Proceedings of ALGORITMY 2002
Conference on Scientific Computing, pp. 78–85

ANISOTROPIC MESH ADAPTATION FOR TRANSONIC AND SUPERSONIC FLOW SIMULATION*

VÍT DOLEJŠÍ AND JIŘÍ FELCMAN†

Abstract. We present an efficient tool for the numerical simulation of high speed flows in two and threedimensional domains. The space discretization is carried out with the finite volume method on unstructured triangular and tetrahedral meshes. In order to achieve sufficiently accurate capturing of shock waves, we have applied the anisotropic mesh adaptation technique which seems to be very suitable for problems with complicated geometries. The anisotropic mesh adaptation is a universal adaptive method which can be used without any modification for the numerical solution of arbitrary boundary value problem. Numerical examples of supersonic and transonic flows are presented.

Key words. anisotropic mesh adaptation, Euler equations, supersonic and transonic flow

AMS subject classifications. 76M12, 65M50, 65N50

1. Introduction. The investigation and numerical simulation of high speed flows play an important role in the *computational fluid dynamics.* The motion of compressible liquid is described by conservation laws, namely the conservation of mass, momentum and energy. Neglecting viscous terms, which are small in comparison with convective ones for transonic or supersonic flows, we obtain the hyperbolic system of the *Euler equations.*

Our goal is to develop a sufficiently accurate and robust method for the numerical solution of the Euler equations. It is a known fact that the solution of hyperbolic equations may be discontinuous even for smooth initial data. The discontinuities of the solution represent physical phenomena which are called the *shock waves.* The accurate capturing of shock waves requires the application of an adaptive method in order to avoid enormous requirements for the memory and CPU-time.

In this paper we combine the *finite volume method*, which is sufficiently robust for subsonic, transonic and supersonic flow regimes, with the *anisotropic mesh adaptation* technique, which is very efficient for problems with complicated geometries. The context of the paper is the following. We present the system of the Euler equations in Section 2 and their discretization by the finite volume method in Section 3. The basic ideas of the anisotropic mesh adaptation method and its application for the computational fluid dynamics is given in Section 4. In the last section we present two examples of inviscid flow simulation in two and three dimensional geometries.

2. Governing equations. The system of the Euler equations describing the motion of an inviscid compressible fluid in a bounded domain $\Omega \subset \mathbb{R}^d$, $d = 2,3$ and time interval $(0,\bar{T})$ can be written in the form

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{d} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = 0 \quad \text{in } \Omega \times (0,\bar{T}), \tag{2.1}$$

*This work was supported under the Grant No. 275/2001/B–MAT/MFF of the Grant Agency of the Charles University Prague and the Grant No. MSM 113200007.

†Charles University Prague, Faculty of Mathematics and Physics, Sokolovska 83, 18675 Prague, Czech Republic({dolejsi,felcman}@karlin.mff.cuni.cz).

where

$$\boldsymbol{w} = (\rho,\, \rho v_1,\, \ldots,\, \rho v_d,\, e)^{\mathrm{T}}, \quad \boldsymbol{w} = \boldsymbol{w}(\boldsymbol{x},t), \quad \boldsymbol{x} \in \Omega, \quad t \in (0,\bar{T}),$$
$$\boldsymbol{f}_s(\boldsymbol{w}) = (\rho v_s,\, \rho v_s v_1 + \delta_{s1} p,\, \ldots, \rho v_s v_d + \delta_{sd} p,\, (e+p)\, v_s)^{\mathrm{T}},\ s = 1,\ldots,d.$$

In order to close the system we add the state equation for perfect gas

$$p = (\gamma - 1)\,(e - \rho|\boldsymbol{v}|^2/2). \tag{2.2}$$

We use the standard notation: t – time, $\boldsymbol{x} = (x_1,\ldots,x_d)$ – Cartesian coordinates, ρ – density, p – pressure, e – total energy, $\boldsymbol{v} = (v_1,\ldots,v_d)$ – velocity, δ_{ij} – Kronecker delta, $\gamma > 1$ – the Poisson adiabatic constant.

The system (2.1) – (2.2) is equipped with the initial condition

$$\boldsymbol{w}(\boldsymbol{x},0) = \boldsymbol{w}^0(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Omega, \tag{2.3}$$

where $\boldsymbol{w}^0$ is a given function, and a set of boundary conditions. For their description see, e. g., [8], [12].

3. Discretization. Let the domain Ω be approximated by a polygonal domain Ω_h. We construct a triangulation (for $d = 2$) or tetrahedrization (for $d = 3$) $\mathcal{T}_h$ of Ω_h with the usual properties (in the sense of the finite element method) which defines a *finite volume* partition $\mathcal{T}_h = \{T_i\}_{i\in J}$ of the closure $\overline{\Omega}_h$ into a finite number of closed volumes (triangles or tetrahedra) T_i; J is a suitable index set. The boundary ∂T_i can be expressed in the form $\partial T_i = \bigcup_{j\in S(i)} \Gamma_{ij}$, where Γ_{ij} is either the common boundary of T_i and T_j or $\Gamma_{ij} \subset \partial\Omega_h$. We set $|T_i| = d$-dimensional Lebesgue measure of T_i, $\boldsymbol{n}_{ij} =$ unit outer normal to ∂T_i on Γ_{ij}, $d(\Gamma_{ij}) = (d-1)$-dimensional Lebesgue measure of Γ_{ij}, $S(i)$ is a suitable index set. The time discretization of (2.1) is carried out with the use of a partition $0 = t_0 < t_1 < t_2 < \ldots$ of the time interval $[0,T]$. We set $\tau_k = t_{k+1} - t_k$. We use a cell centered finite volume method (see [8]), with constant values $\boldsymbol{w}_i^k$ representing the volume average of the vector of conserved quantities $\boldsymbol{w}(\cdot,t_k)$ on T_i at time t_k. Integrating (2.1) over the set $T_i \times (t_k, t_{k+1})$ and using the Green theorem we obtain the following explicit scheme:

$$\boldsymbol{w}_i^{k+1} = \boldsymbol{w}_i^k - \frac{\tau_k}{|T_i|} \sum_{j\in S(i)} \boldsymbol{H}(\boldsymbol{w}_i^k, \boldsymbol{w}_j^k, \boldsymbol{n}_{ij})\, d(\Gamma_{ij}), \quad i \in J,\ k = 0,1,\ldots. \tag{3.1}$$

The term $\boldsymbol{H}(\boldsymbol{w}_i^k, \boldsymbol{w}_j^k, \boldsymbol{n}_{ij})$ is the so-called *numerical flux* which approximates the flux $\sum_{s=1}^d n_s \boldsymbol{f}_s(\boldsymbol{w})$ on Γ_{ij} in the direction $\boldsymbol{n}_{ij} = (n_1,\ldots,n_d)$. The values $\boldsymbol{w}_i^k$ and $\boldsymbol{w}_j^k$ are the values of the function $\boldsymbol{w}(\cdot,t_k)$ approximated from the volumes T_i and T_j on the face Γ_{ij}, respectively. We evaluate the numerical flux in (3.1) by the the *Osher–Solomon scheme* [14] and/or the exact solver for the Euler equations [16]. For more detail see [9], [11].

4. Anisotropic mesh adaptation. In this section we present a brief description of the anisotropic mesh adaptation method for $d-$dimensional ($d = 2,3$) problems. This approach is based on the control of the *interpolation error* of the solution of the considered problem and therefore it can be used without any modification for the numerical solution of wide range problems of physics and engineering described by partial differential equations. As the mesh adaptation criterion is independent on the form of the governing equations (2.1), we have to choose only a significant scalar quantity which is characteristic for a considered problem.

In this paper, we have chosen the density as the significant quantity. For viscous flow problem (see [7]), it is better to chose the Mach number. In the case of heat conduction problem (see [2]), we used a temperature. The complete description of two and three dimensional version can be found in [6] and [10], respectively. Similar adaptive strategies can be found in [1], [3], [13], [15], [17] and in the references therein.

4.1. Necessary condition. Let u be the significant scalar flow quantity in which we measure the discretization error. For a given $t_k \in [0,T]$, u^k means the function $\boldsymbol{x} \in \Omega \to u^k(\boldsymbol{x}) = u(\boldsymbol{x}, t_k)$. The function u^k is supposed to be an element of the space of trial functions U which has to be specified. Let us consider the space of piecewise linear discontinuous functions

$$U_h = \left\{ u_h \in L^1(\Omega_h); u_h\big|_T \in P_1(T)\ \forall T \in \mathcal{T}_h \right\}, \tag{4.1}$$

where $P_1(T)$ denotes the space of all linear polynomials on T. Let $u_h^k \in U_h$ be the approximation of u^k computed by a numerical method. To measure how close the approximate solution u_h is to the exact solution u we define the *discretization error* as

$$e_h \equiv \|u - u_h\|_X, \tag{4.2}$$

where $\|\cdot\|_X$ is a suitable norm such that the expression $\|v\|_X$ has a sense $\forall v \in U \cup U_h$. The ultimate goal of a numerical method is to compute the approximate solution satisfying

$$e_h \le \omega \tag{4.3}$$

where ω is is a prescribed tolerance.

In what follows we present the necessary condition for the relation (4.3) and show how its satisfaction can be used for the construction of adaptive meshes. Let Π_h: $U \to U_h$ be an operator such that

$$\|w - \Pi_h w\|_X = \min_{w_h \in U_h} \|w - w_h\|_X \quad \forall w \in U. \tag{4.4}$$

It is evident that

$$\|u - \Pi_h u\|_X \le e_h. \tag{4.5}$$

This is the crucial point of the proposed adaptation strategy: Any numerical method computes the approximate solution u_h with the discretization error which is bounded from below by $\|u - \Pi_h u\|_X$. Then the *necessary condition* to fulfill (4.3) is

$$\|u - \Pi_h u\|_X \le \omega. \tag{4.6}$$

In order to give the numerical method the chance to satisfy (4.3), the mesh on which (4.6) holds is needed. It is evident that the magnitude of $\|u - \Pi_h u\|_X$ strongly depends on the choice of $\mathcal{T}_h$ and it is completely independent of the numerical solution u_h itself. We will modify the given mesh $\mathcal{T}_h$ in order to satisfy the necessary condition (or its modification). Moreover we require that $\#\mathcal{T}_h$ is as small as possible (in order to save the memory and CPU-time), where $\#\mathcal{T}_h$ denotes the number of elements of $\mathcal{T}_h$.

4.2. Interpolation operator. We have introduced the operator Π_h in (4.4) but the construction of $\Pi_h u$, which depends on the choice of the norm $\|\cdot\|_X$, is difficult. Therefore we introduce a new simpler operator which will be used for further consideration. Let $\boldsymbol{x}_i \in T_i$ be a given point $\forall T_i \in \mathcal{T}_h$ and $u \in C^1(\Omega)$ then we define the *interpolation operator* $r_h : U \to U_h$ such that

$$\begin{aligned} r_h u(\boldsymbol{x}_i) &= u(\boldsymbol{x}_i) \qquad \forall T_i \in \mathcal{T}_h, \\ \nabla r_h u(\boldsymbol{x}_i) &= \nabla u(\boldsymbol{x}_i) \quad \forall T_i \in \mathcal{T}_h. \end{aligned} \tag{4.7}$$

The definition of r_h is unique for a given set of points $\{\boldsymbol{x}_i\}$ and $r_h u \in U_h$ can be simply constructed for given $u \in C^1(\Omega)$. As $r_h u \in U_h$ we have from (4.4)

$$\|w - \Pi_h w\|_X \le \|w - r_h w\|_X \qquad \forall w \in U. \tag{4.8}$$

Therefore the condition

$$\|u - r_h u\|_X \le \omega \tag{4.9}$$

is stronger than the necessary condition (4.6) but as it is easily computable and we use it for further consideration.

4.3. Mathematical background of AMA. Our aim is to adapt a given mesh $\mathcal{T}_h$ so that after adaptation it satisfies the stronger condition (4.9) and $\#\mathcal{T}_h$ is minimal. Let us suppose that the dependence of $\|u - r_h u\|_X$ on $\#\mathcal{T}_h$ is monotone. Then the inverse problem can be formulated: Adapt a given mesh $\mathcal{T}_h$ in such a way that the interpolation error $\|u - r_h u\|_X$ is minimal for a fixed $\#\mathcal{T}_h$.

The basic idea is the following: In order to minimize the interpolation error for a given $\#\mathcal{T}_h$, the *interpolation error function* defined by

$$E_I(\boldsymbol{x}) \equiv |u(\boldsymbol{x}) - r_h u(\boldsymbol{x})| \tag{4.10}$$

should be equidistantly distributed over the whole computational domain Ω, i.e.

$$E_I(\boldsymbol{x}) \approx C \qquad \forall \boldsymbol{x} \in \Omega, \tag{4.11}$$

where $C > 0$ is a constant. If $u \in C^2(\Omega)$ then using a Taylor series expansion at a point $\boldsymbol{x}_0 \in \Omega$ we have

$$u(\boldsymbol{x}) - r_h u(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^{\mathrm{T}} I\!\!H(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) + o(|\boldsymbol{x} - \boldsymbol{x}_0|^2), \tag{4.12}$$

where

$$I\!\!H(\boldsymbol{x}) \equiv \left\{ \frac{\partial^2 u(\boldsymbol{x})}{\partial x_i\, \partial x_j} \right\}_{i,j=1}^{d} \tag{4.13}$$

is the Hesse matrix of the function u.

In order to minimize the interpolation error function E_I, we introduce the discrete version of the condition (4.11). We consider the interpolation error function E_I over edges for practical reasons. Let $\boldsymbol{e}$ be an edge of $\mathcal{T}_h$ (connecting two nodes of $\mathcal{T}_h$), $\ell_{\boldsymbol{e}}$ notes the Euclidean length of $\boldsymbol{e}$ and let $\boldsymbol{x}_{\boldsymbol{e}}$ be a centre of $\boldsymbol{e}$. We approximate $E_I|_{\boldsymbol{e}}$ by the mean value of E_I over $\boldsymbol{e}$. Then the omitting the terms of higher order yields

$$\begin{aligned} E_I\big|_{\boldsymbol{e}} &\approx \frac{1}{\ell_{\boldsymbol{e}}} \int_{\boldsymbol{e}} |u(\boldsymbol{x}) - r_h u(\boldsymbol{x})| \,\mathrm{d}S \;\approx \\ &\approx \frac{1}{\sqrt{2}} \frac{1}{\ell_{\boldsymbol{e}}} \int_{\boldsymbol{e}} |(\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{e}}) I\!\!H(\boldsymbol{x}_{\boldsymbol{e}})(\boldsymbol{x} - \boldsymbol{x}_{\boldsymbol{e}})| \,\mathrm{d}S = \frac{1}{\sqrt{24}} \left| \vec{\boldsymbol{e}}^{\mathrm{T}} I\!\!H(\boldsymbol{x}_{\boldsymbol{e}}) \vec{\boldsymbol{e}} \right|, \end{aligned} \tag{4.14}$$

where $\vec{e} \in \mathbb{R}^d$ is the vector parallel with the edge $\boldsymbol{e}$ and with the Euclidean norm equal to $\ell_{\boldsymbol{e}}$ We define the norm of the edge $\boldsymbol{e}$ corresponding to the matrix $\mathbb{H}$ by

$$\|\boldsymbol{e}\|_{\mathbb{H}} \equiv \left(\left|\vec{e}^{\mathrm{T}}\mathbb{H}(\boldsymbol{x}_{\boldsymbol{e}})\vec{e}\right|\right)^{1/2}. \tag{4.15}$$

We have from (4.14) and (4.15), that the interpolation error function is uniformly distributed over the mesh $\mathcal{T}_h$ if

$$E_I|_{\boldsymbol{e}} \approx \frac{1}{\sqrt{24}}\|\boldsymbol{e}\|^2_{\mathbb{H}} \approx \omega \qquad \text{for any edge } \boldsymbol{e} \text{ of } \mathcal{T}_h, \tag{4.16}$$

where $\omega > 0$ is the given accuracy. With the aid of (4.16) we define the following:

DEFINITION 4.1. *The mesh* $\mathcal{T}_h$ *is* optimal *iff*

$$\|\boldsymbol{e}_k\|_{\mathbb{H}_k} = \epsilon \quad \forall \boldsymbol{e}_k \in \mathcal{T}_h, \qquad \boldsymbol{e}_k \textit{ edge of } \mathcal{T}_h, \quad \mathbb{H}_k \equiv \mathbb{H}(\boldsymbol{x}_{\boldsymbol{e}_k}), \tag{4.17}$$

where $\epsilon > 0$ *is a given constant which plays a role of accuracy.*

In order to measure how the mesh is close to the optimal one we define the *quality parameter* of $\mathcal{T}_h$ by

$$Q_{\mathcal{T}_h} \equiv \frac{1}{\#\mathcal{E}} \sum_{\boldsymbol{e}_k \in \mathcal{T}_h} \left(\|\boldsymbol{e}_k\|_{\mathbb{H}_k} - \epsilon\right)^2, \tag{4.18}$$

where sum is taken over all edges of $\mathcal{T}_h$ and $\#\mathcal{E}$ is their number. The quality parameter is always nonnegative and it is equal to zero if the mesh is optimal in the sense of the Definition 4.1. Therefore to improve the quality of the mesh we modify $\mathcal{T}_h$ in order to decrease $Q_{\mathcal{T}_h}$. The mesh adaptation is performed by an iterative process consisting of the combination of the following local operations:

- **Case** $d = 2$**:** removing an edge, inserting a node in the centre of an edge, swapping a diagonal of quadrilateral formed be two adjacent triangles and moving a node, see [5], [6].
- **Case** $d = 3$**:** removing an edge, inserting a node in the centre of an edge, swapping a face for boundary tetrahedra, swapping an edge for internal tetrahedra and moving a node, see [10].

In order to evaluated the second order derivatives in (4.13), we approximated u by the numerical solution u_h and use some smoothing technique, see [6]. Then the computational process consists of multiplicative application of the finite volume solver in combination with the anisotropic mesh adaptation. We stop the computational process when two successive meshes are almost identical.

5. Numerical examples.

In this section we present two examples of compressible inviscid flow. The finite volume method was used as an iterative time marching process with $k \to \infty$ for obtaining the steady state solution.

5.1. 2D supersonic scramjet inlet.

The first case consists of an internal supersonic flow at Mach number ($= |\boldsymbol{v}|/(\gamma p/\rho)^{1/2}$) equal to 3 in a scramjet inlet, Figure 5.1 shows the geometry of the problem. The supersonic inlet and several obstacles with sharp angles give the solution with a few shock waves, see [4]. Although the configuration is symmetric, the nonsymmetric mesh of the whole domain has been computed to observe if the solution remains symmetric or not.

Using five automatic mesh adaptations we have obtained the final mesh. Figures 5.2 and 5.3 show the final triangular mesh (52285 elements) and the corresponding

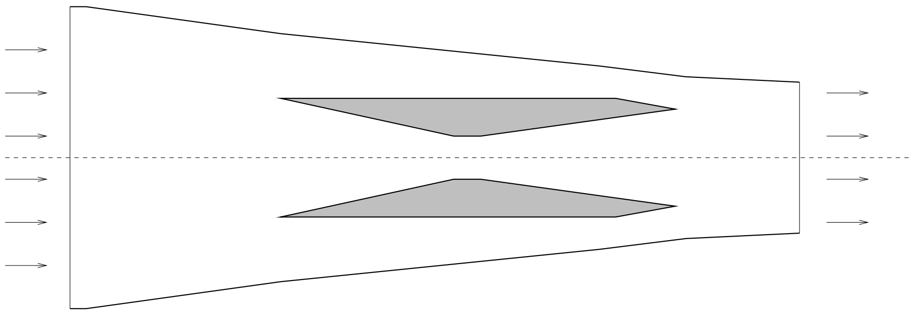

FIG. 5.1. *Geometry of the problem 5.1*

isolines of Mach number, respectively. We see rather complicated geometry of shock waves and it is interesting to notice that there are no oscillations in the solution and that the finite volume solver is robust and accurate even for such meshes with very anisotropic triangles. Moreover the numerical solution stays symmetric.

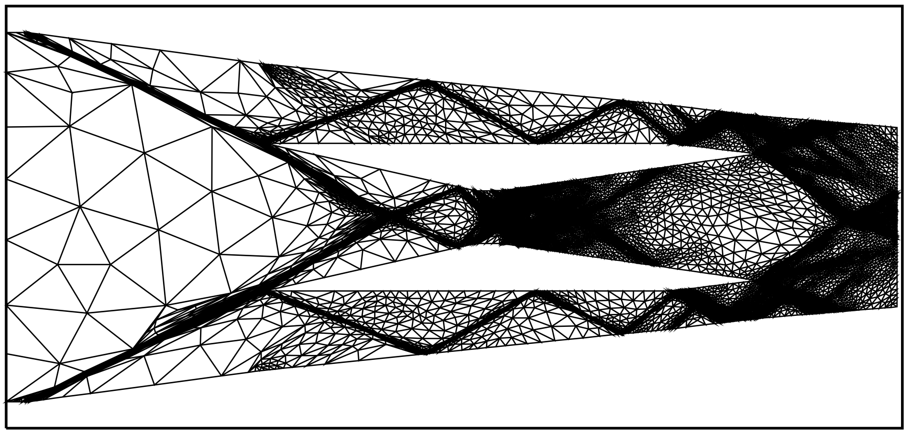

FIG. 5.2. *Final triangulation for a supersonic scramjet inlet*

5.2. 3D transonic flow through 3D GAMM channel. The three-dimensional transonic inviscid flow through the channel (with 25 % spherical bump on the lower wall) of air with inlet Mach number $M_{\text{inlet}} = 0.67$ was solved. The size of the channel is $x_1 \in [0, 2]$, $x_2 \in [0, 1.5]$ and $x_3 \in [0, 1]$. The direction of the flow is parallel with x_1 axis and goes from left to right. Figures 5.4 and 5.5 show the final tetrahedral mesh (with 8786 elements) achieved after five mesh adaptations and the corresponding isolines of Mach number, respectively.

Despite the small number of elements used in the computation of the 3D test channel flow, the anisotropic mesh adaptation method leads to satisfactory results.

REFERENCES

[1] T. Apel. *Anisotropic Finite Elements: Local Estimates and Applications.* TU Chemnitz, Preprint SFB393/99-03, 1999.

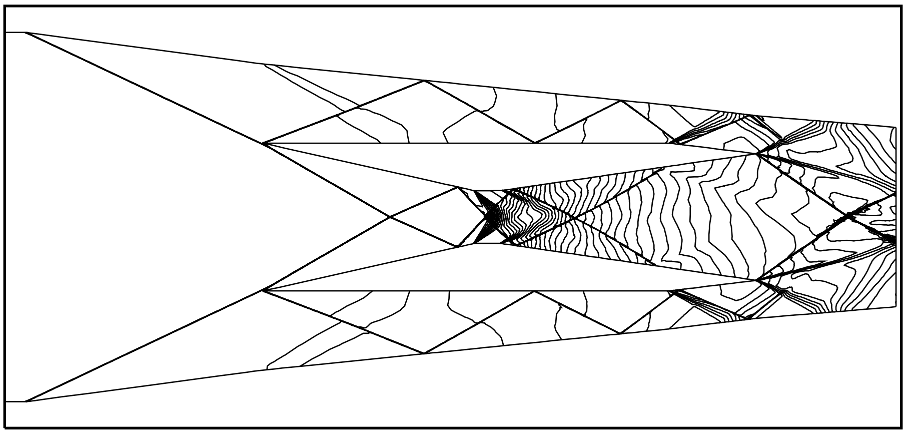

FIG. 5.3. *The corresponding isolines of Mach number for a supersonic scramjet inlet*

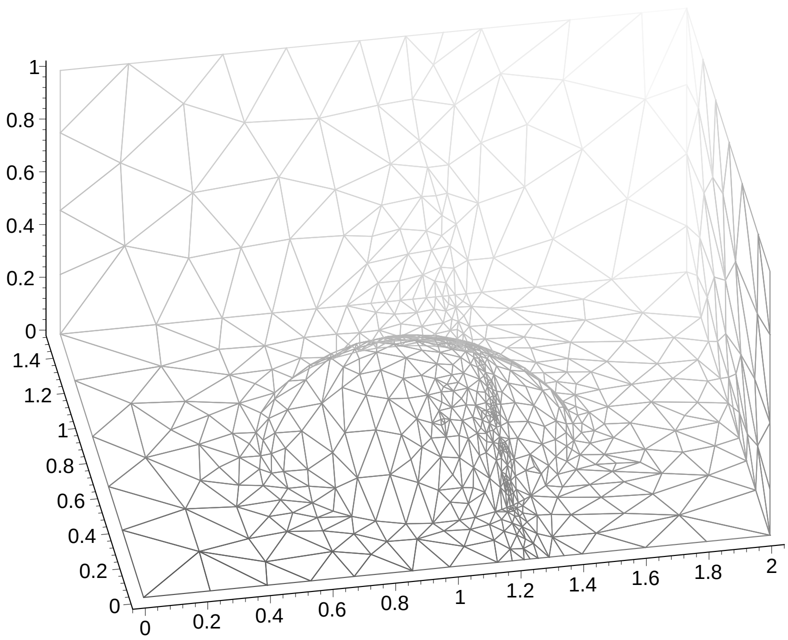

FIG. 5.4. *Final tetrahedrization of 3D GAMM channel*

[2] M. Breuss, V. Dolejší, and A. Meister. On an adaptive method for heat conduction problems with boundary layers. *Computing and Visualization in Science*, (submitted).
[3] M. J. Castro Díaz, H. Borouchaki, P. L. George, F. Hecht, and B. Mohammadi. Anisotropic Adaptive Mesch Generation in Two Dimensions for CFD. In *Proceeding of ECCOMAS96 – Computational Fluid Dynamics, Paris*, pages 181–186, 1996.
[4] M. J. Castro Díaz, F. Hecht, and B. Mohammadi. New progress in anisotropic grid adaptation for inviscid and viscous flows simulations. In *Proceedings of the 4th Annual International Meshing Roundtable*. Sandia National Laboratories, 1995.
[5] V. Dolejší. Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Computing and Visualization in Science*, 1(3):165–178, 1998.

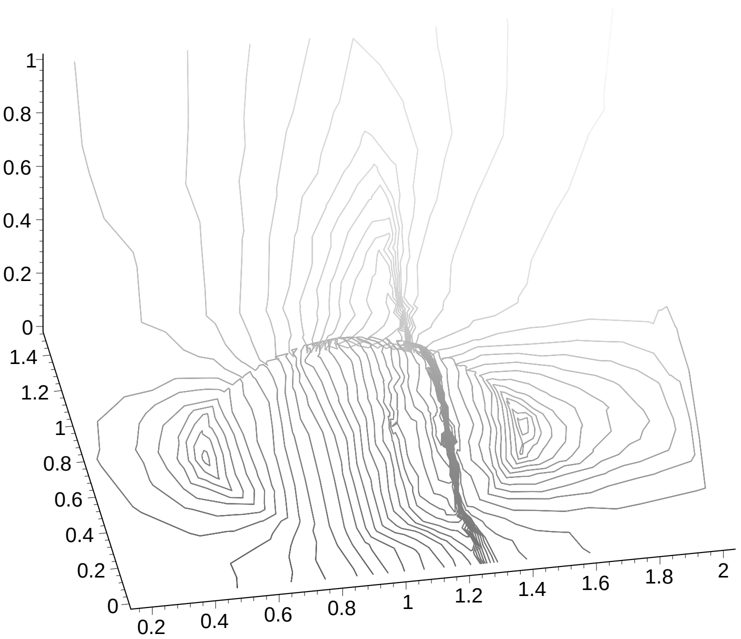

FIG. 5.5. *The corresponding isolines of Mach number for 3D GAMM channel*

[6] V. Dolejší. Anisotropic mesh adaptation technique for viscous flow simulation. *East-West Journal of Numerical Mathematics*, 9(1):1–24, 2001.
[7] V. Dolejší and J. Felcman. Anisotropic mesh adaptation and its application for scalar diffusion equations. *Numerical Methods for Partial Differential Equations*, (submitted).
[8] M. Feistauer. *Mathematical Methods in Fluid Dynamics.* Longman Scientific & Technical, Harlow, 1993.
[9] M. Feistauer and J. Felcman. Convection-Diffusion Problems and Compressible Navier–Stokes Equations. In J. R. Whitemen, editor, *The Mathematics of Finite Elements and Applications*, pages 175–194. John Wiley & Sons, 1996.
[10] J. Felcman. Grid refinement–alignment in 3D flow computations. *Mathematics and Computers in Simulation*, (submitted).
[11] J. Felcman, V. Dolejší, and M. Feistauer. Adaptive Finite Volume Method for the Numerical Solution of the Compressible Euler Equations. In J. Périaux S. Wagner, E. H. Hirschel and R. Piva, editors, *Computational Fluid Dynamics '94*, pages 894–901. John Wiley and Sons, Stuttgart, 1994. ISBN 0-471-950637.
[12] E. Godlewski, P. A. Raviart: Hyperbolic systems of conservation laws. Mathématiques & Applications (Paris). 3-4. Paris: Ellipses, 1991.
[13] W. G. Habashi, M. Fortin, D. Ait ali Yahia, S. Boivin, Y.Bourgault, J. Dompierre, M. P. Robichaud, A. Tam, and M. G. Vallet. Anisotropic mesh optimization: Towards a solver-independent and mesh-independent CFD. In *Compressible Fluid Dynamics, VKI Lecture Seies 1996-06.* von Karman Institute for Fluid Dynamics - Concordia University, 1996.
[14] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comp.*, 38:339–374, 1982.
[15] R. B. Simpson. Anisotropic mesh transformations and optimal error control. *Applied Numer. Math.*, 14:183–198, 1994.
[16] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer-Verlag, 1997.
[17] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Internat. J. Numer. Methods Engrg.*, 24:337–357, 1987.