A MULTI-BLOCK ORTHOGONAL GRID GENERATION USING CAD SYSTEM

Vladimir V. Chudanov[1], Anna E.Aksenova, Valerii A. Pervichko, Alexander G.Churbanov, Irina G.Plotnikova, Vladimir V. Varenkov and Petr N. Vabishchevich

[1]Nuclear Safety Institute, Moscow, Russia, chud@ibrae.ac.ru

ABSTRACT

In this work there is presented a non-traditional approach to generation of block-structured orthogonal grids on the basis of up-to-date CAD systems. The features of the algorithm of constructing orthogonal grids are explained. Examples of constructed grids are provided.

Keywords: mesh generation, computational geometry

1. INTRODUCTION

Generation of grids and their processing become more and more important fields of scientific researches in connection with increasing needs of industry. An improvement of processes of grid generation and processing depends on improvement of mathematical, algorithmic and program concepts.

Grid generation — the process allowing to obtain discrete models of computational domains from the common geometrical configurations. The geometrical configuration can be discretized into grids of two principal types: structured and unstructured.

Structured grids consist of quadrilateral in the 2D or hexahedral in 3D case cells. Connectivity of structured grid is provided by the trivial identification of adjacent nodes with increasing of coordinate indices.

Unstructured grids are characterized by cells of an arbitrary form (usually triangles, quadrilateral in the 2D and tetrahedrons, hexahedrons in the 3D case) and have not any trivial identification of neighbours with increasing of an index. For unstructured grids the set of nodes and the corresponding connectivity matrix are generated to define irregular forms of cells as well as neighbour nodes. In practice sometimes there are used combinations of structured and unstructured grids called as hybrid grids and grids consisting of structured grids with zones of overlapping (referred to as Chimera, composite or patched grids).

From all listed above grids block-structured grids appear to be optimal in sense of compromise between possibilities to handle complex geometrical objects and possibilities to employ a wide variety of developed early programs of numerical analysis. The application of block-structured grids to modeling of complex geometrical configurations consists in representation of the initial complicated domain via a set of more simple subregions — blocks. In this case a grid is generated in each subregion and the link between subregions is controlled according to the connectivity matrix for subregions. As the node connectivity in such grids is defined in a very simple way via indices, it makes possible to employ a lot of well developed and easy checked algorithms. Moreover, block-structured grids indicated some advantages in implementation on parallel computing systems.

The overlapped grid approach has great versatility [1][2][3] and demonstrates both merits and demerits. The primary bottleneck lies here in the necessity to connect components of a grid composition via interpolation that can yield the lack of conservation. Different variants of hyperbolic and elliptic generators are used in this approach.

Experience of numerical simulation shows that quality of computational grids makes an essential impact on accuracy of numerical solutions. In this sense orthogonal or quasi-orthogonal grids are more preferable and provide significant advantages in solution of systems of differential partial equations:

- Transformation of governing equations performed in order to use such grids leads to appearing a small enough number of the additional terms,
- Orthogonal grids provides the highest accuracy of calculations,
- Boundary conditions of various type can be implemented in a very simple way.

Elliptic generators are in common use to construct orthogonal grids (see, for example, [4][5]). The basic difficulties of such an approach are connected with non-standard nonlinear boundary conditions and fulfillment of orthogonality on boundaries of domains where the grid is designed. The primary merit is in the fact that the problem of grid construction is solved in a regular domain.

The alternative approach is connected with searching functions $G \in R^3$ (orthogonal set). For these functions the classical mixed boundary value problem is formulated on the basis of the selfadjoint elliptical equations of second order. However, this well-posed problem is solved in an irregular domain which, as a rule, is curvilinear and multiply connected. Precisely this approach is considered in the present work.

The problem of solution of the mixed boundary value problem in an irregular computational domain for deriving mapping $G \in R^3$ is resolved here in a very simple fashion. First, in the computational domain there is constructed a non-orthogonal fine enough grid. Further, on the basis of the finite-volume method (or variational-difference method) the difference scheme is created. And finally, the corresponding selfadjoint grid elliptic problem is solved using modern high-performance iterative linear solvers. A general technology of generation of orthogonal grids is based on this computational algorithm. It is very convenient to use this technology of grid generation in a combination with a multi-block representation of a geometrical configuration.

Development of a complete system of grid generation is a very complex problem. Any grid generation system should have all necessary attributes existing in modern well-known systems of generation of block-structured grids, such as NGP, ICEM-CFD, EAGLE, namely: a graphic user interface, a model of grid topology, a built-in CAD system based on NURBS- splines, tools for analysis of constructed grids etc. On the other hand, the most part of modern CAD systems provide many functions from the above list, such as a powerfull graphic interface, a rich toolkit for construction of the complex geometrical configurations based on NURBS representation, a possibility to design grids on domain boundaries, analysis of constructed grid and so on. Besides, CAD systems allows to present a complex configuration as a combination of simple domains. Thus, practically all operations of grid generation are included and so, the only necessary operation which should be implemented is to take some multi-block grid from a CAD system and to construct on its basis a new orthogonal grid using the developed in this work solver.

Below there are presented peculiarities of constructing initial multi-block grids on the basis of CAD systems as well as the algorithm of transformation of initial grids into orthogonal ones. Some examples of constructed grids are also provided along with predictions of flows derived on them.

2. APPLICATION OF CAD SYSTEMS TO CONSTRUCTION OF MULTI-BLOCK GRIDS

For construction of an initial multi-block grid we use modern CAD system — Mechanical Desktop (version 2.0). This choice is based on a number of reasons, and the primary is the fact that the given product satisfies to the majority of the requirements that can be imposed upon to grid generation systems.

The basic requirements to grid generators can be formulated as the following list of necessary possibilities:

- Construction of the initial geometry;
- Editing and completion of the configuration;
- Development of a block structure;
- Production of the final computational grid with a high level of a user's control on grid quality involving smoothness, orthogonality, spacing etc.

Mechanical Desktop meets all the above requirements except the last.

A very important element for the efficient process of grid generation of a grid is the development and implementation of a non-conflicting graphic user interface (GUI) [6]. Performance of GUI operating depends on the following requirements:

- Compatibility — possibility of information exchange with other systems. For example, through the format IGES;
- Consistency of logical organization of technology of supporting and making decisions: decomposition onto subdomains, distribution of the topological information etc.;
- Easy in usage: convenient organization of the dialogue with a user;
- Existence of an expert subsystem: analysis of possibility of splitting onto subdomains, check of curve orientation etc.

Structural possibilities of grid generation in most codes are based on GTM data structure. Usage of topological objects of the system (edges, surfaces and blocks) to control the grid information for each subdomain allows to specify grid nodes only once and to use them in all adjacent objects where these nodes are common. These common nodes are defined by the topological specification of surfaces and blocks in the domain. Mechanical Desktop does have built-in GTM data structure and allows to prescribe topological information through specification of text attributes to edges or surfaces of the geometrical configuration. Further, this CAD system provides possibility to construct multi-block representation of the domain, to check direction of curve orientation etc.

Geometry description in the most of modern commercial grid generators is based on NURBS splines [7][8][9][10][11]. The same way of presentation is used in Mechanical Desktop for the geometrical database.

Generation of surface grids immediately on NURBS surfaces allows surface grids to preserve a high degree of accuracy with respect to the initial geometrical data. However, it also requires high quality of models in CAD systems in order to produce good grids. High quality here means that the model should not contain any undesirable intervals, overlaps or intersections between surfaces. Many grid generators have internal CAD system in order to create, modify and/or restore an imported geometry at preparation for grid generation.

Production of the final computational grid with a high level of user's control on grid quality including smoothness, orthogonality and spacing was incorporated in our approach into the code for constructing orthogonal grids. Thus, the process of grid generation consists of the following four items:

- Construction, editing of topology of block structure using facilities of Mechanical Desktop;
- Saving of derived results in the IGES;
- Conversion of the saved information by means of converter in the format of our external code for construction of orthogonal grids;
- Construction of the final computational grid with high-level control of a user on the grid quality (smoothness, orthogonality and spacing). The given item can be performed several times. In other words, if quality of the already obtained orthogonal grid seems to be unsatisfactory, this grid can be passed on the input of our code again. The repetition of this step allows to obtain an orthogonal grid of desirable quality.

This way of grid generation makes possible to design block-structured orthogonal and quasi-orthogonal grids. In the next section the features of the developed numerical algorithm for constructing orthogonal grids are demonstrated on an example of so-called 2.5D geometry.

3. ALGORITHM OF CONSTRUCTING ORTHOGONAL GRIDS

Let us consider the algorithm of constructing orthogonal curvilinear grid in domain $G \in R^3$ having the form of curvilinear rectangular and lying on a smooth 3D surface.

Assume that in domain $G \in R^3$ there is specified curvilinear coordinate system $\{a(x,y,z), b(x,y,z)\}$ and does exist non-degenerated mapping which transforms curvilinear domain G into unit square Ω so, that vertices of curvilinear rectangular G are transformed into vertices of unit square Ω whereas edges of G - into edges of square Ω. Suppose in addition that Jacobian of this mapping $\Im = \frac{\partial(x,y,z)}{\partial(a,b)} > 0$.

In domain Ω let us construct uniform grid with respect to a, b with spacing h_a, h_b which serves as an image of some curvilinear grid in domain G.

Omitting intermediate transformations, let us write the following representation for gradient (T - additional variable):

$$\begin{pmatrix} grad^a T \\ grad^b T \end{pmatrix} = \begin{pmatrix} 1 & -\cos j \\ -\cos j & 1 \end{pmatrix} \begin{pmatrix} \frac{l_b}{S} \frac{¶T}{¶a} \\ \frac{l_a}{S} \frac{¶T}{¶b} \end{pmatrix}$$

$$\left(x_a x_b + y_a y_b + z_a z_b\right) = l_a l_b \cos j \qquad (1)$$

$$l_\perp = l_a l_b \sin j = S$$

Define functions $q = q(a,b)$ è $h = h(a,b)$ on a smooth 3D surface so that $\vec{u} = grad q$ and $\vec{w} = grad h$.

Then in local contravariant basis $\{\vec{n}^a, \vec{n}^b\}$ vectors $\vec{u}$ and $\vec{w}$ can be written as follows:

$$\begin{matrix} u^a = grad^a q \\ u^b = grad^b q \end{matrix} \quad è \quad \begin{matrix} w^a = grad^a h \\ w^b = grad^b h \end{matrix}.$$

Let us require that orthogonality constraint holds for vectors $\vec{u}, \vec{w}$:

$$\begin{pmatrix} u^a & u^b \end{pmatrix} \begin{pmatrix} 1 & \cos j_{ab} \\ \cos j_{ab} & 1 \end{pmatrix} \begin{pmatrix} w^a \\ w^b \end{pmatrix} = 0 \qquad (2)$$

Taking into account that

$$\begin{pmatrix} w^a \\ w^b \end{pmatrix} = \begin{pmatrix} 1 & -\cos j_{ab} \\ -\cos j_{ab} & 1 \end{pmatrix} \begin{pmatrix} \frac{l_b}{S} \frac{¶h}{¶a} \\ \frac{l_a}{S} \frac{¶h}{¶b} \end{pmatrix},$$

expression (2) can be rewritten like this

$$k \begin{pmatrix} \frac{l_b}{S} u^a & \frac{l_a}{S} u^b \end{pmatrix} \begin{pmatrix} h_a \\ h_b \end{pmatrix} = 0 .$$

and that is equivalent to the following relations:

$$
\begin{aligned}
k\frac{l_b u^a}{S} &= h_b \\
k\frac{l_a u^b}{S} &= -h_a
\end{aligned}
\quad , \tag{3}
$$

$k = k(a, b) = \sin^2 j$ is non-zero function.

Differentiation of the first equation of system (3) with respect to a and the second - to b and summation of the derived equations results in:

$$
\begin{aligned}
&\frac{\P}{\P a} k\frac{l_b u^a}{S} + \frac{\P}{\P b} k\frac{l_a u^b}{S} = 0 \\
&\begin{pmatrix} u^a \\ u^b \end{pmatrix} = \begin{pmatrix} 1 & -\cos j_{ab} \\ -\cos j_{ab} & 1 \end{pmatrix} \begin{pmatrix} \frac{l_b}{S}\frac{\P q}{\P a} \\ \frac{l_a}{S}\frac{\P q}{\P b} \end{pmatrix}
\end{aligned}
\tag{4}
$$

Mixed boundary value problem is imposed at the boundary:

$$q = 0, (a, b) \in ä_1, q = 1, (a, b) \in ä_3, \overset{1}{u} = 0, (a, b) \in ä_2 \cup ä_4, \tag{5}$$

Transform scalar product (2) to the form:

$$\begin{pmatrix} w^a & w^b \end{pmatrix} \begin{pmatrix} 1 & \cos j_{ab} \\ \cos j_{ab} & 1 \end{pmatrix} \begin{pmatrix} u^a \\ u^b \end{pmatrix} = 0$$

or

$$\begin{pmatrix} \frac{l_b}{S} w^a & \frac{l_a}{S} w^b \end{pmatrix} k \begin{pmatrix} q_a \\ q_b \end{pmatrix} = 0 \ ,$$

that is equivalent to system of equations:

$$
\begin{aligned}
\frac{1}{k}\frac{l_b w^a}{S} &= q_b \\
\frac{1}{k}\frac{l_a w^b}{S} &= -q_a
\end{aligned}
\tag{6}
$$

and after similar differentiation and addition we obtain from system (6):

$$
\begin{aligned}
&\frac{\P}{\P a}\frac{1}{k}\frac{l_b w^a}{S} + \frac{\P}{\P b}\frac{1}{k}\frac{l_a w^b}{S} = 0 \\
&\begin{pmatrix} w^a \\ w^b \end{pmatrix} = \begin{pmatrix} 1 & -\cos j_{ab} \\ -\cos j_{ab} & 1 \end{pmatrix} \begin{pmatrix} \frac{l_b}{S}\frac{\P h}{\P a} \\ \frac{l_a}{S}\frac{\P h}{\P b} \end{pmatrix}
\end{aligned}
\tag{7}
$$

Mixed boundary value problem is considered at the domain boundary:

$$h = 0, (a, b) \in ä_2, h = 1, (a, b) \in ä_4, \overset{1}{w} = 0, (a, b) \in ä_1 \cup ä_3, \tag{8}$$

where n is the normal to the corresponding boundary.

The developed algorithm can be easy generalized to the 3D case.

4. VERIFICATION OF THE ALGORITHM

In this section there are presented results of investigation on influence of size of the initial grid on quality of derived orthogonal grids. The results are derived for the surface depicted in Fig.1.

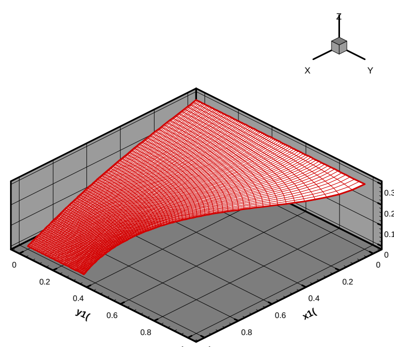


Figure 1. Surface for testing of algorithm

Two sequences of calculations have been conducted. In the first one the following sequence of refining initial grids $11\times 11, 21\times 21, 41\times 41, 81\times 81, 161\times 161, 321\times 321$ has been used in order to derive the orthogonal grid of fixed size. In contrast, the second series is connected with construction of orthogonal grids of different size $11\times 11, 21\times 21, 41\times 41, 81\times 81, 161\times 161, 321\times 321$ using the initial grid of fixed size.

To estimate quality of derived orthogonal grids, we used in both cases grid function $\cos j$ defining in curvilinear coordinates in accordance with expression (1). The results of calculations are presented in Table 1. Values of this 2D function $\cos j$ are shown in this Table as a fraction where the numerator shows the minimal, whereas the denominator – the maximal values of this grid function. It is easy to see from Table 1 that we have the direct dependence of quality of derived orthogonal grids on increasing of size of the initial grid. In other words, the higher size of the initial grid provides the better quality of the derived orthogonal grid.

Table 1

$Base \setminus Orto$	11×11	21×21	41×41
$\frac{11\times11}{-7.041e-1/0.00}$	$\frac{-1.360e-1}{1.556e-2}$	$\frac{-1.640e-1}{6.05e-2}$	$\frac{-1.871e-1}{7.009e-2}$
$\frac{21\times21}{-7.167e-1/1.290e-2}$	$\frac{-1.303e-1}{-8.077e-4}$	$\frac{-6.195e-2}{2.635e-2}$	$\frac{-8.669e-2}{4.641e-2}$
$\frac{41\times41}{-7.230e-1/2.250e-2}$	$\frac{-1.279e-1}{-9.007e-4}$	$\frac{-5.417e-2}{-1.696e-4}$	$\frac{-3.991e-2}{1.736e-2}$
$\frac{81\times81}{-7.262e-1/2.721e-2}$	$\frac{-1.273e-1}{-3.437e-5}$	$\frac{-5.295e-2}{3.694e-4}$	$\frac{-3.635e-2}{4.596e-4}$
$\frac{161\times161}{-7.274e-1/2.957e-2}$	$\frac{-1.271e-1}{9.105e-4}$	$\frac{-5.260e-2}{8.509e-4}$	$\frac{-3.507e-2}{1.224e-3}$
$\frac{321\times321}{-7.280e-1/3.076e-2}$	$\frac{-1.271e-1}{1.272e-3}$	$\frac{-5.249e-2}{9.371e-4}$	$\frac{-3.489e-2}{8.897e-4}$
+	+	+	+
$Base \setminus Orto$	81×81	161×161	321×321
$\frac{11\times11}{-7.041e-1/0.00}$	$\frac{-1.984e-1}{1.093e-1}$	$\frac{-2.041e-1}{1.398e-1}$	$\frac{-2.069e-1}{1.590e-1}$
$\frac{21\times21}{-7.167e-1/1.290e-2}$	$\frac{-9.892e-2}{5.740e-2}$	$\frac{-1.050e-1}{6.696e-2}$	$\frac{-1.080e-1}{7.792e-2}$
$\frac{41\times41}{-7230e-1/2.250e-2}$	$\frac{-4.410e-2}{2.740e-2}$	$\frac{-5.027e-2}{3.446e-2}$	$\frac{-5.335e-2}{4.096e-2}$
$\frac{81\times81}{-7.262e-1/2.721e-2}$	$\frac{-3.728e-2}{9.890e-3}$	$\frac{-3.833e-2}{1.715e-2}$	$\frac{-4.082e-2}{2.001e-2}$
$\frac{161\times161}{-7.274e-1/2.957e-2}$	$\frac{-3.533e-2}{1.334e-3}$	$\frac{-3.575e-2}{6.087e-3}$	$\frac{-3.646e-2}{9.229e-3}$
$\frac{321\times321}{-7.280e-1/3.076e-2}$	$\frac{-3.474e-2}{9.682e-4}$	$\frac{-3.491e-2}{1.081e-3}$	$\frac{-3.515e-2}{3.455e-3}$

Besides, we estimate quality of the derived orthogonal grids in dependence of recurrent employment of our algorithm (see Table 2). At the beginning we used as the initial grid the orthogonal 81×81 derived from the initial grid of the same size, with the following minimal and maximal values of the grid function $\frac{-3.533e-2}{1.334e-3}$ (see Table 1).

Table 2

$Base \setminus Orto$	81×81
$\frac{81\times81}{-7.262e-1/2.721e-2}$	$\frac{-3.728e-2}{9.890e-3}$
$\frac{81\times81}{-3.728e-2/9.890e-3}$	$\frac{-1.266e-2}{7.011e-3}$
$\frac{81\times81}{-1.266e-2/7.011e-3}$	$\frac{-1.057e-2}{6.341e-3}$

After processing of this grid via our orthogonalizing code we obtain new orthogonal grid with a more higher quality in sense of the grid cosine. This operation was repeated several times considering the derived at the previous step orthogonal grid as the initial one at the current step.

5. EXAMPLES OF BLOCK-STRUCTURED GRIDS

Below there are presented examples of constructing block-structured grids on the basis of the presented above technology.

Figures 2 and 3 demonstrate methods of specifying topological data and boundary conditions in the 2D case using Mechanical Desktop (version 2.0).

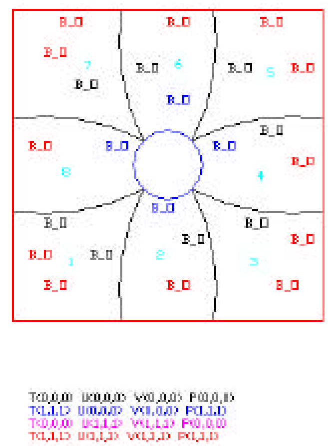


Figure 2. Implicit specification of topological data (attribute with names B_0)

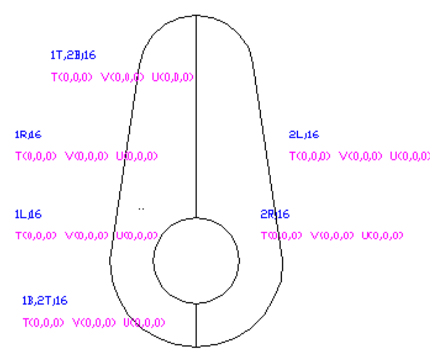


Figure 3. Explicit definition of topological data

Such a representation is constructed by a user via possibilities of Mechanical Desktop.

Then the automatic stage of constructing an orthogonal grid is started.

The visible part of a display containing the geometry presented in the form of blocks, topological data and boundary conditions is saved as a IGES file.

After that this saved data is transformed through the developed converter into the format of our code for constructing orthogonal grids.

This code performs generation of the final orthogonal grid using the developed solver which allows a high-level control of a user on quality of the grid. Some examples of the derived grids are presented below.

Figure 4 shows the surface between side blades of a turbine.

Figure 5 demonstrates an aircraft airfoil with a part of the fuselage.

Stages of constructing a multi-block structure between side surfaces of blades are depicted in Fig. 6.

Finally, an automobile surface and a display body are shown in Fig. 7 and 8, respectively.

6. CONCLUSIONS

- A new recursive-iterative method for constructing orthogonal conformal grid is developed here. It employs solution of elliptic equations on the initial grid.
- This approach can be easy used to construct multi-block structures with the following properties:
  - Continuity of the first and second derivatives at boundaries,
  - Efficient control on grid spacing.
- Here there is proposed a non-traditional approach to use CAD systems as the system for geometrical modeling which does have all attributes of commercial grid generators:
  - Improved GUI,
  - Build-in model of the grid topology,
  - NURBS — as the geometrical base of data,
  - IGES as the format for data exchange with other CAD systems,
  - Powerful toolkit for analysis of constructed grids.

Thus, there is proposed a new method for generation of large-scale multi-block orthogonal grids for applied and industrial CFD applications. This method is automatized enough (no less 90%).

As it was mentioned by Thompson in [12] grid generation tools must be designed to be applied by design engineers rather than grid generation specialists.

There is a clear need for interaction CFD with commercial CAD venders.

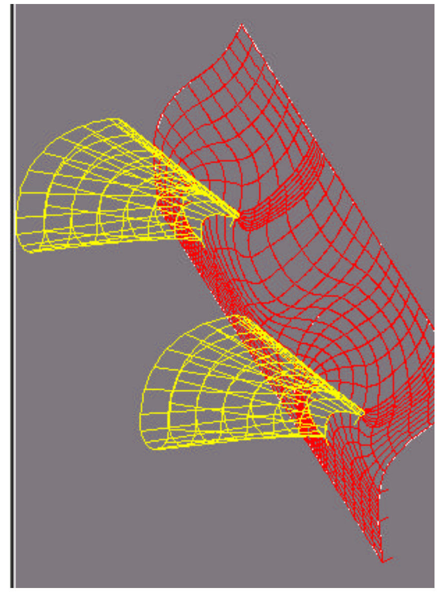


Figure 4. A pair of of turbomachinery blades at the multiblock grid of support

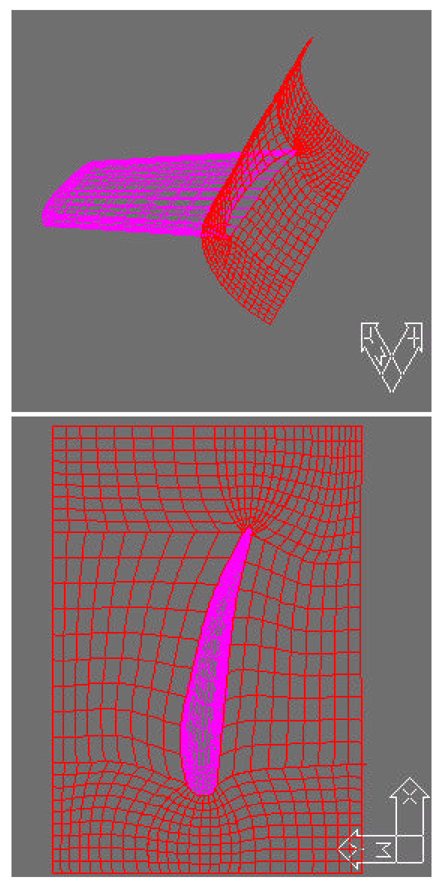
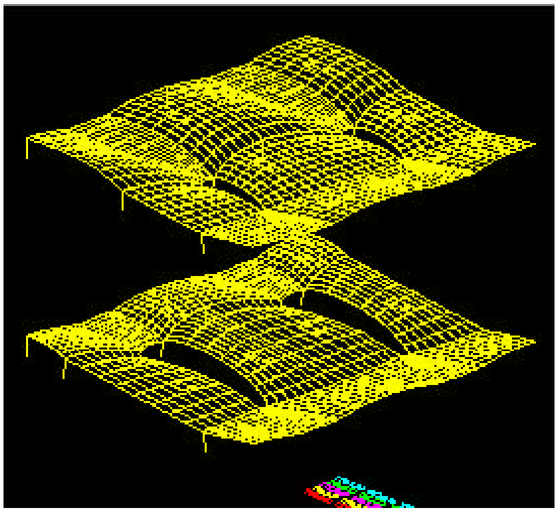
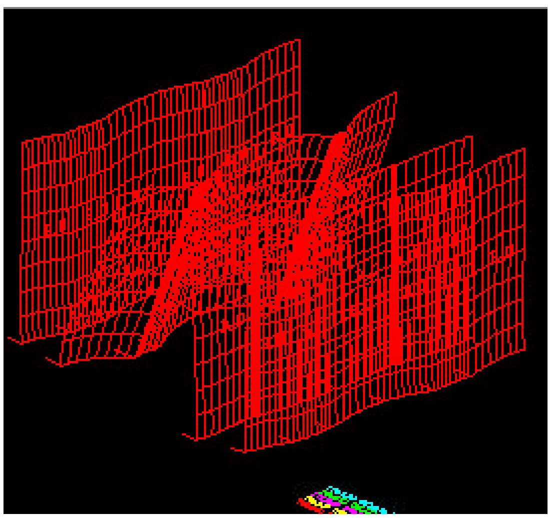


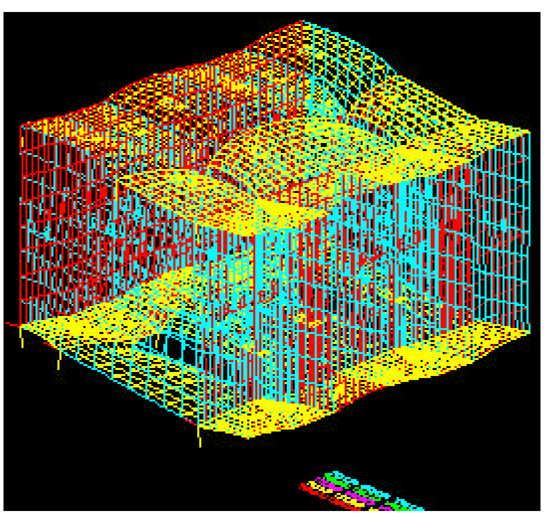
Figure 5. A wing with a part of fuselage at the multiblock grid of support



a) Upper and lower surfaces



b) Left surfaces



c) Common form

Figure 6. A multiblock structure between lateral areas of blades

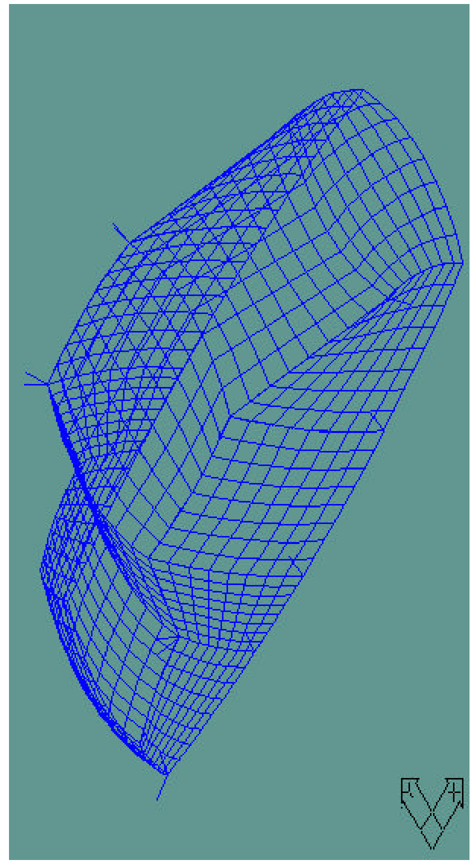


Figure 7. A multiblock grid of support for an automobile surface

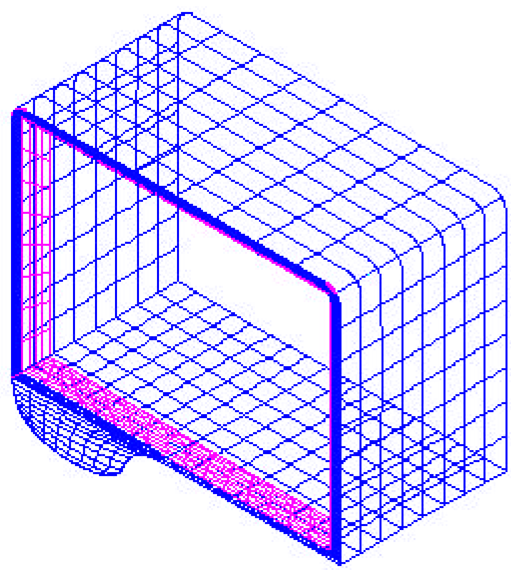


Figure 8. A part body for a display

REFERENCES

[1] M.M. Rai, "A conservative treatment of zonal boundaries for Euler equation calculations", *J. Comput. Phys.*, Vol. 62 pp.472-503 (1986)

[2] J.F.Thompson, "A composite grid generation code for general 3D regions — the Eagle code", *AIAA J.*, Vol. 26 (3) pp.271-272 (1988)

[3] J.L. Steger, F.C. Dougherty, J.A. Benek, "A chimera grid scheme, Advances in Grid Generation", Ghia K.N and Ghia U., *ASME FED*, Vol. 5 (1983).

[4] G. Ryskin, L.G. Leal, "Orthogonal mapping", *J. Comput. Phys.*, Vol.50 pp.71-100 (1983)

[5] R. Duraiswami, A. Prosperetti, "Orthogonal mapping in two dimensions", *J. Comput. Phys.*, Vol.98 pp.254-268 (1992)

[6] A. Gaither and et al. The National Grid Project: a system overview. NASA CP 3291, 1995

[7] G. Farin, Curves and Surfaces for Computer Aided Geometric Design. Thrid Edition, Academic Press, San Diego, CA, 1992

[8] M.E. Mortensen, Geometric Modeling. Wiley & Sons, New York, NY, 1985

[9] L. Piegl, "On NURBS: A Survey". *IEEE Computer Graphics & Applications*, pp. 55. January (1991)

[10] F. Yamaguchi, Curves and Surfaces in Computer Aided Geometric Design. Springer-Verlag, New York, NY, 1988

[11] J. Hoschek, D. Lasser, Fundamentals of Computer Aided Geometric Design. A K Peters, Ltd., Wellesley, MA, 1993

[12] J.F. Thompson, B. Hamman, "A survey of grid generation techniques and systems with emphasis on recent developments", *Surveys on Mathematics for Industry*, Vol.6 pp.289-310 (1997)