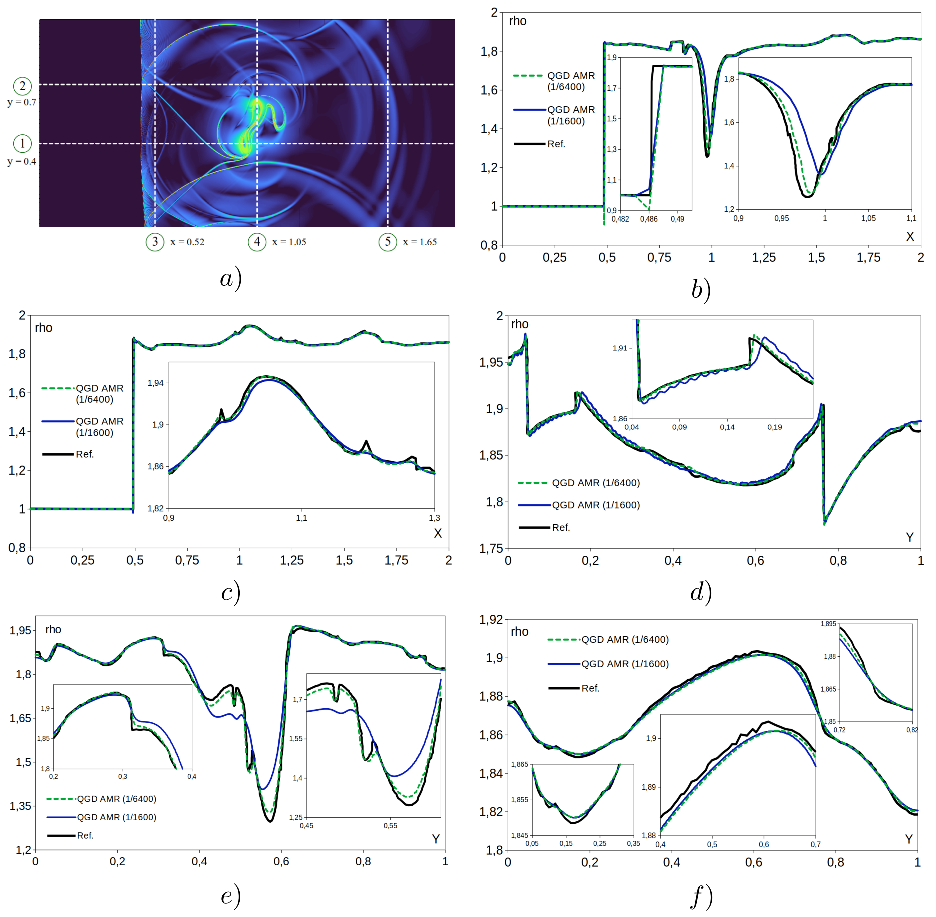


Fig. 10. Comparison of density values on five reference lines: b) Line 1; c) Line 2; d) Line 3; e) Line 3; d) Line 4; f) Line 5;

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 carried out (Fig. 10). The use of AMR technology allowed us to perform the calculation for such a mesh refinement within 6 hours on 24 cores. On the other hand, such a calculation on a stationary grid in the OpenFOAM package would have been computationally expensive.

6 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). It can be seen that on 12 cores, QGD AMReX ($n_{amr} = 2$) computes 9 times faster than upwind, 15 times faster than vanLeer, 32 times faster than QGDFoam and 4.5 times faster than QGD AMReX 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 QGD AMReX and the fact that the algorithm reaches its lower bound on the number of cells per core.

CPU	Cell per CPU	rhoCentralFoam		QGDFoam	QGD AMReX	
		upwind	vanLeer		AMR off	AMR 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 1. The calculation time on the 1/1600 s mesh is $\Delta t = 10^{-5}$ s and $t_{end} = 0.01$ s.

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

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

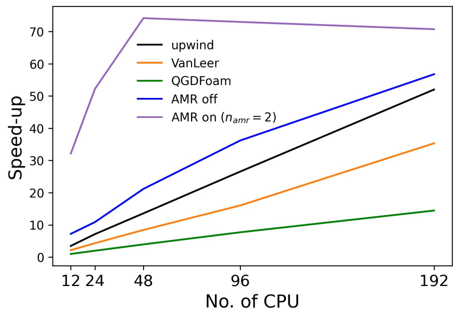


Fig. 11. Speedup of various algorithms.

7 Conclusion and discussion

We have developed a new two-dimensional QGD solver AMReX based on quasi-gas-dynamic equations with the possibility of adaptive mesh refinement and parallelization of computations to graphics kernels and by this article we present and describe it.

Cross-validation of the OpenFOAM and AMReX software packages, as well as the different algorithms upwind, VanLeer and QGD, was made, with the latter one being implemented in two software packages at the same time. It is found that upwind algorithm does not give an acceptable solution on a reasonable grid and time step. The VanLeer algorithm gives an acceptable solution on a grid of 1/1600 and a time step of $\Delta t = 10^{-5}$ s, that corresponds to a Courant number of $Co = 0.03$. A comparable result is given by the QGD algorithm implemented in QGDFoam and AMReX QGD, but 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 OU and with a time step $\Delta t = 10^{-5}$ s, it is found that on 12 cores AMReX QGD with two levels of adaptation is 15 times faster than VanLeer, 32 times faster than QGDFoam and 4.5 times faster than AMReX QGD on a stationary grid of 1/1600. However, the higher the number of cores, the worse is parallelization efficiency of QGD AMReX.

In the future we plan to extend the presented QGD AMReX solver onto three-dimensional flows, optimize the parallelization of the QGD algorithm and test it on a wide range of validation problems.

Acknowledgments. 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/fld.2726
4. QGDSolvers, https://github.com/unicfdlab/QGDsolver Last accessed 29 Feb 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. 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
8. 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
9. 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
10. TheLinuxFoundation, https://www.linuxfoundation.org/press. Last accessed 29 Feb 2024
11. 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).
12. 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
13. Ribner H. S. "The sound generated by interaction of a single vortex with a shock wave." University of Toronto. (1959).
14. 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
15. 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
16. 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
17. 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

18. 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
19. 5th International workshop on hight-order CFD methods, https://how5.cenaero.be/ Last accessed 29 Feb 2024
20. 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
21. 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
22. 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