

FlowModellium Software Package for Calculating High-Speed Flows of Compressible Fluid

M. N. Petrov^a, A. A. Tambova^a, V. A. Titarev^{a,b,*}, S. V. Utyuzhnikov^{a,c,**}, and A. V. Chikitkin^a

^a *Moscow Institute of Physics and Technology, Dolgoprudnyi, Moscow oblast, 141700 Russia*

^b *Federal Research Center “Computer Science and Control,” Russian Academy of Sciences, Moscow, 119333 Russia*

^c *University of Manchester, Manchester, M13 9PL UK*

**e-mail: titarev@ccas.ru*

***e-mail: s.utyuzhnikov@manchester.ac.uk*

Received October 17, 2017

Abstract—The development of the software package FlowModellium designed for simulating high-speed flows of continuum medium taking into account nonequilibrium chemical reactions is described. The numerical method and the two-level parallel algorithm used in the package are presented. Examples of computations are discussed.

Keywords: supersonic flows, computational fluid dynamics, nonequilibrium chemistry, unstructured mesh, TVD scheme, supercomputer computations.

DOI: 10.1134/S0965542518110118

INTRODUCTION

Designing aerial vehicles moving at high supersonic speeds is closely related to the computation of their dynamic and thermal interaction with surrounding air. The experimental investigation of aerodynamic characteristics of aerial vehicles runs into significant technical difficulties and not all airflow regimes can be examined experimentally. To save time and money, it is reasonable to investigate the main thermodynamic and fluid dynamic characteristic using computational fluid dynamics. An important practical task is to create universal software implementing numerical methods for solving models of nonequilibrium chemically active gas. Currently, a number of research teams in Russia are developing universal software packages for simulating the aerodynamics of compressible fluid [1–4]. Among the codes designed for high-speed flows, we mention [5–7].

Currently, the laboratory of mathematical modeling of nonlinear processes in gaseous media of the Moscow Institute of Physics and Technology (Russia) is developing numerical methods and the software package called FlowModellium for simulating external high-speed flows of compressible fluid in a wide range of parameters of the incident flow. The package consists of a mesh generator [8], a modular solver of equations of motion of compressible fluid that takes into account various physical properties of the medium [9], an a visualization module. The use of this software will make it possible to create a simulation environment that includes all phases of solving applied problems—creating a geometric model of the object, generation of mesh in the physical space, computation of airflow on a supercomputer, processing the computation results, and generation of recommendations for developers of aerial vehicles. In this paper, we review the capabilities of the aerodynamic solver included in FlowModellium. Its distinctive feature is the support and use of block structured and hybrid unstructured meshes, the use of an implicit high approximation order scheme on such meshes, and a two-level MPI + OpenMP model of parallel computations adapted for the use on modern supercomputers.

1. BASIC EQUATIONS

Consider external flow around a body by compressible fluid. The state of the fluid at the point $\mathbf{x} = (x_1, x_2, x_3)$ of the physical space at the time t is characterized by its density ρ , velocity $\mathbf{u} = (u_1, u_2, u_3)$, pressure p , and temperature T . We assume that the medium satisfies the perfect gas equation

$$p = \rho \frac{R_u}{M} T, \quad (1)$$

where $R_u \approx 8.31 \text{ J}/(\text{mole} \cdot \text{K})$ is the gas constant, and M is the molar mass. The energy of unit volume of the gas mixture is represented in terms of the mass e , density, and velocity of the gas by the formula

$$E = \rho(e + u_\alpha u_\alpha / 2). \quad (2)$$

In dimensional vector form, the equations are written as

$$\frac{\partial}{\partial t} \mathbf{U} + \nabla(\mathbf{F} - \mathbf{F}^v) = \mathbf{S}(\mathbf{U}), \quad \mathbf{F} = (\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3), \quad \mathbf{F}^v = (\mathbf{F}_1^v, \mathbf{F}_2^v, \mathbf{F}_3^v). \quad (3)$$

Here, \mathbf{U} is the vector of conservative variables, \mathbf{F} and \mathbf{F}^v are the tensors of convective and viscous flows with respect to coordinate directions, and \mathbf{S} is the source term. The specific form of these vectors depends on the physical model. A description of the physical models implemented in the package is given below.

1.1. Laminar Models

The simplest model is the model of perfect gas in which the internal energy of the gas mass unit e in the equation of state (1) is given by

$$e = \frac{RT}{\gamma - 1}, \quad R = \frac{R_u}{M}.$$

The molar mass M and the specific heat ratio γ for air are, respectively, $M = \text{const} \approx 28.98 \times 10^{-3} \text{ kg}/\text{mole}$ and $\gamma = 7/5$. The vectors \mathbf{U} , \mathbf{F}_k , and \mathbf{F}_k^v are given by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \end{pmatrix}, \quad \mathbf{F}_k = \begin{pmatrix} \rho u_k \\ \rho u_1 u_k + \delta_{1k} p \\ \rho u_2 u_k + \delta_{2k} p \\ \rho u_3 u_k + \delta_{3k} p \\ (E + p) u_k \end{pmatrix}, \quad \mathbf{F}_k^v = \begin{pmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \\ u_\alpha \tau_{\alpha k} - q_k \end{pmatrix}. \quad (4)$$

Here summation is made over repeating Greek subscript indexes. The source term is identical zero: $\mathbf{S} \equiv 0$. For the components of the viscous stress tensor τ_{ij} and heat flux q_k , we have:

$$\tau_{ik} = \mu \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} - \frac{2}{3} \text{div} \mathbf{u} \right), \quad q_k = -\lambda \frac{\partial T}{\partial x_k}.$$

For the perfect fluid, the coefficients of viscosity μ and thermal conductivity λ are related by the equations

$$\lambda = \frac{c_p \mu}{\text{Pr}}, \quad c_p = \frac{\gamma R}{\gamma - 1},$$

where $\text{Pr} = 0.75$ is the Prandtl number.

For high speed gas flows in which the temperature can be high, the perfect gas model is inappropriate, and chemical reactions in the gas must be taken into account. A simple model for this case is the equilibrium chemistry model the use of which requires the assumption that the characteristic times of all reactions are large compared with the mean time of free path of particles and that the reaction rates are much lower than the rate of reaching the Maxwell equilibrium due to elastic collisions [10]. The computational equations retain form (4), but the values of the thermodynamic parameters and transport coefficients in the case of taking into account the equilibrium chemical reactions vary depending on the actual values of pressure and density. Thus, by choosing the equilibrium chemistry model for solving the problem, we will actually solve the Euler or Navier–Stokes equations (4) together with the equation of state of the perfect equilibrium chemically reacting gas–plasma mixture.

Table 1. Description of implemented multicomponent models

Number of components N	Composition
5	O, N, O ₂ , N ₂ , NO
7	O, N, O ₂ , N ₂ , NO, NO ⁺ , e
9	O, N, O ₂ , N ₂ , NO, NO ⁺ , O ₂ ⁺ , N ₂ ⁺ , e
11	O, N, O ₂ , N ₂ , NO, O ⁺ , N ⁺ , NO ⁺ , O ₂ ⁺ , N ₂ ⁺ , e

The parameters and transport coefficients are computed by linear interpolation using a table that was preliminary composed based on the working range of pressures 0.01–100 atm and the working range of temperatures 300–20000 K. The interpolation of tabular data gives the maximal error of about 0.5% for the transport coefficients and less than 0.01% for the thermodynamic parameters. The table was constructed using the program package SoVA (Sokolova, Vasil’evskii, and Andriatis) developed for calculating the equilibrium composition and transport coefficients for low-temperature plasma in high-order approximations of the Chapman–Enskog method; this package produces highly accurate tabular data. The basic conditions under which this table was computed are the chemical equilibrium conditions (Guldberg–Waage equations) and the Saha ionization equation. The table used in the software described in the current paper was composed for the 22-component gas–plasma mixture. This table was composed under the assumption that the incident flow is air. However, it can be extended to take into account other components (e.g., for the case of ablation) or replaced by another table if the molar concentrations of the incident flow are different from those characteristic of air.

If the applicability condition of the equilibrium chemical reactions model is not fulfilled, then a non-equilibrium model must be used. In this paper, we use the one-temperature model based on studies [11, 12]. We used 5-, 7-, 9-, and 11-component gas models. The composition of the components is presented in Table 1. The gaseous medium is considered as a perfect quasi-neutral mixture of perfect gases consisting of N components between which nonequilibrium chemical reactions and ionization proceed. The molar mass of the mixture M and the internal gas energy e appearing in Eqs. (1) and (2) are determined by the formulas

$$\frac{1}{M} = \sum_{j=1}^N \gamma_j, \quad e = h - R_u T \sum_{j=1}^N \gamma_j, \quad h = \sum_{j=1}^N \gamma_j h_j. \quad (5)$$

Here $\gamma_j = c_j/m_j$ are the mass-molar concentrations of components (c_j are mass concentrations and m_j is the molar mass of the component); h is the enthalpy of the unit mass of the gas mixture; and e_j and h_j are the molar internal energy and enthalpy of the j th component, respectively. The thermodynamic and thermochemical data are taken from [13].

For the one-temperature (1TM) model of the gas medium, the vector of conservative variables and the vector of fluxes have the form

$$\mathbf{U} = \begin{pmatrix} \rho\gamma_1 \\ \vdots \\ \rho\gamma_N \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_N \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{F}_k = \begin{pmatrix} \rho u_k \gamma_1 \\ \vdots \\ \rho u_k \gamma_N \\ \rho u_1 u_k + \delta_{1k} p \\ \rho u_2 u_k + \delta_{2k} p \\ \rho u_3 u_k + \delta_{3k} p \\ (E + p) u_k \end{pmatrix}, \quad \mathbf{F}_k^v = \begin{pmatrix} \mathbf{K}_{1k} \\ \vdots \\ \mathbf{K}_{Nk} \\ \tau_{xk} \\ \tau_{yk} \\ \tau_{zk} \\ u_\alpha \tau_{\alpha k} - q_k \end{pmatrix}. \quad (6)$$

Here, ω_j are the source terms due to nonequilibrium chemical reactions. The diffusive fluxes \mathbf{K}_j for the j th component and the thermal flux \mathbf{q} are determined by

$$K_{jk} = d_j \frac{\partial \gamma_j}{\partial x_k}, \quad q_k = -\lambda \frac{\partial T}{\partial x_k} + \sum_{j=1}^N K_{jk} h_j.$$

To calculate the viscosity and thermal conductivity coefficients of the gas mixture, approximate Wilke–Vasil’eva formulas [14] are used; τ_{ij} are the ordinary components of the viscous stress tensor.

1.2. RANS Models

To calculate turbulent flows, we use the Spalart–Allmaras (SA) model to close the RANS equations [15–17] and the Menter model (k- ω -SST) [18]. Based on the information about various semi-empirical models, we claim that these models are most appropriate for external fluid dynamics problems; however, in some cases one model can have certain advantages over the other.

The Spalart–Allmaras model combines simplicity (only one additional differential equation) with high potential for taking into account a wide range of turbulent phenomena even in problems with complex geometry. The vectors \mathbf{U} , \mathbf{F}_k , \mathbf{F}_k^v , and \mathbf{S} are given by the formulas (in dimensional form)

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \\ \rho \tilde{v} \end{pmatrix}, \quad \mathbf{F}_k = \begin{pmatrix} \rho u_k \\ \rho u_1 u_k + \delta_{1k} p \\ \rho u_2 u_k + \delta_{2k} p \\ \rho u_3 u_k + \delta_{3k} p \\ (E + p) u_k \\ \rho \tilde{v} u_k \end{pmatrix}, \quad \mathbf{F}_k^v = \begin{pmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \\ u_\alpha \tau_{\alpha k} - q_k \\ \frac{1}{\sigma} \left(\rho (v_{mol} + \tilde{v}) \frac{\partial \tilde{v}}{\partial x_k} \right) \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ S_{sa} \end{pmatrix}. \quad (7)$$

The first five equations are the same as in the perfect gas case. Effectively, the SA model refines the values of the transport coefficients in Eqs. (4) leaving Eqs. (4) themselves intact (4). We assume that the viscosity is given by $\mu = \mu_{mol} + \mu_t$ and the thermal conductivity coefficient is $\lambda = \lambda_{mol} + \lambda_t$. Here μ_{mol} is the molecular viscosity, μ_t is the turbulent viscosity, λ_{mol} is the molecular thermal conductivity coefficient, and λ_t is the turbulent thermal conductivity coefficient. The turbulent viscosity is determined by $\mu_t = \rho \tilde{v} f_{v1}$.

There are a lot of modifications of the basic SA model that take better account of the physics of the phenomenon or improve the stability of computations [19–23]. In this paper, we use the modification SA-nonft2. In this modification, the sixth component of the source term has the form

$$S_{sa} = c_{b1} \rho \tilde{S} \tilde{v} - c_{w1} f_w \rho \left(\frac{\tilde{v}}{d} \right)^2 + c_{b2} \frac{\rho}{\sigma} \frac{\partial \tilde{v}}{\partial x_k} \frac{\partial \tilde{v}}{\partial x_k}.$$

The other terms appearing in the source are given by the formulas

$$\tilde{S} = S + \frac{\tilde{v}}{k^2 d^2} f_{v2}, \quad S = \sqrt{2 W_{ij} W_{ij}}, \quad W_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right),$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi = \frac{\tilde{v}}{v_{mol}}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6},$$

$$g = r + C_{w2} (r^6 - r), \quad r = \min \left(\frac{\tilde{v}}{\tilde{S} k^2 d^2}, 10 \right).$$

The statement of the boundary value problem assumes that the turbulent viscosity on the wall is zero. In the incident flow, the turbulent viscosity is chosen from the range 3–5 laminar viscosities at infinity.

The turbulent thermal conductivity is calculated as $\lambda_t = c_p \mu_t / Pr_t$, where the turbulent Prandtl number Pr_t is equal to 0.85.

As another model of closure of the RANS equations, we use the two-parameter Menter model. In this model, the vectors \mathbf{U} , \mathbf{F}_k , \mathbf{F}_k^v , and \mathbf{S} have the form

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \\ \rho k \\ \rho \omega \end{pmatrix}, \quad \mathbf{F}_k = \begin{pmatrix} \rho u_k \\ \rho u_1 u_k + \delta_{1k} \left(p + \frac{2}{3} \rho k \right) \\ \rho u_2 u_k + \delta_{2k} \left(p + \frac{2}{3} \rho k \right) \\ \rho u_3 u_k + \delta_{3k} \left(p + \frac{2}{3} \rho k \right) \\ \left(E + p + \frac{2}{3} \rho k \right) u_k \\ \rho k u_k \\ \rho \omega u_k \end{pmatrix}, \quad \mathbf{F}_k^v = \begin{pmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \\ u_\alpha \tau_{\alpha k} - q_k \\ (\mu_{mol} + \sigma_k \mu_t) \frac{\partial k}{\partial x_k} \\ (\mu_{mol} + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_k} \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ S_k \\ S_\omega \end{pmatrix}. \quad (8)$$

Here, k is the turbulent energy and ω is its dissipation. The system of equations is closed by the equation of state

$$p = (\gamma - 1) \left[E - \rho \frac{1}{2} (u^2 + v^2 + w^2) - \rho k \right].$$

Under the Boussinesq hypothesis, the stress tensor components are written as

$$\tau_{ij}^{full} = \tau_{ij}^{lam} + \tau_{ij}^{turb},$$

$$\tau_{ij}^{lam} = 2\mu_{mol} \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad \tau_{ij}^{turb} = 2\mu_t \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij}.$$

In turn, the components τ_{ij} in the vector \mathbf{F}_k^v have the form

$$\tau_{ij} = 2(\mu_{mol} + \mu_t) \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right),$$

and the term $-\frac{2}{3} \rho k \delta_{ij}$ is moved to the convective flow.

In this model, the turbulent viscosity is calculated by the formula

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega, S F_2)}, \quad S = \sqrt{2 S_{ij} S_{ij}}, \quad S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

The source terms for the turbulent energy and its dissipation are calculated by

$$S_k = P - \beta^* \rho \omega k, \quad S_\omega = \frac{\gamma}{\nu_t} P - \beta \rho \omega^2 + 2(1 - F_1) \frac{\rho \sigma_{\omega_2}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j},$$

where

$$P = \tau_{ij}^{turb} \frac{\partial u_i}{\partial x_j} = \mu_t \left(S^2 - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \frac{\partial u_k}{\partial x_k} \right) - \frac{2}{3} \rho k \frac{\partial u_k}{\partial x_k}, \quad \phi = F_1 \phi_1 + (1 - F_1) \phi_2,$$

$$F_1 = \tanh(\arg_1^4), \quad \arg_1 = \min \left[\max \left(\frac{\sqrt{k}}{\beta^* \omega d}, \frac{4 \rho \sigma_{\omega_2} k}{CD_{kw} d^2} \right) \right],$$

$$CD_{kw} = \max \left(2 \rho \sigma_{\omega_2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j}, 10^{-20} \right),$$

$$F_2 = \tanh(\arg_2^2), \quad \arg_2 = \max \left(2 \frac{\sqrt{k}}{\beta^* \omega d}, \frac{500 \nu}{d^2 \omega} \right)$$

Here, d is the distance to the body surface. The boundary conditions on the hard wall are written as (δd_1 is the size of the first cell)

$$k = 0, \quad \omega = 10 \frac{6\nu}{\beta_1(\delta d_1)^2}.$$

For infinitely far boundaries, the following formulas are used:

$$\frac{U_\infty}{L} < \omega < 10 \frac{U_\infty}{L}, \quad \frac{10^{-5} U_\infty^2}{Re_L} < k < \frac{0.1 U_\infty^2}{Re_L}.$$

The turbulent thermal conductivity is calculated as in the Spalart–Allmaras model but with the Prandtl number $p_{r_t} = 0.9$.

2. NUMERICAL METHOD

Numerical solution is always constructed in dimensionless variables. The transition to dimensionless variables is specified by choosing the scales of the coordinate L_* , pressure p_* , density ρ_* , velocity v_* , temperature T_* , molar mass m_* , and viscosity μ_* . By default, the length scale is $L_* = 1$ m, but it can also be specified in the program configuration file. In all cases, the scale of the thermal conductivity coefficient is $\lambda_* = \mu_* v_*^2 / T_*$. To choose the other scales, three variants are possible. For the nonequilibrium model, the choice of scales is determined by the concrete implementation of the library of chemical reactions:

$$p_* = 1 \text{ atm}, \quad \rho_* = 1/820.52 \text{ kg/m}^3, \quad v_* = 9118 \text{ m/s},$$

$$T_* = 1000 \text{ K}, \quad m_* = 1 \text{ g/mole}, \quad \mu_* = 3.204 \times 10^{-4} \text{ kg/(m s)}.$$

For the two other variants, the scales of density, temperature, and viscosity are determined by their values in the incident flow. For the scales of velocity and pressure for high speed incident flows, we used $v_* = U_\infty$, $p_* = \rho_* v_*^2$. If the incident flow velocity is low, then we used $v_* = \sqrt{RT_*}$, $p_* = p_\infty$. The equation of state (1) takes the form $p = \rho TS_*$. For flows in which there are chemical reactions, the coefficient $S_* = 1/M$ is recalculated at each point of the space. For the other cases, $S_* = RT_*/v_*^2$ is a constant.

In terms of dimensionless variables, Eqs. (3) take the form

$$\frac{\partial}{\partial t} \mathbf{U} + \nabla \left(\mathbf{F} - \frac{1}{Re_*} \mathbf{F}^v \right) = \mathbf{S}(\mathbf{U}), \quad Re_* = \frac{\rho_* v_* L_*}{\mu_*}. \quad (9)$$

Here the Reynolds number Re_* is formally obtained when we change to the dimensionless variables, and it can be different from the actual Reynolds number of the problem. The steady-state solution to system (9) is constructed using the relaxation method and the second-order Godunov's finite volume scheme.

Let us define a computational mesh consisting of cells V in the space of spatial variables. Each cell can be tetrahedral, pyramidal, hexahedral, or prismatic, and in can be formed by several triangular or quadrilateral faces a_{li} . The total number of cells is N_{tot} . By integrating system (9) and passing to discrete form, we obtain

$$\begin{aligned} \frac{d}{dt} \mathbf{U}_i &= \mathbf{R}_i, \quad \mathbf{R}_i = -\frac{1}{|V_i|} \sum_l (\Phi_{li} - \Phi_{li}^v) + S_i, \\ \Phi_{li} &= \int_{a_{li}} \mathbf{n}_l \cdot \mathbf{F} da, \quad \Phi_{li}^v = \frac{1}{Re_*} \int_{a_{li}} \mathbf{n}_l \cdot \mathbf{F}^v da. \end{aligned} \quad (10)$$

Here, \mathbf{U}_i is the average value over the cell, Φ_{li} is the convective (inviscid) numerical flux through the face l , Φ_{li}^v is the viscous numerical flux through the face l , and \mathbf{n}_l is the external unit normal to the face l . Note that $\mathbf{R}_i \equiv 0$ on the steady-state solution.

The computation of the convective numerical flux Φ_{li} in scheme (10) begins with building a reconstructing polynomial for each cell V_i . This polynomial approximates the vector \mathbf{U} within the cell [24–27]. In the first method, given the average values, the values on the cell faces are constructed using a multidimensional solution reconstruction procedure. To ensure the second order of approximation, a piecewise linear representation of the distribution function in each spatial cell is built; the coefficients of this representation are computed using the least squares method by the values in the cells of the reconstruction stencil. The other method is applicable only to hexahedral cells—it is a locally one-dimensional method of computing the values of the distribution function on faces borrowed from the reconstruction methods of structured meshes.

The reconstruction procedure produces two values of the vector of conservative variables on each face l of the spatial cell i ; the first value, which is denoted by \mathbf{U}^- below, corresponds to the reconstruction polynomial of the cell i . The second value \mathbf{U}^+ is obtained when the polynomial from the cell $\sigma_l(i)$ adjacent to the face l of the cell i is used. In the general case, these two values are different, and the Riemann problem on the face naturally arises. A detailed description of methods for finding its exact and approximate solutions using the property of invariance of the convective flux under rotation of the coordinate system can be found, e.g., in [28, 29].

Let T_{li} be the matrix of rotation of the local system of coordinates to the local system of face l . The convective numerical flux Φ_{li} in scheme (10) has the form (see [29])

$$\begin{aligned} \Phi_{li} &= \int_{a_{li}} T_{li}^{-1} \mathbf{F}(T_{li} \mathbf{U}) da = T_{li}^{-1} \mathbf{f}(\mathbf{Q}^-, \mathbf{Q}^+) |a_{li}|, \\ \mathbf{Q}^- &= T_{li} \mathbf{U}_i = T_{li} \mathbf{U}^-, \quad \mathbf{Q}^+ = T_{li} \mathbf{U}_{\sigma_l(i)l_1} = T_{li} \mathbf{U}^+. \end{aligned} \quad (11)$$

Here, the index of the time step n is omitted for simplicity. The quantity \mathbf{Q} is the vector of conservative variables in the system of coordinates oriented perpendicular to the face, the function \mathbf{f} is the Riemann solver, and l_1 is the index of the face of cell $\sigma_l(i)$ adjacent to the face l of cell i . In this paper, we use generalizations of several Riemann solvers (HLL [30], HLLC [29, 31], Osher's method [32], and exact solver) to the case of chemically reacting gas.

Pay attention how the Riemann solver HLLC for the model k - ω -SST is implemented. In this implementation, the terms appearing due to the Boussinesq hypothesis are approximated by the upwind formula based on the sign of velocity of the central wave and added to the standard value of HLLC flux for the perfect gas [33, 34].

It is well known that when Godunov-type capturing schemes are used, the undesirable effect called the carbuncle phenomenon can occur (e.g., see [35, 37]). In this paper, for the calculation of flows with the Mach number of the incident flow $M_\infty > 10$ and in order to fight this phenomenon, we use the flux HLL on all mesh faces that are father from the body surface than a given distance l_0 ; on the other faces the more accurate three-wave HLLC flux is used. The parameter l_0 must be such that HLLC is used on all faces of the boundary layer. Typically, if the body size is about 1 m, then $l_0 = 1$ cm.

The approximation of viscous numerical fluxes in the solution of the Navier–Stokes equations requires the computation of derivatives of macroscopic quantities (first of all, velocity components and temperature) on cell faces. In the numerical method, we use the direct approximation of the derivatives based on the values of the solution at the centers of the adjacent cells and at the face vertices. This method was proposed in [38, 39] for tetrahedral meshes with triangular faces. In the package described in this paper, we generalize this method to quadrilateral faces. The unknown values at the face vertices are found by averaging over the cells adjacent to the face with geometric weights that are inversely proportional to the distance from the vertex to the cell center.

Our implementation of the implicit version of the numerical method is based on the studies [40–43]. By approximating the time derivative in the semidiscrete scheme (10) by backward differences, we obtain the following implicit one-step scheme on an arbitrary spatial mesh:

$$\frac{\Delta \mathbf{U}_i}{\Delta t} = \mathbf{R}_i^{n+1}, \quad \Delta \mathbf{U}_i = \mathbf{U}_i^{n+1} - \mathbf{U}_i^n.$$

The linearization with respect to time yields the following value in cell i :

$$\left(\mathbf{I} - \Delta t \frac{\partial \mathbf{R}_i^n}{\partial \mathbf{U}} \right) \Delta \mathbf{U}_i = \Delta t \mathbf{R}_i^n \quad (12)$$

During the numerical flux linearization, we take into account the dependence of the flux on the values \mathbf{U} in cell i and in its immediate neighbor $\sigma_l(i)$ on the face l , which corresponds to the upwind spatial first-order approximation. The numerical fluxes on the left-hand side of scheme (12) are approximated using the modified Rusanov flux (see [43])

$$\begin{aligned} f(\mathbf{Q}^-, \mathbf{Q}^+) &= \frac{1}{2}(\mathbf{F}(\mathbf{Q}^-) + \mathbf{F}(\mathbf{Q}^+)) - \frac{1}{2} \hat{v}(\mathbf{Q}^+ - \mathbf{Q}^-), \\ \hat{v} &= \max_k (|\lambda_k(\mathbf{Q}^-)|, |\lambda_k(\mathbf{Q}^+)|) + \frac{1}{\text{Re}_*} \frac{(\mu^- + \mu^+)}{(\rho^- + \rho^+)h}. \end{aligned} \quad (13)$$

Here, λ_k are the eigenvalues of the convective flux Jacobian matrix, and h is the distance between the cell centers x^+ and x^- found by the formula (x^{face} are the coordinates of the face center)

$$h = \|x^+ - x^{\text{face}}\| + \|x^- - x^{\text{face}}\|. \quad (14)$$

The use of expression (14) instead of the conventional formula $h = \|x^+ - x^-\|$ allows us to take into account kinks of the mesh lines in the computation of airflow around bodies of complex shape.

Upon certain transformations, the implicit scheme (12) takes the final form

$$\begin{aligned} D_i \Delta \mathbf{U}_i + \frac{\Delta t}{2|V_i|} \sum_l (T_{li}^{-1} J(T_{li} \mathbf{U}_{\sigma_l(i)}) T_{li} - \hat{v}_{li}) \Delta \mathbf{U}_{\sigma_l(i)} |a_{li}| &= \Delta t \mathbf{R}_i^n \\ D_i &= 1 + \frac{\Delta t}{2|V_i|} \sum_l \hat{v}_{li} |a_{li}|, \quad J(\mathbf{U}) = \partial \mathbf{F}_1(\mathbf{U}) / \partial \mathbf{U}. \end{aligned} \quad (15)$$

Here, J is the Jacobian matrix of the physical flux \mathbf{F}_1 and T_{li} is the rotation matrix of the coordinate system to the local system of face l of the cell V_i (e.g., see [29]). The resulting sparse system of linear equations (15) can be solved by the approximate LU-SGS factorization (see [41–43]). Rewrite the system of equations (15) in the symbolic form

$$A \Delta \mathbf{U} = \mathbf{b}. \quad (16)$$

The diagonal of the matrix A is occupied by D_i , and all off-diagonal nonzero elements have an order Δt . We decompose this matrix into diagonal, strictly lower triangular, and strictly upper triangular matrices: $A = L + D + U$. Instead of system (16), we consider the system

$$(D + U)D^{-1}(D + L)\Delta \mathbf{U} = \mathbf{b} + \mathbf{S}, \quad \mathbf{S} = LD^{-1}U\Delta \mathbf{U}. \quad (17)$$

The matrix of this system is the product of lower triangular, diagonal, and upper triangular matrices. The quantity $\mathbf{S} = O(\Delta t^2)$ is usually neglected. The simplification $\mathbf{S} = 0$ gives an approximate system that is easily solved using forward and backward elimination. Passing to the version of the method that does not require the calculation of the Jacobian matrix

$$J(T_{li} \mathbf{U}_{\sigma_l(i)}) T_{li} \Delta \mathbf{U}_{\sigma_l(i)} \approx \mathbf{F}_1(T_{li} \mathbf{U}_{\sigma_l(i)} + T_{li} \Delta \mathbf{U}_{\sigma_l(i)}) - \mathbf{F}_1(T_{li} \mathbf{U}_{\sigma_l(i)}) = \Delta \mathbf{F}_{li},$$

we obtain the following two-step algorithm:

1. Backward elimination: $i = N_{\text{tot}}, N_{\text{tot}} - 1, \dots, 1$.

$$D_i \Delta \mathbf{U}_i^* = -\frac{\Delta t}{2|V_i|} \sum_{l: \sigma_l(i) < i} \left(T_{li}^{-1} \Delta \mathbf{F}_{li} - \hat{v}_{li} \Delta \mathbf{U}_{\sigma_l(i)}^* \right) |a_{li}| + \Delta t \mathbf{R}_i^n. \quad (18)$$

2. Forward elimination: $i = 1, 2, \dots, N_{\text{tot}}$.

$$D_i \Delta \mathbf{U}_i = \Delta \mathbf{U}_i^* - \frac{\Delta t}{2|V_i|} \sum_{l: \sigma_l(i) > i} \left(T_{li}^{-1} \Delta \mathbf{F}_{li} - \hat{v}_{li} \Delta \mathbf{U}_{\sigma_l(i)} \right) |a_{li}|. \quad (19)$$

The numerical solution is considered convergent to the steady-state mode when the norm of the right-hand side vector becomes less than a given small number.

3. SOFTWARE PACKAGE

The numerical methods described above are implemented in the software developed in the laboratory of mathematical modeling of nonlinear processes in gaseous media of the Institute of Physics and Technology (Russia) [9]. The programs are written in Fortran 2008 that has some object-oriented features. During the development, the requirements of portability and compatibility with different operating systems (Windows 7 and Linux) and capability to run on various computer systems from workstations to supercomputers were taken into account. To make the implementation portable and compatible, we did not use solutions depending on the operating system and specific architecture of computer nodes. The parallel configuration uses MPI for models with distributed memory and OpenMP for models with shared memory. For the development used the development environment Microsoft Visual Studio 2013 and the compiler Intel Fortran 16. The package Git was used for version control.

The package consists of the computational core, mesh preprocessor, and main aerodynamic solver.

The computational core consists of a set of modules implementing the basic. The core implements the procedures of reading spatial meshes in the formats Neutral and StarCD and extracts information about the mesh connectedness (neighbors of faces, cells, and vertices). To approximate the transport operator, algorithms for the reconstruction of scalar functions by the least squares method on an arbitrary mesh (up to the fifth approximation order, inclusive) and algorithms of nonlinear limitation of solution of the type total variation diminishing (TVD). To perform parallel computations using MPI, additional data structures are created and algorithms for the concurrent input–output of the solution to a restart file based on MPI I/O procedures are implemented. The core also includes procedures for the output of spatial and surface data in the Tecplot format.

To solve problems with a large number of computational cells, the package implements a two-level model of parallel computations that has been intensively developed in recent years as applied to fluid dynamics computations [2, 44–46]. In the single-level model, the number of MPI processes N_{mpi} equals the number of physical cores N_{core} . In the two-level model, the number of MPI processes equals the number of employed nodes or processors. In other words, one or two MPI processes and a given number of OpenMP threads $N_{\text{omp}} = N_{\text{core}}/N_{\text{mpi}}$ are run on each node. As before, the MPI interface is used to exchange data between processors or nodes of the supercomputer.

On the upper level (MPI data exchanges), the spatial mesh is decomposed into blocks with certain overlapping taking into account the difference scheme stencil. The preprocessor included in the package is used to create such blocks. The cells of the original single-block unstructured mesh are distributed among processors using the program Metis [47] or the decomposition module described in [48]. Using this decomposition, the preprocessor distributes the mesh cells among processors, renumbers all mesh elements (nodes, faces, and cells), ensures overlapping of blocks needed for the computations, and creates lists of cells for exchanging data between all processors. All the required data is written in blocks so that each thread in the course of parallel computations reads only the part of the mesh assigned to it.

On the lower level within a cluster node, OpenMP is used. At the initialization of computations, the part of the spatial mesh corresponding to an MPI process is decomposed into N_{omp} nonoverlapping blocks. For the difference scheme used in the computations, which stores data in the nodes, there is no data conflict; therefore, the use of a complex two-level decomposition of the computation domain of physical variables proposed in [4, 45] does not decrease the computation time. To compute the numerical fluxes and source, the simple *parallel do* with a dynamic scheduler is used. The size of dynamically distributed tasks (chunks) was a quarter of the loop size divided by the number of threads. In the backward (18) and forward (19) steps of the LU–SGS algorithm, each thread examines only the cells within its block; therefore, the acceleration of this part of the method is limited only by the RAM access speed.

Strictly speaking, the sequential version of the algorithm cannot be directly parallelized because the blocks of the computational mesh overlap. On the whole, each thread performs computations in its part of the spatial mesh; for ghost cells, the Jacobi approximation (see [49]) is used:

$$\Delta U_i = R_i/D_i. \quad (20)$$

If good quality meshes are used, the rate of convergence to the steady-state solution does not decrease. For poor meshes or when the computations are performed on systems with a large number of cores per

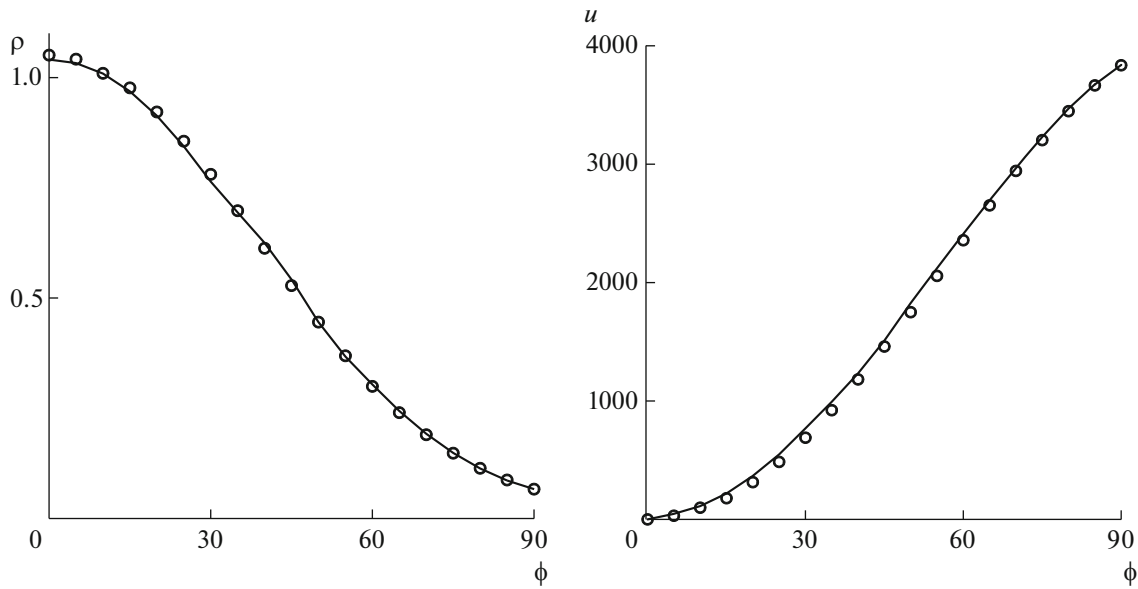


Fig. 1. Comparison of computed (lines) and tabular (open dots, book [54]) values of dimensionless gas density and velocity along the sphere surface depending on the angle ϕ counted from the stagnation point.

node (e.g., based on Intel Xeon Phi), a more reliable modification of LU-SGS based on the recursive decomposition of the computation domain can be used (see [50]).

The use of the two-level model OpenMP + MPI makes it possible to improve the efficiency of computations on systems with a large number of cores per node because the number of MPI processes is reduced by a factor of N_{omp} . However, the use of OpenMP within a node has certain specific features and involves some difficulties (e.g., see the discussion in [45]). The optimal combination of the number of MPI processes N_{mpi} and OpenMP threads N_{omp} depends on the size of problem to be solved and on the number of used processors.

4. EXAMPLES OF COMPUTATIONS

In this section, we describe the results illustrating the scalability of the software package on high performance computers and verification results. In all computations, we used the implicit second-order TVD scheme together with the Riemann solver HLLC or a combination of HLL and HLLC. The major part of computations were performed on the cluster of the laboratory of mathematical modeling of nonlinear processes in gaseous media of the Institute of Physics and Technology and on the system Lomonosov-2 installed in the Research Computer Center of Moscow State University [51, 52] in the framework of the program, *Supercomputer Potential of the Russian Industry* (<http://superprom.hpc-russia.ru>). The code scalability was tested on RSK systems *PetaStream* [53] installed in the Joint Supercomputer Center (JSCC) of the Russian Academy of Sciences, Peter the Great St. Petersburg Polytechnic University.

4.1. Problems with Simple Geometry

Gas flow around a sphere is a classical benchmark problem. Since we are not interested in the reverse flow behind the body, in our computations we used a model geometry consisting of a half-sphere joint to a cylinder of the same radius.

The first series of computations was performed for the perfect inviscid gas model. In each computation, the geometry was flown around by a 22-component inviscid air. The computation results were compared with the tabular data presented in [54]. A mesh consisting of 500 thousand cells without the boundary layer and the second-order scheme were used. Figure 1 shows the plots of the mixture density and velocity on the along the sphere surface at the altitude of 20 km compared with the tabular data (open dots). The parameters of the incident flow were as follows: pressure 5526.87 Pa, density 0.0887 kg/m^3 , and velocity 5901.4 m/s (the Mach number 20). The computation results and the tabular data are very close, which

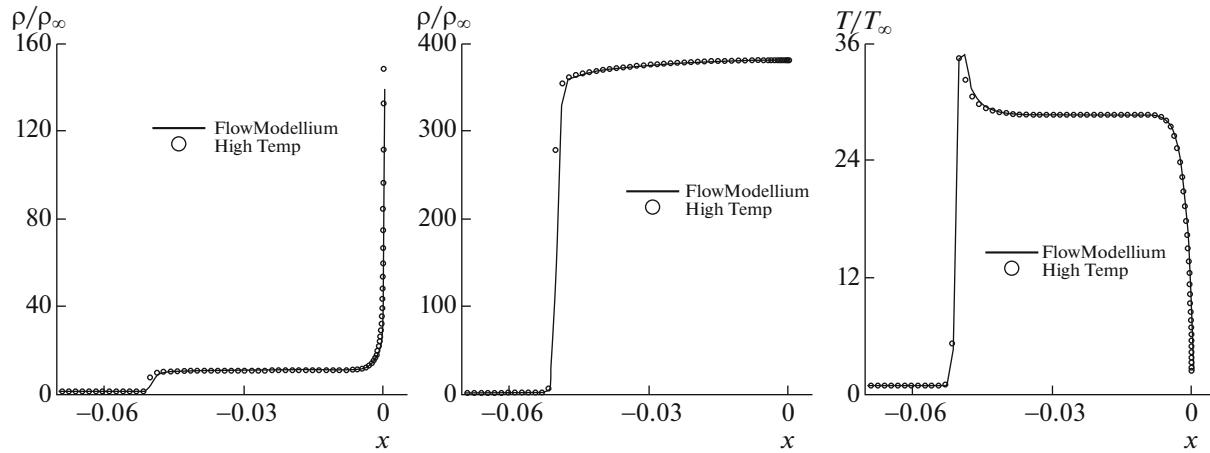


Fig. 2. Comparison of the dimensionless density ρ/ρ_∞ , pressure p/p_∞ , and temperature T/T_∞ on the stagnation line. The data for comparison was obtained using the axisymmetric computations by the HighTemp code [11].

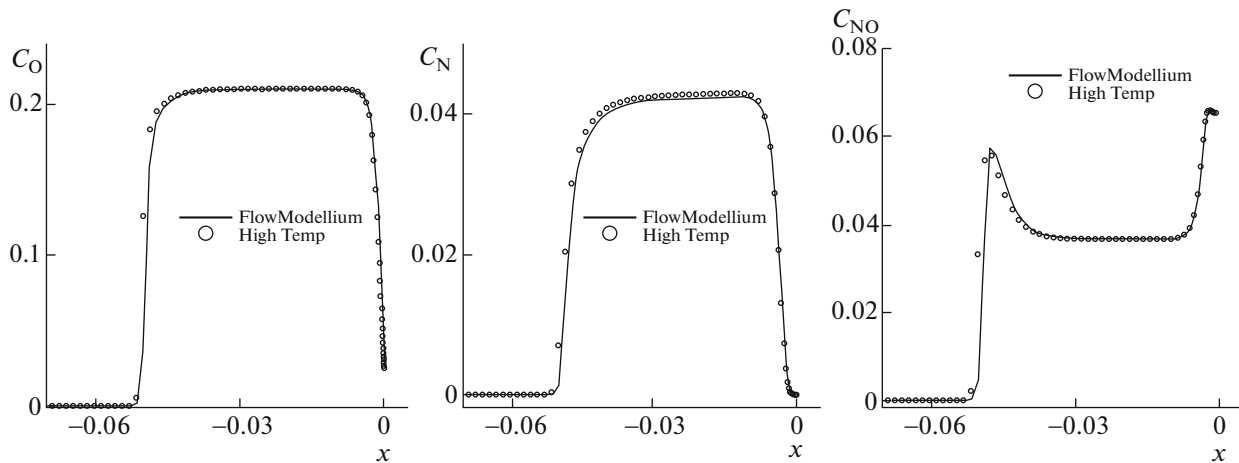


Fig. 3. Comparison of the mass fractions of O, N, and NO on the stagnation line. The data for comparison was obtained using the axisymmetric computations by the HighTemp code [11].

indicates that the implementation of the chosen model is correct. The relative error of density and pressure compared with the tabular data along the sphere surface and on the diffuence line does not exceed 2%. In the vicinity of the stagnation point, the relative error is less than 1%.

In the second series of computations, we considered the flow based on the five component model. The sphere radius was 0.66 cm, and the parameters of the incident flow were as follows: pressure density 0.0645 kg/m^3 and velocity 5022.4 m/s. It was assumed that in the incident flow $c_{O_2} = 0.232$, $c_{N_2} = 0.768$, and the mass fractions of O, N, and NO were zero. The temperature of the sphere surface was $T_w = 550 \text{ K}$, and the surface was assumed to be non-catalytic ($\partial c_k / \partial n = 0$). The corresponding Mach number of the incident flow was $M_\infty = 16.89$. The surface geometry was described by five blocks with the configuration of the type O-mesh; in the center (on the symmetry axis), the square block was used. Thus, the computational mesh was not axisymmetric. The Courant number was 100. Our results were compared with the axisymmetric computations provided by the HighTemp code [11] on a very fine mesh. Note that the computation domain in the axisymmetric solution was an angle of size 70 degrees. Figures 2 and 3 compare the basic computed quantities on the stagnation line. The results are in excellent agreement. On the whole, the typical flow pattern with a bow shock wave and a boundary layer is observed. The gas heating on the shock wave front initiate chemical reactions as a result of which the nitrogen and oxygen molecules dis-

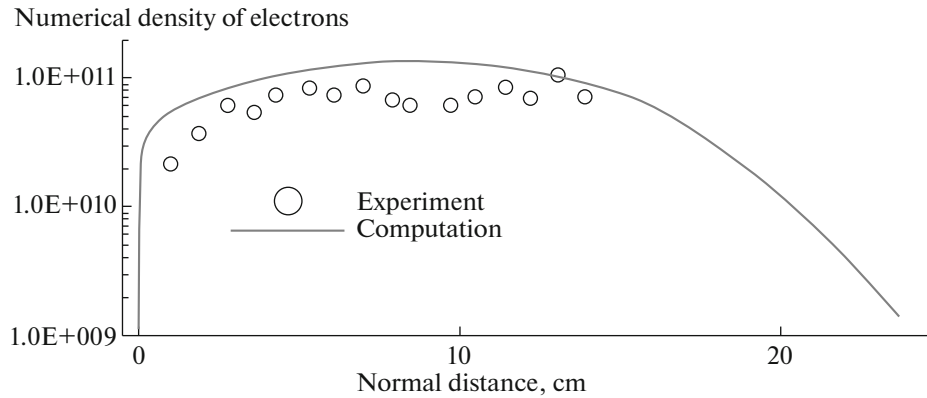


Fig. 4. Comparison of the numerical density of electrons in the perpendicular direction to the body surface at the point $s/R_n = 8.8$ in the computations of RAM-C airflow.

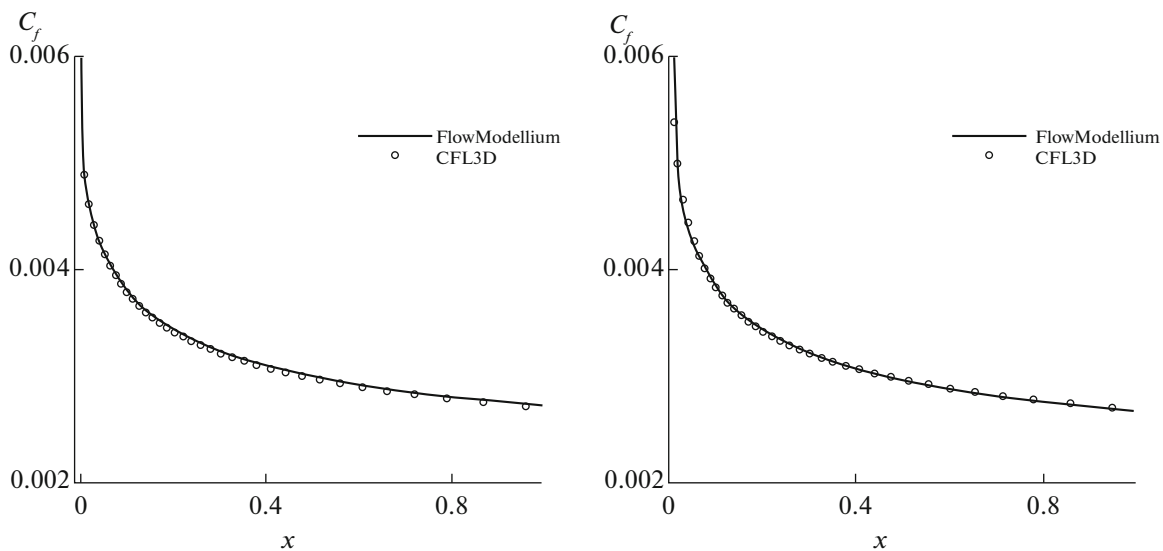


Fig. 5. Distribution of the drag coefficient c_f on the plate; comparison of results obtained by FlowModellium and CFL3D. The SA model is on the left and, k - ω -SST model is on the right.

tegrate, atomic N and O, as well as NO are formed. In the boundary layer, the temperature dramatically decreases to the given surface temperature, as a result of which the mass concentrations of molecular nitrogen and oxygen are restored. Note that the 3D computation reproduces such fine flow details as non-monotonicity of temperature and C_{NO} profiles on the shock wave front.

In the next series of computations we considered the laminar flow around a cone with the half-aperture of 9 degrees, the radius of spherical bluntness $R_n = 0.1524$ m, and the parameters of the incident flow $T_\infty = 218.8$ K, $U_\infty = 8059$ m/s, and surface temperature $T_w = 1000$ K. The 3D computational mesh consisted of about 600 thousand hexahedra with the step in the direction perpendicular to the surface $h_n \approx 10^{-4}$ m. The model 1TM with seven components and the boundary condition of the completely catalytic surface was used. Under this condition, $c_{n_2} = 0.23456$, $c_{o_2} = 0.76544$ on the wall, and the mass fractions of the components were zero. Figure 4 compares our data with the computed and experimental data presented in [55]. The numerical densities of electrons depending on the distance to the surface for the point $s_w/R_n = 8.8$ are compared. It is seen that our data are in good agreement with the experimental data.

Now we test the implementation of the turbulence model.

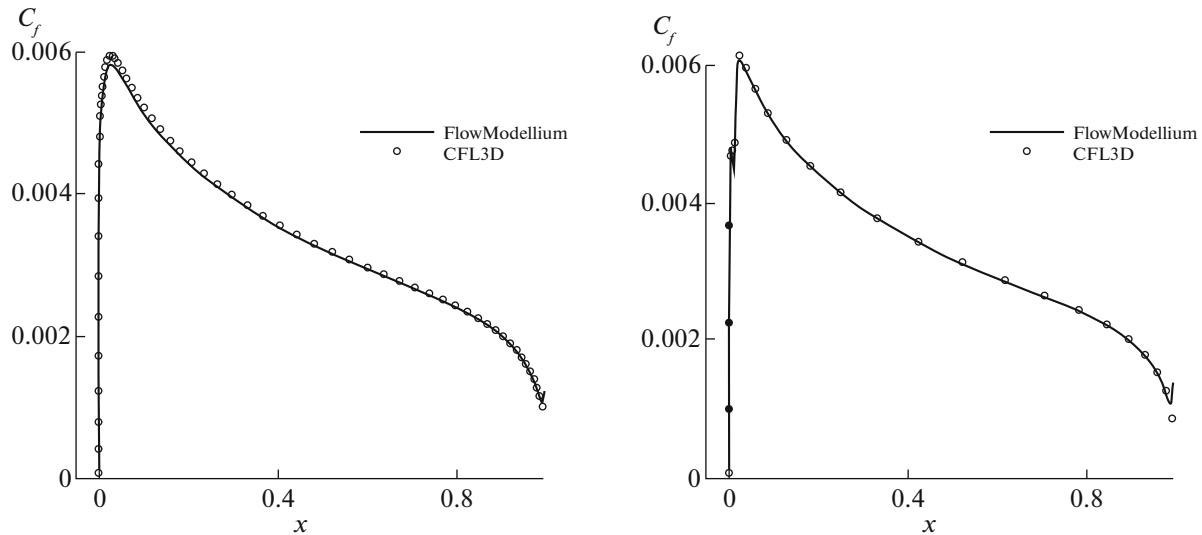


Fig. 6. Distribution of the drag coefficient c_f on the wing airfoil NACA0012; comparison of results obtained by FlowModellium and CFL3D. The SA model is on the left and, k - ω -SST model is on the right.

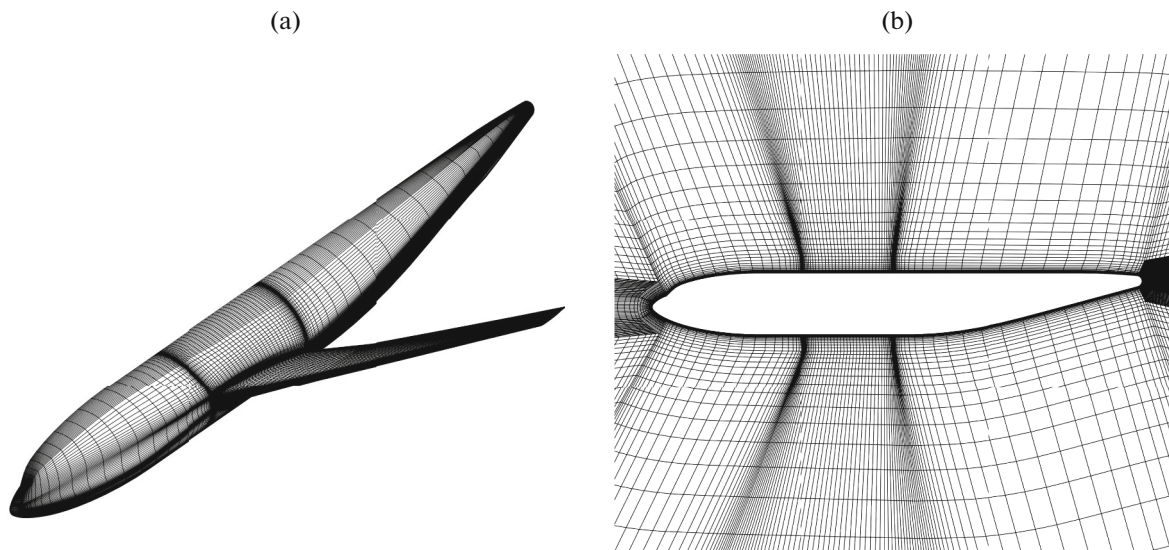


Fig. 7. Computational scheme for the model DLR F6: (a) surface mesh; (b) cut of the mesh near the fuselage.

As test cases, we used two classical benchmarks—flow around a plate and NACA0012 airfoil presented on the site <http://turbmodels.larc.nasa.gov>. The computation of the flow around a plate is the classical test for validating different models. 2D Zero Pressure Gradient Flat Plate is the subsonic flow of compressible perfect gas ($Re_\infty = 5 \times 10^6$, $M = 0.2$, adiabatic wall) around a flat plate. The results of computations within the models SA and k - ω -SST obtained by FlowModellium are compared with the results obtained by the package CFL3D. The computations were performed on a hexahedral mesh refining in the vicinity of the plate (the minimal step is $y = 5 \times 10^{-7}$). Figure 5 plots the drag coefficient obtained by FlowModellium and by CFL3D (the data is taken from the NASA site). The difference in the results is less than one percent.

The next test is the flow around the NACA0012 airfoil. 2D NACA 0012 Airfoil Validation Case is the flow of compressible perfect gas with a zero angle of attack ($Re_\infty = 6 \times 10^6$, $M = 0.15$, adiabatic surface) around the airfoil. Again, the results of computations within the models SA and k - ω -SST obtained by

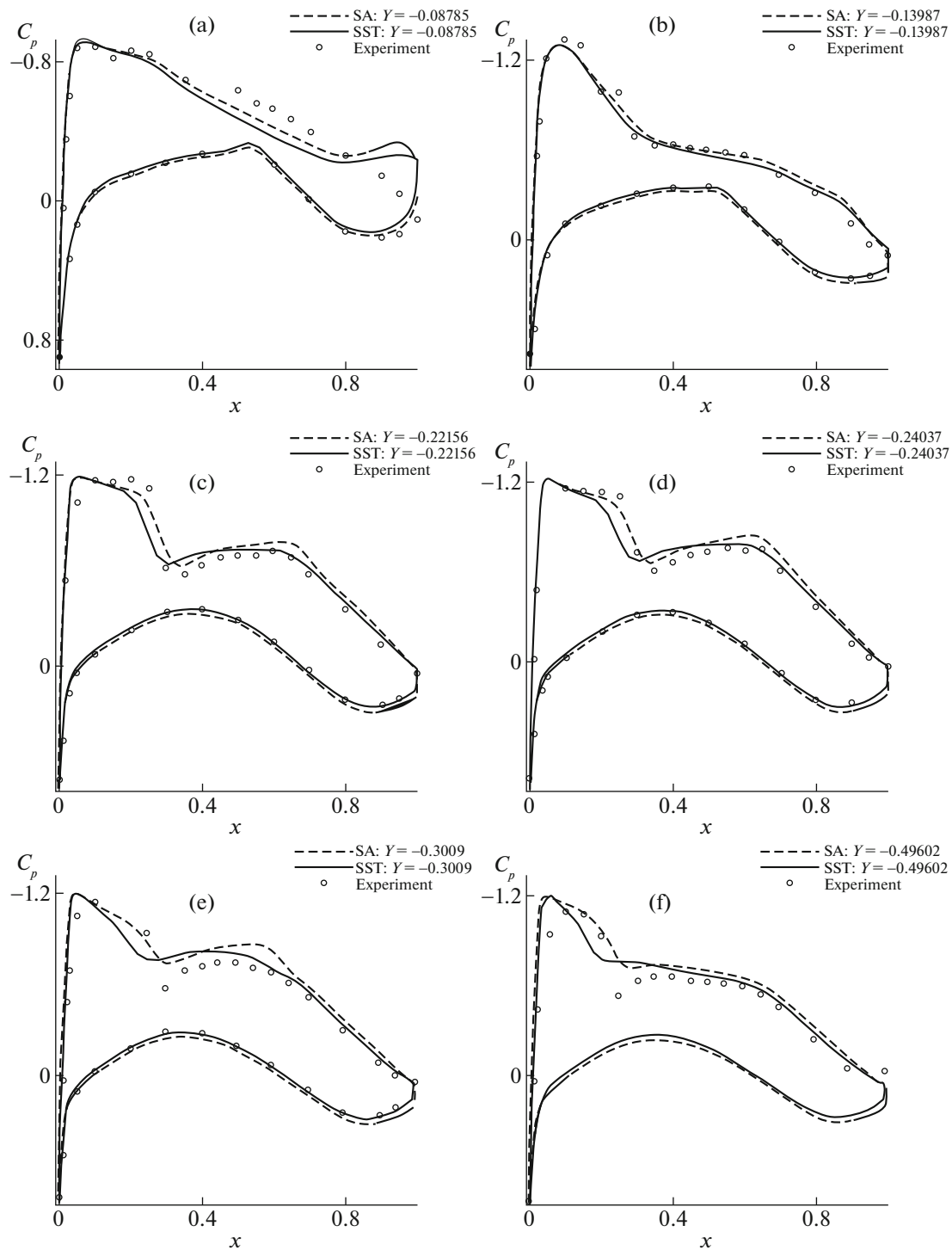


Fig. 8. Distribution of the pressure coefficient over the chord of the DLR F6 profile. The solid line corresponds to the computation within the model $k-\omega$ -SST, the dashed line to the Spalart–Allmaras model, and dots correspond to the experimental data.

FlowModellium are compared with the results obtained by CFL3D. The computations were performed on a hexahedral mesh refining in the vicinity of the airfoil (the minimal step is $y = 5 \times 10^{-7}$). Figure 6 plots the drag coefficient obtained by FlowModellium and by CFL3D (the data is taken from the NASA site).

4.2. Problems with Complex Geometry

In this section, we consider examples of the computation of airflow around a 3D body of complex shape for various flight modes.

Table 2. List of experimental points from [58]

Point	$\rho_\infty, \text{kg/m}^3$	Angle of attack α	$U_\infty, \text{m/s}$	p_∞, Pa	M_∞	Re per meter of length
Run3064	3.721E-02	28.18	1467	5.894E+02	9.85	0.152015E+08
Run3079	3.955E-02	-0.08	1431	5.837E+02	9.96	0.119638E+08
Run3072	0.379	28	1370	8351	7.8	0.102851E+09

The first example is the computation of airflow around the model DLR-F6, which is the assembly of the fuselage and wing of a passenger aircraft. The computation results are compared with the experimental data presented at the 2nd AIAA CFD Drag Prediction Workshop [56]. The geometric parameters of DLR-F6 are as follows: the mean aerodynamic chord is $CAX = 0.1412 \text{ m}$, the wing semispan is 0.5877 m , and the fuselage length is 1.192 m . The flow for $Re_{CAX} = 3 \times 10^6$, $M = 0.75$ at the angle of attack $\alpha = 0.49^\circ$ was considered. The following undisturbed flow deceleration parameters were used in the computations: $p_0 = 90470.781 \text{ Pa}$ and $T_0 = 36.6425 \text{ K}$. The structured hexahedral mesh consisting of 4 million cells (Fig. 7) was used for the computations.

Figure 8 presents the diagrams of the pressure coefficient c_p compared with the experiment. Data for six wing sections $y = 0.08785, 0.13987, 0.22156, 0.24037, 0.30090,$ and 0.49602 are shown. Note that all test results are in good agreement with the results obtained using the package EWT-TsAGI [57].

As the second example, we consider the flow around the shuttle Orion that is under development in the USA. In [58], experimental data for the model on the scale 1:28 for the Mach numbers of the incident flow 8 and 10 are presented. This data can be used for testing computer codes. The model is a segmental-conical body (inverse cone with a spherical bluntness). A hexahedral mesh consisting of 3 million cells constructed for a half of the geometry was used for the computations. The list of flow models for which the computations were performed is shown in Table 2. The body surface temperature was $T_w = 300 \text{ K}$ in all cases. The first two flow modes are laminar, and the third mode is turbulent. Note that the flow parameters at infinity were time-dependent, and a mean value was chosen for the computations.

Figure 9 compares the distribution of the normalized heat flux

$$q' = St/\sqrt{Re_D} = \frac{q_n}{\rho_\infty U_\infty \Delta H} \left(\frac{\rho_\infty U_\infty D}{\mu_\infty} \right)^{1/2}, \quad \Delta H = c_p T_\infty + \frac{1}{2} U_\infty^2 - c_p T_w, \quad (21)$$

where St is the Stanton number and Re_D is the Reynolds number calculated using the model diameter measured across the front cover for the chosen laminar modes. We see that the results are in good agree-

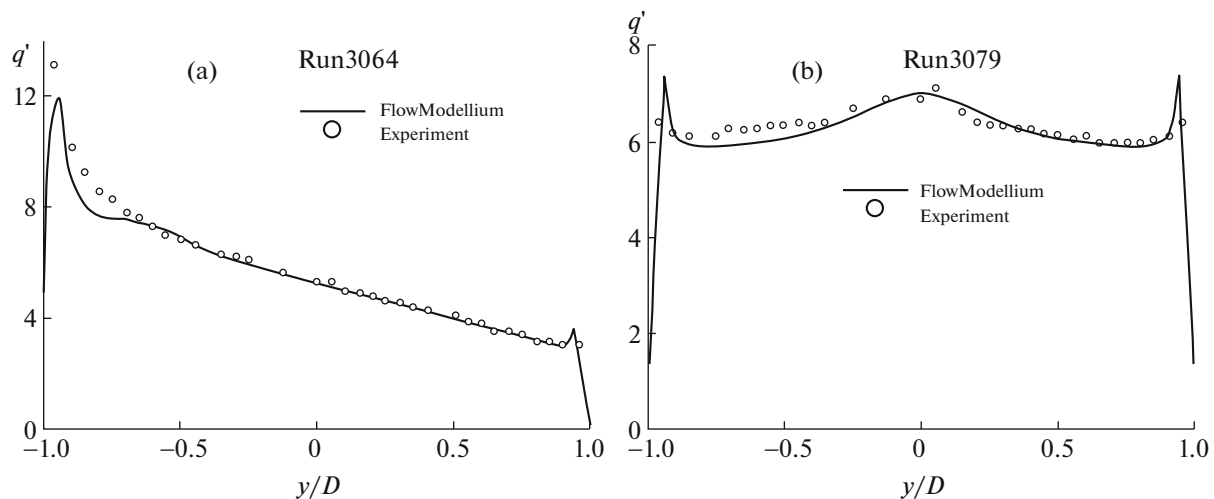


Fig. 9. Distribution of the normalized heat flux q' (see Eq. (21)) over the front cover of Orion; comparison with the experimental data provided by NASA [58] for laminar modes.

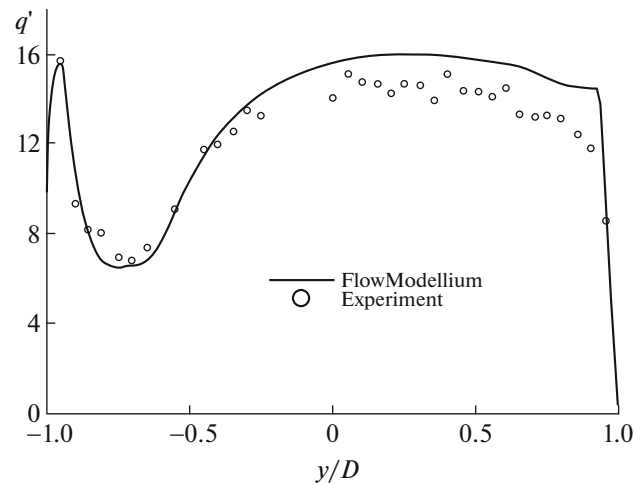


Fig. 10. Distribution of the normalized heat flux q' (see Eq. (21)) over the front cover of Orion; comparison with the experimental data provided by NASA [58] for the turbulent flow mode Run3072.

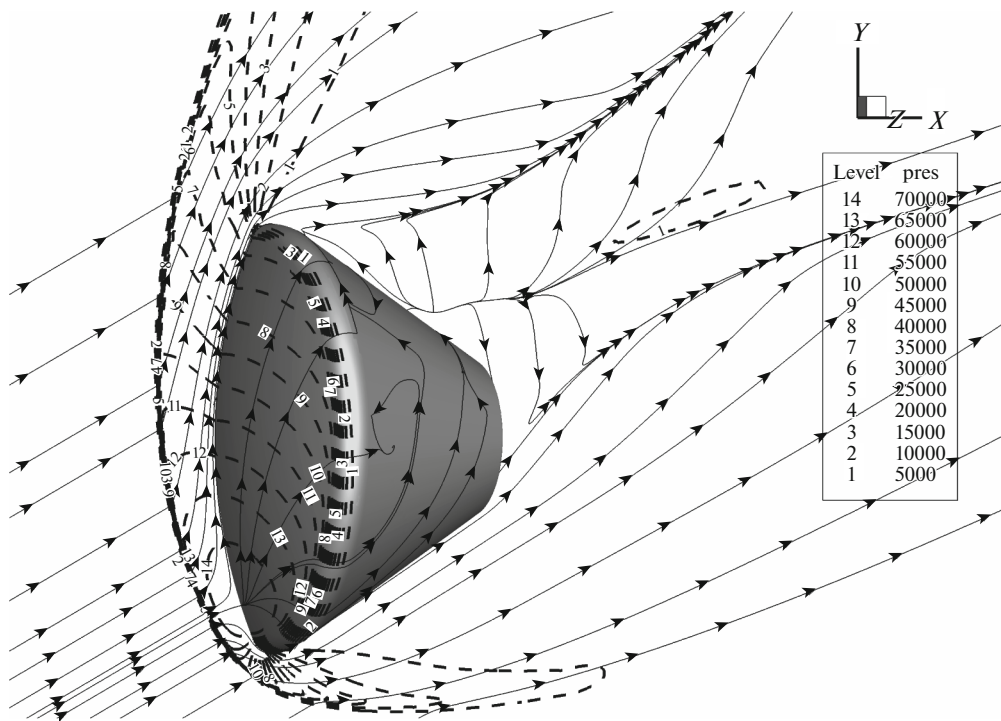


Fig. 11. Pressure and streamline level lines for the mode Run3064.

ment: the accuracy of our results is comparable with the accuracy of the code Laura developed in NASA. Note that in [58] the mesh adapted to the front of the shock wave was used, while we used the same basic mesh for all angles of attack. Figure 10 illustrates the results of computations of flow around a model of Orion based on the RANS Spalart–Allmaras model for the turbulent mode Run3072. In the case of laminar flow at nonzero angles of attack, the heat flux goes from the lower edge of the front shield of the model to its upper edge; in the case of turbulent flow, the turbulent flow dependence curve is nonmonotonic: rapid decrease in the lower part of the shield gives place to increase and then the curve becomes flat. The deviation from the experimental data is 10–15%, which is approximately the same error as is given by the Laura code. Note a fairly strong sensitivity of the results to the choice of concrete values of the flow parameters at infinity from the table of experimental data. The flow pattern for the mode Run3064 is illustrated in Fig. 11. We clearly see the bow shock wave and an intricate pattern of streamlines behind the body.

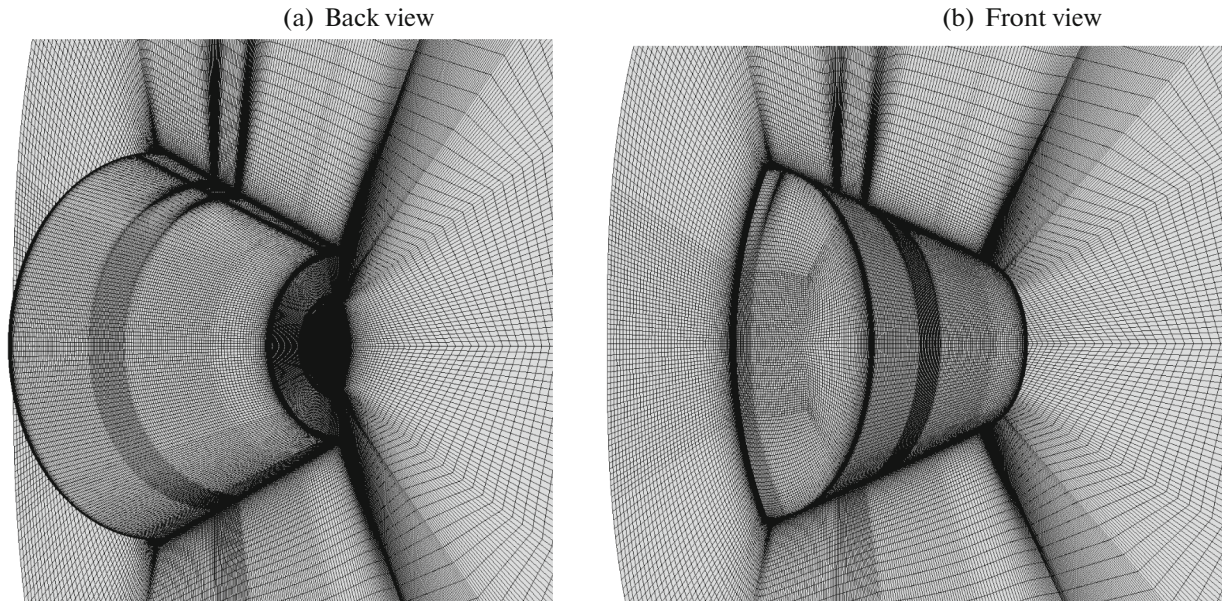


Fig. 12. Details of the computational mesh for computing the flow around the Federation spacecraft.

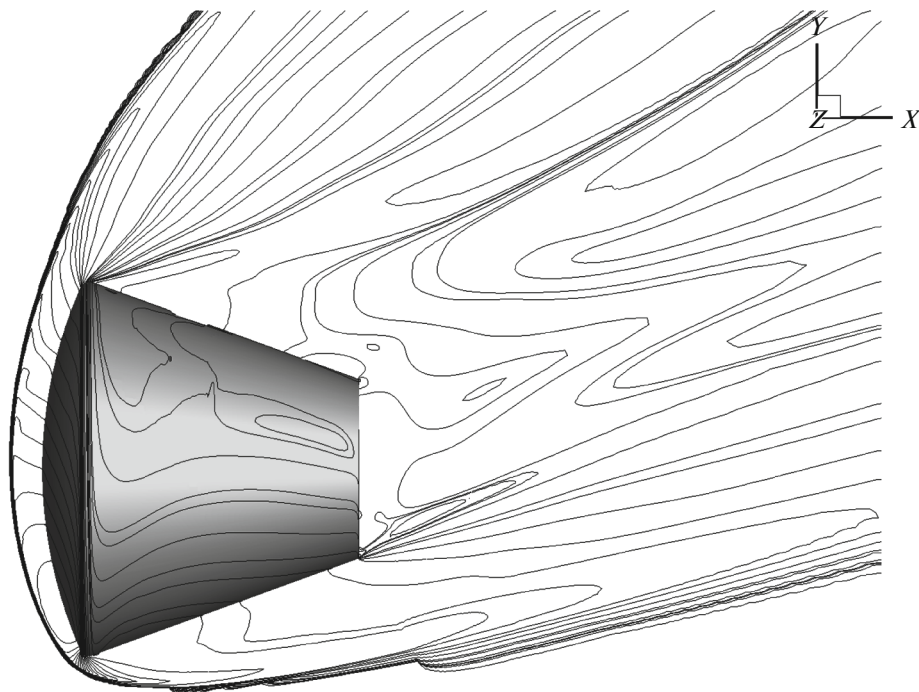


Fig. 13. Flow pattern (pressure level lines) obtained in the computation of airflow around the Federation spacecraft for the altitude of 60 km, $M_\infty = 35$, and the angle of attack $\alpha = 25$, model 1TM.

Finally, we discuss the computation of flow around the prospective spacecraft called *Federation* developed to replace the Soyuz spacecraft [59]. The geometry of this spacecraft is a complex segmental-conical body with a rectangular cut on the upper part of the surface. In the computations, we used a hexahedral mesh consisting of six million cells. The mesh details are shown in Fig. 12. The program package was tested for the flow computation at the altitudes between 50 and 80 km at Mach numbers of the incident flow up to 40.

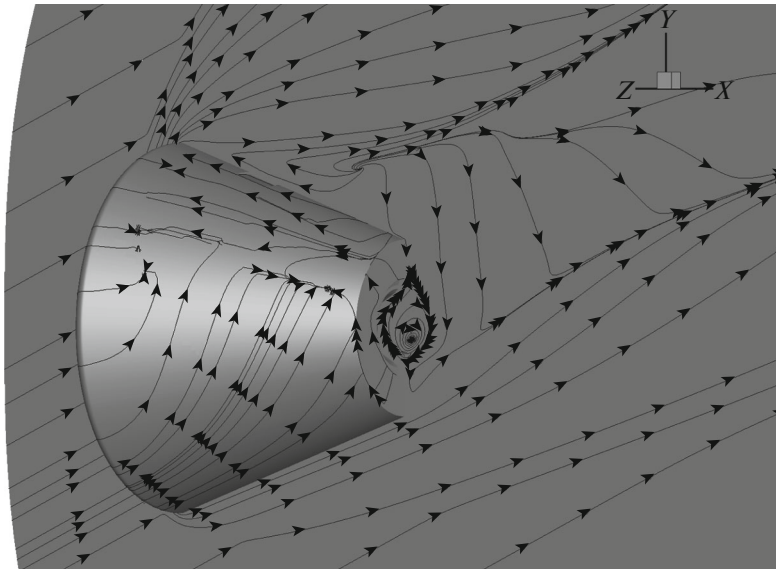


Fig. 14. Flow pattern (streamlines) obtained in the computation of airflow around the Federation spacecraft for the altitude of 60 km, $M_\infty = 35$, and the angle of attack $\alpha = 25$, model 1TM.

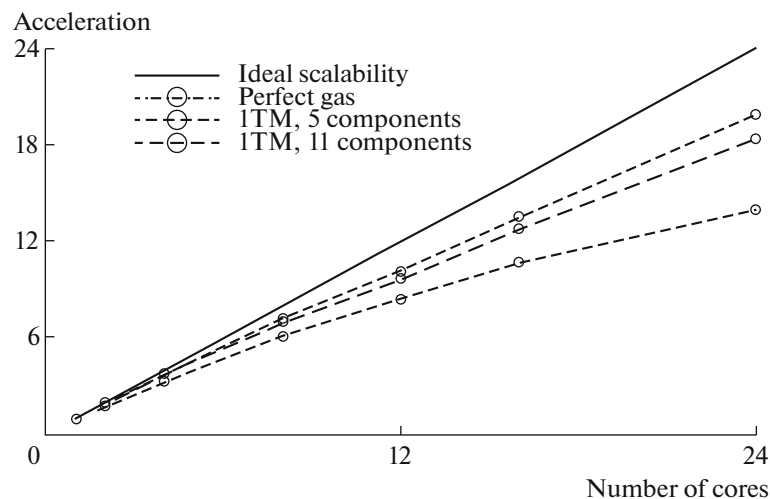


Fig. 15. Code scalability obtained using OpenMP on a single cluster node (two 12-core Intel Xeon E5-2697v2 processors).

Figures 13 and 14 show the field of pressure and streamlines for the altitude 60 km, $M_\infty = 35$, and the angle of attack 25 degrees. The computation was based on the model 1TM (five-component nonequilibrium chemistry). In the results, the bow shock wave and the reverse flow zone behind the body are well resolved. A kink on the level lines in the lower part of the wave front is explained by the nonsmoothness of the computation mesh in this part of the flow. However, this does not affect the main characteristics of the flow, such as the distribution of heat flux over the body surface.

4.3. Scalability Tests

Figure 15 illustrates the scalability of code within one node of the laboratory cluster consisting of two 12-core Intel Xeon E5-2697v2 processors when using the OpenMP technology for various physical models of viscous gas. The computations were performed on a mesh consisting of 600 thousand hexahedra using the second-order implicit TVD scheme. For the perfect gas model, the acceleration is by a factor of 14, and for the model 1TM with eleven components the acceleration is by a factor of almost 20. The

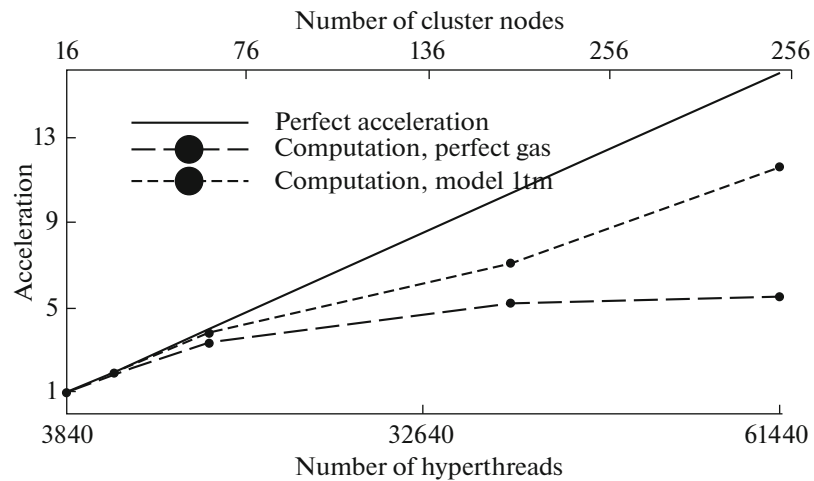


Fig. 16. Code scalability on the cluster Politeknik RSK PetaStream in the Peter the Great St. Petersburg Polytechnic University with the use of up to 256 Intel Xeon Phi 5120D coprocessors.

better scalability of code for the model 1TM is due to a large number of computations needed to find the numerical fluxes, which compensates for the time needed to read data from the node memory.

Figure 16 illustrates the results of scalability testing on the computer system *Politechnik–RSK PetaStream* [53] in the Peter the Great St. Petersburg Polytechnic University. Each cluster node consists of a single coprocessor Intel Xeon Phi 5120D (61 physical cores, 244 hyperthreads). The airflow around the descent module of the Federation spacecraft was computed on a spatial mesh consisting of 6 millions of cells. For the perfect gas model, acceleration by a factor of 5.22 was achieved when using 128 nodes instead of 16 (the efficiency is 65%). As the number of nodes increases further, the acceleration curve becomes flat. For the model 1TM (single-temperature nonequilibrium chemistry), the acceleration by a factor 11.6 when using 256 nodes instead of 16 (the efficiency is 74%).

These results demonstrate excellent scalability of the parallel code described above.

CONCLUSIONS

The software package FlowModellium developed in the laboratory of mathematical modeling of non-linear processes in gaseous media of the Institute of Physics and Technology (Russia) is described. The results demonstrate good parallel scalability of the software, its robustness, and applicability to computing airflow around bodies of complex shape on modern supercomputers.

ACKNOWLEDGMENTS

This work was supported by the Russian Science Foundation, project no. 18-19-00098.

The research was carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University [51, 52], resources of the Joint Supercomputer Center (JSCC) of the Russian Academy of Sciences, and of the Peter the Great St. Petersburg Polytechnic University.

We are grateful to D.A. Zabarko, D.A. Sivkov, M.P. Shuvalov, A.N. Krylov, A.V. Beloshitskii, and I.V. Voronich for useful discussions. The work was accomplished using the resources of the Joint Supercomputer Center (JSCC) of the Russian Academy of Sciences, Peter the Great St. Petersburg Polytechnic University, and Lomonosov Moscow State University [51, 52].

REFERENCES

1. A. S. Kozelkov, Yu. N. Deryugin, D. K. Zelenskii, V. A. Glazunov, A. A. Golubev, O. V. Denisova, S. V. Lashkin, R. N. Zhuchkov, N. V. Tarasova, and M. A. Sizova, "Multipurpose software package LOGOS for solving fluid dynamics and heat and mass transfer problems on supercomputers: Basic technologies and algorithms," *Proc. of the XII Int. Seminar on Super Computations and Mathematical Modeling*, Sarov, Russia, 2010, pp. 215–230.

2. I. V. Abalakin, P. A. Bakhvalov, A. V. Gorobets, A. P. Duben', and T. K. Kozubskaya, "Parallel software package NOISETTE for large-scale computations in fluid dynamics and aeroacoustics," *Vychisl. Metody Proram.* **13** (3), 110–125 (2012).
3. G. A. Faranosov, V. M. Goloviznin, S. A. Karabasov, V. G. Kondakov, V. F. Kopiev, and M. A. Zaitsev, "Cabaret method on unstructured hexahedral grids for jet noise computation," *Comput. Fluids.* **88**, 165–179 (2013).
4. A. V. Gorobets, "Parallel Technology for Numerical Modeling of Fluid Dynamics Problems by High-Accuracy Algorithms," *Comput. Math. Math. Phys.* **55**, 638–649 (2015).
5. A. B. Gorshkov, "Parallelization algorithm for implicit method computation of hypersonic nonequilibrium gas flow past a body, based on Navier–Stokes equations," *Math. Models Comput. Simul.* **2**, 252–260 (2010).
6. A. L. Zheleznyakova and S. T. Surzhikov, "Computation of hypersonic flow around bodies with complex shape on unstructured tetrahedral meshes using the AUSM scheme," *Teplomassoobmen Fiz. Gidrodinam.* **52** (2), 283–293 (2012).
7. A. V. Novikov, A. V. Fedorov, and I. V. Egorov, "Numerical studies of 3D instabilities propagating in supersonic compression-corner flow," *8th European Symp. on Aerothermodynamics for Space Vehicles*, Lisbon, 2015.
8. V. A. Garanzha, L. N. Kudryavtseva, and S. V. Utyuzhnikov, "Variational method for untangling and optimization of spatial meshes," *J. Comp. Appl. Math.* **269**, 24–41 (2014).
9. V. A. Titarev and S. V. Utyuzhnikov, "Software package for calculating hypersonic air flows," Certificate of State Registration of computer programs 2013619670, 2013.
10. S. A. Vasil'evskii, L. G. Efimova, A. F. Kolesnikov, I. A. Sokolova, and G. A. Tirkii, "Highly accurate computation of transport coefficients in a multicomponent plasma. The effect of element separation in chemically and ionization equilibrium plasma," Rep. 2427 of the Institute of Mechanics, Russian Academy of Sciences, 1980.
11. V. I. Sakharov, "Numerical simulation of thermally and chemically nonequilibrium flows and heat transfer in underexpanded induction plasmatron Jets," *Fluid. Dyn.* **42**, 1007–10016 (2007).
12. V. I. Sakharov, "Simulation of nonequilibrium flows of viscous gas in induction plasmatrons and around bodies," *Doctoral (Phys. Math.) Dissertation*, Moscow State Univ., Moscow, 2011.
13. L. V. Gurvich, I. V. Veits, V. A. Medvedev, et al., *Thermodynamic Properties of Individual Materials* (Nauka, Moscow, 1979) [in Russian].
14. R. C. Reid, J. M. Prausnitz, and T. K. Sherwood, *The Properties of Gases and Liquids* (McGraw-Hill, New York, 1977).
15. P. R. Spalart and S. R. Allmaras, "A one-equation turbulence model for aerodynamic flows," AIAA Paper 92-0439, 1992.
16. J. E. Bardina, P. G. Huang, and T. J. Coakley, "Turbulence modeling validation, testing, and development," AIAA Paper 92-0439, 1997.
17. P. G. Huang, J. E. Bardina, and T. J. Coakley, "Turbulence modeling validation, testing, and development," NASA Techn. Rep., 1997.
18. F. Menter, "Zonal two equation kw turbulence models for aerodynamic flows," *23rd Fluid Dynamics, Plasma-dynamics, and Lasers Conf.*, 1993, p. 2906.
19. M. L. Shur, M. K. Strelets, A. K. Travin, and P. R. Spalart, "Turbulence modeling in rotating and curved channels: Assessing the Spalart–Shur Correction," *AIAA J.* **38**, 784–792 (2000).
20. J. R. Edwards and S. Chandra, "Comparison of eddy viscosity–transport turbulence models for three-dimensional, shock-separated flow fields," *AIAA J.* **34**, 756–763 (1996).
21. T. Rung, U. Bunge, M. Schatz, and F. Thiele, "Restatement of the Spalart–Allmaras eddy-viscosity model in strain-adaptive formulation," *AIAA J.* **41**, 1396–1399 (2003).
22. P. R. Spalart, "Trends in turbulence treatments," AIAA 2000-2306, 2000.
23. M. Mani, D. A. Babcock, C. M. Winkler, and P. R. Spalart, "Predictions of a supersonic turbulent flow in a square duct," AIAA Paper 2013-0860, 2013.
24. M. Dumbser, M. Käser, V. A. Titarev, and E. F. Toro, "Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems," *J. Comput. Phys.* **226**, 204–243 (2007).
25. P. Tsoutsanis, V. A. Titarev, and D. Drikakis, "WENO schemes on arbitrary mixed-element unstructured meshes in three space dimensions," *J. Comput. Phys.* **230**, 1585–1601 (2010).
26. V. A. Titarev, "Efficient deterministic modelling of three-dimensional rarefied gas flows," *Commun. Comput. Phys.* **12** (1), 161–192 (2012).
27. V. A. Titarev, M. Dumbser, and S. V. Utyuzhnikov, "Construction and comparison of parallel implicit kinetic solvers in three spatial dimensions," *J. Comput. Phys.* **256**, 17–33 (2014).

28. A. G. Kulikovskii, N. V. Pogorelov, and A. Yu. Semenov, *Mathematical Issues of the Numerical Solution of Hyperbolic Systems of Equations* (Fizmatlit, Moscow, 2001) [in Russian].
29. E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd ed. (Springer, 2009).
30. A. Harten, P. D. Lax, and B. van Leer, "On upstream differencing and Godunov-type schemes for hyperbolic conservation laws," *SIAM Rev.* **25** (1), 35–61 (1983).
31. E. F. Toro, M. Spruce, and W. Speares, "Restoration of the contact surface in the Harten-Lax-van Leer Riemann solver," *J. Shock Waves* **4**, 25–34 (1994).
32. M. Dumbser and E. F. Toro, "On universal Osher-type schemes for general nonlinear hyperbolic conservation laws," *Commun. Comput. Phys.* **10**, 635–671 (2011).
33. P. Batten, M. A. Leschziner, and U. C. Goldberg, "Average-state Jacobians and implicit methods for compressible viscous and turbulent Flows," *J. Comput. Phys.* **137**, 38–78 (1997).
34. V. V. Vlasenko, E. V. Kazhan, E. S. Matyash, S. V. Mikhailov, and A. I. Troshin, "Numerical implementation of an implicit scheme and various turbulence models in the computational module ZEUS," *Trudy Tsentr. Aerogidrodinamich. Inst.* **2735**, 5–49 (2015).
35. M. Dumbser, J.-M. Moschetta, and J. Gressier, "A matrix stability analysis of the carbuncle phenomenon," *J. Comput. Phys.* **192**, 647–670 (2004).
36. I. Yu. Tagirova and A. V. Rodionov, "Application of artificial viscosity for suppressing the carbuncle phenomenon in Godunov-type schemes," *Math. Models Comput. Simul.* **8**, 249–262 (2015).
37. A. V. Rodionov, "Artificial viscosity in Godunov-type schemes to cure the carbuncle phenomenon," *J. Comp. Phys.* **345**, 308–329 (2017).
38. C. R. Mitchell, "Improved reconstruction schemes for the Navier-Stokes equations on unstructured meshes," AIAA-94-0642, 1994.
39. Neal T. Frink, "Assessment of an unstructured-grid method for predicting 3-D turbulent viscous flows," AIAA-96-0292, 1996.
40. S. Yoon and A. Jameson, "Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier–Stokes equations," *AIAA J.* **26**, 1025–1026 (1998).
41. I. S. Men'shov and Y. Nakamura, "An implicit advection upwind splitting scheme for hypersonic air flows in thermochemical nonequilibrium," *Collection of Technical Papers of the 6th Int. Symp. on CFD*, Lake Tahoe, Nevada, 1995, Vol. 2, p. 815.
42. I. S. Men'shov and Y. Nakamura, "On implicit Godunov's method with exactly linearized numerical flux," *Comput. Fluids* **29**, 595–616 (2000).
43. D. Sharov, H. Luo, J. D. Baum, and R. Löhner, "Implementation of unstructured grid GMRES+LU-SGS method on shared-memory, cache-based parallel computers," 38th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2000, AIAA-2000-927, pp. 10–13.
44. M. J. Chorley and D. W. Walker, "Performance analysis of a hybrid MPI/OpenMP application on multi-core clusters," *J. Comput. Sci.* **47**, 168–174 (2010).
45. A. V. Gorobets, A. O. Zheleznyakov, S. A. Sukov, P. B. Bogdanov, and B. N. Chetverushkin, "Extension of the two-level MPI + OpenMP parallelization using OpenCL for fluid dynamics computations on heterogeneous systems," *Vest. Yuzhno-Uralsk. Gos. Univ., Ser. Mat. Model. Progr.*, No. 9, 76–86 (2011).
46. A. V. Gorobets, "Parallel algorithm of the NOISEtte code for CFD and CAA simulations," *Lobachevskii J. Math.* **39**, 524 – 532 (2018).
47. G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme for irregular graphs," *J. Parallel Distrib. Comput.* **48**, 96–129 (1998).
48. I. E. Kaporin and O. Yu. Milyukova, "The massively parallel preconditioned conjugate gradient method for the numerical solution of linear algebraic equations," in *Collection of Papers of the Department of Applied Optimization of the Dorodnicyn Computing Center, Russian Academy of Sciences*, Ed. by V. G. Zhadan (Vychisl. Tsentr, Ross. Akad. Nauk, Moscow, 2011), pp. 132–157.
49. A. M. Wissink, A. S. Lyrintzis, and R. C. Strawn, "Parallelization of a three-dimensional flow solver for Euler rotorcraft aerodynamics predictions," *AIAA J.* **34**, 2276–2283 (1996).
50. M. N. Petrov, V. A. Titarev, S. V. Utyuzhnikov, and A. V. Chikitkin, "A multithreaded OpenMP implementation of the LU-SGS method using the multilevel decomposition of the unstructured computational mesh," *Comput. Math. Math. Phys.* **57**, 1856–1865 (2017).
51. Vl. V. Voevodin, S. A. Zhumatii, S. I. Sobolev, A. S. Antonov, P. A. Bryzgalov, D. A. Nikitenko, K. S. Stefanov, and Vad. V. Voevodin, "Practical usage of the Lomonosov supercomputer," *Otkrytie Sist.*, No. 7, 36–39 (2012).

52. V. Sadovnichy, A. Tikhonravov, Vl. Voevodin, and V. Opanasenko, “Lomonosov: Supercomputing at Moscow State University,” in *Contemporary High Performance Computing: From Petascale toward Exascale (Chapman & Hall/CRC Computational Science)* (CRC, Boca Raton, USA, 2013), pp. 283–307.
53. A. Semin, E. Druzhinin, V. Mironov, A. Shmelev, and A. Moskovsky, “The performance characterization of the RSC petastream module,” *Proc. of the 29th Int. Conf., ISC, Leipzig, 2014*; *Lect. Notes Comput. Sci.* **8488**, 420–429 (2014).
54. A. N. Lyubimov and V. V. Rusanov, *Gas Flows around Blunt Bodies. Part I: Method of Computation and Analysis of Flows. Part II: Tables of Gasdynamic Functions* (Nauka, Moscow, 1970) [in Russian].
55. S. Swaminathan, M. D. Kim, and C. H. Lewis, “Nonequilibrium viscous shock-layer flows over blunt sphericones at angle of attack,” *J. Spacecraft Rockets* **20**, 331–338 (1983).
56. Kelly R. Laflin, Steven M. Klausmeyer, Tom Zickuhr, John C. Vassberg, Richard A. Wahls, Joseph H. Morrison, Olaf P. Brodersen, Mark E. Rakowitz, Edward N. Tinoco, and Jean-Luc Godard, *Data Summary from Second AIAA Computational Fluid Dynamics Drag Prediction Workshop*; *J. Aircraft* **42**, 1165–1178 (2005).
57. S. M. Bosnyakov, V. V. Vlasenko, M. F. Engulatova, E. V. Kazhan, S. V. Matyash, and A. I. Troshin, “Industrial solvers in the software package EWT-TsAGI and their verification on a series of standard benchmarks,” *Trudy Tsentr. Aerogidrodinamich. Inst.* **2735**, 3–91 (2014).
58. B. R. Hollis, T. J. Horvath, K. T. Berger, R. P. Lillard, B. S. Kirk, J. J. Coblisch, and J. D. Norris, “Experimental investigation of project Orion crew exploration vehicle aeroheating in aedc tunnel 9,” *NASA/TP-2008-215547*, 2008, p. 158.
59. N. P. Adamov, A. M. Kharitonov, E. A. Chasovnikov, A. A. Dyad’kin, M. I. Kazakov, A. N. Krylov, and A. Yu. Skorovarov, “Aerodynamic characteristics of reentry vehicles at supersonic velocities,” *Thermophys. Aeromech.* **22**, 557–565 (2015).

Translated by A. Klimontovich