Elliptic Surface Grid Generation in Three-Dimensional Space

Lee Kania
Sverdrup Technology, Inc.
Structures and Dynamics Dept.
Huntsville, Alabama 35806

SUMMARY

A methodology for elliptic surface grid generation in three-dimensional space is described. The method solves a Poisson equation for each coordinate on arbitrary surfaces using successive line over-relaxation. The complete surface curvature terms have been discretized and retained within the nonhomogeneous term in order to preserve surface definition; there is no need for conventional surface splines. Control functions have been formulated to permit control of grid orthogonality and spacing. A method for the interpolation of control functions into the domain has been devised which permits their specification not only at the surface boundaries but within the interior as well. An interactive surface generation code which makes use of this methodology is currently under development.

INTRODUCTION

Developments in grid generation techniques have been a pacing item for application of fluid dynamics codes to complex configurations. In these instances the task of grid generation is as significant a problem as the flow field analysis itself. Clearly, increased emphasis in research and development of grid generation methods is necessary to improve the efficiency with which CFD analyses are conducted. Indeed, recent years have seen the arrival of some powerful and flexible techniques which address this issue.

Significant contributions to the grid generation effort over the years have included, to name only a few, the GRAPE2D code of Sorenson[1], the EAGLE surface/grid code of Thompson[2] and the GRIDGEN codes by Steinbrenner[3] et al. The GRAPE2D code solves Poisson's equation in two-dimensions and utilizes what was a novel approach for determination of the boundary control functions. Use of this code remains widespread throughout the CFD community. Though the code is extremely versatile, grid optimization is somewhat hampered due to the batch mode implementation. The EAGLE code combines techniques in surface grid generation as well as two- and three-dimensional grid generation. As originally released, the code operated in the batch mode on the Cray. Since its inception, attempts to alleviate the inefficiencies of this platform have been made; an interactive version of the code has recently been released. The GRIDGEN series of codes is a more recent appearance. These codes permit interactive surface design and batch grid construction. A host of options are avaliable which afford a high degree of flexibility, however, intelligent use of the majority

of these options requires the user to be well versed in current grid generation techniques. In addition, invocation of the elliptic surface generation mode may become rather cumbersome as the user may be required to intervene in order to ensure construction of well defined surface splines.

The current work builds upon past research efforts and also exploits the capability of present day engineering workstations to produce an easy to use interactive elliptic surface generation code which affords the user a high degree of flexibility and robustness. All data necessary to control grid evolution is specified interactively. Grid optimization is greatly enhanced as solution constraints may be adjusted as the grid evolves. The following sections describe the method and its implementation in greater detail.

DESCRIPTION OF METHOD

Governing Equations

The current method solves the vector Poisson equation

$$\alpha\vec{r}_{\xi\xi} - 2\beta\vec{r}_{\xi\eta} + \gamma\vec{r}_{\eta\eta} = -J^2(P\vec{r}_{\xi} + Q\vec{r}_{\eta}) + [(\alpha\vec{r}_{\xi\xi} - 2\beta\vec{r}_{\xi\eta} + \gamma\vec{r}_{\eta\eta}) \cdot \hat{n}]\,\hat{n} \tag{1}$$

for the surface coordinates where

$$\vec{r} = (x, y, z) \tag{2}$$

and

$$\begin{aligned} \alpha &= \vec{r}_{\eta} \cdot \vec{r}_{\eta} \\ \beta &= \vec{r}_{\xi} \cdot \vec{r}_{\eta} \\ \gamma &= \vec{r}_{\xi} \cdot \vec{r}_{\xi} \end{aligned} \tag{3}$$

In this equation, the Jacobian is given as

$$J = |\vec{r}_{\xi} \times \vec{r}_{\eta}| \tag{4}$$

and the normal vector is given as

$$\hat{n} = (n_x)\hat{i} + (n_y)\hat{j} + (n_z)\hat{k} = \frac{1}{J}(\vec{r}_{\xi} \times \vec{r}_{\eta}) \tag{5}$$

Equation (1) was originally formulated by Warsi[4]. The last term on the right hand side of equation (1) represents the surface curvature terms. Previous researchers have employed a less rigorous approach involving a solution of equation (1) for x and y only, the z-coordinate being provided by splines of the form $z = z(x, y)$. Creation of splines having this functional form will present a problem for surfaces not single valued in z. In the current implementation the complete surface curvature terms are retained in each equation in order to maintain surface definition; construction of surface splines is not required. Equation (1) is solved using successive line over-relaxation. Central differences are used to approximate all derivatives

which results in a scalar tridiagonal system for each sweep through the grid. At each level the Thomas algorithm is used to solve the system of equations for the coordinate vector $\vec{r}^{\,*}$. The coordinates at the previous iterate, n, are then corrected to yield the new coordinate

$$\vec{r}^{\,n+1} = \vec{r}^{\,n} + \omega(\vec{r}^{\,*} - \vec{r}^{\,n}) \tag{6}$$

where ω is the relaxation parameter.

Control Functions

Evaluation of the control functions is patterned after the work of Sorenson and Steger[5]. In their work the control functions are based on a prescribed step size and orthogonality constraint. For surfaces in three-dimensional space we impose these same constraints which, for the $\xi = constant$ boundaries, are expressed as

$$s_\xi = |\vec{r}_\xi| \approx \frac{\Delta s}{\Delta \xi} \tag{7}$$

$$\vec{r}_\xi \cdot \vec{r}_\eta = 0 \tag{8}$$

and a second orthogonality constraint which is expressed as

$$\vec{r}_\xi \cdot \hat{n} = 0 \tag{9}$$

For the $\eta = constant$ boundaries the constraints are written as

$$s_\eta = |\vec{r}_\eta| \approx \frac{\Delta s}{\Delta \eta} \tag{10}$$

$$\vec{r}_\xi \cdot \vec{r}_\eta = 0 \tag{11}$$

and

$$\vec{r}_\eta \cdot \hat{n} = 0 \tag{12}$$

For the $\xi = constant$ boundaries, equations (7) through (9) are solved for $\vec{r}_\xi$ to yield

$$\vec{r}_\xi = \frac{s_\xi(\vec{r}_\eta \times \hat{n})}{|\vec{r}_\eta \times \hat{n}|} \tag{13}$$

A similar procedure for the $\eta = constant$ boundaries using equations (10) through (12) yields

$$\vec{r}_\eta = -\frac{s_\eta(\vec{r}_\xi \times \hat{n})}{|\vec{r}_\xi \times \hat{n}|} \tag{14}$$

These expressions provide the first derivative at the respective surface boundaries without the need to difference into the field. The partial derivatives $\vec{r}_\eta$ and $\vec{r}_{\eta\eta}$ for the $\xi = constant$ boundaries and $\vec{r}_\xi$ and $\vec{r}_{\xi\xi}$ for the $\eta = constant$ boundaries are evaluated using the surface edge distributions. The derivatives are invariant with iteration as long as the prescribed step size and boundary distributions themselves do not change. In the event that boundary points are permitted to "float" with the solution as an alternate means to achieve orthogonality, the

particular derivative must, of course, be re-evaluated after each cycle. The remaining second derivatives $\vec{r}_{\xi\xi}$ and $\vec{r}_{\eta\eta}$ on the $\xi = constant$ and $\eta = constant$ boundaries, respectively, are the only quantities which vary with iteration. They are evaluated using one-sided derivatives as the grid evolves.

With the boundary coordinate derivatives known, the control functions P and Q may now be evaluated from equation (1). The apparent uniqueness problem due to three equations in two unknowns is removed by reducing the set to two equations by formation of the scalar product of equation (1) with $\vec{r}_{\xi}$ and also with $\vec{r}_{\eta}$. Straightforward algebraic manipulation of these equations then yields

$$P = \frac{\beta(\vec{\varrho} \cdot \vec{r}_{\eta}) - \alpha(\vec{\varrho} \cdot \vec{r}_{\xi})}{J^2(\alpha\gamma - \beta^2)} \tag{15}$$

and

$$Q = \frac{\beta(\vec{\varrho} \cdot \vec{r}_{\xi}) - \gamma(\vec{\varrho} \cdot \vec{r}_{\eta})}{J^2(\alpha\gamma - \beta^2)} \tag{16}$$

where

$$\vec{\varrho} = \alpha\vec{r}_{\xi\xi} - 2\beta\vec{r}_{\xi\eta} + \gamma\vec{r}_{\eta\eta} \tag{17}$$

Though the β term vanishes for orthogonal conditions, it is retained for the case where the orthogonality constraint must be necessarily compromised. These expressions for the control functions are, of course, valid anywhere within the field. Typically, the control functions are evaluated at the surface boundaries and interpolated by some means into the interior. However, more flexibility can be obtained by instituting a more generalized procedure in which interior regions can come under direct control as well. This approach, however, introduces additional difficulties as the interpolation for the control functions across the surface is more complicated.

Control Function Interpolation

The solution of equation (1) includes the effects of a source or nonhomogeneous term which involves both P and Q, hence, these quantities must be specified across the grid. Rather than evaluate the interior control functions based on an initial distribution of grid points, we opt for a method of interpolation, similar to the approach taken by Sorenson and Steger[5], in which the value of the control functions within the interior is determined from the boundary values. For the case in which control functions are specified only on the surface boundaries, transfinite interpolation is used to determine the control functions within the interior. This approach yields a reasonably smooth variation and also affords a good measure of flexibility due to the arbitrary form of the blending functions. These functions permit the user to control the damping rate of the control functions. In the current implementation a cosine function is used as the functional form of the blending functions with the decay half-period, in terms of a point count, specified by the user. Outside the region of user control the grid becomes locally Laplacian which yields the smoothest grid possible.

While extremely useful, this approach does have its shortcomings as the blending functions are most easily constructed to be a function of grid index. Repeated iteration on the grid alters the physical distance over which the control functions are damped to zero; this is, to varying degrees, reflected in the resulting grid. A variation on this method which attempts to fix the physical decay distance has also been formulated and will be described in a later section.

For the case in which interior points are to come under direct control and, hence, control functions are to be specified within the interior of the surface, the interpolation for the control functions becomes somewhat more involved. The methodology currently incorporated into the computer code uses a sequential application of transfinite interpolation at the surface boundaries and at all interior regions of the surface. This particular approach is quite limited as it assumes that, for any point within the grid, the control function is determined by a single set of boundaries only, be it the surface boundaries or those of any interior region. Stated another way, this limitation permits repeated application of transfinite interpolation for all regions with the stipulation that the effects of any region vanish before the effects of any other region are introduced. For the region outside that of an interior patch, the control function variation is determined by using a combination of 1D and 2D interpolations

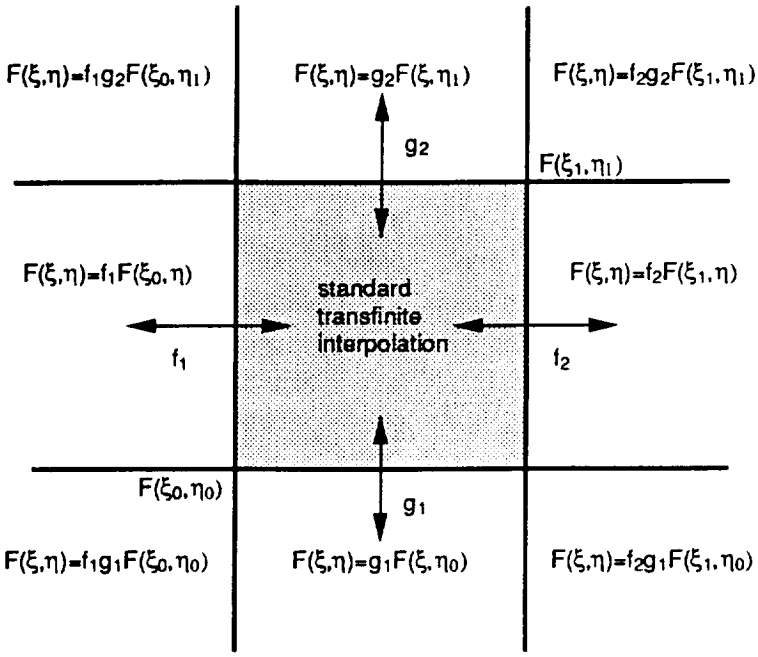

Figure 1. Determination of control functions in the vicinity of an internal patch.

Though this technique is not as robust as we might like, it remains a useful option. This approach is under continued development.

Surface Fitting of Control Functions

A shortcoming inherent in the basing of interpolation blending functions on grid index is evident as the influence of the control functions may be observed to become restricted to an increasingly smaller region of the surface as the grid is refined. A variation of this method which maintains the initial physical decay distance of the control functions has been formulated. Control functions are evaluated as described above though the interpolation is performed on the original surface grid which is, of course, unchanging. At each point on the original grid the control function first- and second-derivatives ($P_\xi, P_{\xi\xi}$, etc.) in transformed space are first evaluated. An interpolation onto the current grid using a Taylor series expansion about the "neighbor" points in the initial surface is then made. These "neighbor" points are determined through a minimization process which will be described in the next section. This method for interpolation tends to provide an additional measure of control as the effects of the control functions are typically felt over a larger area of the surface.

Surface Coordinate Correction

In application it has been found that, in regions of high surface curvature, the evolving surface tends to deviate from the initial or basis surface. To preclude the deviation, a surface coordinate correction scheme has been devised which makes use of "neighbor" point relationships and both physical and transformed quantities to adjust the coordinates to second-order after the grid has converged. A variation of this procedure was used to adjust surface grid points in a three-dimensional adaptive grid scheme with excellent results[6].

The surface correction method is non-iterative and makes use of the basis grid and its mapping to computational space

$$(x, y, z)_o \rightarrow (\xi, \eta)_o \tag{18}$$

where the subscript o denotes the basis surface.

The point on the basis surface closest to (ξ, η) is then determined through a minimization process as (ξ_o, η_o) and distances in the transformed plane are determined

$$\Delta\xi_o = (\xi_x)_o \Delta x + (\xi_y)_o \Delta y + (\xi_z)_o \Delta z \tag{19}$$

$$\Delta\eta_o = (\eta_x)_o \Delta x + (\eta_y)_o \Delta y + (\eta_z)_o \Delta z \tag{20}$$

where

$$\Delta\vec{r} = \vec{r}(\xi, \eta) - \vec{r}_o(\xi_o, \eta_o) \tag{21}$$

The "neighbor" point is obtained when equations (19) and (20) have been minimized. The corresponding surface point is then obtained from the Taylor series expansion about the basis grid point

$$\begin{aligned} \vec{r}(\xi, \eta) = {} & \vec{r}_o(\xi_o, \eta_o) + \left(\frac{\partial \vec{r}_o}{\partial \xi_o}\right) \Delta\xi_o + \left(\frac{\partial \vec{r}_o}{\partial \eta_o}\right) \Delta\eta_o + \left(\frac{\partial^2 \vec{r}_o}{\partial \xi_o \partial \eta_o}\right) \Delta\xi_o \Delta\eta_o \\ & + \frac{1}{2}\left\{\left(\frac{\partial^2 \vec{r}_o}{\partial \xi_o^2}\right) \Delta\xi_o^2 + \left(\frac{\partial^2 \vec{r}_o}{\partial \eta_o^2}\right) \Delta\eta_o^2\right\} \end{aligned} \tag{22}$$

INTERACTIVE CODE DEVELOPMENT

Development of an interactive tool incorporating the surface grid generation methodology is in progress. The final code is to be included as an option in the interactive surface generation code currently under development by Warsi[7] which encompasses aspects of interactive curve generation, surface generation and a novel approach for surface point adjustment termed "surface constrained transfinite interpolation". There are further plans to include surface constrained Bezier curves as well which will provide an invaluable means to effect local grid control. In the implementation of the current approach, pop-up menus are used to promote ease of use and facilitate data entry. Specification of boundary/interior step sizes and control function decay rates are performed interactively with the mouse. Surface evolution is displayed in real time during which the user has the option of revising all constraints and inputs and then restarting or continuing the elliptic generation process. Furthermore, interior surface grid patches and associated constraints are also specified interactively.

APPLICATIONS

To demonstrate the capability of the method, various externally generated surfaces have been subjected to refinement. The results to be presented will highlight the strengths and weaknesses of the approach. The techniques developed to circumvent the shortcomings of the method are also evaluated. In addition, the mode of interior grid control is also demonstrated.

The first surface refined by the current method was constructed using an exponentially damped sine wave, the specific funtion having the form

$$y = 3e^{-\frac{3x}{2x_L}} e^{-2|\frac{z}{z_L}|} sin(\frac{4\pi z}{z_L}) \tag{23}$$

which is defined over $0 \leq x \leq x_L$ and $-z_L \leq z \leq z_L$ where $x_L = 10$ and $z_L = 12$. This particular functional form was selected as it would permit an assessment of the method in regions exhibiting significant curvature variation. The initial surface, which is shown in figure 2, has dimensions of 101×41 and is uniform in each direction. With the elliptic surface generator an attempt was made to increase the leading and trailing near-edge resolution and maintain orthogonality. Figure 3 shows the final surface generated after approximately 50 cycles with $\omega = 0.90$. Note also in this application that the left and right boundary points have been permitted to "float" in order to provide orthogonality and the desired spacing in the vicinity of the corners. The control functions over the interior of the surface were interpolated from the 4 edges using standard transfinite interpolation as described earlier. The functional form of the blending functions utilized the cosine function with the decay half-period specified as 10 points. No control functions were applied at the left or right boundaries.

Since the surface equation is available, the degradation introduced through the solver can be easily examined. Given the new "planform" coordinates x and z, a new y coordinate was calculated using equation (23). The difference between the solved y coordinate and that given by equation (23) is shown as a carpet plot in figure 4 in which the vertical dimension has been scaled by a factor of 50 to highlight the discrepancies. Note the "spikes" present

at regions of significant curvature change, curvature and spike amplitude being directly proportional. Also note that at the leading edge, where the finer resolution has been prescribed, the surface degradation is minimal. This leads to the conclusion that the extent of surface degradation is dependent upon local curvature variation and resolution; a problem inherent in discrete surface representation.

A means to restore the correct surface shape was presented in the section entitled "Surface Coordinate Correction". This scheme was applied after the elliptic generation process, the final surface is shown in figure 5. Visual inspection reveals no change in the surface grid point distribution or the surface shape. The corresponding carpet plot of the surface error on the expanded scale is shown in figure 6. Note that the surface shape has been essentially restored.

In the previous application the prescribed damping period for the spacing control function was set at the relatively high value of 10. Despite this relatively long period, the attraction of points to the leading and trailing edges does not appear to be very strong. In the section of the paper entitled "Surface Fitting of Control Functions" an alternate method of determining the variation of the control functions across the surface was described. Application of this method to the sine wave surface, using constraints identical to those of the previous application, produces the surface shown in figure 7. Note that the near-edge resolution is much finer as the effects of the control function are permitted to propagate further out from the respective edge. A carpet plot of the y surface coordinate error is presented in figure 8. Note that the extent of surface degradation at the leading edge is reduced due to the increased resolution. Application of the surface correction method restored the surface shape, though again, no change was visible.

To contrast the methods presented for the interpolation of the control functions, near-edge detail for the initial surface, the surface created using direct transfinite interpolation and that using the surface fitting method is shown in figure 9. The alternate method clearly provides a more powerful means to achieve increased grid control.

The next surface refined by the method involves a generic fuselage forebody with canopy. This configuration, which is shown in figure 10, features a lobed cross section at the canopy region with circular cross sections at both nose and aft regions. It was expected that the near-canopy region would provide the most difficulty for the method due to the relatively large curvature variation. Increased resolution at the symmetry plane was prescribed and, in this case, there was a visible degradation of the surface in the vicinity of the canopy. Exploded views of the canopy region for the initial, intermediate and corrected surfaces are shown for comparison in figure 11 in which the surfaces have been reflected in order to highlight the degradation. For comparison with the initial surface shown in figure 10, the final corrected surface grid is shown in figure 12. Note the extensive movement of points to the canopy region. This surface was obtained with $\omega = 1.0$ after approximately 75 iterations.

The final application involves an attempt to control the grid within the domain interior. The exponentially damped sine wave surface is used again, though in this case, control

functions are specified rather than interpolated on the boundaries of a rather large section of the interior in order to provide for increased local resolution. The control functions across the remainder of the surface interior are determined using the alternate means described earlier within this paper. The resulting grid is shown in figure 13. Note that the spacial variation of the step size is quite smooth although orthogonality has not been achieved near some portions of the interior boundary. This feature of the code is under continuing development.

CONCLUSIONS

The utility and robustness of the method has been shown. The approach permits elliptic generation on somewhat arbitrary surfaces with relative ease. Though the method does, in essence, require a form of surface splines, their use is transparent to the user. The scheme devised to correct for surface degradation has been found to perform quite well. In addition, the variation on the approach used in the interpolation of the control functions has been found to permit increased grid control. Optional control of interior grid points has been demonstrated although the approach is not yet as general as necessary.

FUTURE WORK

Work to formulate a more robust means to permit user control within the interior of arbitrary surfaces is in progress. This would bring arbitrary grid lines as well as rectangular regions under user control. The means to interpolate the control functions on the remainder of the grid is also under development. Control function damping rates will also be double valued at the desired sections in order to provide bi-directional grid control. Once the method has been fully developed it will be included as an option in the surface code under development by Warsi[7].

REFERENCES

[1] Sorenson, R., "A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation", NASA TM 81198, May 1980.

[2] Thompson, J., "A Composite Grid Generation Code for General 3D Regions-the EAGLE Code", *AIAA Journal*, Vol. 26, No. 3, March 1988.

[3] Steinbrenner, J., Chawner, J. and Fouts, C., "The GRIDGEN 3D Multiple Block Grid Generation System", WRDC TR 90-3022, July 1990.

[4] Warsi, Z.U.A., "Numerical Grid Generation in Arbitrary Surfaces through a Second-Order Differential-Geometric Model", *Journal of Computational Physics*, Vol. 64, No. 1, May 1986.

[5] Sorenson, R. and Steger, J, "Numerical Generation of Two-Dimensional Grids by the Use of Poisson Equations with Grid Control at Boundaries", NASA CP 2166, October 1980.

[6] Kania, L., "Three-Dimensional Adaptive Grid Generation with Applications in Nonlinear Fluid Dynamics", AIAA 92-0661, presented at the 30^{th} Aerospace Sciences Meeting, Reno, NV, January 1992.

[7] Warsi, S. A., "Algebraic Surface Grid Generation in Three-Dimensional Space", In *NASA Workshop on Software Systems for Surface Modeling and Grid Generation*, NASA-CP-3143, April 1992.

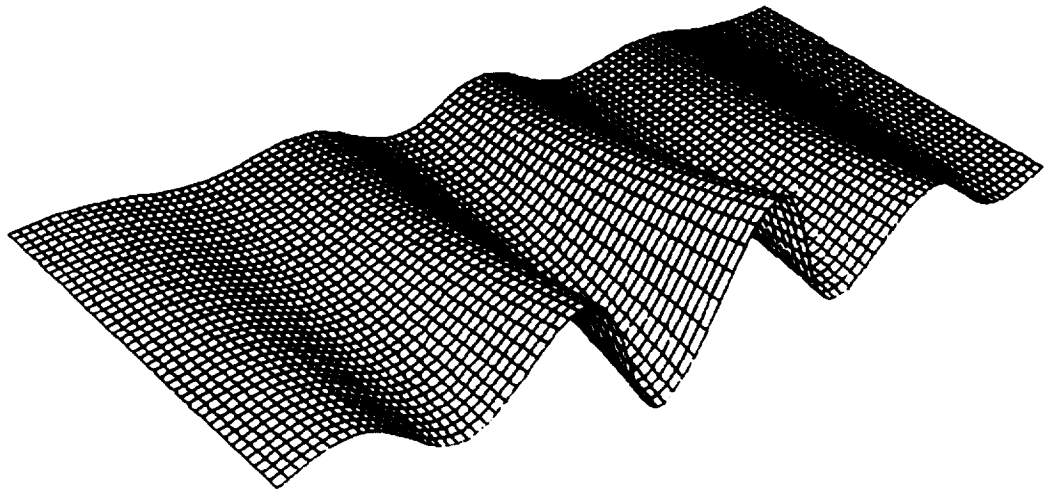

Figure 2. Damped sine wave surface.

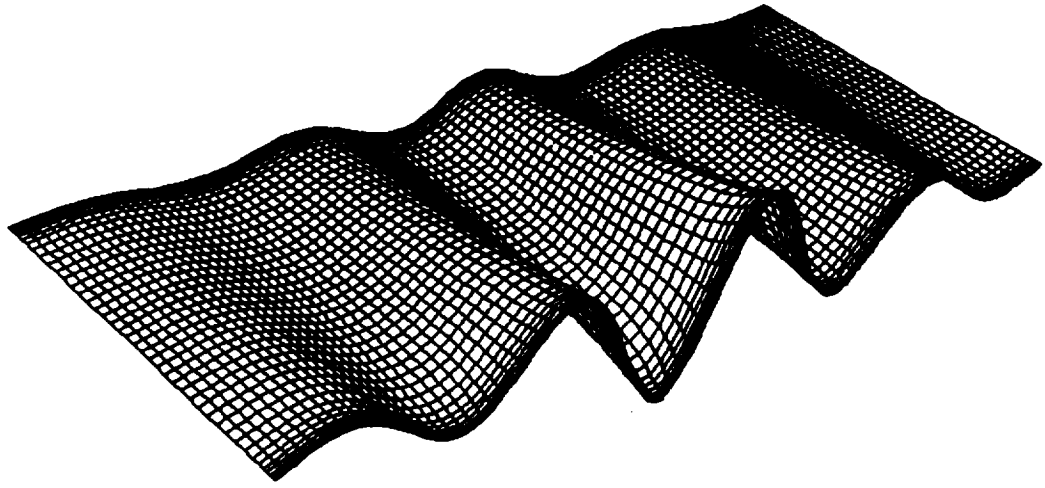

Figure 3. Elliptically generated surface with control functions by standard TFI.

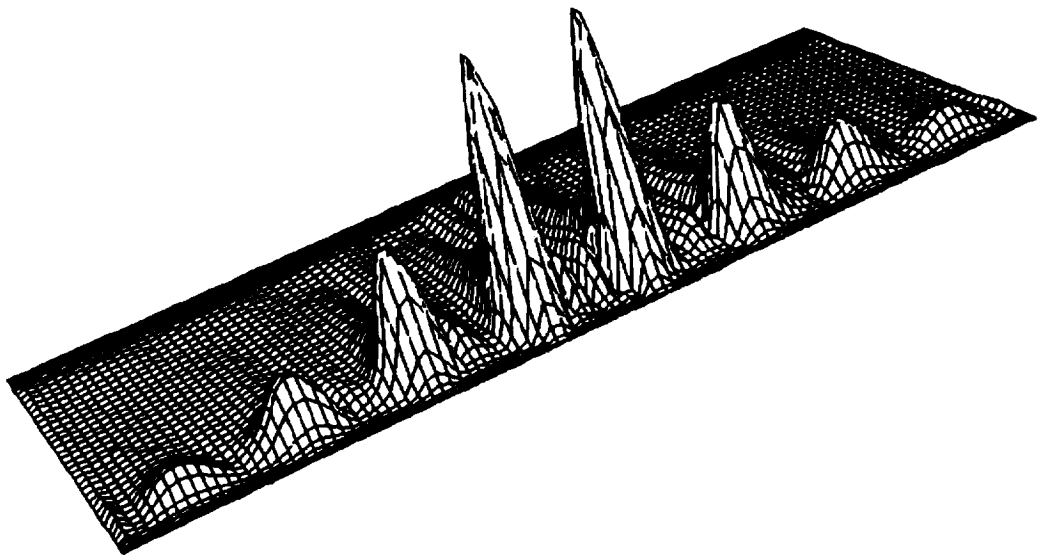

Figure 4. Surface error distribution on elliptic surface.

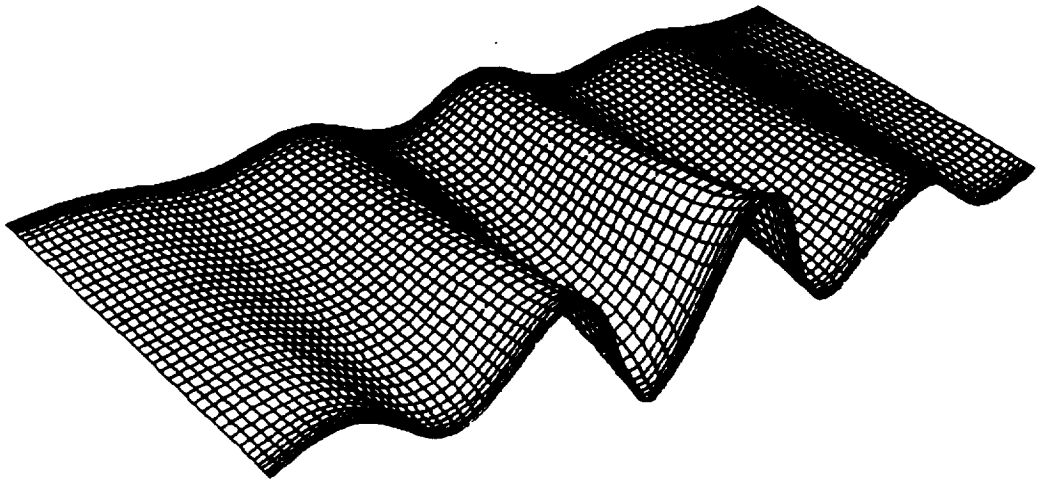

Figure 5. Elliptically generated surface after surface correction.

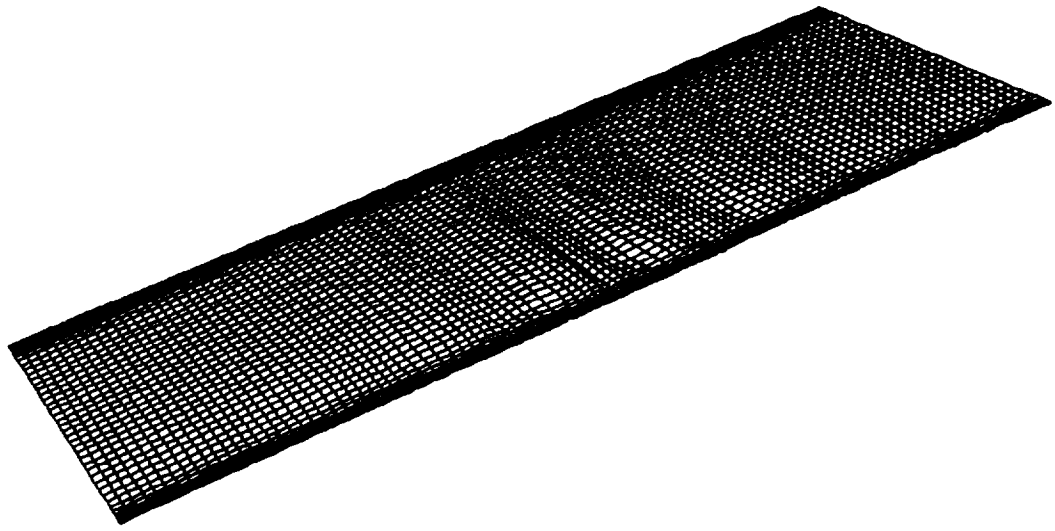

Figure 6. Surface error distribution after surface correction.

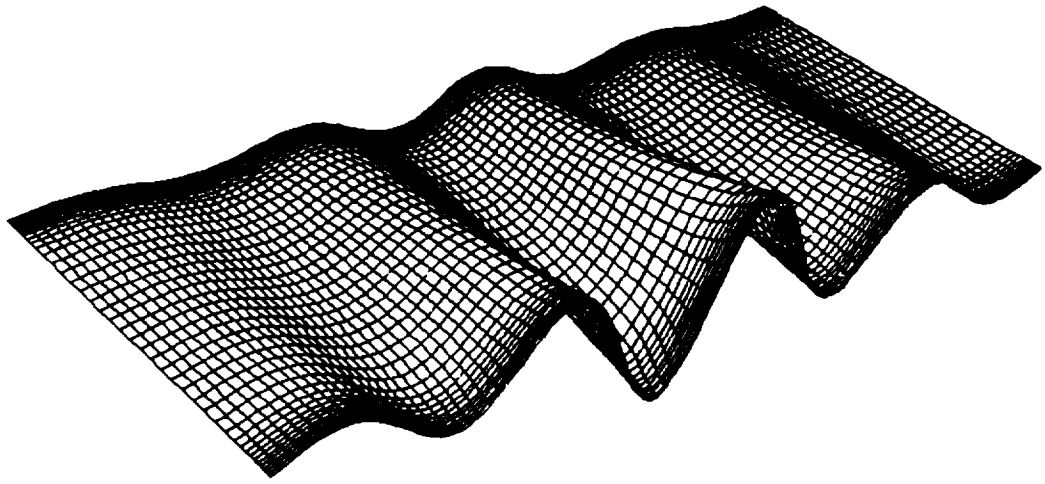

Figure 7. Elliptically generated surface using surface fitting of control functions.

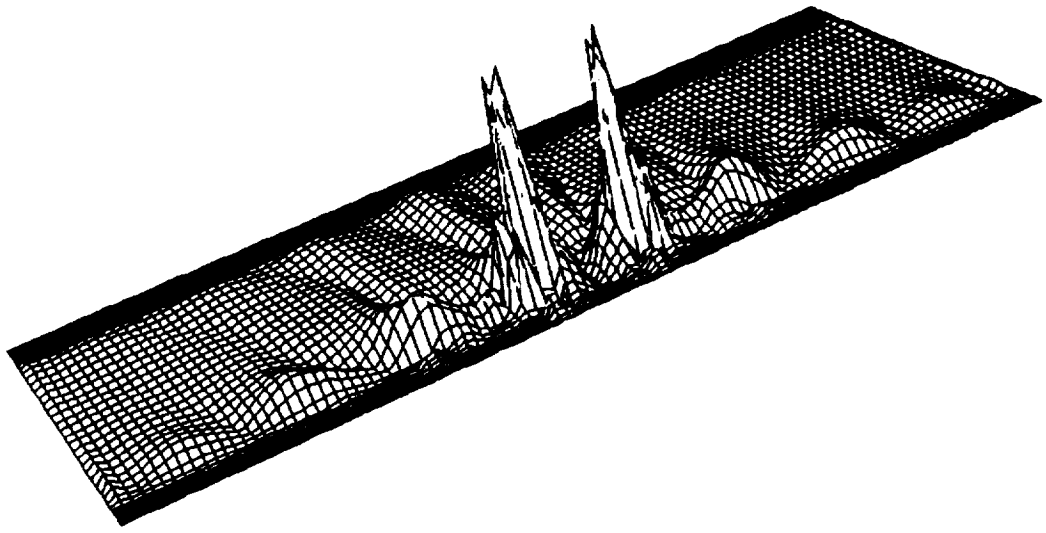

Figure 8. Surface error distribution on elliptic surface.

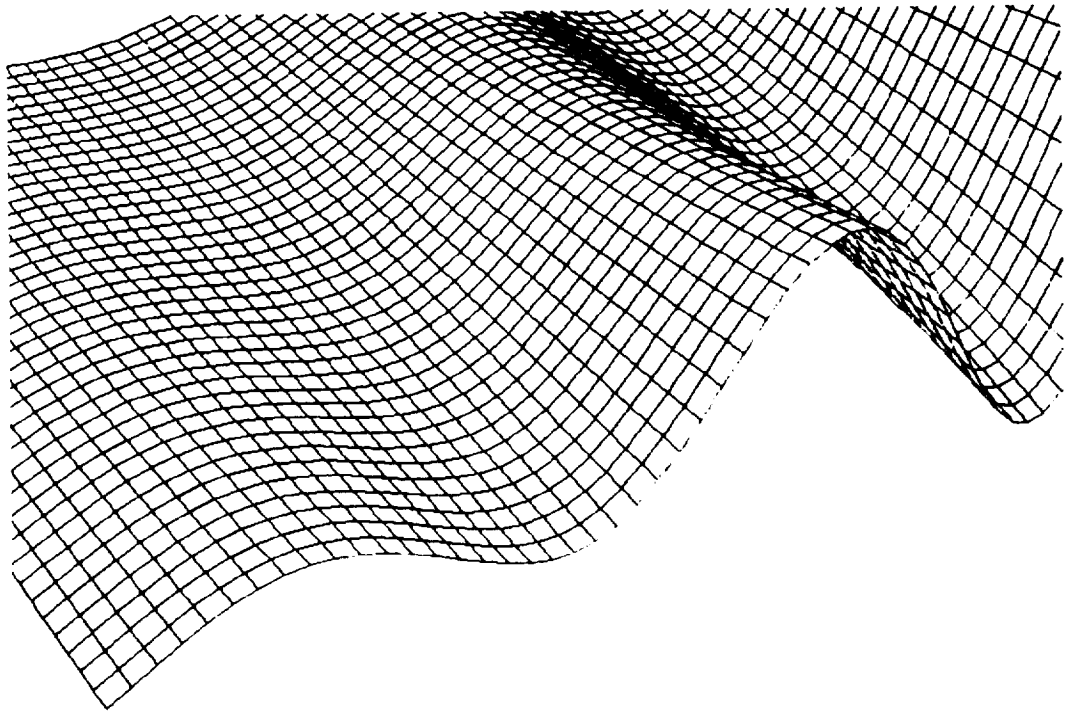

Figure 9a. Damped sine wave surface, near-edge view.

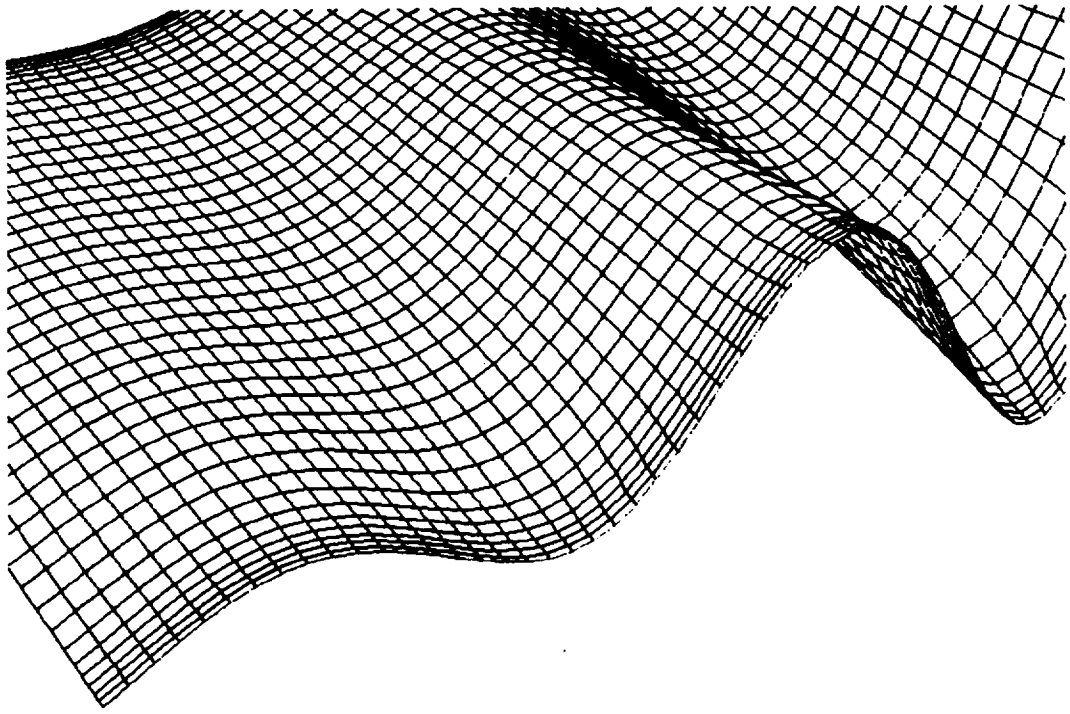

Figure 9b. Elliptic surface, near-edge view (control functions by standard TFI).

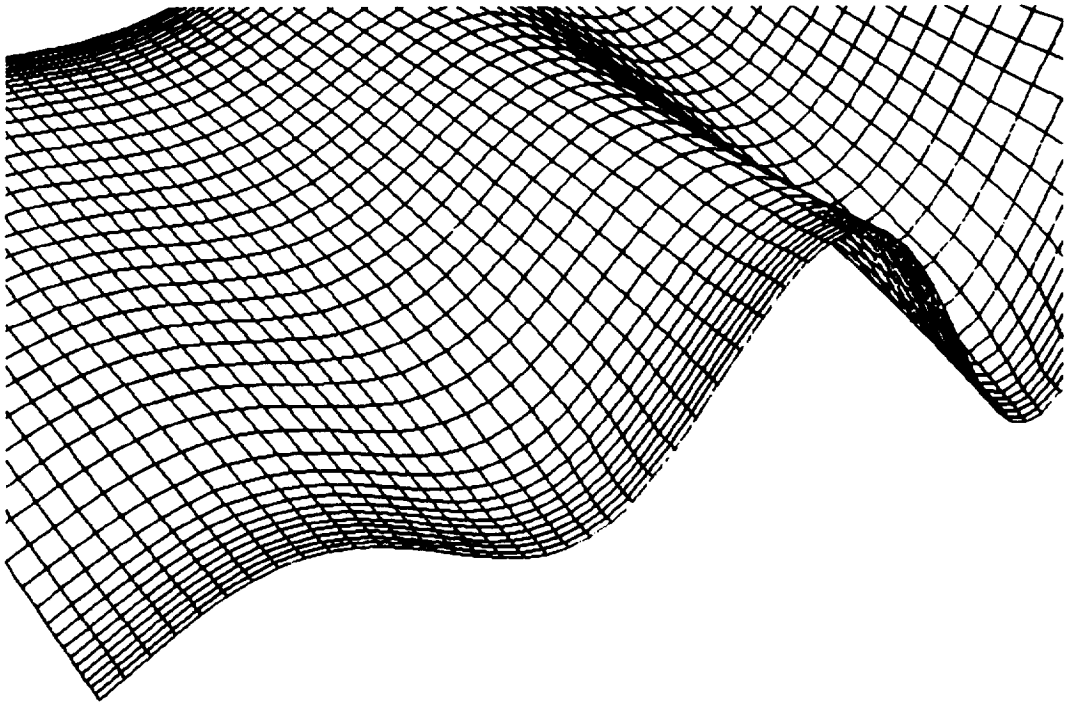

Figure 9c. Elliptic surface, near-edge view (surface fitting of control functions).

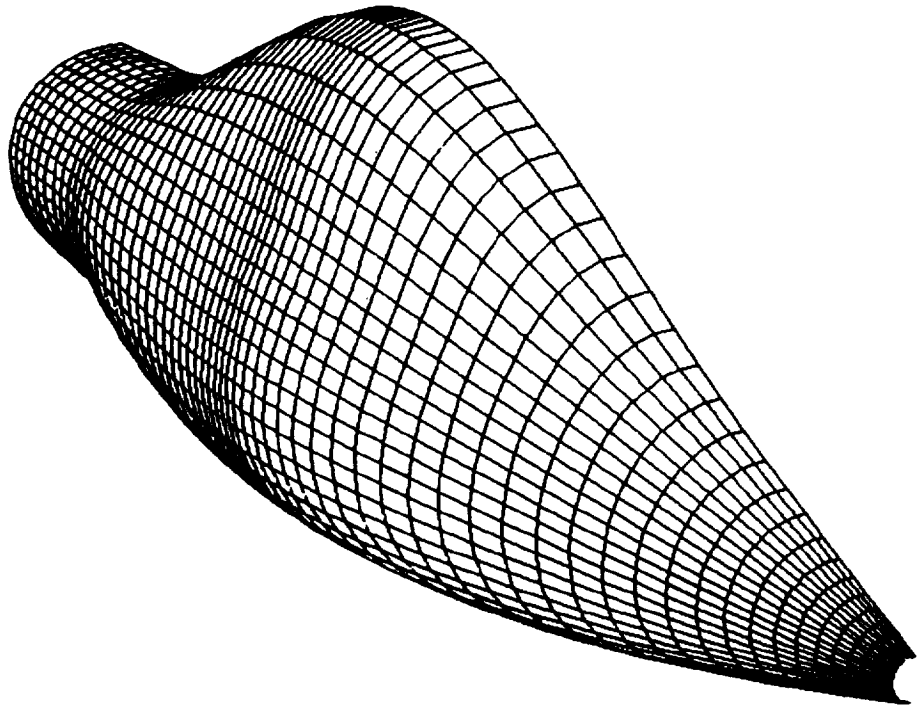

Figure 10. Generic fuselage forebody surface.

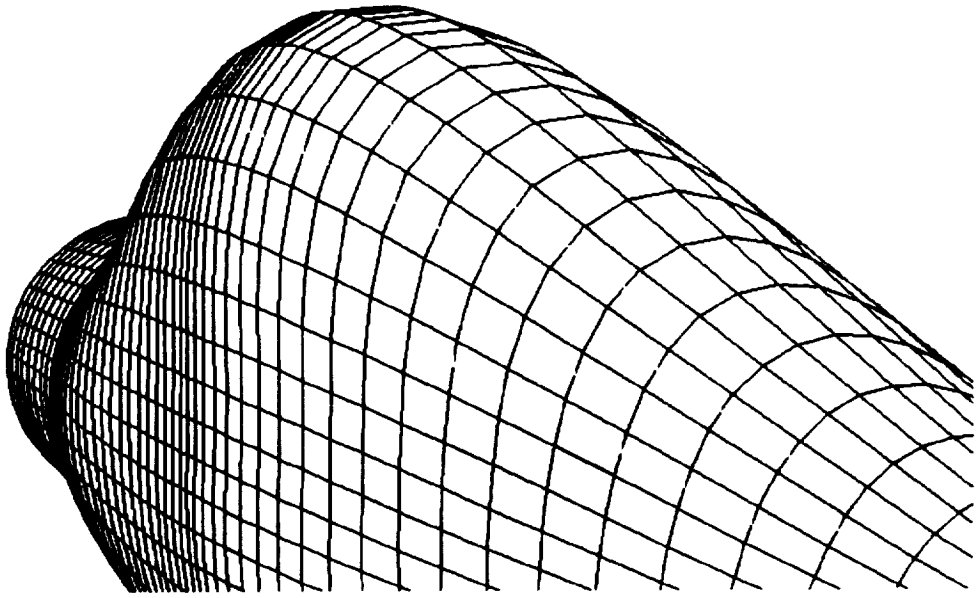

Figure 11a. Generic fuselage forebody surface, near-canopy view.

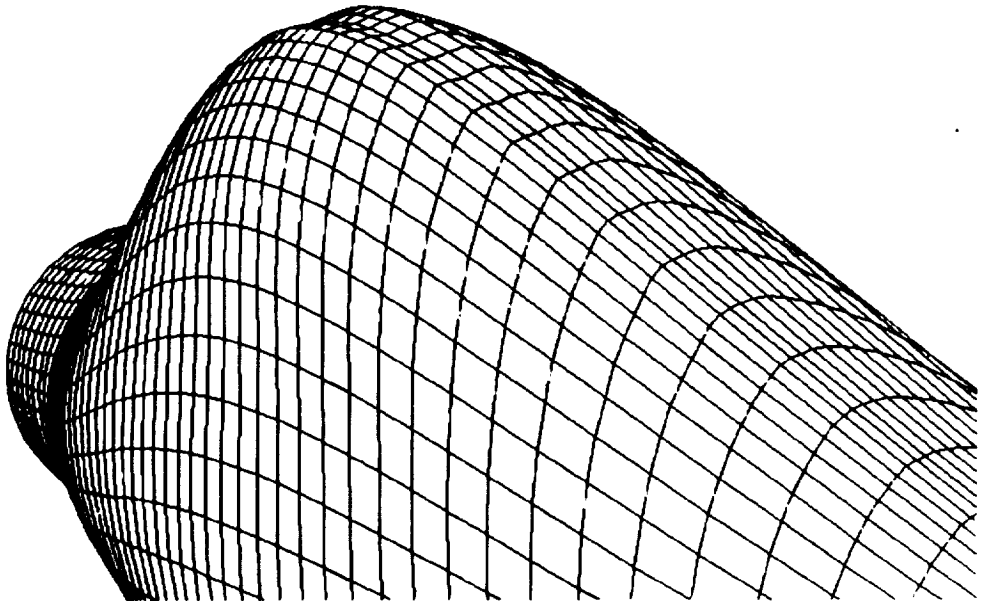

Figure 11b. Elliptic fuselage forebody surface, near-canopy view.

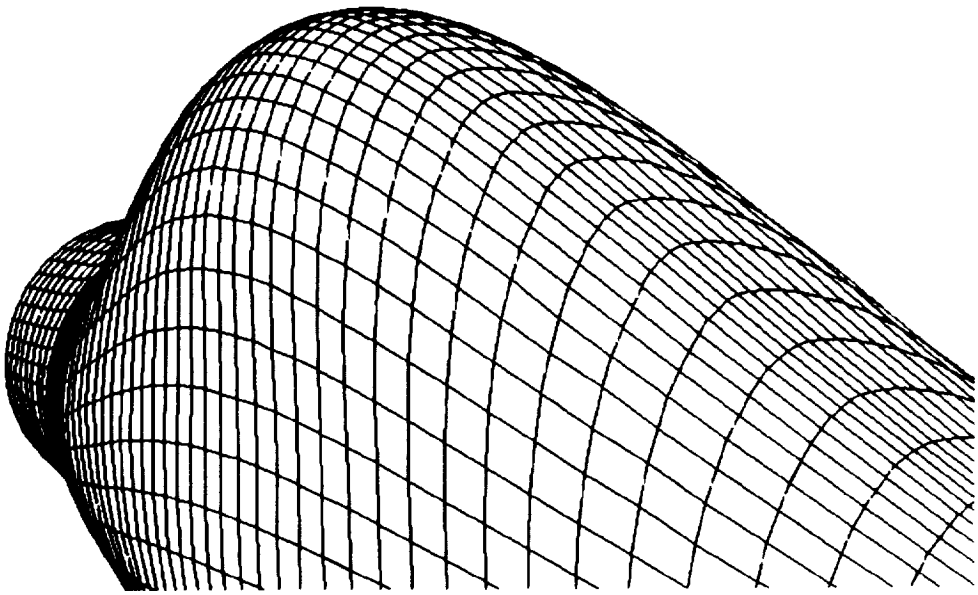

Figure 11c. Surface corrected elliptic fuselage forebody surface, near-canopy view.

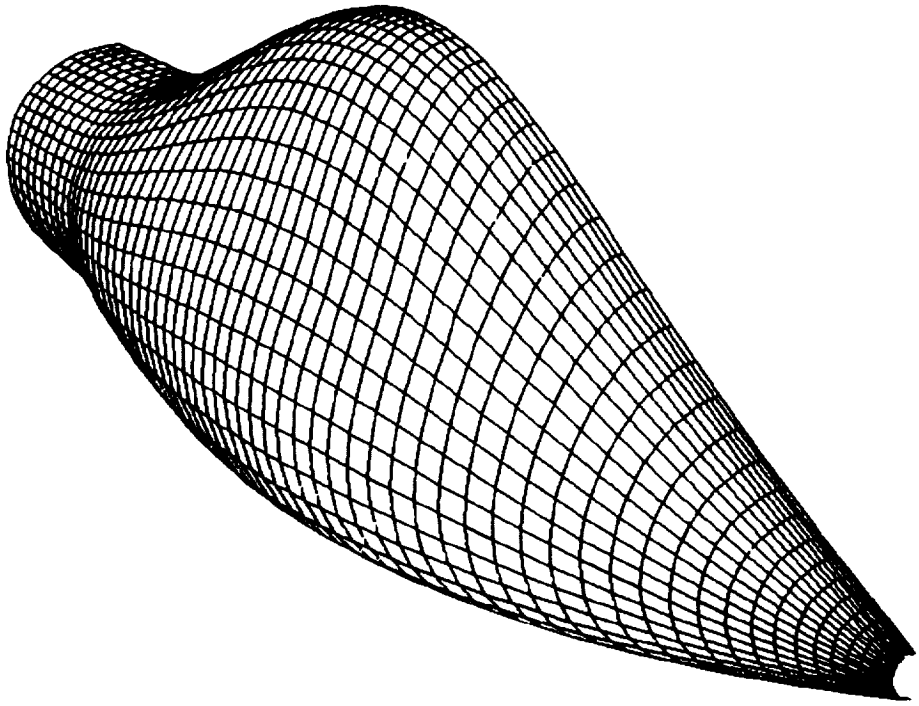

Figure 12. Final elliptic fuselage forebody surface.

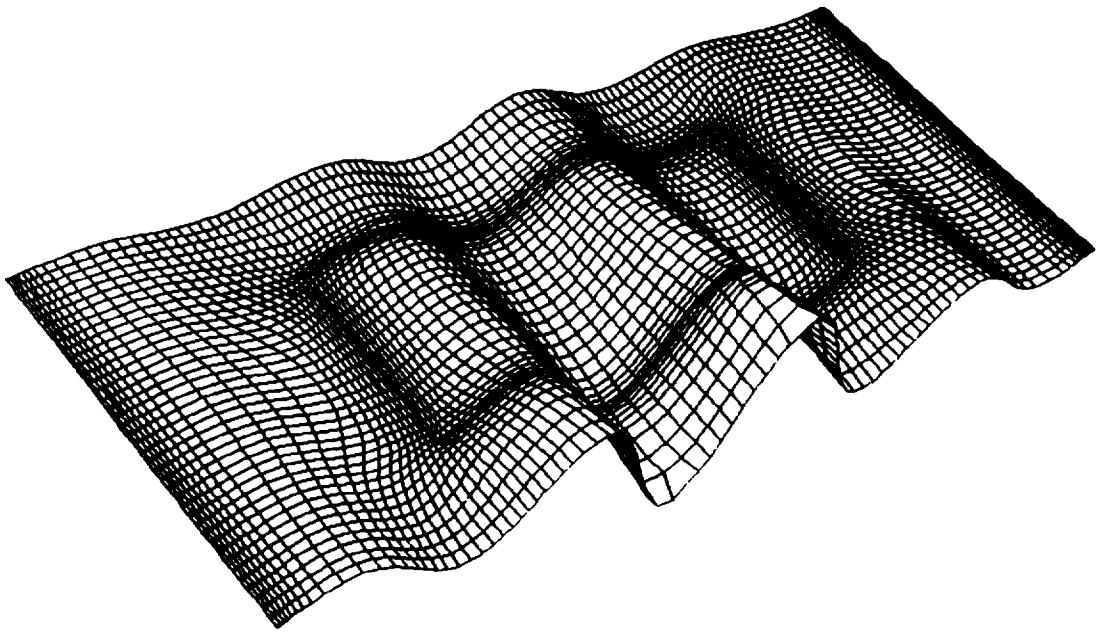

Figure 13. Elliptic surface with interior control option.