ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2008, Vol. 48, No. 9, pp. 1600–1619. © Pleiades Publishing, Ltd., 2008.
Original Russian Text © I.A. Vaseva, V.D. Liseikin, Yu.V. Likhanova, Yu.N. Morokov, 2008, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2008, Vol. 48, No. 9, pp. 1638–1658.

Some Aspects of Grid Generation

I. A. Vaseva, V. D. Liseikin, Yu. V. Likhanova, and Yu. N. Morokov

Institute of Computing Technologies, Siberian Branch, Russian Academy of Sciences, pr. akademika Lavrent'eva 6, Novosibirsk; ul. Pirogova, 2, 630090 Russia

e-mail: vaseva.irina@gmail.com, lvd@ict.nsc.ru, yulia.likhanova@gmail.com, quant@ict.nsc.ru

Received February 26, 2007; in final form, December 20, 2007

Abstract—New results concerning the development of a universal method for grid generation based on the numerical solution of the inverted Beltrami and diffusion equations with respect to the monitor metric are obtained. In order to build monitor metrics, layer-type functions are used. Algorithms for generating smoothly matched block grids are proposed. Examples of two- and three-dimensional grids for the tokamak edge region, for calculation of a passive impurity in the atmosphere, and for the numerical solution of two-dimensional singularly perturbed problems are presented.

DOI: 10.1134/S0965542508090108

Keywords: grid generation methods, numerical solution of inverted Beltrami and diffusion equations, generation of smoothly matched block grids.

INTRODUCTION

Comprehensive grid generation techniques are extremely important for modern computational physics and applied mathematics because they can provide a basis for developing unified automated codes for grid generation. Such techniques and codes promise a considerable advancement in studying physical phenomena having a nonhomogeneous multiscale structure in domains with a complex geometric shape. In particular, this pertains to nuclear power engineering, to studying properties of high-temperature plasma in a tokamak chamber with the purpose of creating energy sources based on thermonuclear fusion and plasma processes, and so on. In designing aircraft, it is very important to have grid generation codes that could, in combination with other numerical techniques, complement or even replace experiments in a wind tunnel; indeed, numerical calculations are much cheaper than experiments in wind tunnels. The development of efficient codes for the rapid generation of grids with prescribed properties in domains around aircraft and spacecraft will enable one to quickly determine promising configurations using numerical methods; then, those configurations can be fine-tuned using wind tunnels and full-scale experiments. As a result, the cost of designing aircraft with optimal characteristics can be significantly reduced and the design and manufacturing time can be shortened. Moreover, automated codes are required for the numerical simulation of chemical reactions and biological processes in such systems as human organs, for predicting natural calamities, and so on.

The problem of developing automated technologies for grid generation has not been adequately solved yet, and it is becoming increasingly urgent in computational mathematics, applied physics, power engineering, chemistry, and biology. This is explained by the need to use complex three-dimensional domains and by the more stringent requirements for the grid quality and generation time so that the calculations can be efficient. The basic grid properties that are important in applications are as follows:

(1) regularity (orthogonality of the grid lines, nondegeneracy, and insignificant cell deformation);

(2) smoothness (small differences in the shape and size of adjacent cells);

(3) grid clustering in the regions where the functions undergo sharp changes (shock and rarefaction waves, boundary and interior layers, phase transition regions, and contact boundaries);

(4) grid clustering in the regions where the function values are of primary concern (regions where the approximation or interpolation errors are large and where the sea is shallow in tsunami study and prediction);

(5) alignment of the grid lines with a vector field (magnetic fields in studying the plasma behavior in a tokamak chamber).

The natural efficiency requirements for the grid generation codes are as follows:

(1) reliability (simplicity and convenience);

(2) comprehensiveness (implementation of all the required grid properties and the ability for rapid and unified grid generation for various applications in domains with boundaries of complex geometry, such as tokamak chambers, regions around aircraft, internal human organs and systems, and so on);

(3) automation (minimal manual work spent for debugging codes when generating grids for new domains and problems);

(4) invariance (the grid point distribution should be independent of parameterizations of physical geometries).

The development of automated codes for the generation of grids having the desired properties is a difficult task. In order to be able to satisfy the requirements for the codes and grid quality, an effective basic mathematical technology is needed that can ensure both basic and balanced grid properties. A promising technology that is able to provide a basis for developing codes satisfying all the above requirements is the technology based on the numerical solution of systems of inverted Beltrami and diffusion equations [1]. This technology is based on achievements in Riemannian geometry, the theory of elliptic equations, variational calculus, and computational mathematics. Fundamental studies on the theoretical justification and numerical implementation of this technology, as well as some experimental results, can be found in [1–3].

In this technology, the monitor metric in the inverted Beltrami and diffusion equations is the main tool for controlling the grid properties. These equations provide a basis for formulating the mathematical model for determining the grid nodes and cells in arbitrary domains and on arbitrary surfaces encountered when studying various phenomena in science and engineering. This model enables one to generate both steady and moving structured and unstructured grids in domains of arbitrary dimensions and on their boundaries independently of the parameterization of the physical geometries. The desired grid properties are achieved by introducing a monitor metric. The Beltrami and diffusion equations and operators are popular in Riemannian geometry, and many of their properties are well understood. In particular, some relationships between the metric invariants and the characteristics of the coordinate surfaces are obtained by solving these equations. For example, such an important grid characteristic as the rate of the grid clustering near a boundary surface is determined in terms of the values of the Beltrami operator (the second differential Beltrami parameter) and the mean curvature of this surface (see [1]). The knowledge of such properties helps explicitly define a metric for controlling grids. Moreover, the solutions to those equations are optimal for the inverted energy and diffusion functionals (see [1, 2]), which enables one to use not only finite difference or finite element approximations of equations but also apply variational methods for finding the location of grid nodes (see [4]). These advantages of the mathematical model based on the Beltrami and diffusion equations provide confidence that the research aimed at creating unified automated program packages for grid generation are promising. Presently, the development of such packages is an important problem in applied mathematics, computational physics, mechanics, biology, and medicine both in Russia and in other countries.

The development of comprehensive technologies and software for automatic grid generation has been mainly performed in the USA beginning in the 1990s (see [5]) in the framework of the global national initiative aimed at creation of computing centers and automated computer technologies for the numerical solution of applied problems (see [6]). Those technologies are based on the numerical solution of the inverted Poisson equations for structured grid generation and on the front and Delaunay methods for the generation of unstructured grids. However, as applied to grid generation, the Poisson equations are not as efficient as the Beltrami and diffusion equations; in particular, they do not guarantee that the coordinate transformation is nonsingular (because the Poisson operator is not conservative); neither do they guarantee that those transformations are independent of the parameterizations of geometries. In addition, controlling the grid properties using weighting functions in the Poisson equations is a difficult problem because no sound theoretical results concerning the relationship between the weighting functions and the grid characteristics are available. As a result, one should select appropriate weighting functions by hand for each new physical domain in order to obtain nondegenerate grids with desired characteristics. Under this approach, debugging computer programs for the generation of grids in domains with complex boundary shapes, such as domains around aircraft, takes several months of collective efforts of large teams of researches. For these reasons, the method proposed and developed by the authors of this paper is more promising in terms of its convenience, comprehensiveness, and possibility to be automated compared to the methods based on the numerical solution of the inverted Poisson equations.

In this paper, we describe the elements of a global grid code designed for grid generation pertaining to the technology of generating local one-block grids and to the methods of their smooth matching for obtaining multiblock grids.

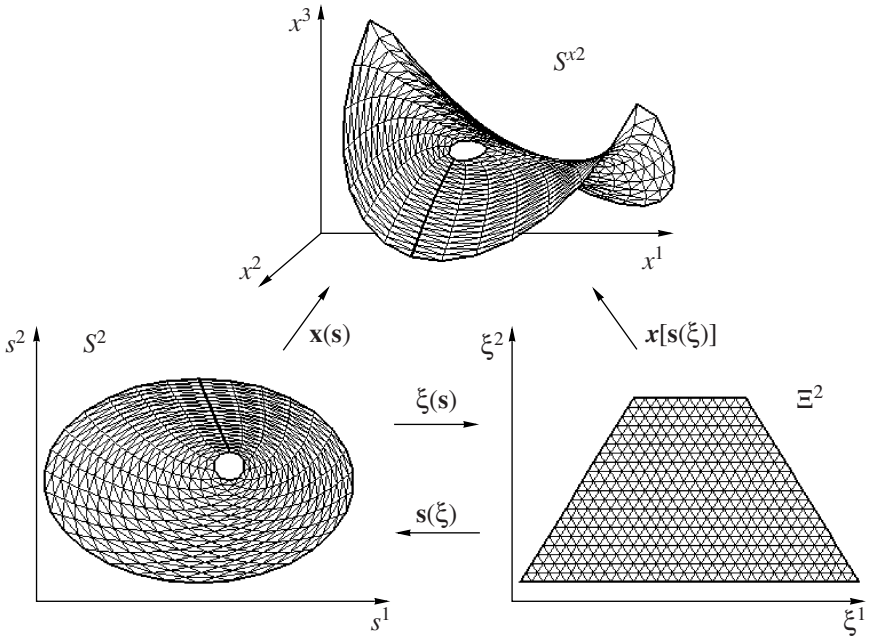

Fig. 1. The scheme of surface triangular grid generation.

1. MATHEMATICAL MODEL

1.1. Scheme of the Mapping Approach

The proposed grid generation mathematical model is formulated in the framework of the mapping technology, which can be used for an arbitrary n-dimensional physical geometry $S^{xn} \subset \mathbb{R}^{n+k}$ (curve, domain, or surface) that is locally specified by the parameterization

$$\begin{gathered}\mathbf{x}(\mathbf{s}) : S^n \longrightarrow \mathbb{R}^{n+k},\\ \mathbf{x} = (x^1, \ldots, x^{n+k}), \quad \mathbf{s} = (s^1, \ldots, s^n), \quad n \geq 1,\end{gathered} \tag{1.1}$$

where S^n is an n-dimensional parametric domain (interval for $n = 1$) and $\mathbf{x}(\mathbf{s})$ is a smooth vector function of rank n at every point $\mathbf{s} \in S^n$ (see Fig. 1). When $k = 0$, the surface S^{xn} is a domain $X^n \subset \mathbb{R}^n$. In this case, X^n can be considered as a parametric domain (see Fig. 2).

In the mapping approach, the grid in physical geometry S^{xn} is generated using the intermediate nonsingular transformation

$$\mathbf{s}(\boldsymbol{\xi}) : \Xi^n \longrightarrow S^n, \quad \boldsymbol{\xi} = (\xi^1, \ldots, \xi^n) \tag{1.2}$$

from the parametric domain S^n to the corresponding computational (logical) domain Ξ^n of a simple shape that makes it possible to analytically determine points of the reference grid (Figs. 1, 2) [1, 3, 7–9]. The grid nodes in S^{xn} are determined by the following mapping of the reference grid defined on Ξ^n:

$$\mathbf{x}[\mathbf{s}(\boldsymbol{\xi})] : \Xi^n \longrightarrow S^{xn} \in \mathbb{R}^{n+k}. \tag{1.3}$$

The shape of the computation domain Ξ^n and of the cells of the reference grid is chosen depending on the shape of S^{xn} and on the numerical algorithm (finite differences, finite elements, spectral methods) used to solve the physical problem. In particular, Ξ^n can be a standard n-dimensional cube with rectangular cells (Fig. 2), a trapezoid (triangle) with triangular cells (Fig. 1), a prism, or a frustum of a pyramid. Moreover, the reference grid in Ξ^n may be unstructured.

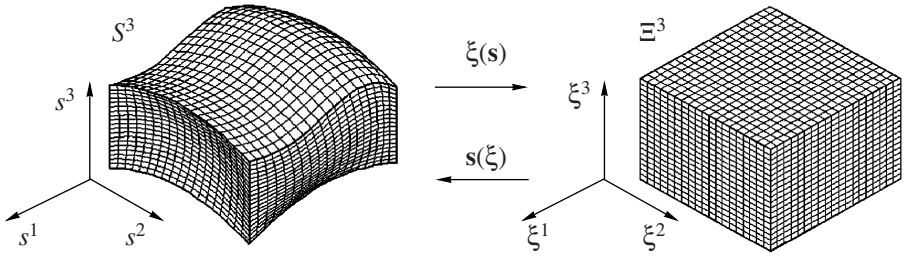

Fig. 2. The scheme of thee-dimensional hexahedral grid generation.

1.2. Grid Equations

In the version of the mapping approach considered in this paper, the intermediate transformation $\mathbf{s}(\boldsymbol{\xi})$) is defined as the mapping inverse of the vector function

$$\boldsymbol{\xi}(\mathbf{s}) : S^n \longrightarrow \Xi^n, \quad \boldsymbol{\xi}(\mathbf{s}) = [\xi^1(\mathbf{s}), \ldots, \xi^n(\mathbf{s})]$$

(see Fig. 1), which is a solution of the Dirichlet problem

$$\begin{gathered}\frac{\partial}{\partial s^j}\left(w(\mathbf{s})g_{\mathbf{s}}^{jk}\frac{\partial \xi^i}{\partial s^k}\right) = 0, \quad i, j, k = 1, 2, \ldots, n, \\ \xi^i\big|_{\partial S^n} = \varphi^i(\mathbf{s}), \quad i = 1, 2, \ldots, n.\end{gathered} \tag{1.4}$$

Here, $g_{\mathbf{s}}^{jk}$ are the contravariant components of the prescribed monitor metric $g_{ij}^{\mathbf{s}}$ on the set of points of the physical geometry S^{xn} defined in the parametric coordinates $s^1, \ldots, s^n$ obtained from (1.1); ∂S^n and $\partial\Xi^n$ are the boundaries of S^n and Ξ^n, respectively; $\boldsymbol{\varphi}(\mathbf{s}) = [\varphi^1(\mathbf{s}), \ldots, \varphi^n(\mathbf{s})]$ is a one-to-one mapping between ∂S^n and $\partial\Xi^n$; and $w(\mathbf{s}) > 0$ is a weighting function that increases or decreases the influence of the metric in the specified regions of S^{xn}.

Here and in what follows, we assume that, in the expressions involving identical indexes and not containing the symbols + or –, the summation is performed over those indexes.

The equations in problem (1.4) are called the diffusion equations with respect to the monitor metric $g_{ij}^{\mathbf{s}}$. When $w(\mathbf{s}) = \sqrt{g^{\mathbf{s}}}$, where $g^{\mathbf{s}} = \det(g_{ij}^{\mathbf{s}})$, the diffusion equations in (1.4) are actually the Beltrami equations in the metric $g_{ij}^{\mathbf{s}}$. Moreover, when $n \neq 2$, the diffusion equations in (1.4) are, in fact, the Beltrami equations in the metric (see [1])

$$g_{ij} = (g^{\mathbf{s}})^{\frac{1}{2-n}}[w(\mathbf{s})]^{\frac{2}{n-2}}g_{ij}^{\mathbf{s}}, \quad i, j = 1, 2, \ldots, n,$$

independently of the weighting function $w(\mathbf{s})$.

Although the Beltrami equations are comprehensive (that is, any nonsingular twice differentiable intermediate transformation (1.2) can be found as the inverse of the solution to these equations, see [1]), the diffusion equations in (1.4) make it possible to more simply ensure the desired grid properties in various regions of S^{xn}, especially in the case $n = 2$.

The diffusion equations in (1.4) are the Euler–Lagrange equations for the diffusion functional

$$I[\boldsymbol{\xi}] = \frac{1}{2}\int\limits_{S^n} w(\mathbf{s})g_{\mathbf{s}}^{jk}\frac{\partial \xi^i}{\partial s^k}\frac{\partial \xi^i}{\partial s^j}d\mathbf{s} = \frac{1}{2}\int\limits_{S^n} w[s(\boldsymbol{\xi})]g_{\boldsymbol{\xi}}^{ij}d\mathbf{s}, \quad i, j, k = 1, 2, \ldots, n, \tag{1.5}$$

where

$$g_{\boldsymbol{\xi}}^{ij} = g_{\mathbf{s}}^{kl}\frac{\partial \xi^i}{\partial s^k}\frac{\partial \xi^j}{\partial s^l}, \quad i, j, k, l = 1, 2, \ldots, n,$$

are the contravariant elements of the monitor metric in the coordinates $\xi^1, \ldots, \xi^n$.

For $w(\mathbf{s}) = \sqrt{g^{\mathbf{s}}}$, (1.5) is the energy functional for the mapping $\boldsymbol{\xi}(\mathbf{s}) : S^n \longrightarrow \Xi^n$.

1.3. Inverted Equations

To find the grid nodes, the equations in (1.4) must be inverted to obtain equations with respect to the intermediate transformation $\mathbf{s}(\boldsymbol{\xi})$. As a result, transformed problem (1.4) has the form (see [1])

$$w[\mathbf{s}(\boldsymbol{\xi})]g_{\boldsymbol{\xi}}^{kl}\frac{\partial^2 s^i}{\partial\xi^k\partial\xi^l} = \frac{\partial}{\partial s^j}(w(\mathbf{s})g_{\mathbf{s}}^{ji}),$$
$$s^i\big|_{\partial\Xi^n} = \psi^i(\boldsymbol{\xi}), \quad i, j, k, l = 1, 2, \ldots, n. \tag{1.6}$$

Equations (1.6) will be called the inverted diffusion equations.

The inverted diffusion equations are equivalent to the following equations in conservative form (see [1]):

$$\frac{\partial}{\partial\xi^j}\{Jw[\mathbf{s}(\boldsymbol{\xi})]g_{\boldsymbol{\xi}}^{ij}\} = 0, \quad i, j = 1, 2, \ldots, n. \tag{1.7}$$

Here, $J = \det(\partial s^i/\partial\xi^j)$. Substituting $\sqrt{g^{\mathbf{s}}}$ for $w(\mathbf{s})$ in these equations, we obtain the Beltrami equations in conservative form.

Equations (1.7) are the Euler–Lagrange equations for the inverted diffusion functional $I[\mathbf{s}]$, which is obtained by replacing the integration domain S^n with Ξ^n in functional (1.5); that is

$$I[\mathbf{s}] = \frac{1}{2}\int\limits_{\Xi^n} Jw[\mathbf{s}(\boldsymbol{\xi})]g_{\mathbf{s}}^{kl}\frac{\partial\xi^i}{\partial s^k}\frac{\partial\xi^i}{\partial s^l}d\boldsymbol{\xi}, \quad i, k, l = 1, 2, \ldots, n. \tag{1.8}$$

When $w(\mathbf{s}) = \sqrt{g^{\mathbf{s}}}$, this is the inverted energy functional. Therefore, the inverted energy functional has the form

$$I[\mathbf{s}] = \frac{1}{2}\int\limits_{\Xi^n} \sqrt{g^{\boldsymbol{\xi}}}g_{\mathbf{s}}^{kl}\frac{\partial\xi^i}{\partial s^k}\frac{\partial\xi^i}{\partial s^l}d\boldsymbol{\xi}, \quad i, k, l = 1, 2, \ldots, n, \tag{1.9}$$

where $g^{\boldsymbol{\xi}} = \det(g_{ij}^{\boldsymbol{\xi}}) = (J)^2 g^{\mathbf{s}}$.

Functional (1.9) for the metric of the monitor surface, that is, for

$$g_{ij}^{\mathbf{s}} = g_{ij}^{xs} + \frac{\partial\mathbf{f}}{\partial s^i}\cdot\frac{\partial\mathbf{f}}{\partial s^j}, \quad i, j = 1, 2, \ldots, n,$$

where $\mathbf{f}(\mathbf{s})$ is a vector function defined at the points of the physical geometry S^{xn} in the coordinates $s^1, \ldots, s^n$, was first proposed in [11] for generating grids with node clustering in the zones of sharp variation of $\mathbf{f}(\mathbf{s})$. For the case when S^{xn} is a two-dimensional domain S^2 and $\mathbf{f}(\mathbf{s})$ is a scalar function $f(\mathbf{s})$, that is, for the case

$$g_{ij}^{\mathbf{s}} = \delta_j^i + \frac{\partial f}{\partial s^i}\frac{\partial f}{\partial s^j}, \quad i, j = 1, 2, \tag{1.10}$$

functional (1.9) proposed in [11] was successfully used for generating adaptive grids in gas dynamics, in particular, in [13, 14]. For the numerical solution of the minimization problem for (1.9) with metric (1.10), a generalization of the method proposed in [15] for the Euler metric (i.e., for $g_{ij}^{\mathbf{s}} = \delta_j^i$, $i, j = 1, 2$) was used.

In the general case, boundary value problem (1.6), as well as the minimization problem for inverted functional (1.9), can have no solutions. Note that, for any monitor metric, the equations in (1.4) are linear and elliptic; hence, boundary value problem (1.4) and the minimization problem of functional (1.5) have a unique smooth solution $\boldsymbol{\xi}(\mathbf{s}) : S^n \longrightarrow \Xi^n$. Therefore, if the solution $\mathbf{s}(\boldsymbol{\xi}) : \Xi^n \longrightarrow S^n$ of problem (1.6) and of the

minimization problem for functional (1.9) exists, then the Jacobian of the transformation $\mathbf{s}(\boldsymbol{\xi})$ is nonsingular because $\mathbf{s}(\boldsymbol{\xi})$ is the inverse of $\boldsymbol{\xi}(\mathbf{s})$.

Note that, for the inverted Beltrami equations, i.e., for $w(\mathbf{s}) = \sqrt{g^{\mathbf{s}}}$, boundary value problem (1.6) and the minimization problem for inverted energy functional (1.9) are well posed for any monitor metric $g^{\mathbf{s}}_{ij}$ for $n = 1$ and 2 if, for $n = 2$, Ξ^2 is convex and homeomorphic to S^2 (see [16]). However, in the case of the Beltrami equations for $n = 3$ and for the diffusion equations for $n = 2$ and 3, the problem of determining the monitor metric that ensures the existence and, therefore, the nonsingularity of the transformation $\mathbf{s}(\boldsymbol{\xi}) : \Xi^n \longrightarrow S^n$, is open.

1.4. Finite Difference Algorithm

Solutions of the nonlinear boundary value problem (1.6) are found using the following algorithm. First, multiply the left- and the right-hand sides of the inverted equations by $(J)^2$ or by $g^{\boldsymbol{\xi}} = \det(g^{\mathbf{s}}_{ij}) = g^{\mathbf{s}}J^2$; thus, we avoid the division by the Jacobian J when changing the derivatives $\partial\xi^i/\partial s^j$, which appear in the expressions for $g^{ij}_{\boldsymbol{\xi}}$ and $\dfrac{\partial}{\partial s^j} = \left(\dfrac{\partial \xi^k}{\partial s^j}\right)\left(\dfrac{\partial}{\partial \xi^k}\right)$ in these equations, for the derivatives $\partial s^k/\partial\xi^m$. For definiteness, multiply system (1.6) by $(J)^2$. This operation enables us to perform numerical computations of the grid nodes using iterative methods for singular initial and intermediate transformations (see [1]). Note that we cannot avoid the division by J in functional (1.5) and in conservative equations (1.7). As a result, in order to generate a grid using the numerical solution of Eqs. (1.7) and of the optimization problem for functional (1.5), one must use a complicated technique (see [1, 12]) for avoiding the division by zero if the transformation is singular at certain points.

The transformed elliptic problem is replaced by the parabolic boundary value problem with respect to the functions $s^i(\boldsymbol{\xi}, t)$ $(i = 1, 2, \ldots, n)$

$$\begin{gathered}\frac{\partial s^i}{\partial t} = J^2\left(w[\mathbf{s}(\boldsymbol{\xi})]g^{kl}_{\boldsymbol{\xi}}\frac{\partial^2 s^i}{\partial\xi^k\partial\xi^l} - \frac{\partial\xi^m}{\partial s^j}\frac{\partial}{\partial\xi^m}\{w[\mathbf{s}(\boldsymbol{\xi})]g^{ij}_{\mathbf{s}}\}\right),\\ s^i(\boldsymbol{\xi}, t) = \psi^i(\boldsymbol{\xi}), \quad \boldsymbol{\xi}\in\partial\Xi^n, \quad t\geq 0,\\ s^i(\boldsymbol{\xi}, 0) = s^i_0(\boldsymbol{\xi}), \quad \boldsymbol{\xi}\in\Xi^n, \quad i, j, k, l, m = 1, 2, \ldots, n,\end{gathered} \tag{1.11}$$

where $s^i_0(\boldsymbol{\xi})$ is the ith component of the initial transformation

$$\mathbf{s}_0(\boldsymbol{\xi}) : \Xi^n \longrightarrow S^n, \quad \mathbf{s}_0(\boldsymbol{\xi}) = [s^1_0(\boldsymbol{\xi}), \ldots, s^n_0(\boldsymbol{\xi})],$$

and the mapping

$$\boldsymbol{\psi}(\boldsymbol{\xi}) = [\psi^1(\boldsymbol{\xi}), \ldots, \psi^n(\boldsymbol{\xi})] : \partial\Xi^n \longrightarrow \partial S^n$$

is the inverse of the mapping in (1.4):

$$\boldsymbol{\varphi}(\boldsymbol{\xi}) : \partial S^n \longrightarrow \partial\Xi^n.$$

In the case when the right-hand side of system (1.11) is an elliptic operator, the solution of problem (1.11) tends to the solution of (1.6) as $t \longrightarrow \infty$.

Define the operator $L[\mathbf{v}]$ and the vector function $\mathbf{f}$ by

$$L[\mathbf{v}] = \sum_{m,\, p\, =\, 1, 2, \ldots, n} L_{mp}[\mathbf{v}], \quad \mathbf{f} = (f^1, \ldots, f^n),$$

where

$$L_{mp}[\mathbf{v}] = J^2 w[\mathbf{s}(\boldsymbol{\xi})]g^{mp}_{\boldsymbol{\xi}}\frac{\partial^2\mathbf{v}}{\partial\xi^m\partial\xi^p},$$

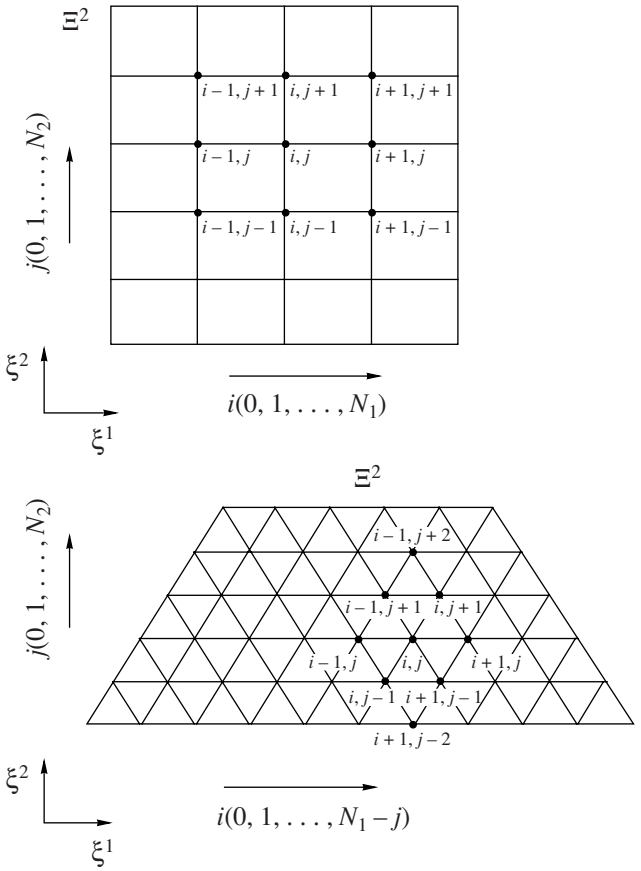

Fig. 3. The two-dimensional square and triangular stencil for the finite difference method.

$$f^i = -J^2 \frac{\partial \xi^k}{\partial s^j} \frac{\partial}{\partial \xi^k}(w[\mathbf{s}(\boldsymbol{\xi})] g_{\mathbf{s}}^{ij}[\mathbf{s}(\boldsymbol{\xi})]),$$

$$i, j, k, m, p = 1, 2, \ldots, n, \quad i, m, p \text{ are fixed.}$$

In this notation, the equations in (1.11) take the form

$$\partial \mathbf{s}/\partial t = L[\mathbf{s}] + \mathbf{f}. \tag{1.12}$$

For $n = 2$, system (1.12) can be numerically solved on the grids depicted in Fig. 3 using the stabilizing correction method (see [10]). More precisely, the solution is found using the scheme

$$\frac{\mathbf{s}^{k+1/2} - \mathbf{s}^k}{\tau} = L_{11}^h[\mathbf{s}^{k+1/2}] + 2L_{12}^h[\mathbf{s}^k] + L_{22}^h[\mathbf{s}^k] + \mathbf{f}^k,$$

$$\frac{\mathbf{s}^{k+1} - \mathbf{s}^{k+1/2}}{\tau} = L_{22}^h[\mathbf{s}^{k+1}] - L_{22}[\mathbf{s}^k].$$

Here, $L_{ij}^h[\mathbf{v}]$ is the finite difference operator that approximates the operator $L_{ij}[\mathbf{v}]$ by central differences. The initial transformation $\mathbf{s}(\boldsymbol{\xi}, 0) = \mathbf{s}_0(\boldsymbol{\xi})$ is found using the transfinite interpolation (see [1]).

In the three-dimensional case, the scheme is similar.

2. MONITOR METRIC

To efficiently control the grid properties, we use the Beltrami and diffusion equations to define monitor metrics on the set of points of the physical geometry S^{xn}. The metrics are defined in terms of the variables

of the physical problem, the geometric characteristics of S^{xn}, given vector fields, and the like; that is, they are defined in terms of the quantities to which the grid should be adapted. The monitor metrics must be simple and sufficiently universal so as to be able to produce all the required grids. We use the following metric tensors (see [1]) as the monitor metric satisfying these requirements:

$$g^{\mathbf{s}}_{ij} = \epsilon(\mathbf{s})g^{xs}_{ij} + B^k_i B^k_j, \quad i, j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, l, \tag{2.1}$$

or

$$g^{ij}_{\mathbf{s}} = z(\mathbf{s})g^{ij}_{sx} + D^i_k D^j_k, \quad i, j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, l. \tag{2.2}$$

Here, g^{xs}_{ij} (g^{ij}_{sx}) are the covariant (contravariant) components of the physical geometry S^{xn}, $\mathbf{B}_k = (B^1_k, \ldots, B^n_k)$ ($\mathbf{D}^k = (D^k_1, \ldots, D^k_n)$) ($k = 1, 2, \ldots, l$) is a covariant (contravariant) vector, and $\epsilon(\mathbf{s}) \geq 0$ and $z(\mathbf{s}) \geq 0$ are weighting functions.

2.1. Metric for Grid Clustering in the Regions of Sharp Variation of Physical Quantities

In the numerical simulation of factors affecting the behavior of aircraft and underwater vehicles, one needs computer codes for generating grids that condense in the regions where some characteristics of the physical medium (density, pressure, and the like) undergo sharp changes. In particular, such codes are required for the investigation of multiple interactions of shock and rarefaction waves or contact boundaries.

In power engineering and kinetics, an important factor for improving numerical results is grid refinement in the regions where some physical characteristics of combustion reactions in boilers of thermoelectric power stations and in chambers of aviation engines and space rockets undergo sharp changes; it is also important for taking into account kinetic reactions in discontinuity regions.

In the problem of determining the pollution of the environment by the remains of rocket propellants leaving tanks of falling rocket stages, it is important to find the distribution of the fuel that reached the ground in drop form and to describe the evolution of the cloud consisting of the propellant vapor formed when the falling drops evaporate in the atmosphere. In the second case, the wind drift of the cloud parts and its smearing due to the atmospheric turbulent diffusion plays an important role. This problem is three-dimensional, and a substantial factor is the moving source of drops (the falling rocket stage) and the smearing source of the vapor formed by the falling ensemble of evaporating drops. The use of moving adapted grids that condense in the regions where the drops and propellant vapor are formed makes it possible to considerably improve the efficiency of numerical algorithms in solving this class of problems.

Three-dimensional grids that condense in boundary layers are also important for studying global atmospheric phenomena and, in particular, for precise weather prediction based on numerical calculations.

To generate grids with node clustering in the regions of large gradients of a function $\mathbf{f}(\mathbf{s}) = (f^1(\mathbf{s}), \ldots, f^l(\mathbf{s}))$, monitor metric (2.1) is defined by (see [1, 11]

$$g^{\mathbf{s}}_{ij} = \epsilon(\mathbf{s})g^{xs}_{ij} + \frac{\partial f^k}{\partial s^i}\frac{\partial f^k}{\partial s^j}, \quad i, j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, l. \tag{2.3}$$

The efficiency of metric (2.3) for $\epsilon(\mathbf{s}) = \text{const}$ in the numerical solution of gas dynamics problems in which grids refining in the regions of shock waves and boundary layers are required was demonstrated in [12, 14].

2.2. Metric for Node Clustering in the Regions of Large or Small Function Values

To refine a grid in the regions where the interpolation error of the function $u(\mathbf{s})$ and the equations' approximation error are large, the monitor metric is defined in form (2.1) (see [17]) with

$$B^k_i = \frac{\partial^2 u}{\partial s^k \partial s^i}, \quad i, k = n.$$

In prediction of natural calamities (in particular, in tsunami prediction), the efficiency of numerical calculations substantially depends on the adaptation of the grid to the shape of a basin, its depth, and its shoreline; in particular, the grid must be refined in the regions of small depth. To generate grids for the physical geometry S^{xn} with node clustering in the regions of large values of a function $v(\mathbf{s})$ (for example, error solution or the reciprocal of the basin depth), the contravariant components of the monitor metric are defined by (see [12])

$$g_{\mathbf{s}}^{ij} = \Psi[v](\mathbf{s})g_{sx}^{ij}, \quad i, j = 1, 2, \ldots, n, \tag{2.4}$$

where $\Psi[v] > 0$ is a positive operator such that $\Psi[v](\mathbf{s})$ takes large (small) values where $v(\mathbf{s})$ is small (large). A generalization of metric (2.4) is given in [1].

Metric (2.4) can also be used for generating grids refining in the regions where the gradient of a function $\mathbf{f}(\mathbf{s})$ is large; for this purpose, the operator $A[\mathrm{grad}\mathbf{f}]$ is defined such that $A[\mathrm{grad}\mathbf{f}](\mathbf{s})$ takes small values where $|\mathrm{grad}\mathbf{f}|(\mathbf{s})$ is large and vice versa. In this case, the monitor metric has the form

$$g_{\mathbf{s}}^{ij} = A[\mathrm{grad}\mathbf{f}](\mathbf{s})g_{sx}^{ij}, \quad i, j = 1, 2, \ldots, n.$$

This metric was used for generating adaptive grids in two-dimensional domains in problems concerning axially symmetric gas flows [12].

2.3. Metric for the Generation of Field-Aligned Grids

The contravariant elements of the monitor metric in form (2.2) are also useful for generating grids in which the angle between the given vector field $\mathbf{B} = (B^1, \ldots, B^n)$ and the normal to a grid hypersurface $\xi^i = \mathrm{const}$ is close to $\pi/2$. In this case, the contravariant elements of the monitor metric have the form (see [1])

$$g_{\mathbf{s}}^{ij} = \epsilon(\mathbf{s})g_{sx}^{ij} + B_k^i B_k^j, \quad i, j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, l, \tag{2.5}$$

where $\epsilon(\mathbf{s})$ and $\mathbf{B}_k$ $(k = 2, 3, \ldots, l)$ are small and $\mathbf{B}_1 = \mathbf{B}$. The application of metric (2.5) for generating magnetic field-aligned grids used in plasma numerical calculations was demonstrated in [18].

2.4. Metric for Generating Balanced Grids

To generate balanced adaptive grids that are simultaneously field-aligned and adapt to the values of one function and to the gradient of another, the contravariant elements $g_{\mathbf{s}}^{ij}$ of the monitor metric can be defined in the form of a linear combination of the corresponding contravariant elements g_{al}^{ij}, g_{adg}^{ij}, and g_{adv}^{ij} and basis metrics (2.5), (2.3), and (2.4); therefore, they can be defined by

$$g_{\mathbf{s}}^{ij} = \epsilon_1(\mathbf{s})g_{al}^{ij} + \epsilon_2(\mathbf{s})g_{adg}^{ij} + \epsilon_3(\mathbf{s})g_{adv}^{ij}, \quad i, j = 1, 2, \ldots, n, \tag{2.6}$$

where $\epsilon_i(\mathbf{s}) \geq 0$ $(i = 1, 2, 3)$ are weighting functions satisfying the condition $\Sigma_{i=1}^{3}\epsilon_i(\mathbf{s}) > 0$.

The covariant elements of the balanced metric are defined similarly (see [1]).

2.5. Application of Layer-Type Functions for Defining the Monitor Metric

The definitions of monitor metrics (2.3), (2.4), and (2.5) include weighting and control functions to adapt to the gradients of physical quantities, to align the grid lines with a vector field, and to balance the grid (2.6). These functions either define the metric itself or regulate the contribution of basis metrics in the final result. The set of functions that can be used for such purposes can be specified using three basis layer-type functions $\varphi_i(x, \varepsilon)$ $(i = 1, 2, 3; x \geq 0; 0 < \varepsilon \ll 1)$. These functions describe the qualitative behavior of solutions of singularly perturbed problems that have narrow zones (layers) in which the gradient of the solution undergoes sharp changes (see [3]). The basis layer-type functions are as follows:

$$\begin{gathered}\varphi_1(x, \varepsilon) = \exp(-b|x - x_0|/\varepsilon^k), \\ k > 0, \quad b > 0, \quad 0 \leq x \leq 1, \quad 0 < \varepsilon \ll 1,\end{gathered} \tag{2.7}$$

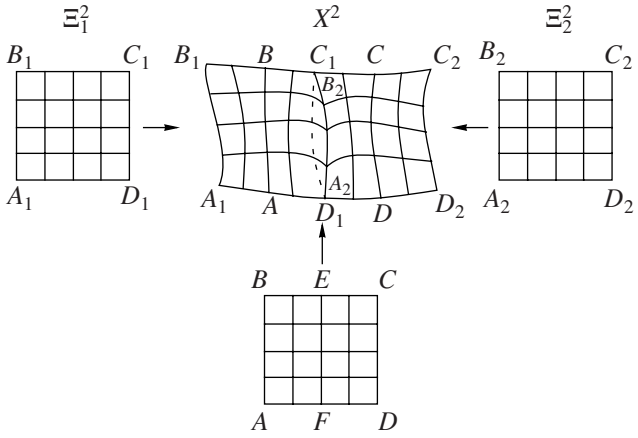

Fig. 4. The scheme of smooth grid generation by solving grid equations.

is the exponential layer-type function;

$$\varphi_2(x,\varepsilon) = \frac{\varepsilon^{kb}}{(\varepsilon^k + |x - x_0|)^b}, \quad k > 0, \quad b > 0, \quad 0 \le x \le 1, \quad 0 < \varepsilon \ll 1, \tag{2.8}$$

and is the power layer-type function;

$$\varphi_3(x,\varepsilon) = \frac{\ln(1 + |x - x_0|\varepsilon^{-k})}{\ln(1 + \varepsilon^{-k})}, \quad k > 0, \quad 0 \le x \le 1, \quad 0 < \varepsilon \ll 1, \tag{2.9}$$

is the logarithmic layer-type function.

The parameter ε determines the steepness of the function near the point $x = x_0$. k and b are positive constants used to control some characteristics of the functions and layers. In particular, k affects the scale of the layer, and b affects the structure of the layer and the layer width.

If ε is small, basis layer functions (2.7)–(2.9) change sharply in a very narrow region. Therefore, they can be used to specify weighting functions for localizing the contribution of the corresponding basis metrics into the total monitor metric (2.6) for generating balanced grids. Examples of using layer-type functions for generating balanced grids are discussed in Section 4.6.

3. GENERATION OF SMOOTH BLOCK GRIDS

In Section 1, we described the technology for generating single-block grids. However, it is often required to use multiblock grids. In this case, there are difficulties in the smooth matching of grid lines of adjacent blocks. In this section, we present two new iterative approaches to generating block grids in which the grid lines match smoothly at the boundary of the adjacent blocks. These approaches can also be used for generating smooth grids of the *O*- and *C*-types in the vicinity of fictitious edges and faces. The main feature of these methods is that both the boundary segments of the adjacent blocks and the distribution of the grid nodes in these segments are determined in the course of the iterations.

In each block, the local grid is determined by solving the Dirichlet problem for inverted equations (1.6) with respect to the monitor metric. The location of the boundary segments in the adjacent blocks and the distribution of the grid nodes in these segments are determined by the numerical solution of grid equations and by the interpolation by the nodes belonging to the grid hypersurfaces near the common boundary segments.

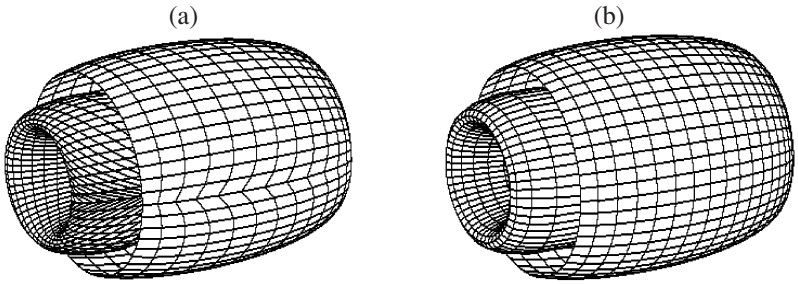

Fig. 5.

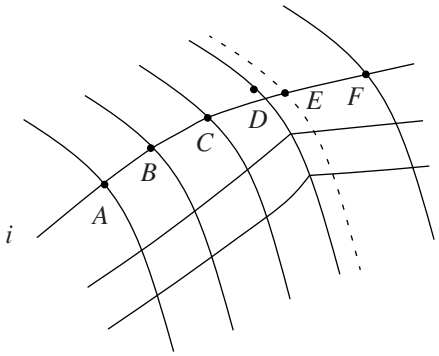

Fig. 6.

3.1. Smoothing Using Grid Equations

The idea of this approach is illustrated in Fig. 4, where the scheme of generating a smooth grid with quadrilateral cells in two blocks of the two-dimensional domain X^2 is illustrated. The left-hand block is bounded by the curves A_1B_1, B_1C_1, C_1D_1, and D_1A_1. The right-hand block is bounded by the curves A_2B_2, B_2C_2, C_2D_2, and D_2A_2. The curves C_1D_1 and B_2A_2 coincide and are common for both blocks. First, the grid in X^2 is generated independently in each block using the numerical solution of the Beltrami or diffusion equations at the nodes of the corresponding domains Ξ_1^2 and Ξ_2^2 with the identical distribution of nodes on the common boundary in X^2. Then, the third (additional) block $ABCD$ is introduced that contains the joining line EF (the point E coincides with the points $C1$ and $B2$, and the point F, with $D1$ and $A2$), and, between every two (or every several) iterations performed for the main blocks, the same equations are solved for the new block $ABCD$. As a result, the grid points on the line EF will be mapped (depending on the adjacent nodes and grid equations) to the dashed line that is the new boundary between the blocks. Next, the nodes inside the new blocks are found, and so on. The process is continued for the new blocks until convergence is achieved. Similarly, smooth block structured triangular and spatial grids are generated.

The main advantage of this approach is the use of the same grid equations for finding new nodes both on the joining line and inside the blocks. As a result, the process of the joining line recalculation is fairly simple, and the conditions imposed on the grid by the monitor metric remain valid.

Figure 5 shows a nonsmooth (a) and a smooth (b) grid on a surface generated using the proposed algorithm.

3.2. Smoothing by Interpolation

In the framework of this approach, a node on the ith grid line of the new common boundary (the dashed line in Fig. 6) is found using the following procedure. Upon determining the grid points inside both blocks that have a common boundary with the node D on it, we choose the ith grid line the two nearest nodes to D inside one block (the points B and C in Fig. 6) and the one nearest point in the other block (the point D in Fig. 6). Then, we draw a smooth curve (for example, an arc of a circle) through these points. The new common boundary point E of the ith level belongs to this arc. Its location on the arc is determined by the equation $\overline{CE}/\overline{EF} = \overline{AB}/\overline{BC}$, where A is the third grid point of the first block on the ith level (the bar over a segment

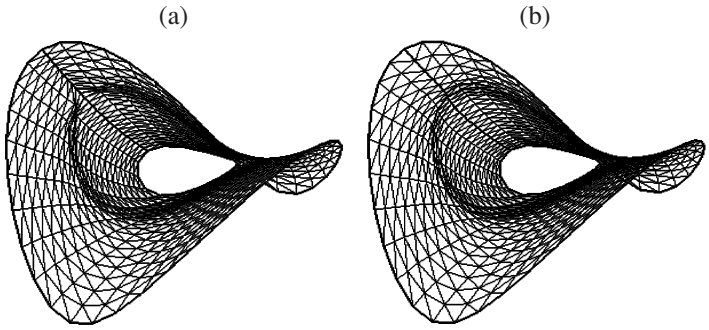

Fig. 7.

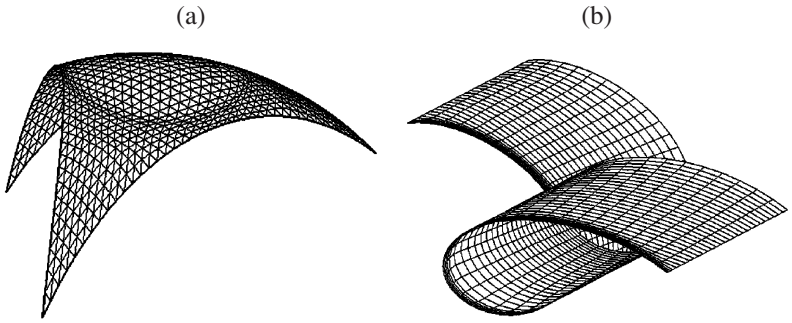

Fig. 8. Examples of triangular and quadrilateral grids generated using metric (2.3) that ensure grid clustering in the neighborhood of certain specified curves.

denotes the distance between its endpoints). When all the grid points on the new common boundary are found, the grid points inside both new blocks are found independently by solving grid equations. The process is continued until a certain convergence condition is satisfied.

Figure 7 shows nonsmooth (a) and smooth (b) triangular *O*-type grids on a surface generated using the proposed approach.

4. NUMERICAL RESULTS

In this section, we present examples of grids generated by the numerical solution of the inverted Beltrami and diffusion equations with respect to the monitor metric.

4.1. Generation of Adaptive Grids

To generate a grid refining in the neighborhood of the hypersurface determined by the equation $\phi(\mathbf{s}) = 0$, monitor metric (2.3) or (2.4) can be used. The monitor function can be specified using the basis layer-type functions $\varphi_i(x, \varepsilon)$ ($i = 1, 2, 3$), see (2.7)–(2.9).

If metric (2.3) is used to cluster the grid nodes near the hypersurface $\phi(\mathbf{s}) = 0$, the monitor function $f(\mathbf{s})$ must be chosen such that it changes sharply near this hypersurface. For example, we can use the function $f(\mathbf{s}) = \tanh[\phi(\mathbf{s})/\delta]$, where $0 < \delta \ll 1$, which includes the layer-type mapping $\varphi_1(x, \delta)$.

Figure 8a shows the grid generated on the monitor surface

$$v(\mathbf{s}) = 1.5\sqrt{1 - 2(s_1 - 0.5)^2 - (s_2 - 0.28)^2},$$

defined over the domain with the boundaries

$$s^2 = -0.4(s^1 - 0.5)^2 + 0.1, \quad 0 \le s^1 \le 1,$$

$$s^2 = \sqrt{3}s^1, \quad 0 \le s^1 \le 0.5,$$

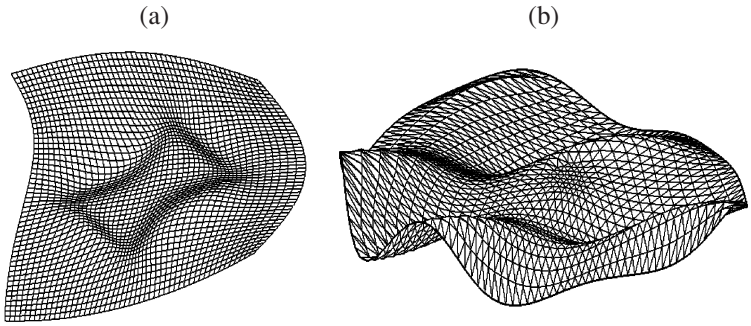

Fig. 9. Adaptive quadrilateral (a) and triangular (b) grids in a domain and on a surface generated using metric (2.4).

$$s^2 = \sqrt{3}(1 - s^1), \quad 0.5 \le s^1 \le 1.$$

This grid is refined in the neighborhood of the curve $\varphi_1(\mathbf{s}) = 0$, where

$$\varphi_1(\mathbf{s}) = 20((s^1 - 0.5)^2 + (s^2 - 0.33)^2 - 0.07).$$

Figure 8b shows the grid generated on the monitor surface

$$x^1(\mathbf{s}) = s^1 - 0.5\cos(8s^1),$$

$$x^2(\mathbf{s}) = s^2 + 0.5\sin(8s^1),$$

$$x^3(\mathbf{s}) = (s^1)^2,$$

defined over the domain with the boundaries

$$0 \le s^1 \le 1, \quad (s^1 - 0.5)^2 - 0.25 \le s^2 \le (s^1 - 0.5)^2 + 0.75.$$

This grid is refined in the neighborhood of the curve $\varphi_2(\mathbf{s}) = 0$, where

$$\varphi_2(\mathbf{s}) = (s^1 - 0.5)^2 - s^2 - 0.25.$$

The function $f(\mathbf{s})$ for generating the grids shown in Fig. 8 was specified by the equation

$$f(\mathbf{s}) = \begin{cases} 0.4\tanh[\varphi_1(\mathbf{s})/0.5] & (\text{Fig. 8a}), \\ 0.17\tanh[\varphi_2(\mathbf{s})/0.5] & (\text{Fig. 8b}). \end{cases}$$

Figure 9a depicts the adaptive grid generated in the domain with the boundaries

$$s^1 = 0.2\sin(\pi t^2) - 0.9, \quad s^2 = 2t - 0.9,$$

$$s^1 = 0.4\sin(\pi t) + 1.1, \quad s^2 = 1.4t - 0.3,$$

$$s^1 = 2t - 0.9, \quad s^2 = 0.6t^2 - 0.9,$$

$$s^1 = 2t - 0.9, \quad s^2 = 0.2\sin(\pi t) + 1.1,$$

where $0 \le t \le 1$. The grid points cluster in the neighborhood of the curve $\phi(\mathbf{s}) = 0$, where

$$\phi(\mathbf{s}) = (s^2)^2(s^1 - 0.5)^2 + (s^1)^2(s^2 - 0.4)^2 - 0.02.$$

Let us write (2.4) in the form

$$g_{\mathbf{s}}^{ij} = \omega[z(\phi)]g_{sx}^{ij}, \quad i, j = 1, 2, \ldots, n,$$

where $\phi(\mathbf{s}) = 0$ is the hypersurface near which the grid should be refined.

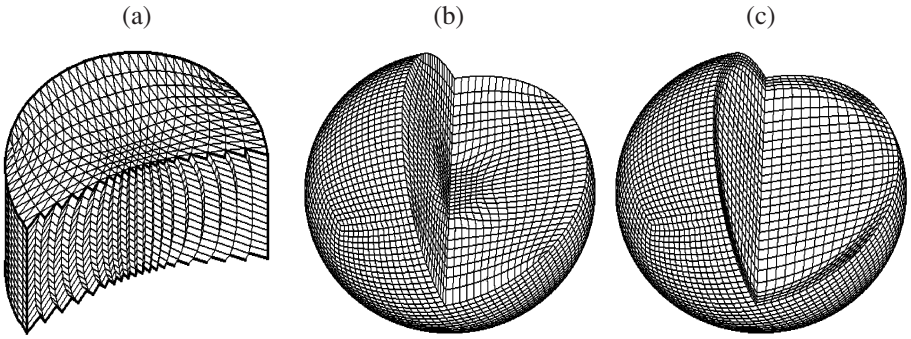

Fig. 10.

The monitor functions are defined as follows:

$$\omega[z] = 0.003\exp(-\exp(-z/0.5)/0.5),$$

$$z(\phi) = \tanh[(\phi(\mathbf{s})/0.03].$$

Thus, the grid in Fig. 9a is refined near the hypersurface $\phi(\mathbf{s}) = 0$; near this surface, the gradient of the function $z(\phi)$ is large.

Figure 9b depicts the surface

$$v(\mathbf{s}) = 0.01(\sin(15s^1)\sin(5s^2)),$$

defined over the domain bounded by the curve

$$(s^1 - 0.5)^2 + (s^2 - 0.5)^2 = 0.25.$$

The functions $\phi(\mathbf{s})$ and $\omega[z(\phi)]$ were defined by

$$\omega[z(\phi)] = 0.5\exp\left[-1.5\exp\left(-\frac{[\phi(\mathbf{s})]^2}{0.01}\right)\right],$$

$$\phi(\mathbf{s}) = (s^1 - 0.475)^2/0.03 + (s^2 - 0.45)^2/0.07.$$

The resulting grid clusters in the neighborhood of the point (0.475, 0.45).

A prismatic grid in a cylinder is shown in Fig. 10a. Figures 10b and 10c use cross sections to demonstrate examples of adaptive hexahedral grids in a ball. The monitor function was defined as

$$f(\mathbf{s}) = \begin{cases} 2\exp[-\exp(-R^2/0.001)/0.5] & \text{(Fig. 10b)}, \\ 2\exp[-\exp(-(R^2 - 0.75)/0.001)/0.5] & \text{(Fig. 10c)}, \end{cases}$$

where $R^2 = (s^1 - 0.5)^2 + (s^2 - 0.5)^2$.

4.2. Application of the Method for Calculating the Propagation of Passive Impurities in the Atmosphere

The proposed grid generation method can be used to construct efficient algorithms for modeling the evolution of a cloud of a passive impurity produced by a moving source in the atmosphere. For example, such a source can be a flying object (aircraft, rocket, a falling rocket stage, and the like). Such a source ejects into the atmosphere either light particles of a passive impurity or liquid or gaseous matter; while falling, this matter evaporates and forms a cloud of the impurity vapor. A specific feature of this problem is the presence of a moving localized source (aircraft, rocket, or falling rocket stage) and a moving distributed source (falling vaporizing drops). This feature stipulates the use of moving adaptive grids because a fixed spatial grid does not make it possible to adequately describe the corresponding physical processes.

The problem of the Earth pollution by the remains of rocket fuel (unsymmetrical dimethylhydrazine) flowing out of the tanks of a spent rocket stage was considered in [19]. When flowing out, the fuel disintegrates into drops, which then undergo wind drift, and their mass decreases due to evaporation. As the mass

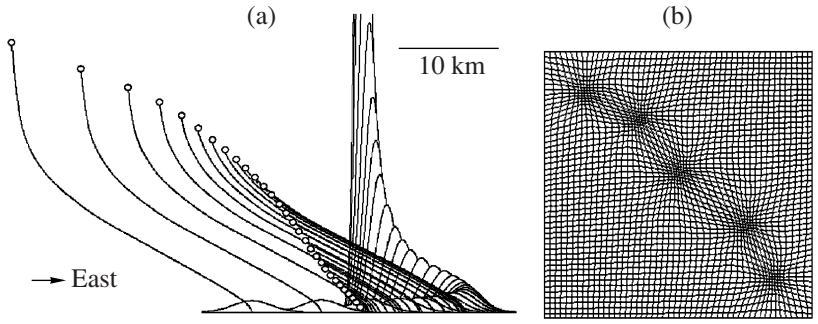

Fig. 11.

of the drops decreases, the vertical rate of fall decreases. Figure 11a shows the falling trajectories of individual drops as they are ejected one-by-one from the tank of a falling rocket stage (open dots). Only the trajectories of the drops with the same initial diameter of 3 mm are depicted. The corresponding Gaussian distributions of the surface density of the fuel that reached the ground in drop form are shown in the same figure. Such a distribution corresponds to the situation when the turbulent atmospheric diffusion is taken into account, which smears the ensemble of falling drops. A large series of calculations for falling second stages of the Proton carrier rockets launched from the Baikonur launching site was performed. Real-life rocket launches (90 launches) were considered in which the second stages fall in three regions of the Altai. The meteorological data concerning these launches were provided by the hydrometeorological service of Russia. Since 90% of the fuel evaporates when the drops are falling, it is of interest to investigate the evolution of the fuel vapor cloud. The propagation of the impurity vapors is determined by the wind drift and the atmospheric turbulent diffusion. The corresponding dynamics of the impurity vapor density are governed by the transport–diffusion equation (see [3])

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial x^i}\left(D\frac{\partial \rho}{\partial x^i}\right) - \frac{\partial}{\partial x^i}(\rho u^i) + S, \quad i = 1, 2, 3,$$

where D is the impurity diffusion coefficient in the atmosphere, S is the density of the impurity source, u^i are the components of the local velocity, and ρ is the impurity density. In the framework of the approach used in this paper, the variables are changed for the new variables ξ^1, ξ^2, ξ^3 in which the transport–diffusion equation has the form (see [3])

$$\frac{\partial}{\partial t}(J\rho) = -\frac{\partial}{\partial \xi^j}\left(J\rho(\bar{u}^j - \bar{w}^j) - Jg^{kj}D\frac{\partial \rho}{\partial \xi^k}\right) + JS, \quad j, k = 1, 2, 3,$$

where

$$\bar{u}^j = u^k\frac{\partial \xi^j}{\partial x^k}, \quad \bar{w}^j = \frac{\partial \xi^i}{\partial t}\frac{\partial \xi^j}{\partial x^i}.$$

This equation is solved using the finite difference method simultaneously with the parabolized grid equations (1.11) in the variables ξ^1, ξ^2, ξ^3. The monitor metric in form (2.4), which depends on the vapor density, yields the desired grid refinement in the regions with high impurity density (see Fig. 11b).

4.3. Application of the Method for Solving Singularly Perturbed Diffusion Problems

Monitor metric (2.4) is effective for the numerical solution of problems with boundary and interior layers. By way of example, Figs. 12b and 13b show the results of solving the singularly perturbed diffusion problem

$$\begin{aligned} -\varepsilon\Delta u + \mathbf{a}(\mathbf{x})\cdot\nabla u + f(\mathbf{x}, u) &= 0, \quad \mathbf{x} \in G, \\ u(\mathbf{x}) &= u_0(\mathbf{x}), \quad \mathbf{x} \in \partial G, \end{aligned} \tag{4.1}$$

where $G \subset \mathbb{R}^2$, $\partial G = \bar{G}\backslash G$, $\mathbf{x} = (x^1, x^2)$, and $\mathbf{a}(\mathbf{x}) = (a_1(x^1), a_2(x^2))$.

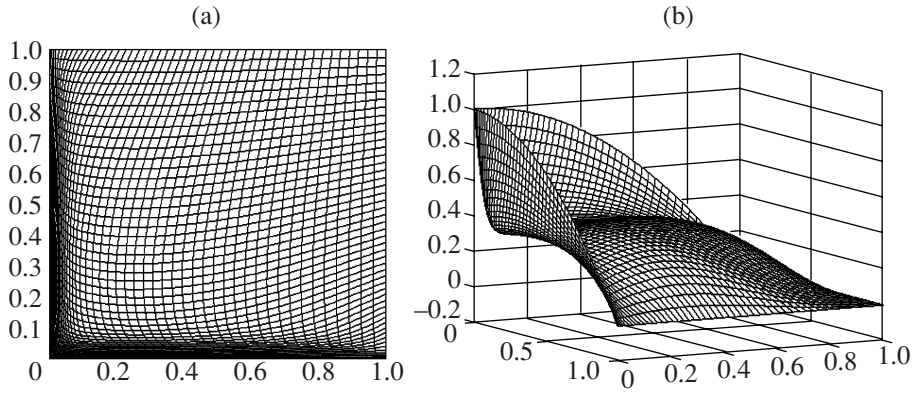

Fig. 12. Application of metric (2.3) for the generation of an adaptive quadrilateral grid (a) and for solving two-dimensional singularly perturbed problem (4.1) (b).

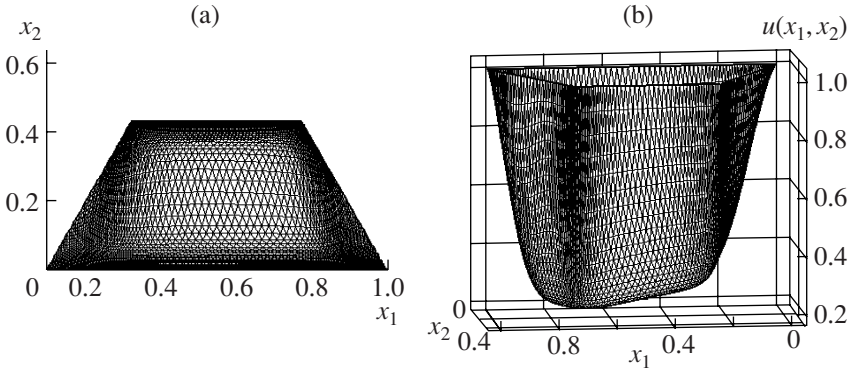

Fig. 13. Application of metric (2.3) for the generation of an adaptive triangular grid (a) and for solving two-dimensional singularly perturbed problem (4.1) (b).

Figure 12 shows the grid (Fig. 12a) used to solve problem (4.1) and its results (Fig. 12b) with the parameters

$$\varepsilon = 0.01, \quad a_1(x^1) = -\sqrt{x^1}, \quad a_2(x^2) = -\sqrt{x^2}, \quad f(\mathbf{x}, u) = 4u - \sin(5x^1x^2) + 1,$$

$$u_0(\mathbf{x}) = 0, \quad \text{if } x^1 = 1 \quad \text{or} \quad x^2 = 1,$$

$$u_0(\mathbf{x}) = 1 - (x^1)^2, \quad \text{if} \quad x^2 = 0,$$

$$u_0(\mathbf{x}) = 1 - (x^2)^2, \quad \text{if } x^1 = 0.$$

The grid shown in Fig. 13a was used to solve problem (4.1) with the parameters

$$\varepsilon = 0.005, \quad \mathbf{a}(\mathbf{x}) \equiv \mathbf{0}, \quad f(\mathbf{x}, u) = \sqrt{x_1}(\sin(x^1) + 3)u - \cos(x^1), \quad u_0(\mathbf{x}) = 1.$$

4.4. Field-Aligned Grids

Figure 14 shows the integral lines of a vector field **B** (Fig. 14a) in the two-dimensional domain S^2 and the corresponding grid (Fig. 14b). This grid was generated using monitor metric (2.5) with $k = 1$. The field **B** is defined using the model function $\psi(\mathbf{s})$ as

$$\mathbf{B} = \left(-\frac{\partial \psi}{\partial s^2}, \frac{\partial \psi}{\partial s^1}\right). \tag{4.2}$$

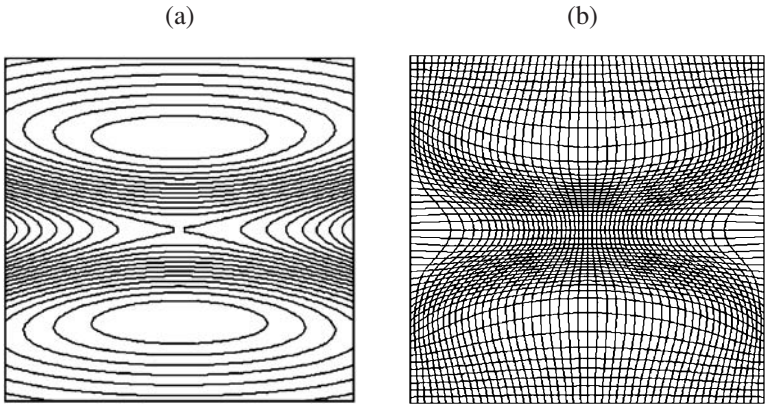

Fig. 14.

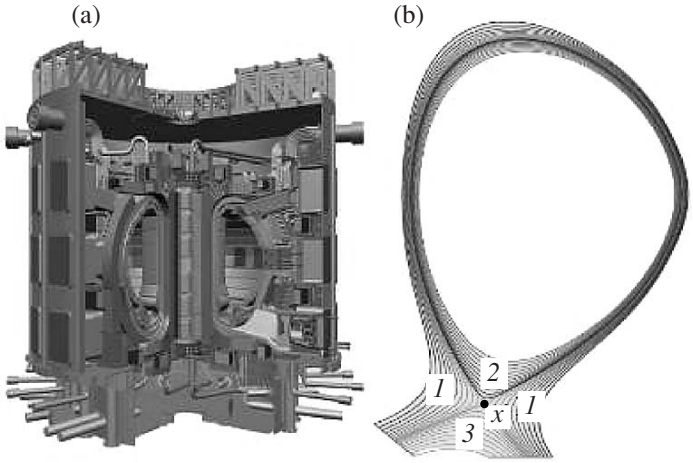

Fig. 15.

This vector field satisfies the natural (for vector fields) condition $\operatorname{div}\mathbf{B} = 0$. The grid was obtained using the following expressions for $\epsilon(\mathbf{s})$ and $\psi(\mathbf{s})$:

$$\epsilon(\mathbf{s}) = 0.05\left(\frac{0.3}{0.3 + |\mathbf{B}|^2}\right)^2,$$

$$\psi(\mathbf{s}) = \phi(s^2)(1 - \phi(s^2))[(s^1 - 0.5)^2 - 6(\phi(s^2) - 0.5)^2],$$

$$\phi(s^2) = 0.5\left[1 + \tanh\left(\frac{s^2 - 0.5}{0.2}\right)\right].$$

4.5. Grid in a Tokamak Edge Region

Generation of magnetic field-aligned grids in a tokamak chamber is important in plasma physics; in particular, it is important in the research aimed at creating a nuclear fusion reactor [20, 21]. To understand the processes occurring in a tokamak, one needs highly accurate numerical models. The application of adaptive grids considerably improves the accuracy and efficiency of such models.

Figure 15a depicts the tokamak design, and Fig. 15b shows the magnetic field lines in an axially symmetric cross section of the tokamak. Such a field has a singular point x at which it degenerates. At this point, the magneticline called the separatrix has a self-intersection. The topology of the domain under consider-

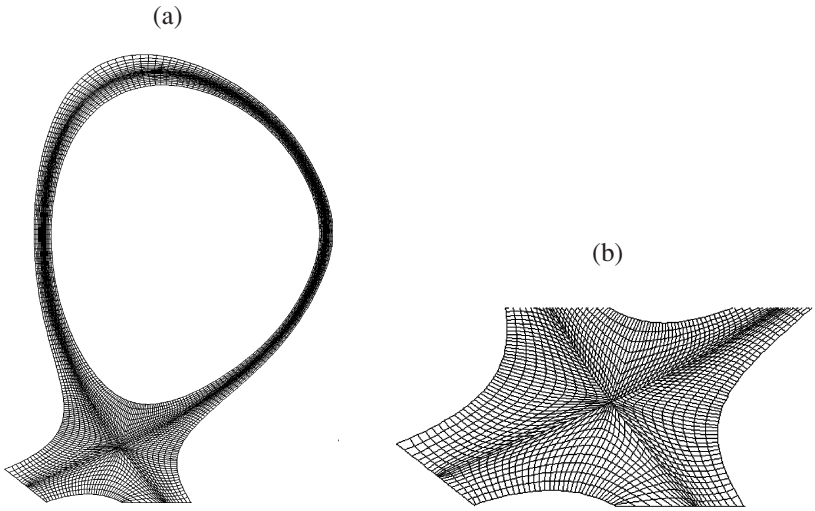

Fig. 16.

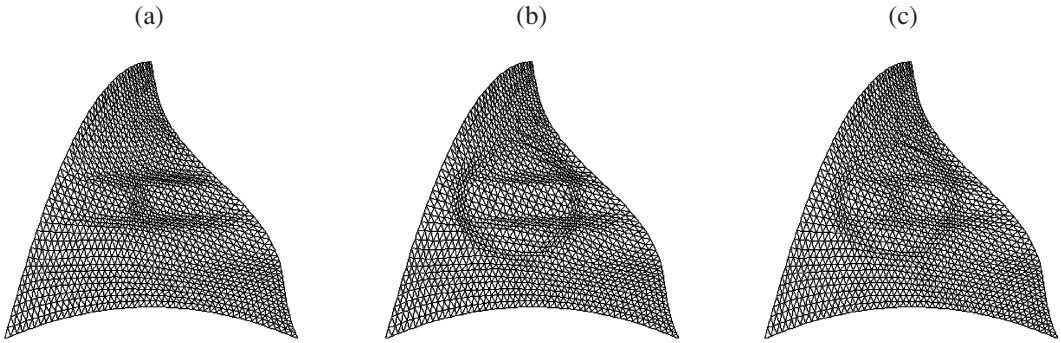

Fig. 17.

ation is defined by three blocks (see Fig. 15b). The magnetic lines in the second blocks are self-intersecting; therefore, an *O*-type grid should be constructed in this block.

Figure 16a shows the grid generated in the tokamak edge region using metric (2.4), which enables one to refine the grid in the neighborhood of the point x. Figure 16b shows a fragment of this grid. The grid nodes on the separatrix were clustered in the neighborhood of x by solving one-dimensional diffusion equations.

4.6. Generation of Balanced Grids

To generate balanced field-aligned grids that adapt to the values of one function and to the derivatives of another, formula (2.6) for the contravariant metric components of the monitor metric was used. This formula was written in the form

$$g^{ij}(\mathbf{s}) = (1-\alpha)g^{ij}_{al} + \alpha((1-\beta)g^{ij}_{adg} + \beta g^{ij}_{adv}), \quad i, j = 1, 2, \ldots, n. \tag{4.3}$$

Examples of balanced two-dimensional grids in the domain S^2 are shown in Fig. 17. In Fig. 17a, a grid aligned with the vector field $\mathbf{B} = \left(-\frac{\partial\psi}{\partial s^2}, \frac{\partial\psi}{\partial s^1}\right)$ (formula (4.2)) and adapted to the values of the function $\varphi_1(\mathbf{s})$ is shown. The grid depicted in Fig. 17b is aligned with the same vector field and adapted to the gradient of the function $f_2[\varphi_2(\mathbf{s})]$. The grid in Fig. 17c is aligned with the same field and is adapted to the values of one function and to the gradient of the other. These grids were generated using metric (4.3). The functions

appearing in this metric for generating the grids in Fig. 17 were defined using the basis layer-type functions (2.7) in the form

$$f_1(\varphi_1) = \exp(-\varphi_1),$$

$$\varphi_1(\mathbf{s}) = 50\exp\left(-\frac{((s_1-0.45)^2+(s_2-0.5)^2)^2}{0.000001}\right),$$

$$f_2(\varphi_2) = 20\tanh\frac{\varphi_2}{0.025},$$

$$\varphi_2(\mathbf{s}) = (s_1-0.45)^2+(s_2-0.5)^2-0.04,$$

$$\psi(\mathbf{s}) = v(1-v)[(s^1-0.5)^2+1.5(v-0.5)^2],$$

$$\text{where}\quad v = 0.5\left[1+\tanh\frac{s^2-0.5}{0.2}\right].$$

The coefficients α and β in (4.3) were as follows:

$$\alpha = \varepsilon(\mathbf{s}) = 0.1\exp(-|\mathbf{B}|^2/0.04),\quad \beta = 1\quad \text{(Fig. 17a)},$$

$$\alpha = \varepsilon(\mathbf{s}) = 0.9\exp(-|\mathbf{B}|^2/0.03),\quad \beta = 0\quad \text{(Fig. 17b)},$$

$$\alpha = \varepsilon(\mathbf{s}) = 0.9\exp(-|\mathbf{B}|^2/0.03),\quad \beta = 0.5\quad \text{(Fig. 17c)}.$$

ACKNOWLEDGMENTS

This work was supported by the Russian Foundation for Basic Research, project no. 07-01-00336, and by the Complex Integration Project of the Siberian Branch of the Russian Academy of Sciences, project no. 1.8.

REFERENCES

1. V. D. Liseikin, *A Computational Differential Geometry Approach to Grid Generation* (Springer, Berlin, 2007).
2. Yu. I. Shokin, V. D. Liseikin, A. S. Lebedev, et al., *Riemanian Geometry Techniques in Grid Generation* (Nauka, Novosibirsk, 2005) [in Russian].
3. V. D. Liseikin, *Grid Generation Methods* (Springer, Berlin, 1999).
4. S. K. Godunov and G. P. Prokopov, "On Calculation of Conformal Mappings and Grid Generation," Zh. Vychisl. Mat. Mat. Fiz. **7** (5), 1031 (1967).
5. J. F. Thompson, B. K. Soni, and J. Hauser, "A Reflection on Grid Generation in the 90s: Trends, Needs Influences," in *Numerical Grid Generation in Computational Field Generation,* Ed. by B. K. Soni, J. F. Thompson, J. Hauser, and P. R. Eiseman (ERC Press, Mississippi State University, Starkville, 1996), no. 1, p. 1029.
6. H. J. Raveche, D. H. Lawrie, and A. M. Despain, *A National Computing Initiative* (SIAM, Philadelphia, 1987).
7. P. Knupp and S. Steinberg, *Fundamentals of Grid Generation* (CRC, Boca Raton, Florida, 1993).
8. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation. Foundations and Applications* (Elsevier Science, New York, 1985).
9. Z. U. A. Warsi, "Tensors and Differential Geometry Applied to Analytical and Numerical Coordinate Generation," MSSU-EIRS-81-1 (Mississippi State University, Starkville, 1996).
10. N. N. Yanenko, *The Split-Step Method in Multidimensional Problems of Mathematical Physics* (Nauka, Novosibirsk, 1967) [in Russian].
11. V. D. Liseikin, "On the Generation of Regular Grids on *n*-Dimensional Surfaces," Zh. Vychisl. Mat. Mat. Fiz. **31** (11), 41 (1991).
12. N. T. Danaev, V. D. Liseikin, and N. N. Yanenko, "On the Numerical Computation of the Motion of Viscous Gas around a Bodies of Revolution Using A Moving Grid," in *Numerical Methods in Mechanics of Continua* (ITPM SO AN SSSR, Novosibirsk, 1980), vol. 11, no. 1, p. 51 [in Russian].
13. A. A. Charakch'yan and S. A. Ivanenko, "A Variational Form of the Winslow Grid Generator," J. Comput. Phys. **136**, 385 (1997).

14. B. N. Azarenok, "Variational Barrier Method of Adaptive Grid Generation in Hyperbolic Problems of Gas Dynamics," SIAM J. Numer. Analys. **40** (4), 651 (2002).
15. S. A. Ivanenko and A. A. Charakhch'yan, "Curvilinear Quadrilateral Grids," Zh. Vychisl. Mat. Mat. Fiz. **28** (4), 503 (1988).
16. G. Liao, "On Harmonic Maps," in *Mathematical Aspects of Numerical Grid Generation, Frontiers in Appl. Math.* (SIAM, Philadelphia, 1991), vol. 8, no. 123.
17. P. J. Frey and P. L. George, *Mesh Generation, Application to Finite Elements* (Hermes Sci., Paris, 2000).
18. A. G. Glasser, V. D. Liseikin, and I. A. Kitaeva, "Control of the Properties of Numerical Grids through Monitor Metrics," Zh. Vychisl. Mat. Mat. Fiz. **45** (8), 1466 (2005) [Comput. Math. Math. Phys. **45**, 1416–1432 (2005)].
19. E. G. Klimova, Yu. N. Morokov, G. S. Rivin, et al., "Mathematical Evaluation of Regions of Pollution by Rocket Fuel Caused by Falling Spent Parts of Carrier Rockets," Opt. Atmos. Okeana **18** (5-6), 525 (2005).
20. A. H. Glasser and X. Z. Tang, "The SEL Macroscopic Modeling Code," Comput. Phys. Commun. **164**, 237 (2004).
21. T. D. Rognlien, X. Q. Xu, and A. C. Hinmarsh, "Application of Parallel Implicit Methods to Edge-Plasma Numerical Simulations," J. Comput. Phys. **175**, 249 (2002).