

The universal algorithm for solving the gas dynamics equations on the mesh with arbitrary number of cell faces

K V Kostyushin¹, V A Kotonogov, A M Kagenov, I M Tyryshkin, A A Glazunov, I V Eremin and K N Zhiltsov

Research Institute of Applied Mathematics and Mechanics of *Tomsk State University*,
36 Lenin Avenue, Tomsk 634050, Russian Federation.

¹Corresponding author: kostushink@niipmm.tsu.ru

Abstract. The paper presents methodology and algorithm for calculating the equations of gas dynamics on arbitrary computational meshes with a mixed type of cells. The calculation method is based on the method of linear reconstruction proposed by Barth and Jespersion. The algorithm for determining the geometric parameters of arbitrary computational cell is presented. To implement the calculation algorithm, a data storage system has been proposed and tested. The algorithm of the solver and the algorithm of docking the computational meshes in the case of using block-structured meshes are proposed. The efficiency of methodology and developed program of calculation are demonstrated by the calculation example of the air flow in flat air intakes. The structure of flow and position of the bow shock wave are determined. These results with the theoretical values were compared. The application of the proposed methodology and calculation algorithm to arbitrary computational meshes with a mixed cell type makes it possible to optimize the process of constructing computational mesh and conduct numerical studies of gas dynamics in regions of complex geometry.

1. Introduction

For simulation of combustion products flow in the gas-dynamic tracts of rocket engines numerical methods of increased accuracy are widely used based on the Godunov scheme [1-3]. Methods of this type have been developing for more than forty years, and by now a huge number of their variants and implementations have been developed [4]. For the first time, a scheme of the second order of accuracy in space based on Godunov's method was proposed by Kolgan in 1972 [5, 6]. The main idea of the Godunov-Kolgan scheme was in to replace the piecewise constant distribution of a function inside a cell by a piecewise linear distribution. To meet the conditions of preservation of the total variation (Total Variation Diminishing, TVD), Kolgan proposed to use the principle of the minimum value of the derivative. In 1979, the approach proposed by Kolgan was developed in the works of Van Lier [7, 8], in which the MUSCL (Monotonic Upstream-Centered Scheme for Conservation Laws) scheme was proposed. The scheme became very popular, and as a result of its development, a large family of schemes of the MUSCL type was born [9]. However, the use of schemes such as MUSCL on unstructured computational mesh is accompanied by a number of computational difficulties [10]. In this connection, the method of linear reconstruction of the solution (Piecewise linear reconstruction) Barth and Jespersion [11], which is the development of the approach of Telyaeva [12], is often used to calculate currents on unstructured mesh. Using the Barth and Jespersion approach it is easy enough to obtain solutions on unstructured computational meshes. However, for the development of universal



solver that allows calculations on arbitrary computational meshes with different number of faces of the computational cell, it is necessary to implement special solver algorithms and data storage systems. The present work is devoted to solving these problems.

2. Calculation methodology

The Euler equations, which govern the compressible, inviscid gas flow in three-dimensions, are [13]:

$$\frac{d}{dt} \left(\iiint_G \rho dG \right) + \oint_S \rho \mathbf{v} \cdot d\mathbf{S} = 0, \quad (1)$$

$$\frac{d}{dt} \left(\iiint_G \rho \mathbf{v} dG \right) + \oint_S (\rho \mathbf{v} \mathbf{v} + p \hat{\mathbf{I}}) \cdot d\mathbf{S} = 0, \quad (2)$$

$$\frac{d}{dt} \left(\iiint_G e dG \right) + \oint_S (e + p) \mathbf{v} \cdot d\mathbf{S} = 0, \quad (3)$$

where: ρ – density; t – time; p – pressure; $\mathbf{v} = [u, v, w]^T$ – gas velocity; $\hat{\mathbf{I}}$ – 3×3 identity tensor; ε – specific internal energy; $e = \rho \varepsilon + \rho(u^2 + v^2 + w^2)/2$ – total energy per unit volume; G – final volume in three-dimensional space; $dG = dx dy dz$ – element of volume; S – surface limiting the volume G ; $d\mathbf{S} = \mathbf{n} dS$ – orientable surface element S , where \mathbf{n} – external normal unit vector to the surface S ; dS – differential surface element.

System (1) – (3) is closed by equation of state:

$$\varepsilon = \frac{p}{(k-1)\rho}. \quad (4)$$

In order to build difference scheme the finite volume method is applied. Therefore, we will cover all computational domains with discrete cells consisting of arbitrary convex polyhedrons with volume G_i , where $i = 1, 2, \dots$ – volume number, and $m(i)$ – face number of i finite volume. Each face of i finite volume has surface S_j , where $j = 1, 2, \dots, m(i)$. Approximation of integral equations in each of polyhedrons obtains in the following way:

$$G_i \frac{\rho_i^{k+1} - \rho_i^k}{\Delta t} + \sum_{j=1}^{m(i)} R_j (\mathbf{V}_j \cdot \mathbf{S}_j) = 0, \quad (5)$$

$$G_i \frac{(\rho \mathbf{v})_i^{k+1} - (\rho \mathbf{v})_i^k}{\Delta t} + \sum_{j=1}^{m(i)} (R_j \mathbf{V}_j) (\mathbf{V}_j \cdot \mathbf{S}_j) + \sum_{j=1}^{m(i)} P_j \mathbf{S}_j = 0, \quad (6)$$

$$G_i \frac{e_i^{k+1} - e_i^k}{\Delta t} + \sum_{j=1}^{m(i)} (E_j + P_j) (\mathbf{V}_j \cdot \mathbf{S}_j) = 0. \quad (7)$$

As a result we will get Godunov's scheme [14] for custom computational mesh. Here $\mathbf{S}_j = \mathbf{n}_j S_j$, and Δt – time step. Lower index i signifies value of functions, which are referred to mass center of i polyhedrons, and lower index j signifies values referred to center of j face of discrete cell. Upper index k signifies the number of time step. "Large" values of R , \mathbf{V} , P , E signify correspondingly density, velocity, pressure and total energy at faces of control volume. These values are defined from the solution of Riemann problem [14] in the direction of outward normal. Also for the flow calculations can be used other methods, based on approximate solutions of Riemann problem: Roe, Osher, HLL, HLLC, HLLE and others [15]. Scheme (5) – (7) is easily generalized for two dimensional and axisymmetric approaches [14].

2.1. Piecewise linear reconstruction

To increase the order of accuracy in space on unstructured computational mesh, the method of linear reconstruction of the solution, proposed by Barth and Jespersen in 1989 [11], is widely used. The parameters of linear reconstruction method on the right and left boundaries of the computational cell are determined using the following relationships:

$$\mathbf{Q}^L = \mathbf{Q}_i + \psi_i (\nabla \mathbf{Q}_i \vec{\mathbf{r}}_L), \quad (8)$$

$$Q^R = Q_j + \psi_j(\nabla Q_i \vec{r}_L), \tag{9}$$

where Q_i – parameter in center of computational cell i , Q_j – parameter in center of computational cell j , $\nabla Q_i = \left[\frac{\partial Q}{\partial x}, \frac{\partial Q}{\partial y}, \frac{\partial Q}{\partial z} \right]^T$ gradient Q in center of cell i , \vec{r}_L – vector-centroid connecting center cell i and center of the face, ψ_i – function limiter for cell i .

To determine the gradients, the least squares method is used:

$$\begin{bmatrix} \Delta x_{i1} & \Delta y_{i1} & \Delta z_{i1} \\ \Delta x_{i2} & \Delta y_{i2} & \Delta z_{i2} \\ \vdots & \vdots & \vdots \\ \Delta x_{ij} & \Delta y_{ij} & \Delta z_{ij} \\ \vdots & \vdots & \vdots \\ \Delta x_{iN} & \Delta y_{iN} & \Delta z_{iN} \end{bmatrix} \begin{bmatrix} \partial_x Q \\ \partial_y Q \\ \partial_z Q \end{bmatrix} = \begin{bmatrix} Q_1 - Q_i \\ Q_2 - Q_i \\ \vdots \\ Q_j - Q_i \\ \vdots \\ Q_N - Q_i \end{bmatrix}, \tag{10}$$

where $\Delta(\cdot)_{ij} = (\cdot)_i - (\cdot)_j$, N – number of neighboring cells.

To satisfy TVD conditions, a limiter function “Barth and Jespersion” is used:

$$\psi_j = \min_{j \in N(i)} \begin{cases} \min \left(1, \frac{Q_{\max} - Q_i}{\Delta_2} \right), & \Delta_2 > 0 \\ \min \left(1, \frac{Q_{\min} - Q_i}{\Delta_2} \right), & \Delta_2 < 0 \\ 1, & \Delta_2 = 0 \end{cases} \tag{11}$$

where $\Delta_2 = \nabla Q_i \cdot \vec{r}_L$, $\Delta_2 = \nabla Q_i \cdot \vec{r}_L$,
 $Q_{\max} = \max (Q_i, \max_{j \in N(i)}(Q_j))$, $Q_{\min} = \min (Q_i, \min_{j \in N(i)}(Q_j))$.

2.2. Topology of arbitrary computational

For simplicity of presentation, consider a two-dimensional computational mesh with an arbitrary number of cell faces. Let the cell C of an arbitrary mesh be represented as a list of vertices $(p_1, p_2, p_3, \dots, p_n)$, faces $(e_1, e_2, e_3, \dots, e_n)$ and neighboring cells $(c_1, c_2, c_3, \dots, c_n)$, having a common face with the cell under consideration, where n – number of cell vertices (figure 1). The vertices are numbered by traversing the cell counterclockwise. Cell faces are indexed similarly, starting with the face passing through the vertices p_1 and p_2 . Neighboring cell is assigned an index common face.

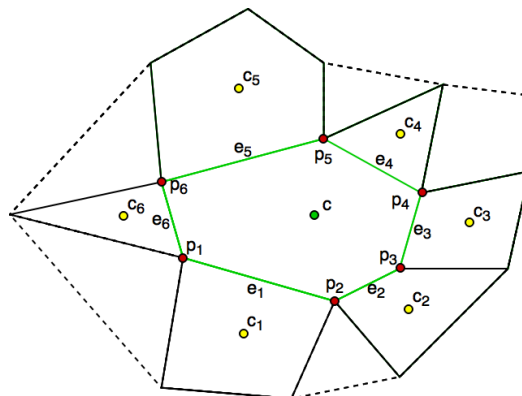


Figure 1. Topology of an arbitrary computational mesh.

Then the coordinates of the center of an arbitrary cell are calculated as follows:

$$c = \frac{1}{n} \sum_{i=1}^n p_i.$$

The lengths of the faces can be calculated as follows:

$$L_{ei} = |P_{ij}| = \sqrt{(x_{p_i} - x_{p_j})^2 + (y_{p_i} - y_{p_j})^2}, j = \begin{cases} i + 1, & i \neq n \\ 1, & i = n \end{cases}$$

where i – number of face [1..n].

The unit normal vectors are calculated using the following relation:

$$n_{ei} = (x_{p_j} - x_{p_i}, y_{p_i} - y_{p_j}) / L_{ei}, j = \begin{cases} i + 1, & i \neq n \\ 1, & i = n \end{cases}$$

To calculate the area of an arbitrary cell, we divide it into k triangles using any triangulation algorithm (figure 2).

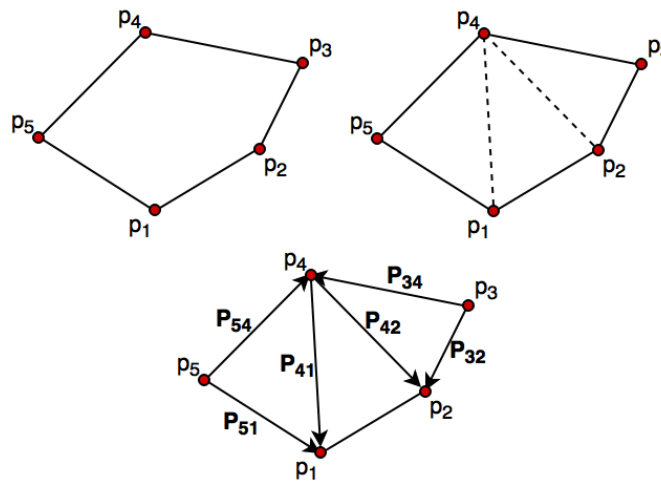


Figure 2. Example of calculating area of an arbitrary cell.

Then area of the cell can be defined as the sum of the areas of the triangles:

$$S_c = \sum_{j=1}^k S_j.$$

The algorithm described above makes it possible to construct all the basic types of two-dimensional computational grids.

3. System of data storage and solver algorithm

The simplest approach for organization the solver algorithm is the approach based on data storage in isolated arrays and their further correlation by using indexes of corresponding elements. This approach quite quickly allows developing solvers for meshes with topology known in advance, however, even at the slightest alteration of mesh topology it is necessary to modify the data storage system and solver algorithm. In this regard, for realization of solver algorithm, adapted to arbitrary face number in cells, it is suggested the following data structure should be used (figure 3):

Class «Cell». Instances of class «Cell» keep information about cell geometry (mass center, area and so on) and gas parameters in cell.

Class «Face». Instances of class «Face» keep information about geometry (area or length, normal vector) and gas parameters (big values) at the cell face.

Data model of computational mesh is presented in the form of collections of class instances «Cell» and «Face».

Interconnection «Cell» - «Face» is realized by storage of collection references on class instances «Face» in class «Cell», and connection «Face» – «Cell», by storage in class «Face» references on class instances «Cell».

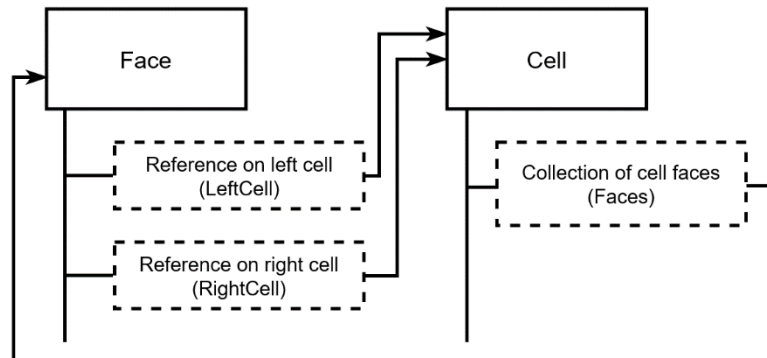


Figure 3. Interconnection between classes «Cell» and «Face».

Such system of data storage does not have any dependence on computational mesh topology, and allows conducting calculations on meshes of custom configuration. When using object-oriented programming languages, scheme of solver (5) – (7) can be written in a quite simple and compact way (figure 4).

```

foreach (var cell in cells)
{
    double stream_1 = 0;
    Vector<double> stream_2 = new Vector(0,0,0);
    double stream_3 =0;

    foreach (var face in cell.faces)
    {
        stream_1 += face.Gas.Ro * (face.Gas.V * face.N * face.S);
        stream_2 += (face.Gas.Ro * face.Gas.V) * (face.Gas.V * face.N * face.S) + face.Gas.P *
face.N * face.S;
        stream_3 += (face.Gas.E + face.Gas.P) * (face.Gas.V * face.N * face.S);
    }

    cell.Gas_Next.Ro = cell.Gas.Ro - dt / cell.G * stream_1;
    cell.Gas_Next.V = cell.Gas.Ro * cell.Gas.V - dt / cell.G * stream_2;
    cell.Gas_Next.E = cell.Gas.E - dt / cell.G * stream_3;
}

```

Figure 4. A listing of the example of solver source code of time step. Language C#.

Here «cells» are collection of class instance cell, «cell.faces» are collection of reference on cell faces, «stream_1», «stream_2», «stream_3» – sum of flux through faces of computational cell for conservation equation of mass, momentum and energy correspondingly. Fields «face.Gas.Ro», «face.Gas.V», «face.Gas.P» signify density, velocity and pressure at the face of computational cell and calculate by solving Riemann problem (not given in the listing). «face.S», «face.N» – area and unit vector of outward normal. «cell.Gas», «cell.Gas_Next» are gas parameters in the cell center at the current and next time step correspondingly, «cell.G» is cell volume, «dt» is current time step, which calculated from Courant–Friedrichs–Lewy condition. Fields «V», «N» and variable «stream_2» – instances of class «Vector» for which standard vector operations are identified. The use of suggested data storage system and solver algorithm allow conducting calculations at meshes with arbitrary topology.

3.1. Mesh coupling

At the usage of blocked structured or unstructured meshes, the task of meshes coupling among sub domains comes up. In operation, for meshes coupling among sub domains the algorithm, which is based on edges division of the cells (on the coupling line) into composite edges, is used. As an example, let look at the coupling algorithm of two two-dimensional computational sub domains shown in figure 5.

For each boundary cell (c) from the first sub domain, the edge (p2-p4) is given out, which join on the boundary of sub domains separation.

Boundary cells of the second sub domain (c2, c3) are defined, the ones that have common boundaries with the analyzed cell (c).

The vertex (p3) of cells (c2, c3) is defined, which situated on the edge (p2-p4).

The edge (p2-p4) of the cell (c) is broken into several edges (e2, e3) along the vertex (p3).

Algorithm for each cell from the first domain is repeated.

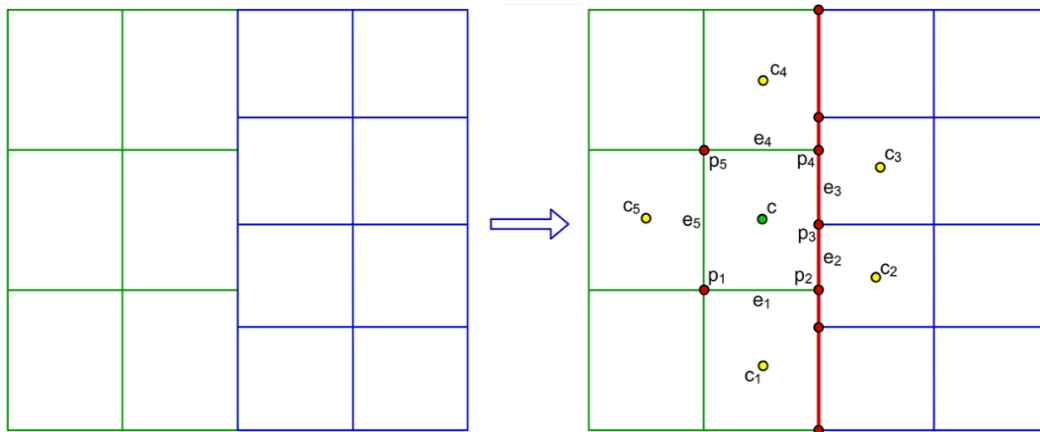


Figure 5. Coupling scheme of two computational sub domains.

For more complex configurations of couplings the algorithm is analogous. Algorithm can be used for coupling of sub domains with different topology, and also it is easily generalized on three-dimensional computational meshes.

4. Calculation of supersonic flow in air intake

To verify algorithm and methodology of calculation series of calculations on simple flat hypersonic air intake was conducted. General scheme of air intakes under consideration is given in figure 6.

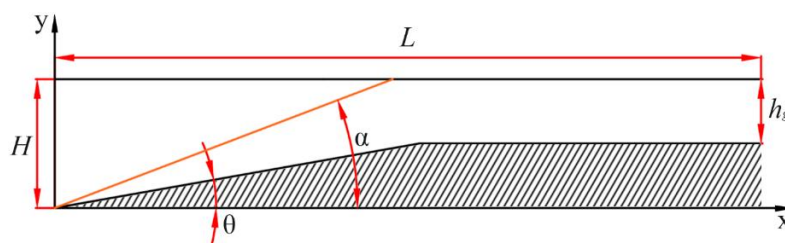


Figure 6. Air intake scheme.

Where $H = 1$ – characteristic dimension; $L = 8$ – length of air intake; θ – inclination angle at inlet of air intake, α – angle of reflected shock wave, $h_g = 0.5$ – height of straight section of air intake.

In describing the flow of an inviscid ideal gas, the relationship between the inclination angle θ and the angle of the reflected shock wave α has the form [16]:

$$\text{ctg}\theta = \left[\frac{\frac{k+1}{2}M^2}{M^2\sin^2\alpha - 1} \right] \text{tg}\alpha \tag{12}$$

Using the dependence (12), air intake configurations for the angles of the reflected shock wave 20, 25 and 30 and the speed $M=5$ of the incoming air flow are constructed. The geometric parameters of the air intakes are given in table 1.

Table 1. Geometric parameters of air intake.

Type of configuration	θ	α
Type 1	10.665	20
Type 2	15.644	25
Type 3	20.173	30

To build the computational mesh the computational domain was divided into 3 computational sub domains, which are shown on figure 7. Structured computational grid was built in each subdomain. A computational mesh coupling among sub domains was conducted with the use of above-mentioned algorithm. As an example, figure 8 shows a part of computational mesh in A-B coupling area. Table 2 shows the mesh size of A, B and C of computational domain.

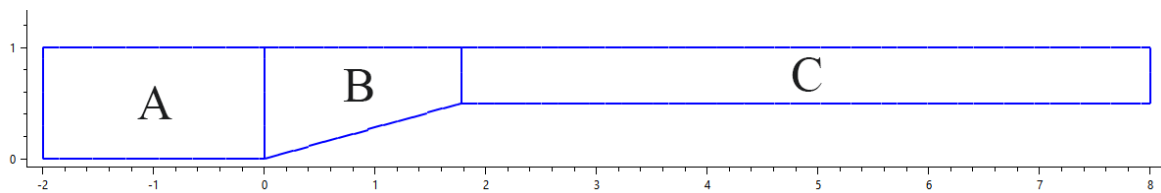


Figure 7. Computational domains for air intake’s configuration with $\alpha = 25^\circ$.

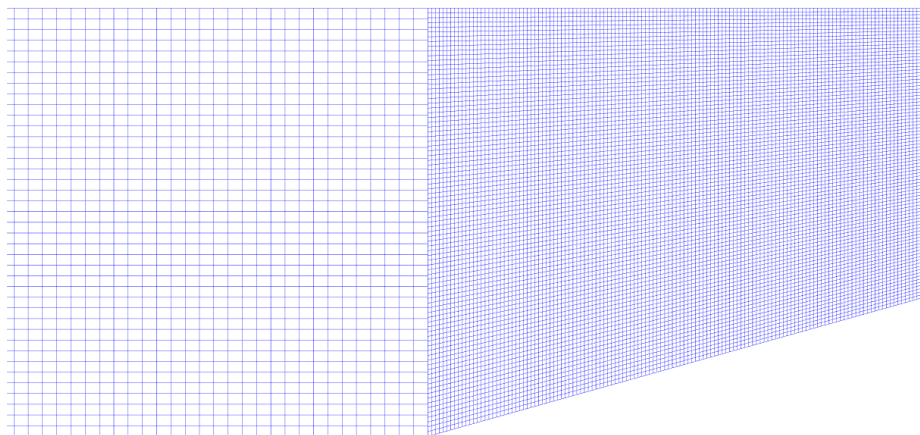


Figure 8. Computational mesh in A-B coupling area.

Table 2. Mesh size parameters.

Type of configuration	Area A	Area B	Area C
Type 1	60×40	350×90	650×90
Type 2	60×40	200×90	700×90
Type 3	60×40	200×90	950×90

At all inner elements of air intake's construction, the boundary condition is solid wall. At exit section of air intake – supersonic outflow. At other boundaries of outer domains – supersonic inflow with the speed $M_0 = 5$. Working gas is air. At figure 9 distribution of Mach number is shown for three configurations of air intakes.

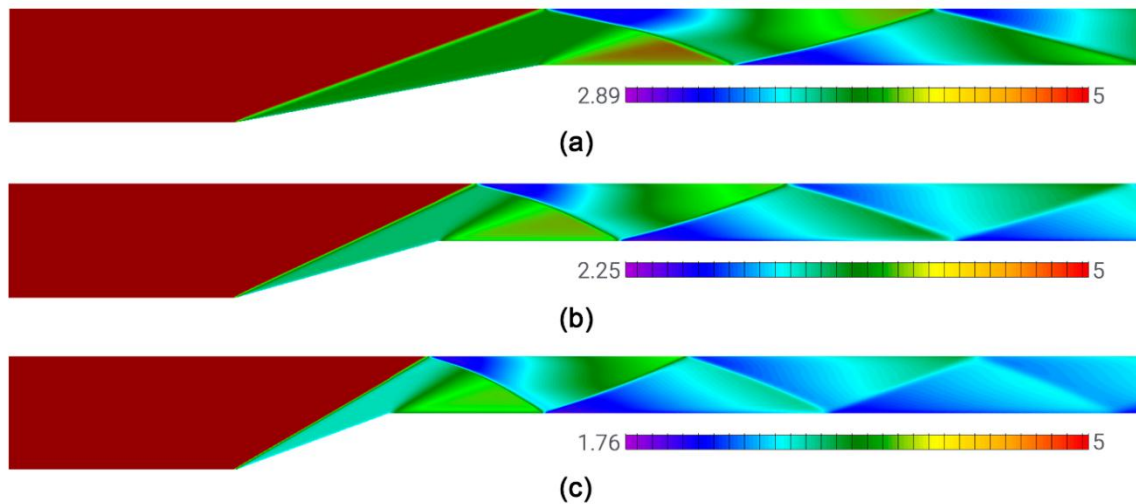


Figure 9. Distribution of Mach number for air intake Type 1 (a), Type 2 (b) and Type 3 (c).

Table 3 shows the calculated α_c and projected α flows deflection angles for all air intake configurations.

Table 3. Angle of the reflected shock wave.

Type of configuration	α	α_c
Type 1	20	20.13
Type 2	25	25.16
Type 3	30	30.09

It can be seen that the results of the calculation are in good agreement with the analytical procedure (12).

5. Conclusion

The method of calculation of ideal gas flows on an arbitrary computational mesh is developed. An algorithm for determining the geometric parameters of an arbitrary computational cell is presented. The data storage system, solver algorithm, and the algorithm for coupling computational meshes are shown. Calculations of gas flow in a supersonic air intake are carried out. The calculation results are in good agreement with the analytical method [16]. The application of the proposed methodology and calculation algorithm to arbitrary computational mesh with a mixed cell type makes it possible to optimize the process of constructing computational mesh and conduct numerical studies of gas dynamics in regions of complex geometry.

Acknowledgments

This study (research grant No 8.2.12.2018) was supported by The Tomsk State University competitiveness improvement program.

References

- [1] Vasenin I M, Shrager E R, Glazunov A A, Krainov A Yu and Krainov D A 2013 Modeling of the Process of Motion of a Scramjet in the Atmosphere *Russ Phys J.* **56** 908-913 DOI: 10.1007/s11182-013-0117-2
- [2] Burkin V V, Diachkovsky A S, Ishchenko A N, Kasimov V Z and Rogaev K S 2015 Investigation of gas-dynamic features of a moving model with a pulse jet engine (CPJE) inside a barrel *Tomsk State University Journal of Mathematics and Mechanics* [Vestnik Tomskogo Gosudarstvennogo Universiteta, Matematika i Mekhanika – in Russian] **35** 45-51 DOI 10.17223/19988621/35/6
- [3] Minkov L L, Shrager E R and Kiryushkin A E 2016 Two Approaches for Simulating the Burning Surface in Gas Dynamics. *Key Engineering Materials* **685** 114-118 DOI: 10.4028/www.scientific.net/KEM.685.114
- [4] Rodionov A V 2016 Correlation between the discontinuous Galerkin method and MUSCL-type schemes *Math Models Comput Simul.* **8(3)** 285-300 DOI: 10.1134/S207004821603008X
- [5] Kolgan V P 1972 Application of the principle of minimizing the derivative to the construction of finite difference schemes for computing discontinuous solutions of gas dynamics [Primenenie principa minimal'nyh znachenij proizvodnoj k postroeniyu konechnoraznostnykh skhem dlya raschyota razryvnykh reshenij gazovoj dinamiki – in Russian]. *TsAGI Science Journal* [Uchenye zapiski TsAGI – in Russian] **3(6)** 68-77
- [6] Kolgan V P 2011 Application of the principle of minimizing the derivative to the construction of finite difference schemes for computing discontinuous solutions of gas dynamics *J. Comput. Phys.* **230(7)** 2384-2390 DOI: 10.1016/j.jcp.2010.12.033
- [7] Van Leer B 2011 A historical oversight: Vladimir P. Kolgan and his high-resolution scheme *J. Comp. Phys.* **230(7)** 2378-2383 DOI: 10.1016/j.jcp.2010.12.032
- [8] Van Leer B 1979 Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comp. Phys.* **32** 101-136 DOI: 10.1016/0021-9991(79)90145-1
- [9] Toro E F 2009 *Riemann solvers and numerical methods for fluid dynamics* (Third Edition. Berlin: Springer-Verlag Berlin Heidelberg) DOI: 10.1007/b79761
- [10] Darwish M S and Moukalled F 2003 *International Journal of Heat and Mass Transfer* **46** 599-611
- [11] Barth T and Jespersen D 1989 The design and application of upwind schemes on unstructured meshes *27th Aerospace Sciences Meeting* <https://doi.org/10.2514/6.1989-366>
- [12] Tilliayeva N I 1986 Generalization of the modified Godunov's scheme on arbitrary irregular grids [Obobshcheniye modifitsirovannoy skhemy S.K. Godunova na proizvol'nyye neregulyarnyye setki – in Russian] *TsAGI Science Journal* [Uchenye zapiski TsAGI – in Russian] **17(2)** 18-26
- [13] Kulikovskiy A G, Pogorelov N V and Semenov A. J 2001 *Mathematical Aspects of Numerical Solution of Hyperbolic Systems* (London: Chapman & Hall/CRC)
- [14] Godunov S K, Zabrodin A V, Ivanov M Y, Kraiko A N and Prokopov G P 1976 *Numerical Solution of Multidimensional Problems of Gas Dynamics* [in Russian – Chislennoe reshenie mnogomernykh zadach gazovoj dinamiki] (Moscow: Nauka)
- [15] Toro E F 2009 *Riemann Solvers and Numerical Methods for Fluid Dynamics* (Berlin: Springer-Verlag)
- [16] German R 1959 *Supersonic Inflow Diffusers* (Moscow: AN USSR Publisher)