Scientific Visualization, 2024, volume 16, number 4, pages 25 - 36, DOI: 10.26583/sv.16.4.03

Integrating Scientific Visualization in the Assessment of Openfoam Solvers for the Flow Around a Spherically Blunted Cone

A.E. Bondarev[1,A], A.E. Kuvshinnikov[2,A]

Keldysh Institute of Applied Mathematics RAS

[1] ORCID: 0000-0003-3681-5212, bond@keldysh.ru
[2] ORCID: 0000-0003-1667-6307, kuvsh90@yandex.ru

Abstract

This study presents a comprehensive comparative analysis of four OpenFOAM solvers for simulating supersonic flow around a Spherically Blunted Cone. Utilizing the generalized computational experiment technique, we evaluated solver performance across varying Mach numbers and cone half-angles. The research aimed to provide clear recommendations for solver selection in high-speed flow simulations. Results show that rhoCentralFoam exhibited the lowest deviation from the exact solution, followed closely by pisoCentralFoam. The sonicFoam solver demonstrated limitations at higher Mach numbers. The QGDFoam solver, while showing promise, requires further optimization of its smoothing parameter for improved accuracy. This study offers valuable insights for OpenFOAM users, enabling them to make informed decisions when choosing solvers for compressible gas dynamics problems.

Keywords: comparative accuracy assessment, computational fluid dynamics, supersonic flow, spherically blunted cone, OpenFOAM.

1. Introduction

Researchers working with the OpenFOAM software package [1] are concerned with comparing accuracy and efficiency across various solvers. Using OpenFOAM for solving computational fluid dynamics (CFD) problems involves selecting an appropriate solver that meets the expectations of accuracy, efficiency, and stability. To achieve this, it is necessary to analyse the performance of the solver based on some reference solutions. However, few studies are offering a detailed comparison of OpenFOAM solvers based on these characteristics. Some examples of such comparisons can be found in [2] and [3], but they do not provide clear recommendations for solver selection. A more detailed attempt at comparing solvers was made in a series of works where the reference problems included supersonic flow around a cone at an angle of attack [4, 5], shock wave modelling [6], and the formation of a two-dimensional rarefaction wave [7]. A review of these studies can also be found in [8]. The method proposed in these works gained popularity in [9], [10]. By employing the generalised computational experiment technique [11-14], results were obtained that allow users to navigate the wide range of numerical methods developed and select the most accurate and efficient ones for their calculations. The generalised computational experiment involves dividing the governing parameters of a problem within a specific range, followed by a parametric study and visualisation of multidimensional results [15].

OpenFOAM (Open Source Field Operation And Manipulation CFD Toolbox) is a popular software package for computational fluid dynamics (CFD), widely used for modelling liquid and gas flows, heat transfer, and a variety of other tasks related to physical processes. OpenFOAM is an open-source package, meaning that it is available for free, and its source code is

open. This allows users to modify the code for their specific tasks and add custom equations, turbulence models, or other physical processes. For example, if you have a specific task, such as modelling gas-dynamic shock waves with chemical reactions, you can independently create a solver that takes these features into account. This flexibility distinguishes OpenFOAM from commercial software, where many features are closed to modification. OpenFOAM also supports parallel computing, enabling the solution of large-scale problems with complex physics. This is critical when calculations need to be performed on a grid consisting of millions of cells. In OpenFOAM, the computational domain can be easily divided among multiple processors, thereby accelerating the calculations. The comparison included four solvers: two standard solvers – rhoCentralFoam and sonicFoam and two custom solvers – pisoCentralFoam [16] and QGDFoam [17]. The latter two solvers were developed by research teams from the Institute for System Programming of the Russian Academy of Sciences and the Keldysh Institute of Applied Mathematics of the Russian Academy of Sciences.

The rhoCentralFoam solver utilises a central-upwind scheme of the Godunov type, originally proposed by Kurganov and Tadmor (KT) [18], later modified in the work by Kurganov, Noelle, and Petrova (KNP) [19]. The Kurganov-Tadmor scheme bypasses the Riemann problem by using an approximation of local wave propagation speeds and a central approach. Instead of solving the Riemann problem to determine the flux at the cell interface, the KT scheme employs an estimate of the maximum wave propagation speed (typically determined through the maximum characteristic speed of the equations) on both sides of the cell. This speed corresponds to the rate at which information propagates through the cell boundary. For example, in the case of the Euler equations, this might be the speed of sound plus the flow velocity, as this is the maximum speed of information propagation in a compressible gas. This approach helps to avoid many difficulties associated with computing shock waves and steep gradients in density or pressure fields, which can lead to oscillations in numerical solutions. The KNP scheme was implemented for OpenFOAM by Greenshields [20].

The sonicFoam solver uses the PIMPLE algorithm [21] to link pressure and velocity. This algorithm combines elements of the PISO and SIMPLE methods. At its core, the PIMPLE algorithm features an iterative structure designed to refine the velocity and pressure fields within the fluid domain. The algorithm begins with an initial assumption about the velocity field, which is often derived from previous time steps or estimated using simpler approaches. This initial assumption is crucial as it lays the foundation for the subsequent iterative process. In the first step, the algorithm predicts the velocity field based on the existing pressure distribution, applying an explicit scheme that facilitates direct calculation of the velocity. After obtaining the initial velocity forecast, the algorithm then updates the pressure field. This is where the strength of the PISO method comes into play. At this stage, the algorithm calculates a pressure correction that ensures compliance with the conservation of mass. The pressure correction is essential as it helps eliminate discrepancies between the predicted and actual behaviour of the gas, especially in regions where flow velocities can change rapidly or where complex boundary conditions exist. After correcting the pressure, the velocity field is updated once again, taking into account the new pressure information. This iterative process is repeated several times, typically within a single time step, allowing for continuous refinement of the velocity and pressure fields. Each iteration helps improve the accuracy of the solution, leading to greater stability.

When using explicit methods based on solving the Riemann problem or flux vector splitting, the following issues arise: at low Mach numbers, the solution either loses stability, or an extremely small time step is required, which increases computational costs and makes such methods inefficient for the considered problems. On the other hand, implicit splitting methods such as PISO and SIMPLE can cause non-physical oscillations in the numerical solution and loss of stability at high Mach numbers. These phenomena significantly limit the effectiveness of these methods in problems where compressibility plays a major role, making them unsuitable for studying high-speed flows or acoustic phenomena. The solution could be to use a hybrid approach, where PISO/SIMPLE algorithms for the implicit integration of mass,

momentum, and energy conservation equations are combined with non-oscillatory methods for discretizing convective terms. The authors of [16] selected the Kurganov-Tadmor scheme as the non-oscillatory method; it is already implemented as an explicit scheme in OpenFOAM, has undergone successful testing, and is simple enough for integration into a hybrid approach. One of the key advantages of this scheme is the independence of flux approximation for physical quantities from the system of equations' characteristics, eliminating the need to find Riemann invariants and decompose solutions by characteristics. The main idea of the hybrid method is to introduce a switching function that translates the flux approximation from the KT scheme to the form used in the PIMPLE method, depending on the proximity to subsonic speeds [22]. All these hybrid solvers are not included in OpenFOAM, but are available in the authors' public repository [23].

The quasi-gas-dynamic (QGD) equations system was developed by a team led by B.N. Chetverushkin [24]. It is based on the idea that gas can be considered not as a continuous medium, but as a set of particles moving and interacting with each other at the microscopic level. The QGD model serves as an intermediate approach between classical hydrodynamics and more complex models that account for microscopic effects. The introduction of additional terms into the quasi-gas-dynamic equations aims to capture small-scale effects that cannot be directly resolved within the framework of traditional Navier-Stokes equations [25]. These additional terms are often referred to as dissipative terms, and they are introduced through various averaging and approximation techniques that reflect physical processes. The inclusion of these terms allows for a more accurate description of complex phenomena and interactions in gases, especially in conditions where non-equilibrium processes and energy transformation are significant. Based on this system, the QGDFoam solver was developed. The parameter τ plays a key role in accounting for small-scale effects and determines the time scale for dissipative processes. The controlled parameter α (associated with τ), in dissipative terms, provides this solver with adjustable numerical viscosity, which helps to reduce unwanted oscillations at discontinuities.

2. Problem Statement

In this study, a two-dimensional inviscid problem of steady flow formation around a blunt circular cone with a half-angle of β is used to compare solvers. The flow is generated by a supersonic gas flow with a Mach number M at zero angle of attack. The defining parameters of the problem, in terms of the generalised computational experiment, are the Mach number M and the half-angle of the cone β . The ranges for the varying parameters and their increments were chosen the following way: the Mach number M is set to 2, 4, and 6, and the half-angle of the cone β is set to 5°, 10°, 15°, and 25°. The overall flow scheme is presented in Figure 1. For all selected parameters, there exists a tabulated solution for the problem [26]. The Euler equations system, closed by the ideal gas equation of state, is taken for the calculations.

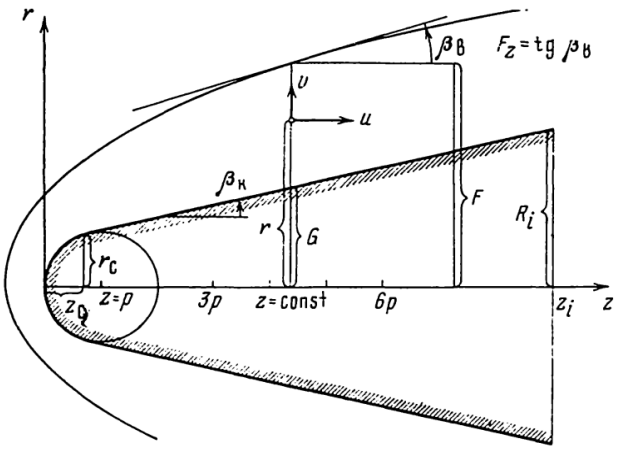

Figure 1. Flow Diagram

3. Managing Calculations

The computational domain, as shown in Figure 2, is divided into cells. To work with the OpenFOAM package, it is necessary to define the boundary and initial conditions. At the inlet boundary, designated as "inlet," the characteristics of the undisturbed flow are specified: the pressure P is 101,325 Pa, the temperature T is 300 K, and the x-component of velocity Ux varies from 694.5 m/s (2 Mach) to 2083.5 m/s (6 Mach), while the y-component of velocity Uy is 0 m/s. At the outlet and the top, conditions are set so that the derivatives of gas dynamic functions normal to the boundary are equal to zero, defined in OpenFOAM as «zeroGradient». At the cone boundary, a zero gradient condition is applied for pressure and temperature, while a «slip» condition is used for velocity, which corresponds to the non-penetrating condition in the Euler equations. To model the axisymmetric geometry in OpenFOAM, this special «wedge» condition is applied at the front and back boundaries. The axis in OpenFOAM uses a specific boundary condition «empty», which is employed when calculations in that direction are not conducted.

The computational domain (Figure 2) is divided into cells. The OpenFOAM package requires the definition of the boundary and initial conditions for the solution. At the inlet boundary, the parameters of the undisturbed incoming flow are specified (pressure P = 101325 Pa, temperature T = 300 K, the x-component of velocity varies from 694.5 m/s (2 Mach) to 2083.5 m/s (6 Mach), and the y-component of velocity is 0 m/s). At the outlet and the top, boundary conditions are set such that the derivatives of gas dynamic functions normal to the boundary are equal to zero, defined in OpenFOAM as «zeroGradient». At the cone, a zero-gradient condition is also applied for pressure and temperature, while a «slip» condition is used for velocity, corresponding to the no-penetration condition in the Euler equations. For modelling the axisymmetric geometry, a special «wedge» condition is applied at the front and back boundaries. This special boundary condition called «empty» is applied at the axis. This condition is used in cases where calculations are not performed in that direction.

The scheme of the computational domain for the cone with a half-angle $\beta = 15^\circ$ is presented in Figure 2. It is worth noting that in the displayed image, the grid appears larger than in the actual calculations for clarity.

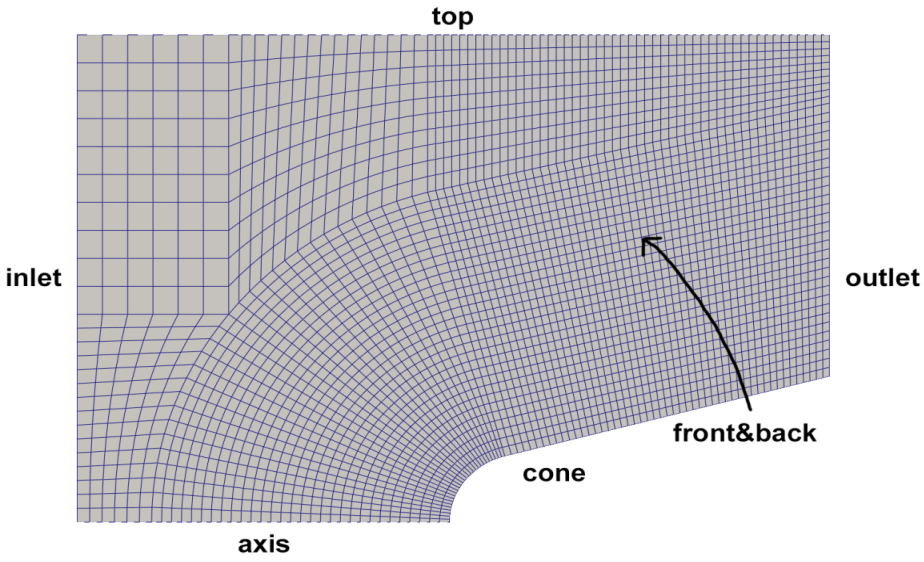

Figure 2. Computational Domain Scheme

The number of grid cells is 35,950. The initial conditions correspond to the boundary conditions at the inlet face, meaning the parameters of the incoming flow serve as the initial conditions. In the QGDFoam solver, a smoothing coefficient of $\alpha = 0.1$ was applied across the en-

tire computational domain. Additionally, the values for molar mass M = 28.96 and specific heat capacity at constant pressure C_p = 1004 were set.

Unlike many other software packages, OpenFOAM manages simulations through text files. This provides flexibility, as it is easy to automate task launches, modify simulation parameters, and process results. The standardisation of calculations plays a crucial role in organising the comparison of solvers, as it ensures uniform conditions for assessing their performance and accuracy. With standardised methodologies, grids, boundary conditions, and physical models, the results become comparable and reliable. This allows researchers to eliminate the influence of external factors and focus on the characteristics of each specific solver. Furthermore, standardisation helps in better understanding the strengths and weaknesses of each solver, which can be useful for selecting the optimal tool based on the specifics of the task. In the OpenFOAM package, we used the same parameters for the configuration files fvSchemes and fvSolution as in [5].

4. Experimental Results

The simulations performed with all solvers resulted in the well-known qualitative flow pattern for the considered problem. An example is shown in Figure 3, depicting the pressure distribution in the computational domain. The presented pressure distribution was obtained using the rhoCentralFoam solver. No solution breakdown was observed for any of the solvers, indicating the high stabilising properties of all solvers involved in the study.

Next, the deviation is evaluated from the exact solution for the entire computational domain using the L_2 norm. To do this, we define the relative error Err for the L_2 norm as follows:

$$L_2: \; Err = \sqrt{\sum_m |y_m - y_m^{exact}|^2 V_m} \Big/ \sqrt{\sum_m |y_m^{exact}|^2 V_m}$$

Here, y_m^{exact} represents the pressure p, density ρ, x and y components of velocity (Ux and Uy). V_m^{exact} is the cell volume. The values of y_m^{exact} were obtained by interpolating the tabulated solution of the problem. The solvers involved in the comparative accuracy analysis were sonicFoam, QGDFoam, rhoCentralFoam, and pisoCentralFoam. For convenience, the solvers are denoted by abbreviations in the tables: rCF (*rhoCentralFoam*), pCF (*pisoCentralFoam*), sF (*sonicFoam*), QGDF (*QGDFoam*). The deviation values from the exact solution for all values over the entire computational domain are provided in Tables 1–3. The smallest values in each row are highlighted in bold.

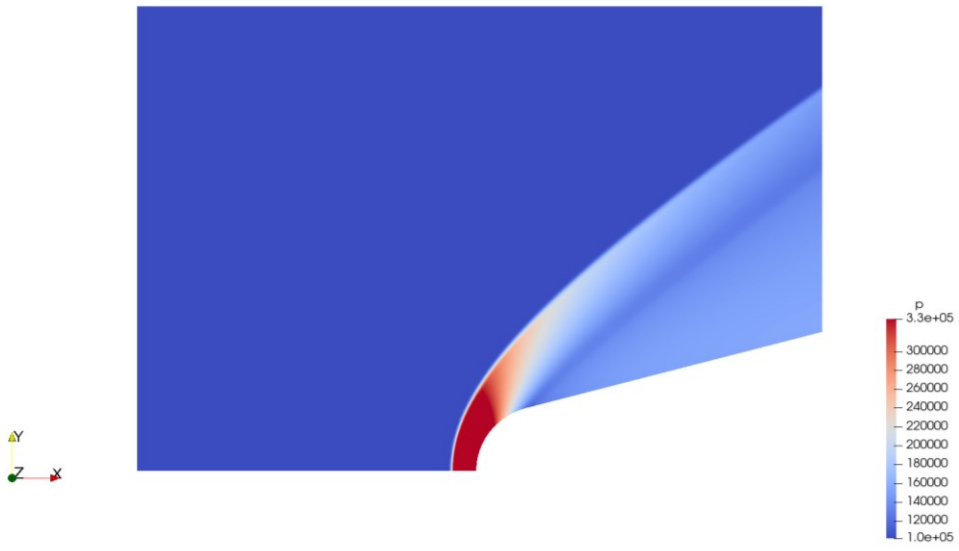

Figure 3. Field steady-state flow pressure for the rhoCentralFoam Solver

TABLE 1 ERRORS FOR M = 2

Value	Half-cone angle	rCF	pCF	sF	QGDF
p	5	**0.017028**	0.022472	0.031921	0.025685
	10	**0.027280**	0.033699	0.055463	0.042351
	15	**0.034960**	0.040031	0.080514	0.046906
	25	**0.036960**	0.043399	0.085460	0.049201
ρ	5	**0.013049**	0.016927	0.026125	0.019875
	10	**0.021151**	0.025949	0.046866	0.033427
	15	**0.027018**	0.030899	0.069235	0.037590
	25	**0.028358**	0.033815	0.074514	0.040236
Ux	5	0.003332	**0.002882**	0.003195	0.003090
	10	0.005847	**0.005424**	0.006937	0.005807
	15	0.009694	**0.008440**	0.012859	0.009248
	25	0.012425	**0.012269**	0.016679	0.012820
Uy	5	**0.049208**	0.058461	0.073065	0.061710
	10	**0.046843**	0.054963	0.073083	0.060938
	15	**0.043771**	0.048582	0.071450	0.048474
	25	**0.036296**	0.044299	0.057872	0.040218

TABLE 2 ERRORS FOR M = 4

Value	Half-cone angle	rCF	pCF	sF	QGDF
p	5	**0.031943**	0.040018	0.067521	0.051875
	10	**0.045202**	0.053764	0.125038	0.062389
	15	**0.051245**	0.060396	0.154412	0.074341
	25	**0.051458**	0.061273	0.161013	0.077917
ρ	5	**0.024840**	0.030584	0.056221	0.040623
	10	**0.035139**	0.041356	0.106082	0.049628
	15	**0.039569**	0.046781	0.134346	0.059978
	25	**0.039651**	0.047775	0.142780	0.063810
Ux	5	0.005091	**0.003401**	0.004375	0.003689
	10	0.008348	**0.006382**	0.011029	0.006035
	15	0.011934	**0.009961**	0.019059	0.009639
	25	0.015016	**0.014323**	0.024822	0.013115
Uy	5	**0.070407**	0.076146	0.105546	0.087226
	10	**0.060978**	0.069945	0.115991	0.065847
	15	**0.053252**	0.063890	0.102818	0.059047
	25	**0.043076**	0.058077	0.081359	0.050234

TABLE 3 ERRORS FOR M = 6

Value	Half-cone angle	rCF	pCF	sF	QGDF
p	5	**0.057095**	0.064405	0.174251	0.083892
	10	**0.065626**	0.067459	0.183761	0.095616
	15	**0.068654**	0.071534	0.194212	0.099098
	25	**0.071054**	0.072618	0.204263	0.100022
ρ	5	**0.039118**	0.044436	0.141836	0.062416
	10	**0.050310**	0.052647	0.158682	0.076522
	15	**0.053411**	0.056018	0.168282	0.080265
	25	**0.055427**	0.057246	0.182445	0.082623
Ux	5	0.006597	**0.003779**	0.007974	0.004184
	10	0.010538	**0.006571**	0.012771	0.007293
	15	0.014176	**0.009545**	0.017492	0.010283
	25	0.018507	**0.012650**	0.024849	0.013967
Uy	5	**0.090816**	0.092901	0.158190	0.101646
	10	**0.075070**	0.077949	0.124047	0.082505
	15	**0.064468**	0.070916	0.097257	0.065471
	25	**0.056260**	0.068021	0.085289	0.055883

The error surface visualization shown in Figures 4–7.

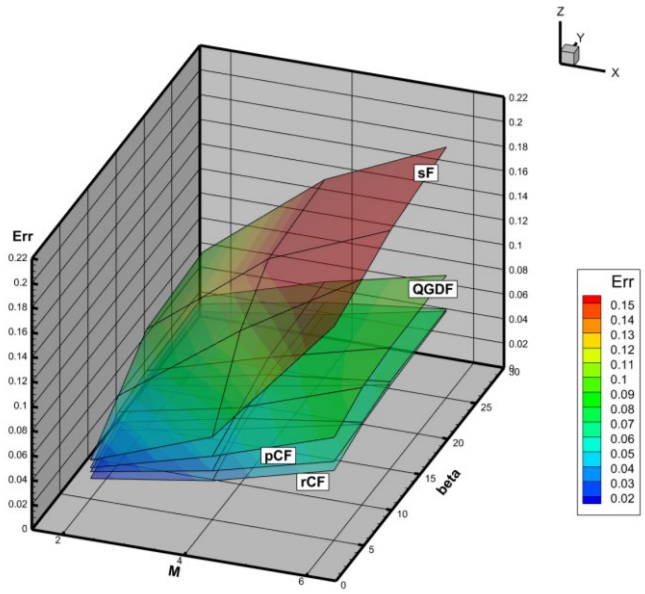

Figure 4. Variation of the deviation from the exact solution for pressure depending on the Mach number and cone half-angle for all solvers in L_2 norm

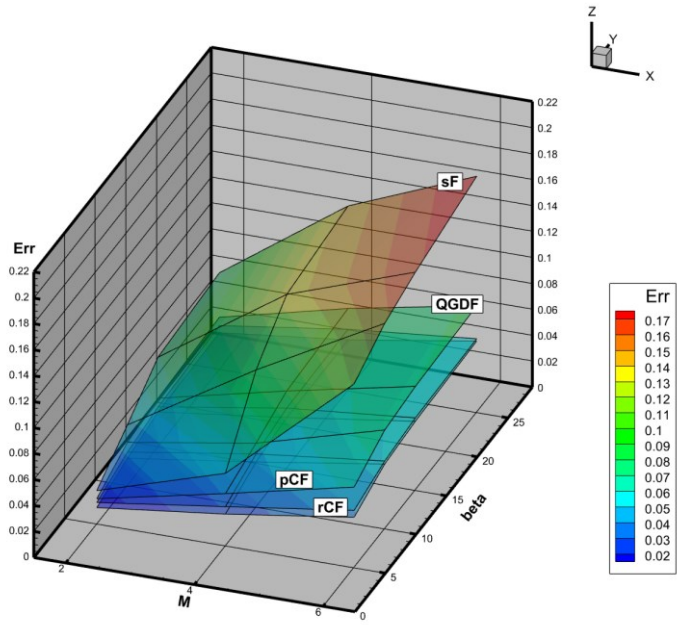

Figure 5. Variation of the deviation from the exact solution for density depending on the Mach number and cone half-angle for all solvers in L_2 norm

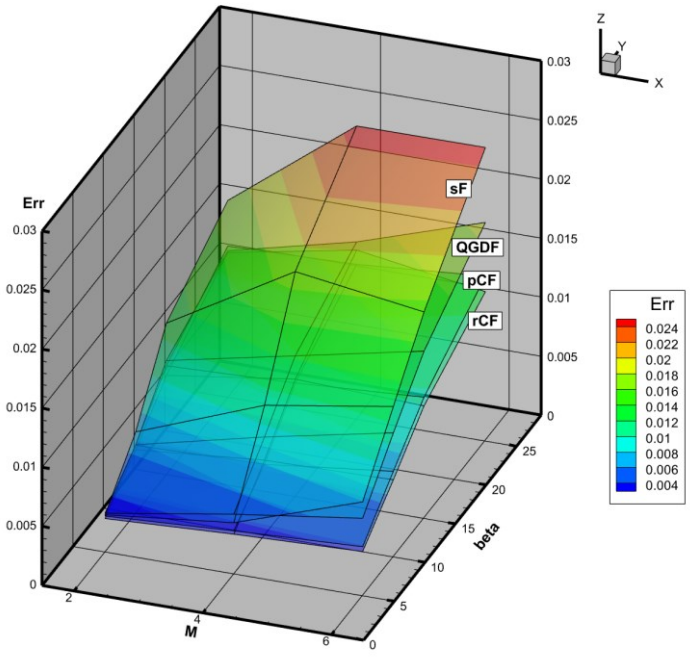

Figure 6. Variation of the deviation from the exact solution for velocity x-component depending on the Mach number and cone half-angle for all solvers in L_2 norm

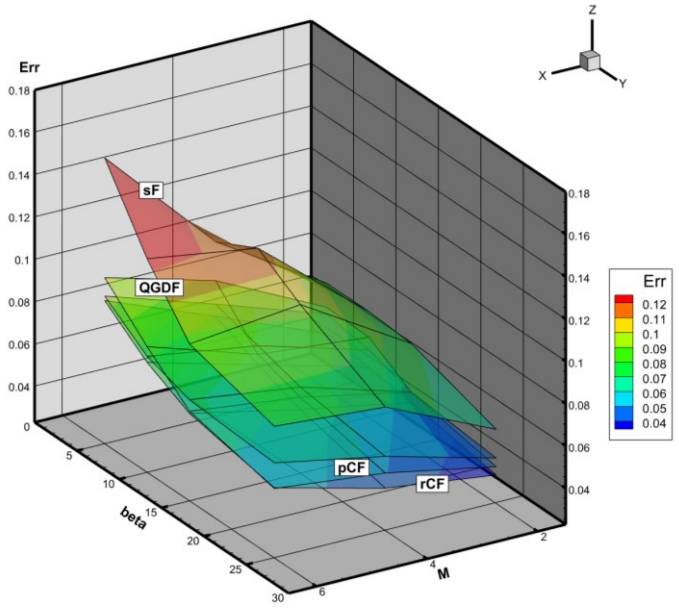

Figure 7. Variation of the deviation from the exact solution for velocity y-component depending on the Mach number and cone half-angle for all solvers in L_2 norm

However, it should be noted that in this study, the smoothing parameter α for the QGDFoam solver was not adjusted, and as shown in [27], modifying this parameter affects the solution's accuracy. α = 0.1 may not be an optimal value for this type of problem. This issue will be addressed in our future research.

The largest absolute increase in error is observed for pressure at an angle of 5° for the sF method when going from M=4 to M=6. The error increases from 0.067521 to 0.174251, giving an absolute increase of 0.10673. The largest relative increase in error at this transition is observed for the velocity Ux at an angle of 5° for the sF method. The error increases from 0.004375 to 0.007974, which corresponds to a relative increase of 1.82 times. It is interesting to note that for the Uy velocity, there is a decrease in error when going from M=4 to M=6 for some methods and angles. For example, for the sF method at an angle of 15°, the error decreases from 0.102818 to 0.097257, this corresponds to a decrease of 5.4%. The rCF method shows the most stable results for both transitions. For example, for pressure at an angle of 25°, the increase in error is only 39% for the transition from M=2 to M=4 and 38% for the transition from M=4 to M=6, which is much smaller than the other methods.

5. Conclusion

This comprehensive study of OpenFOAM solvers for supersonic flow around a Spherically Blunted Cone provides valuable insights into their accuracy across various conditions. A notable trend is the increase in errors with rising Mach numbers from 2 to 6 for all considered quantities. For pressure and density, errors typically increase with increasing half-cone angle up to a certain value, after which the growth slows or stops. Interestingly, for the y-component of velocity, errors generally decrease with increasing cone half-angle. The study highlights that the choice of the most appropriate solver may depend on the specific problem and the physical quantity of interest. The difference in accuracy between methods becomes more pronounced at higher Mach numbers, especially for pressure and density calculations. Future research opportunities include investigating the accuracy of the solvers in more complex geometries and flow conditions, such as unsteady flows or flows with strong shock-boundary layer interaction.

This study provides crucial guidance for researchers and engineers in selecting the most accurate OpenFOAM solver for specific supersonic flow problems, emphasizing the im-

portance of considering both the flow conditions and the physical quantities of primary interest in their simulations.

Acknowledgements

Calculations were performed on the hybrid supercomputer K-100 installed in the Supercomputer Centre of Collective Usage of KIAM RAS.

References

1. OpenFOAM Foundation: [Online]. URL: http://www.openfoam.org (Accessed: 02.07.2024).

2. Gutierrez L. F., Tamagno J. P., Elaskar S. A. High speed flow simulation using OpenFOAM // Mecanica Computacional. 2012. Vol. XXXI. P. 2939–2959.

3. Lorenzon D., Elaskar S. A. Simulacion de flujos supersonicos bidimensionales y axialmente simetricos con OpenFOAM // Revista de la Facultad de Ciencias Exactas, Fisicas y Naturales. 2015. Vol. 2. no. 2. P. 65–76.

4. Bondarev A. E., Kuvshinnikov A. E. Analysis of the behavior of OpenFOAM solvers for 3D problem of supersonic flow around a cone at an angle of attack // CEUR Workshop Proceedings, 2020, V. 2763, p.48–51, CPT2020, Proceedings of the 8th International Scientific Conference on Computing in Physics and Technology, Moscow region, Russia, November 09-13, 2020.

5. Alekseev A. K., Bondarev A. E., Kuvshinnikov A. E. On uncertainty quantification via the ensemble of independent numerical solutions // Journal of Computational Science. 2020. Vol. 42 101114.

6. Alekseev A. K., Bondarev A. E., Kuvshinnikov A. E. Comparative analysis of the accuracy of openfoam solvers for the oblique shock wave problem // Mathematica Montisnigri, 2019, Vol. XLV. P. 95-105.

7. Bondarev A. E., Kuvshinnikov A. E. Analysis and Visualization of the Computational Experiments Results on the Comparative Assessment of OpenFOAM Solvers Accuracy for a Rarefaction Wave Problem // Scientific Visualization. 2021. Vol. 13. № 3. P. 34–46.

8. Бондарев А.Е., Кувшинников А.Е. Задачи сравнительной оценки численных методов на референтных решениях // Труды Семнадцатой Международной научно-технической конференции «Оптические методы исследования потоков». М.: Научно-технологический центр уникального приборостроения РАН, 2023. С. 517–528.

9. Canteros M.A., Polanský J. Review and comparison of two OpenFOAM® solvers: rhoCentralFoam and sonicFoam // EPJ Web of Conf. 2024. Vol. 299 01005.

10. Analysis of the oscillations induced by a supersonic jet applied to produce nanofibers / Quintero F., Doval A.F., Goitia A. et al. // International Journal of Mechanical Sciences. 2022. Vol. 238. 107826.

11. Бондарев А. Е., Галактионов В. А. Построение и обработка результатов вычислительного эксперимента на основе параллельных решений для оптимизационных и параметрических исследований в газовой динамике (пленарный доклад) // Оптические методы исследования потоков: Труды XIV Международной научно-технической конференции. М.: Издательство «Перо», 2017. С.7–15.

12. Bondarev A. E. On visualization problems in a generalized computational experiment //Scientific Visualization. 2019. Vol. 11.2 P. 156–162.

13. Alekseev A. K., Bondarev A. E., Galaktionov V. A., Kuvshinnikov A.E. On the construction of a generalized computational experiment in verification problems // Matematica Montisnigri. 2020. Vol. XLVIII. P. 19–31.

14. Alekseev A. K., Bondarev A. E., Galaktionov V. A., Kuvshinnikov A. E. Generalized Computational Experiment and Verification Problems // Program. Comput. Soft. 2021. Vol. 47 P. 177–184.

15. Zakharova A. A, Korostelyov D. A., Podvesovskii A .G., Bondarev A. E., Galaktionov V. A. Generalized Computational Experiment State Analysis Using Three-Dimensional Visual Maps // Scientific Visualization. 2022 Vol. 14.4 P. 12–23.

16. Kraposhin M., A. 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 Computer Science. 2015. Vol. 66. P. 43–52.

17. Istomina M.A. About realization of one-dimensional quasi-gas dynamic algorithm in the open program OpenFOAM complex // Preprinty IPM im. M.V.Keldysha. 2018. № 001. [In Russian]

18. Kurganov A., Tadmor E. New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations // J. Comput. Phys. 2000. Vol. 160. № 1. P. 241–282.

19. Kurganov A., Noelle S., Petrova G. Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton–Jacobi equations // SIAM J Sci Comput. 2001. Vol. 23 P. 707–740.

20. Greenshields C.J., Wellerr H.G., Gasparini L., Reese J.M. Implementation of semi-discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high-speed viscous flows // Int. J. Numer. Meth. Fluids. 2010. Vol. 63. № 1. P. 1–21.

21. Issa R. Solution of the implicit discretized fluid flow equations by operator splitting // J. Comput. Phys. 1986. Vol. 62. № 1. P. 40–65.

22. Kraposhin M. V., Banholzer M., Pfitzner M., Marchevsky I. K. A hybrid pressure-based solver for nonideal single-phase fluid flows at all speeds // Int. J. Numer. Meth. Fluids. 2018. Vol. 88. № 2. P. 79–99.

23. United collection of hybrid Central solvers — one-phase, two-phase and multicomponent versions: [Online]. URL: https://github.com/unicfdlab/hybridCentralSolvers (Accessed: 02.07.2024).

24. Chetverushkin B. N. Kinetic schemes and quasi-gas-dynamic system of equations. CIMNE, Barcelona, Spain, 2008. 298 p.

25. Elizarova T.G. Quasi-Gas Dynamic Equations. Springer, Berlin, Heidelberg, 2009. 286 p.

26. Lyubimov A.N., Rusanov V.V. Gas Flows Past Blunt Bodies: Part II. Tables of the Gasdynamic Functions. NASA Technical Translation F-714, 1973.

27. Bondarev A. E., Kuvshinnikov A. E. Comparative Estimation of QGDFoam Solver Accuracy for Inviscid Flow Around a Cone // IEEE The Proceedings of the 2018 Ivannikov ISPRAS Open Conference (ISPRAS-2018). P. 82–87.