N95- 28760

Elliptic Surface Grid Generation on Minimal and Parametrized Surfaces [1]

S.P. Spekreijse, G.H. Nijhuis, J.W. Boerstoel
National Aerospace Laboratory NLR
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands

Summary

- An elliptic grid generation method is presented which generates excellent boundary conforming grids in domains in 2D physical space. The method is based on the composition of an algebraic and elliptic transformation. The composite mapping obeys the familiar Poisson grid generation system with control functions specified by the algebraic transformation. New expressions are given for the control functions. Grid orthogonality at the boundary is achieved by modification of the algebraic transformation.
- It is shown that grid generation on a minimal surface in 3D physical space is in fact equivalent to grid generation in a domain in 2D physical space.
- A second elliptic grid generation method is presented which generates excellent boundary conforming grids on smooth surfaces. It is assumed that the surfaces are parametrized and that the parametrization is a smooth mapping from a unit square onto the surface. A generated surface grid only depends on the shape of the surface and is independent of the parametrization.
- Concerning surface modeling, it is shown that bicubic Hermite interpolation is an excellent method to generate a smooth surface which is passing through a given discrete set of control points. In contrast to bicubic spline interpolation, there is extra freedom to model the tangent and twist vectors such that spurious oscillations are prevented.

1 Introduction

A flow simulation system for the computation of flows about complete aircraft configurations including propulsion aircraft components has been developed at NLR. The system is known as the ENFLOW (Euler/Navier-Stokes FLOW) system [1]. Fig.4 shows the layout of the system and summarizes its use for CFD work.

Surface modeling of the original aerodynamic input configuration surfaces is done with the commercial ICEM-CFD software. The subdivision of a three dimensional flow domain into blocks is done with the graphical interactive domain modeler ENDOMO. The computation of structured grids in the interior of the blocks is done with the graphical interactive grid generator ENGRID. Given a multi-block grid, the flow solver ENSOLV computes the solution of the Euler and/or Navier-Stokes equations with respect to specified boundary conditions.

In this paper we focus on surface grid generation. Output of ICEM-CFD is a set of discrete

[1]This investigation was partly performed under contract 01105N with the Netherlands Agency for Aerospace Programs (NIVR).

surfaces. A discrete surface is a two-dimensional array of so-called control points. The complete set of discrete surfaces approximates the original aerodynamic input configuration surfaces.

A discrete surface must be interpolated during grid generation. For this purpose, bicubic Hermite interpolation is used to define a smooth surface which is passing through the set of control points without introducing spurious oscillations. A C^1 parametrization is constructed which maps a unit square onto the interpolated surface. Bicubic Hermite interpolation is discussed in Section 5.

Multi-block grid generation proceeds from the inside out, starting with the generation of grids in edges, followed by the grid generation in faces, and ending with the grid generation in blocks. For this reason, a surface grid generation method is needed to generate interior grids in surfaces with a prescribed boundary grid point distribution as Dirichlet boundary condition.

In Section 4 it is shown how elliptic surface grid generation can be used to generate an interior surface grid on a parametrized surface with a prescribed boundary grid point distribution. A generated surface grid is independent of the parametrization. Thus the interior surface grid will only depend on the shape of the surface and the prescribed boundary grid points.

For surfaces in the interior of a flow domain, often only the boundary shape is defined. For a surface with only a prescribed boundary shape and a prescribed boundary grid point distribution, it is possible to generate an interior surface grid on a minimal surface. The shape of the surface becomes a soap film bounded by the prescribed boundary of the surface. It appears that surface grid generation on minimal surfaces is in fact a straightforward extension of grid generation in 2D.

Grid generation in a domain in 2D is treated in Section 2, the extension to minimal surface grid generation is discussed in Section 3.

Concerning grid generation, the main emphasis lies on the derivation of the elliptic grid generation systems. The discretization and solution of the systems of partial differential equations is not discussed. We refer to [2] for details about the applied solution methods. Grid generation in 3D domains is also discussed in [2].

2 2D grid generation

Consider a simply connected bounded domain $\mathcal{D}$ in two dimensional space with Cartesian coordinates $\vec{x} = (x, y)^T$. Suppose that $\mathcal{D}$ is bounded by four edges E_1, E_2, E_3, E_4. Let (E_1, E_2) and (E_3, E_4) be the two pairs of opposite edges as shown in Fig.1.

Introduce the parameter space $\mathcal{P}$ as the unit square in a two dimensional space with Cartesian coordinates $\vec{s} = (s, t)^T$. Require that the parameters s and t obey:

- $s \equiv 0$ at edge E_1 and $s \equiv 1$ at edge E_2, (1)
- s is the normalized arc length along edges E_3 and E_4, (2)
- $t \equiv 0$ at edge E_3 and $t \equiv 1$ at edge E_4, (3)
- t is the normalized arc length along edges E_1 and E_2. (4)

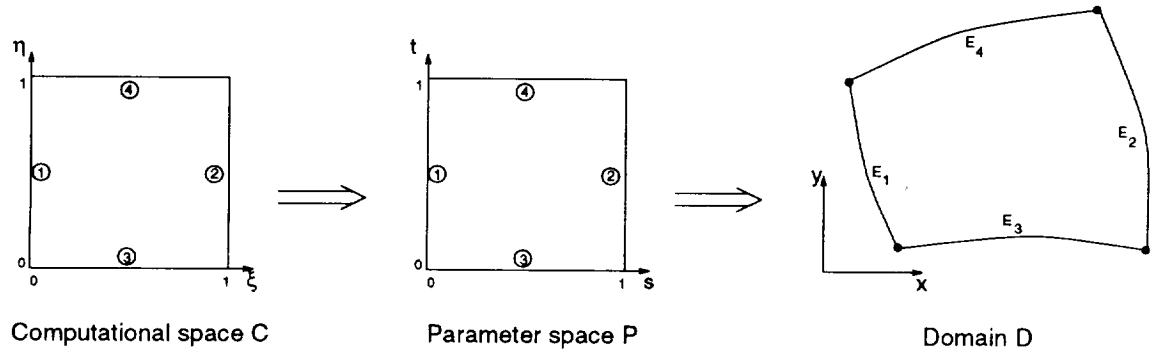

Figure 1: Transformation from computational (ξ, η) space to a domain $\mathcal{D}$ in Cartesian (x, y) space.

The mapping $\vec{s}: \partial\mathcal{D} \mapsto \partial\mathcal{P}$ is defined by these requirements. In the interior of $\mathcal{D}$ we require that s and t are harmonic functions of x and y, thus obey the Laplace equations:

$$\triangle s = \frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} = s_{xx} + s_{yy} = 0, \tag{5}$$

$$\triangle t = \frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} = t_{xx} + t_{yy} = 0. \tag{6}$$

The two Laplace equations $\triangle s = 0$ and $\triangle t = 0$, together with the above specified boundary conditions, define the mapping $\vec{s} : \mathcal{D} \mapsto \mathcal{P}$. Note that this mapping only depends on the shape of domain $\mathcal{D}$. By interchanging the dependent and independent variables, a non-linear elliptic partial differential equation can be derived for $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$. This mapping is called the elliptic transformation. It is well known that this mapping is differentiable and one-to-one [3].

Define the computational space $\mathcal{C}$ as the unit square in a two dimensional space with Cartesian coordinates $\vec{\xi} = (\xi, \eta)^T$. Assume that a mapping $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{D}$ is prescribed which maps the boundary of $\mathcal{C}$ one-to-one on the boundary of $\mathcal{D}$. This mapping defines the boundary grid point distribution. Assume that

- $\xi \equiv 0$ at edge E_1 and $\xi \equiv 1$ at edge E_2,
- $\eta \equiv 0$ at edge E_3 and $\eta \equiv 1$ at edge E_4.

We wish to construct a mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ which obeys the boundary conditions and which is a differentiable one-to-one mapping. Furthermore, we require that the interior grid point distribution describes a smooth transition between the prescribed grid point distribution in the four edges.

A natural mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ exists which obeys these requirements. This mapping will be the composition of an algebraic transformation and the elliptic transformation based on the Laplace equations. The algebraic transformation is a differentiable one-to-one mapping from computational space $\mathcal{C}$ onto the parameter space $\mathcal{P}$. The composition of these two mappings defines the interior grid point distribution and is a differentiable one-to-one mapping from computational domain $\mathcal{C}$ onto domain $\mathcal{D}$.

The algebraic transformation is defined as follows. Because $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{D}$ is prescribed and $\vec{x} : \partial\mathcal{P} \mapsto \partial\mathcal{D}$ is defined as described above, it follows that $\vec{s} : \partial\mathcal{C} \mapsto \partial\mathcal{P}$ is also defined.

From the preceding requirements it follows that

$$s(0,\eta) = 0 \ , \ s(1,\eta) = 1 \ , \ s(\xi,0) = s_{E_3}(\xi) \ , \ s(\xi,1) = s_{E_4}(\xi), \tag{7}$$

where the functions s_{E_3}, s_{E_4} are monotonically increasing, and

$$t(\xi,0) = 0 \ , \ t(\xi,1) = 1 \ , \ t(0,\eta) = t_{E_1}(\eta) \ , \ t(1,\eta) = t_{E_2}(\eta), \tag{8}$$

where the functions t_{E_1}, t_{E_2} are also monotonically increasing. Thus the four functions $t_{E_1}(\eta)$, $t_{E_2}(\eta)$, $s_{E_3}(\xi)$, $s_{E_4}(\xi)$ are defined by the boundary grid point distribution.

The mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is now defined by the following two algebraic equations:

$$s = s_{E_3}(\xi)(1-t) + s_{E_4}(\xi)t, \tag{9}$$

$$t = t_{E_1}(\eta)(1-s) + t_{E_2}(\eta)s. \tag{10}$$

Eq.(9) implies that a coordinate line ξ = constant is mapped to the parameter space $\mathcal{P}$ as a straight line: s is a linear function of t, and Eq.(10) implies that a grid line η = constant is also mapped to $\mathcal{P}$ as a straight line: t is a linear function of s. For given values of ξ and η, the corresponding s and t values are found as the intersection point of the two straight lines. For this reason, the system defined by Eqs.(9),(10) is called the "algebraic straight line transformation". It can be easily verified that this system defines a differentiable one-to-one mapping because of the positiveness of the Jacobian: $s_\xi t_\eta - s_\eta t_\xi > 0$.

In the remainder of this section, we will derive the set of non-linear elliptic partial differential equations which the composite mapping $\vec{x} = \vec{x}(\vec{s}(\vec{\xi}))$ has to fulfill. First introduce the two covariant base vectors

$$\vec{a}_1 = \frac{\partial\vec{x}}{\partial\xi} = \vec{x}_\xi \ , \ \vec{a}_2 = \frac{\partial\vec{x}}{\partial\eta} = \vec{x}_\eta, \tag{11}$$

and define the covariant metric tensor components as the inner product of the covariant base vectors

$$a_{ij} = (\vec{a}_i, \vec{a}_j) \ , \ i = \{1,2\} \ , \ j = \{1,2\}. \tag{12}$$

Then the contravariant base vectors $\vec{a}^1$ and $\vec{a}^2$ are defined according to the rules

$$(\vec{a}^i, \vec{a}_j) = \delta^i_j \ , \ i = \{1,2\} \ , \ j = \{1,2\}, \tag{13}$$

with δ^i_j the Kronecker symbol. Define the contravariant metric tensor components

$$a^{ij} = (\vec{a}^i, \vec{a}^j) \ , \ i = \{1,2\} \ , \ j = \{1,2\}, \tag{14}$$

so that

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} a^{11} & a^{12} \\ a^{12} & a^{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{15}$$

and

$$\vec{a}^1 = a^{11}\vec{a}_1 + a^{12}\vec{a}_2 \ , \ \vec{a}^2 = a^{12}\vec{a}_1 + a^{22}\vec{a}_2. \tag{16}$$

Introduce the determinant J^2 of the covariant metric tensor: $J^2 = a_{11}a_{22} - a_{12}^2$.

Now consider an arbitrary function $\phi = \phi(\xi, \eta)$. Then ϕ is also defined in domain $\mathcal{D}$ and the Laplacian of ϕ is expressed as

$$\triangle\phi = \phi_{xx} + \phi_{yy} = \frac{1}{J}\left\{\left(Ja^{11}\phi_\xi + Ja^{12}\phi_\eta\right)_\xi + \left(Ja^{12}\phi_\xi + Ja^{22}\phi_\eta\right)_\eta\right\}, \tag{17}$$

which may be found in every textbook on Tensor Analysis and Differential Geometry (for example see [4], page 227). Take as special cases respectively $\phi \equiv \xi$ and $\phi \equiv \eta$. Then Eq.(17) yields

$$\triangle\xi = \frac{1}{J}\left\{\left(Ja^{11}\right)_\xi + \left(Ja^{12}\right)_\eta\right\}\ ,\ \triangle\eta = \frac{1}{J}\left\{\left(Ja^{12}\right)_\xi + \left(Ja^{22}\right)_\eta\right\}. \tag{18}$$

Thus the Laplacian of ϕ can also be expressed as

$$\triangle\phi = a^{11}\phi_{\xi\xi} + 2a^{12}\phi_{\xi\eta} + a^{22}\phi_{\eta\eta} + \triangle\xi\ \phi_\xi + \triangle\eta\ \phi_\eta. \tag{19}$$

Substitution of respectively $\phi \equiv s$ and $\phi \equiv t$ in this equation yields

$$\triangle s = a^{11}s_{\xi\xi} + 2a^{12}s_{\xi\eta} + a^{22}s_{\eta\eta} + \triangle\xi\ s_\xi + \triangle\eta\ s_\eta, \tag{20}$$

$$\triangle t = a^{11}t_{\xi\xi} + 2a^{12}t_{\xi\eta} + a^{22}t_{\eta\eta} + \triangle\xi\ t_\xi + \triangle\eta\ t_\eta. \tag{21}$$

Using these equations and the requirement that s and t are harmonic in domain $\mathcal{D}$, thus $\triangle s = 0$ and $\triangle t = 0$, we find the following expressions for the Laplacian of ξ and η

$$\begin{pmatrix} \triangle\xi \\ \triangle\eta \end{pmatrix} = a^{11}\vec{P}_{11} + 2a^{12}\vec{P}_{12} + a^{22}\vec{P}_{22}, \tag{22}$$

where

$$\vec{P}_{11} = -T^{-1}\begin{pmatrix} s_{\xi\xi} \\ t_{\xi\xi} \end{pmatrix}\ ,\ \vec{P}_{12} = -T^{-1}\begin{pmatrix} s_{\xi\eta} \\ t_{\xi\eta} \end{pmatrix}\ ,\ \vec{P}_{22} = -T^{-1}\begin{pmatrix} s_{\eta\eta} \\ t_{\eta\eta} \end{pmatrix}, \tag{23}$$

and the matrix T is defined as

$$T = \begin{pmatrix} s_\xi & s_\eta \\ t_\xi & t_\eta \end{pmatrix}. \tag{24}$$

The six coefficients of the vectors $\vec{P}_{11} = (P_{11}^1, P_{11}^2)^T$, $\vec{P}_{12} = (P_{12}^1, P_{12}^2)^T$ and $\vec{P}_{22} = (P_{22}^1, P_{22}^2)^T$ are so called control functions. These six control functions are completely defined and easily computed for a given algebraic transformation $\vec{s} = \vec{s}(\vec{\xi})$. Different and less useful expressions of these control functions can also be found in [5, 6].

Finally, substitution of $\phi \equiv \vec{x}$ in Eq.(19) yields

$$\triangle\vec{x} = a^{11}\vec{x}_{\xi\xi} + 2a^{12}\vec{x}_{\xi\eta} + a^{22}\vec{x}_{\eta\eta} + \triangle\xi\ \vec{x}_\xi + \triangle\eta\ \vec{x}_\eta. \tag{25}$$

Substitution of Eq.(22) into this equation and using the fact that $\triangle\vec{x} \equiv 0$ we arrive at the following Poisson grid generation system

$$\begin{aligned} a^{11}\vec{x}_{\xi\xi} + 2a^{12}\vec{x}_{\xi\eta} + a^{22}\vec{x}_{\eta\eta} &+ \left(a^{11}P_{11}^1 + 2a^{12}P_{12}^1 + a^{22}P_{22}^1\right)\vec{x}_\xi \\ &+ \left(a^{11}P_{11}^2 + 2a^{12}P_{12}^2 + a^{22}P_{22}^2\right)\vec{x}_\eta = 0. \end{aligned} \tag{26}$$

Using Eqs.(12),(15), we find the following well known expressions for the contravariant metric tensor components:

$$J^2 a^{11} = a_{22} = (\vec{x}_\eta, \vec{x}_\eta)\ ,\ J^2 a^{12} = -a_{12} = -(\vec{x}_\xi, \vec{x}_\eta)\ ,\ J^2 a^{22} = a_{11} = (\vec{x}_\xi, \vec{x}_\xi). \tag{27}$$

Thus the Poisson grid generation system defined by Eq.(26) can be simplified by multiplication with J^2. Then we obtain:

$$\begin{aligned} a_{22}\vec{x}_{\xi\xi} - 2a_{12}\vec{x}_{\xi\eta} + a_{11}\vec{x}_{\eta\eta} \quad &+ \quad \left(a_{22}P_{11}^1 - 2a_{12}P_{12}^1 + a_{11}P_{22}^1\right)\vec{x}_\xi \\ &+ \quad \left(a_{22}P_{11}^2 - 2a_{12}P_{12}^2 + a_{11}P_{22}^2\right)\vec{x}_\eta = 0. \end{aligned} \tag{28}$$

This equation, together with the expressions for the control functions P_{ij}^k given by Eq.(23), forms our 2D grid generation system. Grids are computed by solving this quasi-linear system of elliptic partial differential equations.

Orthogonality at boundaries

Grid orthogonality at boundaries can be achieved as follows.

Redefine the elliptic transformation $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ by imposing the following new set of boundary conditions for the harmonic functions s and t:

- $s \equiv 0$ at edge E_1 and $s \equiv 1$ at edge E_2,
- $\frac{\partial s}{\partial n} = 0$ along edges E_3 and E_4, where n is the outward normal direction,
- $t \equiv 0$ at edge E_3 and $t \equiv 1$ at edge E_4,
- $\frac{\partial t}{\partial n} = 0$ along edges E_1 and E_2, where n is the outward normal direction.

These new boundary conditions define a new mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$.

The Neumann boundary condition $\frac{\partial s}{\partial n} = 0$ along edges E_3 and E_4 imply that a parameter line $s =$ constant is a curve in domain $\mathcal{D}$ which is orthogonal at those edges. Similarly, a parameter line $t =$ constant is a curve in $\mathcal{D}$ which is orthogonal at edge E_1 and edge E_2.

The algebraic transformation $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is redefined according to the following two algebraic equations:

$$s = s_{E_3}(\xi)H_0(t) + s_{E_4}(\xi)H_1(t), \tag{29}$$

$$t = t_{E_1}(\eta)H_0(s) + t_{E_2}(\eta)H_1(s), \tag{30}$$

where H_0 and H_1 are cubic Hermite interpolation functions defined in Eq.(52) below. Note that $H_0(0) = 1$, $H_0'(0) = 0$, $H_0(1) = 0$, $H_0'(1) = 0$ and $H_1(0) = 0$, $H_1'(0) = 0$, $H_1(1) = 1$, $H_1'(1) = 0$. It follows from Eq.(29) that a coordinate line $\xi =$ constant is mapped to parameter space $\mathcal{P}$ as a cubic curve which is orthogonal at both edge E_3 and edge E_4 in $\mathcal{P}$. Such a curve in parameter space $\mathcal{P}$ will thus be mapped by the new elliptic transformation $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ as a curve which is orthogonal at both edge E_3 and edge E_4 in $\mathcal{D}$. Similar observations can be made for coordinate lines $\eta =$ constant. Thus the grid will be orthogonal at all four edges in domain $\mathcal{D}$.

The composite mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ still obeys the Poisson grid generation system defined by Eq.(28). Thus the same system of elliptic equations can be solved to generate an orthogonal grid at the boundary. The only difference is that now $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is defined by Eqs.(29),(30) instead of Eqs.(9),(10).

Figs.5,6,7 are demonstrations of the robustness of this elliptic grid generation method. The boundary grid point distribution is prescribed and the interior grids are obtained by solving Eq.(28). The interior grid point distributions were verified to be foldfree by zooming into regions where the grid is very dense.

3 Surface Grid Generation on Minimal Surfaces

Grid generation on a minimal surface in 3D physical space is in fact equivalent to grid generation in a domain in 2D physical space.

As in the two dimensional case, again consider four curved edges E_1, E_2, E_3, E_4 but now situated in the three dimensional physical space with Cartesian coordinates $\vec{x} = (x, y, z)^T$. Let (E_1, E_2) and (E_3, E_4) be the two pairs of opposite edges as shown in Fig.2.

Introduce the parameter space $\mathcal{P}$ as the unit square in a two dimensional space with Cartesian coordinates $\vec{s} = (s, t)^T$. Again require that the parameters s and t obey the boundary equations specified in Eqs.(1),(2),(3),(4). Furthermore, require that

$$\triangle s = 0, \tag{31}$$

$$\triangle t = 0, \tag{32}$$

$$H = 0, \tag{33}$$

where $\triangle$ is the Laplace-Beltrami operator for surfaces and H is the mean curvature.

These three requirements, together with the specified boundary conditions define a unique mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{R}^3$. The shape of the surface defined by this mapping is a minimal surface (soap film) because of the requirement that the mean curvature H is zero. The parametrization of the surface is defined by Eqs.(31),(32). Define the minimal surface $\mathcal{S} = \{\vec{x}(s, t) \mid (s, t) \in \mathcal{P}\}$.

Consider a prescribed boundary grid point distribution at the four edges E_1, E_2, E_3, E_4 of the minimal surface $\mathcal{S}$. The boundary grid point distribution can be defined as a mapping $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{S}$ where $\mathcal{C}$ is the computational space defined as the unit square in a two dimensional space with Cartesian coordinates $\vec{\xi} = (\xi, \eta)^T$. Because $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{S}$ is prescribed and $\vec{x} : \partial\mathcal{P} \mapsto \partial\mathcal{S}$ is defined as described above, it follows that $\vec{s} : \partial\mathcal{C} \mapsto \partial\mathcal{P}$ is also defined.

In exactly the same way as for the two dimensional case, the mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is defined by the algebraic straight line transformation defined by Eqs.(9),(10). The mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{S}$ is defined by Eqs.(31),(32),(33). The composite mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{S}$ is defined as $\vec{x} = \vec{x}(\vec{s}(\vec{\xi}))$ and describes the interior grid point distribution on the minimal surface $\mathcal{S}$. Note that this composite mapping will be differentiable and one-to-one.

We will now show that the set of non-linear elliptic partial differential equations which the composite mapping has to fulfill is the same Poisson system as defined by Eq.(28) but with $\vec{x} = (x, y, z)^T$ instead of $\vec{x} = (x, y)^T$. Thus grid generation on a minimal surface in 3D physical space is in fact

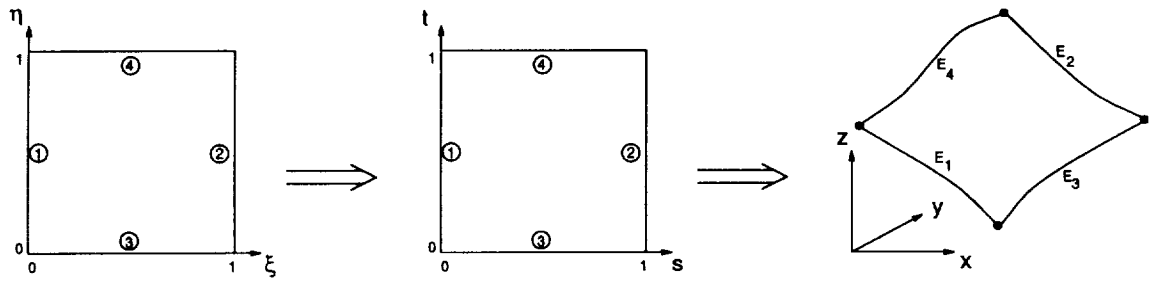

Figure 2: Transformation from computational (ξ, η) space to a minimal surface $\mathcal{S}$ in Cartesian (x, y, z) space.

equivalent to grid generation in a domain in 2D physical space. The result that a Poisson system of the form as defined by Eq.(28) can be used to compute a grid on a minimal surface can also be found as a special application of the formulas derived in [7].

For this purpose, introduce the two covariant base vectors

$$\vec{a}_1 = \vec{x}_\xi \ , \ \vec{a}_2 = \vec{x}_\eta. \tag{34}$$

The two covariant base vectors span the tangent plane of $\mathcal{S}$ at the corresponding point P. Define the unit surface normal as

$$\vec{n} = \frac{\vec{a}_1 \wedge \vec{a}_2}{\| \vec{a}_1 \wedge \vec{a}_2 \|}, \tag{35}$$

where $\wedge$ is the vector product operator. The contravariant base vectors $\vec{a}^1$ and $\vec{a}^2$ are defined according to the rules

$$(\vec{a}^i, \vec{a}_j) = \delta^i_j \ , \ i = \{1,2\} \ , \ j = \{1,2\}, \tag{36}$$

and

$$(\vec{a}^1, \vec{n}) = 0 \ , \ (\vec{a}^2, \vec{n}) = 0. \tag{37}$$

Thus the two contravariant base vectors are also lying in the tangent plane of $\mathcal{S}$ at the corresponding point P. Define the covariant metric tensor components by Eq.(12) and the contravariant metric tensor components by Eq.(14). Then Eqs.(15),(16) are still valid. Again introduce the determinant J^2 of the covariant metric tensor: $J^2 = a_{11}a_{22} - a_{12}^2$.

Now consider an arbitrary function $\phi = \phi(\xi, \eta)$. Then ϕ is also defined on surface $\mathcal{S}$ and the Laplace-Beltrami operator of ϕ is expressed as

$$\triangle\phi = \frac{1}{J}\left\{\left(Ja^{11}\phi_\xi + Ja^{12}\phi_\eta\right)_\xi + \left(Ja^{12}\phi_\xi + Ja^{22}\phi_\eta\right)_\eta\right\} \tag{38}$$

(see [4], page 227). As in the two-dimensional case, substitution of $\phi \equiv \xi$ and $\phi \equiv \eta$ into this equation yields Eq.(18). Thus the Laplace-Beltrami operator of ϕ can also be expressed as defined by Eq.(19). Substitution of respectively $\phi \equiv s$ and $\phi \equiv t$ in Eq.(19) and using the requirements expressed by Eqs.(31),(32) yields exactly the same expressions for $\triangle\xi$ and $\triangle\eta$ given by Eqs.(22),(23). Finally, substitution of $\phi \equiv \vec{x}$ in Eq.(19) yields Eq.(25).

The Laplace-Beltrami operator applied on $\vec{x}$ obeys a famous relation expressed by

$$\triangle \vec{x} = 2H\vec{n}, \tag{39}$$

where the mean curvature H is defined as

$$H = \frac{1}{2}\left(a^{11}\vec{x}_{\xi\xi} + 2a^{12}\vec{x}_{\xi\eta} + a^{22}\vec{x}_{\eta\eta}, \vec{n}\right). \tag{40}$$

(for example see [8], Theorem 1, page 71). Using the requirement $H = 0$ yields

$$\triangle \vec{x} = 0. \tag{41}$$

Thus Eq.(22) and Eq.(25) with $\triangle \vec{x} = 0$ are also valid for minimal surfaces. Following the same derivation as given in Section 2, we arrive at exactly the same non-linear system of elliptic partial differential equations as expressed by Eq.(28). Thus an interior grid point distribution on a minimal surface is found by solving Eq.(28). The only difference compared to the two dimensional case is that now $\vec{x} = (x, y, z)^T$ instead of $\vec{x} = (x, y)^T$.

Grid orthogonality at boundaries can be obtained in the same way as described in Section 2.

One may ask whether it is useful to implement a method to compute grids on minimal surfaces in a 3D multi-block grid generator code. The answer is yes. Minimal surfaces may be used to define the geometry and grid for a block-face of which only the four face-edges are given. It is also possible to apply minimal surface grid generation when a grid must be generated in a block-face with four face-edges lying in a plane. Then the minimal surface is a plane surface bounded by the four edges. The grids in the 2D domains depicted in Figs.5,6,7 were generated in this way and are in fact grids on minimal surfaces.

An example of a grid on a characteristic minimal surface is shown in Fig.8. This is a so-called square Scherck surface [8]. Fig.9 illustrates what happens when the prescribed boundary grid point distribution is changed. This figure clearly shows that the shape of the minimal surface is independent of the prescribed boundary grid point distribution.

4 Surface Grid Generation on Parametrized Surfaces

In this section we develop a method to generate a grid on a parametrized surface which is independent of the parametrization. A generated grid only depends on the shape of the surface and the prescribed boundary grid point distribution at the four edges of the surface.

Consider a bounded surface $\mathcal{S}$ with a prescribed geometrical shape in three dimensional physical space with Cartesian coordinates $\vec{x} = (x, y, z)^T$. Assume that $\mathcal{S}$ is parametrized by a differentiable one-to-one mapping

$$\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}, \tag{42}$$

where $\mathcal{P}_{uv}$ is the unit square in two dimensional space with Cartesian coordinates $\vec{u} = (u, v)^T$. Define the four edges E_1, E_2, E_3, E_4 of surface $\mathcal{S}$ by

- $u \equiv 0$ at edge E_1 and $u \equiv 1$ at edge E_2,

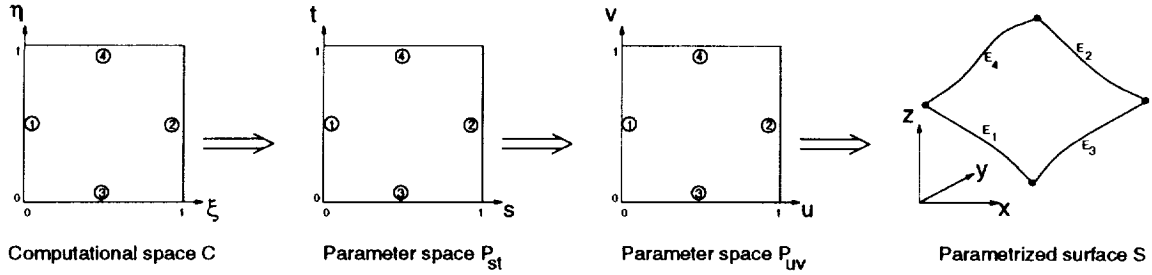

Figure 3: Transformation from computational (ξ, η) space to a parametrized surface $\mathcal{S}$ in Cartesian (x, y, z) space.

- $v \equiv 0$ at edge E_3 and $v \equiv 1$ at edge E_4.

Thus (E_1, E_2) and (E_3, E_4) are the two pairs of opposite edges of surface $\mathcal{S}$ as shown in Fig.3. Introduce the parameter space $\mathcal{P}_{st}$ as the unit square in a two dimensional space with Cartesian coordinates $\vec{s} = (s,t)^T$. Again require that the parameters s and t obey the boundary equations specified in Eqs.(1),(2),(3),(4). Furthermore, require that $\triangle s = 0$ and $\triangle t = 0$ where $\triangle$ is the Laplace-Beltrami operator for surfaces. Hence the parameters s and t obey

$$\left(Ja^{11}s_u + Ja^{12}s_v\right)_u + \left(Ja^{12}s_u + Ja^{22}s_v\right)_v = 0, \tag{43}$$

$$\left(Ja^{11}t_u + Ja^{12}t_v\right)_u + \left(Ja^{12}t_u + Ja^{22}t_v\right)_v = 0, \tag{44}$$

where a^{ij} are the contravariant tensor components and J^2 is defined as the determinant of the covariant metric tensor. The contravariant tensor components a^{ij} are related to the covariant tensor components a_{ij} according to Eq.(15). The covariant metric tensor components are defined by Eq.(12), where the two covariant base vectors are now given by

$$\vec{a}_1 = \vec{x}_u \ , \ \vec{a}_2 = \vec{x}_v \ . \tag{45}$$

The coefficients Ja^{11}, Ja^{12} and Ja^{22} in Eqs.(43),(44) are thus functions of u and v, and Eqs.(43),(44) are therefore two uncoupled second-order linear partial differential equations for the functions $s = s(u,v)$ and $t = t(u,v)$.

Each boundary point of surface $\mathcal{S}$ has a unique (s,t) parameter value at $\partial\mathcal{P}_{st}$ and a unique (u,v) parameter value at $\partial\mathcal{P}_{uv}$. Thus each (u,v) parameter value at $\partial\mathcal{P}_{uv}$ has also a unique (s,t) parameter value at $\partial\mathcal{P}_{st}$. Thus the functions s and t are prescribed at the boundary of $\mathcal{P}_{uv}$. Hence, Eq.(43) together with the Dirichlet boundary conditions for s can be used to compute $s = s(u,v)$ and Eq.(44) together with the Dirichlet boundary conditions for t can be used to compute $t = t(u,v)$. Only two linear partial differential equations have to be solved to define these mappings. These two mappings are compactly written as $\vec{s} : \mathcal{P}_{uv} \mapsto \mathcal{P}_{st}$. Note that $\vec{s} : \mathcal{P}_{uv} \mapsto \mathcal{P}_{st}$ is a differentiable one-to-one mapping so that the inverse mapping $\vec{u} : \mathcal{P}_{st} \mapsto \mathcal{P}_{uv}$ also exists.

Thus the composite mapping $\vec{x} : \mathcal{P}_{st} \mapsto \mathcal{S}$, defined as $\vec{x} = \vec{x}(\vec{u}(\vec{s}))$ also exists and is differentiable and one-to-one. Note that this mapping $\vec{x} : \mathcal{P}_{st} \mapsto \mathcal{S}$ only depends on the shape of surface $\mathcal{S}$ and is independent of the original parametrization $\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}$. The mapping $\vec{x} : \mathcal{P}_{st} \mapsto \mathcal{S}$ may thus be considered as a property of surface $\mathcal{S}$ and defines a new unique parametrization of $\mathcal{S}$.

Consider a prescribed boundary grid point distribution at the four edges E_1, E_2, E_3, E_4. The boundary grid point distribution can be defined as a mapping $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{S}$ where $\mathcal{C}$ is the computational space defined as the unit square in a two dimensional space with Cartesian coordinates $\vec{\xi} = (\xi, \eta)^T$. Because $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{S}$ is prescribed and $\vec{x} : \partial\mathcal{P}_{st} \mapsto \partial\mathcal{S}$ is defined as described above, it follows that $\vec{s} : \partial\mathcal{C} \mapsto \partial\mathcal{P}_{st}$ is also defined.

In exactly the same way as for the two dimensional case, the mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}_{st}$ is now defined by the algebraic straight line transformation defined by Eqs.(9),(10). The composition of the mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}_{st}$ and the mapping $\vec{x} : \mathcal{P}_{st} \mapsto \mathcal{S}$ defines $\vec{x} : \mathcal{C} \mapsto \mathcal{S}$ and describes the interior grid point distribution on surface $\mathcal{S}$. Note that this composite mapping will also be differentiable and one-to-one.

Grid orthogonality at boundaries can be obtained by the same procedure as described in Section 2 for 2D domains.

Fig.10 shows an irregularly distributed control point mesh on a smooth surface. The surface is defined as $z = \frac{1}{8} tanh(15(\frac{1}{4} - (x-1)^2 - (y-1)^2)), (x, y) \in [0, 1]^2$. In section 5.2 is described how bicubic Hermite interpolation is used to define the mapping $\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}$. The parametrization depends on the control point distribution. Fig.11 shows an elliptic grid. Equidistributed boundary grid points are used as Dirichlet boundary condition. This figure clearly demonstrates that the interior surface grid only depends on the shape of the surface and is independent of the parametrization.

Less academic surface grids are shown in Fig.13.

5 Surface modeling

Consider a discrete surface defined as a two-dimensional array of control points. An interpolated surface is obtained by bicubic Hermite interpolation. Non-linear averaging formulas for the tangent and twist vectors are used to prevent spurious oscillations. The interpolation method is explained first for curves and then extended to surfaces.

5.1 Piecewise cubic Hermite interpolation for curves

Consider a set of control points $\{\vec{x}_i = (x, y, z)_i^T \mid i = 0 \ldots N\}$. We wish to construct a smooth C^1 curve $\vec{x} : u \in [0, 1] \mapsto \mathcal{R}^3$ which is passing through the set of control points with a geometrical shape as one would intuitively expect. Furthermore, spurious oscillations should be prevented. For this reason, cubic spline interpolation is not safe. Instead, piecewise cubic Hermite interpolation is applied. The extra freedom to model the tangent vectors is used to prevent unwanted oscillations. The parameter u is defined as normalized arc length.

Compute the distance between succeeding control points:

$$\bar{d}_i = \| \vec{x}_i - \vec{x}_{i-1} \| , \; i = 1 \ldots N. \tag{46}$$

Define the length of the curve by

$$L = \sum_{i=1}^{N} \bar{d}_i, \tag{47}$$

and the normalized distances as

$$d_i = \bar{d}_i / L \ , \ i = 1 \ldots N. \tag{48}$$

Define the knot sequence $\{u_i \mid i = 0 \ldots N\}$ by $u_0 = 0$ and

$$u_i = u_{i-1} + d_i \ , \ i = 1 \ldots N. \tag{49}$$

Hence, $0 = u_0 < u_1 < \ldots < u_N = 1$. Patch i is defined as the interval $[u_{i-1}, u_i]$. In patch i, we relate a local variable $\alpha \in [0, 1]$ to the global variable u by

$$u = u_{i-1} + \alpha(u_i - u_{i-1}) = u_{i-1} + \alpha d_i. \tag{50}$$

For the moment, suppose that the tangent vectors $\{\vec{x}_{u_i} = \frac{d\vec{x}}{du}(u_i), i = 0 \ldots N\}$ are known. How these tangent vectors are modeled is shown below.

The shape of the curve at patch i is then defined as

$$\vec{x}(\alpha) = \vec{x}_{i-1} H_0(\alpha) + \vec{x}_i H_1(\alpha) + d_i \vec{x}_{u_{i-1}} H_2(\alpha) + d_i \vec{x}_{u_i} H_3(\alpha), \tag{51}$$

where H_0, H_1, H_2, H_3 are cubic Hermite interpolation functions defined as

$$\begin{aligned} H_0(\alpha) &= (1 + 2\alpha)(1 - \alpha)^2, \\ H_1(\alpha) &= (3 - 2\alpha)\alpha^2, \\ H_2(\alpha) &= \alpha(1 - \alpha)^2, \\ H_3(\alpha) &= (\alpha - 1)\alpha^2, \end{aligned} \tag{52}$$

with $0 \leq \alpha \leq 1$.

It can be easily verified that $\frac{d\vec{x}}{du}(u_i-) = \frac{d\vec{x}}{du}(u_i+) = \vec{x}_{u_i}$, so that the piecewise cubic curve $\vec{x}(u)$ is indeed C^1.

The tangent vectors $\{\vec{x}_{u_i}, i = 0 \ldots N\}$ are computed as follows. Define the chord vectors

$$\vec{x}_{u_{i-\frac{1}{2}}} = \frac{\vec{x}_i - \vec{x}_{i-1}}{d_i} \ , \ i = 1 \ldots N. \tag{53}$$

Note that $\| \vec{x}_{u_{i-\frac{1}{2}}} \| = L$. The tangent vectors at the interior knots $i = 1 \ldots N - 1$ are modeled as

$$\vec{x}_{u_i} = \vec{x}_{u_{i-\frac{1}{2}}} c_i + \vec{x}_{u_{i+\frac{1}{2}}} (1 - c_i) \ , \ i = 1 \ldots N - 1, \tag{54}$$

with

$$c_i = \frac{\| \vec{x}_i - \vec{x}_{i-1} \|^2}{\| \vec{x}_i - \vec{x}_{i-1} \|^2 + \| \vec{x}_{i+1} - \vec{x}_i \|^2} = \frac{d_i^2}{d_i^2 + d_{i+1}^2} \ , \ i = 1 \ldots N - 1. \tag{55}$$

If $\| \vec{x}_i - \vec{x}_{i-1} \| \ll \| \vec{x}_{i+1} - \vec{x}_i \|$ then $c_i \approx 0$ and $\vec{x}_{u_i} \approx \vec{x}_{u_{i+\frac{1}{2}}}$. This implies that high curvature will be restricted to small patches which is a behaviour as one would intuitively expect. Spurious oscillations are also prevented.

Quadratic end conditions are used to compute the end tangent vectors $\vec{x}_{u_0}$ and $\vec{x}_{u_N}$. The quadratic end conditions require that the cubic polynomial function $\vec{x}(\alpha)$ is a quadratic function of α in patch 1 and in patch N. It is easily verified that this implies that

$$\vec{x}_{u_0} = 2\vec{x}_{u_{\frac{1}{2}}} - \vec{x}_{u_1} \ , \ \vec{x}_{u_N} = 2\vec{x}_{u_{N-\frac{1}{2}}} - \vec{x}_{u_{N-1}}. \tag{56}$$

Fig.12 illustrates that cubic Hermite interpolation prevents spurious oscillations, in contrast to cubic spline interpolation.

5.2 Piecewise bicubic Hermite interpolation for surfaces

Consider a set of control points $\{\vec{x}_{i,j} = (x,y,z)^T_{i,j} \mid i = 0 \ldots N, j = 0 \ldots M\}$. We wish to construct a smooth C^1 surface $\vec{x} : (u,v) \in [0,1]^2 \mapsto \mathcal{R}^3$ which is passing through the set of control points with a geometrical shape as one would intuitively expect. As for curves, spurious oscillations should be prevented. For this reason, bicubic spline interpolation is not safe. Instead, piecewise bicubic Hermite interpolation is applied. The extra freedom to model the tangent vectors and twist vectors is used to prevent unwanted oscillations.

Consider a row of points $\{\vec{x}_{i,j} \mid i = 0 \ldots N\}$ with $j \in \{0 \ldots M\}$ fixed. This row of points is considered as a discrete curve and it is therefore possible to compute a knot sequence $\{u_{i,j} \mid i = 0 \ldots N\}$ in exactly the same way as described in the previous section. In the same way, consider a column of points $\{\vec{x}_{i,j} \mid j = 0 \ldots M\}$ with $i \in \{0 \ldots N\}$ fixed, and compute the knot sequence $\{v_{i,j} \mid j = 0 \ldots M\}$.

To construct a smooth surface, one knot sequence is needed for all rows and all columns. These two knot sequences are obtained by averaging:

$$u_i = \frac{1}{M+1}\sum_{j=0}^{M} u_{i,j} \ , \ i = 0 \ldots N, \ v_j = \frac{1}{N+1}\sum_{i=0}^{N} v_{i,j} \ , \ j = 0 \ldots M. \tag{57}$$

Patch (i,j) is defined as the rectangle $[u_{i-1}, u_i] \times [v_{j-1}, v_j]$. In patch (i,j) we relate local variables $(\alpha, \beta) \in [0,1]^2$ to the global variables (u,v) by

$$u = u_{i-1} + \alpha d^u_i \ , \ v = v_{i-1} + \beta d^v_j, \tag{58}$$

with $d^u_i = u_i - u_{i-1}$ and $d^v_j = v_j - v_{j-1}$.

For the moment, suppose that the tangent vectors $\vec{x}_{u_{i,j}} = \frac{\partial \vec{x}}{\partial u}(u_i, v_j)$, $\vec{x}_{v_{i,j}} = \frac{\partial \vec{x}}{\partial v}(u_i, v_j)$, and twist vectors $\vec{x}_{uv_{i,j}} = \frac{\partial^2 \vec{x}}{\partial u \partial v}(u_i, v_j)$ are known for all knots (i,j). How these tangent and twist vectors are modeled is shown below.

The shape of the surface at patch (i,j) is then defined as

$$\vec{x}(\alpha,\beta) = (H_0(\alpha), H_1(\alpha), H_2(\alpha), H_3(\alpha)) M^H_{i,j} \begin{pmatrix} H_0(\beta) \\ H_1(\beta) \\ H_2(\beta) \\ H_3(\beta) \end{pmatrix}, \tag{59}$$

where the matrix $M^H_{i,j}$ is defined by

$$M^H_{i,j} = \begin{pmatrix} \vec{x}_{i-1,j-1} & \vec{x}_{i-1,j} & d^v_j \vec{x}_{v_{i-1,j-1}} & d^v_j \vec{x}_{v_{i-1,j}} \\ \vec{x}_{i,j-1} & \vec{x}_{i,j} & d^v_j \vec{x}_{v_{i,j-1}} & d^v_j \vec{x}_{v_{i,j}} \\ d^u_i \vec{x}_{u_{i-1,j-1}} & d^u_i \vec{x}_{u_{i-1,j}} & d^u_i d^v_j \vec{x}_{uv_{i-1,j-1}} & d^u_i d^v_j \vec{x}_{uv_{i-1,j}} \\ d^u_i \vec{x}_{u_{i,j-1}} & d^u_i \vec{x}_{u_{i,j}} & d^u_i d^v_j \vec{x}_{uv_{i,j-1}} & d^u_i d^v_j \vec{x}_{uv_{i,j}} \end{pmatrix}. \tag{60}$$

From these definitions, it can be easily verified that the piecewise bicubic surface $\vec{x}(u,v)$ is C^1.

The tangent vectors $\vec{x}_{u_{i,j}}$ are computed as follows (the tangent vectors $\vec{x}_{v_{i,j}}$ are computed in the same way). Define the chord vectors

$$\vec{x}_{u_{i-\frac{1}{2},j}} = \frac{\vec{x}_{i,j} - \vec{x}_{i-1,j}}{d_i^u} \ , \ i = 1 \ldots N \ , \ j = 0 \ldots M, \tag{61}$$

and use the same non-linear averaging formula as used for curves, thus

$$\vec{x}_{u_{i,j}} = \vec{x}_{u_{i-\frac{1}{2},j}} c_{i,j} + \vec{x}_{u_{i+\frac{1}{2},j}} (1 - c_{i,j}) \ , \ i = 1 \ldots N-1 \ , \ j = 0 \ldots M. \tag{62}$$

with

$$c_{i,j} = \frac{\| \vec{x}_{i,j} - \vec{x}_{i-1,j} \|^2}{\| \vec{x}_{i,j} - \vec{x}_{i-1,j} \|^2 + \| \vec{x}_{i+1,j} - \vec{x}_{i,j} \|^2} \ , \ i = 1 \ldots N-1 \ , \ j = 0 \ldots M. \tag{63}$$

Quadratic end conditions are used again to compute the end tangent vectors.

A modification of Adini's method [9] is used to compute the twist vectors. Consider patch (i,j) with local variables (α, β). Assume that the tangent vectors $\vec{x}_u$, $\vec{x}_v$ are known at the four corner points of the patch. Introduce the abbreviate notation $\vec{x}_{00} = \vec{x}_{i-1,j-1}$, $\vec{x}_{10} = \vec{x}_{i,j-1}$, $\vec{x}_{01} = \vec{x}_{i-1,j}$, $\vec{x}_{11} = \vec{x}_{i,j}$. Use Eqs.(59),(60) to find $\vec{x}_\alpha(0,0) = d_i^u \vec{x}_{u_{i-1,j-1}}$, $\vec{x}_\alpha(1,0) = d_i^u \vec{x}_{u_{i,j-1}}$, $\vec{x}_\alpha(0,1) = d_i^u \vec{x}_{u_{i-1,j}}$, $\vec{x}_\alpha(1,1) = d_i^u \vec{x}_{u_{i,j}}$, $\vec{x}_\beta(0,0) = d_j^v \vec{x}_{v_{i-1,j-1}}$, $\vec{x}_\beta(1,0) = d_j^v \vec{x}_{v_{i,j-1}}$, $\vec{x}_\beta(0,1) = d_j^v \vec{x}_{v_{i-1,j}}$, $\vec{x}_\beta(1,1) = d_j^v \vec{x}_{v_{i,j}}$. Compute the boundary curves of the patch by cubic Hermite interpolation. Thus, for example

$$\vec{x}(\alpha, 0) = \vec{x}_{00} H_0(\alpha) + \vec{x}_{10} H_1(\alpha) + \vec{x}_\alpha(0,0) H_2(\alpha) + \vec{x}_\alpha(1,0) H_3(\alpha), \tag{64}$$

and the other three boundary curves are computed by similar formulas. Define the shape of the surface patch as a bilinearly blended Coon's patch

$$\vec{x}(\alpha,\beta) = (1-\alpha, \alpha) \begin{pmatrix} \vec{x}(0,\beta) \\ \vec{x}(1,\beta) \end{pmatrix} + (1-\beta, \beta) \begin{pmatrix} \vec{x}(\alpha,0) \\ \vec{x}(\alpha,1) \end{pmatrix} - (1-\alpha, \alpha) \begin{pmatrix} \vec{x}_{00} & \vec{x}_{01} \\ \vec{x}_{10} & \vec{x}_{11} \end{pmatrix} \begin{pmatrix} 1-\beta \\ \beta \end{pmatrix}. \tag{65}$$

Compute the corner twist vectors $\vec{x}_{\alpha\beta}(0,0), \vec{x}_{\alpha\beta}(1,0), \vec{x}_{\alpha\beta}(0,1), \vec{x}_{\alpha\beta}(1,1)$ from Eq.(65). Use Eq.(58) to find the corresponding twist vectors w.r.t. the global variables (u,v): $\vec{x}_{uv_{i-1,j-1}} = \vec{x}_{\alpha\beta}(0,0)/d_i^u d_j^v$ etc..

Thus at the four corners of each patch, an estimation is found for the twist vector $\vec{x}_{uv}$. Consider an interior knot (i,j). Then four estimations for the twist vector $\vec{x}_{uv_{i,j}}$ are found in respectively patches (i,j),$(i+1,j)$,$(i,j+1)$ and $(i+1,j+1)$. Write those estimations as respectively $\vec{x}_{uv}^{SW}$, $\vec{x}_{uv}^{SE}$, $\vec{x}_{uv}^{NW}$, $\vec{x}_{uv}^{NE}$. A similar averaging procedure as applied for tangent vectors is used to define a unique value $\vec{x}_{uv_{i,j}}$:

$$\vec{x}_{uv_{i,j}} = \frac{\vec{x}_{uv}^{SW} A_{i,j}^2 + \vec{x}_{uv}^{SE} A_{i+1,j}^2 + \vec{x}_{uv}^{NW} A_{i,j+1}^2 + \vec{x}_{uv}^{NE} A_{i+1,j+1}^2}{A_{i,j}^2 + A_{i+1,j}^2 + A_{i,j+1}^2 + A_{i+1,j+1}^2}, \tag{66}$$

where $A_{i,j}$ is the area of patch (i,j) defined as

$$A_{i,j} = 0.5 \, \| \, (\vec{x}_{i,j} - \vec{x}_{i-1,j-1}) \wedge (\vec{x}_{i-1,j} - \vec{x}_{i,j-1}) \, \| \, . \tag{67}$$

This non-linear averaging procedure guarantees that large changes in twist will be restricted to small patches. At a boundary knot, there are only two estimations for the twist vector. It is evident how the averaging procedure must be applied in that case. At the four corner knots, only one estimation is available and averaging is thus not needed.

Figs.10,11 illustrate that bicubic Hermite interpolation gives a smooth surface shape, even if the control points are irregularly distributed.

Concluding remarks

An elliptic grid generation method is presented to generate boundary conforming grids in domains in 2D physical space and on minimal surfaces and parametrized surfaces in 3D physical space. The method is based on the use of composite mappings and produce excellent grids in the sense of smoothness, grid point distribution and regularity.

The described elliptic grid generation method has been implemented into NLR's multi-block grid generation code ENGRID and is extensively used for the generation of boundary conforming Navier-Stokes grids in block-faces with complex shapes.

Concerning surface modeling, it is shown that bicubic Hermite interpolation is an excellent method to define interpolated surfaces.

References

[1] J.W. Boerstoel, S.P. Spekreijse, P.L. Vitagliano, The Design of a System of Codes for Industrial Calculations of Flows around Aircraft and other Complex Aerodynamic Configurations, NLR-TP-92190, AIAA-92-2619-CP, 10th Applied Aerodynamics Conference, Palo Alto, 1992.

[2] S.P. Spekreijse, Elliptic Grid Generation Based on Laplace Equations and Algebraic Transformations, NLR-TP-94102,1994. (Accepted for publication in Journal of Computational Physics).

[3] C. W. Mastin and J. F. Thompson, Transformation of Three-Dimensional Regions onto Rectangular Regions by Elliptic Systems, Numerische Mathematik, vol.29,pp.397-407,1978.

[4] E. Kreyszig, Introduction to Differential Geometry and Riemannian Geometry, Mathematical Expositions no.16, University of Toronto Press,1967.

[5] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, Numerical Grid Generation: Foundations and Applications, Elsevier, New York, 1985.

[6] Z.U.A. Warsi, Basic Differential Models for Coordinate Generation, in: Numerical Grid Generation, Ed. J.F. Thompson, Elsevier, North-Holland,pp.41-70,1982.

[7] Z.U.A. Warsi, Numerical Grid Generation in Arbitrary Surfaces through a Second-Order Differential-Geometric Model, Journal of Computational Physics 64,pp.82-96,1986.

[8] U.Dierkes, S. Hildebrandt, A. Kuster, O. Wohlrab, Minimal Surfaces I, Grundlehren der Mathematischen Wissenschaften 295, Springer Verlag, Berlin, 1991.

[9] F.Yamaguchi, Curves and Surfaces in Computer Aided Geometric Design, Springer Verlag, Berlin, 1988.

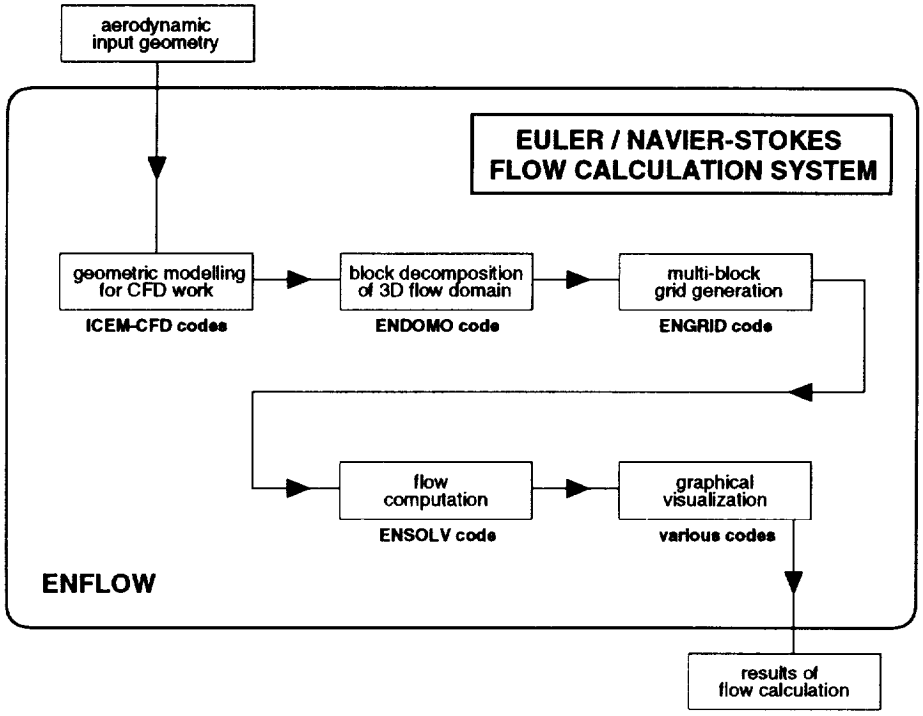

Figure 4: The NLR ENFLOW system

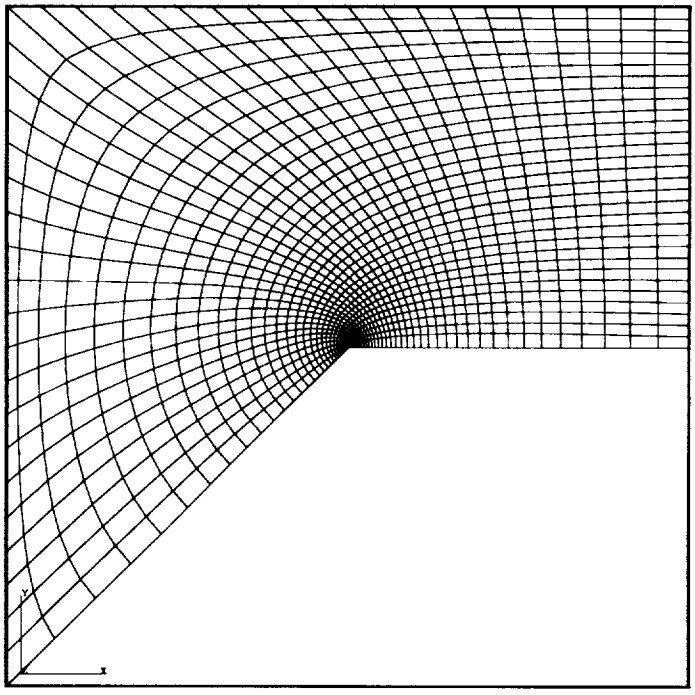

Figure 5: Elliptic grid with grid orthogonality at the lower boundary of the domain.

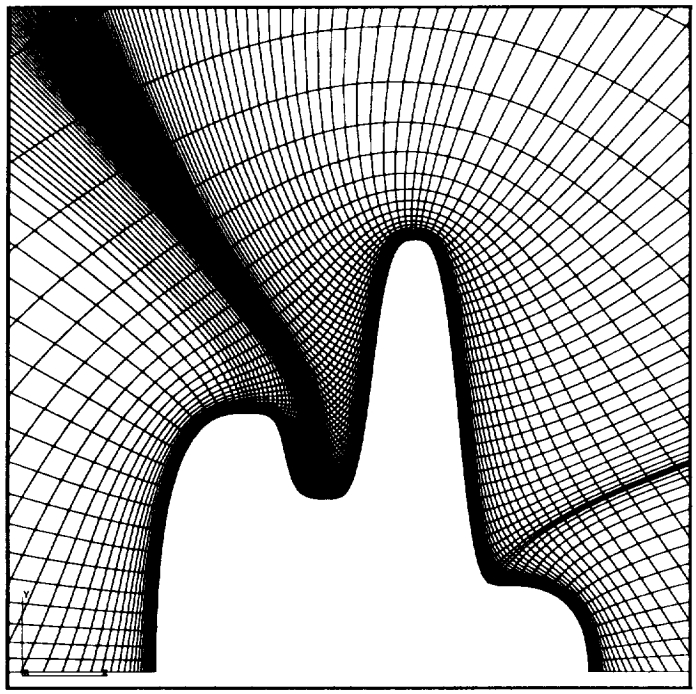

Figure 6: Elliptic grid with boundary layer and orthogonality.

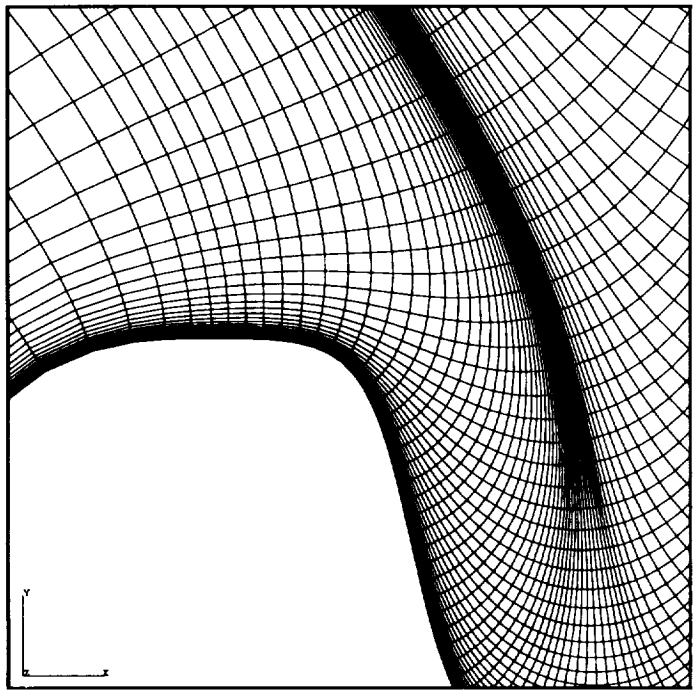

Figure 7: Detail of elliptic grid at convex part of the boundary.

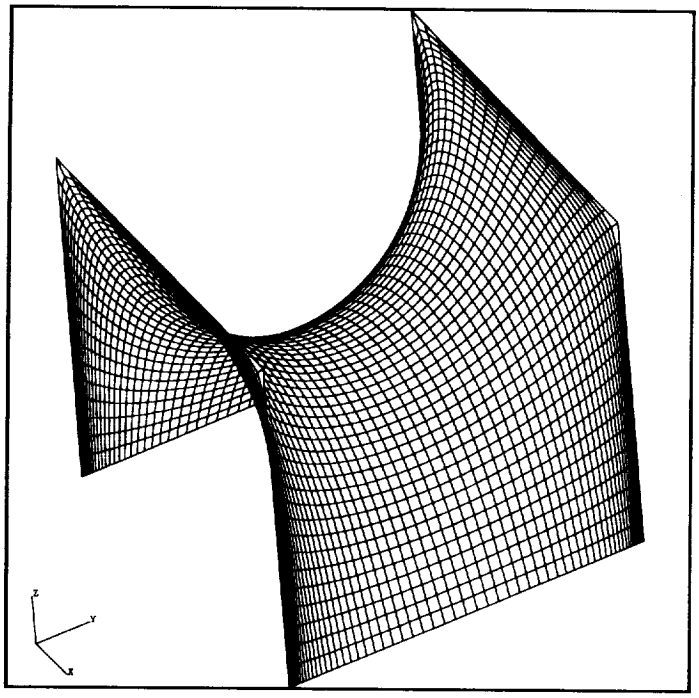

Figure 8: Minimal surface grid. Surface is a square Scherck surface.

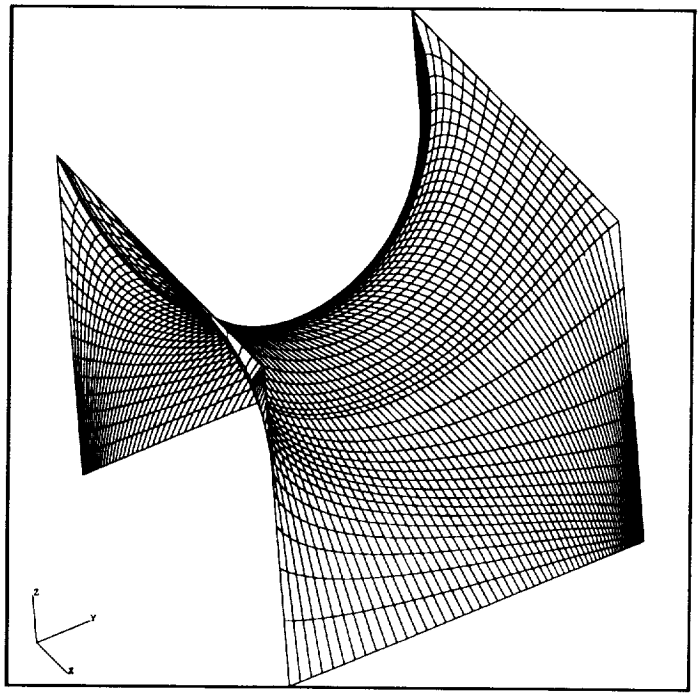

Figure 9: Minimal surface grid. Shape of surface is independent of the boundary grid point distribution.

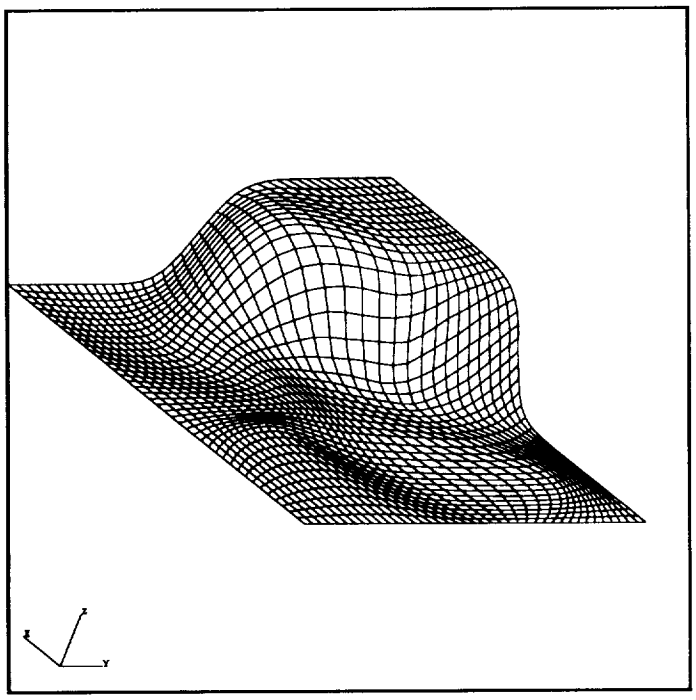

Figure 10: Irregularly distributed control point mesh on a smooth surface.

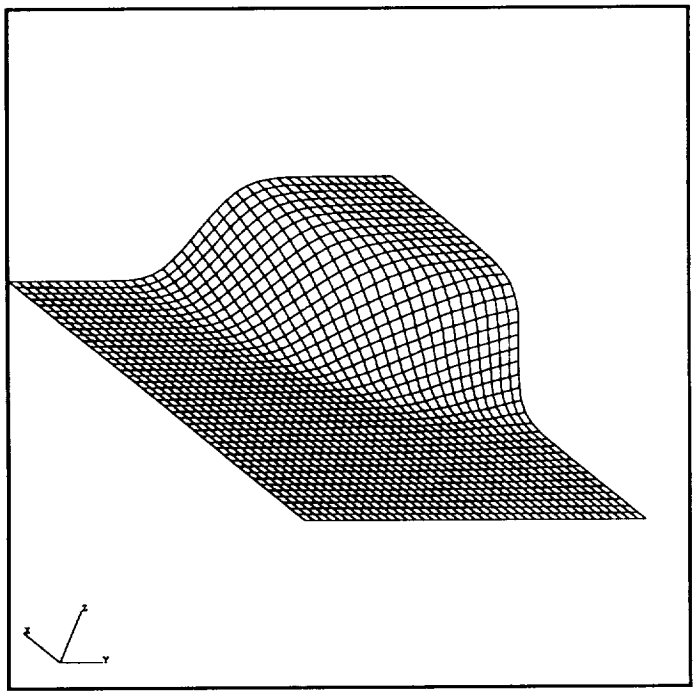

Figure 11: Elliptic grid on the surface. Grid is independent of the parametrization.

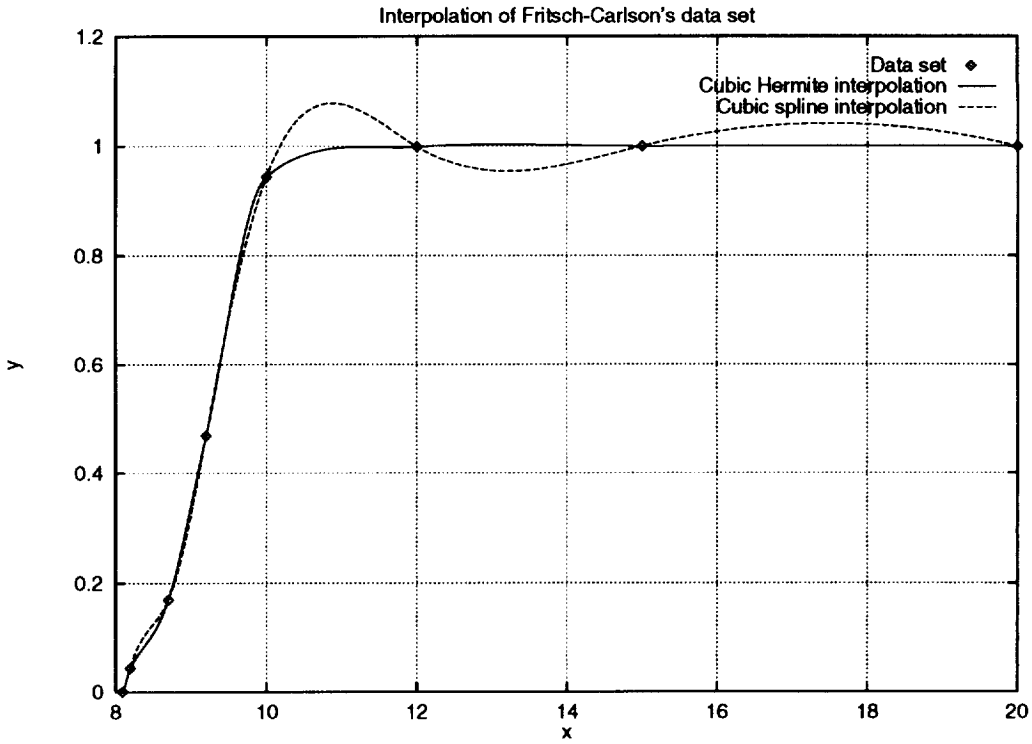

Figure 12: Comparison of cubic Hermite and cubic spline interpolation.

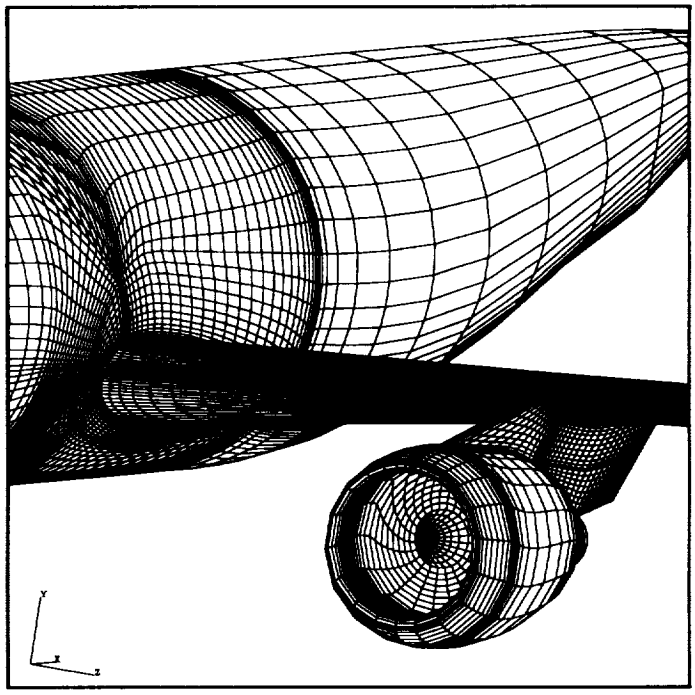

Figure 13: Surface grids of a wing-body-pylon-nacelle configuration.