MATHEMATICA MONTISNIGRI
Vol XLII (2018)

ON THE CONSTRUCTION OF THE GENERALIZED NUMERICAL EXPERIMENT IN FLUID DYNAMICS

A.E. BONDAREV[1]

[1] Keldysh Institute of Applied Mathematics Russian Academy of Sciences
Miusskaya sq. 4, 125047 Moscow, Russia
e-mail: bond@keldysh.ru

Summary. The paper considers the construction of a generalized numerical experiment for problems of computational fluid dynamics (CFD). Generalized numerical experiment allows to obtain a solution not for one specific mathematical modeling problem, but for a class of problems defined in the multidimensional space of defining parameters. The basis of this approach is the use of parallel computing for the organization of multitasking. We consider the construction of interfaces for the organization of calculations, processing and analysis of the results. Some examples are given illustrating the application of the approach for constructing such an experiment for various classes of CFD problems in computational gas dynamics. Also examples of such experiment application to problems of analyzing the accuracy of numerical methods and the effectiveness of parallelization tools are considered.

1 INTRODUCTION

Long before the advent of the computer age, the main source of information in the problems of gas dynamics was a physical experiment. It was the experiment that made it possible to obtain the necessary information about flows and their properties, to obtain a visual representation of the flow pattern, and to obtain the relationships between gas-dynamic quantities characteristic of this picture. The results of such experiments are extensively presented in papers [1, 2]. However, in practical applications it was always not enough simply to obtain in the experiment the flow field for some single case. The main goal of a physical experiment has always been not the modeling of the physical phenomenon itself, but the elucidation of the circumstances under which it occurs, i.e. obtaining the dependence of the appearance of the phenomenon on the defining parameters of the problem, such as Mach numbers, Reynolds, Prandtl numbers, etc., and geometric parameters of the problem. In fact, the establishment of such physical laws for shock waves, separated flows, characteristic configurations of streamlined bodies was the main task of fluid and gas mechanics. Accordingly, it was necessary to carry out a series of physical experiments where the determining parameters of the flow varied, such as the velocity, viscosity, properties of the medium, etc. Such large-scale experimental work made it possible to obtain key relationships for the dependence of the gasdynamic functions of interest or the conditions for the appearance of a physical effect on the key determining parameters.

As a striking example of such a dependence, one can cite the famous formula of G.I. Petrov, representing the fundamental law on the ultimate pressure drop in the shock, which the turbulent boundary layer is able to withstand without detachment from the wall [3]:

2010 Mathematics Subject Classification: 49Q99, 76M27.
Key words and Phrases: generalized numerical experiment, parallel computations, parametric studies

$$P_2 / P_1 = 0.713M_e + 0.213$$

Here P_2 / P_1 is the pressure drop, M_e is the Mach number before the separation point, varying from 1.5 to 4.

Another example is the famous Kozlov formula [4], which represents the dependence of surface friction on Mach numbers, Reynolds numbers and the temperature factor:

$$c_{fw} = 0{,}085 Re_w^{-0{,}29+0{,}01\lg Re_w} \bar{T}_{we}^{0{,}39} \bar{T}_e^{0{,}2}$$

Here c_{fw}, Re_w is the coefficient of surface friction and the Reynolds number calculated with reference to the wall temperature, T_e is the temperature at the outer boundary of the boundary layer, and is the temperature factor.

The advent of computer technology allowed solving the problems of mathematical modeling of currents, which sharply reduced the need for large-scale physical experiments. However, in the problems of mathematical modeling, the main tendency of carrying out series of calculations with the variation of the defining parameters of the problem also remained. The main goal was the same - to determine the conditions for the appearance of a physical phenomenon when the external conditions of the problem are varied. The determination of such conditions and their approximation with the aid of an analytical expression was the main goal of practical computational fluid dynamics. There are many examples of such studies. Here we give an author's example [5], which presents a series of numerical experiments on the flow of a backward ledge by a viscous gas flow. As a result of the experiments, a generalized formula is obtained that represents the characteristic time of the establishment of the flow as a function of the Mach and Reynolds numbers of the external flow:

$$\tau^* = 10{,}7[1 - 0{,}14(M_\infty - 2)]\left[1 + 19{,}45\left(\frac{1}{\sqrt{Re}} - \frac{1}{\sqrt{10^3}}\right)\right]$$

Here τ^* is the characteristic settling time, M_∞ , Re are the Mach and Reynolds numbers. The generalized formula is given for the range $2 \le M_\infty \le 3$, $1000 \le Re \le 5000$.

Nevertheless, obtaining such dependencies required a huge number of computational experiments and was very laborious.

The emergence of high-performance computing that allows for parallel computations has ensured the possibility of parallel calculation of the same problem with different input data in multitask mode. This makes it possible at the present time to construct and carry out *a generalized computational experiment*.

2 FORMULATION OF THE PROBLEM

A generalized computing experiment involves splitting each of the defining parameters of a problem within a certain range. Thus, a grid decomposition is formed for some multidimensional parallelepiped composed of the defining parameters of the considered

problem of gas dynamics. For each point of this grid, the problem is calculated in the space of the determining parameters. Formally, this can be written as follows.

Suppose that there is a reliable numerical method for solving two-dimensional and three-dimensional nonstationary problems of computational gas dynamics. Then we can obtain a numerical solution $F(x, y, z, t, A_1, \dots, A_N)$ for any point in the space of a computational domain, where *x, y, z* are the spatial coordinates, *t* is the time, $A_1, \dots, A_N$ are the defining parameters of the problem. As defining parameters of the problem, we will keep in mind the characteristic numbers describing the properties of the flow under consideration, such as the Mach numbers, Reynolds, Prandtl, Strouhal, etc., and the characteristic geometric parameters. Each of the characteristic parameters is limited in a certain range:

$$A_i^{min} \le A_i \le A_i^{max}, \quad i = 1, \dots, N$$

We divide each of the parameters A_i into *k*-1 parts, so we obtain for each parameter a partition consisting of *k* points. The volume of an *N*-dimensional space formed by a set of defining parameters A_i is filled with a set of k^N points.

Denoting the point from the given set, as $(A_1^*, \dots, A_N^*)$, we arrive at the fact that for each point of the collection it is necessary to obtain a numerical solution of the gas-dynamic problem $F(, x, y, z, t, A_1^*, \dots, A_N^*)$.

It is easy to see that this will require solving k^N gasdynamic problems, which is impossible without the use of parallel calculations in a multitask mode. In practice, the number *N* usually does not exceed 5, which corresponds to the computing capabilities at the current time.

It should also be noted that we formulated the classical problem of parametric study. Parametric numerical studies allow one to obtain a solution not for one particular mathematical modeling problem, but for a class of problems defined in a multidimensional space of defining parameters. Also, such a formal formulation allows numerical study of optimization analysis problems, when the inverse problem is solved at each point of the grid partition of the multidimensional space of the determining parameters. Both types of similar problems are considered in a series of papers [6 - 9].

3 PARALLEL SOLUTIONS

The only way to effectively carry out a generalized numerical experiment is applying of parallel computations. The problem of the optimal and effective way of parallelization was thoroughly discussed in the papers [7,8]. There were considered parts of the whole algorithm for parameter optimization and analysis. For these parts the main criterion of applicability for parallelizing is independence of specific numerical method. From this point of view the most perspective way for parallelizing is applying the approach of multitask parallelism using the principle “one task – one process”. Due to minimal quantity of internal exchanges between the processes we are able to create an effective practical tool for generalized numerical experiment.

The general parallel computing scheme used for such experiment is shown in Figure 1.

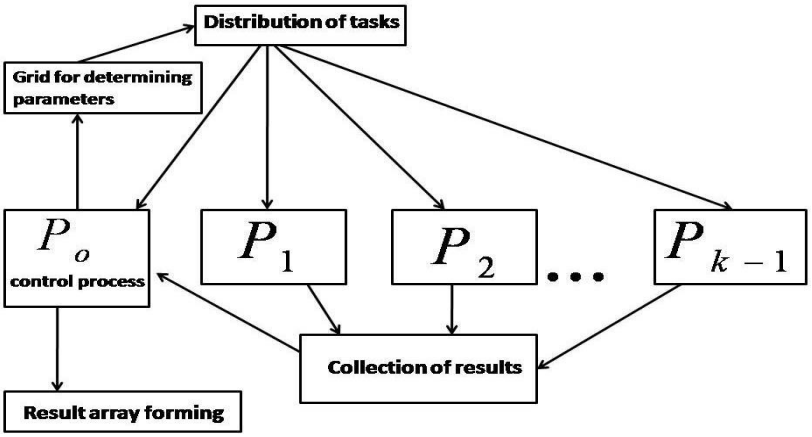

Figure 1: Parallel computing scheme for generalized numerical experiment.

We assume that k processes are provided for parallel computation. The control process P_0 creates the grid in the multidimensional space of determining parameters, then P_0 forms tasks and sends the tasks to others processes and to itself also. After task completion P_0 collects the results and implements all procedures defined by user, such as data processing and transformation.

Due to the absence of internal exchanges between the processes the procedure of parallelizing amounts to creation of control interface for tasks distribution and data collecting in one multidimensional array.

There are two effective and easy ways to create such interface for parallel computations. The first way is to apply MPI (Message Passing Interface) [10]. This variant of parallelizing allows implementing a program tool for generalized numerical experiment. The computation can be carried out k times faster according to the number of provided processes.

The other way of parallelization is application of DVM technology [11, 12], elaborated in Keldysh Institute of Applied Mathematics RAS. DVM-system provides unified toolkit to develop parallel programs of scientific-technical calculations in C and Fortran. DVM parallel model is based on data parallel model. The DVM name reflects two names of the model - Distributed Virtual Memory and Distributed Virtual Machine. These two names show that DVM model is adopted both for shared memory systems and for distributed memory systems. DVM high level model allows not only to decrease cost of parallel program development but provides unified formalized base for supporting Run-Time System, debugging, performance analyzing and prediction. Unified parallel model is built in C and Fortran languages on the base of the constructions, that are "transparent" for standard compilers, that allows to have single version of the program for sequential and parallel execution. C-DVM and Fortran DVM compilers translate DVM-program in C or Fortran program correspondingly, including parallel execution Run-Time Support system calls. So only requirement to a parallel system is availability of C and Fortran compilers. This way of code parallelizing allows one to save a lot of human resources for coding and debugging. At the same time DVM parallelization provides less speed of computations in comparison with MPI.

For both types of parallel technologies special control interfaces for parameter optimization and analysis were designed [8]. Both control interfaces were applied to jet

interaction problem for testing. Testing computations were carried out for 20 processors. According to test results the time of computations for DVM method is 205 seconds. The same test for MPI case requires 144 seconds. At the same time DVM application allowed to decrease human expenses for coding and debugging up to ten times as against MPI. So both types of parallel technologies are quite applicable for problems in question. The calculations were performed using a hybrid supercomputer K-100 [13] in Keldysh Institute of Applied Mathematics RAS.

With the help of the constructed interfaces, a series of calculations were carried out, realizing the concept of a generalized numerical experiment for various classes of problems. The results of the calculations will be shown in the following sections. Both developed interfaces are very versatile. They can be applied to almost any software code for solving the CFD problem chosen as the base one.

It should be noted that modern version of Aiwlib library [14] has some similar properties which make this library applicable for construction of generalized numerical experiment. Aiwlib library is a library for C++11 and Python languages, which is aimed for the development of high-performance computing numerical simulation applications running under GNU/Linus OS. It also provides means for batch calculations.

4 PROCESSING AND ANALYSIS OF THE RESULTS

As a result of implementing the construction of a generalized numerical experiment and performing parallel calculations, we obtain a large data set representing a set of numerical solutions $F(x, y, z, t, A_1, \dots, A_N)$ for each point $(A_1^*, \dots, A_N^*)$ of the partition of the multidimensional volume of the defining parameters $(A_1, \dots, A_N)$ of the problem under consideration. This volume in its original form is rather difficult to use, although its availability for further purposes is necessary. As a rule, when studying the conditions for the appearance of a physical effect (for example, the emergence of a space-time structure), the object of primary interest is not gas-dynamic fields, but certain markers or objective functions determined with the help of these fields. The creation of such markers and goal functions is an extensive topic that deserves a separate discussion and is beyond the scope of this article. Suppose that some function P that plays the role of a marker is defined and can be calculated for each solution $F(x, y, z, t, A_1, \dots, A_N)$ obtained from the calculated fields of gas-dynamic quantities. Then the main task is to analyze the multidimensional array $P(A_1, \dots, A_N)$. This problem is considerably covered by the fact that, as indicated above, in practice the number N usually does not exceed 5. Practical approaches to solving problems of this kind are described in [8, 9, 15]. As the basic method of solution, the following is indicated: decreasing the dimensionality of the investigated array to 3, visual representation of the new array, approximation of the dependence using geometric primitives of the first or second order.

There are some ways to decrease the array dimensionality. These ways are well known from the group of methods for multidimensional data processing and analysis. Being frank we should note the fact that most of these methods were used for a long time before computers appearance. This field of science was known as “experimental data processing”.

The first way is the analysis of variances for each characteristic parameter. Characteristic parameter is considered as coordinate direction. Data variances D_1, D_2, ... D_N are computed

along the each direction. Then the variances should be arranged. The direction with minimal variance $D_{\min} = \min\{D_i\}$ is rejected. This procedure sometimes is called as compactification.

More radical kind of compactification can be implemented as follows. After variances computing and arranging one chooses three directions with maximal variances. If other variances are much less than this triplet one changes the data for directions corresponding to other variances by means. After such decreasing of dimensionality one can operate in standard 3D space. This approach has one disadvantage - it does not work if multidimensional data are close to hypersphere. Nevertheless for many practical cases with small dimensionality (4 or 5) the approach works well enough.

Another way is the construction of different 3D data projections for various triplets of determining parameters. If the data on projections for some direction are close to constant then this direction can be rejected.

Also, to reduce the dimension of a multidimensional array, methods of mapping into embedded manifolds of smaller dimension are very effective [16, 17]. Among them, the most common method is the principal component method (PCA). The essence of the method consists in the transition from the initial coordinate system to the new orthogonal basis in the multidimensional space under consideration, whose axes are oriented along the directions of maximum dispersion. The possible scheme of working with an array in this case is the approximation by primitives of the data array in the space of the first three main components and the subsequent transition to the initial space of the determining parameters.

Among the modern methods of reducing the dimension that are common in solving the problems of computational gas dynamics, it is necessary to mention the POD-method (Proper Orthogonal Decomposition) [18]. It involves the creation of a functional orthogonal basis of a smaller dimension, to which the multidimensional dynamic volume under investigation is "pulled on". In this formulation, the eigenvectors of the covariance matrix serve as the vectors of the desired basis. Also, in recent years, dynamical mode decomposition (DMD) has become increasingly popular in the analysis of flows [19, 20]. The dynamic mode decomposition method (DMD) is an algorithm for searching for an evolution operator (inverse operator problem solutions) in a finite-dimensional space of solution of a problem (numerical or experimentally obtained) in a set of solutions (slices, "snapshots") at some successive instants.

5 EXAMPLES OF THE GENERALIZED NUMERICAL EXPERIMENT

This section contains the examples of the proposed above approach applied to some practical problems. It is applied in some variations due to different aims for each class of problems.

The first example is the problem of unsteady interaction of the supersonic viscous flow with jet obstacle [7]. Figure 2 illustrates the example. The obstacle appears due to co-current underexpanded jet exhausting from the nozzle. The nozzle is placed to external supersonic viscous flow. Expanding jet propagates on the external surface of the nozzle creating obstacle in external flowfield. Typical flow structure is shown in Figure 2 (a) by streamlines. Time-dependent control action (the velocity of pressure ratio growth in underexpanded jet) allows to change time-space structure of flowfield (Figure 2 (b)). New space-time structure presents

specific flow regime where jet propagates upwind on the external wall of the nozzle. We consider crucial velocity of jet pressure ratio growth as control parameter. The main target of research is estimating and defining the control parameter dependence on four characteristic parameters of the problem – Mach, Reynolds, Prandtl and Strouhal numbers. These parameters are varied in definite ranges creating four-dimensional space. We want to find for each point in this space the crucial velocity corresponding to a new time-space structure appearance. According to the scheme presented in the previous chapters parallel algorithm is implemented for computations. For the space of determining parameters two types of grids are chosen: 5 and 10 points for each determining parameter. It requires computing 625 and 10000 problems. Both MPI and DVM technologies were applied to control parallel computations.

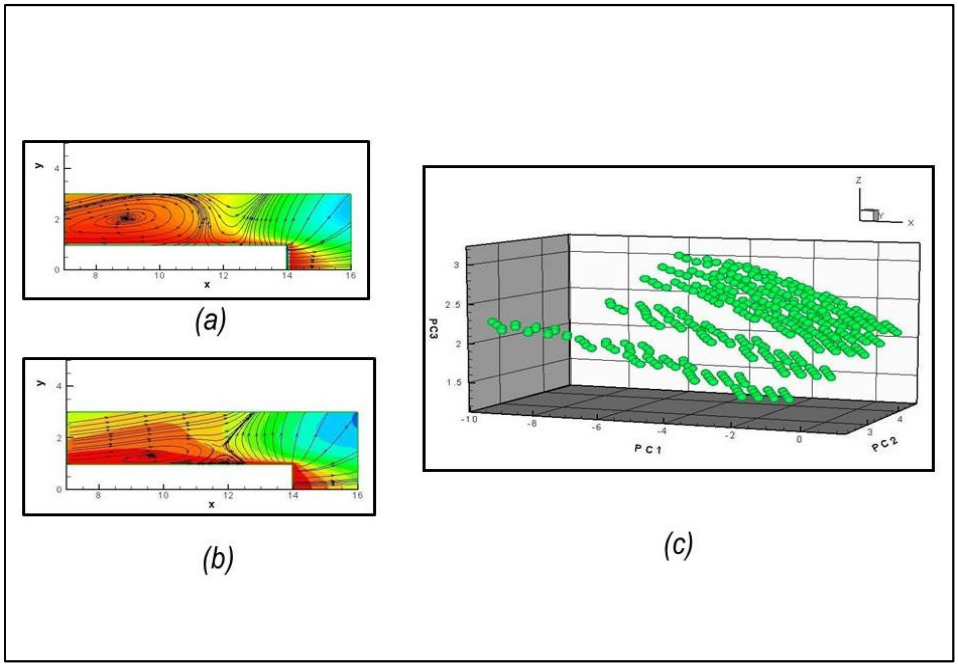

Figure 2: Parameter optimization and analysis applied to jets interaction

As a result of approach application five-dimensional data array is obtained, where variables are four characteristic parameters M_∞, Re_∞, Pr, Sh_∞ and crucial velocity V^*. For obtained data three principal components are defined and we construct data visual presentation in principal components (Figure 2 (c)). The presentation allows us to suppose that the points of data volume can be roughly approximated by parametric plane. After defining the coefficients for plane and inverse transformation to the original variables we obtain the sought-for dependence $V^*= F(M_\infty, Re_\infty, Pr, Sh_\infty)$ in analytical form. The dimensionality of four-dimensional array under consideration can be decreased up to three, because characteristic parameter Re_∞ has a very small influence on the solution. So the final form for V^* cam be written as follows:

$$V^*= - 0.1\, M_\infty + 0.115\, Pr + 0.24\, Sh_\infty$$

Obtained results present a result of generalized numerical experiment for the class of problems, where the class is defined by multidimensional volume of characteristic parameters.

The second example of application of general numerical experiment is devoted to optimization problem. The example presents a search the optimal shape of a power plant three-dimensional blade assembly [21]. This experiment is based on developed computational technology for the computation of power loads on the 3D blade assembly of a power plant in a wind flow. The calculation for various combinations of the key geometric parameters of the assembly using parallel computations makes it possible to find the optimal shape of the assembly with respect to its power characteristics. A virtual experimental facility for simulating the flow around the power plant based on the solution of the Navier–Stokes equations was created. Computations aimed at determining the optimal shape of the blade assembly taking into account constraints on its design were carried out, and the results were thoroughly analyzed using the proposed optimization procedure. The solution of the optimization problem is based on the parameterization of the design using three key parameters. On the discrete set of values of these parameters, the maximums of two objective functions—the magnitude of the total aerodynamic force and the magnitude of the rotation torque—determining the lift-to-drag ratio of the power plant are found. Figure 3 presents the shape of 3D blade assembly and pressure distribution on its surface.

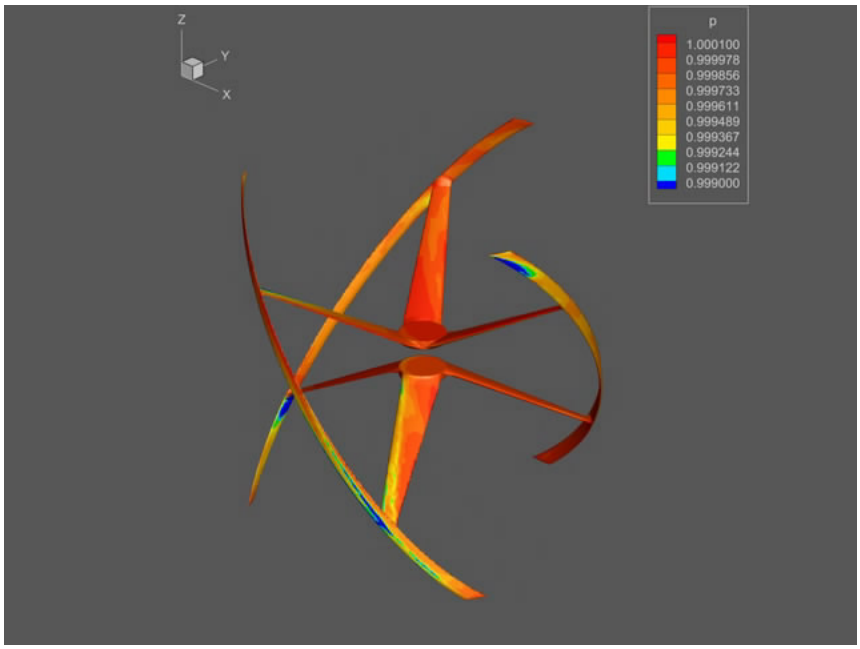

Figure 3: Pressure distribution on the surface of 3D blade assembly.

The next example is addressed to the problem of the evaluation of the accuracy for different numerical methods. The problem of inviscid compressible flow around a cone at zero angle of attack is used as a base one. The results obtained with the help of various OpenFOAM solvers are compared with the known numerical solution of the problem with the variation of cone angle and flow velocity [22]. Cone angle β changes from 10° to 35° in steps

of 5°. Mach number varies from 2 to 7. The scheme of a flow around a cone is presented in Fig. 4. Here angle β is a half of cone angle as shown in Fig. 4. For comparison, four solvers were selected from the OpenFOAM software package: *RhoCentralFoam, SonicFoam, RhoPimpleFoam, RhoPimpleFoam.* The results of such kind of numericsl experiment were presented as errors in the form of an analog of the L2 norm for all solvers. Fig.5 illustrates the results in a form of a change in deviation from the exact solution for pressure depending on the cone angle and the velocity for the solver *rhoCentralFoam*. Such changes were obtained for all solvers.

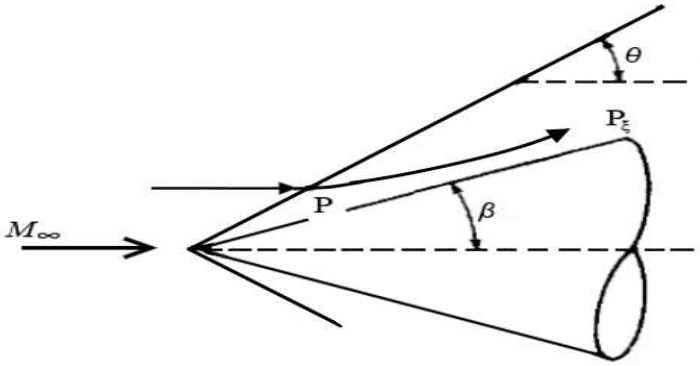

Figure 4: Scheme of a flow around a cone.

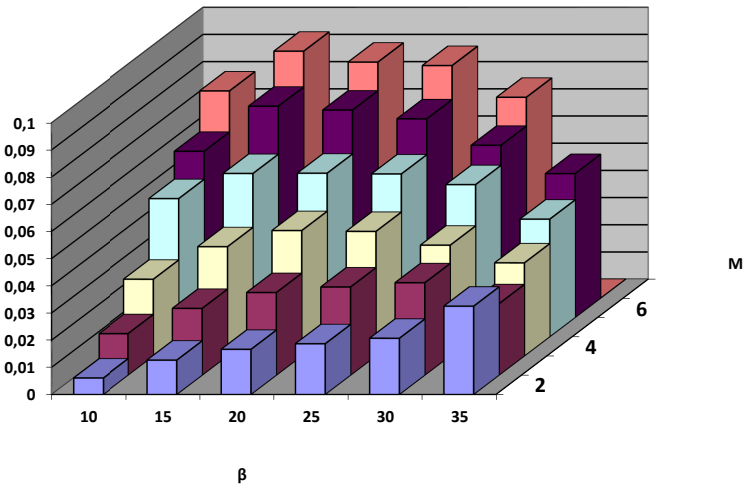

Figure 5: Change in deviation from the exact solution for pressure depending on the cone angle and the velocity for the solver *rhoCentralFoam*.

This methodical research can serve as a basis for selecting the OpenFoam solver for calculating the inviscid supersonic flow around the elongated bodies of rotation. The results of solvers comparison can also be useful for developers of OpenFoam software content. The results obtained made it possible to get a general idea of the calculation errors for all solvers.

Another one example of generalized numerical experiment is devoted to the problem of tuning the properties of hybrid finite-difference schemes [23]. The paper [23] contains the description of developed program tool Burgers2. This program tool is intended for tuning and optimization of computational properties for hybrid finite-difference schemes applied to Burgers equation. One-dimensional model Burgers equation describes propagation of disturbances for dissipative medium. The equation has exact solution, so it is widely used for tuning-up of computational tools. Described program tool is based on combining of optimization problem solution and visual data presentation. Visual presentations of maximal error surface and error function are implemented as program tool features. User is able to visualize error function distribution for any chosen moment of time. These visual presentations allow analyzing and control computational properties of hybrid finite-difference schemes under consideration. Users have possibility of creating hybrid finite-difference schemes and analyzing computational properties for chosen grid template provided by program tool. Visual presentation of optimization problem solution allows finding of suitable weight coefficients for hybrid finite-difference scheme under consideration. The user can make the calculations simultaneously different sets of weight coefficients in accordance with the concept of generalized numerical experiment. Figure 6 presents the surface of absolute error for one of the hybrid scheme variants. The negative data area indicates where the oscillations occur.

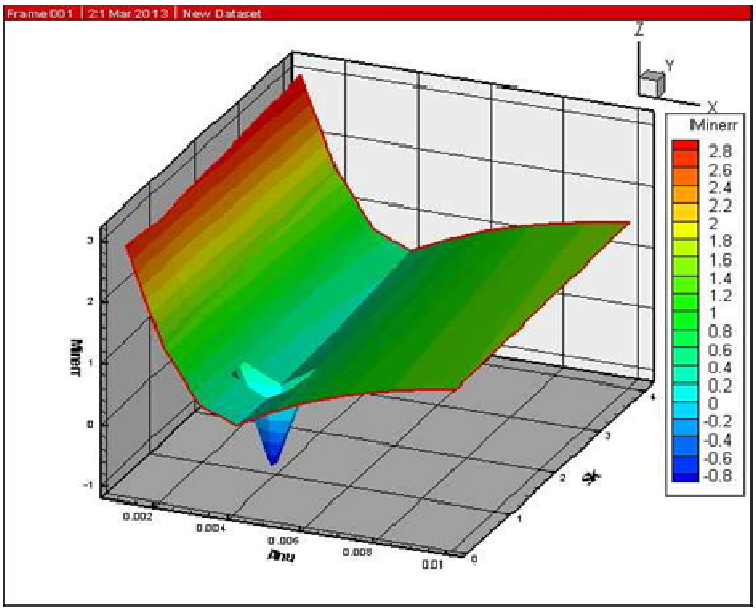

Figure 6: Surface of absolute error.

The next example is aimed to evaluation of parallelization effectiveness. Here the problem of Burgers equation from previous example is used as a base one. To solve this problem, we used an implicit finite difference scheme, described in detail in [23]. When solving the parametric problem, the viscosity coefficient and weight coefficient of the difference scheme split in certain ranges, and for each pair of values the problem described above was solved. During the experiments on parallelization of this program code with DVM, the following

parameters varied: N is the number of MPI processes, PPN is the number of MPI processes per compute node. In the conducted experiments, the number of MPI processes N varied from 1 to 32, and the number of MPI processes per one PPN computing node ranged from 1 to 8. The results are presented in Table 1 below. In the table, the following notation is used: N is the number of MPI processes, PPN is the number of MPI processes running on one node, T is the time in seconds, S is the acceleration T/Tserial, E is the efficiency of parallelization.

N	PPN	T	S	E
serial	------	51,4	1	1
1	1	51,4	1,001	1,001
2	1	25,7	1,997	0,998
2	2	25,7	1,998	0,999
4	1	12,9	3,988	0,997
4	2	12,9	3,987	0,997
4	4	12,9	3,991	0,998
8	1	7,61	6,753	0,844
8	2	6,71	7,662	0,958
8	4	6,73	7,641	0,955
8	8	6,98	7,363	0,920
16	2	4,09	12,580	0,786
16	4	3,64	14,129	0,883
16	8	3,76	13,652	0,853
32	2	2,08	24,684	0,769
32	4	2,09	24,613	0,769
32	8	2,16	23,782	0,743

Table 1 : Burgers equation, implicit scheme.

The obtained results of calculations allow to estimate the effectiveness of the implemented solution for multi-tasking parallelization.

The examples show applicability of presented approach for a wide range of practical applications, so the approach can be considered as quite universal one.

6 CONCLUSIONS

The concept of generalized computing experiment presented in the article has a wide range of possible applications. First of all, for the problems of computational fluid dynamics such an approach makes it possible to obtain a solution not only for one, separately taken, problem, but for a whole class of problems defined in a certain range of the complex of determining parameters. A generalized computing experiment involves splitting each of the defining parameters of a problem within a certain range. Thus, a grid decomposition is formed for some multidimensional parallelepiped composed of the defining parameters of the considered problem of gas dynamics. For each point of this grid, the problem is calculated in the space of

the determining parameters. Practical implementation of the approach becomes possible with the use of parallel calculations in multitask mode. The results of calculations are multidimensional volumes of data that can be processed using data analysis tools and visual analytics It should be noted that the application of the approach makes it possible to conduct exploratory calculations on coarse grids for a class of problems with subsequent refinement for sets of determining parameters of special interest.

Acknowledgements: Author acknowledges the support by grants of RFBR № 16-01-00553A and 17-01-444A.

REFERENCES

[1] M. D. Van Dyke, *Album of Fluid Motion* (Parabolic, California, 1982; Mir, Moscow, 1986).
[2] T. J. Mueller, "Flow Visualization by Direct Injection," in *Fluid Mechanics Measurements* (Hemisphere, 1983), pp. 307–375.
[3] E.N. Bondarev, V.T. Dubasov, Y.A. Ryzhov et al., *Aerigidromeckanika.* M.: Mashinostroenie, (1993).
[4] L.V. Kozlov, "Experimental investigation of surface friction on a flat plate in a supersonic flow in the presence of heat transfer". *Izvestiya AN SSSR, OTN, Mekhanika i mashinostroenie*, 2, (1963).
[5] A.E. Bondarev, "Effect of supersonic stream parameters on the characteristic time of transient step flow". *Izvestiya AN SSSR, Mekhanika zhidkosti I gaza*, 4, 137-140 (1989).
[6] A.E. Bondarev, "Analysis of Space-Time Flow Structures by Optimization and Visualization Methods", *Transactions on Computational Science XIX, LNCS* 7870, 158-168 (2013).
[7] A.E. Bondarev and V.A. Galaktionov, "Parametric Optimizing Analysis of Unsteady Structures and Visualization of Multidimensional Data", *International Journal of Modeling, Simulation and Scientific Computing*, V.04, supp01, (2013) DOI 10.1142/S1793962313410043.
[8] A.E. Bondarev and V.A. Galaktionov, "Analysis of Space-Time Structures Appearance for Non-Stationary CFD Problems", *Procedia Computer Science*, 51, 1801–1810 (2015).
[9] A.E. Bondarev, "Design and analysis of multidimensional parametrical solutions for time-dependent CFD problems", *Mathematica Montisnigri*, XXXIII, 58-68 (2015).
[10] P. Pacheco, *Programming Parallel with MPI.* San Francisco, CA: Morgan Kaufmann, 1997.
[11] DVM-system. [Online]. Available: http://dvm-system.org/
[12] V.A. Bakhtin, B.N. Chetverushkin, V.A. Kryukov, E.V. Shil'nikov, "Extension of the DVM parallel programming model for clusters with heterogeneous nodes", *Doklady Mathematics*, 84 (3), 879-881 (2013).
[13] Hybrid supercomputer K-100. [Online]. Available: http://www.kiam.ru/MVS/resourses/k100.html
[14] A. V. Ivanov and S. A. Khilkov, "Aiwlib library as the instrument for creating numerical modeling applications", *Scientific Visualization*, 10(1), 110 - 127 (2018). DOI: 10.26583/sv.10.1.09
[15] A. Bondarev, "Multidimensional Data Analysis in CFD Problems", *Scientific Visualization*, 6(5), 61-68 (2014).
[16] A. Gorban, B. Kegl, D. Wunsch, A. Zinovyev (Eds.), *Principal Manifolds for Data Visualisation and Dimension Reduction* (LNCSE 58 ed.). Berlin – Heidelberg – New York: Springer, (2007).
[17] A. Zinovyev, *Vizualization of multidimensional data,* Krasnoyarsk: Krasnoyarsk Univ., (2000).
[18] G. Kerschen, J.-C. Golinval, A. F. Vakakis, L. A. Bergman, "The Method of Proper Orthogonal Decomposition for Dynamica Characterization and Order Reduction of Mechanical Systems: An Overview", *Nonlinear Dynamics* 41, 147–169 (2005).

[19] P.J. Schmid, “Dynamic mode decomposition of numerical and experimental data”, *Journal of Fluid Mechanics* 656(1), 5-28 (2010).
[20] A. K. Alekseev, D. A Bistrian, A. E. Bondarev, I. M. Navon, “On Linear and Nonlinear Aspects of Dynamic Mode Decomposition”, *International Journal for Numerical Methods in Fluids*, 82(6), 348–371 (2016). DOI: 10.1002/fld.4221.
[21] Andreev S.V. et al., “A Computational Technology for Constructing the Optimal Shape of a Power Plant Blade Assembly Taking into Account Structural Constraints”, *Programming and Computer Software*, 43(6), 345-352 (2017). DOI: 10.1134/S0361768817060020
[22] Alexander E. Bondarev and Artem E. Kuvshinnikov, “Analysis of the Accuracy of OpenFOAM Solvers for the Problem of Supersonic Flow Around a Cone”, *Lecture Notes in Computer Science (LNCS)*, 10862, 221–230, (2018). DOI: 10.1007/978-3-319-93713-7_18
[23] A. E. Bondarev et al., “Design of program tool BURGERS2 for hybrid finite-difference schemes optimization and visualization”, *Scientific Visualization*, 5(1), 26-37 (2013).

Received May 15, 2018