RUSSIAN ACADEMY OF SCIENCE
COMPUTING CENTER

REPORT ON APPLIED MATHEMATICS

Sergey A. Ivanenko
Boris N. Azarenok

APPLICATION OF MOVING ADAPTIVE GRIDS FOR NUMERICAL SOLUTION OF 2-D NONSTATIONARY PROBLEMS IN GAS DYNAMICS

(to be submitted for publication)

MOSCOW 2000

Boris N. Azarenok & Sergey A. Ivanenko
Computing Center of RAS
Vavilov str. 40, GSP-1, Moscow, Russia
e-mail: azarenok@ccas.ru , sivan@ccas.ru

UDC 517.958:533.7

ABSTRACT

Solution-adaptive grid generation procedure is coupled with the Godunov-type solver of the second order accuracy. Dynamically adaptive grids, clustered to singularities, allow to increase the accuracy of numerical solution. The theory of harmonic maps is used as a theoretical framework for grid generation. The problem of constructing harmonic coordinates on the surface of the graph of control function is formulated. The projection of these coordinates onto a physical domain produces an adaptive-harmonic structured grid. A variational grid generator which can be used also in the case of unstructured meshes with adaptation to a vector-function is described in detail. The discrete functional has an infinite barrier on the boundary of the set of grids with all convex cells and this guarantees unfolded (nondegenerate) grid generation at every time step. Results of test computations are presented.

Key words. High-order scheme, adaptive grid, variational barrier method, harmonic mapping, folding-free grid generator

---

The present work was performed under support of Russian Fund of Fundamental Research (project code 99-01-00264).

Introduction

The main purpose of the present work is to investigate possibility to use the method of adaptive-harmonic grid generation, which has shown essential advantages in the stationary problems [17],[2], to the nonstationary problems of gas dynamics.

It is well known that grid adjustment (adaptation) to singularities of the numerical solution allows to increase greatly the accuracy of computations [1]. However, there are some additional numerical problems here. It is necessary to obtain a correct resolution of singularities and suppress nonphysical spurious oscillations. In the present work the method of adaptive grid generation, based on the theory of harmonic maps, ensuring generation of unfolded grids at every time step is presented. Method is variational, i.e. we consider the problem of minimizing a finite-difference function approximating the Dirichlet's functional written for surfaces. The discrete functional has an infinite barrier on the boundary of the set of grids with all convex cells and this guarantees nondegenerate (unfolded) grid generation during computations. This folding-resistant property is very important in nonstationary problems since if at least one cell becomes folded we have to stop calculation and use special procedures to continue modeling.

Although in the report we present results obtained on structured grids, the method can be extended to unstructured meshes as well.

Moving grid techniques have been developed in a number of works [4, 3, 5, 6, 7]. All these methods, in contrast to presented in this report, do not ensure the generated meshes to be unfolded at discrete level.

To execute calculations on the moving curvilinear grids a $GLFC$ scheme (Godunov Linear Fluxes Correction) is used, being a modification of the Godunov scheme with fluxes correction and time splitting, that provides the second-order accuracy in time and space to the smooth solutions [8]. In order to suppress nonphysical spurious oscillations in the vicinity of the discontinuities a monotonicity algorithm is applied.

1. Flow solver

In this section we shall describe the $GLFC$ scheme used as a flow solver [8].

We write the variational formulation for the hyperbolic system of gas dynamics equations in integral form of conservation laws

$$\oiint_A \boldsymbol{\sigma} dxdy + \boldsymbol{a} dydt + \boldsymbol{b} dtdx = \mathbf{o} \,, \tag{1}$$

where

$$\boldsymbol{\sigma} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \boldsymbol{a} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{bmatrix},$$

here u and v are the velocity components, p and ρ are the pressure and density. Total energy $E=\rho[e+0.5(u^2+v^2)]$, here e is the specific internal energy. Equation of state is $p=(\gamma-1)\rho e$, where γ is the ratio of specific heats. Denote the vector of unknown functions as $\boldsymbol{f} = (u, v, p, \rho)^T$.

Let us introduce the curvilinear moving grid in space $x-y-t$, see Fig.1a. Bottom face of the computing cell (or control volume) is taken at time level n and the top face at level $n+1$.

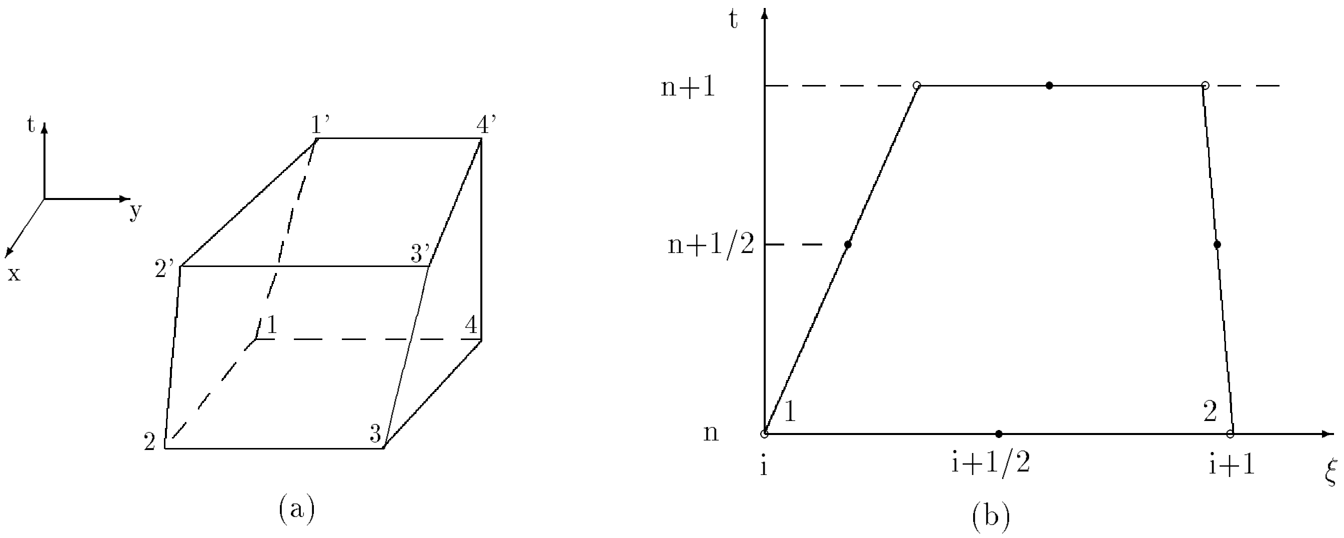

Fig.1. Computational cell

Integrating the system (1) over the oriented surface being the boundary of the control volume we get a cell-centered finite-volume discretization of the governing equations:

$$\boldsymbol{\sigma}^{n+1}A^{n+1} - \boldsymbol{\sigma}^{n}A^{n} + \boldsymbol{Q}_{11'4'4} + \boldsymbol{Q}_{233'2'} + \boldsymbol{Q}_{122'1'} + \boldsymbol{Q}_{433'4'} = \mathbf{o}\ , \tag{2}$$

where $\boldsymbol{\sigma}^{n+1}$ and $\boldsymbol{\sigma}^{n}$ are the average values at time t^{n+1} and t^{n} in the center of the top and bottom faces, A^{n+1} and A^{n} are the areas of these faces. Each of four vector values $\boldsymbol{Q}_{11'4'4}$, $\boldsymbol{Q}_{233'2'}$, $\boldsymbol{Q}_{122'1'}$ and $\boldsymbol{Q}_{433'4'}$ is an average flux of mass, impulse and energy through the corresponding intercell surface towards the outward normal vector.

For example, to the face $122'1'$ the value $\boldsymbol{Q}_{122'1'}$ has the following structure

$$\boldsymbol{Q}_{122'1'} = \boldsymbol{\sigma}_{122'1'}A^{xy}_{122'1'} + \boldsymbol{a}_{122'1'}A^{yt}_{122'1'} + \boldsymbol{b}_{122'1'}A^{tx}_{122'1'}\ , \tag{3}$$

where $A^{xy}_{122'1'}$, $A^{yt}_{122'1'}$, $A^{tx}_{122'1'}$ are the areas of projections of the face $122'1'$ onto the coordinate planes $x-y$, $y-t$ and $t-x$, respectively. In (3) all values are defined in the center of the face $122'1'$ using the post-wave values after solving the Riemann problem. Pre-wave values $\boldsymbol{f}$ in the center of the face are obtained by time splitting and a special interpolation procedure that provides the scheme with the second-order accuracy in time and space in the domains of smooth solution.

We shall describe the algorithm to calculate the pre-wave values $\boldsymbol{f}$. For clarity let us consider 1-D case. In 2-D case such an algorithm is executed along each of the curvilinear coordinates.

Let us consider a cell of the 1-D grid along the curvilinear coordinate ξ shown in Fig.1b. Assume the function $\boldsymbol{f}$ to be linear within the zone $(i, i+1)$. Then $\boldsymbol{f}^n_{1,i+1/2}$ and $\boldsymbol{f}^n_{2,i+1/2}$, the values at the left and right zone edges, are defined by

$$\boldsymbol{f}^n_{1,i+1/2} = \boldsymbol{f}^n_{i+1/2} - \frac{1}{2}\delta\boldsymbol{f}^n_{i+1/2}h^n_{i+1/2}\ \ ,\quad \boldsymbol{f}^n_{2,i+1/2} = \boldsymbol{f}^n_{i+1/2} + \frac{1}{2}\delta\boldsymbol{f}^n_{i+1/2}h^n_{i+1/2}\ . \tag{4}$$

Here $h^n_{i+1/2}$ is the spacing in the ξ direction. The derivative $\delta\boldsymbol{f}^n_{i+1/2}$ is defined via the auxiliary values $\tilde{\boldsymbol{f}}_{1,i+1/2}$ and $\tilde{\boldsymbol{f}}_{2,i+1/2}$ at the right and left zone edges

$$\delta\boldsymbol{f}^n_{i+1/2} = (\tilde{\boldsymbol{f}}_{2,i+1/2} - \tilde{\boldsymbol{f}}_{1,i+1/2})/h^n_{i+1/2} \,. \tag{5}$$

The auxiliary value at the left zone edge $\tilde{\boldsymbol{f}}_{1,i+1/2}$ is obtained using the Taylor expansion about the point ξ^n_i within of $O(h^3)$ and $\tilde{\boldsymbol{f}}_{2,i+1/2}$ using the Taylor expansion about ξ^n_{i+1}

$$\tilde{\boldsymbol{f}}_{1,i+1/2}=\boldsymbol{f}_{i+1/2} - \frac{h_{i+1/2}}{h_{i+1/2}+h_{i-1/2}}\left(\boldsymbol{f}_{i+1/2} - \boldsymbol{f}_{i-1/2}\right) - \frac{h^2_{i+1/2}}{4(h_{i+1/2}+h_{i-1/2})}\left(\boldsymbol{f}^{(1)}_{i+1/2} - \boldsymbol{f}^{(1)}_{i-1/2}\right) \,,$$

$$\tilde{\boldsymbol{f}}_{2,i+1/2}=\boldsymbol{f}_{i+1/2} + \frac{h_{i+1/2}}{h_{i+3/2}+h_{i+1/2}}\left(\boldsymbol{f}_{i+3/2} - \boldsymbol{f}_{i+1/2}\right) - \frac{h^2_{i+1/2}}{4(h_{i+3/2}+h_{i+1/2})}\left(\boldsymbol{f}^{(1)}_{i+3/2} - \boldsymbol{f}^{(1)}_{i+1/2}\right) . \tag{6}$$

Here the superscript n is omitted for convenience and all values are taken at time level n.

Every components of the derivative $\boldsymbol{f}^{(1)}$ is defined by the Kolgan's principle of minimal derivatives [9]. For example, to velocity u it looks as

$$u^{(1)}_{i+1/2} = \begin{cases} u^{(1)}_i \,, & \text{if } |u^{(1)}_i| \le |u^{(1)}_{i+1}| \\ u^{(1)}_{i+1} \,, & \text{otherwise} \,, \end{cases} \tag{7}$$

where $u^{(1)}_i$ is given by relation $u^{(1)}_i=2(u_{i+1/2}-u_{i-1/2})(h_{i+1/2}+h_{i-1/2})$.

Afterwards we apply the Collela-Woodward's monotonicity algorithm [13] to reset the auxiliary values $\tilde{\boldsymbol{f}}_{1,i+1/2}$ and $\tilde{\boldsymbol{f}}_{2,i+1/2}$. We construct an interpolating parabola within the zone for every component of the vector $\boldsymbol{f}$. For example, to u we have

$$U(\zeta) = a_0 + a_1\zeta + a_2\zeta^2 \,, \quad \zeta = (\xi - \xi_{i+1/2})/h_{i+1/2} \,, \quad \xi_i \le \xi \le \xi_{i+1} \,, \quad -1/2 \le \zeta \le 1/2 \,.$$

The parabola takes on the following values at the zone edges $U(-1/2)=u_{1,i+1/2}$, $U(1/2)=u_{2,i+1/2}$ and for it the value $u^n_{i+1/2}$ is a zone average $u^n_{i+1/2} = \int_{-1/2}^{1/2} U(\xi)d\xi$. Coefficients of the parabola are equal $a_1=\tilde{u}_{2,i+1/2}-\tilde{u}_{1,i+1/2}$, $a_2=3(\tilde{u}_{1,i+1/2}+\tilde{u}_{2,i+1/2})-6u_{i+1/2}$.

A condition to be imposed is that the interpolating parabola shall not achieve its maximum or minimum within the zone. Otherwise the values $\tilde{u}_{1,i+1/2}$, $\tilde{u}_{2,i+1/2}$ are corrected to ensure the parabola to be monotone. There are two possible cases when it takes place:

a) $u^n_{i+1/2}$ does not lie inside the interval from $\tilde{u}_{1,i+1/2}$ to $\tilde{u}_{2,i+1/2}$. In such a case the interpolating function is set to be a constant, i.e.

$$\tilde{u}_{1,i+1/2} = \tilde{u}_{2,i+1/2} = u_{i+1/2} \,, \text{ if } \quad (\tilde{u}_{2,i+1/2} - u_{i+1/2})(u_{i+1/2} - \tilde{u}_{1,i+1/2}) \le 0 \,;$$

b) $u^n_{i+1/2}$ lies inside the above interval, but sufficiently close to one of these values. Then we reset $\tilde{u}_{1,i+1/2}$ or $\tilde{u}_{2,i+1/2}$ (which is farther from $u^n_{i+1/2}$) so that the interpolating parabola becomes monotone as follows:

$$\begin{cases} \tilde{u}_{1,i+1/2} = u_{i+1/2} + 2(u_{i+1/2} - \tilde{u}_{2,i+1/2}) \,, & \text{if } a_1^2 < -a_1a_2 \,, \\ \tilde{u}_{2,i+1/2} = u_{i+1/2} + 2(u_{i+1/2} - \tilde{u}_{1,i+1/2}) \,, & \text{if } a_1^2 < a_1a_2 \,. \end{cases}$$

Next from (5) we get $\delta \boldsymbol{f}^n_{i+1/2}$ and from (4) the auxiliary values $\tilde{\boldsymbol{f}}^n_{1,i+1/2}$ and $\tilde{\boldsymbol{f}}^n_{2,i+1/2}$, which in 2-D case correspond to the values at the edges in one of the curvilinear coordinates direction. By analogy we get the values to the other pair of the edges. Afterwards we calculate the fluxes through the intercell faces by formulae (3) and then from (2) the intermediate value $\tilde{\boldsymbol{f}}^{n+1}_{i+1/2}$ at time t^{n+1}.

Pre-wave values $\boldsymbol{f}$ at the center of the intercell faces at time $t^{n+1/2}$ are given by

$$\boldsymbol{f}^{n+1/2}_{1,i+1/2} = \left[\boldsymbol{f}^n_{i+1/2} + \tilde{\boldsymbol{f}}^{n+1}_{i+1/2} - \delta \boldsymbol{f}^n_{i+1/2}(h^n_{i+1/2} + h^{n+1}_{i+1/2})\right]/2 \,,$$

$$\boldsymbol{f}^{n+1/2}_{2,i+1/2} = \left[\boldsymbol{f}^n_{i+1/2} + \tilde{\boldsymbol{f}}^{n+1}_{i+1/2} + \delta \boldsymbol{f}^n_{i+1/2}(h^n_{i+1/2} + h^{n+1}_{i+1/2})\right]/2 \,.$$

Next we solve the Riemann problem to each of four intercell faces taking into account the velocity of the corresponding cell edge [8]. As a result we obtain the post-wave values $\boldsymbol{f}$ at the center of the intercell faces.

Calculating the fluxes through the intercell faces via (3) and substituting these values into (2) we get the final values of mass, impulse and energy $\boldsymbol{\sigma}^{n+1}$ at time t^{n+1}.

Note that when calculating by (6) the auxiliary values at the boundary nodes, for example, $\boldsymbol{f}_{1,i+1/2}$ at the left node i=0, in case the boundary condition is defined as $\boldsymbol{f}_{left}$, we set $\boldsymbol{f}_{1,1/2}$=$\boldsymbol{f}_{left}$, and, in case the boundary condition is defined to the fluxes, for example, reflecting boundary condition, we set $\boldsymbol{f}_{1,1/2}$=$\boldsymbol{f}^{n-1/2}_0$, where $\boldsymbol{f}^{n-1/2}_0$ are the post-wave values at the preceding time step.

In every cell the admissible time step $\triangle t_{i+1/2,j+1/2}$ is defined by [16]

$$\triangle t^{'}_{i+1/2,j+1/2} = \frac{h^{'}_{i+1/2,j+1/2}}{\max(D^{II}_{i,j+1/2} - W_{i,j+1/2}\ ,\ -D^{I}_{i+1,j+1/2} - W_{i+1,j+1/2})}\ ,$$

$$\triangle t^{''}_{i+1/2,j+1/2} = \frac{h^{''}_{i+1/2,j+1/2}}{\max(D^{II}_{i+1/2,j} - W_{i+1/2,j}\ ,\ -D^{I}_{i+1/2,j+1} - W_{i+1/2,j+1})}\ , \qquad (8a)$$

$$h^{'}_{i+1/2,j+1/2} = \frac{A_{1234}}{\sqrt{(x_{i+1/2,j+1} - x_{i+1/2,j})^2 + (y_{i+1/2,j+1} - y_{i+1/2,j})^2}}\ ,$$

$$h^{''}_{i+1/2,j+1/2} = \frac{A_{1234}}{\sqrt{(x_{i+1,j+1/2} - x_{i,j+1/2})^2 + (y_{i+1,j+1/2} - y_{i,j+1/2})^2}}\ ,$$

$$\triangle t_{i+1/2,j+1/2} = \frac{\triangle t^{'}_{i+1/2,j+1/2}\ \triangle t^{''}_{i+1/2,j+1/2}}{\triangle t^{'}_{i+1/2,j+1/2} + \triangle t^{''}_{i+1/2,j+1/2}}\ .$$

Here $\triangle t'$ and $\triangle t''$ are the admissible time steps to 1-D schemes, $h^{'}, h^{''}$ are the "average heights" of the bottom face A_{1234} , W is the velocity of the corresponding cell edge (e.g. $W_{i+1/2,j}$ is the velocity of the edge 12 which defines inclination of the face $122'1'$), see Fig.1a. Next, $D^{II}_{i+1/2,j}$ and $D^{II}_{i,j+1/2}$ are the "extreme right wave" speeds defined from solving the Riemann problem to the faces $122'1'$ and $11'4'4$, respectively; $D^{I}_{i+1,j+1/2}$ and $D^{I}_{i+1/2,j+1}$ are the "extreme left wave" speeds to the faces $233'2'$ and $433'4'$, respectively.

As an admissible time step we take the minimal $\triangle t_{i+1/2,j+1/2}$ of all cells of the grid

$$\triangle t = c_{cfl} \ \min_{i,j} \triangle t_{i+1/2,j+1/2} \ . \tag{8b}$$

The Courant number coefficient c_{cfl} is less then 1 (usually close to 1) and it is introduced as a correction to non-linearity of the problem. Note that the time step depends on both the post-wave values and velocity of every intercell face. In computations the value of $\triangle t$, obtained at the preceding time step, is used to the next time step. By this reason in the case of essentially nonstationary processes the c_{cfl} coefficient should be greatly decreased.

2. Unfolded meshes

Two types of grids/meshes are used in computations: regular (structured) and irregular (unstructured). Regular grids contain only regular nodes, or nodes whose neighbors are known only from the indexing. A typical example is a curvilinear grid constructed by a mapping of a parametric square onto a physical domain. Grid nodes are enumerated with double indices in two-dimensional and by triple indices in 3-D case. This is not the case of irregular meshes. For such a mesh, neighbors of nodes must be specified.

The condition of the Jacobian positiveness is considered as the condition for a regular grid to be invertible (nondegenerate). An irregular mesh can be assumed as a set of local coordinates, so the condition of the Jacobian positiveness can be used also to define discrete conditions for an irregular mesh to be unfolded (nondegenerate).

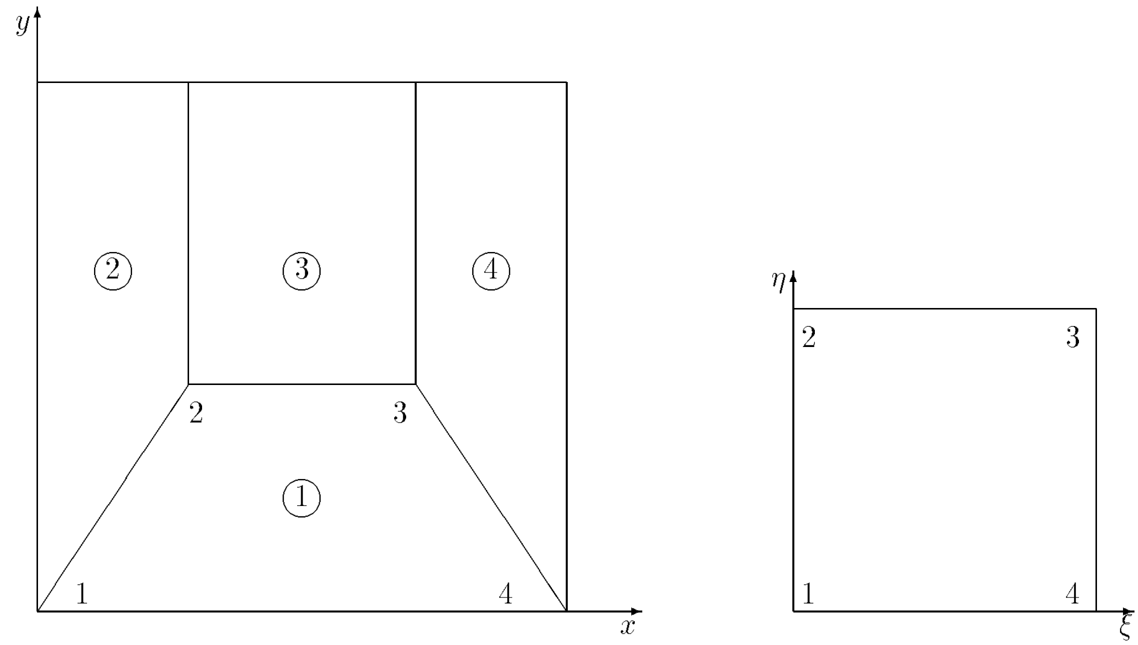

Fig.2. Correspondence of node numbers for a mapping of the unit square in the plane ξ, η onto the quadrilateral cell 1 of an irregular mesh in the plane x, y.

In the employment of irregular meshes we must define the correspondence between local (for each element) and global node enumeration. In Fig.2 the simplest example of an irregular mesh is shown. Element numbers are shown in circles. The local enumeration is shown only for the element 1. The global enumeration is shown with a bold font.

The function $COR(N,k)$ is introduced to define a correspondence between local and global node numbers:

$$COR(N,k) = n,\ n = 1,\dots,N_n,\ N = 1,\dots,N_e,\ k = 1,2,3,4,$$

where n is the global node number, N_n is the total number of mesh nodes, N is an element number, N_e is the number of elements, k is a local node number in the element. This function is implemented in the computer program as a function for a regular grid and as an array for an irregular mesh. For example, for the irregular mesh shown in Fig.2 the correspondence between local and global enumeration is defined as follows

$$COR(1,1) = 1,\ COR(1,2) = 3,\ COR(1,3) = 4,\ COR(1,4) = 2.$$

For irregular meshes the array COR is filled up during the mesh construction, for example, by a marching method. It is often necessary to use another correspondence functions, for example when we must define numbers of two elements from the number of their common edge or to define the neighbor numbers for a given node. The choice of these functions depends on the type of elements used and on the solver peculiarities. We shall consider below only the simplest data structure, defined by $COR(N,k)$, which is enough for our purposes.

For regular grids we can use the function with the same name instead of the array COR. It is convenient to use one-dimensional numeration instead of double indices. For node numbers of a regular grid we have

$$N(i,j) = i + (j-1)(i^*-1),\ i = 1,\dots,i^*-1;\ j = 1,\dots,j^*-1,$$

$$n(i,j) = i + (j-1)i^*,\ i = 1,\dots,i^*;\ j = 1,\dots,j^*,$$

where $n(i,j)$ corresponds to the node i,j, and $N(i,j)$ corresponds to the cell number $i+1/2, j+1/2$. Then the correspondence function is defined as follows

$$COR(N(i,j),1) = n(i,j),\ COR(N(i,j),2) = n(i,j+1),$$

$$COR(N(i,j),3) = n(i+1,j+1),\ COR(N(i,j),4) = n(i+1,j).$$

Now we consider conditions for the mesh node coordinates to assure a mesh to be nondegenerate. Note, that in case of a regular grid instead of the mapping $x(\xi,\eta)$, $y(\xi,\eta)$ of the parametric rectangle onto a domain Ω, a bilinear mapping of the same unit square onto each quadrilateral cell can be considered (see [2]). All argumentation will be true in this case, since the Jacobian of the mapping $x^h(\xi,\eta)$, $y^h(\xi,\eta)$ is not changed if the square cell is shifted in the plane ξ,η. Hence, for each cell of irregular mesh a bilinear mapping of the unit square on the plane ξ, η onto this cell can be introduced (see Fig.2). The condition of the Jacobian positiveness can be written as follows

$$[J_k]_N > 0, \quad k = 1, 2, 3, 4; \; N = 1, \ldots, N_e, \tag{9}$$

where $J_k = (x_{k-1} - x_k)(y_{k+1} - y_k) - (y_{k-1} - y_k)(x_{k+1} - x_k)$ is the doubled area of the triangle, written in local enumeration. Consequently, all mesh cells satisfying inequalities for J_k will be convex quadrilaterals.

Meshes, satisfying inequalities (9) will be called convex or unfolded meshes and denoted by $\mathcal{D}$. This set belongs to the Euclidean space $\mathrm{R}^{\mathrm{N_{in}}}$, where N_{in} is the total number of degrees of freedom of the mesh equal to double the number of its internal nodes. In this space $\mathcal{D}$ is an open bounded set. Its boundary $\partial\mathcal{D}$ is the set of meshes for which at least one of the inequalities (9) becomes an equality.

3. Problem formulation for grid generation

To generate a regular adaptive-harmonic mesh we formulate the problem to minimize the harmonic (Dirichlet's) functional, written for a surface [2]. Notations are shown in Fig.3.

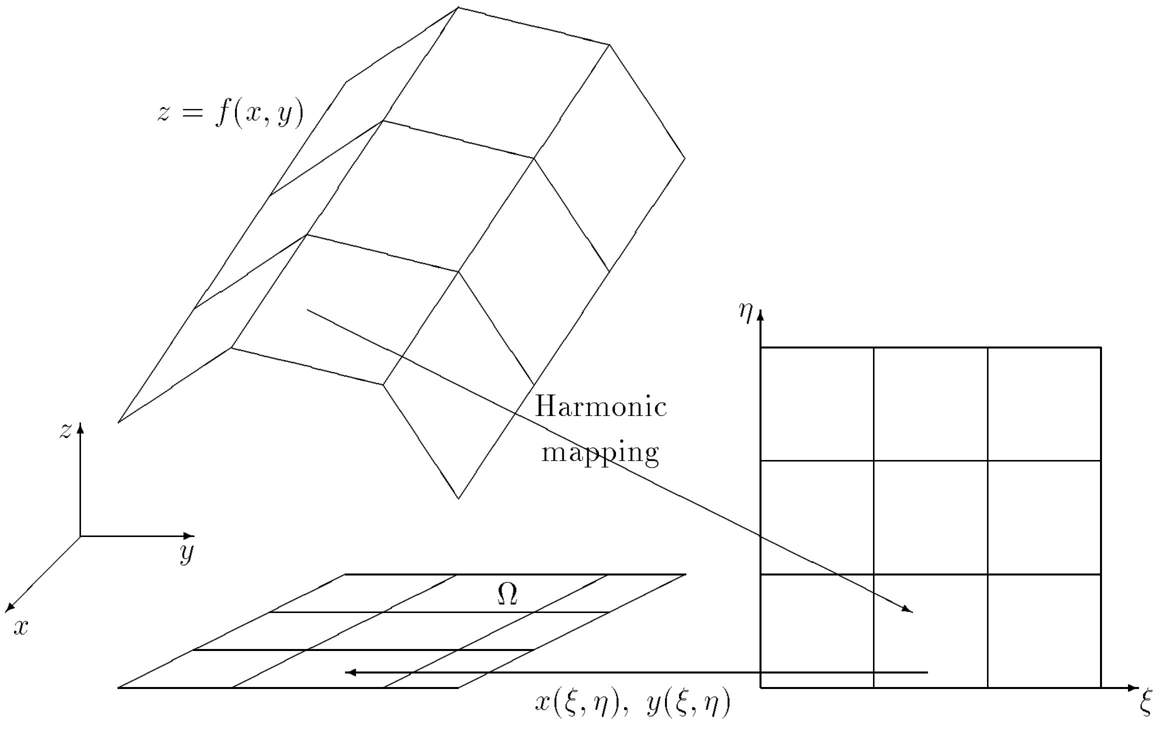

Fig.3. Harmonic coordinates on the surface of the graph of a function $z = f(x, y)$.

Let us consider the functional defining the adaptive-harmonic mesh, clustered in regions of high gradients of the function $f(x, y)$

$$I = \iint \frac{((1 + f_x^2)(x_\xi^2 + x_\eta^2) + 2 f_x f_y (x_\xi y_\xi + x_\eta y_\eta) + (1 + f_y^2)(y_\xi^2 + y_\eta^2)}{(x_\xi y_\eta - x_\eta y_\xi)\sqrt{1 + f_x^2 + f_y^2}} d\xi d\eta. \tag{10}$$

The problem of irregular mesh adaptation (r-refinement) is formulated as follows. Let the coordinates of the irregular mesh nodes be given. The mesh is formed by quadrilateral elements, i.e. the array COR(N,k) is also defined. The problem is to find new coordinates of the mesh nodes minimizing the functional (10) values computed for a mapping of the unit square onto every grid cell. In nonstationary problems such a formulation is considered at every time step and grid adaptation is performed to the control function considered as a function $f(x, y)$ at this time level.

The functional (10) can be minimized both with given fixed location of the boundary nodes and when they moves along the boundary. In the latter case a new position of the boundary point, obtained after one step of the minimization procedure, is projected onto the boundary [11].

In order to control the number of grid nodes in the layer of high gradients, earlier it has been suggested to use the function $c_a f(x, y)$ instead of $f(x, y)$ [2]. Here the function $f(x, y)$ is scaled so that the difference between maximal and minimal values of it is equal to the length of the diagonal of a rectangle enclosing all the boundary points that describe the geometry in the plane x, y, i.e.

$$f_{max} - f_{min} = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}.$$

The larger the coefficient of adaptation c_a the greater the number of grid nodes in the layer of high gradients. Usually c_a is in the range from 0.1 to 0.5. In the case of elliptic problems the share of points in the layer is approximately equal to $c_a/(c_a+1)$, i.e. if c_a=0.5 then about one third of the points is in the layer of high gradients.

Note if the algorithm of adaptive-harmonic grid generation is applied to the elliptic problems, it is assumed the function $f(x, y)$ to be continuously differentiable. If we solve the hyperbolic systems of gas dynamics, the function $f(x, y)$ can be discontinuous and in this case the coefficient c_a changes its essence. Numerical experiments show that in the case of adaptation to a smooth function, when increasing the coefficient c_a the grid begins "to feel" new subdomains where the gradient is rather essential. Moreover, in [17] it was shown that in the case of continuously differentiable function of one variable $f(x)$ in the limit of $c_a \to \infty$ we obtain a grid being optimal in norm L_∞ in a sense that the error of piecewise constant interpolation on such a grid in the norm L_∞ will be minimal. In fact, in 1-D case the functional (10), in which the function $f(x)$ is changed to $c_a f(x)$ and f is scaled to be in the range $0 \leq f(x) \leq 1$, $x, \xi \in [0, 1]$, takes the form

$$I = \int \frac{1}{x_\xi \sqrt{1 + c_a^2 f_x^2}} d\xi,$$

and the Euler equation to this functional, defining asymptotic behavior of the Jacobian of the adaptive-harmonic mesh, may be written as

$$x_\xi = const(1/c_a^2 + f_x^2)^{-1/2}.$$

It can be shown [17] if the grid is optimal in the norm L_∞ for piecewise constant interpolation, at the continuous limit the following expression for the Jacobian holds

$$x_\xi = const|f_x|^{-1}.$$

Hence, the adaptive-harmonic grid is optimal in the norm L_∞ at the limit $c_a \to \infty$.

From the above expressions for x_ξ it follows that if the function f is discontinuous, then, independently on the value $c_a > 0$ at adaptation, the cell size might approach zero in the vicinity of the discontinuity. Thus, it is necessary to modify the discontinuous function f in such a way that, firstly, to get a smooth function and, secondly, the regions of high gradients of the modified function would correspond to the discontinuities of the function f.

It can be achieved if, for example, to introduce an artificial viscosity or change problem formulation, i.e. instead of the Euler equations consider the Navier-Stokes equations. Here we use another approach based on introducing an additional parameter restricting the maximal value of the control function gradient. In such an approach the number of grid points in the vicinity of the shock turns should be limited and does not depend on c_a . Now c_a plays the following role: when increasing the coefficient c_a the grid begins "to feel" the discontinuities of less intensity. Decrease of c_a results in decreasing sensitivity of the grid to the discontinuities of greater intensity and in the limit of c_a=0 we obtain a quasiuniform harmonic grid not depending of $f(x,y)$.

Thus, the coefficient c_a is only responsible for selecting those discontinuities of the function $f(x,y)$ to which the mesh will be adapted. Therefore, it is necessary to have one more controlling parameter determining the thickness of the layer of high gradients that approximates the discontinuity line in the solution and simultaneously restricts maximum gradient of the grid control function. It is accomplished as follows. First, on the existing mesh we define the function $\tilde{f} = c_a f^h$, where f^h is an interpolant of the function f whose value in the nodes coincides with the value of the function f. When solving the problem of gas dynamics, the function f is defined in the center of cells, therefore, we shall update the values of f to the nodes. We apply the simplest interpolation formulae using the area of cells surrounding a given node. Then we specify D_{max}, maximal value of modulus of gradient to the control function f, for example, as D_{max}=$\lambda \max(|\nabla \tilde{f}|)$, where the coefficient λ<1 and $|\nabla \tilde{f}|$=$\sqrt{\tilde{f}_x^2 + \tilde{f}_y^2}$. Next the gradient of the function is updated as follows

$$\nabla \tilde{f}^* = \begin{cases} D_{max} \nabla \tilde{f} / |\nabla \tilde{f}| & \text{if } |\nabla \tilde{f}| > D_{max} \ , \\ \nabla \tilde{f} & \text{otherwise} \ . \end{cases}$$

Then obtained values $\tilde{f}_x^*$ and $\tilde{f}_y^*$ are substituted into (10) instead of f_x and f_y.

It is not necessary to use this procedure in simulation of smooth subsonic flows, however, in the case of supersonic flows we shall apply it to prevent cells to be folded in the vicinity of the shock waves and tangential discontinuities.

If it is necessary to generate the grid with adaptation to the vector-function with components

$f_i(x,y)$, the functional (10) can be generalized as

$$I = \iint \frac{g_{11}(x_\xi^2 + x_\eta^2) + 2g_{12}(x_\xi y_\xi + x_\eta y_\eta) + g_{22}(y_\xi^2 + y_\eta^2)}{(x_\xi y_\eta - x_\eta y_\xi)\sqrt{g_{11}g_{22} - g_{12}^2}} d\xi d\eta \ ,$$

where

$$g_{11} = 1 + \sum_i c_i^2 (f_i)_x^2, \quad g_{12} = \sum_i c_i^2 (f_i)_x (f_i)_y, \quad g_{22} = 1 + \sum_i c_i^2 (f_i)_y^2.$$

Geometrical meaning of the problem to minimize the above functional is the following. This problem is equivalent to the one on constructing the harmonic coordinates on the surface of the graph of the control function with the components $r(x,y)=(c_1 f_1, c_2 f_2, ...)(x,y)$ (see [12]). The harmonic coordinates are defined using a harmonic mapping of the surface onto the parametric square, see. Fig.3. As a result we obtain a parametrization of the surface $(x, y, c_1 f_1, c_2 f_2 ...)(\xi, \eta)$. In case of an irregular mesh we consider the harmonic mapping of every cell on the surface of the vector-function onto the same square cell.

In such an approach the functions $f_i(x,y)$ can be both the gas dynamics values u, v, p, ρ and some function depending on them, for example, $|V|=\sqrt{u^2+v^2}$. Derivatives of those functions are normalized as above using the corresponding maximal modulus of the gradients $D_{i,max}$.

4. Approximation and minimization of the functional

The functional can be approximated in such a way that its minimum is attained on a grid/mesh of convex quadrilaterals:

$$I^h = \sum_{N=1}^{N_e} \sum_{k=1}^{4} \frac{1}{4} [F_k]_N \ , \tag{11}$$

where

$$F_k = \frac{D_1[1 + (f_x)_k^2] + D_2[1 + (f_y)_k^2] + 2D_3 (f_x)_k (f_y)_k}{J_k [1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

$$D_1 = (x_{k-1} - x_k)^2 + (x_{k+1} - x_k)^2, \quad D_2 = (y_{k-1} - y_k)^2 + (y_{k+1} - y_k)^2,$$

$$D_3 = (x_{k-1} - x_k)(y_{k-1} - y_k) + (x_{k+1} - x_k)(y_{k+1} - y_k),$$

$$J_k = (x_{k-1} - x_k)(y_{k+1} - y_k) - (x_{k+1} - x_k)(y_{k-1} - y_k).$$

Here $(f_x)_k$ and $(f_y)_k$ are the values of derivatives at node number k of cell number N.

The function I^h possesses the following very important property (see [2]).

The function I^h has an infinite barrier at the boundary of the set of unfolded meshes, i.e. if at least one of the quantities J_k in (9) tends to zero for some cell while remaining positive, then $I^h \to +\infty$.

From this it follows, that if the set of unfolded meshes $\mathcal{D}$ is not empty, the system of algebraic equations

$$R_x = \frac{\partial I^h}{\partial x_n} = 0, \ R_y = \frac{\partial I^h}{\partial y_n} = 0,$$

has at least one solution which is an unfolded mesh. To find it, one must first find a certain initial unfolded mesh, and then use some method of unconstrained minimization of the function I^h. Since this function has an infinite barrier on the boundary of the set $\mathcal{D}$, each step of the method can be chosen so that the mesh always remains unfolded (see [11]).

Suppose the mesh at the l-th step of iterations is determined. We use the quasi-Newtonian procedure when the $(l+1)$-th step is accomplished as follows

$$x_n^{l+1} = x_n^l - \tau \left(R_x \frac{\partial R_y}{\partial y_n} - R_y \frac{\partial R_x}{\partial y_n} \right) \left(\frac{\partial R_x}{\partial x_n}\frac{\partial R_y}{\partial y_n} - \frac{\partial R_y}{\partial x_n}\frac{\partial R_x}{\partial y_n} \right)^{-1},$$

$$y_n^{l+1} = y_n^l - \tau \left(R_y \frac{\partial R_x}{\partial x_n} - R_x \frac{\partial R_y}{\partial x_n} \right) \left(\frac{\partial R_x}{\partial x_n}\frac{\partial R_y}{\partial y_n} - \frac{\partial R_y}{\partial x_n}\frac{\partial R_x}{\partial y_n} \right)^{-1}, \tag{12}$$

where τ is the iteration parameter, which is chosen so that the mesh remains unfolded. For this purpose after every step conditions (9) are checked and if they are not satisfied, this parameter is multiplied by 0.5. Then conditions (9) are checked for the grid, computed with a new value of τ and if they are not satisfied, this parameter is multiplied by 0.25 and so on.

As a result the variational method of adaptive-harmonic grid generation has immanent guarantee to produce grids free of folding at every time step.

5. Derivation of computational formulas

Four triangles, introduced in sec.2, are considered for the quadrilateral cell number N. Each of these triangles corresponds to a corner with the number k and gives a proper contribution to the functional and also to the values of its derivatives. Since the integrand in (10) does not depend on the rotation of the coordinate system ξ, η, then all the computational formulas will be the same for all triangles. We enumerate nodes of the triangle which corresponds to the corner with the local number k from 1 to 3 as follows:
node 1 corresponds to the local node number $k-1$ of the cell N,
node 2 corresponds to the local node number k of the cell N,
node 3 corresponds to the local node number $k+1$ of the cell N.

Then in the new enumeration the expression for F_k will be

$$F_k = \frac{D_1[1+(f_x)_k^2] + D_2[1+(f_y)_k^2] + 2D_3(f_x)_k(f_y)_k}{J_2[1+(f_x)_k^2+(f_y)_k^2]^{1/2}}, \tag{13}$$

where

$$D_1 = (x_1-x_2)^2 + (x_3-x_2)^2, \quad D_2 = (y_1-y_2)^2 + (y_3-y_2)^2,$$

$$D_3 = (x_1-x_2)(y_1-y_2) + (x_3-x_2)(y_3-y_2), \quad J_2 = (x_1-x_2)(y_3-y_2) - (x_3-x_2)(y_1-y_2).$$

We introduce notations

$$U = \frac{D_1[1+(f_x)_k^2] + D_2[1+(f_y)_k^2] + 2D_3(f_x)_k(f_y)_k}{[1+(f_x)_k^2+(f_y)_k^2]^{1/2}},$$

$$V = (x_1 - x_2)(y_3 - y_2) - (x_3 - x_2)(y_1 - y_2).$$

We use formulas for the derivatives of the relation of two functions. Differentiating, we obtain

$$F = \frac{U}{V},$$

$$F_x = \frac{U_x - FV_x}{V}, \quad F_y = \frac{U_y - FV_y}{V}, F_{xx} = \frac{U_{xx} - 2F_xV_x - FV_{xx}}{V},$$

$$F_{yy} = \frac{U_{yy} - 2F_yV_y - FV_{yy}}{V}, F_{xy} = F_{yx} = \frac{U_{xy} - F_xV_y - F_yV_x - FV_{xy}}{V}. \tag{14}$$

For the triangle vertex with the number 1 we should substitute appropriate expressions instead of U and V, U_x and V_x and so on into (14) and replace x and y by x_1 and y_1.

For the vertex 1 we have

$$V_x = y_3 - y_2, \quad V_y = x_2 - x_3, \; V_{xx} = 0, \quad V_{xy} = 0, \quad V_{yy} = 0.$$

$$U_x = 2\frac{[1 + (f_x)_k^2](x_1 - x_2) + (f_x)_k(f_y)_k(y_1 - y_2)}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

$$U_y = 2\frac{[1 + (f_y)_k^2](y_1 - y_2) + (f_x)_k(f_y)_k(x_1 - x_2)}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

$$U_{xx} = 2\frac{1 + (f_x)_k^2}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}}, \; U_{xy} = 2\frac{(f_x)_k(f_y)_k}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}}, \; U_{yy} = 2\frac{1 + (f_y)_k^2}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

For the vertex 2 we have

$$V_x = y_1 - y_3, \quad V_y = x_3 - x_1, \; V_{xx} = 0, \quad V_{xy} = 0, \quad V_{yy} = 0.$$

$$U_x = 2\frac{[1 + (f_x)_k^2](2x_2 - x_1 - x_3) + (f_x)_k(f_y)_k(2y_2 - y_1 - y_3)}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

$$U_y = 2\frac{[1 + (f_y)_k^2](2y_2 - y_1 - y_3) + (f_x)_k(f_y)_k(2x_2 - x_1 - x_3)}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

$$U_{xx} = 4\frac{1 + (f_x)_k^2}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}}, \; U_{xy} = 4\frac{(f_x)_k(f_y)_k}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}}, \; U_{yy} = 4\frac{1 + (f_y)_k^2}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

For the vertex 3 we have

$$V_x = y_2 - y_1, \quad V_y = x_1 - x_2, \; V_{xx} = 0, \quad V_{xy} = 0, \quad V_{yy} = 0.$$

$$U_x = 2\frac{[1 + (f_x)_k^2](x_3 - x_2) + (f_x)_k(f_y)_k(y_3 - y_2)}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

$$U_y = 2\frac{[1 + (f_y)_k^2](y_3 - y_2) + (f_x)_k(f_y)_k(x_3 - x_2)}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

$$U_{xx} = 2\frac{1 + (f_x)_k^2}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}}, \; U_{xy} = 2\frac{(f_x)_k(f_y)_k}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}}, \; U_{yy} = 2\frac{1 + (f_y)_k^2}{[1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}},$$

Computations are performed as follows. Let F and its derivatives by x_1 and y_1 be computed with the use of formulas (14) for the cell number N and triangle number k. Then the computed values are added to the appropriate array elements

$$I^h += F, \quad [R_x]_n += F_x, \quad [R_y]_n += F_y,$$

$$[R_{xx}]_n + = F_{xx}, \quad [R_{xy}]_n + = F_{xy}, \quad [R_{yy}]_n + = F_{yy},$$

where $n = COR(N, k-1)$.

Similarly for vertex 2, the correspondence between local and global number is $n = COR(N, k)$.

Similarly for vertex 3, the correspondence is defined as $n = COR(N, k+1)$.

Derivatives $[f_x]_n$ and $[f_y]_n$ are computed as follows. All triangles of the mesh are scanned and for the triangle number k of the cell number N the following values are computed

$$fx = (f_1 - f_2)(y_3 - y_2) - (f_3 - f_2)(y_1 - y_2),$$

$$fy = (x_1 - x_2)(f_3 - f_2) - (x_3 - x_2)(f_1 - f_2),$$

$$J_2 = (x_1 - x_2)(y_3 - y_2) - (x_3 - x_2)(y_1 - y_2).$$

Where f_1, f_2 and f_3 are values of the function f at vertices of the triangle, numbered 1, 2 and 3, corresponding to local numbers of corners of a quadrilateral cell $k-1$, k and $k+1$. Computed values are added to corresponding array elements (which were first cleared)

$$[f_x]_n + = fx, \quad [f_y]_n + = fy, \quad [J]_n + = J_2, \quad n = COR(N, k).$$

New values of derivatives are computed as follows

$$[f_x]_n / = [J]_n, \quad [f_y]_n / = [J]_n.$$

Here, according to C-language notation, $a+ = b$ means, that the new value of a becomes equal to $a + b$, and $a/ = b$ means, that the new value of a becomes equal to a/b.

Values of derivatives, multiplied by c_a, are recalculated in such a way, that maximum of gradient modulus does not exceed D_{max} as described in sec.3. Obtained values of derivatives are substituted into computational formulae.

In the case of adaptation to the vector-function we replace $[1 + (f_x)_k^2]$ by g_{11}, $[(f_x)_k (f_y)_k]$ by g_{12}, $[1 + (f_y)_k^2]$ by g_{22}, $\sqrt{1 + (f_x)_k^2 + (f_y)_k^2}$ by $\sqrt{g_{11} g_{22} - g_{12}^2}$. Here g_{11},g_{12} and g_{22} are computed by using the formulae from sec.3.

6. Dynamic solution-adaptive coupled algorithm

The process of grid generation usually contains preliminary and two main stages. At the preliminary stage coordinates of the boundary points are determined. At the first stage using coordinates of the boundary points we obtain coordinates of internal nodes. This is initial mesh generation. Here we use folding-free harmonic grid generation algorithm, ensuring all grid cells to be convex quadrilaterals [2].

At the second stage we perform simulation of the physical process on the mesh which can be adjusted to the singularities of the numerical solution with purpose to increase the accuracy of computations. Further it is possible to return to the preliminary or first main stages if some correction of the boundary or initial mesh is required.

Note the process of grid generation is multi-component and, therefore, different algorithms must be developed and applied at different stages.

One time step to solve the 2-D equations of gas dynamics with automatic grid adaptation contains the following stages:

1. Generate the mesh at the next time step. As an initial approach we can either use the mesh from the previous time level or, that is most effective, define the node coordinates by their velocity from the previous time level. Value of time step $\triangle t$, defined via (8), is calculated at the previous step from the stability condition.

2. Compute the gas dynamics values at the next time level using the $GLFC$ scheme described in sec.1.

3. Update the gas dynamics value chosen as a control function from the cells center to nodes. The result is a control function value f_{ij} at every mesh node.

4. Evaluate the derivatives $(f_x)_{ij}$ and $(f_y)_{ij}$ at every mesh node.

5. Make one iteration step and compute the new values of x_{ij} and y_{ij} by formulae (12).

6. Repeat starting with step 2 to convergence.

7. Compute the final gas dynamics values at the next time level using the flow solver.

To accelerate the adaptation procedure we apply the following algorithm [10]. At every time step after the first main iteration performed for all grid nodes, further additional iterations (usually 3 to 4 iterations), including solution of the problem of gas dynamics, are performed only to those nodes that displaced more than 0.1 of maximal node shift. It allows to decrease the run time about three times.

7. Flow in channel

The method presented here was applied to the well known test of the planar nonstationary supersonic flow in the wind tunnel containing a step [14]. To estimate the accuracy of numerical solution we use the values which can be easily estimated analyzing the density, pressure or Mach number contours and positions of shocks.

The wind tunnel is 1 unit wide and 3 units length, step is 0.2 units height and begins at x=0.6, see Fig.4. Initial conditions are $\boldsymbol{f}=(3,0,1,1.4)^T$ that corresponds to the flow-in boundary condition at x=0 as well. Ratio of specific heats γ=1.4. The exit boundary has the transmissive conditions since at x=3 the flow is supersonic. Along the walls of the tunnel reflecting boundary conditions are applied.

The process is nonstationary, a steady flow develops only by time 12. However, after time 4 the structure of the flow changes weakly and we shall consider the time evolution up to t=4. The corner of the step is the center of a rarefaction fan and, hence, is a singular point of the flow. Just above the step there is a thin numerical "boundary layer". Shock wave interacts with this layer and qualitative nature of the flow near the step is altered. When computing on the rectangular mesh we apply the additional boundary condition to the density and velocity near the corner of the step [14] to minimize numerical errors generated by the corner.

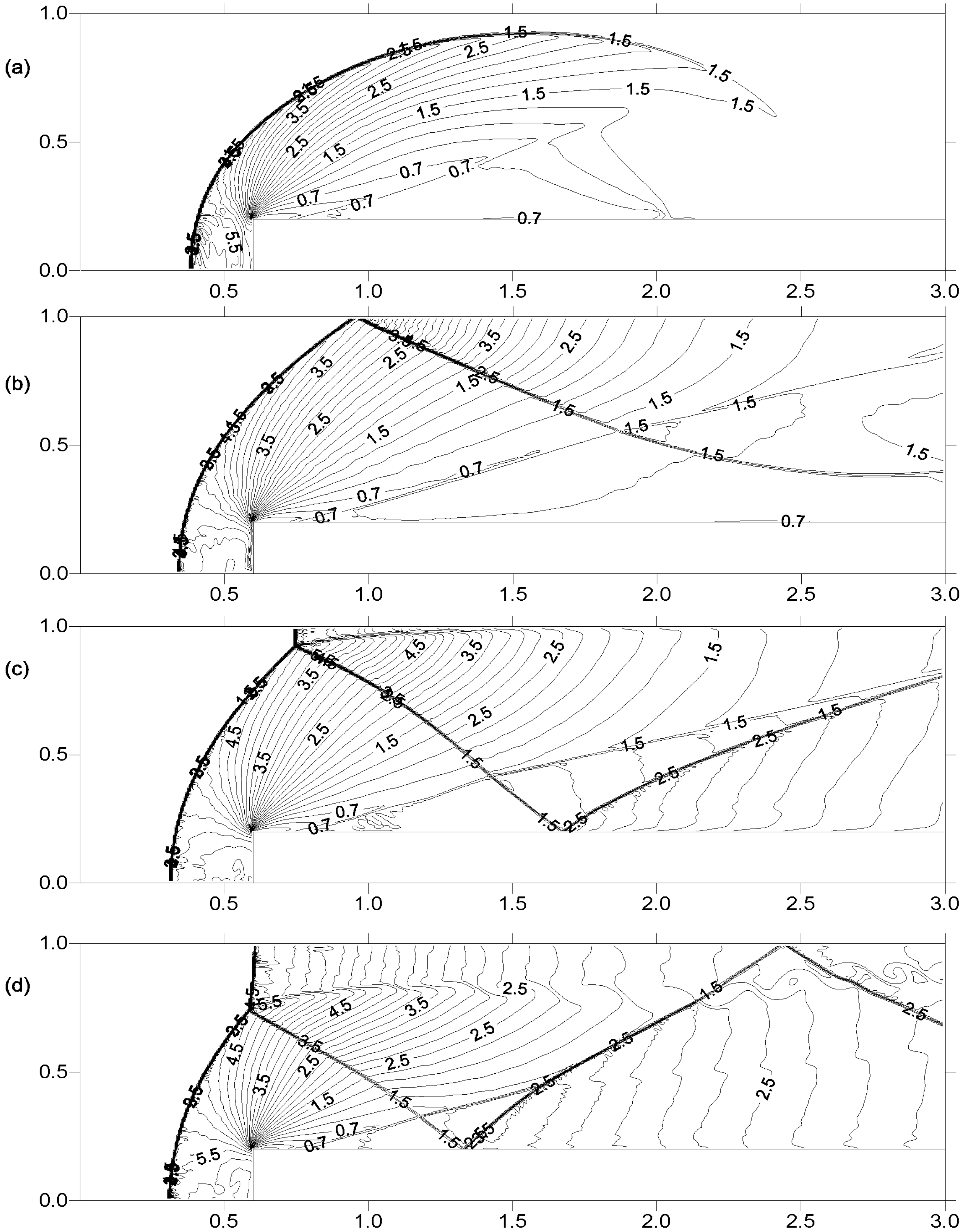

Fig.4. Density contours at t=0.5(a), 1(b), 2(c), 4(d) on the uniform rectangular 480×160 grid.

In Fig.4a-d the density contours are shown at time moments t=0.5, 1, 2, 4. Computations were performed on the rectangular uniform 480×160 grid with the spacings h_x=h_y=1/160 by the $GLFC$ scheme. Note that both in this case and in all further computations the results obtained by the first-order scheme show the great thickness of smearing to the reflected shock waves and they were recognized improper to be presented here. The results in Fig.4 are the same that in [14]. At time t=0.5 (Fig.4a) the bow shock is formed, then extending by t=1 it reaches the top wall and reflects from it (Fig.4b). In some time, about t=2 (Fig.4c), the shock reflects from the step and near the top wall the Mach stem emerges and the contact discontinuity emanates from the triple point. By time t=4 (Fig.4d) the shock wave has already reflected triply, the contact discontinuity has intersected twice the shock. These results obtained on such a refined grid we will consider as a standard to estimate the accuracy of calculations on the moving and adaptive grids.

In the first set of computations we have studied possibility to increase the accuracy by using the technology on moving grids and consisting in cutting the physical domain into subdomains with boundaries being the shock waves or contact discontinuities. However, in this example it is easy to capture only the bow shock. Attempt to capture the reflecting shocks as the boundary of subdomains runs into serious difficulties since both the boundary of subdomains and their number are changed during the process.

In Fig.5a,b the 150×60 grid and density contours are shown at time t=4. When calculating we apply the procedure to capture the bow shock and Mach stem without using the grid adaptation procedure. Additional boundary condition imposed near the corner of the step [14] on such a curvilinear grid leads to increasing the errors caused by the corner influence and we do not apply it. Nevertheless, we obtain the reflection from the step similar to obtained on the rectangular grid with applying that boundary condition. Thus, using the proper grid conforming with the solution, when the grid lines align with the stream lines near the corner, enables to reduce those numerical errors. If to calculate on the rectangular mesh without above additional boundary condition, we shall have irregular reflection from the step [15]. We can estimate the error comparing the results shown in Fig.5b with standard results shown in Fig.4d. It can be seen the thickness of the reflected shock wave about 3 to 4 times larger and intensity is the same as for the standard.

First, we have performed computations with adaptation to the vector-function $(u, v, p, \rho)^T$. The adapted grid at t=4 and u, v, p, ρ contours are shown in Fig.6. The adaptation procedure was switched on at time 3.95, before it the calculations were performed only with fitting the bow shock. Since time 3.95 every time step includes one additional iteration for the grid and during it we solve the gas dynamics problem for entire grid. Thus, the total number of iterations for the grid at every time step N_{iter} was equal to 2. Adaptation was performed along the boundaries as well, including the moving bow shock, with the use of 1-D algorithm [11]. The iteration parameter τ=0.15, coefficient of adaptation was defined as follows:

$$c_a = \begin{cases} 0.2 & \text{if} \quad x \leq 0.6 \,, \\ 0.2 - 0.05(x - 0.6)/(3 - 0.6) & \text{if} \quad x > 0.6 \,, \end{cases}$$

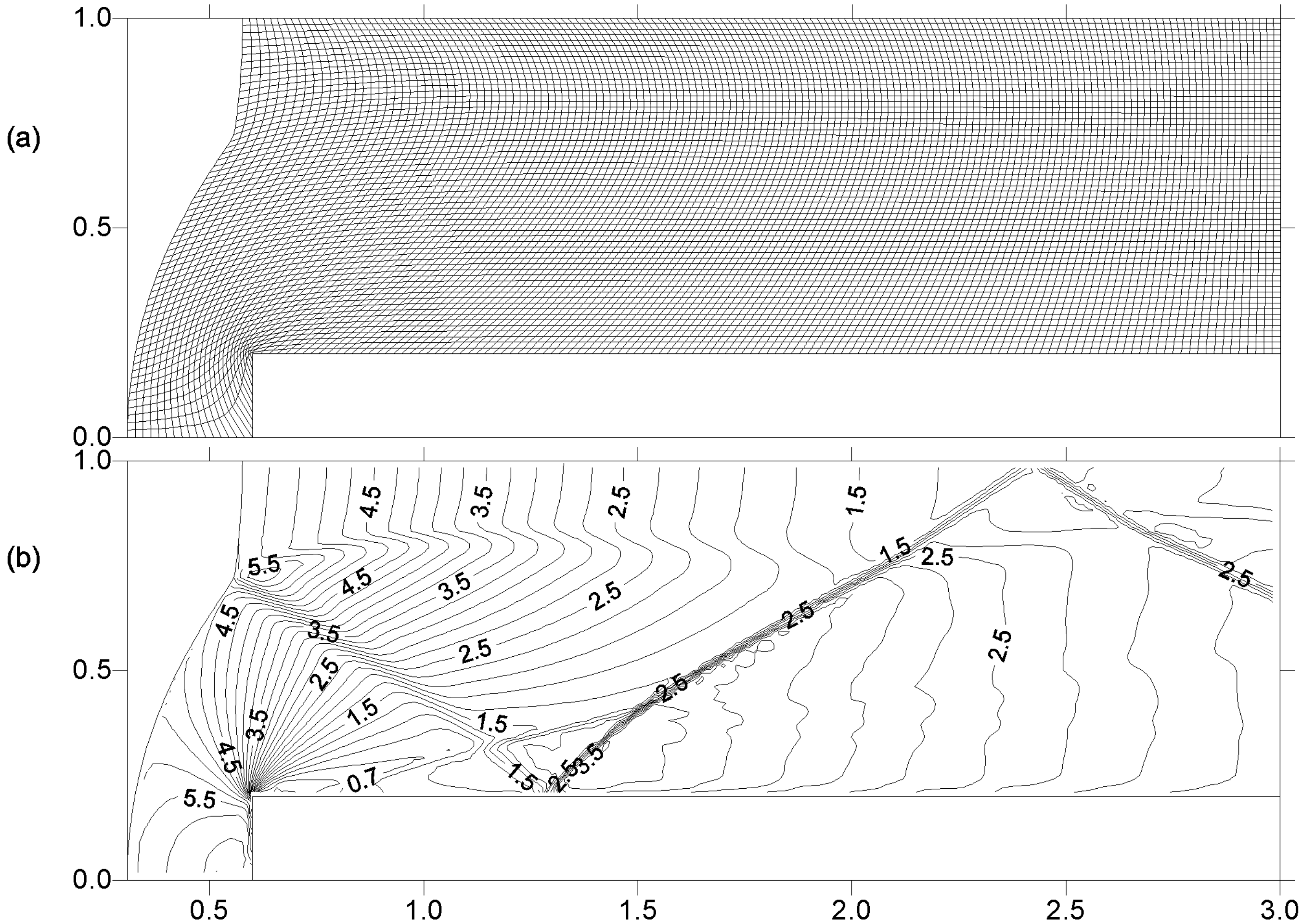

Fig.5. Moving quasiuniform 150×60 grid (a) applied together with the shock fitting procedure to the left bow shock, Mach stem and density contours (b) at t=4.

and for each component of the vector-function c_i=0.25c_a, coefficient D_{max}=100 in all variants. We can see grid lines compression to the shock wave emanating from the triple point is strong enough meanwhile the contact discontinuity is weakly indicated. Disturbances from the corner distort strongly the grid cells and we had to decrease c_a down to 0.05 in the vicinity of the point where the shock wave reflects from the step. Otherwise we would get distorted cells. Furthermore, we switched off adaptation (c_a=0) in the vicinity of the corner in all variants since the rarefaction wave affects the shape of cells in the same manner.

Results of computations with adaptation to scalar functions ρ and $|V|$ are presented in Fig.7a and Fig.7b, correspondingly. Contours of $|V|$ are presented in Fig.7,c. Fragment of the grid from Fig.7b is shown in Fig.7d. Coefficient of adaptation c_a was defined as above. In the case of adaptation to ρ the grid does not "feel" the contact discontinuity, but in the case of adaptation to $|V|$ the grid compression to the contact discontinuity was obtained. Comparing Fig.6e and Fig.7c we can see that the reason lays in different behavior of the ρ and $|V|$ contours in the vicinity of the contact discontinuity.

We see that computations including adaptation coupled with the shock-fitting procedure have been quite successful. It can be recommended in various applications where the bow shocks can be captured as boundary lines.

Easy way to obtain nearly the same resolution is to perform global adaptation without shock fitting.

In the next calculations with global adaptation the grid has been adapted to the vector-function $(u, v, p, \rho)^T$. Figs.8,9 present the grids and density contours at the same time moments as in Fig.4. The main calculation was performed with $N_{iter} = 2$, D_{max}=5 and c_a=0.1. In order to increase the accuracy at 0.05 before every time point the adaptive parameters were changed to:

1) t=0.5 (see Fig.8a,b) $N_{iter} = 10$, D_{max}=100, c_a=0.075 ,

2) t=1 (see Fig.8c,d) $N_{iter} = 6$, D_{max}=100

$$c_a = \begin{cases} 0.075 & \text{if } \ x \leq 1 \ , \\ 0.075 + 0.175(x-1)/(3-1) & \text{if } \ x > 1 \ ; \end{cases}$$

3) t=2 (see Fig.9a,b) $N_{iter} = 4$, D_{max}=100

$$c_a = \begin{cases} 0.05 & \text{if } \quad x \leq 0.8 \ , \\ 0.05 + 0.15(x-0.8)/(1.5-0.8) & \text{if } \ 0.8 < x < 1.5, \\ 0.2 & \text{if } \quad x > 1.5 \ ; \end{cases}$$

4) t=4 (see Fig.9c,d) $N_{iter} = 2$, D_{max}=100

$$c_a = \begin{cases} 0.05 & \text{if } \quad x \leq 0.6 \ , \\ 0.05 + 0.15(x-0.8)/(1-0.6) & \text{if } \ 0.6 < x < 1., \\ 0.2 & \text{if } \quad x > 1 \ . \end{cases}$$

For each component of the vector-function c_i=0.25c_a. The choice of different values of N_{iter} is caused by the different speed of the flow evolution.

We were forced to pay particular attention to the subdomain where the grid lines "formed" the triple point, see Figs.9a,c, since it is a very sensitive place and large value of c_a can cause the grid lines to overlap. One more way to prevent grid folding in this subdomain is to decrease temporally the iteration parameter τ when the triple point is being formed.

The contours of u, v, p, ρ at t=4 are presented in Fig.10. As in computations with shock-fitting (see Fig.7b) we expect the grid will condense strongly to the tangential discontinuity with $|V|$ as an adaptation criterion. Parameters of adaptation in this case were defined as above. The grid, ρ, $|V|$ contours and fragment of the grid at t=4 are shown in Fig.11. In whole, due to grid adaptation the shock waves thickness decreases greatly. In the fragment of the grid, Fig.11c, we see all the shocks are indicated by very compressed grid lines. Accuracy of computations increases due to appearing the very narrow stretched cells in the vicinity of the bow shock, Mach stem and reflected shocks. Maximal aspect ratio achieves 50. Approximately by this factor the width of smearing of the shocks is decreased. This can cause the accuracy of numerical solution to be increased by several times [10]. Presence of distorted cells in the vicinity of the shock wave, emanating from the triple point, probably influences insignificantly the quality of the solution. In the region, where the shock reflects from the step, Fig.11a, the line of nodes clustering is a bit distorted due to the mentioned numerical boundary layer and as above we set here c_a=0.05.

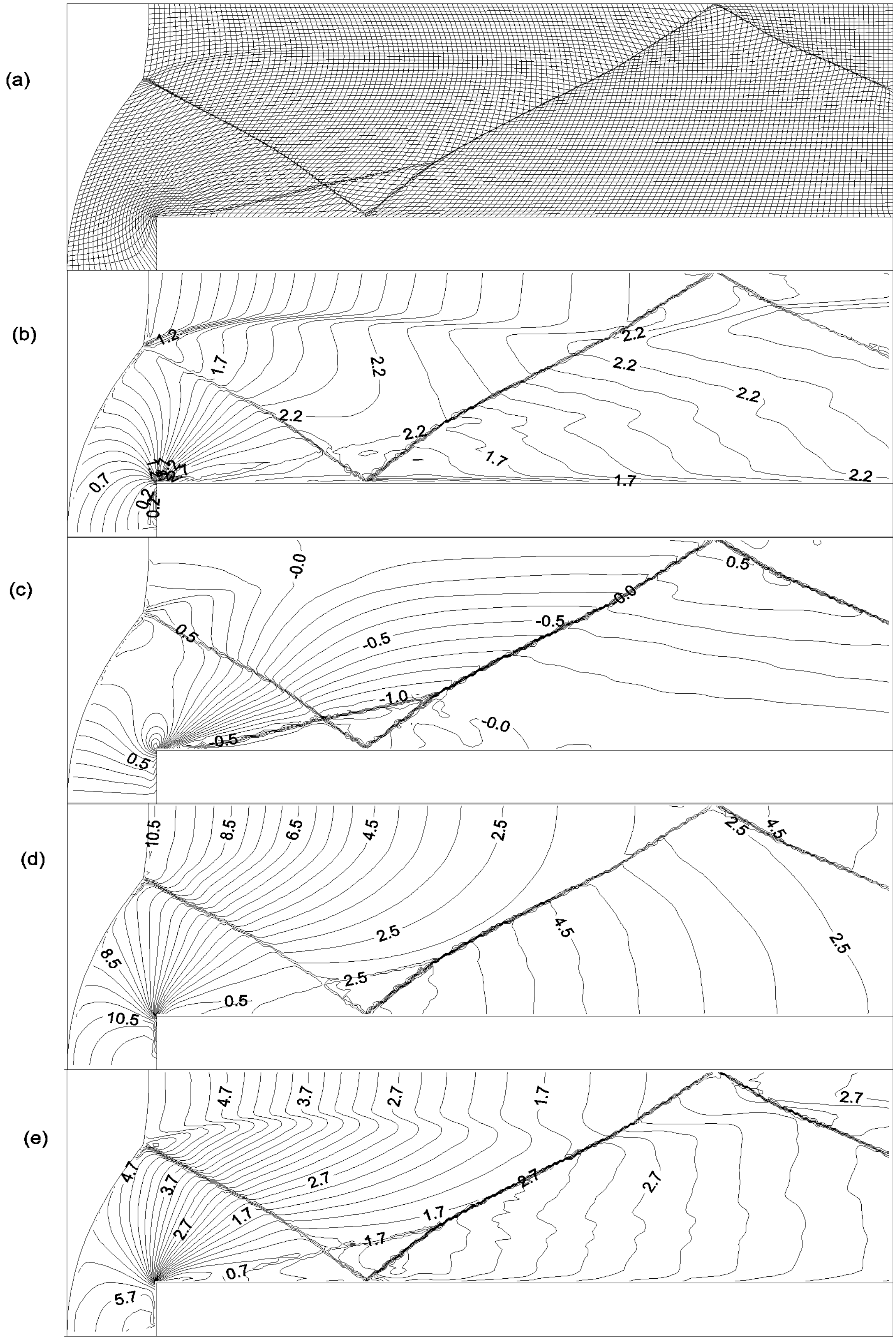

Fig.6. Adaptive 150×60 grid (a) applied together with the shock-fitting procedure at t=4; velocity components u (b) and v (c), pressure (d) and density (e) contours. Adaptation to the vector-function $(u, v, p, \rho)^T$.

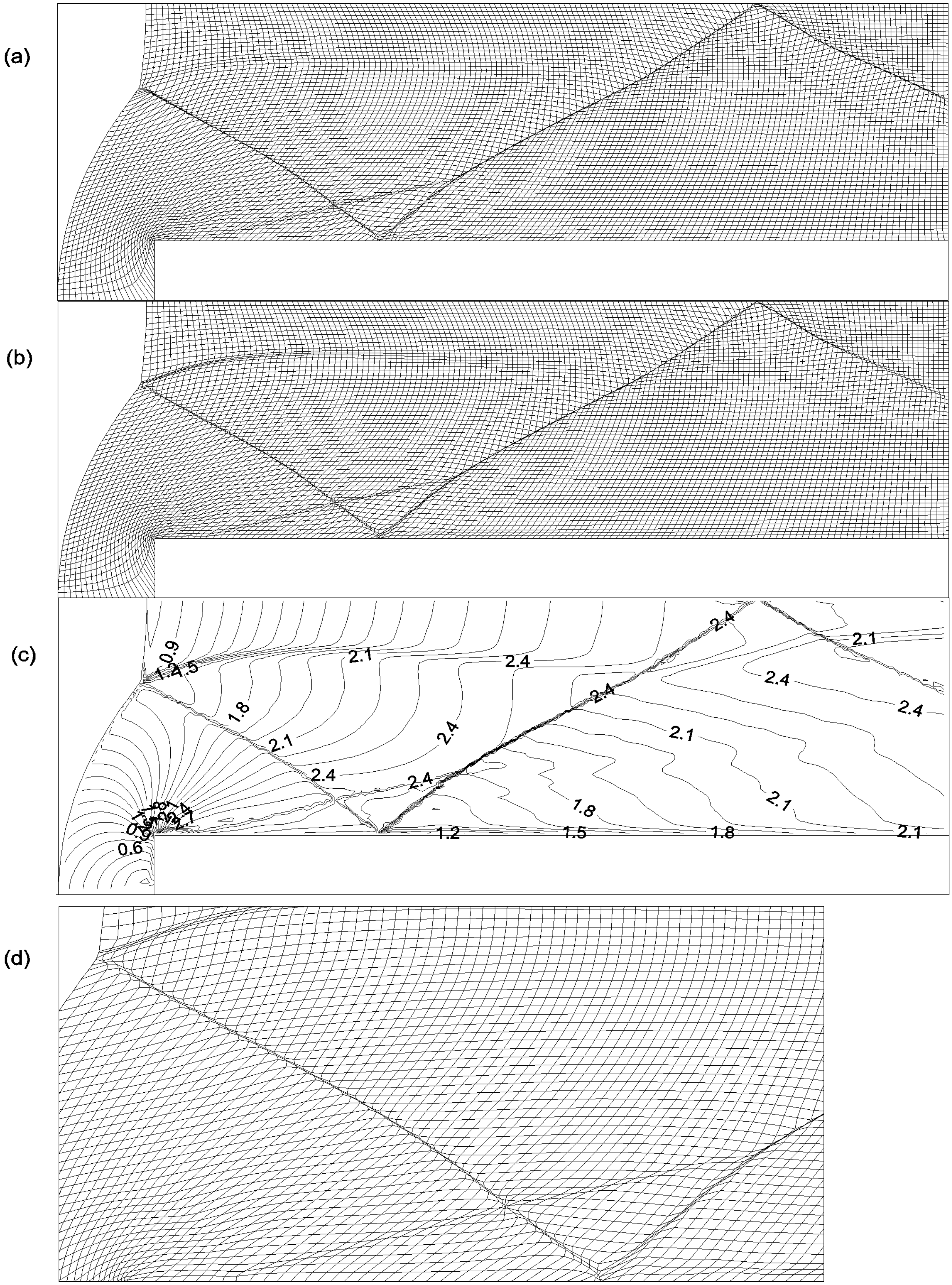

Fig.7. Adaptive 150×60 grids at t=4 applied together with the shock-fitting procedure when adapting to ρ (a) and $|V|$ (b) as a control function; $|V|$ contours (c) in the second case and fragment of the grid (d).

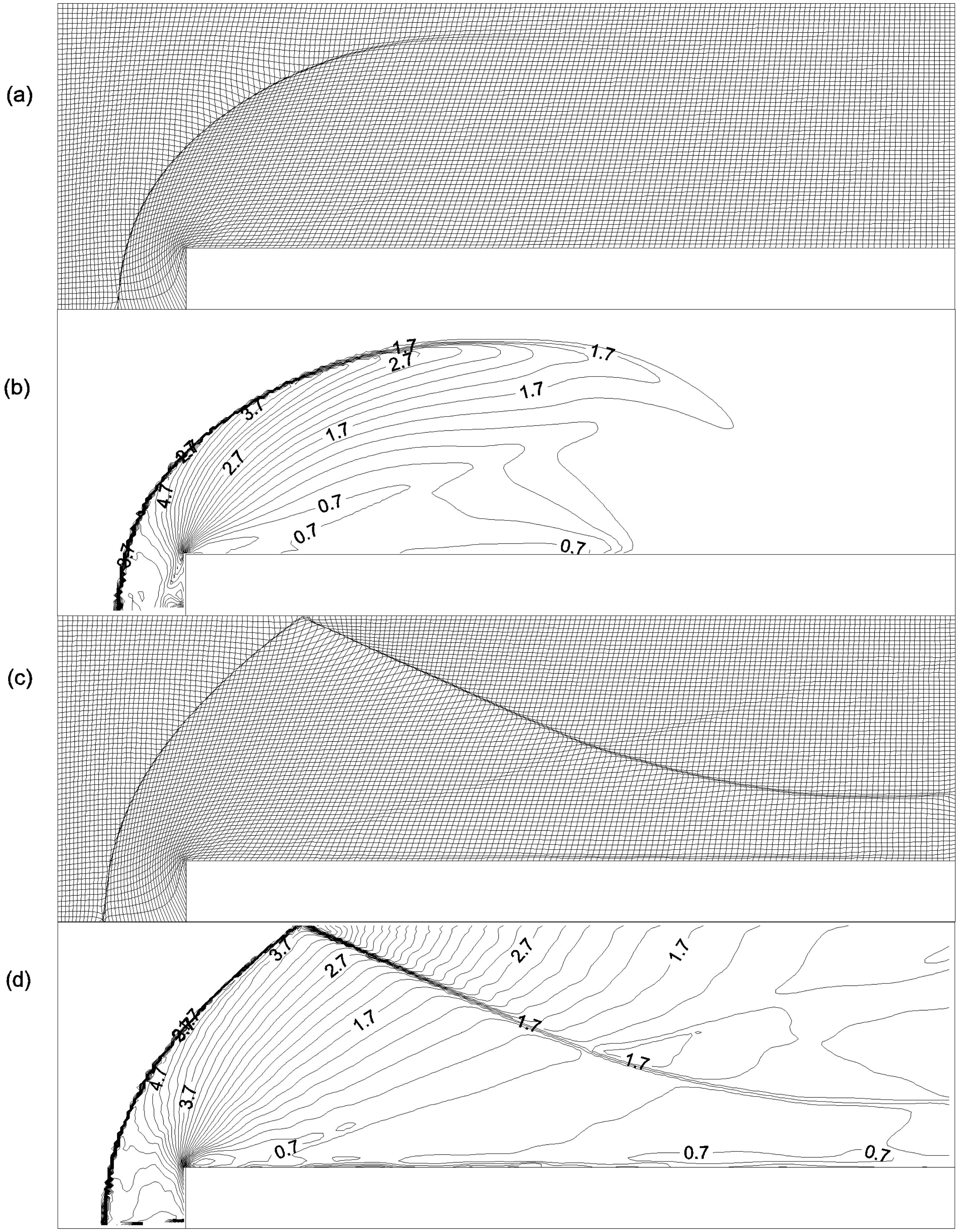

Fig.8. Global adaptation. Adaptive 180×60 grids and density contours at t=0.5 (a,b) and 1 (c,d). Adaptation to the vector-function $(u, v, p, \rho)^T$.

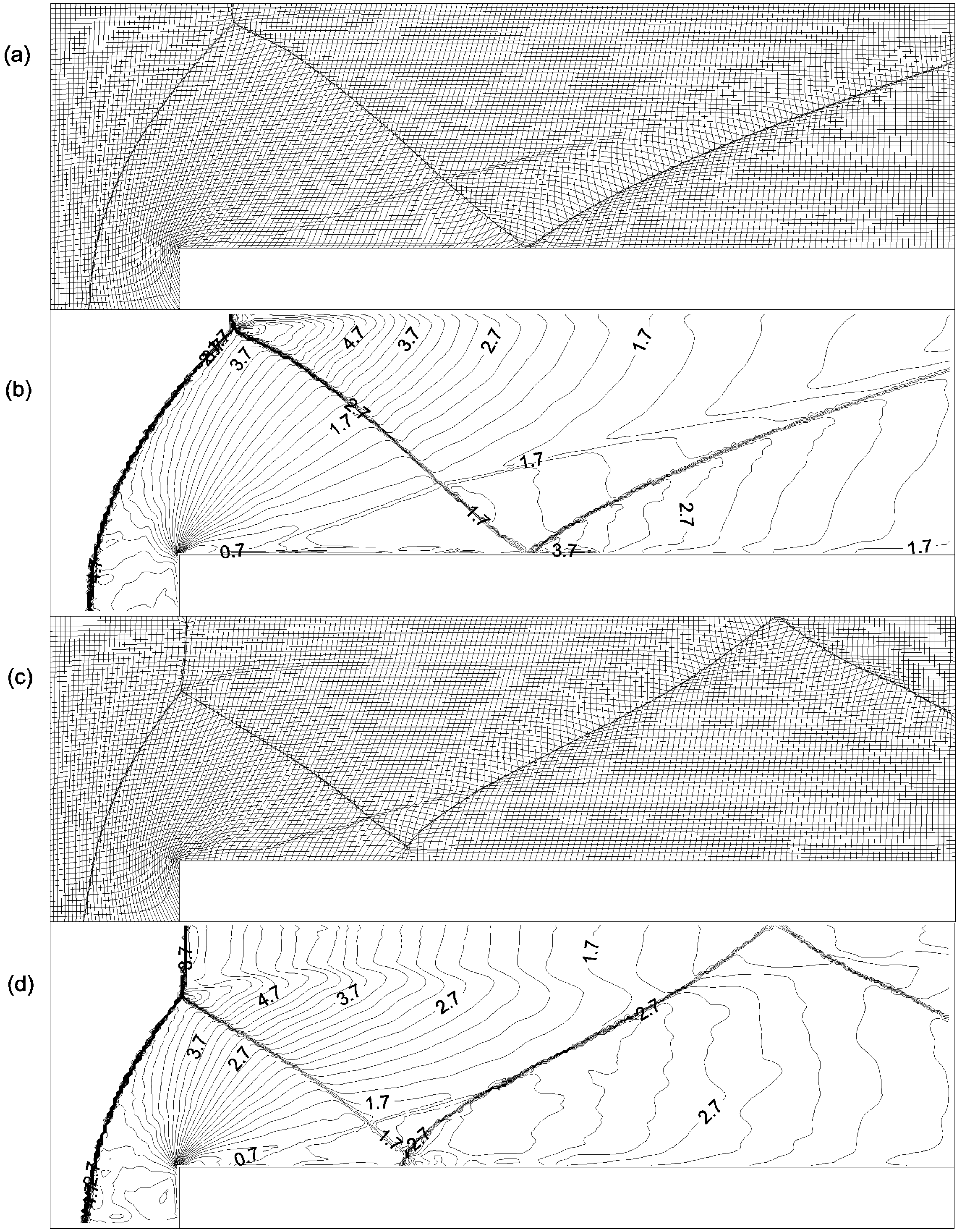

Fig.9. Global adaptation. Adaptive 180×60 grids and density contours at t=2 (a,b) and t=4 (c,d). Adaptation to the vector-function $(u, v, p, \rho)^T$.

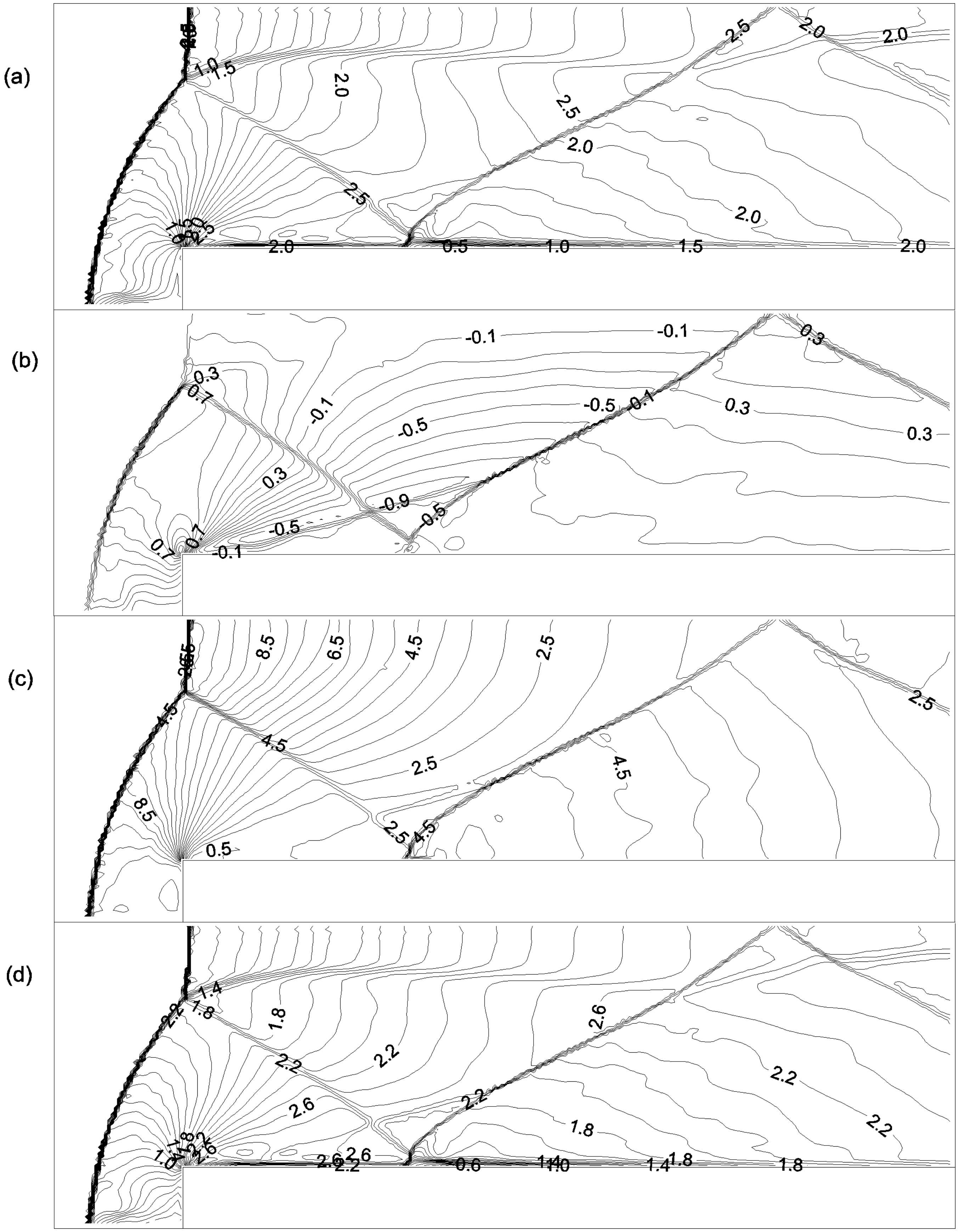

Fig.10. Global adaptation. Velocity components u (a) and v (b), pressure (c) and $|V|$ (d) contours at t=4. Adaptation to the vector-function $(u, v, p, \rho)^T$.

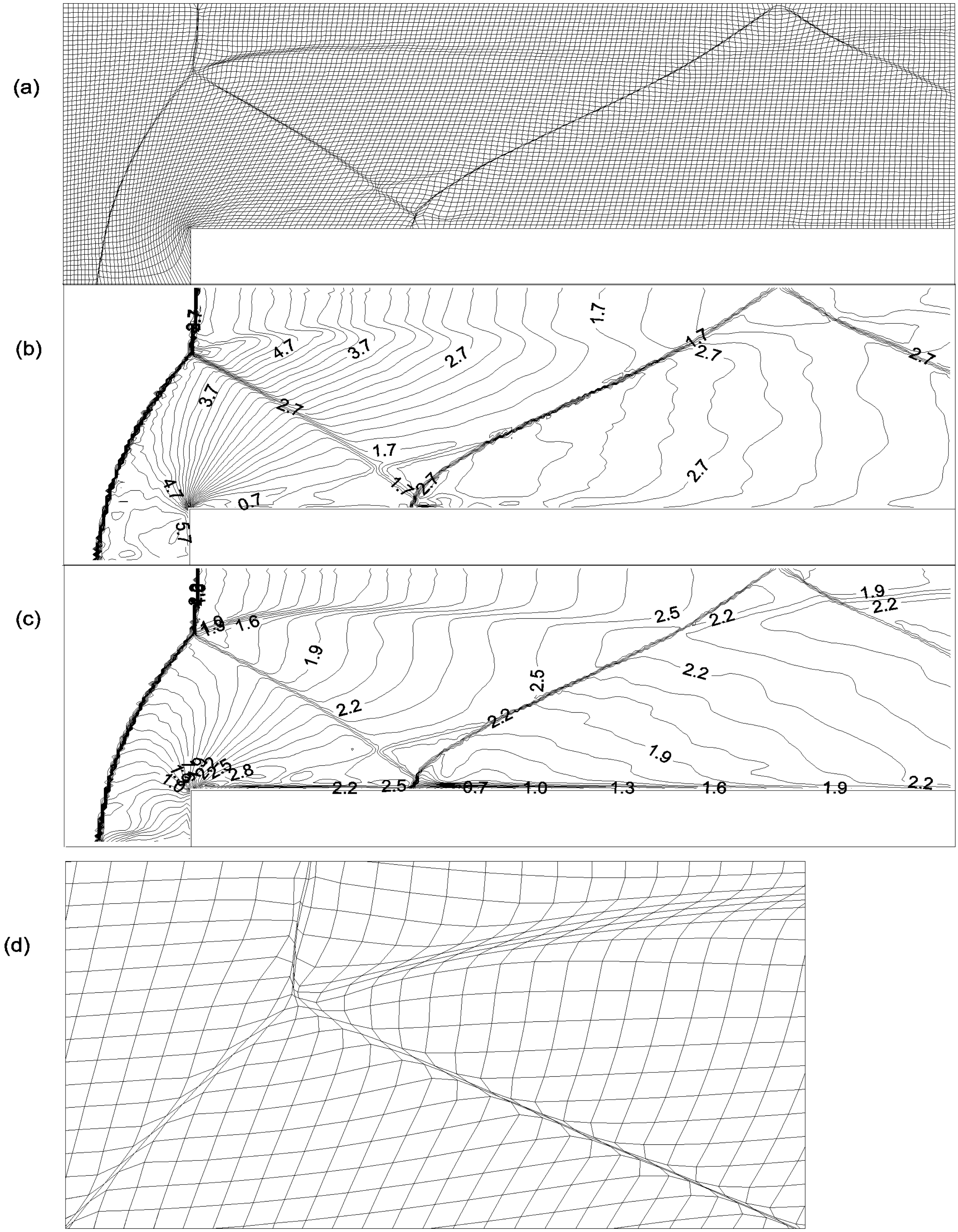

Fig.11. Global adaptation with $|V|$ as an adaptation criterion. Adaptive 180×60 grid (a), density (b) and $|V|$ (c) contours at t=4. Fragment of the grid in the vicinity of the triple point (d).

Performed computations confirm presence of an additional smoothing. When using an adaptive grid the coordinate lines align along the shock waves, forming the stretched cells oriented along the shocks. The amplitude of spurious oscillations is decreased by several factors of ten in comparison with the results obtained on the rectangular grid, see Fig.4d and Figs.9d,11b.

8. Concluding remarks

Results of computations presented in this paper show that the method of adaptive-harmonic grid generation can be successfully applied to nonstationary problems of gas dynamics.

The main conclusion is the following. In nonstationary problems it is necessary to use special procedures in adaptation algorithm to prevent grid folding. Our experience has shown that in the case of high compression of grid lines, the property to produce grids free of folding is the most important. The present method is based on minimization of the functional with infinite barrier at the boundary of the set of grids with all convex cells and this guarantees unfolded grid generation during computations at every time step.

Another important question discussed in many papers is the choice of adaptation criterion. Our experience has shown that it is better to start with common criterion, i.e., to perform adaptation to the vector-function of all dependent variables $(u, v, p, \rho)^T$. Then we can adjust coefficients of adaptation c_i separately to achieve better resolution of singularities.

Grid is clustered in such a way that we can obtain quite a satisfactory resolution of the shocks with not a great number of grid nodes. For the above test the accuracy on the adaptive grid is higher, due to decreasing the thickness of shocks, than on the rectangular grid with greater number of nodes by a factor of 7.1. At the same time in case of adaptation it is required less run time by a factor of 1.5. Note the method is not quite automatic. Success of adaptation depends on the coefficient of adaptation c_a to be selected during computations.

As in the case of a uniform grid, here the shock wave is smeared within 3 cells and the grid with strong clustering looks like a set of blocks with the boundaries defined automatically as the lines of grid nodes clustering. Inside every "block" the grid is quasiuniform and the solution is smooth. This approach may be considered as a further development of the moving block technology presented in the monograph [16], where the flow domain is divided into blocks with boundaries being the shock waves or contact discontinuities.

Advantage of the moving adaptive grid technology is that such a "cutting" is performed automatically. In real-world computations we can obtain such a complex nonstationary configuration of the blocks that even if we would be able to capture their boundaries "by hand", we will have serious problems in attempt to join those blocks. In complex situations it is necessary to adjust the parameters of the algorithm to provide both unfolded grid generation at every time step and smooth variation of the grid control parameters. For example, at Mach stem emerging in the vicinity of the triple point strongly distorted cells appear up to their degeneration. Hence, in this subdomain we shall not specify too large value of c_a.

Formally moving adaptive-harmonic grid generation method can be extended to the 3-D case [2], however, it is necessary to perform additional theoretical and experimental investigations with the purpose to create algorithms with immanent guarantee to produce grids free of folding.

References

1. Handbook of Grid Generation. J.F. Thompson, B.K. Soni and N.P. Weatherill eds. (CRC Press, Boca Raton, Fl, 1999).

2. Ivanenko S.A. Harmonic mappings, Chapt. 8 in Handbook of Grid Generation. (CRC Press, Boca Raton, Fl, 1999).

3. D.S. McRay, Kelly R. Laflin. Dynamic Grid Adaptation and Grid Quality, Chapt. 34 in Handbook of Grid Generation. (CRC Press, Boca Raton, Fl, 1999).

4. J.F.Thompson. A survey of dynamically-adaptive grids in the numerical solution of partial differential equations, Appl. Numer. Maths, 1985, Vol.1, 3-27.

5. C.L. Ingram, D.S. McRay. Extension of a dynamic solution - adaptive grid algorithm and solver to general structured milti-block configurations, AIAA 96-0294, AIAA 34th Aerospace Science Meeting, Reno, NV, Jan. 1996.

6. P.A.Zegeling. Moving Grid Techniques, Chapt. 37 in Handbook of Grid Generation. (CRC Press, Boca Raton, Fl, 1999).

7. Liu F, Ji S, Liao G, An adaptive grid method and its application to steady Euler flow calculations, SIAM J. Sci. Comput. 1998; 20: 811-825.

8. B.N. Azarenok, Realization of a second-order Godunov's scheme, Computer Methods in Applied Mechanics and Engineering, 2000, 189(3), pp 1031-1052.

9. V.P. Kolgan, Application of the pinciple of minimal derivatives towards constructing finite difference schemes to compute discontinuous solutions of fluid dynamics, TSAGI Scientific Notes, 3 (1972), N 6, pp. 68-77.

10. B.N. Azarenok and S.A. Ivanenko, Application of Adaptive Grids in Numerical Analysis of Time-Dependent Problems in Gas Dynamics, Comput. Maths. Math. Phys., 2000, 40(9), pp 1330-1349.

11. Charakhch'yan AA, Ivanenko SA. A variational form of the Winslow grid generator, J.Comput.Phys. 1997; 136: 385-398.

12. Liseikin VD. Grid Generation Methods, Springer-Verlag: New York, 1999.

13. P. Colella and P.R. Woodward, The piecewise parabolic method (PMM) for gas-dynamical simulations, J.Comput.Phys., 54 (1984), pp. 174-201.

14. P. Colella and P.R. Woodward, The numerical simulation of two-dimansional fluid flow with strong shocks, J.Comput.Phys., 54 (1984), pp. 115-173.

15. P.R. Woodward and P. Colella, "Lectures notes in physics No. 141", 1981. Springer-Verlag, New York/Berlin. P.434.

16. S. K. Godunov (Ed.), et al, Numerical solution of multi-dimensional problems in gas dynamics, Nauka, Moscow, 1976, (In Russian); S. K. Godunov, A. V. Zabrodin, M. Ya. Ivanov, A. N. Kraiko and G. P. Prokopov, Résolution Numérique des Problémes Multidimensionnels de la Dynamique des Gaz, Mir, Moscou, 1979 (In French).

17. S.A.Ivanenko. Adaptive - harmonic grid generation and its application for numerical solution of the problems with boundary and interior layers, Comput. Maths. Math. Phys. 35(10), 1203-1220, 1995.