

UDC 519.688

10.23947/2587-8999-2017-2-148-155

Efficient parallel software system for solving Navier-Stokes equations by the discontinuous Galerkin method***M.M. Krasnov, P.A. Kuchugov, M.E. Ladonkina, V.F. Tishkin****

Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), Moscow, Russian Federation
Lavrentyev Institute of Hydrodynamics, Novosibirsk, Russian Federation

Algorithms for solving the Navier-Stokes equations on a three-dimensional tetrahedral grid by the discontinuous Galerkin method were realized. Under the code development a new approach to programming problems of mathematical physics was used which allows one compactly write and effectively implement mathematical expressions in particular due to introduction of the concept of «grid operator» similar to mathematical one and uniformly realize algorithms for various grid types and computing architectures. The efficiency of this numerical code is investigated.

Keywords: Navier-Stokes equations, discontinuous Galerkin method, parallel programming, templated meta-programming

Introduction. At present, numerical methods of high accuracy must be used to solve a number of applied problems of mathematical physics. This is especially true for solving complex, multiscale problems, in which it is not enough to obtain a solution by grinding the mesh and methods of the first order of accuracy. An example of a method providing high accuracy is the Discontinuous Galerkin Method (DGM) [1], which has proved itself to solve a wide class of problems in mathematical physics. This method has a number of advantages inherent in both finite-element and finite-difference approximations. In particular, it provides a given order of accuracy, can be used for grids of arbitrary structure and is theoretically justified [1-5]. Discontinuous Galerkin Method, with all its merits, has a significant computational complexity, so the question of maximizing the use of all the possibilities of modern computer technology is very acute. The dynamics of the development of high-performance computing technology dictates the creation of software complexes that are relatively easily adaptable for work on various hybrid parallel architectures. For this purpose, when creating calculation modules, it is necessary to use new approaches to programming. When creating programs based on numerical methods of high accuracy, such as RMG, it is necessary to use the new capabilities of modern programming languages. Thus, with the advent of templates in new versions of the C ++ language, it became possible to reduce the amount of computation by transferring some of the computation to the compilation stage and, as a consequence, accelerating the computations. In addition, template metaprogramming makes it possible to simplify the algorithm due to its generalization [6].

* The research is done with the financial support from RSF Projects No. 16-11-10033.

** E-mail: kmm@kiam.ru, pkuchugov@gmail.com, ladonkina@imamod.ru, tishkin@mail.ru

1. **Structure of the program code DG3D.** Let's consider the three-dimensional Navier-Stokes equation, written in a strong conservative form and supplemented by suitable initial-boundary conditions.

$$\tau = \left(\lambda - \frac{2}{3} \mu \right) (\operatorname{div} v) + 2\mu S(v),$$

$$S(v) = \frac{\nabla v + (\nabla v)^*}{2}, \quad q(U) = -k \nabla T,$$

where U – the vector of conservative variables, $F(U)$, $G(U, \tau)$ – the functions of the inviscid and viscous flow, respectively, v – the velocity vector, μ – dynamic viscosity coefficient, λ – bulk viscosity, τ – the tensor of viscous stresses.

We assume that the liquid obeys arbitrary equations of state.

To apply the discontinuous Galerkin method, we cover the region Ω , on which the solution is sought by the tetrahedral grid T_h . On each element T_j the approximate solution of the system of equations (1) will be sought in the form of polynomials $P^{(x)}$ of degree P with time-dependent coefficients. In this paper, the Taylor basis is used as the basis functions.

As is customary in solving second-order equations by discontinuous Galerkin method, in this paper the Navier-Stokes equations are written in the form of a system of equations of the first order and the solution occurs in two stages [1, 7]. At the first stage, the components of the viscous stress tensor are computed, and the components of the temperature gradient vector, the approximation of which, like the approximation of the solution, within the grid cell is in the form of polynomials of degree P with time-dependent coefficients, the components of the vector of conservative variables are determined in the next step.

When solving this system, it is necessary to correctly determine the flow functions calculated at the element boundary. Inviscid flows can be calculated using various versions of the exact or approximate solution of the Riemann problem. In the developed software package, a library of software implementations of these approximations has been created [8-10]. At the moment, various discrete approximations are used for both viscous [7, 11, 12] and non-viscous flow terms.

To ensure the monotonicity of the solution obtained by the DGM, it is necessary to introduce so-called slope limiters, or limiters, in particular for cases when the solution contains strong discontinuities. In this paper we use the classical Cockburn limiter [1, 13-15], which is reliable in the work and is easily realized in the three-dimensional case on grids of arbitrary structure.

When creating a software package that implements the discontinuous Galerkin method on unstructured tetrahedral grids [16], a fundamentally new approach to programming problems of mathematical physics was used, which allows one to compactly write and effectively implement mathematical formulas, in particular, by introducing the notion of "grid operator" uniformly implement the approach on different types of grids and for various computing architectures. The grid-operator approach to programming is implemented as a class library for the C++ language. The main tool of the C++ language used in implementing this approach is template metaprogramming. This

tool implements a problem-oriented language (DSL) for manipulating grid functions. With the help of this language, grid expressions of any complexity can be built from grid functions, which can include arithmetic operations with brackets, as well as the application of grid operators to grid expressions. These grid expressions can be assigned to grid functions, and the calculations themselves in accordance with the given expression are started only at the moment of assignment. Prior to this, the chain of calculations is only remembered. The concept of so-called «deferred» calculations is realized. This allows, in particular, to get rid of additional arrays for storing intermediate results.

To implement the grid-operator approach to programming, in the IMM. M.V. Keldysh RAS, a gridmath library was created. This library implements grids, grid functions, grid calculators, arithmetic of calculated objects and grid operators, parallelization of calculations on shared memory using OpenMP or CUDA.

2. Analysis of the efficiency of the program code. According to Amdahl's law, the acceleration in the parallelization of computations by p processes is limited by a fraction of α from the volume of computations, which can be obtained only by consecutive calculations. The proportion of computations that can be paralleled ideally is in this case $1 - \alpha$ and can be accelerated by a factor of p . Then the acceleration that can be obtained on a computer system from p processes, in comparison with a single process solution, will not exceed the value

$$S_p = \frac{p}{\alpha + \frac{p-1}{p}}.$$

If the number α is small, then approximately

$$S_p \approx \frac{p}{\alpha p + 1}.$$

It follows from this formula that an ideal acceleration (in p times) can be obtained only if the fraction of successive calculations is zero. And, for example, if the proportion of consecutive calculations is half, for any number of processes, the acceleration can not be more than two. Thus, as p increases, the acceleration asymptotically approaches the number $1 / \alpha$.

In well-scalable programs, the number α is determined, first of all, by the exchange of boundaries between processes and depends on the ratio of the number of boundary cells to the total number of grid cells, but it is difficult to calculate in advance. But with a ready-made configuration of grid distribution by processes, it can be experimentally evaluated. If acceleration is known on p processes, then the number α can be estimated as

$$\alpha = \frac{p - S_p}{(p-1) \cdot S_p}.$$

Or, approximately, for $p \gg 1$.

After that, knowing α , we can estimate the acceleration for the other p , and, most importantly, understand to what extent this task is scalable and on what number of processes it makes sense to run it. Consider an additional acceleration when the number of processes is doubled (we use the approximate formula):

$$\frac{S_{2p}}{S_p} = \frac{2(\alpha p + 1)}{2\alpha p + 1} = 1 + \frac{1}{2\alpha p + 1}.$$

We choose as the maximum such p , at which the doubling of the number of processes accelerates the computation by no more than 20%:

$$2\alpha p + 1 = 5; \quad p = 2 / \alpha.$$

To estimate the effectiveness of the software complex, we consider the solution of the problem of flow past a rigid body by discontinuous Galerkin method on an irregular grid containing 3×10^6 cells.

Calculations were performed on K-100 and K-60 clusters. The K-100 cluster consists of 64 computational nodes, each node contains 2 Intel Xeon X5670 CPUs with 6 cores per processor, only 12 cores and 96 GB of RAM, and 3 nVidia Fermi C2050 graphics cards with 2.8 GB of operational memory on each. The cluster K-60 consists of 66 computational nodes. Each node is a dual-processor Intel Xeon E5-2690 v4 server with 256GB of RAM. The results of calculations on the K-60 cluster are given in Table 1, on the K-100 cluster in Table 2, p – the number of processes, t – the time in seconds, S_p = acceleration, $E_p = S_p 100 / p$ – efficiency, α – the fraction of consecutive calculations.

Table 1

p	t_1, c	S_{p1}	$E_{p1}, \%$	α_1	t_2, c	S_{p2}	$E_{p2}, \%$	α_2
1	1015,05	1,00			1012,68	1,00		
2	508,00	2,00	100	0,0009	504,00	2,01	100	-0,0046
4	264,13	3,84	96	0,0136	257,50	3,93	98	0,0057
8	149,49	6,79	85	0,0255	139,01	7,28	91	0,0140
16	83,80	12,11	75	0,0214	69,92	14,48	90	0,0070
32	54,69	18,56	58	0,0234	35,43	28,58	90	0,0039
64	42,95	23,63	37	0,0271	17,41	58,17	91	0,0016
128	41,61	24,39	20	0,0334	8,35	121,28	95	0,0004
256	49,73	20,41	8	0,0453	3,92	258,34	101	0,0000

Table 2

p	t_1, c	S_{p1}	$E_{p1}, \%$	α_1	t_2, c	S_{p2}	$E_{p2}, \%$	α_2
1	582,57				35,2			
2	302,85	1,9	96	0,0198	18,16	1,9	97	0,0159
4	150,6	3,9	97	0,0085	10,49	3,3	84	0,0480
8	80,61	7,2	90	0,0134	5,47	6,4	80	0,0304
12	55,55	10,5	87	0,0120	3,76	9,3	78	0,0235
16	42,16	13,8	86	0,0098	2,91	12,1	76	0,0202
32	22,51	25,9	80	0,0074	1,54	22,8	71	0,0125

In Table. 1 in the first series of calculations marked with the digit "1" took into account the total execution time of the program from the moment of initialization and recording the calculation results in a file. Since the results are written to a single file sequentially from all processes, the efficiency drops to 8%. This is not quite a correct estimate of the effectiveness of the program, since the actual estimated time is several orders of magnitude higher, and the time of writing to the file remains the same. Evaluation of the effectiveness of the directly calculated part of the program is given in the second part of Table. 1. The most reliable figures for assessing effectiveness are in the 90% area. The further growth of E_p can be explained by the acceleration of computations occurring in fast memory, as well as by fast exchanges.

Consider computing on graphics accelerators using CUDA. The results of the calculation are given in Table. 2 in the second part and denoted by the number «2». Despite the decrease in efficiency with the growth in the number of processes with respect to calculations performed using the MPI and OpenMP libraries, the computational speed on graphics accelerators exceeds the program execution speed by 15 or more times without the use of CUDA. In addition, these calculations allow to optimally choose the configuration for calculating a specific task. In this case, with $p = 12$ and above $a \approx 0,02$. This gives the maximum reasonable $p = 2 / a = 100$. On a relatively good scale, you can expect about to $p = 128$.

Thus, after conducting a series of calculations on a small number of processes, you can evaluate the dynamics of the scalability of the calculation and choose the optimal configuration for launching the software package.

The study of thermoelasticity problems, taking into account the interaction of deformation and temperature fields, began with [1–3]. This line of research was called the coupled thermoelasticity. Generalization and solution of particular problems of the new direction of research was continued in [3–5]. In subsequent years, both analytical, starting with [4, 5], and numerical methods [6] were developed to solve problems of coupled thermoelasticity. In the latter paper the authors were one of the first who developed a scheme of application of the finite element method and gave its implementation for solving the coupled problems of thermoelasticity. Analysis shows that in the overwhelming majority of studies in the solution of coupled thermoelasticity problems the finite element models of a fairly general purpose were developed, for example [7–11]. The analytical methods for solving this class of problems did not become as widespread as the numerical ones. The results obtained with their help were summarized in [12]. Beginning with papers [13–18], scientists consider uncoupled problems of thermoelasticity about the sliding contact of a rigid body with an elastic coating, taking into account friction, heating of the coating from friction, and abrasive wear. Because of the large number of parameters of the problem, the one-dimensional and quasi-static problems were considered. In [15–18], for their solution the integral Laplace transform with a solution in the form of functional series along the poles of the integrands of the contour quadratures of the inverse Laplace transform were used. The solution method allows establishing the parametric boundaries of the thermoelastic instability of a sliding contact, to investigate the properties of the solutions obtained. Beginning with [20–22], a new direction of the development of the model of the

sliding contact of two elastic bodies arose, taking into account friction, wear and heat generation, built on the principle of virtual energy and the basic laws of thermodynamics. The solution of problems on the basis of this model is carried out by the finite element method [22]. The present paper demonstrates the application of the Laplace integral transform and complex analysis methods to solution of the coupled thermoelastic problem on the coating wear occurring during sliding frictional contact with frictional heating.

References

1. Cockburn B. Introduction to the Discontinuous Galerkin Method for Convection Dominated Problems, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics, 1998, v.1697, pp.151-268.
2. A.K. Pany and S.Yadav. An hp-Local Discontinuous Galerkin method for Parabolic Integro-Differential Equations, OCCAM, Report No. 09/30.
3. Ladonkina M.E., Neklyudova O.A., Tishkin V.F., Utyralov D.I. Realization of the boundary conditions of adhesion for the Galerkin discontinuous method // Prep. M.V. Keldysh, 2014, 16 p.
4. Ladonkina M.E., Tishkin V.F. On methods of Godunov type of high accuracy order // Reports of the Academy of Sciences, 2015, T.461, No. 4, pp. 390-393.
5. Ladonkina M.E., Tishkin V.F. A generalization of the Godunov method, using piecewise polynomial approximations, Differentsial'nye Uravneniya, 2015, Vol. 51, No. 7, pp. 899-907.
6. Krasnov M.M. Operator library for solving three-dimensional grid problems of mathematical physics using graphics cards with CUDA architecture. // Matematicheskoe Modelirovanie, 2015, vol.27, no. 3, pp.109-120.
7. Bassi F., Rebay S. A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations // Journal of Computational Physics, 1997, 131, pp. 267-279.
8. SK Godunov. Difference method for numerical calculation of discontinuous solutions of hydrodynamics equations // Matem. sb., 1959. 47 (89): 3, pp. 271-306.
9. Rusanov V.V. Calculation of the interaction of non-stationary shock waves with obstacles. // Journal of Computational Mathematics and Mathematical Physics, 1961. T.I., №2, pp. 267-279.
10. Lax P.D. Weak solutions of nonlinear hyperbolic equations and their numerical computation // Communications on Pure and Applied Mathematics. 1954,7, No. 1, pp. 159 -193.
11. Arnold D.N., Brezzi F., Cockburn B., Marini L.D. Unified analysis of discontinuous Galerkin methods for elliptic problems. // SIAM Journal on Numerical Analysis, 2002, 29, pp. 1749-1779.
12. A.K. Pany and S. Yadav An hp-Local Discontinuous Galerkin method for Parabolic Integro-Differential Equations, OCCAM, Report No. 09/30
13. M.E. Ladonkina, O.A. Neklyudova, V.F. Tishkin. Investigation of the influence of the limiter on the order of the accuracy of the solution by the Galerkin discontinuous method. // KIAM Preprint. M.V. Keldysh, 2012, No. 34, pp. 31.
14. M.E. Ladonkina, O.A. Neklyudova, V.F. Tishkin. Investigation of the influence of the limiter on the order of accuracy of the solution by the Galerkin discontinuous method, Mat. Model., 2012, T.24, №12, pp. 124-128.

15. M.E. Ladonkina, OA Neklyudova, V.F. Tishkin. High accuracy limiter for the Galerkin discontinuous method on triangular meshes. // KIAM Preprint. M.V. Keldysh, 2013, No. 53, 26c.

16. Krasnov MM, Kuchugov PA, Ladonkina ME, Tishkin VF Galerkin discontinuous method on three-dimensional tetrahedral grids. Using the Operator Programming Method. // Mathematical Modeling, 2017, Vol. 29, No. 2, P. 3-22.

Authors:

Krasnov Mikhail Mikhailovich, Senior Researcher, Institute of Applied Mathematics, Keldysh Institute of Applied Mathematics Russian Academy of Sciences (4 Miusskaya pl., Moscow, Russian Federation)

Kuchugov Pavel Aleksandrovich, Candidate of Science in Physics and Maths, Researcher, Keldysh Institute of Applied Mathematics Russian Academy of Sciences (4 Miusskaya pl., Moscow, Russian Federation)

Ladonkina Marina Evgenyevna, Candidate of Science in Physics and Maths, Keldysh Institute of Applied Mathematics Russian Academy of Sciences (4 Miusskaya pl., Moscow, Russian Federation)

Tishkin Vladimir Fedorovich, Corresponding Member of the Russian Academy of Sciences, Professor, Doctor of Science in Physics and Maths, Institute of Applied Mathematics, Lavrentyev Institute of Hydrodynamics (Siberian Branch of the Russian Academy of Science), (Academician Lavrentiev Avenue, 15, Novosibirsk, Russian Federation)

УДК 004.942

Эффективный параллельный программный комплекс для решения уравнений Навье-Стокса разрывным методом Галеркина*

М.М. Краснов, П.А. Кучугов, М.Е. Ладонкина, В.Ф. Тишкин **

Институт прикладной математики им. М.В. Келдыша РАН, Москва, Россия

Институт гидродинамики им. М.А. Лаврентьева СО РАН, Новосибирск, Россия

Реализованы алгоритмы решения уравнения Навье-Стокса на трехмерной тетраэдральной сетке методом Галеркина с разрывными базисными функциями. При создании программного кода использован новый подход к программированию задач математической физики, позволяющий компактно записывать и эффективно реализовывать математические формулы, в частности, за счет введения понятия «сеточного оператора», аналогичного математическому, единообразно реализовывать подход на разных типах сеток и для различных вычислительных архитектур. Исследуется эффективность созданного программного кода.

Ключевые слова: уравнения Навье-Стокса, разрывный метод Галеркина, параллельное программирование, шаблонное метапрограммирование

Авторы:

Краснов Михаил Михайлович, старший научный сотрудник, Институт прикладной математики им. М.В. Келдыша Российской академии наук (125047, Москва, Миусская пл., д.4)

Кучугов Павел Александрович, кандидат физико-математических наук, научный сотрудник, Институт прикладной математики им. М.В. Келдыша Российской академии наук (125047, Москва, Миусская пл., д. 4)

Ладонкина Марина Евгеньевна, кандидат физико-математических наук, Институт прикладной математики им. М.В. Келдыша Российской академии наук (125047, Москва, Миусская пл., д. 4)

Тишкин Владимир Федорович, член-корреспондент РАН, профессор, доктор физико-математических наук, Институт прикладной математики им. М.В. Келдыша Российской академии наук (125047, Москва, Миусская пл., д. 4)

* Работа выполнена при финансовой поддержке Российского научного фонда (код проекта 16-11-10033).

**E-mail: kmm@kiam.ru, pkuchugov@gmail.com, ladonkina@imamod.ru, tishkin@mail.ru