DOI: 10.14529/jsfi240202

Assessment of Several Advanced Numerical Algorithms Implemented in the CFD Code SINF/Flag-S for Supercomputer Simulations

Evgueni M. Smirnov[1], *Dmitri K. Zaitsev*[1], *Alexander A. Smirnovsky*[1], *Elizaveta V. Kolesnik*[1], *Aleksei A. Pozhilov*[1]

Computational Fluid Dynamics demands substantial computational resources and advanced numerical algorithms for accurate simulation of fundamental and industrial problems. This paper presents an experience in assessing several numerical algorithms implemented recently into the in-house finite-volume code SINF/Flag-S developed at the Peter the Great St. Petersburg Polytechnic University for supercomputer simulation. Three topics are covered: (i) implementation and testing of an original geometric multigrid method for solving linear algebraic equations; (ii) application of a fractional step method for solving unsteady incompressible fluid motion equations; and (iii) description and testing of a density-based solver for compressible gas viscous flow simulation across a wide Mach number range. For each of the topics considered, the results of the calculations of some testing problems are presented, namely: a model problem of heat transfer in a cubic domain, turbulent Rayleigh–Bénard convection in a slightly tilted cylindrical container, free convective flow around a subsea cooler model, high-speed gas flow with strong effects of viscous-inviscid interaction. The parallel efficiency of the implemented algorithms is demonstrated, and their significance for large-scale simulations on supercomputers is highlighted.

Keywords: CFD, multigrid method, incompressible fluid, fractional step method, compressible gas, density-based solver.

Introduction

Computational Fluid Dynamics (CFD) is one of the most resource-consuming domains in the scientific field, which requires a large amount of computational power to perform accurate numerical simulations for various physical problems and tasks. In particular, huge computing power is necessary to perform high-precision numerical simulation of turbulent flows and complex 3D high-speed flows of viscous gas. Therefore, it is of great interest to both improve the technical characteristics of supercomputers and introduce increasingly efficient and reliable numerical methods into codes.

In this paper, we represent our recent experience in assessing some advanced numerical techniques implemented in the in-house finite-volume code SINF/Flag-S, developed at the SPbPU (Peter the Great St. Petersburg Polytechnic University). A brief description of the code is given in Section 1. In the following sections, three topics are considered: (i) implementation and testing of the original geometric multigrid method for solving systems of linear algebraic equations, (ii) approbation of the original fractional step method for solving unsteady equations of the incompressible fluid motion and (iii) description and testing of an effective density-based solver for the compressible gas viscous flows in a wide range of Mach number (sub-, trans- and supersonic). Some data are also provided to illustrate the parallel efficiency of the implemented algorithms.

Multigrid methods are very effective for numerically solving large-size systems of equations. Although the basic idea was proposed back in 1961 [12], the development of effective multigrid methods is still the subject of up-to-date scientific studies [13, 14, 29]. In particular, multigrid methods demonstrate high efficiency at numerically solving systems of linear algebraic equations

[1]Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation

obtained by discretizing a Poisson-like equation. Typical problems, in which the Poisson-like equation arises, are as follows: steady heat transfer in solids, a pressure equation in SIMPLE-like algorithms for pressure-velocity coupling [36] and in fractional-step methods. It is known that the time required to solve the pressure equation in a SIMPLE-like algorithm can reach 90% of the total calculation. It is also known that the complexity of the multigrid algorithm increases as $O(N)$, where N is the number of grid cells, while the complexity of the Krylov sub-space methods (like Conjugated Gradients method, see [41]) increases at best as $O(N^{3/2})$. Thus, the necessity of implementing multigrid method in the numerical codes is essential when dealing with large grids and calculations on supercomputers. In Section 2 of the paper, we present our experience in developing and approbating an original parallelized Geometric Multigrid Method for solving systems of linear algebraic equations that arise when solving elliptical-type governing equations using the Finite Volume Method.

For faster simulation of unsteady incompressible fluid flows, especially in case of turbulent motion, effective and robust numerical methods are needed. Traditional SIMPLE-like algorithms require a number of iterations at each current time layer while advancing in physical time. In the so called non-iterative methods, intermediate iterations between time-layers are almost absent (except the iterations due to the grid "non-orthogonality" issues). In the class of the non-iterative methods, fractional-step methods are very widespread [3, 10, 21]. Usually, these methods are partially or fully implicit; in the latter case, relatively large time steps can be achieved. Section 3 covers a brief description of the implicit fractional-step method implemented in the SINF/Flag-S code for solving unsteady equations of incompressible fluid motion and two examples of its application for performing Direct Numerical Simulation of free convective flows.

In industry and aero-engineering, problems often arise, where compressible flows are present in a wide range of velocities. Up-to-date numerical methods for calculating compressible flows provide the ability to accurately resolve different gas-dynamic discontinuities. There are many different schemes for calculating convective fluxes in case of trans- and supersonic motion, which are currently being actively developed [31, 45]. On the other hand, there are many applications where the role of viscous motion in subsonic regions are also very significant, for example, the flow of gas or vapor in a turbine vane cascade [30] or the flow around a helicopter blade [1]. For these applications, simulation should be performed with a reliable and robust numerical method that allows a smooth transition from subsonic to supersonic flow, taking into account viscous effects. Section 4 briefly describes our experience in implementing robust numerical techniques for calculating viscous compressible gas flows in a wide range of the Mach number followed by an example of using these techniques for accurate test computations of a high-speed 3D problem with complex effects of viscous-inviscid interaction.

1. General Description of the SINF/Flag-S Code

The finite-volume "unstructured" code SINF/Flag-S operates on unstructured meshes with cells of arbitrary topology (tetrahedrons, hexahedrons, prisms, polyhedra). The code is written primarily in the Fortran 90 programming language, with some functions written in C programming language. The SINF/Flag-S program is a console application; all information is read from control text files. The ANSYS Fluent format is used for mesh files and results files; to allow restarting from saved fields, a special file format is used.

The SINF/Flag-S code is intended for simulation the following classes of problems:

- steady and unsteady flows of viscous fluid;

- compressible viscous gas flows in a wide range of the Mach number;
- conjugate or non-conjugate heat transfer in liquids, gases and solids;
- simulation of turbulent flows using different approaches, such as: (U)RANS, LES, IDDES, DNS;
- heat buoyancy in the mass force fields of different kinds: gravitational, centrifugal, Coriolis;
- flows in the porous medium with accounting for phase interchange (evaporation and condensation);
- multiphase flow simulation using the Volume of Fluid approach;
- two phase filtration flow simulation using the "black oil" model;
- calculation of flows in domains with moving boundaries.

In the case of incompressible or low-Mach flows, different methods of solving governing equations can be used, in particular: the SIMPLEC method [48], the Rogers–Kwak scheme [40], the coupled method based on the Rhie–Chow interpolation [9]. To calculate unsteady flows, iterative algorithms based on the above-mentioned methods or the "nominally" non-iteration fractional step method (see Section 3.1) can be used. For simulation of the compressible viscous gas flows in a wide range of the Mach number (from sub- to supersonic), the coupled density-based solver is used (see Section 4.1).

Discretization of spatial operators is performed using the Finite Volume Method on collocated grids. For the approximation of viscous (diffusion) terms, the hybrid central scheme of 2nd order with non-orthogonality correction is used. Different interpolation schemes can be used for the approximation of the convective terms in governing equations. For the incompressible or low-Mach flows there are following interpolation schemes: central (linear, 2nd or 4th order using gradients), second order upwind, QUICK, weigthed schemes, bounded schemes with Jameson or scalar limiters. In the density-based solver, different schemes for inviscid fluxes calculation have been implemented, which are briefly described in Section 4.

For solving systems of linear algebraic equations, in the SINF/Flag-S code there are different linear solvers, in particular: CG, BiCG, GMRES, Geometrical Multigrid (see description in Section 2.1), which can be preconditioned, in particular, by Symmetric Gauss–Seidel preconditioner, Incomplete Cholesky or LU factorization.

To ensure the ability to solve the problem in parallel mode, the domain decomposition approach based on task parallelism using the MPI (Message Passing Interface) library of the MPI 2.0 standard is used [15]. The division of the problem into separate subtasks is carried out based on blocks of a multi-block computational mesh, for which the code contains an auxiliary utility Par4Flag; the latter uses the ParMetis library [27] to decompose the computational mesh. To exchange information between processes, an additional row of "halo" cells is used, geometrically duplicating the border cells of the adjacent block.

2. Geometric Multigrid Method and Evaluation of Its Performance

2.1. Description of Implemented Geometric Multigrid Method

As it was shown by the practice of hydrodynamic calculations, the most resource-intensive part of the SIMPLE-like algorithms is solving the Poisson equation for pressure correction. The Poisson equation also describes thermal conductivity in solid structural elements and the filtration movement of the working fluid in a porous wick. Thus, using a specialized algorithm in a

linear solver that speeds up obtaining a Poisson equation solution can significantly reduce the time spent on solving the problem and, accordingly, significantly increase the efficiency of using available computing resources. When solving linear systems of large size, multigrid methods demonstrate the greatest efficiency, since their use allows achieving a linear increase in the time required to solve a problem as its size increases. The main difficulty in applying multigrid methods on unstructured meshes is in constructing grids for coarse levels. One of the most promising is the agglomeration method [34], in which the cells of coarse computational grids are obtained by agglomeration (combining) the cells of the original grid. The question of the most effective method of cell agglomeration still remains open and is the subject of a number of modern studies, for example [32, 35].

An original agglomeration multigrid method was implemented in the SINF/Flag-S code. The agglomeration method is based on the algorithm proposed in [37]. At the same time, it includes several original modifications, which significantly improve the quality of the resulting agglomerates and, accordingly, increase the stability and speed of convergence of the multigrid method. The modified agglomeration algorithm consists of the following steps, repeated cyclically after the initial creation of a queue of border cells:

1. Construction of the basis. A cell from the queue is merged with two (2D case) or three (3D case) neighboring cells that have a common node, and the distance to which does not exceed the distance to the nearest cell by more than 4 times. If there are several options for merging, then the one that contains the largest number of cells already in the queue is selected (if there are still several options, then the first one is taken). If a suitable set of cells is not found and the distance to at least one of the neighboring cells exceeds the distance to the nearest cell by more than 4 times, then the first step of the algorithm is repeated, with an attempt to construct a basis containing one less cell. If it is possible to form a basis in the end, then proceed to step 2; otherwise – to consider the next cell from the queue.
2. Addition of cells. The cells adjacent to the resulting agglomerate are examined. If among them there is a cell adjacent to at least two cells from the agglomerate, and the addition of this cell reduces the value of the parameter Ar (aspect ratio, defined as the ratio of linear dimensions characterizing the surface area and volume of the agglomerate), then this cell is added to the agglomerate; otherwise, the transition to the next step is carried out. If there are several suitable cells, then the one whose addition leads to a smaller value of the Ar parameter is selected. If the number of cells included in the agglomerate is less than 4 (2D case) or 8 (3D case), then the step is repeated.
3. Queue replenishment. Cells that are adjacent to the resulting agglomerate and have not yet been combined with other cells are added to the queue (if they have not already been added to it). Then the next cell from the queue is processed (step 1).
4. Holes patching. All unmerged cells are marked. The selected unmerged cell is merged with one of the neighboring agglomerates. It is forbidden to add an unmerged cell to the agglomerate if all the neighbors of this cell included in the agglomerate were marked (i.e., they were also unmerged cells). Among the agglomerates allowed for combining, the one for which the addition of this cell will most strongly decrease (or least increase) the value of the Ar parameter is selected. Then proceed to the next unmerged cell.

The number of grid levels is selected automatically, and the construction of new grids continues until one cell remains in each calculation block. The difference operator on coarse grids is formed by summing the equations with scaling coefficients [26], however, instead of the global scaling

proposed in [26] (by the number of grid cells), local scaling is used (based on the ratio of the distances between the centers of the cells of the fine and coarse grids), which increases efficiency of the method in the case of grids with highly elongated cells. The Symmetric Gauss–Seidel method (SGS) is used as a smoother. Various types of multigrid cycles (V-, W-, F-cycle) with a fixed number of iterations have been implemented. The developed multigrid method can be used both as an independent solver and as a preconditioner for the GMRES method.

2.2. Evaluation of Method Performance and Parallelization Efficiency

The effectiveness of the developed multigrid method was evaluated on a model problem of heat transfer in a cubic domain with constant thermal conductivity (see Fig. 1). Almost all the boundaries of the region were considered adiabatic, except two small regions adjacent to the opposite vertices, in one of which the temperature value was specified, and in the second the heat flux was specified. The dimensionless formulation of the problem is illustrated in Fig. 1.

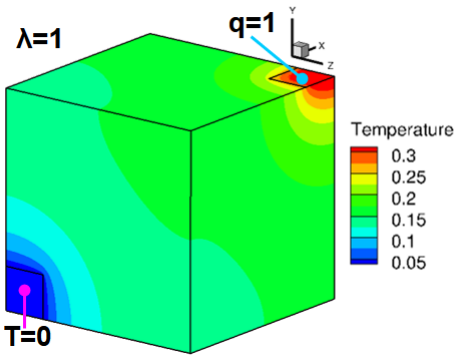


Figure 1. Model problem of conductivity heat transfer in a cube

Comparative calculations were performed using the SINF/Flag-S code and the ANSYS Fluent 23R1 code on uniform (isotropic) hexahedral and tetrahedral computational grids of different spatial resolutions, as well as on polyhedral grids obtained from the tetrahedral one using the ANSYS Fluent package. When using the SINF/Flag-S code, the two different methods were used as linear solvers: the developed multigrid method (with W-cycle) and the GMRES method (with a SGS preconditioner). Of the options available in the Fluent code, the BCGSTAB linear solver method with the multigrid preconditioner using a W-cycle has been selected as the most effective one for solving the formulated task (based on the results of preliminary tests). Calculations were performed in single-threaded mode. The time required to solve the problem with a prescribed accuracy was being determined (the problem was considered solved when the value of the relative integral imbalance of the heat flux became less than 10^{-4}). Figure 2 shows the dependencies of the problem computational time on the number of computational grid cells obtained with various codes and numerical methods.

Here, the data obtained in calculations with polyhedral grids are presented; calculations with hexahedral and tetrahedral grids gave similar results. It can be seen that when using the GMRES method (code SINF/Flag-S), the time to solve the problem increases proportionally to the number of cells to the power of 1.5. The use of both multigrid algorithms provides, as expected, a linear increase in computational time as the number of grid cells increases. The multigrid method developed and implemented in the SINF/FlagS code made it possible to solve this problem about 2 times faster than using the Fluent code.

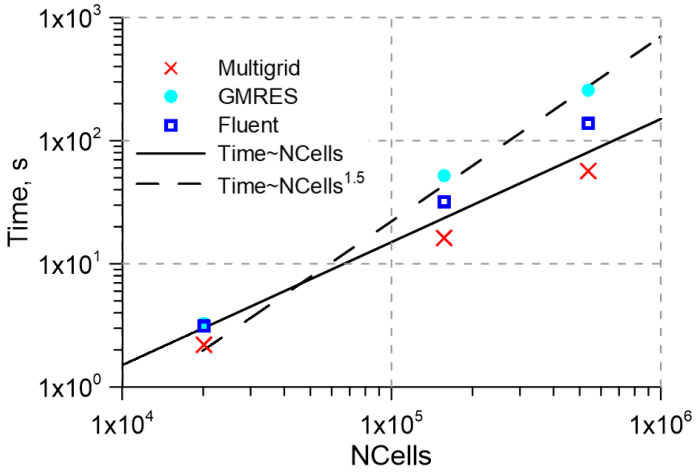
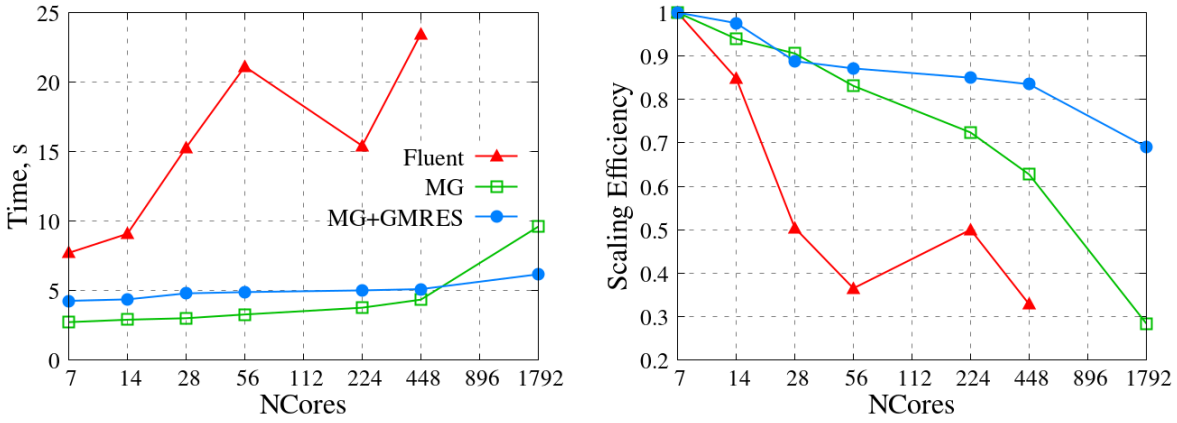


Figure 2. Efficiency of different solvers with increasing the grid size

The scalability of the developed multigrid method was evaluated as well. The considered model problem was solved using 7 to 1792 cores while maintaining the number of cells per computing core; accordingly, the grid size varied from four million to a one billion cells. The calculations were carried out on the SPbPU Polytechnic RSC Tornado supercomputer. Figure 3 shows the dependencies of the problem-solving time and the scaling efficiency on the number of cores involved when using the SINF/Flag-S and ANSYS Fluent 23R1 codes. In calculations with the SINF/Flag-S code, the developed multigrid method was used both independently (MG) and as a preconditioner for the GMRES method (in the latter case, a more economical V-cycle was used instead of the W-cycle). When using the Fluent code, as before, the BCGSTAB method with a multigrid preconditioner turned out to be the most effective. From Fig. 3 one can see that when using a relatively small number of cores (up to 500), the MG+GMRES combination in the SINF/Flag-S code works up to one and a half times slower than the pure multigrid method, but the scalability of such combination is better. One can conclude also that for the considered model problem, the Fluent code is significantly inferior to the SINF/Flag-S code both in terms of problem solving time and scalability.



(a) (b)

Figure 3. "Weak" parallel scalability: computational time (a) and scaling efficiency (b)

3. Fractional Step Method and Its Testing

3.1. Description of Method

As it was mentioned in Introduction, in some cases, when solving unsteady equations of incompressible fluid dynamics, the use of iterative time advancement methods (the SIMPLE method and similar) can be very resource-consuming. In particular, simulation of turbulent flows using eddy-resolving approaches such as LES, DES, etc., requires small time steps to adequately resolve the dynamics of turbulent eddy structures. To speed up the solution, it is reasonable to use faster numerical approaches based on a non-iterative transition from one time layer to another. One such approach is the so-called projection method or fractional step method [8, 10, 21, 33].

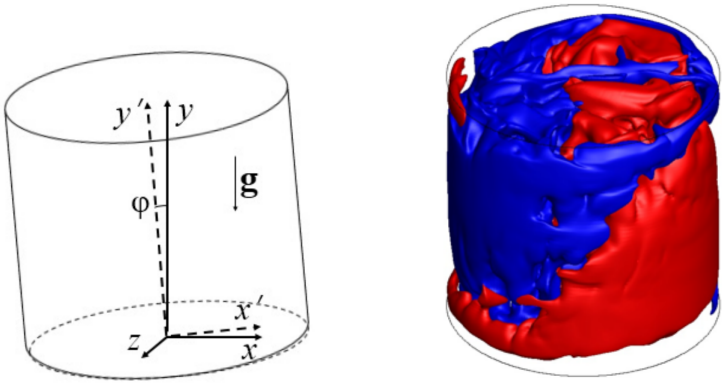
The fundamental principle of the fractional step method (FSM) lies in the separation of the spatial operators in the momentum equation. The pressure gradient is interpreted as a projection operator that transforms an arbitrary velocity field into a solenoidal one. The momentum balance equation is thus split into two equations. The first one is solved either without the pressure gradient at all or with the pressure gradient taken from the previous time layer and serves to determine the predictor value of the velocity. This predictor velocity, of course, does not necessarily satisfy the continuity equation. The second equation, which is written for the corrector velocity vector, relates the pressure gradient at the new time layer to the velocity vector at the new time layer. The combination of the second equation and the continuity equation leads to the Poisson equation for the pressure at the new time layer. It is important to emphasize that, unlike SIMPLE-like algorithms, where a Poisson-like equation also arises (it is written for pressure correction), the resulting equation for pressure at the new time layer is physically justified and should be solved with good accuracy to ensure the mass balance equation at the new time layer is satisfied. So, the use of advanced numerical techniques like multigrid methods is rather essential for the FSM.

Another important aspect that affects the effectiveness of the FSM is the evaluation of the convective terms in the predictor velocity equation. The convective term can be calculated by different methods. The most common approach includes the extrapolation of the convective term from the two preceding time levels using the Adams–Bashforth scheme. For the better stability of the numerical algorithm, implicit schemes for calculating the convective term are preferable; they can be introduced in different ways, for example, as in [6, 19]. The SINF/Flag-S code uses an original scheme for computing the convective terms, which combines extrapolation from two preceding time levels with introducing implicitness. Namely, the mass flux through the face of the finite volume is extrapolated from previous time layers using the Adams–Bashforth scheme (at the intermediate time layer in the case of the Crank-Nicolson scheme or at a new time layer in the case of the backward 2nd order scheme for time derivative approximation), while the transported quantity (velocity vector) is written implicitly at the desired time layer (intermediate or new). Our experience in using the proposed algorithm in the SINF/Flag-S code shows that the introduced implicitness can essentially increase the FSM stability in the case of highly skewed and non-orthogonal cells. For more details concerning the realized FSM see [44].

Below, the performance of the fractional step method implemented in the SINF/Flag-S code is illustrated by two examples: turbulent Rayleigh–Bénard convection in a slightly tilted cylindrical container and free convective transitional flow near a subsea cooler model.

3.2. Turbulent Rayleigh–Bénard Convection in a Slightly Tilted Cylindrical Container

Rayleigh–Bénard convection of fluid in a vertical cylindrical container is one of the canonical model problems in the investigations of natural convection [2]. Experimental and numerical studies show that large-scale circulation (e.g. in the form of large-scale vortex covering the entire region of convective flow) is a characteristic feature of such flow. Using the SINF/Flag-S code, Direct Numerical Simulation of water turbulent convection in a slightly tilted cylindrical container heated from below was performed recently [43] at the Rayleigh number $Ra = 10^8$ and the Prandtl number $Pr = 6.4$ (the container height was equal to its diameter). As shown in [42], tilting of a cylinder leads to "locking" of the large-scale vortex in a certain azimuthal position. In Fig. 4, the computational domain as well as large-scale vortex vizualization are presented. The simulation was carried out based on the Navier–Stokes equations written with the Boussinesq approximation for including the buoyancy effects.

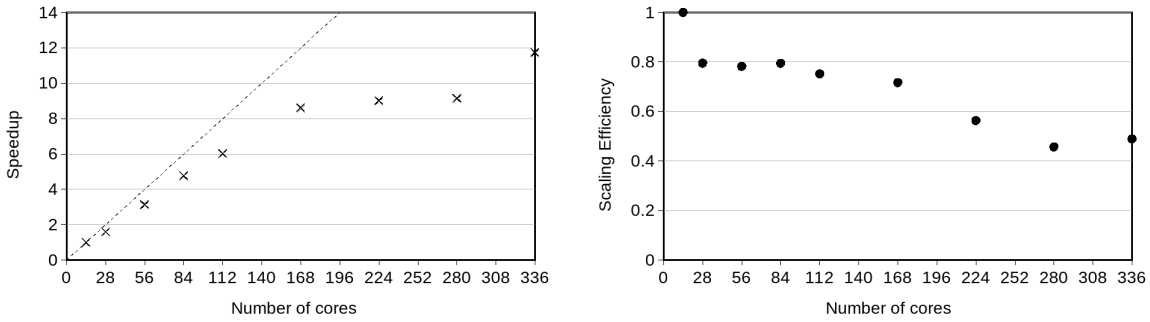


(a) (b)

Figure 4. Geometry of computational domain for tilted container (a) and visualization of large-scale vortex by isosurfaces of positive (red) and negative (blue) instantaneous vertical velocity component (b)

The fractional step method with the Crank–Nicolson scheme was used for advancing in time. A central difference scheme was used to approximate the convection terms in the governing equations. The computational grid consisted of approximately 1.5×10^7 cells. The time step was chosen so that the local values of the Courant number were less than unity; characteristic time of the buoyancy process (buoyancy time) was estimated as about 25000 time steps. The computed fields were averaged over time starting after a transient process; the sample for averaging was about 150 buoyancy times.

The calculations were performed using resources of the Polytechnic Supercomputer Center, real time of the calculation of 150 buoyancy time (about 4×10^6 time steps) using 12 nodes (28 cores on each node) of the supercomputer took about two months. More details about the study and results can be found elsewhere [43]. Testing of the parallelization efficiency was performed as well. The results on the calculated speedup and scaling efficiency are presented in Fig. 5. It can be seen that a strong degradation in scalability is observed in the interval from 168 to 224 cores. Nevertheless, the scaling efficiency in case of a large number of cores remains about 0.5 that is a relatively good level.



(a) (b)

Figure 5. Parallel scalability: speed-up (a) and scaling efficiency (b)

3.3. Free Convective Flow Around a Subsea Cooler Model

The problem of free convective flow around a subsea cooler arises from the practical need to have reliable data on the heat transfer efficiency of a heat exchanger located on the seabed and intended to cool natural gas produced from the shelf [16]. Such passive heat exchangers consist of a bank of pipes that use the surrounding cold water as a coolant (see Fig. 6). The heating of the surrounding water induces a buoyancy-driven draft flow, resulting in mixed convection conditions for the inner-rows tubes; this convection is characterized by relatively low Reynolds numbers, regarding the flow around the pipes. When the heat exchanger design relies primarily on the data for natural convection, the heat transfer coefficient at the external surface of the pipe becomes a critical uncertainty. Up-to-date CFD methods can provide valuable data on external heat transfer for the low-Reynolds-number regimes commonly encountered in buoyancy-dominated coolers. More details about the conditions typical for subsea cooler simulation can be found elsewhere [17, 18].

Here we present some results of the 3D simulation of water flow near a subsea cooler consisting of ten and twenty-two pipes with inlet and outlet manifolds (Fig. 6). No-slip conditions and a constant temperature were set on the surface of heated pipes and collectors. It was assumed that the exchanger was located in a boundless water basin, the outer boundaries of the computational region were moved away from the cooler to a large distance, and conditions for the free flow of water were imposed on them.

The simulation was carried out based on the direct solution of the unsteady Navier–Stokes equations; buoyancy effects were taken into account in the Boussinesq approximation; typical values of the Grashof number evaluated for a pipe was about 10^5. The calculations were performed using the SINF/Flag-S code with the fractional step method. Resources of the Polytechnic Supercomputer Center were used. For the bank of ten pipes, the grid consisted of more than 75 million cells, and to obtain a representative sample, the calculations using 448 cores were being carried out in two weeks. For the bundle of 22 pipes, the grid size was about 150 mln cells. Using 896 cores, a representative sample was also obtained in two weeks.

Visualization of the flow is shown in Fig. 7, where streamlines colored by temperature as well as temperature distributions in cross sections are presented. It can be seen that in the center of the tube bank, the flow is nearly stationary. Contrary to that, the flow in the upper pipes region is highly unsteady, with almost turbulent behavior.

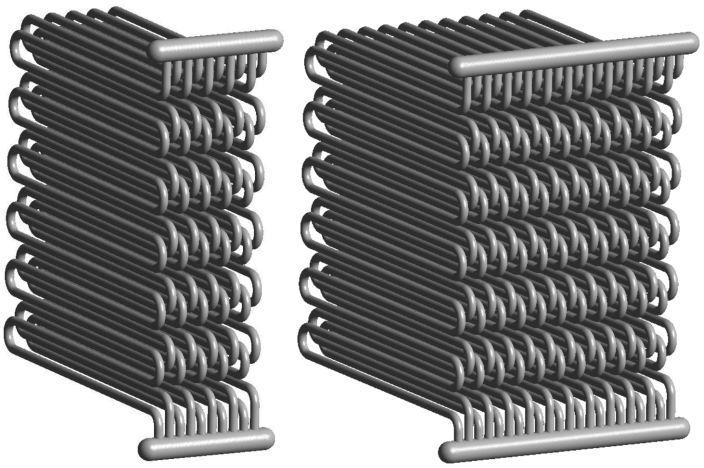
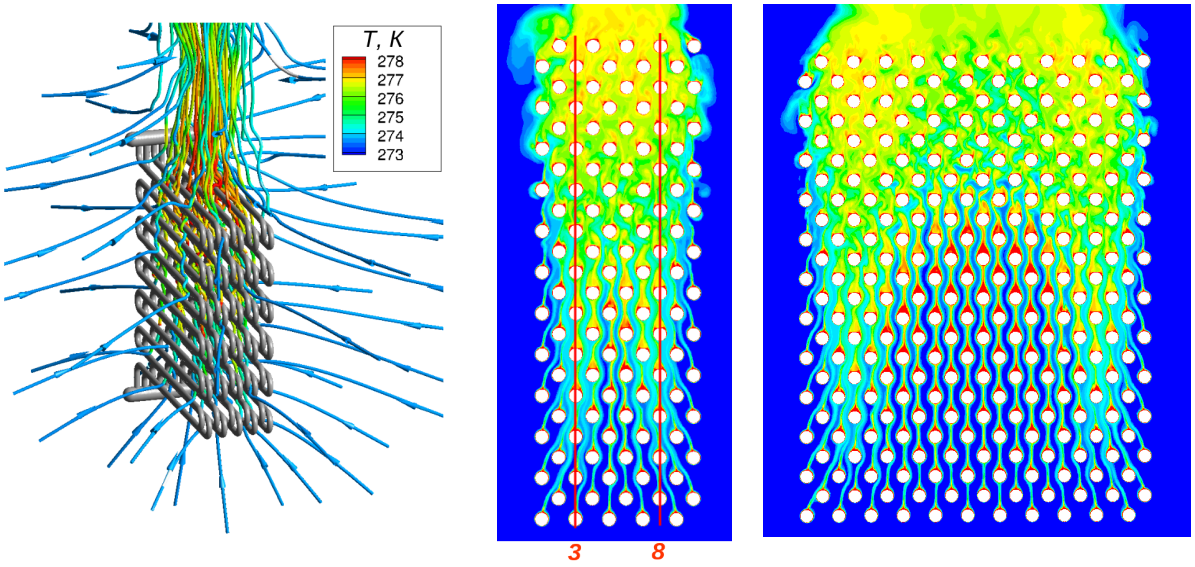


Figure 6. The computational geometry of subsea cooler consisting of 10 (left) and 22 (right) pipes



(a) (b) (c)

Figure 7. Flow visualization: a – streamlines colored by temperature for the bank of 10 pipes; b and c – temperature in cross section for two geometries (red lines show the position of 3rd and 8th row)

4. Coupled Density-based Solver for Viscous Compressible Gas Flows in a Wide Range of Mach Number

4.1. General Remarks for Developed Density-based Solver

Compressible gas flows are characterized by the presence of discontinuities of various kinds, which especially manifest themselves at high flow velocities (when Mach number greater than unity) in the form of shock waves. In numerical simulation of such flows, the method of approximating convective flux is especially important, and the scheme used must in one way or another take into account the characteristic properties of the systems of equations. In the SINF/Flag-S code, the following schemes are available for convective flux evaluation: Godunov, Roe, HLL, HLLC, and schemes of the AUSM family.

High order of accuracy is achieved using the MUSCL slope-limiting approach, according to which piecewise-linear reconstruction of the solution is built for each control volume. In the SINF/Flag-s code, there are two approaches for constructing a high-order accuracy scheme for the case of unstructured grids: a scalar approach with second-order limiters [5], and a quasi-one-dimensional approach, consisting in applying quasi-one-dimensional computations for locally selected directions (i.e., for each face). The proposed in [4, 28] version of the quasi-one-dimensional approach, where additional virtual points are reconstructed, is implemented in the SINF/Flag-S code. To preserve monotonicity of the solution near discontinuities, second-order TVD schemes are used. Thanks to the reconstruction, refined values of the variables to the left and right of the control volume face are calculated, which are then used to calculate the inviscid fluxes when selecting one of the above-mentioned schemes.

In the case of very high Mach numbers, the so-called "carbuncle" instability may appear in the numerical solution [38], which leads to a strong distortion of the shock wave front in the numerical solution. In the SINF/Flag-S code, additional artificial viscosity, introduced according to [39], is used to suppress this instability.

On the other hand, the main difficulty in modeling low-speed compressible flows lies in the "acoustic stiffness" (a strong difference in the magnitude of the eigenvalues) of the system of equations for the dynamics of a compressible gas, which arises at substantially subsonic flow velocities [7, 11, 20, 46]. This is due to the very different characteristic times of convective transport and propagation of acoustic disturbances and leads to the fact that in the system of equations the matrix for time derivatives becomes degenerate as the Mach number tends to zero. Under these conditions, the stability and speed of convergence of traditional methods of numerical integration of the full Navier–Stokes equations, many of which were originally developed to calculate supersonic flows, sharply deteriorate, and in the limiting case of the Mach number approaching zero, these methods generally lose their functionality. To overcome this difficulty, regularization based on the Turkel preconditioning is implemented into the SINF/Flag-S code, according to [49].

The implemented in the code SINF/Flag-S density-based solver was successfully approved on the series of different problems, including the problem of the sub- or supersonic flow past the blunt-fin body mounted on a plate. The detailed description of the problem and numerical results both for incompressible fluid flows with different Prandtl numbers and the compressible gas flow with the Mach number ranging from 0.01 to 0.5 are presented in [24], with the focus on the effect of compressibility on heat transfer intensification in the region occupied by horseshoe-shaped vortex structures. In the next section, we present the results of numerical simulations and parallel efficiency testing for a representative three-dimensional problem of high-speed gas flow with strong effects of viscous-inviscid interaction.

4.2. Test Problem

The problem formulation is based on the data of the computational and experimental study [47], which investigated the structure of laminar supersonic air flow (Mach number $M_\infty = 6.7$, Prandtl number $Pr = 0.71$, and ratio of specific heats $\gamma = 1.4$) past a fin mounted on a plate (Fig. 8). Following [47], the flow is assumed to be laminar throughout the domain. The Reynolds number, calculated from the bluntness diameter ($D = 2.5$ mm), corresponds to the minimum value considered in [47]: $Re_D = 1.25 \times 10^4$. The free-stream total temperature is $T_0 = 630$ K. The computational domain is shown in Fig. 8. The fin is positioned at a distance

of $L_{plate} = 145$ mm from the plate's leading edge. The other dimensions of the computational domain are taken from [47] as follows: $R = 76.5$ mm, $L_f = 60$ mm, and $H = 25$ mm.

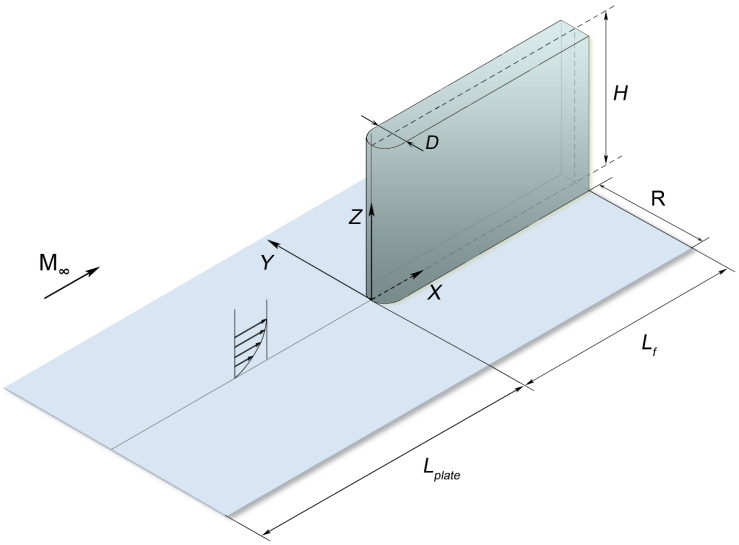


Figure 8. Test problem definition

The full three-dimensional Navier–Stokes equations were solved treating air as a perfect gas. The temperature dependence of the viscosity coefficient was evaluated using Sutherland's law. An uniform flow was specified at the inlet boundary of the computational domain, and the no-slip condition was applied on the body surface and the plate. Non-reflecting boundary conditions were specified on the lateral and upper boundaries, and the zero-gradient condition was specified at the outlet. The body surface and the plate were maintained at a constant temperature, $T_w = 300\ K$, corresponding to a temperature ratio of $T_w/T_\infty = 4.76$.

The convective fluxes on the faces of control volumes were evaluated using the second-order AUSM scheme with the van Albada TVD limiter. A nominally second-order scheme was used for evaluating viscous fluxes. The simulation was performed based on the time-dependent formulation, using an implicit time-advancing scheme of second-order accuracy (the backward three-layer scheme). Under considered conditions, the transient process converged to a steady-state solution. The basic simulation was performed using a quasi-structured grid with 20 million cells. The grid had approximately 70 cells across the incoming boundary layer, which had a thickness of about $1.2D$ before separation. The first near-wall cell is approximately 0.01 mm in the normal direction. For testing grid sensitivity, the simulation was also performed on a finer grid, which consisted of 50 mln cells.

Figure 9 provides a general overview of the flow structure. The figure shows the volume streamlines and the relative heat flux distribution over the body and plate surfaces (q_{w0} is the heat flux calculated for a plate without an obstacle). The flow's primary characteristics include the formation of a separation region with a system of horseshoe vortices that extend around the fin's side, and a pronounced non-monotonicity in the heat flux distribution within the junction region.

Figure 10 presents the calculated distributions of the plate-surface nondimensional heat flux (normalized by the heat flux local values q_{wp}, which were calculated for the smooth plate without an obstacle at the corresponding points) along the symmetry line (a) and along a cross line $x/D = 1$ (b). Additionally, experimental and numerical data from [47] are included. The

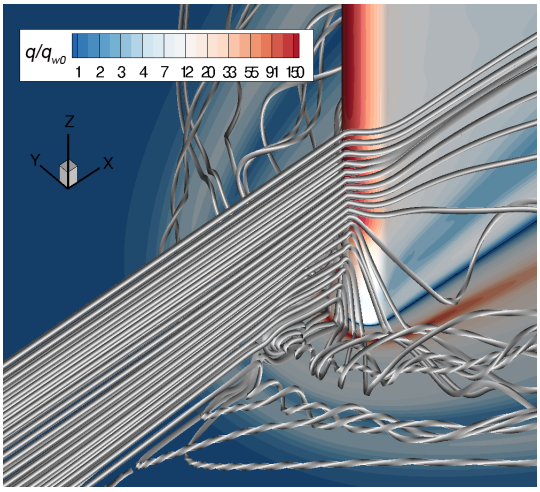


Figure 9. Visualization of the three-dimensional flow structure

primary specificity in the calculated heat flux distributions along the symmetry line is that the distributions exhibit two local maxima, in addition to the global one (it is outside the field of the Fig. 10). Our computational results align well with the computational data reported in [47]. Let us note that the cited paper authors believed that good agreement of their numerical data with the experimental results had been achieved. Also, there is negligible difference between the results obtained using the base grid and the refined grid for both solutions, confirming the adequacy of the grid resolution.

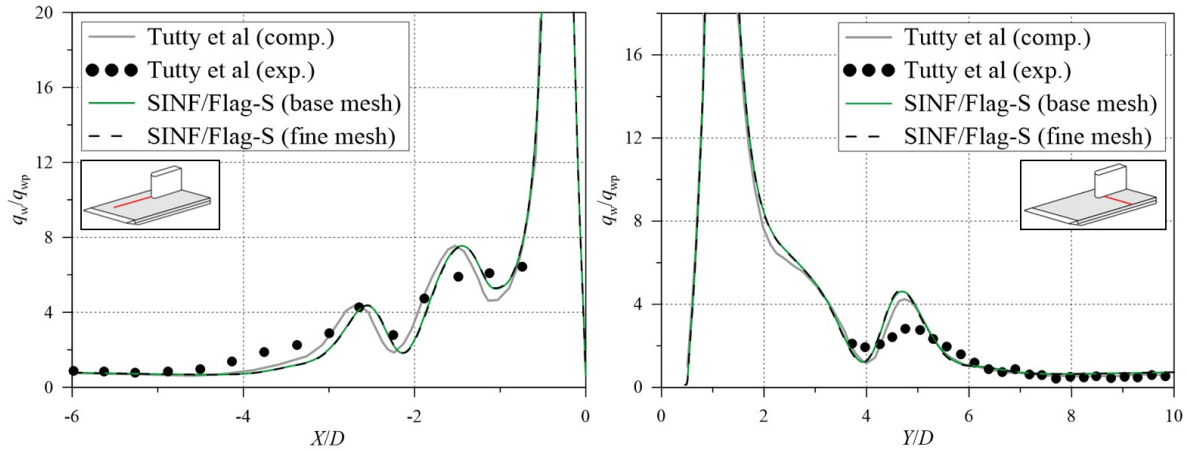
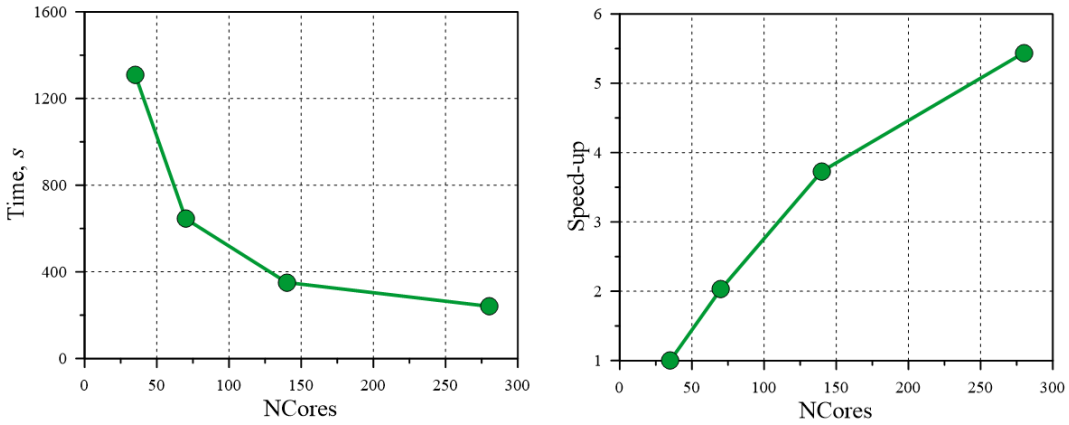


Figure 10. Relative heat flux distribution: (left) along the line of symmetry on the plate $Y/D = 0$; (right) along line $X/D = 1$

Testing the efficiency of code parallelization was carried out by running simulation using a different number of cores of the Polytechnic Supercomputer Center cluster. The time for 100 iterations was calculated. The results are presented in Fig. 11.

Concluding this section, one should note that the problems of non-uniqueness of the solution can occur when studying this or a related class of gas flow (see [22, 23, 25]). Since, additionally to geometry, there are many other parameters determining such tasks (Reynolds number, Mach number, temperature factor, relative thickness of incoming boundary layer), getting knowledge on the challenging problem of the non-uniqueness of the solution assumes running a huge amount of parametric refined calculations on power supercomputers.



(a) (b)

Figure 11. Parallel scalability: calculation time (a) and speed-up (b)

Conclusion

In the paper, we have presented our recent experience in assessing several advanced numerical techniques implemented in the in-house SINF/Flag-S code.

To improve the efficiency for solving systems of linear algebraic equations, which appears as a result of the Poisson equation discretization, an original parallelized geometric multigrid method was implemented in the SINF/Flag-S code. Dealing with a model problem of conduction heat transfer in a cubic domain, it was shown that the implemented multigrid method, used both as a separate linear solver and as a preconditioner to the GMRES method, demonstrates very good performance and scalability both in single-threaded and in parallel mode. It was also established that the multigrid method implemented in the last versions of the ANSYS Fluent code is significantly inferior to the one implemented in the SINF/Flag-S, both in terms of problem-solving time and scalability.

To improve the efficiency of solving essentially non-stationary problems (in particular, in turbulent flow simulation using eddy-resolved approaches), an original fully-implicit fractional step method was implemented in the SINF/Flag-S. The algorithm approbation was illustrated by two examples of unsteady flows. The first one is the DNS of turbulent Rayleigh–Bénard convection in a slightly tilted cylindrical container. Here, the computational domain has a relatively simple geometry, but the flow is characterized by intense turbulent motion with strongly different turbulent scales: a large-scale circulation has the form of a vortex covering the entire volume, while the size of the vortices responsible for dissipation is several orders of magnitude smaller. Accurate simulation of all turbulent scales requires very small time steps, so the non-iterative solver is preferable, especially when a large enough sample is needed for reliable statistics. Results of testing of the implemented fraction step method in parallel mode showed relatively good scaling efficiency, even in the case of numerous cores: although at first it degrades with increasing numbers of cores, it later goes to a level of about 0.5, which is quite acceptable. The second example is related to simulation of unsteady free convective flow around a subsea cooler model. This industrial problem is characterized by a numerous heated elements (pipes) interacting with the surrounding water, so grids of very large size should be used for simulation. On the other hand, the flow in such configurations may be simultaneously laminar in some regions of the whole flow and transient or even turbulent in other regions. The use of the fractional step method also gives a significant increase in computational performance here.

A reliable tool for numerical simulation of compressible gas flows in a wide range of the Mach numbers is important for various practical applications, including turbomachinery and aeronautics. With considering a complex 3D model problem of the viscous-inviscid interaction of supersonic flow past a blunt fin mounted on a plate, we have demonstrated the ability of the coupled density-based solver implemented in the SINF/Flag-S code to accurately resolve both the shock wave structure and the regions of substantially subsonic flow, where viscous effects play a significant role (in the boundary layer and within horseshoe vortices). The performed investigation of parallel scalability showed fairly good speed-up of the calculations when using up to several hundred computational cores.

Acknowledgments

The results of the work were obtained using computational resources of Peter the Great Saint-Petersburg Polytechnic University Supercomputing Center (www.spbstu.ru).

References

1. Abalakin, I.V., Bobkov, V.G., Kozubskaya, T.K., *et al.*: Numerical simulation of flow around rigid rotor in forward flight. Fluid Dyn. 55(4), 534–544 (Jul 2020). https://doi.org/10.1134/S0015462820040011

2. Ahlers, G., Grossmann, S., Lohse, D.: Heat transfer and large scale dynamics in turbulent Rayleigh–Bénard convection. Rev. Mod. Phys. 81, 503–537 (Apr 2009). https://doi.org/10.1103/RevModPhys.81.503

3. Anunciação, M., Augusto Villela Pinto, M., Neundorf, R.: Solution of the Navier–Stokes equations using projection method and preconditioned conjugated gradient with multigrid and ILU solver. Rev. Int. Métodos Numér. Cálc. Diseño Ing. 36(1), 17 (2020). https://doi.org/10.23967/j.rimni.2020.01.003

4. Bakhvalov, P.A., Kozubskaya, T.K.: Cell-centered quasi-one-dimensional reconstruction scheme on 3D hybrid meshes. Math. Models Comput. Simul. 8(6), 625–637 (Nov 2016). https://doi.org/10.1134/S2070048216060053

5. Barth, T., Linton, S.: An unstructured mesh Newton solver for compressible fluid flow and its parallel implementation. In: 33rd Aerospace Sciences Meeting and Exhibit. https://doi.org/10.2514/6.1995-221

6. Bevan, R.L.T., Boileau, E., van Loon, R., *et al.*: A comparative study of fractional step method in its quasi-implicit, semi-implicit and fully-explicit forms for incompressible flows. Int. J. Numer. Methods Heat Fluid Flow 26(3/4), 595–623 (May 2016). https://doi.org/10.1108/HFF-06-2015-0233

7. Carrión, M., Woodgate, M., Steijl, R., Barakos, G.: Implementation of all-Mach Roe-type schemes in fully implicit CFD solvers – demonstration for wind turbine flows. International

Journal for Numerical Methods in Fluids 73(8), 693–728 (2013). https://doi.org/10.1002/fld.3818

8. Chorin, A.J.: Numerical solution of the Navier–Stokes equations. Math. Comput. 22(104), 745 (Oct 1968)

9. Darwish, M., Moukalled, F.: A Fully Coupled Navier–Stokes Solver for Fluid Flow at All Speeds. Numerical Heat Transfer, Part B: Fundamentals 65(5), 410–444 (2014). https://doi.org/10.1080/10407790.2013.869102

10. De Michele, C., Capuano, F., Coppola, G.: Fast-Projection Methods for the Incompressible Navier–Stokes Equations. Fluids 5(4) (2020). https://doi.org/10.3390/fluids5040222

11. Dellacherie, S., Jung, J., Omnes, P., Raviart, P.A.: Construction of modified Godunov-type schemes accurate at any Mach number for the compressible Euler system. Mathematical Models and Methods in Applied Sciences 26(13), 2525–2615 (2016). https://doi.org/10.1142/S0218202516500603

12. Fedorenko, R.P.: A relaxation method for solving elliptic difference equations. USSR Computational Mathematics and Mathematical Physics 1(4), 1092–1096 (1962). https://doi.org/10.1016/0041-5553(62)90031-9

13. Gorobets, A.V.: An approach to the implementation of the multigrid method with full approximation for CFD problems. Computational Mathematics and Mathematical Physics 63(11), 2150–2161 (2023). https://doi.org/10.1134/S0965542523110106

14. Gorobets, A.V., Soukov, S.A., Magomedov, A.R.: Heterogeneous parallel implementation of a multigrid method with full approximation in the NOISETTE code. Matem. Mod. 36(2), 129–146 (2024). https://doi.org/10.20948/mm-2024-02-08

15. Gropp, W., Lusk, E., Skjellum, A.: Using MPI (2nd ed.): portable parallel programming with the message-passing interface. MIT Press, Cambridge, MA, USA (1999)

16. Gyles, B.R., Haegland, B., Dahl, T.B., *et al.*: Natural Convection - Subsea Cooling: Theory, Simulations, Experiments and Design. International Conference on Offshore Mechanics and Arctic Engineering, vol. Volume 1: Offshore Technology; Polar and Arctic Sciences and Technology, pp. 11–20 (Jun 2011). https://doi.org/10.1115/OMAE2011-49030

17. Ivanov, N., Ris, V., Tschur, N., Zasimova, M.: Numerical simulation of conjugate heat transfer in a tube bank of a subsea cooler based on buoyancy effects. J. Phys. Conf. Ser. 745, 032058 (Sep 2016). https://doi.org/10.1088/1742-6596/745/3/032058

18. Ivanov, N., Ris, V.V., Tschur, N.A., Zasimova, M.: Effect of gas flow direction on passive subsea cooler effectiveness. Comput. Therm. Sci. Int. J. 11(1-2), 1–16 (2019). https://doi.org/10.1615/COMPUTTHERMALSCIEN.2018024704

19. Jan, Y.J., Sheu, T.W.H.: A quasi-implicit time advancing scheme for unsteady incompressible flow. Part I: Validation. Computer Methods in Applied Mechanics and Engineering 196(45), 4755–4770 (2007). https://doi.org/10.1016/j.cma.2007.06.008

20. Jung, J., Perrier, V.: Steady low Mach number flows: Identification of the spurious mode and filtering method. Journal of Computational Physics 468, 111462 (2022). https://doi.org/10.1016/j.jcp.2022.111462

21. Kim, J., Moin, P.: Application of a fractional-step method to incompressible Navier–Stokes equations. Journal of Computational Physics 59(2), 308–323 (1985). https://doi.org/10.1016/0021-9991(85)90148-2

22. Kolesnik, E.V., Smirnov, E.M.: Supersonic laminar flow past a blunt fin: Duality of the numerical solution. Tech. Phys. 66(6), 741–748 (Jun 2021). https://doi.org/10.1134/S1063784221050133

23. Kolesnik, E.V., Smirnov, E.M.: Duality of the stream pattern of supersonic viscous gas flow past a blunt-fin junction: The effect of a low sweep angle. Fluid Dyn. 58(1), 1–8 (Feb 2023). https://doi.org/10.1134/S0015462822601887

24. Kolesnik, E.V., Smirnovsky, A.A., Smirnov, E.M.: Compressibility effect on heat transfer intensified by horseshoe vortex structures in turbulent flow past a blunt-body and plate junction. Journal of Physics: Conference Series 1565(1), 012104 (jun 2020). https://doi.org/10.1088/1742-6596/1565/1/012104

25. Kolesnik, E., Smirnov, E., Babich, E.: Dual numerical solution for 3D supersonic laminar flow past a blunt-fin junction: Change in temperature ratio as a method of flow control. Fluids 8(5), 149 (May 2023)

26. Koobus, B., Lallemand, M.H., Dervieux, A.: Unstructured volume-agglomeration MG: Solution of the Poisson equation. International Journal for Numerical Methods in Fluids 18(1), 27–42 (1994). https://doi.org/10.1002/fld.1650180103

27. Lasalle, D., Karypis, G.: Multi-threaded graph partitioning. In: 2013 IEEE 27th International Symposium on Parallel and Distributed Processing. pp. 225–236 (2013). https://doi.org/10.1109/IPDPS.2013.50

28. Le Touze, C., Murrone, A., Guillard, H.: Multislope MUSCL method for general unstructured meshes. Journal of Computational Physics 284, 389–418 (2015). https://doi.org/10.1016/j.jcp.2014.12.032

29. Lien, F.S., Ji, H., Ryan, S.D., *et al.*: A coupled multigrid solver with wall functions for three-dimensional turbulent flows over urban-like obstacles. International Journal for Numerical Methods in Fluids 95(9), 1349–1371 (2023). https://doi.org/10.1002/fld.5189

30. Liu, Y.: An efficient numerical method for highly loaded transonic cascade flow. J. Eng. Math. 60(1), 115–124 (Jan 2008)

31. Mani, M., Dorgan, A.J.: A perspective on the state of aerospace computational fluid dynamics technology. Annual Review of Fluid Mechanics 55, 431–457 (2023). https://doi.org/10.1146/annurev-fluid-120720-124800

32. Marmignon, C., Cantaloube, B., Le Pape, M.C., *et al.*: Development of an agglomeration multigrid technique in the hybrid solver elsA-H. In: Seventh International Conference on Computational Fluid Dynamics (ICCFD7). p. 15 (2012)

33. Matsui, K.: A projection method for Navier–Stokes equations with a boundary condition including the total pressure. Numerische Mathematik 152(3), 663–699 (2022). https://doi.org/10.1007/s00211-022-01323-x

34. Mavriplis, D.: Multigrid techniques for unstructured meshes. ICASE Report (1995)

35. Nishikawa, H., Diskin, B., Thomas, J.L.: Critical study of agglomerated multigrid methods for diffusion. AIAA J. 48(4), 839–847 (Apr 2010). https://doi.org/10.2514/1.J050055

36. Patankar, S.V., Spalding, D.B.: A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. International Journal of Heat and Mass Transfer 15(10), 1787–1806 (1972). https://doi.org/10.1016/0017-9310(72)90054-3

37. Patel, A.: Development of an adaptive RANS solver for unstructured hexahedral meshes. PhD thesis, Université libre de Bruxelles, Brussels, Belgium (2003)

38. Quirk, J.J.: A contribution to the great Riemann solver debate. Int. J. Numer. Methods Fluids 18(6), 555–574 (Mar 1994). https://doi.org/10.1002/fld.1650180603

39. Rodionov, A.V.: Artificial viscosity to cure the carbuncle phenomenon: The three-dimensional case. J. Comput. Phys. 361, 50–55 (May 2018). https://doi.org/10.1016/j.jcp.2018.02.001

40. Rogers, S.E., Kwak, D.: Upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations. AIAA Journal 28(2), 253–262 (1990). https://doi.org/10.2514/3.10382

41. Saad, Y.: Iterative methods for sparse linear systems. SIAM, Philadelphia, MS, 2 edn. (2003)

42. Smirnov, S.I., Abramov, A.G., Smirnov, E.M.: Numerical simulation of turbulent Rayleigh–Bénard mercury convection in a circular cylinder with introducing small deviations from the axisymmetric formulation. J. Phys. Conf. Ser. 1359(1), 012077 (Nov 2019). https://doi.org/10.1088/1742-6596/1359/1/012077

43. Smirnov, S., Smirnov, E.: Direct numerical simulation of the turbulent Rayleigh–Bénard convection in a slightly tilted cylindrical container. St. Petersburg State Polytechnical University Journal. Physics and Mathematics 13, 13–23 (2020). https://doi.org/10.18721/JPM.13102

44. Smirnov, S.I., Smirnov, E.M., Smirnovsky, A.A.: Endwall heat transfer effects on the turbulent mercury convection in a rotating cylinder. St Petersbg. Polytech. Univ. J. Phys. Math. 3(2), 83–94 (Jun 2017). https://doi.org/10.1016/j.spjpm.2017.05.009

45. Toro, E.F.: Riemann solvers and numerical methods for fluid dynamics. Springer, Berlin, Germany, 3 edn. (Apr 2009)

46. Turkel, E.: Preconditioned methods for solving the incompressible and low speed compressible equations. Journal of Computational Physics 72(2), 277–298 (1987). https://doi.org/10.1016/0021-9991(87)90084-2

47. Tutty, O.R., Roberts, G.T., Schuricht, P.H.: High-speed laminar flow past a fin–body junction. Journal of Fluid Mechanics 737, 19–55 (2013). https://doi.org/10.1017/jfm.2013.541

48. Van Doormaal, J.P., Raithby, G.D.: Enhancements of the SIMPLE method for predicting incompressible fluid flows. Numer. Heat Trans. 7(2), 147–163 (Apr 1984). https://doi.org/10.1080/01495728408961817

49. Weiss, J.M., Smith, W.A.: Preconditioning applied to variable and constant density flows. AIAA Journal 33(11), 2050–2057 (1995). https://doi.org/10.2514/3.12946