ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2016, Vol. 56, No. 6, pp. 1048–1064. © Pleiades Publishing, Ltd., 2016.
Original Russian Text © I.V. Egorov, A.V. Novikov, 2016, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2016, Vol. 56, No. 6, pp. 1064–1081.

In memory of O.M. Belotserkovskii

Direct Numerical Simulation of Laminar–Turbulent Flow over a Flat Plate at Hypersonic Flow Speeds

I. V. Egorov[a, b] and A. V. Novikov[a, c]

[a] *Central Aerodynamic Institute, ul. Zhukovskogo 1, Zhukovsky, Moscow oblast, 140180 Russia*
[b] *Institute for Analytical Instrumentation, Russian Academy of Sciences, 2-ya Brestskaya ul. 19/18, Moscow, 123056 Russia*
[c] *Moscow Institute of Physics and Technology (State University), Institutskii per. 9, Dolgoprudny, Moscow oblast, 147700 Russia*
e-mail: AndrewNovikov@yandex.ru

Received November 9, 2015

Abstract—A method for direct numerical simulation of a laminar–turbulent flow around bodies at hypersonic flow speeds is proposed. The simulation is performed by solving the full three-dimensional unsteady Navier–Stokes equations. The method of calculation is oriented to application of supercomputers and is based on implicit monotonic approximation schemes and a modified Newton–Raphson method for solving nonlinear difference equations. By this method, the development of three-dimensional perturbations in the boundary layer over a flat plate and in a near-wall flow in a compression corner is studied at the Mach numbers of the free-stream of M = 5.37. In addition to pulsation characteristic, distributions of the mean coefficients of the viscous flow in the transient section of the streamlined surface are obtained, which enables one to determine the beginning of the laminar–turbulent transition and estimate the characteristics of the turbulent flow in the boundary layer.

Keywords: direct numerical simulation, laminar–turbulent transition, hypersonic flows, boundary layer.

DOI: 10.1134/S0965542516060129

INTRODUCTION

The problem of calculating the laminar–turbulent transition (LTT) in a hypersonic flow around bodies is one of the main problems of high-speed aerodynamics. The LTT leads to a significant increase in the heat fluxes and friction drag of a flying vehicle and influences the efficiency of the propulsion unit and control devices. Solving this problem requires a deep insight into the physical mechanisms initiating the transition to turbulence. In the case of small external perturbations, which is typical of the flight conditions, the LTT has three main stages (see [1]): the receptivity to external perturbations; the development of unstable modes, such as the first and second Mack modes, cross-flow instability, and Görtler vortices; and the nonlinear breakdown of perturbations, terminating with developed turbulent flow. In Belotserkovskii's works, much attention was paid to the initiation and development of turbulence for a wide class of problems of fluid mechanics (see [2]).

Belotserkovskii called attention to the fact that the integrated calculation of all stages of the LTT is possible only by the direct numerical simulation (DNS) in which the full unsteady Navier–Stokes equations are solved without restrictions on the main (unperturbed laminar) flow and the amplitude of perturbations. In addition, in contrast to physical experiments, the DNS gives the full information on the perturbation field, which enables one to distinguish and analyze various mechanisms of the LTT. The modern methods of parallel computations and the rapid development of multiprocessor systems made possible numerical experiments for hypersonic boundary layers on simple configurations such as a plate and cone at a zero angle of attack (see [3]).

In this work, we present a mathematical formulation of the problem of simulation of viscous compressible gas flows, a method for the numerical integration of the Navier–Stokes equations, and examples of

application of the approach developed. The numerical method is based on an implicit finite-volume method of the second-order approximation in both space and time, a Godunov-type quasi-monotonic TVD scheme, and the implementation on a multiprocessor cluster supercomputer. The suggested approach is especially efficient if the system of equations in the problem has an increased stiffness, e.g., in the cases when the computational region contains spacial flow inhomogeneities (such as shock waves and separated-flow regions) [4] and when modeling gas flows with allowance for nonequilibrium physicochemical processes (see [5]). In [4, 5], this approach was used in the implementation on one-processor computers.

To verify the method proposed, the comparison with the results of direct numerical simulation of wave packets developing on a plate at the free-stream Mach number $M_\infty = 5.35$ [6] was performed. For example, the development of three-dimensional perturbations in the boundary layer on a flat plate at $M_\infty = 5.373$ was studied. Similar calculations for two-dimensional perturbations are presented in [7–9].

With the help of visualization of the fields of three-dimensional perturbation, specificities of the development of instability in the linear and nonlinear stages are revealed and discussed. In addition to the pulsation characteristics, distributions of the mean viscous friction coefficients in the laminar and transient sections of the streamlined surface are considered, which makes it possible to determine the beginning of the LTT and estimate the length of the transition region.

1. PROBLEM FORMULATION

In mechanics of continuous media, the gas motion is generally described by transient three-dimensional Navier–Stokes equations, which also serve as a basis for the direct numerical simulation of a turbulent flow.

The Navier–Stokes equations in an arbitrary curvilinear coordinate system ξ,η,ζ, where $x = x(\xi,\eta,\zeta)$, $y = y(\xi,\eta,\zeta)$, and $z = z(\xi,\eta,\zeta)$ are the Cartesian coordinates, are written in the divergent form as follows:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial \xi} + \frac{\partial \mathbf{G}}{\partial \eta} + \frac{\partial \mathbf{F}}{\partial \zeta} = \mathbf{0}; \tag{1}$$

here, $\mathbf{Q}$ is the vector of conservative dependent variables of the problem and $\mathbf{E}$, $\mathbf{G}$, and $\mathbf{F}$ are the vectors of fluxes in the curvilinear coordinates. The vectors $\mathbf{E}$, $\mathbf{G}$, and $\mathbf{F}$ are related to the corresponding vectors $\mathbf{E}_c$, $\mathbf{G}_c$, and $\mathbf{F}_c$ in the Cartesian coordinates by the formulas

$$\begin{gathered}\mathbf{Q} = J\mathbf{Q}_c, \quad \mathbf{E} = J\left(\mathbf{E}_c\frac{\partial \xi}{\partial x} + \mathbf{G}_c\frac{\partial \xi}{\partial y} + \mathbf{F}_c\frac{\partial \xi}{\partial z}\right),\\ \mathbf{G} = J\left(\mathbf{E}_c\frac{\partial \eta}{\partial x} + \mathbf{G}_c\frac{\partial \eta}{\partial y} + \mathbf{F}_c\frac{\partial \eta}{\partial z}\right), \quad \mathbf{F} = J\left(\mathbf{E}_c\frac{\partial \zeta}{\partial x} + \mathbf{G}_c\frac{\partial \zeta}{\partial y} + \mathbf{F}_c\frac{\partial \zeta}{\partial z}\right),\end{gathered} \tag{2}$$

where $J = \partial(x,y,z)/\partial(\xi,\eta,\zeta)$ is the Jacobian of the transformation.

The curvilinear coordinate system (ξ,η,ζ) is used for the discretization on a uniform grid. To this end, a given arbitrary computational grid in the Cartesian coordinates is mapped onto a uniform grid in the curvilinear coordinates. The Cartesian components of the vectors $\mathbf{E}_c$, $\mathbf{G}_c$, and $\mathbf{F}_c$ for the three-dimensional Navier–Stokes equations have the form

$$\mathbf{Q}_c = \begin{Vmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho e\end{Vmatrix}, \quad \mathbf{E}_c = \begin{Vmatrix}\rho u\\ \rho u^2 + p + \tau_{xx}\\ \rho uv + \tau_{xy}\\ \rho uw + \tau_{xz}\\ \rho uH + I_x\end{Vmatrix}, \quad \mathbf{G}_c = \begin{Vmatrix}\rho v\\ \rho uv + \tau_{xy}\\ \rho v^2 + p + \tau_{yy}\\ \rho vw + \tau_{yz}\\ \rho vH + I_y\end{Vmatrix}, \quad \mathbf{F}_c = \begin{Vmatrix}\rho w\\ \rho wu + \tau_{xz}\\ \rho wv + \tau_{yz}\\ \rho w^2 + p + \tau_{zz}\\ \rho wH + I_z\end{Vmatrix},$$

where ρ is the gas density; u, v, and w are the Cartesian components of the velocity vector $\mathbf{V}$; p is pressure; $e = h - \frac{p}{\rho} + \frac{1}{2}(u^2 + v^2 + w^2)$ is the total energy per unit volume; $H = h + \frac{1}{2}(u^2 + v^2 + w^2)$ is the total enthalpy, $h = c_pT$ is the static enthalpy; T is temperature; c_p is the specific heat capacity at a constant

pressure, $\boldsymbol{\tau}$ is the symmetric viscous stress tensor related to the strain rate tensor $\mathbf{s}$ by a linear dependence. The components of the strain rate tensor $\mathbf{s}$ for compressible gas have the form

$$\mathbf{s}_{xx} = 2\frac{\partial u}{\partial x} - \frac{2}{3}\operatorname{div}\mathbf{V}, \quad \mathbf{s}_{yy} = 2\frac{\partial v}{\partial y} - \frac{2}{3}\operatorname{div}\mathbf{V}, \quad \mathbf{s}_{zz} = 2\frac{\partial w}{\partial z} - \frac{2}{3}\operatorname{div}\mathbf{V},$$

$$\mathbf{s}_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}, \quad \mathbf{s}_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}, \quad \mathbf{s}_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y},$$

and the heat flux vector $\mathbf{I}$ is defined by the expression

$$\mathbf{I} = -\lambda \operatorname{grad}(T) + \boldsymbol{\tau}\mathbf{V},$$

where μ and λ are the coefficient of molecular viscosity and the heat conductivity, respectively.

System of equations (1) is closed by the equation of state and dependences of the transfer coefficients on temperature and pressure. In this work, we use the model of ideal gas with the equation of state

$$p = \rho RT/M,$$

where R is the universal gas constant and M is the molecular weight of gas. The coefficient of molecular viscosity is assumed to be temperature-dependent and is calculated according to the Sutherland law:

$$\frac{\mu}{\mu_\infty} = \frac{1 + \dfrac{T_\mu}{T_\infty}}{\dfrac{T}{T_\infty} + \dfrac{T_\mu}{T_\infty}}\left(\frac{T}{T_\infty}\right)^{\frac{3}{2}},$$

where $T_\mu = 110.4\ K$ for air. The Prandtl number $\mathrm{Pr} = \mu c_p/\lambda$ is assumed to be constant.

For the numerical integration, we use the dimensionless form of the Navier–Stokes equations. The Cartesian coordinates $x = \bar{x}L$, $y = \bar{y}L$, $z = \bar{z}L$ are related to the characteristic linear size L; time $t = \bar{t}L/V_\infty$, to the characteristic time L/V_∞; the velocity vector components $u = \bar{u}V_\infty$, $v = \bar{v}V_\infty$, and $w = \bar{w}V_\infty$, to the absolute value of the free-stream velocity V_∞; pressure $p = \bar{p}(\rho_\infty V_\infty^2)$, to twice the impact air pressure; and the remaining gas dynamics variables, to their values in the free-stream flow. The bar over a symbol means that the corresponding variable is dimensionless and symbol "∞" marks the value of this variable in the unperturbed flow. After this non-dimensionalization, the Navier–Stokes equations acquire the following similarity parameters: $\gamma = c_p/c_v$, the adiabatic exponent; $\mathrm{M}_\infty = V_\infty/a_\infty$, the free-stream Mach number (a is the speed of sound); $\mathrm{Re}_\infty = (\rho_\infty V_\infty L)/\mu_\infty$, the Reynolds number; and Pr, the Prandtl number. Most of the calculated data are presented in the dimensionless variables and the overbar is suppressed for simplicity.

On the boundary of the computational region, which coincides with the rigid surface, we impose the boundary conditions: the no-slip and nonpermeability conditions $u = 0$, $v = 0$, $w = 0$; the conditions of adiabaticity ($\partial T/\partial n\big|_w = 0$) or isothermality ($T = T_w = \mathrm{const}$) of the streamlined surface, or some conditions of thermal balance. On the outer boundary of computational region, radiation conditions corresponding to a diverging wave are imposed. These boundary conditions in terms of the Riemann invariants have the form

$$\alpha_1 = \frac{2a}{\gamma - 1} - \left(u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} + w\frac{\partial \eta}{\partial z}\right)\frac{1}{\Delta}, \quad \alpha_2 = \frac{p}{\rho^\gamma}, \quad \alpha_3 = u\frac{\partial \xi}{\partial x} + v\frac{\partial \xi}{\partial y} + w\frac{\partial \xi}{\partial z},$$

$$\alpha_4 = u\frac{\partial \zeta}{\partial x} + v\frac{\partial \zeta}{\partial y} + w\frac{\partial \zeta}{\partial z}, \quad \alpha_5 = \frac{2a}{\gamma - 1} + \left(u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} + w\frac{\partial \eta}{\partial z}\right)\frac{1}{\Delta},$$

$$\Delta = \sqrt{\left(\frac{\partial \eta}{\partial x}\right)^2 + \left(\frac{\partial \eta}{\partial y}\right)^2 + \left(\frac{\partial \eta}{\partial z}\right)^2}.$$

In this case, at each point on the boundary of the computational region, at $\eta = \eta_{\max}$, the signs of the eigenvalues

$$\lambda_1 = \left(u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} + w\frac{\partial \eta}{\partial z}\right)\frac{1}{\Delta} - a, \quad \lambda_2 = \left(u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} + w\frac{\partial \eta}{\partial z}\right)\frac{1}{\Delta},$$

$$\lambda_3 = \lambda_2, \quad \lambda_4 = \lambda_2, \quad \lambda_5 = \left(u\frac{\partial\eta}{\partial x} + v\frac{\partial\eta}{\partial y} + w\frac{\partial\eta}{\partial z}\right)\frac{1}{\Delta} + a,$$

which determine the direction of propagation of a perturbation with respect to $\eta = \text{const}$, are determined. For $\lambda_i \le 0$ ("input boundary"), the corresponding invariant on the input boundary is calculated from the values of the gas-dynamics variables of the free-stream flow; for $\lambda_i > 0$ ("output boundary"), we use the linear extrapolation of α_i from the values of the gas-dynamics variables corresponding to internal points of the computational region.

As the initial approximation, one can use the condition of a uniform free-stream flow followed by the development of the flow field in the process of solving the unsteady problem. In this case, with forming the flow pattern, the time step gradually increases, which makes it possible to obtain the solution of the steady problem.

2. APPROXIMATION OF THE EQUATIONS

The initial boundary value problem formulated above was solved numerically by an integro-interpolation method (the finite-volume method). Being applied to the Navier–Stokes equations (1), it enables one to obtain finite-difference analogs of the conservation laws:

$$\frac{3\mathbf{Q}^{n+1}_{i,j,k} - 4\mathbf{Q}^{n}_{i,j,k} + \mathbf{Q}^{n-1}_{i,j,k}}{\Delta t} + \frac{\mathbf{E}^{n+1}_{i+\frac{1}{2},j,k} - \mathbf{E}^{n+1}_{i-\frac{1}{2},j,k}}{h_\xi} + \frac{\mathbf{G}^{n+1}_{i,j+\frac{1}{2},k} - \mathbf{G}^{n+1}_{i,j-\frac{1}{2},k}}{h_\eta} + \frac{\mathbf{F}^{n+1}_{i,j,k+\frac{1}{2}} - \mathbf{F}^{n+1}_{i,j,k-\frac{1}{2}}}{h_\zeta} = 0, \tag{3}$$

where n is the index of the time layer; Δt is the time step; and i, j, k and h_ξ, h_η, h_ζ are indices of the nodes and step sizes in the coordinates ξ, η, ζ, respectively.

For a monotonic difference scheme, the fluxes at semi-integer nodes are calculated by solving the Riemann problem of the breakdown of an arbitrary discontinuity. Mathematically, this problem is reduced to solving a nonlinear system of algebraic equations. An approximate method for this problem may be the splitting in the generalized coordinates and representation of some averaged state of the corresponding Jacobi matrix $\mathbf{A}$ (for example, $\mathbf{A} = \partial\mathbf{E}/\partial\mathbf{Q}$ for the direction ξ, where $\mathbf{E}$ is the convective component of the corresponding flux in the diagonal form: $\mathbf{A} = \mathbf{R}\Lambda\mathbf{R}^{-1}$, Λ is a diagonal matrix whose entries are six eigenvaues of the matrix $\mathbf{A}$).

When approximating the convective component of the vector of fluxes $\mathbf{E}$, $\mathbf{G}$, and $\mathbf{F}$ at semi-integer nodes, we use a Godunov-type monotonic scheme (see [10, 11]) and the Roe approximate method (see [12]) for solving the Riemann problem of the breakdown of an arbitrary discontinuity. In this case, the computational formulas for the vectors $\mathbf{E}$, $\mathbf{G}$, and $\mathbf{F}$ are similar; therefore, below we consider the vector $\mathbf{E}$. On a face of a cell (at a semi-integer node), we have

$$\mathbf{E}_{i+\frac{1}{2}} = \frac{1}{2}(\mathbf{E}(\mathbf{Q}_L) + \mathbf{E}(\mathbf{Q}_R) - \mathbf{B}(\mathbf{Q}_{LR})\Lambda(\varphi(\lambda_i))\mathbf{B}(\mathbf{Q}_{LR})^{-1}(\mathbf{Q}_R - \mathbf{Q}_L)),$$

where $\Lambda(\phi(\lambda_i))$ is a diagonal matrix with the entries $\varphi(\lambda_i)$ and λ_i are the eigenvalues of the operator $\mathbf{A} = \partial\mathbf{E}/\partial\mathbf{Q}$, $\mathbf{B}_{LR} = \mathbf{B}(\mathbf{Q}_{LR})$ is a matrix whose columns are the eigenvectors of the operator $\mathbf{A}$, and $\mathbf{B}^{-1}_{LR} = \mathbf{B}(\mathbf{Q}_{LR})^{-1}$ is the matrix of left eigenvectors of the operator $\mathbf{A}$ (the inverse of the matrix $\mathbf{B}_{LR}$).

When calculating the eigenvalues and eigenvectors of the operator $\mathbf{A}$, the matrices $\Lambda(\varphi(\lambda_i))$, $\mathbf{B}_{LR}$, and $\mathbf{B}^{-1}_{LR}$ are determined from the values of the dependent variables, which have the form (see [12])

$$u_{LR} = \frac{u_L\sqrt{\rho_L} + u_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \quad v_{LR} = \frac{v_L\sqrt{\rho_L} + v_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \quad w_{LR} = \frac{w_L\sqrt{\rho_L} + w_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}},$$

$$H_{LR} = \frac{H_L\sqrt{\rho_L} + H_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \quad a^2_{LR} = (\gamma - 1)\left(H_{LR} - \frac{1}{2}(u^2_{LR} + v^2_{LR} + w^2_{LR})\right),$$

where a is the local speed of sound.

As the function $\varphi(\lambda_i)$, which provides the entropy condition for the physically correct choice of the numerical solution, we use the function

$$\varphi(\lambda) = \begin{cases} |\lambda|, & |\lambda| > \varepsilon, \\ \dfrac{\lambda^2 + \varepsilon^2}{2\varepsilon}, & |\lambda| \le \varepsilon, \end{cases}$$

where ε is the parameter responsible for the dissipative properties of the difference scheme (in most computations, we used $\varepsilon = 0.1$).

To increase the order of approximation (to the third one), the dependent variables are interpolated to the face of an elementary cell using the WENO (Weighted Essentially Non-Oscillatory scheme) principle [13]:

$$\mathbf{Q}_R = \omega_{0R}\mathbf{Q}_{0R} + \omega_{1R}\mathbf{Q}_{1R}, \quad \mathbf{Q}_{0R} = \frac{1}{2}(\mathbf{Q}_{i+1} + \mathbf{Q}_i), \quad \mathbf{Q}_{1R} = -\frac{1}{2}\mathbf{Q}_{i+2} + \frac{3}{2}\mathbf{Q}_{i+1},$$

$$\mathbf{Q}_L = \omega_{0L}\mathbf{Q}_{0L} + \omega_{1L}\mathbf{Q}_{1L}, \quad \mathbf{Q}_{0L} = \frac{1}{2}(\mathbf{Q}_i + \mathbf{Q}_{i+1}), \quad \mathbf{Q}_{1L} = -\frac{1}{2}\mathbf{Q}_{i-1} + \frac{3}{2}\mathbf{Q}_i,$$

$$\omega_{0R} = \frac{\alpha_{0R}}{\alpha_{0R} + \alpha_{1R}}, \quad \omega_{1R} = \frac{\alpha_{1R}}{\alpha_{0R} + \alpha_{1R}}, \quad \alpha_{0R} = \frac{d_0}{(\varepsilon + \beta_{0R})^2}, \quad \alpha_{1R} = \frac{d_1}{(\varepsilon + \beta_{1R})^2},$$

$$\omega_{0L} = \frac{\alpha_{0L}}{\alpha_{0L} + \alpha_{1L}}, \quad \omega_{1L} = \frac{\alpha_{1L}}{\alpha_{0L} + \alpha_{1L}}, \quad \alpha_{0L} = \frac{d_0}{(\varepsilon + \beta_{0L})^2}, \quad \alpha_{1L} = \frac{d_1}{(\varepsilon + \beta_{1L})^2},$$

$$\beta_{0R} = (\mathbf{Q}_{i+1} - \mathbf{Q}_i)^2, \quad \beta_{1R} = (\mathbf{Q}_{i+2} - \mathbf{Q}_{i+1})^2,$$

$$\beta_{0L} = (\mathbf{Q}_i - \mathbf{Q}_{i+1})^2, \quad \beta_{1L} = (\mathbf{Q}_i - \mathbf{Q}_{i-1})^2,$$

$$d_0 = \frac{2}{3}, \quad d_1 = \frac{1}{3}, \quad \varepsilon = 10^{-4}.$$

For approximating the diffusion component of the fluxes $\mathbf{E}$, $\mathbf{G}$, and $\mathbf{F}$ on the face of an elementary cell, we use the central-difference scheme, accurate in the second order. The derivatives are calculated by the formulas

$$\left.\frac{\partial \mathbf{U}}{\partial \xi}\right|_{i+\frac{1}{2},j,k} = \frac{1}{h_\xi}(\mathbf{U}_{i+1,j,k} - \mathbf{U}_{i,j,k}),$$

$$\left.\frac{\partial \mathbf{U}}{\partial \eta}\right|_{i+\frac{1}{2},j,k} = \frac{1}{4h_\eta}(\mathbf{U}_{i+1,j+1,k} + \mathbf{U}_{i,j+1,k} - \mathbf{U}_{i+1,j-1,k} - \mathbf{U}_{i,j-1,k}),$$

$$\left.\frac{\partial \mathbf{U}}{\partial \varsigma}\right|_{i+\frac{1}{2},j,k} = \frac{1}{4h_\varsigma}(\mathbf{U}_{i+1,j,k+1} + \mathbf{U}_{i,j,k+1} - \mathbf{U}_{i+1,j,k-1} - \mathbf{U}_{i,j,k-1});$$

here, $\mathbf{U} = (u, v, w, p, T)^{\mathrm{T}}$ is the vector of nonconservative ("primitive") dependent variables of the problem.

The stencil of the difference scheme on which the full Navier–Stokes equations are approximated consists of 25 points. The resulting completely implicit nonlinear difference scheme, apparently, is stable on the linear problem.

3. SOLVING THE NONLINEAR DIFFERENCE EQUATIONS

After applying the above finite-difference approximation to the Navier–Stokes equations and the corresponding boundary conditions on a certain grid, the integration of the nonlinear equations is reduced to solving a system of nonlinear algebraic equations

$$\mathbf{R}(\mathbf{X}) = 0, \tag{4}$$

where $\mathbf{X}$ is the vector of the sought variables (the node values of the gas dynamics variables, including the boundary nodes of the computational grid). The above-formulated problem is efficiently solved by the

well-known iterative Newton method, whose main advantage is the quadratic convergence rate. For solving nonlinear difference equations (4), we used a modified Newton–Raphson method:

$$\mathbf{X}^{[k+1]} = \mathbf{X}^{[k]} - \tau_{k+1}\mathbf{D}_{k_o}^{-1}\mathbf{R}(\mathbf{X}^{[k]}),$$

where $\mathbf{D}_{k_0} = (\partial\mathbf{R}/\partial\mathbf{X})_{k_0}$ is the Jacobi matrix, k and k_0 are the iteration indices, $k_0 \le k$, and $\mathbf{R}(\mathbf{X}^{[k]})$ is the residual vector. The expression $\mathbf{D}_{k_o}^{-1}\mathbf{R}(\mathbf{X}^{[k]}) = \mathbf{Y}^{[k]}$ is the solution to the system of linear equations $\mathbf{D}_{k_o}\mathbf{Y}^{[k]} = \mathbf{R}(\mathbf{X}^{[k]})$. In the process of numerical solution, the parameter of regularization of the Newton method with respect to the initial approximation τ_k is determined by the formula (see [14])

$$\tau_{k+1} = \frac{(\mathbf{Y}^{[k]} - \mathbf{Y}^{[k-1]}, \mathbf{Y}^{[k]})}{(\mathbf{Y}^{[k]} - \mathbf{Y}^{[k-1]})^2}.$$

With the convergence of the iterative process, we have $\tau_k \to 1$ and the convergence rate theoretically becomes quadratic.

The most computation-intensive elements of the algorithm in the implementation of the Newton method is generating the matrix $\mathbf{D}_{k_0} = (\partial\mathbf{F}/\partial\mathbf{X})_{k_0}$ and solving the system of algebraic equations with this matrix.

Since the approximation of the equations in each of the computational cells involves only several neighboring nodes (in the spatial case, 25 nodes for the TVD scheme), the computational complexity of generating the Jacobi matrix is on the order of $O(N)$, where N is the number of nodes of the difference problem. The Jacobi matrix is formed in each iteration by finite increments of the residual vector in the vector of sought grid variables. This method is universal, because it is easily extended to an arbitrary system of difference equations of a preliminary undefined form. Rather frequently, the difference equations obtained by the approximation of differential equations are complicated and the forming the Jacobi matrix analytically is very labor-consuming. In particular, this occurs when the Navier–Stokes equations are solved by means of monotonized schemes. Moreover, when forming the Jacobi matrix analytically, the necessary number of arithmetic and logical computer operations, generally speaking, may be greater than when this matrix is formed by the method of finite increments.

The storage and processor time required for solving the system of linear algebraic equations in a nonlinear iteration,

$$(\partial\mathbf{R}/\partial\mathbf{X})_{k_0}\mathbf{Y}^{[k]} = \mathbf{R}(\mathbf{X}^{[k]}),$$

essentially depend on the sparseness of the matrix $(\partial\mathbf{R}/\partial\mathbf{X})_{k_0}$. If the Navier–Stokes equations are approximated by the second-order difference scheme described in Section 2, the operator $(\partial\mathbf{R}/\partial\mathbf{X})_{k_0}$ has a sparse 25-diagonal block structure and its elementary block is a 5×5 dense matrix. Preliminary calculations have shown that the convergence of the iterative method essentially depends on the points in the stencil of approximation used for the convective component and the direct derivatives of the dissipative component of the Navier–Stokes equations. The use of "corner" points in the stencil of approximation of mixed derivatives makes only a small effect on the convergence rate of the nonlinear iterative process. Therefore and in order to reduce almost by a half the required computer storage and the total number of arithmetic operations in a nonlinear iteration, the diagonals corresponding to mixed derivatives in the operator $(\partial\mathbf{R}/\partial\mathbf{X})$ were omitted. As a result, the operator $(\partial\mathbf{R}/\partial\mathbf{X})$ for the spatial case has a sparse 13-diagonal block structure.

The system of linear algebraic equations obtained in a nonlinear iteration is solved by the generalized minimum residual method *GMRes* (see [14]), which, in a number of numerical experiments (see [16]), has been recognized as most reliable and fast.

4. COMPUTATIONAL GRIDS AND THE INTERNAL DATA STRUCTURE

The calculations are performed on structured multi-block grids. The blocks of the grid must join one another with entire faces, node to node.

Inside each block of the grid, independently of other blocks, we introduce an individual curvilinear coordinate system (ξ, η, ζ), in which the computational grid is uniform on each direction with the step sizes h_ξ, h_η, h_ζ, respectively. It is assumed that the curvilinear coordinates vary within each block from 0 to 1.

In this curvilinear coordinate system, discretization (3) of the gas dynamics equations is performed. The conversion to the Cartesian coordinates is carried out according to (2) by means of the inverse metric coefficients $\partial\xi/\partial x$, $\partial\xi/\partial y$, $\partial\xi/\partial z$, $\partial\eta/\partial x$, $\partial\eta/\partial y$, $\partial\eta/\partial z$, $\partial\eta/\partial z$, $\partial\zeta/\partial x$, $\partial\zeta/\partial y$, and $\partial\zeta/\partial z$, determined from the equation

$$\begin{pmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\xi}{\partial y} & \frac{\partial\xi}{\partial z} \\ \frac{\partial\eta}{\partial x} & \frac{\partial\eta}{\partial y} & \frac{\partial\eta}{\partial z} \\ \frac{\partial\zeta}{\partial x} & \frac{\partial\zeta}{\partial y} & \frac{\partial\zeta}{\partial z} \end{pmatrix} \times \begin{pmatrix} \frac{\partial x}{\partial\xi} & \frac{\partial x}{\partial\eta} & \frac{\partial x}{\partial\zeta} \\ \frac{\partial y}{\partial\zeta} & \frac{\partial y}{\partial\eta} & \frac{\partial y}{\partial\zeta} \\ \frac{\partial z}{\partial\xi} & \frac{\partial z}{\partial\eta} & \frac{\partial z}{\partial\zeta} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Here, the forward metric coefficients (the derivatives $\partial x/\partial\xi$, $\partial y/\partial\xi$, $\partial z/\partial\xi$, $\partial x/\partial\eta$, $\partial y/\partial\eta$, $\partial z/\partial\eta$, $\partial z/\partial\xi$, $\partial z/\partial\eta$, $\partial z/\partial\zeta$) are approximated by finite differences with a second order of accuracy.

The nodes of the grid inside each block have a three-dimensional index (i, j, k) corresponding to the directions of the grid lines ξ, η, ζ, respectively. In addition, to each node, a global one-dimensional index unique for all blocks of the computational grid is assigned. This global index determines the index of an element in the vector of solution $\mathbf{X}$ and the residual vector $\mathbf{R}$ when solving the system of difference equations (4).

In the discretization of the equations (calculation of the residual vector), near the boundaries of each block, the values of the dependent variables at the nodes of the adjacent blocks are needed. The access to these nodes is performed with using the concept of shadow values. Each block is preliminary extended to several grid surfaces according to the number of points in the discretization stencil so that these new shadow values correspond to the nodes of the neighboring block. The dependent variables at the shadow nodes are not calculated in the process of solving but are taken from the adjacent block in a special procedure of exchange with shadow values, which is executed before each time iteration. The additional shadow nodes are considered belonging to the current block, take into account its sizes, and have their local three-dimensional indices (i, j, k), but their global indices remain the same as those of "real" nodes in the adjacent block. Therefore, the total number of nodes in the computational region remains the same.

The mapping between the local three-dimensional indices (including shadow nodes) and global indices for each block of the grid is stored in a special list of indices. This list is the base for the work of the procedure of exchange with the shadow values.

5. IMPLEMENTATION OF PARALLEL COMPUTATIONS ON MULTIPROCESSOR SUPERCOMPUTERS

The distributed computations were implemented with the help of the parallelization on the block level. During the computation, each processor processes at least one block, and the data of blocks (the grid and the solution at the nodes) is independently stored by each processor. If the number of processors exceeds the number of blocks in the given grid, an additional division of the grid into a necessary number of blocks is needed. The computation results and the computational grid are stored in the form of CGNS (CFD General Notation System) files [17].

In the case of parallel computations, the calculated vector of solution with the global numeration of elements becomes distributed, i.e., its parts are stored in different processors. Therefore, the residual vectors of the nonlinear grid equations and the Jacobi matrix must be distributed too. Operations with such vectors and matrices are implemented with the help of subroutines from the PETSc freeware package (Portable, Extensible Toolkit for Scientific Computation) [18]. For parallel computations, the PETSc employs the Message Passage Interface (MPI). In the current computational code, some procedures of the MPI are employed in pure form, but all parallel operations are mainly implemented by means of PETSc commands.

The discretization of the equations in each block is performed independently by each processor. In this case, a fraction of the globally distributed residual vector is formed. The connection between processors corresponds to the connectivity between blocks and is provided by the conception of shadow nodes. The exchange with shadow values is implemented by means of intrinsic structures and the "Scatter" functions from the PETSc. In this case, the mapping list between the global interblock indices and the local intrablock indices, formed on the preprocessing of the grid, is used.

The Jacobi matrix in each iteration of solving the difference equations is formed in a parallel manner by finite increments of the residual vector in the sought grid variables. The resulting Jacobi matrix is fully distributed. The linear system with this matrix is solved by a parallel version—implemented in PETSc—of the generalized minimum residual method *GMRes* (see [15]) with a block Jacobi-type preconditioner.

Thus, each stage of the implemented numerical method is executed in parallel in each block of the computational domain. Therefore, the acceleration of computations is close to perfectly linear, if the delays for the communication between processors are neglected. Preliminary computations have shown that the PETSc parallel procedures efficiently pass data between processors so that the scaling proves to be really good.

The relative delays for the interprocessor communication can be reduced by increasing the time spent by each processor for the processing of its own set of data independently from the others. In other word, one should increase the dimensions of grid blocks and/or their number per processor.

The method described above was implemented in the HSFlow (High Speed Flow) software package.

6. VERIFICATION OF THE METHOD

To verify the method developed, we used the results of [6], obtained by another numerical method for solving the Navier–Stokes equations. In [6], hybrid nonmonotonic difference schemes accurate in the fourth and higher orders were employed. The characteristics of the free-stream flow in this work were the same as in [6], namely, $M_\infty = 5.35$, $Re_{\infty,1} = 14.3\times10^6$ 1/m, $\gamma = 1.4$, $Pr = 0.71$, and $T_\infty^* = 64.3163$ K. The non-dimensionalization length in the calculations was $L = 0.1$ m; in this case, $Re_\infty = 1.43\times10^6$. These parameters correspond to their experimental values on the edge of the boundary layer on a sharp cone, which was studied in a shock tube at $M_\infty = 6.0$, $Re_{\infty,1} = 11.0\times10^6$ 1/m, and $T_0^* = 433.0$ K.

The computational domain was a rectangular parallelepiped with the dimensions $4.5\times0.9293\times0.4$ in the horizontal, vertical, and cross directions, respectively, in dimensionless units. On the lower boundary of the domain, coinciding with the surface of the plate, the no-slip conditions were imposed. The wall was assumed isothermic with a temperature $T_w = 300$ K ($\overline{T}_w = 4.664$). On the boundaries $x = x_{max}$ (right) and $z = z_{max}$ (front), we used the linear extrapolation of the dependent variables u, v, w, p, and T (the nonreflecting boundary condition in a supersonic flow). On the boundary $z = 0$ (back), the condition of symmetry was imposed. The computations were performed on an orthogonal grid with the dimensions $2501\times251\times141$ in the horizontal, vertical, and cross directions, respectively (with a total of 88.5 million nodes). Near the streamlined surface, the grid was condensed in the vertical direction so that 120 nodes fell within the boundary layer. The computations were performed on a high-performance computer cluster involving 512 processor cores.

Perturbations were introduced into the boundary layer through a small round hole on the surface of the plate by forced pulsations of the vertical velocity component. These pulsations were specified by the transient boundary condition on the wall:

$$v_w = A\cos^3\left(\pi\frac{x-0.5(x_2+x_1)}{x_2-x_1}\right)\cos^3\left(\pi\frac{z}{z_2-z_1}\right)\sin(-\omega t),$$
$$x_1 \le x \le x_2, \quad z_1 \le z \le z_2, \quad 0 \le t \le t_1,$$

where $x_1 = 1.36$, $x_2 = 1.40$, $z_1 = -0.02$, $z_2 = 0.02$, $t_1 = 2\pi/\omega$, and $\omega = \omega^* L^*/U_\infty^*$ is the dimensionless frequency. In order to provide the nonlinear development of perturbations, the amplitude was taken sufficiently large: $A = 0.05$ (5% of the free-stream velocity). The above form and parameters of perturbations fully correspond to those used in [6].

The problem was solved in two stages. At first, the stationary unperturbed flow field was calculated. Then perturbations were introduced by switching the unsteady boundary condition simulating an actuator.

The forced perturbations introduced into the boundary layer by a short local pulse form a three-dimensional wave packet propagating downstream. Figure 1 shows an instant pressure perturbation field on the surface of the plate after a sufficiently large time since the pulse. Hereinafter, the perturbation field is calculated by subtracting the basic steady field from the instant perturbed field. We see that the wave fronts forming a packet are rectified near the central line $z = 0$. Here, the domination of plane waves with the amplitudes increasing downstream begins. This behavior agrees with the linear theory of stability, according to which plane waves (the so-called second Mack mode waves) are amplified faster than oblique ones.

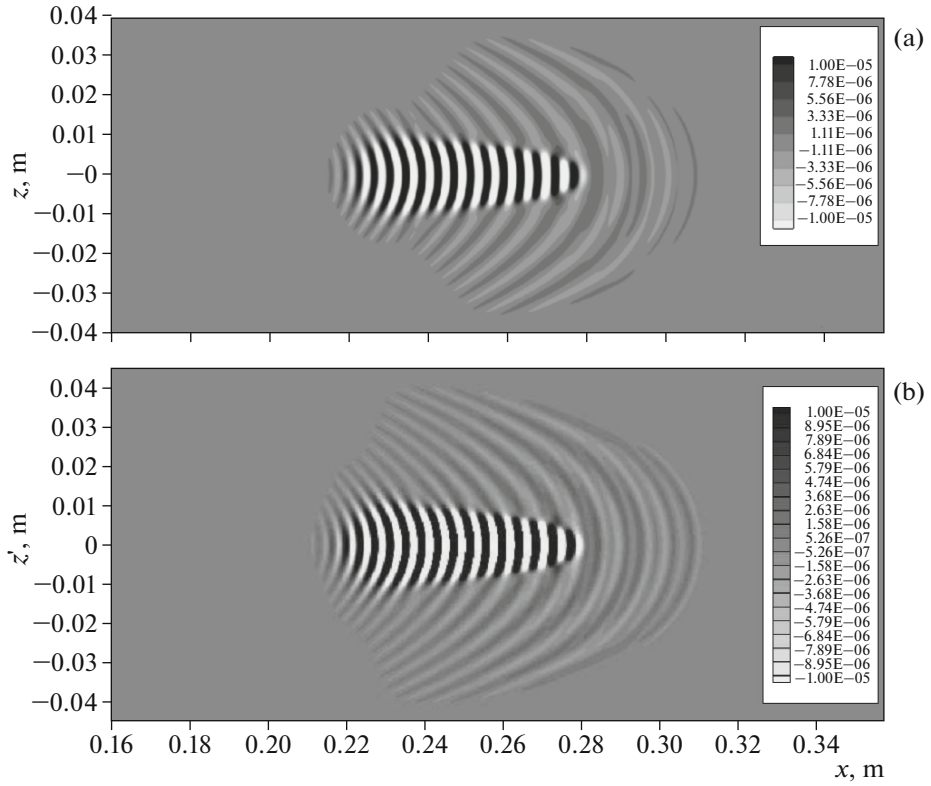

Fig. 1. Verification of computational codes; the comparison with [6]. Instant perturbation pressure field (the trace of a wave packet) at the time $t = 0.174$ ms: (a) – the current work and (b) [6].

Beside the central part of the packet with pronounced plane fronts, waves running in front of the main perturbation are observed.

For comparison, Fig. 1 also shows the perturbation field obtained in [6]. We see not only a good qualitative agreement between the results but also the same structural features of the perturbation and a quantitative agreement in the dominating wave length. Thus, the numerical method applied in this work can be used for studying the development of three-dimensional perturbations in hypersonic flows.

7. DIRECT NUMERICAL SIMULATION OF THE INITIAL STAGE OF THE LAMINAR–TURBULENT TRANSITION

As an example of calculation of the laminar–turbulent transition, we consider a hypersonic laminar boundary layer over a plate in which perturbations are artificially introduced by periodic injection–suction of gas through a local region on the wall.

7.1. Parameters of the Flow and the Computational Domain

The calculations are performed for a hypersonic flow above a sharp flat plate with a free-stream Mach number $M_\infty = 5.373$, a Reynolds number $\mathrm{Re}_{\infty,1} = U^*_\infty \rho^*_\infty / \mu^*_\infty = 17.9 \times 10^6$ 1/m, and temperature $T^*_\infty = 74.194$ K. The wall is isothermic with a surface temperature $T^*_w = 300$ K; $T_w = 4.043$. These parameters of the flow fully correspond to the parameters used in [9], where the compression angle was studied. The coordinates are related to the characteristic scale $L = 0.3161$ m (in [9], this is the distance from the leading edge to the line of fracture of the streamlined surface), which corresponds to a Reynolds number $\mathrm{Re}_\infty = 5.667 \times 10^6$. The computational domain in the plane (x, y) is shown in Fig. 2; in the cross direction (z) it has a depth of 0.2 dimensionless units.

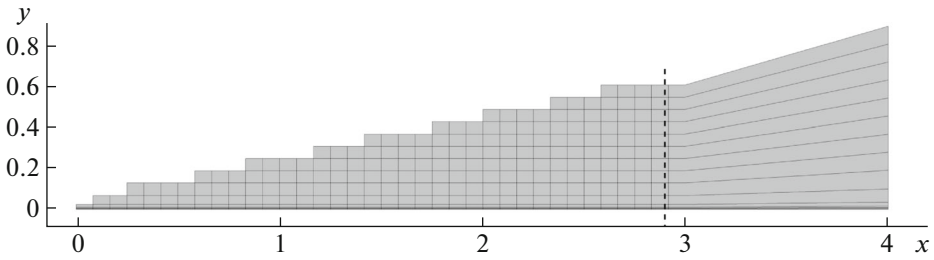

Fig. 2. Computational domain in the plane (x, y). The boundaries of blocks in which the grid is divided for parallel computations are shown. The dashes line marks the boundary of the confidence domain.

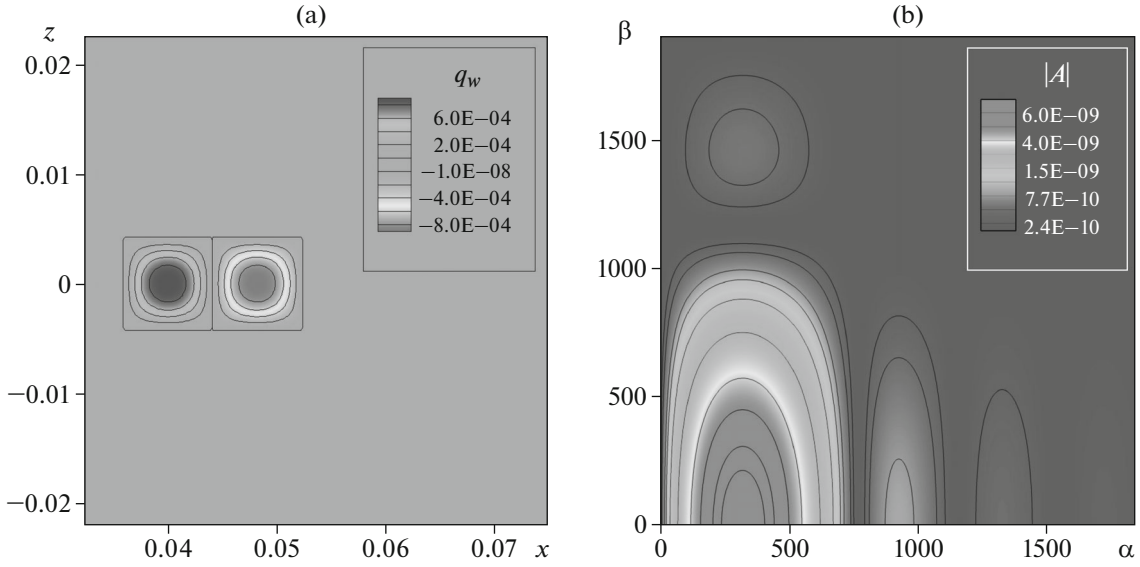

Fig. 3. (a) Spatial form and (b) spectrum of the signal generator.

The boundary conditions are as follows: the no-slip conditions $u = v = w = 0$ on the lower ($y = y_{\min}$) boundary of the computational domain; the free-stream conditions $u = 1$, $v = 0$, $p = 1/\gamma M_\infty^2$, and $T = 1$ on the left ($x = x_{\min}$) and upper ($y = y_{\max}$) boundaries; the linear extrapolation $f_i - 2f_{i-1} + f_{i-2} = 0$ from inside the domain for the dependent variables u, v, w, p, and T on the right ($x = x_{\max}$) and front ($z = z_{\max}$) boundaries ("soft" boundary conditions); the conditions of symmetry $\partial u/\partial n = \partial v/\partial n = \partial p/\partial n = \partial T/\partial n = 0$ and $w = 0$ on the rear ($z = z_{\min}$) boundary.

The computations were performed on an orthogonal $6001 \times n_y \times 151$ grid, in which the number of nodes n_y in the vertical direction varied from 126 to 376 over the length of the computational domain, depending on the position of the shock wave. The grid contains a total of 251×10^6 nodes. The grid is condensed near the surface so that 55% of nodes lie within the boundary layer. The size of 160 cells near the right edge of the computational domain gradually increases from $\Delta x = 0.0005$ to $\Delta x = 0.1$. This part of the computational domain (for $x > 2.917$) forms a "buffer" zone, the solution in which is not used for the analysis but where the unsteady perturbations must dissipate due to grid viscosity. The resulting three-dimensional grid is obtained by the equidistant multiplication of the 2D grid.

7.2. The Generator of Perturbations

The problem is solved in two steps. At first, the stationary laminar flow field (the main flow) is calculated by the relaxation method. Then, on the steady solution, we impose a unsteady perturbation: a per-

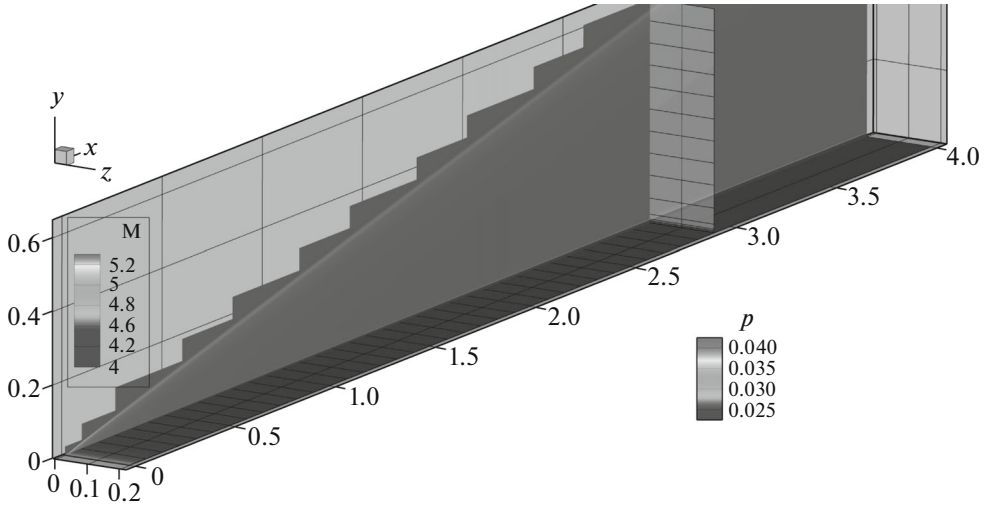

Fig. 4. Steady laminar streamlining of a plate. The Mach number field in the plane of symmetry and the pressure field on the surface. The vertical plane at $x \approx 2.9$ marks the beginning of the "buffer" zone of the computational domain excluded from the analysis, where the cells of the grid have increased sizes for dissipation of ingoing perturbations.

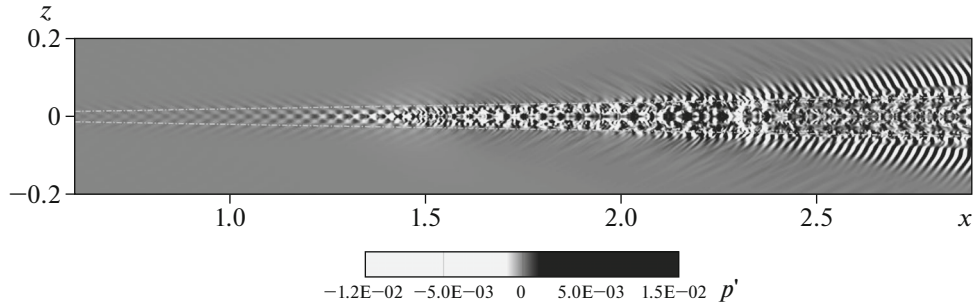

Fig. 5. Instant pressure perturbation field on the surface at $t = 5.704$. The dashed lines mark the boundaries of the turbulent wedge within which the averaging is performed.

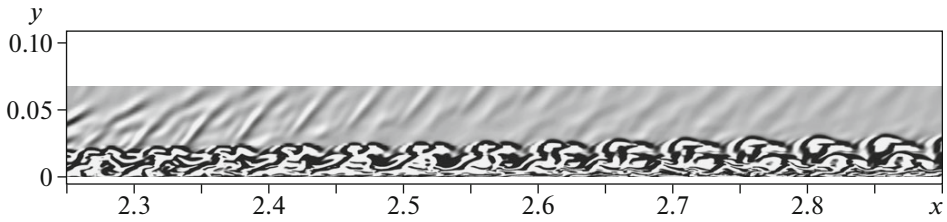

Fig. 6. Instant shadow pattern of perturbations of the boundary layer in the plane of symmetry $z = 0$, $t = 7.3125$.

manently acting injection–suction through two holes in the wall, which is simulated by the following boundary condition on the perturbation of the vertical mass flux:

$$q_w(x,z,t) = (\rho v)_w = \varepsilon \sin\left(2\pi \frac{x - x_1}{x_2 - x_1}\right) \sin\left(\pi \frac{z - z_1}{z_2 - z_1}\right) \sin(\omega_0 t),$$

$$x_1 \le x \le x_2, \quad z_1 \le z \le z_2, \quad 0 \le t < \infty,$$

where $x_1 = 0.0358$ and $x_2 = 0.0521$ are the boundaries of the domain of the action in the longitudinal direction (the same as in [9]) with the center at $x_0 \simeq 0.044$ and $z_2 = -z_1 = (x_2 - x_1)/4 = 4.075 \times 10^{-3}$ are the boundaries in the cross direction. This gives a rectangular region with an aspect ratio of 2 (see Fig. 3a).

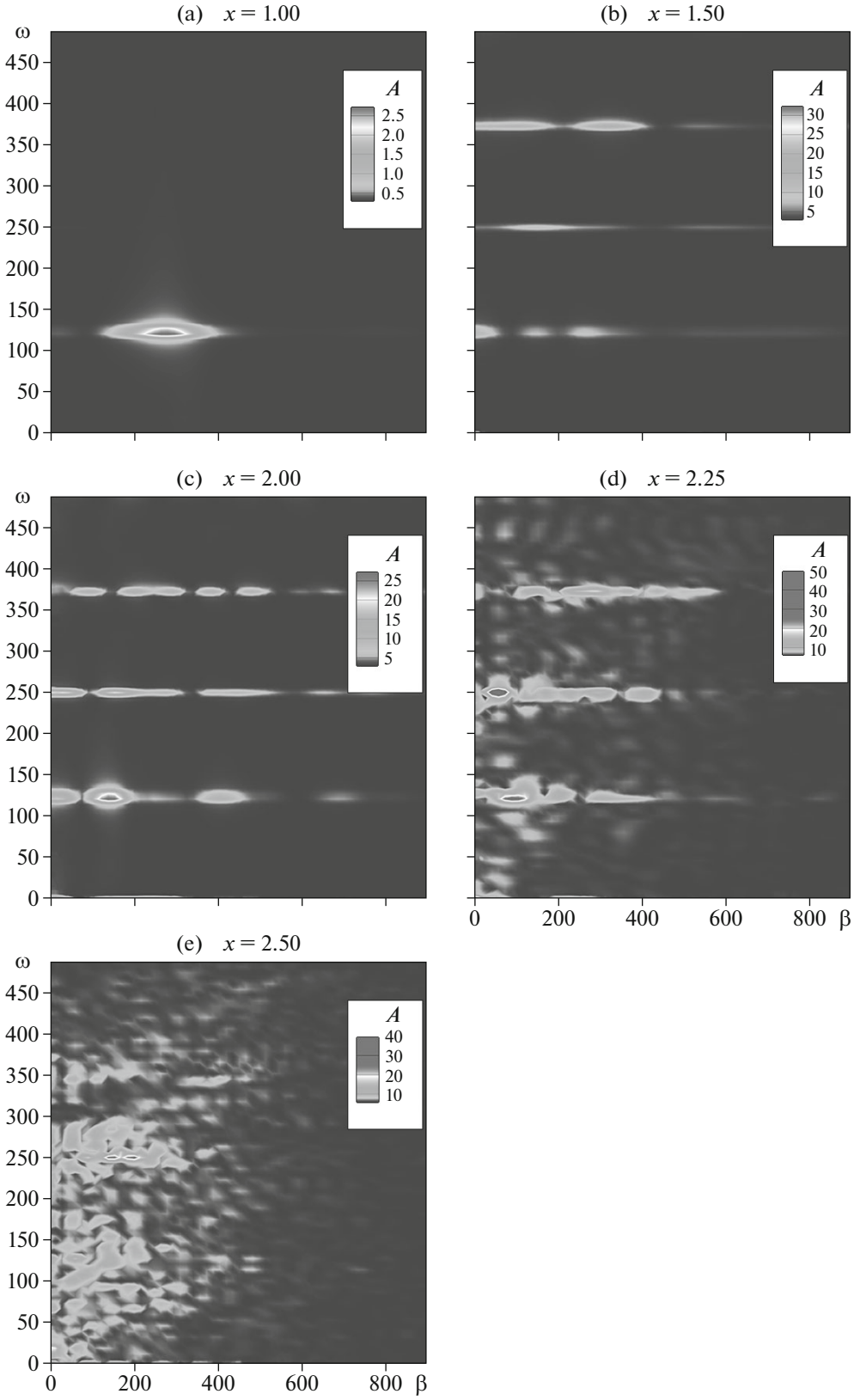

Fig. 7. Frequency and wavenumber spectra for different x.

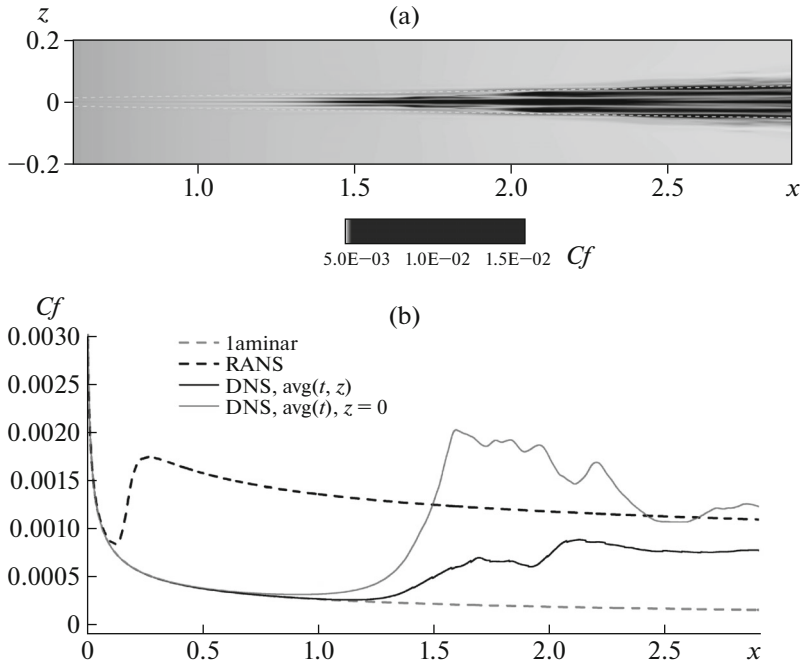

Fig. 8. Time-averaged friction coefficient *Cf* on the surface: (a) the *Cf* field in the (x–z) plane; (b) (laminar) *Cf* in the unperturbed flow; (RANS) *Cf* in a two-dimensional flow calculated via the Reynolds-averaged Navier–Stokes equations; (DNS, avg(t, z)) *Cf* time- and amplitude-averaged within the wedge marked by the dashed lines; (DNS, avg(t), $z = 0$) *Cf* time-averaged on the line $z = 0$.

The perturbation from the generator develops into a three-dimensional wave train propagating downstream. The generator is situated at a relatively small distance from the front edge so that the induced perturbation propagates through the regions of instability of both the first and second modes. This choice is also motivated by the assumption that, in natural conditions, instabilities are excited most efficiently near the front edge.

The calculations were performed at a frequency $\omega_0 = \omega_0^* L^*/U_\infty^* = 125$, corresponding to the frequency parameter $F_0 = \omega_0/\mathrm{Re}_\infty = 2.206\times10^{-5}$, typical of the instability of the first Mack mode (see [1]). We consider the level of forcing $\varepsilon = 10^{-3}$, at which perturbations initially develop in a linear manner. It should be noted that the numerical error of the steady solution should be much smaller than the amplitude of the perturbation, i.e., the steady field should be calculated with a high accuracy.

The spatial form of the forcing action and its Fourier spectrum (the absolute values of the Fourier transform) are shown in Fig. 3, where α is the x-component and β is the z-component of the wavenumber. In the spectrum of the generator in the (α,β)-plane (see Fig. 3b), we see the presence of the components of the action at $\beta = 0$ with a wide spectrum with respect to α. They can generate longitudinal vortices.

7.3. Results

The calculated stationary laminar flow field around the plate is shown in Fig. 4. We see a front shock wave forming on the leading edge as a result of the displacing action of the boundary layer. The vertical size of the computational domain was chosen so that the shock wave do not intersect the upper face, in order to avoid the interaction with the boundary condition and possible nonphysical reflection. In the figure, we see a damping "buffer" zone with strongly increased grid cells. The solution in it is not exact and is not shown in the subsequent figures.

Figure 5 shows the instant perturbation field of nondimensional pressure on the surface of the plate. Hereinafter, the field calculated in the region $0 \le z \le 0.2$ is mirrored to $-0.2 \le z \le 0$ (in accordance with the symmetry condition at $z = 0$). The perturbation field is obtained by subtracting the main flow field from the field at a given instant of time.

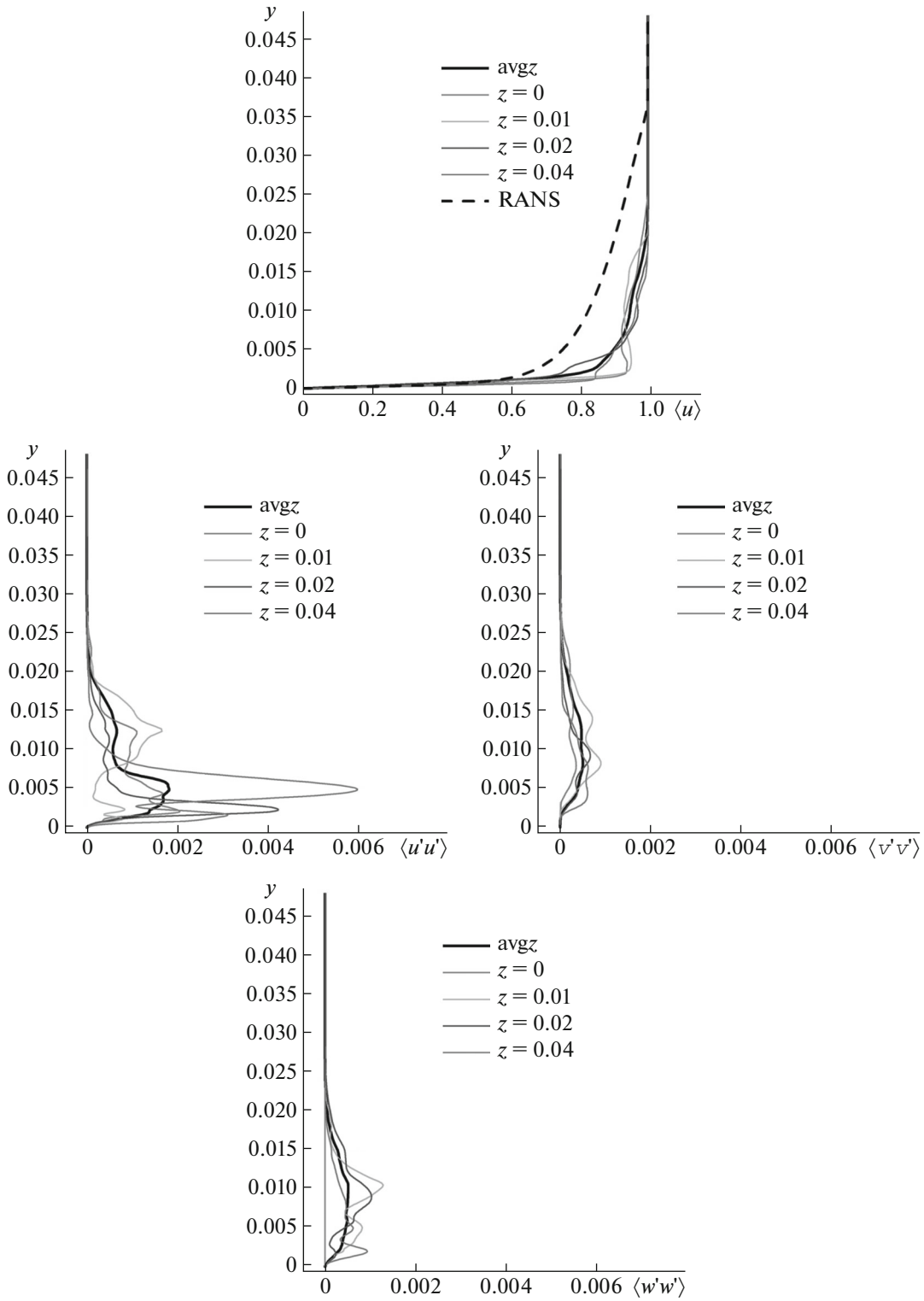

Fig. 9. Vertical profiles of quantities time-averaged in the cross-section $x = 2.5$; avg$z$ is the profile amplitude-averaged within the turbulent wedge: $z = 0.01$, 0.02, 0.03, and 0.04 are the profiles at the corresponding values of the z-coordinate corresponding to the amplitude.

We see that, up to the cross section $x = 1.5$, the core of the perturbation has a chess-like structure and, on the edges of the core, oblique waves dominate. This is typical of perturbations of the first unstable mode of the boundary layer. We also see low-amplitude circular wave fronts propagating far from the wave

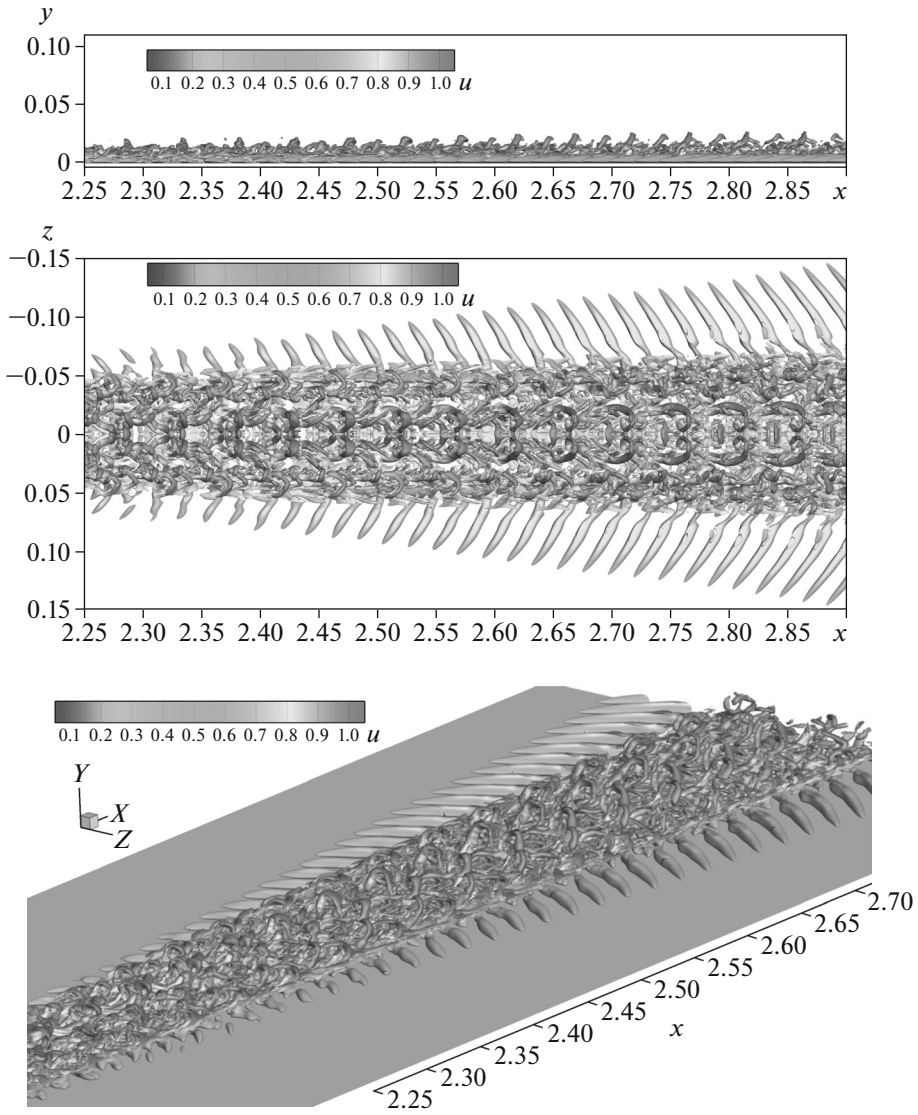

Fig. 10. Vortex structures in the perturbed boundary layer at the time $t = 7.3125$: visualization by a Q-criterion isosurface ($Q = 100$) colored in accordance with the horizontal velocity component; (top) side view, (middle) top view, and (bottom) perspective view.

train. Apparently, these waves are fast acoustic waves of the continuous spectrum. Downstream from the cross section $x = 1.5$, the regular structure of the core of perturbations begins the breakdown, so that downstream from $x = 2.25$, the perturbations at the center take a completely chaotic form: the laminar–turbulent transition begins and a "young" turbulent wedge forms. The form of perturbations in the core of the turbulent wedge in the plane of symmetry $z = 0$ is shown in Fig. 6 by a shadow pattern (the Laplacian of density, $\nabla^2\rho$).

The two-dimensional spectra in the $\beta - \omega$ plane are shown in Fig. 7 in different cross sections with respect to x. In the cross section $x = 1.00$, in the perturbation, oblique waves with the transverse wavenumber $\beta \approx 300$ and the frequency $\omega = 125$ dominate. These waves correspond to the instability of the first mode. At $x = 1.50$, harmonics with frequencies multiple of the main frequency appear, which is connected with the nonlinear interaction of harmonics. At $x = 2.00$, the waves of multiple frequencies are amplified and the zero-frequency harmonic corresponding to the variation in the mean field appears—the boundary layer begins reconstruction. At $x > 2.25$, the spectrum becomes substantially wider and harmonics of aliquant frequencies appear, which indicates nonlinear breakdown.

The onset of the laminar-turbulent transition can be confirmed by the value of the surface friction coefficient

$$c_f = \tau_w^* / \frac{1}{2}\rho^* V_\infty^* = \left[\mu \frac{\partial u}{\partial n}\right]_w / \frac{1}{2}\mathrm{Re}_\infty$$

on the wall of the plate. Figure 8 shows the field of the friction coefficient $c_f(x, z)$ time-averaged over the interval $5.7005 \le t \le 6.6265$ of dimensionless units. We see that, downstream from the cross section $x \approx 1.0$, $c_f(x, z)$ begins to increase in the longitudinal direction, which is typical of the laminar–turbulent transition. The increase varies in the spanwise direction: in the $(x–z)$ plane, we see banded structures. Figure 8 also shows the distribution $c_f(x)$ obtained on solving the Reynolds-averaged Navier–Stokes equations with using the two-parametric $q - \omega$ model of turbulence. This gives an estimate of $c_f(x)$ for a fully turbulent boundary layer. We see that the $c_f(x)$ obtained in the direct numerical simulation approaches the turbulent value.

Figure 9 shows the pulsation characteristics of a perturbed flow in the cross section $x = 2.5$. Here, the time averaging is performed in the interval $6.6255 \le t \le 7.3205$. We see that the profile of the mean velocity qualitatively resembles the case of a fully turbulent boundary layer (the RANS line).

Figure 10 shows three-dimensional vortex structures in the form of Q-criterion isosurfaces $Q = 100$, colored according to the horizontal velocity component. Small-scale hairpin vortices fill the central part of perturbations and are surrounded with oblique waves. As a whole, vortices of many scales are present, which is typical of a turbulent boundary layer, but some regular structure is observed, i.e., the flow is not yet fully turbulized.

This example demonstrates that the numerical method enables one to simulate the process of transition from the linear phase up to the end of the nonlinear breakdown.

CONCLUSIONS

A method for the direct numerical simulation of unsteady three-dimensional perturbations leading to the laminar–turbulent transition at hypersonic flow velocities has been developed and implemented. The simulation is based on solving the full three-dimensional transient Navier–Stokes equations. The numerical method is oriented to application of supercomputers and is based on implicit monotonic approximation schemes and a modified Newton–Raphson method for solving nonlinear difference equations.

The direct numerical simulation of the propagation of three-dimensional perturbations in the boundary layer on a plate at a free-stream Mach number 5.35 has been performed. Perturbations were introduced into the boundary layer by means of an unsteady boundary condition on the vertical velocity component, which simulate a short pulse through a circular hole. This pulse formed a three-dimensional wave packet propagating downstream in the boundary layer. It has been shown that, in this packet, plane waves corresponding to the second unstable Mack mode dominate. Weak nonlinear effects of the interaction between plane and oblique waves have been demonstrated. A good agreement with a similar numerical study from [6] has been obtained, which confirms the correctness of the method employed in this work and its applicability to the study of the development of three-dimensional perturbations in hypersonic boundary layers.

With the help of this method, the development of three-dimensional perturbations in the boundary layer on a plane plate at a free-stream Mach number M = 5.37 has been studied. Typical structures of the perturbation field in the linear and nonlinear stages of the laminar–turbulent transition have been revealed. The mean viscous friction coefficient distribution in the linear and transient parts of the streamlined surface has been obtained, making it possible to determine the beginning of the LTT.

ACKNOWLEDGMENTS

This work was carried out at the Moscow Institute of Physics and Technology and was supported by the Russian Science Foundation (project no. 14-19-00821, numerical study and analysis of results) and the Russian Foundation for Basic Research (project no. 14-08-00793, development of the algorithm and programs for numerical simulation).

REFERENCES

1. A. Fedorov, "Transition and stability of high-speed boundary layers," Annu. Rev. Fluid Mech. **43**, 79–95 (2011).
2. O. M. Belotserkovskii, *Constructive Modeling of Structural Turbulence and Hydrodynamic Instabilities* (World Scientific, Singapore, 2009).
3. X. Zhong and X. Wang, "Direct numerical simulation on the receptivity, instability, and transition of hypersonic boundary layers," Annu. Rev. Fluid Mech. **44**, 527–561 (2012).
4. I. V. Yegorov and O. L. Zaitsev, "An approach to the numerical solution of the bidimensional Navier–Stokes equations using the direct calculation method," USSR Comput. Math. Math. Phys. **31** (2), 80–89 (1991).
5. I. V. Yegorov and D. V. Ivanov, "The use of fully implicit monotone schemes to model plane internal flows," Comput. Math. Math. Phys. **36** (12), 1717–1730 (1996).
6. J. Sivasubramanian and H. F. Fasel, "Transition initiated by a localized disturbance in a hypersonic flat-plate boundary layer," AIAA Paper, No. 2011-374 (2011).
7. I. V. Egorov, A. V. Fedorov, and V. G. Soudakov, "Direct numerical simulation of disturbances generated by periodic suction-blowing in a hypersonic boundary layer," Theor. Comput. Fluid Dyn. **20** (1), 41–54 (2006).
8. I. V. Egorov, A. V. Novikov, and A. V. Fedorov, "Numerical simulation of stabilization of the boundary layer on a surface with a porous coating in a supersonic separated flow," J. Appl. Mech. Tech. Phys. **48** (2), 176–183 (2007).
9. I. V. Egorov, A. V. Novikov, and A. V. Fedorov, "Numerical modeling of the disturbances of the separated flow in a rounded compression corner," Fluid Dyn. **41** (4), 521–530 (2006).
10. S. K. Godunov, "Difference method for computing discontinuous solutions of fluid dynamics equations," Mat. Sb. **47** (3), 271–306 (1959).
11. S. K. Godunov, A. V. Zabrodin, M. Ya. Ivanov, A. N. Kraiko, and G. P. Prokopov, *Numerical Solution of Multidimensional Problems in Gas Dynamics* (Nauka, Moscow, 1976) [in Russian].
12. P. L. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes," J. Comput. Phys. **43**, 357–372 (1981).
13. C. W. Jiang, "Shu efficient implementation of weighted ENO schemes," J. Comput. Phys. **126**, 202–228 (1996).
14. T. Kh. Karimov, "On certain iterative methods for solving nonlinear equations in Hilbert spaces," Dokl. Akad. Nauk SSSR **269** (5), 1038–1046 (1983).
15. Y. Saad and M. H. Shultz, "GMRES: A generalized minimal residual algorithm for solving non symmetric linear systems," SIAM J. Sci. Stat. Comput. **7** (3), 856–869 (1986).
16. I. Yu. Babaev, V. A. Bashkin, and I. V. Egorov, "Numerical solution of the Navier–Stokes equations using variational iteration methods," Comput. Math. Math. Phys. **34** (11), 1455–1462 (1994).
17. CFD General Notation System, http://cgns.org/ (April 28, 2015).
18. S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, PETSc web page, http://www.mcs.anl.gov/petsc (September 20, 2015).

Translated by E. Chernokozhin