Numerical solution of the Navier-Stokes equations by discontinuous Galerkin method

M M Krasnov¹, P A Kuchugov¹,², M E Ladonkina¹,², A E Lutsky¹, V F Tishkin¹,²
¹Keldysh Institute of Applied Mathematics of RAS, 4 Miusskaya Sq., 125047, Moscow, Russia
²Lavrentyev Institute of Hydrodynamics of SB RAS, 15 ac. Lavrentyeva Av., 630090, Novosibirsk, Russia
E-mail: lodonkina@imamod.ru

Abstract. Detailed unstructured grids and numerical methods of high accuracy are frequently used in the numerical simulation of gasdynamic flows in areas with complex geometry. Galerkin method with discontinuous basis functions or Discontinuous Galerkin Method (DGM) works well in dealing with such problems. This approach offers a number of advantages inherent to both finite-element and finite-difference approximations. Moreover, the present paper shows that DGM schemes can be viewed as Godunov method extension to piecewise-polynomial functions.

As is known, DGM involves significant computational complexity, and this brings up the question of ensuring the most effective use of all the computational capacity available. In order to speed up the calculations, operator programming method has been applied while creating the computational module. This approach makes possible compact encoding of mathematical formulas and facilitates the porting of programs to parallel architectures, such as NVidia CUDA and Intel Xeon Phi.

With the software package, based on DGM, numerical simulations of supersonic flow past solid bodies has been carried out. The numerical results are in good agreement with the experimental ones.

1. Introduction
The solution of real-world aerothermodynamics problems in areas with complex geometry demands the use of unstructured grids, which often should also consist of mesh elements of different scale, and the application of high accuracy numerical methods. Galerkin method with discontinuous basis functions or Discontinuous Galerkin Method (DGM) [1] works well in dealing with such problems. This approach offers a number of advantages inherent to both finite-element and finite-difference approximations. In particular, it provides an adjusted order of accuracy, is applicable for meshes of arbitrary structure, allows implement boundary conditions of various kind relatively easily and reliably theoretically grounded [1–5].

While using DGM, the approximation of the solution within the cell is in the form of polynomials of degree \( N \) with time-dependent coefficients. During the numerical solution it is necessary to correctly calculate flux functions on cell’s faces. The approach for calculation of flux function values for diffusion terms in the form of half sum has been proposed in work [6]. In [2] it is shown the necessity of using stabilizing additions for calculation of numerical fluxes while solving heat conductivity equations. In the present paper, it is proposed to use the second method for calculation values of flux functions for diffusion terms [7].
For all its merits, DGM involves significant computational complexity, and this brings up the question of ensuring the most effective use of all the computational capacity available. In the present paper, in order to speed up the calculations, grid-operator approach to the programming has been applied while creating the computational module for solving three-dimensional Navier-Stokes equations based on DGM. This approach makes possible compact encoding of mathematical formulas and facilitates the porting of programs to parallel architectures, such as NVidia CUDA and Intel Xeon Phi. Moreover, the usage of meta-programming features of C++ allows to do a lot of preliminary calculations on the compilation stage.

2. The description of DG method for the Navier-Stokes equations

Let’s consider the Navier-Stokes equations, written in the form of first-order equations:

$$\partial_t U + \nabla \cdot F(U) - \nabla \cdot G(U, \tau, q) = 0,$$

$$\tau = \left( \lambda - \frac{2}{3} \mu \right) \hat{I} (\nabla \cdot v) + 2\mu S(v), \quad S(v) = \frac{1}{2} (\nabla v + (\nabla v)^T), \quad q = -k\nabla T,$$

and supplemented with the appropriate initial and boundary conditions, the type of which depends on the specific objectives and will be specified further. Vector of conservative variables $U$ is given in the following form:

$$U = (\rho, \rho u, \rho v, \rho w, E)^T,$$

where $\rho$ is the density of the fluid, $u, v, w, p$ are the components of velocity $v, p$ is the pressure, $\epsilon$ is the specific internal energy and $E = \rho \left( \epsilon + \frac{u^2 + v^2 + w^2}{2} \right)$, is total energy per unit volume. Perfect gas equation of state will be used to determine the pressure value $p$

$$p = (\gamma - 1) \rho \epsilon,$$

where $\gamma$ is the specific heats ratio. For components of flux functions $F(U)$ and $G(U, \tau, q)$ one can obtain:

$$F_x(U) = (\rho u, \rho u^2 + p, \rho uv, \rho uw, (E + p)u)^T,$$

$$F_y(U) = (\rho u v, \rho u^2 + p, \rho uw, (E + p)v)^T,$$

and

$$G_x(U, \tau, q) = (0, \tau_{xx}, \tau_{xy}, \tau_{xz}, u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x)^T,$$

$$G_y(U, \tau, q) = (0, \tau_{yx}, \tau_{yy}, \tau_{yz}, u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y)^T,$$

$$G_z(U, \tau, q) = (0, \tau_{zx}, \tau_{zy}, \tau_{zz}, u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z)^T.$$

To apply DGM, the area $\Omega$, where the solution is sought, will be covered with a tetrahedral mesh $T_h$. An approximate solution of the set of the equations (1) will be sought in the form of polynomials $P(x)$ of degree $N$ with time-dependent coefficients on each mesh element $T_j$:

$$U_h(x, t) = \sum_{k=0}^{D_P} U_k(t) \phi_k(x),$$

$$q_{ih}(x, t) = \sum_{k=0}^{D_P} q_{ik}(t) \phi_k(x),$$

$$\tau_{ijh}(x, t) = \sum_{k=0}^{D_P} \tau_{ijk}(t) \phi_k(x),$$

(6)
where \( i, j = x, y, z, D_P = C_{N+3}^3 - 1 \), is the dimension of the space of polynomials, \( \phi_k \) is the corresponding basis function. In the present paper, Taylor basis is being used as the basis functions:

\[
\phi_k = \left( \frac{x - x_c}{\Delta x} \right)^\alpha \left( \frac{y - y_c}{\Delta y} \right)^\beta \left( \frac{z - z_c}{\Delta z} \right)^\delta,
\]

where \( \alpha + \beta + \delta = 0, 1, 2, \ldots N, x_c, y_c, z_c \), are the coordinates of the mass center of particular tetrahedron, \( \Delta x, \Delta y \) \( \text{and} \ \Delta z \), are the projections of the cell on the coordinate axis \( x, y \) \( \text{and} \ \text{z} \).

An approximate solution of (1) in DGM is sought as a solution of the following system [1,6]:

\[
\frac{d}{dt} \int_{T_j} \phi_k(x) U_h(x, t) d\Omega + \oint_{\partial T_j} \phi_k(x) h_F \left( U_h^+, U_h^-, n \right) d\sigma - \oint_{\partial T_j} \left( \frac{\partial \phi_k(x)}{\partial x} F_x(U_h(x, t)) + \frac{\partial \phi_k(x)}{\partial y} F_y(U_h(x, t)) + \frac{\partial \phi_k(x)}{\partial z} F_z(U_h(x, t)) \right) d\Omega - \oint_{\partial T_j} \phi_k(x) h_G \left( U_h^+, \tau^+, q^+, U_h^-, \tau^-, q^-, n \right) d\sigma + \oint_{\partial T_j} \left( \frac{\partial \phi_k(x)}{\partial x} G_x(U_h(x, t), \tau(x, t), q(x, t)) + \frac{\partial \phi_k(x)}{\partial y} G_y(U_h(x, t), \tau(x, t), q(x, t)) + \frac{\partial \phi_k(x)}{\partial z} G_z(U_h(x, t), \tau(x, t), q(x, t)) \right) d\Omega = 0, \quad (8)
\]

\[
\int_{T_j} \tau \cdot \Phi_k d\Omega = \oint_{\partial T_j} \left( \lambda - \frac{2}{3} \mu \right) \left( \Phi_k \cdot \hat{I} \right) h_v(v^+, v^-, n) d\sigma - \oint_{T_j} \left( \nabla \left( \left( \lambda - \frac{2}{3} \mu \right) \left( \Phi_k \cdot \hat{I} \right) \right), v \right) d\Omega + \oint_{\partial T_j} \mu \left[ h_S(v^+, v^-, n) \cdot \Phi_k \right] d\sigma - \oint_{T_j} (v, \nabla \cdot (\mu \Phi_k)) d\Omega, \quad \forall \phi_k(x), k = 0 \ldots D_P \quad (9)
\]

\[
\int_{T_j} \phi_l(x) q_h(x, t) d\Omega + \oint_{\partial T_j} \phi_l(x) h_T \left( T_h^+, T_h^-, k_h^+, k_h^- \right) n d\sigma - \oint_{T_j} \nabla (k(x) \phi_l(x)) T_h(x, t) d\Omega = 0, \quad (10)
\]

where \( U_h(x, t) \) is the solution vector, \( n \) is the outward normal vector to the element boundary \( \partial T_j \), \( h_F \left( U_h^+, U_h^-, n \right) \), \( h_G \left( U_h^+, \tau^+, q^+, U_h^-, \tau^-, q^-, n \right) \), \( h_v(v^+, v^-, n) \), \( h_S(v^+, v^-, n) \) and \( h_T \left( T^+, T^-, n \right) \) are flux functions calculated at element boundary \( \partial T_j \). Quantities denoted by \( U_h^- \) are calculated at the boundary \( \partial T_j \) of the element \( T_j \) using the values within the cell, while ones indicated by \( U_h^+ \) are calculated using the values of the adjunction cell. Function \( h_F \left( U_h^+, U_h^-, n \right) \) is monotone and it satisfies the matching condition:

\[
h_F \left( U_h(x, t), U_h(x, t), n \right) = F \left( U_h(x, t) \right). \quad (11)
\]

In the present paper, Godunov [8] and Rusanov–Lax–Friedrichs [9,10] fluxes have been used:

\[
h_F \left( U_h^+(x, t), U_h^-(x, t), n \right) = \frac{1}{2} \left( F \left( U_h^+(x, t) \right) + F \left( U_h^-(x, t) \right) \right) - B \left( U_h^+(x, t) - U_h^-(x, t) \right), \quad B = \max \left( |v^+| + c^+, |v^-| + c^- \right), \quad (12)
\]

where \( v \) is the velocity and \( c \) is the speed of sound.
It is necessary to calculate flux functions $h_T$, $h_v$ and $h_S$ at element boundary in equations (9)–(10). The following expressions are used for this aim:

$$h_T(T^+_h, k^+_h, T^-_h, k^-_h) = \left(\frac{k^+T^+ + k^-T^-}{2}\right),$$

$$h_v(v^+, v^-, n) = \left(\frac{v^+ + v^-}{2}\right),$$

$$h_S(v^+, v^-, n) = \frac{v^+ + v^-}{2} \otimes n + n \otimes \frac{v^+ + v^-}{2}.\quad (13)$$

It is necessary to calculate the flow values of $h_G(U^+_h, \tau^+, q^+, U^-_h, \tau^-, q^-, n)$ at the element boundary in (8) using stabilizing additions [7] as for heat conductivity equation [2,3]:

$$\hat{q}(T^+_h, q^+, T^-_h, q^-, n) = \frac{q^+ + q^-}{2} + C_3 (k^+T^+ - k^-T^-) n,$$

$$\hat{\tau}(U^+_h, \tau^+, U^-_h, \tau^-, n) = \frac{\tau^+ + \tau^-}{2} + C_1 ((v^+ - v^-), n) \hat{I} + C_2 (v^+ - v^-) \otimes n + n \otimes (v^+ - v^-),$$

$$h_G(U^+_h, \tau^+, q^+, U^-_h, \tau^-, q^-, n) = G_x(U^+_h, U^-_h, \tau, q, n_x) + G_y(U^+_h, U^-_h, \tau, q, n_y) + G_z(U^+_h, U^-_h, \tau, q, n_z).\quad (14)$$

All the integrals appearing in equations (8)–(10) are calculated using numerical Gaussian quadrature formulas, the number of integration points corresponding to the required calculation accuracy [11,12].

As is known, to ensure the monotonicity of the numerical solution obtained by this method, it is necessary to apply so called slope limiters, especially in the case of strong discontinuities [13–15]. In the present paper the classical Cockburn limiter [1] has been used due to the fact that it is easily implemented in the multidimensional case on grids of arbitrary structure.

The system of ordinary differential equations (8), which determines the changes in time of numerical solution, can be written as

$$\frac{dU^p}{dt} = A^{-1}R(U^p),\quad (15)$$

where $A$ is the mass matrix, $U^p$ is the global vector of degrees of freedom (the expansion coefficients of all the sought-for functions), $k$ is the equation number in the system (1) and $R(U^p)$ is the vector of the right hand sides.

**Remark.** For high order accuracy methods it is necessary to use higher-order scheme in time. In this paper we use the Runge–Kutta scheme of the third order [1].

In the case when viscosity and heat conductivity are vanishing, it is possible to show that equations (8) can be obtained by sequential projection of initial data on the space of piecewise-polynomial functions $E_P$ and by the subsequent solution of Riemann problem on the cell’s faces and that these solution can be used as the new initial data at the time moment $t + \Delta t$.

Thus, we consider the problem with the initial data in the form of piecewise-polynomial function $U_p(x, 0)$. If during some time period $\Delta t_0$ the exact solution of this problem is known, then, receiving it at time moment $\Delta t_0$, it can be again projected to $E_P$, and so on. As is known, from works [16,17] the exact solution with piecewise-polynomial data for sufficiently small values $\Delta t$ can be obtained with accuracy $O(\Delta t)$ as the solution of Riemann problem with the data to the left and to the right of the discontinuity. It is known that in the multidimensional case around the vertices and the edges of the cells a more complex problem arises which has no exact
solution at the moment. However, due to the finite speed of propagation of perturbations for hyperbolic problems, the area of influence of these data near these points and edges does not exceed \( c \Delta t \), where \( c \) is the maximum speed of propagation of disturbances. In the limit \( \Delta t \to 0 \) we obtain the function \( U_p(x, t) = \sum_{k=0}^{Dp} U_{kj}(t) \phi_{kj}(x) \). Coefficients \( U_{kj}(t) \) are calculated using the following equations:

\[
\frac{d}{dt} U_{kj}(t) \left( \int_{T_j} \phi_{kj}^2(x, t) dx \right) - \int_{T_j} \left( \frac{\partial \phi_{kj}(x, t)}{\partial x} F_x(U(x, t)) + \frac{\partial \phi_{kj}(x, t)}{\partial y} F_y(U(x, t)) + \frac{\partial \phi_{kj}(x, t)}{\partial z} F_z(U(x, t)) \right) dx + \int_{\partial T_j} \left( F_x^*(U(x, t)) n_x + F_y^*(U(x, t)) n_y + F_z^*(U(x, t)) n_z \right) \phi_{kj}(x, t) d\sigma = 0,
\]

where the flow values \( F_x^*(U(x, t)) \), \( F_y^*(U(x, t)) \), \( F_z^*(U(x, t)) \) are obtained as a result of Riemann problem solution with the initial data \( U_j^+ \) and \( U_j^- \) on the left and right sides of discontinuity. It should be noted that, in the case when \( \phi_{kj}(x, t) \) doesn’t depend on time \( t \), equations (16) coincide with expressions (7)-(10) if Godunov fluxes [8] are used as flow functions. Thus, discontinuous Galerkin schemes can be regarded as Godunov method extension [4, 5] to piecewise-polynomial functions.

3. Program realization of DG method

As is known, DGM involves significant computational complexity, and this brings up the question of ensuring the most effective use of all the computational capacity available. In order to speed up the calculations, the grid-operator approach to the programming of mathematical physics problems [18] has been applied while creating the computational module. This method is aimed at maximizing parallelization on shared memory and facilitates programs porting to parallel architectures, such as NVidia CUDA and Intel Xeon Phi. It also makes possible compact encoding of mathematical formulas.

In order to use operator method, it is necessary to clearly identify the parts, which may be designed in the form of operators, in general algorithm. In DGM they are volume integration and calculation of fluxes through cell faces and limiting. Their precise mathematical formulation has allowed to implement them relatively easily and very efficiently [19, 20].

Let’s illustrate the approach feasibility on an example of volume integrals calculations in equations (8). The expansion coefficients of \( U_h \) by basis functions serve to ensure the input of the corresponding procedure. Domain-specific language (DSL) with simple grammar is implemented by means of meta-programming features of C++ (using expression templates), and it allows to input the expressions, which will be later used for computations, as the integration function parameters. Finally, the program appears as follows:

\[
\begin{align*}
11ev &= \text{volume Integral}(\rho_u, \rho_v, \rho_w)(U); \\
12ev &= \text{volume Integral}(\rho_u * u + p, \rho_u * v, \rho_u * w)(U); \\
13ev &= \text{volume Integral}(\rho_v * u, \rho_v * v + p, \rho_v * w)(U); \\
14ev &= \text{volume Integral}(\rho_w * u, \rho_w * v, \rho_w * u + p)(U); \\
15ev &= \text{volume Integral}(E + p * u, E + p * v, E + p * w)(U);
\end{align*}
\]

All the other operators are implemented in the same manner. It should be noted here, that according to DGM, surface integrals of flux functions through the particular cell face are calculated twice. However, it doesn’t contradict the operator programming method and doesn’t significantly degrade the performance.
Thus, one can get simply readable and effective numerical code portable to various parallel architectures.

4. Some examples of calculations using DG method
This section illustrates numerical results of calculations of several model problems using numerical method and its software implementation presented above.

4.1. Flow past a triangular prism
One of such examples may be taken in the form of the problem of shock wave interaction with a stationary obstacle, such as a triangular prism. In this case, a complex pattern is formed by secondary shock waves and weak discontinuities, as well as the result of their interaction with each other and with vortices formed in the flow at the base of the prism. A mesh with a total number of about $10^7$ cells has been used in calculation. Four basis functions have been used to represent the mesh function in the cell. Numerical domain has the size of $L_x = 0.3$ m, $L_y = 0.15$ m, $L_z = 0.1$ m, the prism apex is located at the distance of $L_V = 0.1$ m from the left border of the domain on symmetry axis $y = L_y/2$, the prism base $b = 0.02$ m and half apex angle is $30^\circ$. The background pressure was $0.05$ MPa, temperature $-300$ °K, Mach number of primary shock wave was $1.3$, specific heats ratio $\gamma = 1.4$.

Figure 1 shows the numerical results of the corresponding calculation at different time moments. The comparison of the resulting flow structure with experimental shadowgraphs, see, for example, [21], suggests their good agreement. One of the inconsistencies lies in the lack of the secondary vortices, which may be due to the insufficient resolution of the used numerical grid.

![Figure 1](image1.png)

**Figure 1.** Numerical Schlieren with marked density contours illustrating the problem of the shock wave interaction with a solid triangular prism at time moments $t = 53, 102, 130$ and $172$ $\mu$s.

Another exponential problem is the modeling of the supersonic inviscid flow past another triangular prism [22], which this time allows to assess numerical results in a quantitative sense. In this case, oblique bow shock wave is formed and its slope angle $\beta$ to the axis of symmetry...
can be calculated analytically. Also a characteristic pattern of shock waves reflected from the boundaries is formed, as well as the result of their interaction. Initial setup parameters for this problem are different, in particular, $L_x = 0.12$ m, $L_y = 0.03$ m, $L_z = 0.01$ m, $L_V = 0.005$ m, the height of the prism is $l = 0.011$ m. The density distribution of the steady flow for half apex angles of the prism of $20^\circ$ is shown below, on figure 2. The numerical value of the angle $\beta$ for the considered option is $34.9^\circ$ and its theoretical value – $34.58^\circ$, indicating a good agreement of the numerical experiment with analytical data within 1.5%.

4.2. Flow in the compression corner

In this section a more complex problem which illustrates the structure of the supersonic separated flow and shock waves interaction with the boundary layer is being considered. According to the experiment described in [23], the flow in the compression corner has been modelled. Geometric configuration of the solid body model with sizes in millimeters is shown on figure 3. It consists of a horizontal plate with a ramp which is inclined at $30^\circ$. The following wind tunnel flow parameters have been used: Mach number of 6.01, pressure $p_{0\infty} = 9.72 \cdot 10^5$ Pa, temperature $T_{0\infty} = 380$ °K and Reynolds number calculated using the model length $L = 50$ mm as a spatial scale $Re_L = 6 \cdot 10^5$. The calculations have been performed for the angle of attach $\alpha = 0^\circ$ using unstructured tetrahedral mesh with $\sim 10^6$ cells. The viscous and thermal conductivity have been taken into account. Dynamic viscosity coefficient was calculated according to Sutherland’s law.

At high flow velocity, shock waves, which have been formed over the separation zone, interact with the shock wave at the reattachment point which leads to the appearance of a complicated flow structure affecting the shear layer above the region of the reverse flow. Numerical experiments have been aimed at reproducing this pattern. As shown on figure 4a,
all the waves, as well as the separation and reattachment points, can be clearly evidenced together with the shear layer above the region of the reverse flow. The distribution of the pressure coefficient $C_p$ on the wall along the symmetry axis of the model, shown on figure 4b, qualitatively agrees with the numerical and experimental results given in [23]. It should be noted that the boundary layer exceeds the experimental one, but this fact can be attributed to the use of a coarse mesh in calculations. The same reason can be given for some smoothing of $C_p$ profile at the location of the ramp mounting.

5. Conclusions
In this paper numerical simulations of hypersonic flow past various solid bodies have been carried out. The interaction of the considered flow with the obstacle generates a set of shock and rarefaction waves, numerous Mach legs, triple points as well as vortices that develop as time goes and interact with each other. In this case the steady flow is of particular interest. Despite the fact that computations performed have been based on coarse mesh, their results are in good agreement with the experimental and numerical ones and the complex flow pattern is accurately captured. This fact, in turn, confirms the effectiveness of DGM application to solving aerodynamic problems in areas with complex geometry.

The use of the meta-programming of C++ language has allowed to perform part of the calculations at compilation stage, and thus to reduce overall program execution time, while the use of the operator approach has permitted to create an efficient software implementation of the algorithm, which can be easily ported to the modern parallel computing architecture, including graphics accelerators.

Acknowledgments
The research has been performed under the financial support of Russian Science Foundation, grant No. 16-11-10033.
References
[1] Cockburn B 1998 An introduction to the Discontinuous Galerkin method for convection-dominated problems vol 1697 (Springer Berlin Heidelberg) chap Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, pp 150–268
[2] Arnold D N, Brezzi F, Cockburn B and Marini L D 2001 SIAM Journal on Numerical Analysis 39 pp 1749–1779
[3] Pani A K and Yadav S 2011 Journal of Scientific Computing 46 pp 71–99
[4] Ladonkina M E and Tishkin V F 2015 Doklady Mathematics 91 pp 189–192
[5] Ladonkina M E and Tishkin V F 2015 Differential Equations 51 pp 895–903
[6] Bassi F and Rebay S 1997 Journal of Computational Physics 131 pp 267–279
[7] Ladonkina M E, Neklyudova O A, Tishkin V F and Utiralov D I 2014 The no-slip boundary conditions for discontinuous Galerkin method Preprint 32 Keldysh Institute of Applied Mathematics of RAS
[8] Godunov S K 1959 Mat. Sb. (N.S.) 47(89) pp 271–306
[9] Rusanov V V 1962 USSR Computational Mathematics and Mathematical Physics 1 pp 304–320
[10] Lax P D 1954 Communications on Pure and Applied Mathematics VII pp 159–193
[11] Myskovskikh I P 1981 Interpoliationshnice khabiturnie formuli (M.:Nauka)
[12] Li B Q 2006 Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer (Springer-Verlag London)
[13] Ladonkina M E, Neklyudova O A and Tishkin V F 2012 Research of the impact of different limiting functions on the order of solution obtained by RKDG Preprint 34 Keldysh Institute of Applied Mathematics of RAS
[14] Ladonkina M E, Neklyudova O A and Tishkin V F 2013 Mathematical Models and Computer Simulations 5 pp 346–349
[15] Ladonkina M E, Neklyudova O A and Tishkin V F 2013 The high order limiter for RKDG on triangular meshes Preprint 53 Keldysh Institute of Applied Mathematics of RAS
[16] Teshukov V M 1980 Journal of Applied Mechanics and Technical Physics 21 pp 261–267
[17] Men’shov I S 1990 USSR Computational Mathematics and Mathematical Physics 30 pp 54–65
[18] Krasnov M M 2015 Matem. Mod. 27 pp 109–120
[19] Krasnov M M and Ladonkina M E 2016 Discontinuous Galerkin method on three-dimensional tetrahedral meshes. the usage of C++ template metaprogramming Preprint 24 Keldysh Institute of Applied Mathematics of RAS
[20] Krasnov M M, Ladonkina M E and Tishkin V F 2016 Discontinuous Galerkin method on three-dimensional tetrahedral meshes. the usage of the operator programming method Preprint 23 Keldysh Institute of Applied Mathematics of RAS
[21] Chang S M and Chang K S 2000 Shock Waves 10 pp 333–343
[22] Chaudhuri A, Hadjadj A and Chinnayya A 2011 Journal of Computational Physics 230 pp 1731–1748
[23] Zapryagaev V, Kavun I and Lipatov I 2013 Progress in Flight Physics 5 pp 349–362