HypGrid2D a 2-D mesh generator

Sørensen, Niels N.

Publication date:
1998

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Sørensen, N. N. (1998). *HypGrid2D a 2-D mesh generator*. Denmark. Forskningscenter Risoe. Risoe-R No. 1035(EN)

RISØ

HypGrid2D
a 2-D Mesh Generator

Niels N. Sørensen

Risø National Laboratory, Roskilde, Denmark
March 1998

Risø-R-1035(EN)

HypGrid2D
a 2-D Mesh Generator

Niels N. Sørensen

Risø National Laboratory, Roskilde, Denmark
March 1998

Abstract The implementation of a hyperbolic mesh generation procedure, based on an equation for orthogonality and an equation for the cell face area is described. The method is fast, robust and gives meshes with good smoothness and orthogonality. The procedure is implemented in a program called HypGrid2D. The HypGrid2D program is capable of generating C-, O- and "H"-meshes for use in connection with the EllipSys2D Navier-Stokes solver.

To illustrate the capabilities of the program, some test examples are shown. First a series of C-meshes are generated around a NACA-0012 airfoil. Secondly a series of O-meshes are generated around a NACA-65-418 airfoil. Finally "H"-meshes are generated over a gaussian hill and a linear escarpment.

The report has passed an internal review at the Wind Energy and Atmospheric Physics Department performed by:

Kristian S. Dahl

Jeppe Johansen

ISBN 87-550-2368-1
ISSN 0106-2840

Afdelingen for Informationsservice · Risø · 1998

Contents

1 Introduction

Mesh generation is an essential part of computational fluid dynamics. The quality of a solution is strongly influenced by the quality of the mesh. There are several requirements that a mesh must fulfill. It must be orthogonal at the surfaces, resolve the flow gradients, be smooth and well aligned with the flow.

Many different mesh generation techniques have been developed; transfinite interpolation methods, conformal mapping, and methods based on hyperbolic and elliptic equations. For a good introduction to the various methods of grid generation, see [8].

The present work is carried out to provide a mesh generation tool, for use in connection with the Navier-Stokes solver EllipSys2D [3] [2]. The mesh generator is especially suited for external aerodynamics, e.g. flow around airfoils, and flow over natural terrain.

2 Hyperbolic mesh generation

In the present work a 2-D hyperbolic mesh generator will be described. The hyperbolic mesh generation technique has several advantages; it generates highly orthogonal meshes, the method is simple, relatively robust and very fast. The main disadvantages are the lack of control of the outer boundary of the mesh and the propagation of boundary discontinuities into the internal part of the mesh.

In connection with external aerodynamics, the control of the exact location of the outer boundary is of little consequence. Here one simply has to assure that the outer boundary is located sufficiently far away to have negligible influence on the area of interest. For internal flows the hyperbolic technique can be combined with other methods, using the hyperbolic equations close to the surfaces. Alternatively, blending of hyperbolic meshes generated from opposing surfaces can be used to obtain control over all boundaries of the mesh, see Jeng and Liou [1].

The propagation of boundary discontinuities, can to some extend be removed using inherent or adaptive dissipation as described in Tai et al. [7]. Additionally, explicit smoothing of the mesh layers before generating the next layer can help to limit the problem, see Tai et al. [6]. The hyperbolic equation system used in the present work is based on an equation for orthogonality and an equation for the cell area as described in the original work of Steger and Sorenson [4] and Steger and Chaussee [5], namely

$$\begin{aligned} x_\xi x_\eta + y_\xi y_\eta &= 0 \ , & (1) \\ x_\xi y_\eta - y_\xi x_\eta &= \text{Area} \ , & (2) \end{aligned}$$

where x and y are Cartesian coordinates, and ξ and η are the coordinates in the transformed space. In the following η will be used as the marching direction.

The equations (1) and (2) are linearized around a known state x_0, y_0 using $x = x_0 + x'$ and $y = y_0 + y'$. Expressions of the following form are obtained for the individual terms in the equations

$$\begin{aligned} x_\xi x_\eta &= (x^0 + x')_\xi (x^0 + x')_\eta \\ &= x^0_\xi x^0_\eta + x^0_\xi x'_\eta x'_\xi x^0_\eta + x'_\xi x'_\eta \\ &= x^0_\xi x^0_\eta + x^0_\xi (x - x^0)_\eta x^0_\eta (x - x^0)_\xi + O(\Delta^2) \end{aligned}$$

$$= \quad x^0_\xi x_\eta + x^0_\eta x_\xi - x^0_\xi x^0_\eta + O(\Delta^2) \ , \tag{3}$$

and in analog way the following expressions are obtained for the remaining terms in the equations (1) and (2)

$$y_\xi y_\eta = y^0_\xi y_\eta + y^0_\eta y_\xi - y^0_\xi y^0_\eta + O(\Delta^2) \tag{4}$$

$$x_\xi y_\eta = x^0_\xi y_\eta + y^0_\eta x_\xi - x^0_\xi y^0_\eta + O(\Delta^2) \tag{5}$$

$$y_\xi x_\eta = y^0_\xi x_\eta + x^0_\eta y_\xi - y^0_\xi x^0_\eta + O(\Delta^2) \tag{6}$$

Substituting the expressions (3) to (6) into the equations (1) and (2) the following equations are generated

$$x^0_\xi x_\eta + x^0_\eta x_\xi + y^0_\xi y_\eta + y^0_\eta y_\xi \quad = \quad x^0_\xi x^0_\eta + y^0_\xi y^0_\eta \ , \tag{7}$$

$$x^0_\xi y_\eta + y^0_\eta x_\xi - y^0_\xi x_\eta - x^0_\eta y_\xi \quad = \quad x^0_\xi y^0_\eta - y^0_\xi x^0_\eta + \mathrm{Area} \ . \tag{8}$$

Using the fact that the mesh is orthogonal $x^0_\xi x^0_\eta + y^0_\xi y^0_\eta = 0$, and the expression for the area of the cell $\mathrm{Area}^0 = x^0_\xi y^0_\eta - y^0_\xi x^0_\eta$ we can rewrite the equations, namely

$$x^0_\xi x_\eta + x^0_\eta x_\xi + y^0_\xi y_\eta + y^0_\eta y_\xi \quad = \quad 0 \ , \tag{9}$$

$$x^0_\xi y_\eta + y^0_\eta x_\xi - y^0_\xi x_\eta - x^0_\eta y_\xi \quad = \quad \mathrm{Area} + \mathrm{Area}^0 \ . \tag{10}$$

Or, in vector notation,

$$A\boldsymbol{r}_\xi + B\boldsymbol{r}_\eta \quad = \quad \boldsymbol{V} \ , \tag{11}$$

$$B^{-1}A\boldsymbol{r}_\xi + \boldsymbol{r}_\eta \quad = \quad B^{-1}\boldsymbol{V} \ , \tag{12}$$

$$C\boldsymbol{r}_\xi + \boldsymbol{r}_\eta \quad = \quad \boldsymbol{S} \ , \tag{13}$$

where

$$A = \begin{bmatrix} x^0_\eta & y^0_\eta \\ y^0_\eta & -x^0_\eta \end{bmatrix} , \qquad B = \begin{bmatrix} x^0_\xi & y^0_\xi \\ -y^0_\xi & x^0_\xi \end{bmatrix} , \qquad \boldsymbol{V}_\eta = \begin{bmatrix} 0 \\ \mathrm{Area} + \mathrm{Area}^0 \end{bmatrix} ,$$

$$\boldsymbol{r}_\xi = \begin{bmatrix} x \\ y \end{bmatrix}_\xi , \qquad \boldsymbol{r}_\eta = \begin{bmatrix} x \\ y \end{bmatrix}_\eta ,$$

$$C = B^{-1}A = \frac{1}{\det B} B^T A = \frac{1}{\gamma} \begin{bmatrix} \alpha & \beta \\ \beta & -\alpha \end{bmatrix} ,$$

$$\boldsymbol{S} = B^{-1}\boldsymbol{V} = \frac{1}{\det B} B^T \boldsymbol{V} = \frac{\mathrm{Area} + \mathrm{Area}^0}{\gamma} \begin{bmatrix} -y^0 \\ x^0 \end{bmatrix}_\xi ,$$

$$\begin{aligned} \alpha &= x^0_\xi x^0_\eta - y^0_\xi y^0_\eta \ , \\ \beta &= x^0_\xi y^0_\eta + x^0_\eta y^0_\xi \ , \\ \gamma &= x^0_\xi x^0_\xi + y^0_\xi y^0_\xi \ . \end{aligned}$$

2.1 Discretization of the governing equations

The partial derivatives in equation 13 are approximated with finite differences. For the inner part of the domain second order central differences are used for the ξ derivatives, while first order forward or backward differences are used at the left and right hand sides of the domain, respectively. For the η derivatives first order forward differences are used. This results in the following expressions for the partial derivatives

$$r_\xi = \frac{r(\xi+1,\eta) - r(\xi-1,\eta)}{2} \qquad , \xi \in [2 : \mathrm{NI}-1] \; , \tag{14}$$

$$r_\xi = r(\xi+1,\eta) - r(\xi,\eta) \qquad , \xi = 1 \; , \tag{15}$$

$$r_\xi = r(\xi,\eta) - r(\xi-1,\eta) \qquad , \xi = \mathrm{NI} \; , \tag{16}$$

and

$$r_\eta = r(\xi,\eta) - r(\xi,\eta-1) \qquad , \xi \in [1 : \mathrm{NI}] \; . \tag{17}$$

The coefficients are calculated using information from the previous layer, with the ξ differences given by equation (14) to (16), but the η derivatives are computed from equation (1) and (2) as given below

$$x^0_\eta = \frac{-y^0_\xi \mathrm{Area}^0}{\xi^0_\xi x^0_\xi + y^0_\xi y^0_\xi} = \frac{-y^0_\xi \mathrm{Area}^0}{\gamma} \; , \tag{18}$$

$$x^0_\eta = \frac{x^0_\xi \mathrm{Area}^0}{\xi^0_\xi x^0_\xi + y^0_\xi y^0_\xi} = \frac{x^0_\xi \mathrm{Area}^0}{\gamma} \; . \tag{19}$$

The area function is user specified, and the actual function used in the present approach is given in section 2.3.

Inserting the finite differences into equation 13, a NI × NI block tridiagonal system with 2 × 2 blocks is generated. As pointed out by Steger and Chaussee[5], a dissipation term is necessary in order to make the equations stable. In the present approach an adaptive second-order dissipation term, originally proposed by Tai et al. [7] is used, namely

$$\text{adaptive dissipation} = \frac{1}{2}\lambda(\Delta\nabla)r \; . \tag{20}$$

Where λ is the eigenvalue of the matrix $\mathcal{C}$, and Δ and ∇ are the forward and backward differences with respect to ξ, respectively. The eigenvalue can be computed as

$$\lambda = \sqrt{(\alpha^2 + \beta^2)/\gamma^2} \tag{21}$$

Adding the dissipation term to the discretized equation 13, and rearranging we get the following expression

$$\frac{1}{2}(\mathcal{C}(\xi,\eta) + \lambda(\xi,\eta-1)\mathcal{I})r(\xi+1,\eta) - \frac{1}{2}(\mathcal{C}(\xi,\eta) - \lambda(\xi,\eta-1)\mathcal{I})r(\xi-1,\eta)$$
$$+(\mathcal{I} - 2\lambda(\xi,\eta-1)\mathcal{I})r(\xi,\eta) = \mathcal{S}(\xi,\eta) - r(\xi,\eta-1) \; , \tag{22}$$

which still results in a block tridiagonal system. In the present approach the system is solved using a standard block tridiagonal solver.

2.2 Boundary conditions

Boundary conditions are needed at $\xi = 1$ and $\xi = \mathrm{NI}$. Depending on the actual mesh geometry, two different types of boundary conditions will be considered here, one corresponding to "H"- and C-meshes and one corresponding to an O-mesh.

Boundary conditions for "H"- and C-meshes

For an "H"- or a C-mesh, the physical conditions to be enforced at the boundaries of the domain is $\frac{\partial x}{\partial \eta} = 0$. Written in discrete form this results in the following conditions:

$$x(1, \eta) = x(1, \eta - 1) \qquad \wedge \qquad x(\mathrm{NI}, \eta) = x(\mathrm{NI}, \eta - 1) \ . \tag{23}$$

Incorporating the one sided difference at $\xi = 1$, equation (13) can be rewritten in the following way

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{i,j} + \frac{1}{2} \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{i+1,j}$$

$$-\frac{1}{2} \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{i,j} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} ,$$

using the boundary condition stated above we now get

$$\begin{bmatrix} 2 & 0 \\ -c_{21} & 2 - c_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{i,j} + \begin{bmatrix} 0 & 0 \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{i+1,j} = \begin{bmatrix} 2x_{i,j-1} \\ 2s_2 - c_{21}x_{i,j-1} \end{bmatrix}$$

This is a fully implicit boundary conditions, and this treatment have proven to be extremely stable and produce meshes of good quality.

Boundary conditions for O-meshes

For an O-mesh, the physical conditions to be enforced at the boundaries $\xi = 1$ and $\xi = \mathrm{NI}$ of the domain are $\frac{\partial y}{\partial \eta} = 0$. Written in discrete form this results in the following conditions:

$$y(1, \eta) = y(1, \eta - 1) \qquad \wedge \qquad y(\mathrm{NI}, \eta) = y(\mathrm{NI}, \eta - 1) \ . \tag{24}$$

This can again be used to rewrite the discrete equations at $\xi = 1$ and $\xi = \mathrm{NI}$ as for the C- and "H"-meshes, the result for the $\xi = 1$ boundary is

$$\begin{bmatrix} 2 - c_{11} & -c_{12} \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{i,j} + \begin{bmatrix} c_{11} & c_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{i+1,j} = \begin{bmatrix} 2s_1 - c_{12}y_{i,j-1} \\ 2y_{i,j-1} \end{bmatrix}$$

The described boundary condition do not inforce c_0-continuity at the wake-cut of the O-mesh. The following explicit procedure is used to obtain the c_0-continuity: Having obtained the coordinates for the new layer from the hyperbolic equations, the difference between the distance from the trailing edge to the points on the upper and lower side of the cut is computed:

$$\delta = \sqrt{(x_{up}^2 + y_{up}^2)} - \sqrt{(x_{lo}^2 + y_{lo}^2)} \ ,$$

where (x_{up}, y_{up}) and (x_{lo}, y_{lo}) are the coordinates of the points on the upper and lower side of the mesh cut, respectively.

To obtain c_0 continuity the shortest position vectors is extended by the difference $(|\delta|)$. By scaling $(|\delta|)$ by a factor decreasing when moving away from the cut in the ξ-direction, the original hyperbolic mesh is preserved away from the cut.

2.3 Cell area

Besides the boundary point distribution, the main control parameter for the grid generation is the specification of the cell area as function of the computational coordinates (ξ, η).

The most important region of the grid to control is the part near the surface. Here it is often desirable to control the normal spacing to resolve the boundary layer. To facilitate this the present area function is computed as the cell length times a desired cell height. The desired cell height of the cell next to the surface is specified as function of ξ. Based on this and a desired maximum height of the mesh, the cell area function Area(ξ, η) is computed using a suitable stretching function.

Stretching functions

Two stretching functions are implemented in the present grid generator, based on the hyperbolic tangent and hyperbolic sine, respectively. Both streching functions are well suited for aerodynamic applications. The hyperbolic tangent function has the best overall point distribution, while the hyperbolic sine has the best point distribution close two the surface, see Vinokur [9], Thompson [8].

Given the slope of the distribution function at $\eta = 1$ ($\partial s/\partial \eta$ is approximately equal to the normalized height of the first cell $\Delta_1/\Delta s$), the length of the curve (Δs), and the number of points in the distribution function (NJ). The hyperbolic tangent stretching function can then be constructed by the following approach, see Thompson [8]:
Start by calculating the parameter B

$$B = \frac{\Delta s}{(\mathrm{NJ} - 1)\Delta_1} \tag{25}$$

For B > 1 , which will be the case for the present application we must solve the transcendental equation

$$B = \frac{\sinh(\delta)}{\delta} \tag{26}$$

The stretching function $f(\eta)$ is then given by

$$f(\eta) = \Delta s \left(1 + \frac{\tanh\left(\delta \frac{\eta - 1}{\mathrm{NJ}-1} - 1\right)}{\tanh(\delta)} \right) \ , \eta = [1, \mathrm{NJ}] \ . \tag{27}$$

The transcendental equation is solved using a Newton method, and can generally be solved within machine accuracy in around 10 iterations.

The hyperbolic sine can be constructed by again using equations (25) and (26), but with the following expression for the stretching function

$$f(\eta) = \Delta s \frac{\sinh\left(\delta \frac{\eta - 1}{\mathrm{NJ}-1}\right)}{\sinh(\delta)} \ , \eta = [1, \mathrm{NJ}] \ . \tag{28}$$

Area blending

To obtain equal size cells in the farfield, and to avoid clustering and damp oscillation of grid lines, an area blending technique is used. The blending is performed between the local cell area Area$_{cell}$ and the mean cell area of the present layer Area$_{mean}$ in the following way.

$$\mathrm{Area} = \mathrm{factor} \times \mathrm{Area}_{cell} + (1 - \mathrm{factor}) \times \mathrm{Area}_{mean} \ , \tag{29}$$

where

$$\text{factor} = 0.5 \times \left(1 + (1 - \text{factor}_{blend})^{\eta-2}\right) \ . \tag{30}$$

As seen from equation (30), the local cell area is used for the cell next to the surface ($\eta = 2$), and depending on the user specified parameter factor_{blend} the area will approach the averaged cell area when η approaches η_{max}. Generally, a value of $\text{factor}_{blend} = [0 : 0.4]$ is recommended. Some examples of adjustment of the blending factor is given in the example section.

2.4 Smoothing algorithm

To limit the problem with propagation of boundary discontinuities and the problem of crossing grid lines, explicit smoothing is applied to the mesh layer ($\eta =$ const.) before generation of the next layer. As recommended by Tai et al. [6], the smoother is based on a Laplacian smoother. Instead of the computational coordinates they recommend an expression based on arc length of the curve, as described below.

$$\tilde{r} = r + \alpha r_{ss} \ ,$$

where $\tilde{\boldsymbol{r}}$ is the smoothed coordinate vector, and $\boldsymbol{r}_{ss}$ is the second derivative with respect to curve length. With l_p being the distance between $\boldsymbol{r}(\xi+1,\eta)$ and $\boldsymbol{r}(\xi,\eta)$, and l_m being the distance between $\boldsymbol{r}(\xi,\eta)$ and $\boldsymbol{r}(\xi-1,\eta)$ the following expression can be derived for the partial derivative

$$r_{ss} = 2\frac{\left(\boldsymbol{r}(\xi+1,\eta) - \boldsymbol{r}(\xi,\eta)\right)/l_p - \left(\boldsymbol{r}(\xi,\eta) - \boldsymbol{r}(\xi-1,\eta)\right)/l_m}{l_p + l_m}$$

$$r_{ss} = \frac{-2}{l_p l_m} r(\xi,\eta) + \frac{2}{l_p l_m} \frac{l_m r(\xi+1,\eta) + l_p r(\xi-1,\eta)}{l_p + l_m}$$

$$\tilde{r}_{i,j} = (1-\alpha')r(\xi,\eta) + \alpha' \frac{l_m r(\xi+1,\eta) + l_p r(\xi-1,\eta)}{l_p + l_m} \qquad \text{, where } \alpha' = \frac{2\alpha}{l_p l_m}$$

If smoothing is applied close to a sharp convex corner, overlapping of grid lines may result, and to avoid this problem the smoothing is turned on as the mesh marches away from the surface using the following function for the smoothing parameter:

$$\alpha' = \min(\alpha'_0, \alpha'_0(\eta - 2)/\eta_s) \ .$$

The value of α'_0 is user specified, and must be in the range [0:.5], and the value η_s is set equal to η_{max}. Using this the smoothing parameter α' will be equal to 0 for the first mesh layer above the surface ($\eta = 2$) and increases slowly towards the maximum value α'_0. The number of smoothing passes for each mesh layer is also given by the user and is typical in the range [0:10].

2.5 Distribution function

The points on the bottom boundary (the $\eta = 1$ face) must be distributed before the mesh generation can take place. The surface point distribution has a large impact on the final mesh. The main impact of the boundary distribution is the tendency of the hyperbolic method to conserve the surface point distribution in the inner part of the mesh. A typical example is the high concentration of grid lines emerging normal to the airfoil chord near the trailing edge in a C-mesh. The smoothness of the boundary point distribution is also very important, as discontinuities in the point distribution can lead to oscillations and in the worst case breakdown of the mesh generation algorithm. The volume averaging technique described in connection with the volume function, can to some extend limit the previous described problems.

Many different techniques can be used to distribute the points on the surface, in the present implementation a point distribution technique based on a spline description of the normalized arclength as function of normalized computational coordinate ξ_n is used. The method generate a smooth point distribution, is flexible, easy to use, and gives good control over the point distribution.

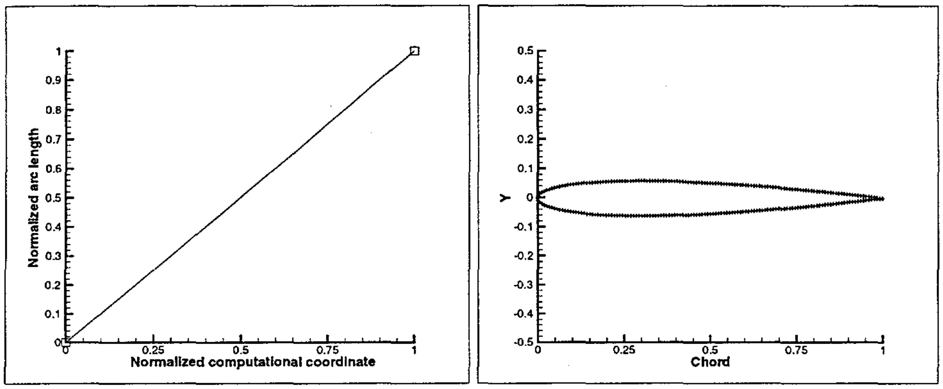

Figure 1. Point distribution with constant spacing based on only two control points.

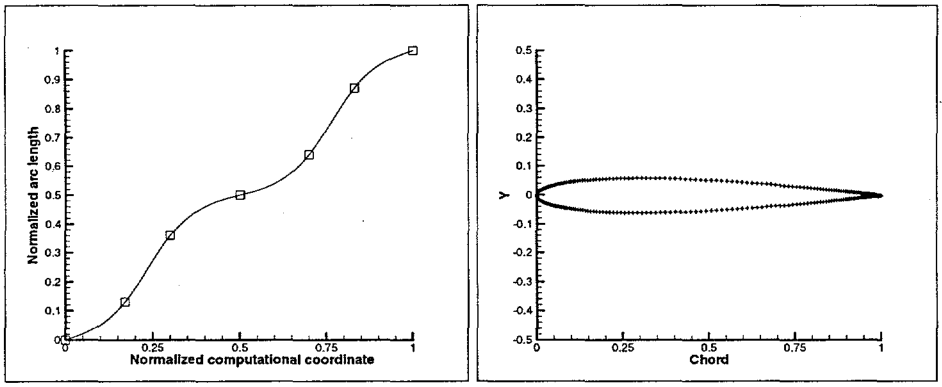

Figure 2. Point distribution with cell concentration near trailing and leading edge based on 7 control points.

The method is illustrated in figure 1 to 3 where three different point distributions on an airfoil are shown. The point distributions are generated by the splines shown at the left side of the figures. In figure 1 a distribution with constant spacing is generated by a spline based on two control points. The control points are indicated on the spline curve. A more adequate point distribution for the airfoil is obtained in figure 2, where the points are concentrated near the leading and trailing edge. The corresponding spline curve is constructed from only 7 control points, see the left side of the figure. By adding more control points nearly any desired point distribution can be obtained, see figur 3. The spline has to fulfill three conditions:

$fs(\xi_n) = 0$, for $\xi_n = 0$,

$fs(\xi_n) = 1$, for $\xi_n = 1$,

$fs(\xi_n)$ must be a monotonic increasing function,

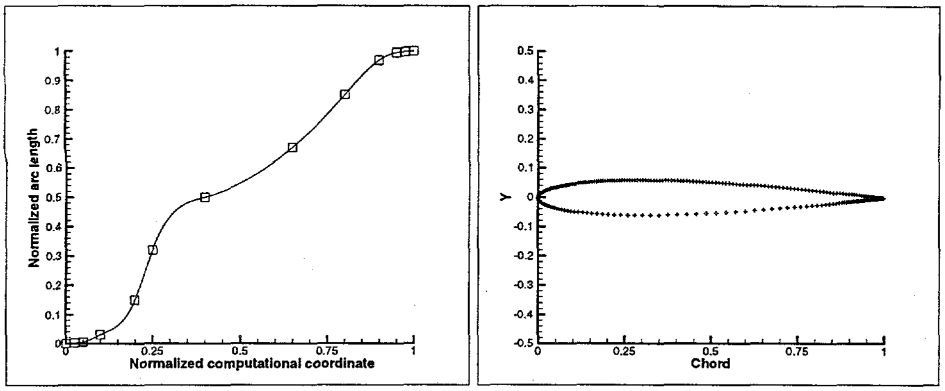

Figure 3. Point distribution with cell concentration near trailing, leading and on upper side of the airfoil based on 13 control points.

where the normalized computational coordinate ξ_n is defined by

$$\xi_n = \frac{\xi - 1}{\mathrm{NI} - 1} \; .$$

The first two conditions are easily satisfied by always specifying the first control point as (0,0) and the last as (1,1). The third condition is more intricant to assure, and here some trial and error may be necessary.

3 Manual for HypGrid2D

The described hyperbolic equation system is implemented in a program called HypGrid2D. HypGrid2D is capable of generating C-, O- and "H"-meshes. The program is controled by four input files:

1 gcf.dat : The main control file for the mesh generator. The file consists of 19 command lines, see below.

2 prof.dat: A file holding the geometry description, for the η= 1 curve. It is assumed that the ξ, η system forms a right hand system together with a ζ axis pointing out of the paper. As a consequence the description of a airfoil must be given i clockwise direction, starting from the trailing edge on the bottom side.

3 dist.dat: A file holding the discrete points used for controlling the distribution function.

4 high.dat: A file holding the discrete points used for specifying the scaling of the cell height as function of ξ relative to the height given in the "gcf.dat" file. This file is optional and no scaling is performed if the file do not exist.

The program delivers four output files:

1 grid.X2D: A file holding the resulting mesh in Basis2D format

2 grid.DAT: A file holding the resulting mesh in single block format

3 prof.DAT: A file holding the resulting surface mesh at $\eta = 1$.

4 dist.DAT: A file holding the resulting spline through the points given in the "dist.dat" file.

gcf.dat

The file named "gcf.dat" is the main control file for the mesh generator. The file consists of 19 command lines with the following information.

Line	Description	Type
1	Comment line	
2	The type of mesh to generate. The present version of HypGrid2D has three possibilities, a C-mesh indicated by the value 1, an O-mesh indicated by the value 2, and an "H"-mesh indicated by the value 3.	Integer
3	The number of cells in the ξ and η direction respectively. The present version of HypGrid2D allows for mesh sizes up to 500 × 200.	Integer,Integer
4	Approximate height of the mesh.	Real*8
5	Approximate height of the first cell.	Real*8
6	Stretching function 1=tanh, 2=sinh	Integer
7	Surface representation, use spline representation if ".true." else use linear interpolation	Logical
8	Volume blending factor, used to obtain equal sized cells in the farfield. The value must be in the range [0:0.4]	Real*8
9	Dissipation factor. This factor is used to scale the adaptive dissipation, and most often a value of 1 will give satisfactory results.	Real*8
10	Number of smoothing passes used in the 'Laplacian' smoother, typical values from 4 to 8 is used.	Integer
11	Smooth factor for the explicit smoother, typical values in the range [0:0.4].	Real*8
12	Comment line	
13	The number of cells in the wake (ξ-direction) in the case of a C-mesh, in case of an O-mesh this information is not used.	Integer
14	Length of the wake region in case of a C-mesh, in case of an O-mesh this information is not used.	Real*8
15	Intended flow angle. Is used in order to align the wake cut of a C-mesh with the flow. In case of an O-mesh this value should be set equal to zero to start with.	Real*8
16	Wake angle at trailing edge. Is used in order to align the wake-cut of a C-mesh with the flow close to the trailing edge. This information is not used in case of an O-mesh.	Real*8
17	Comment line	
18	The number of outlet cells on upper and lower side of the wake cut in case of an O-mesh, in case of a C-mesh this information is not used	Integer
19	Wake contraction is used for an O-mesh to obtain clustering near the trailing edge cut, typical values in the range [0:0.8].	Real*8

4 C-mesh

In the following a series of C-meshes will be generated around a NACA-0012 airfoil to demonstrate some of the possibilities of the HypGrid2D program. The meshes are for demonstration purpose only. For actual flow computations some parameters (domain height, cell height at the wall etc.) will have to be adjusted to other values.

4.1 Baseline mesh

The "gcf.dat" file used for the baseline mesh is given below. The mesh has 288 cells in the ξ-direction and 48 in the η-direction, resulting in a Basis2D mesh of 6 blocks of 48 × 48 cells. With 48 cells in the wake region, this assures that block edges are placed directly at the trailing edge of the airfoil.

```
*********** General commands *******************************************
1          : Meshtype 1=cmesh, 2=omesh, 3=hmesh
288 48     : Number of cells in ksi, and eta (ni,nj)
8.d0       : Approximate height of domain
2.d-5      : Approximate height of first cell
1          : Stretching function tanh=1, sinh=2
.true.     : Surface representation (.true.-> spline or linear)
0.0d       : Blending of volume and mean volume
1.d0       : Numerical dissipation factor
0          : Number of smoothing sweeps
0.d0       : Smooth factor
*********** C-mesh commands *******************************************
48         : Cells in wake (c/h-mesh)
8.d0       : Approximate lenght of wake (c/h-mesh)
0.d0       : Intended flow angle
0.d0       : Wake angle at trailing edge
*********** O-mesh commands *******************************************
0  0       : Outlet part of boundary (o-mesh) (ioutu,ioutl)
0.d0       : Wake contraction (o-mesh)
```

The "dist.dat" file used to distribute the points on the airfoil is given below, and figure 4 shows is along with the resulting distribution curve. Notice that the first line holds the number of entries in the file. The format of the line is: first character is #, followed by a single space, and finally the number of entries in the file.

```
# 13
0.00e00 0.0e00
7.00e-2 8.00e-3
1.20e-1 3.00e-2
2.00e-1 1.50e-1
2.80e-1 3.50e-1
3.90e-1 4.70e-1
5.00e-1 5.00e-1
6.10e-1 5.30e-1
7.20e-1 6.50e-1
```

```
8.00e-1 8.50e-1
8.80e-1 9.70e-1
9.30e-1 9.92e-1
1.00e00 1.00e00
```

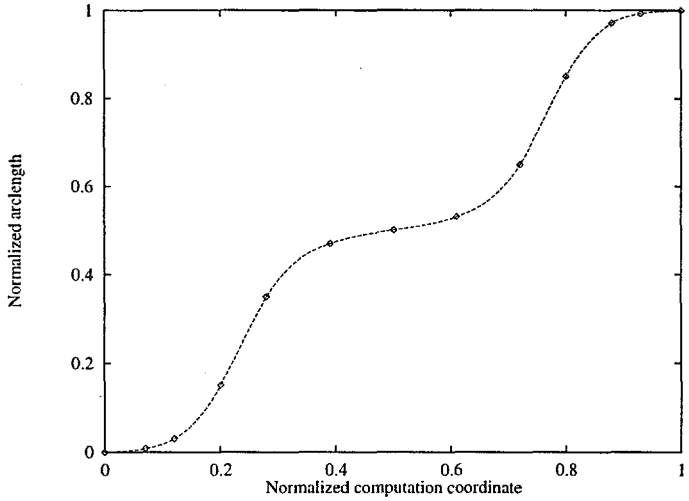

Figure 4. Figure showing the distribution function generated from the "dist.dat" file

The scaling of the cell height obtained by using the "high.dat" file is not used in the present example and the following simple "high.dat" file is used. The number of entries in the file is given in accordance with the format described for the "dist.dat" file.

```
# 2
0.00 1.0
1.00 1.0
```

The description of the airfoil surface is given in the "prof.dat" file in clockwise direction starting from the bottom side of the trailing edge:

```
# 201
     1.000000000          .0000000000E+00
     .9997532964         -.3521839608E-04
     .9990133643         -.1408203680E-03
     .9977809787         -.3165146627E-03
     .9960573316         -.5617689458E-03
     .9938441515         -.8758968324E-03
     .9911436439         -.1258070697E-02
     .9879583716         -.1707236865E-02
     .9303709865          .9537035599E-02
     .
     .
     .
     .9755282402          .3443375695E-02
     .9801468253          .2801339841E-02
     .9842915535          .2222120529E-02
     .9879583716          .1707236865E-02
```

```
.9911436439          .1258070697E-02
.9938441515          .8758968324E-03
.9960573316          .5617689458E-03
.9977809787          .3165146627E-03
.9990133643          .1408203680E-03
.9997532964          .3521839608E-04
1.000000000          .0000000000E+00
```

The resulting mesh is shown in figure 5, having good orthogonality and smoothness.

With respect to Basis2D related boundary conditions the following attributes are specified. For ξ=1,η=[2;NJ-1] and ξ=NI η=[2;NJ-1] outlet conditions (attribute=401) are specified. For η=NJ,ξ=[1,NI] inlet conditions (attribute=201) are specified. For η=1, wall conditions (attribute=101) are specified for the ξ=[IWAKE+1,NI-IWAKE], while the remaining part of η=1 are set as internal interface (attribute=1).

4.2 Stretching function

In figure 6 the result of using the sinh stretching function is shown for comparison. This is obtained by changing line 6 in the "gcf.dat" file into:

```
2           : Stretching function tanh=1, sinh=2 (dist_func)
```

The result is clearly visible in top of figure 6, where obviously more cells are clustered near the airfoil in the η-direction, giving smaller stretching near the airfoil and large cells in the farfield region. For flows where the region of interest is located close to the airfoil surface, this mesh may be superior to the Baseline mesh with the tanh stretching.

4.3 Volume blend

The previous meshes both have a unfortunate clustering of grid lines in the ξ-direction perpendicular to the airfoil originating at the trailing edge. This feature can be removed, by using the volume blending feature of the HypGrid2D program. By making the following change to line 8 in the "gcf.dat" file:

```
1.d-3       : Blending of volume and mean volume
```

The results shown in figure 7 is obtained, where it is obvious that the clustering is removed.

4.4 Alignment of wake

For high angles of attack, the flow will intersect the mesh cut at a angle. To align the mesh cut with the flow direction, it is possible to specify an angle of attack in the "gcf.dat" file in line 15:

```
25.d0       : Intended flow angle
```

As seen in figure 8 the wake is now aligned with a flow angle of 25 degrees. Aligning the mesh cut and thereby the mesh with the flow, will result in less numerical diffusion.

4.5 Wake near trailing edge

The angle at which the wake leaves the airfoil can also be controlled, this can be necessary at high angles of attack, where the use of the previous angle of attack specification may result in large angles between the airfoil and the wake. By specifying the following command in line 16 of the "gcf.dat" file this feature can be controlled.

```
25.d0        : wake_angle    # wake angle
```

The effect of the above command can be seen in figure 9.

4.6 Final mesh

The combination of the last three modification can now be combined into a final mesh, with alignment of wake in the farfield and near the airfoil, and volume blending to avoid cell clustering. Line 8, 15 and 16 in the "gcf.dat" file are modified in the following way:

```
.
.
1.d-3        : Blending of volume and mean volume
.
.
25.d0        : Intended flow angle (flow_angle)
25.d0        : wake_angle    # wake angle
```

The final mesh is shown in figure 10.

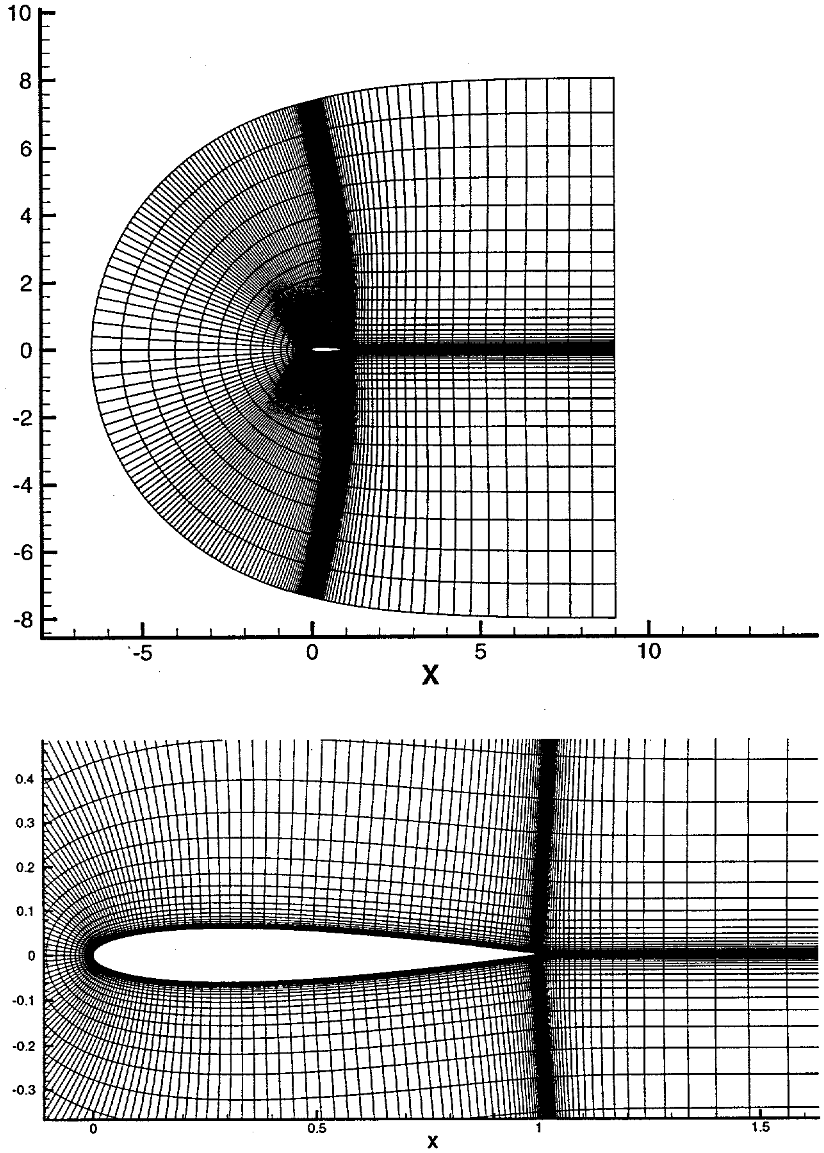

Figure 5. Baseline mesh

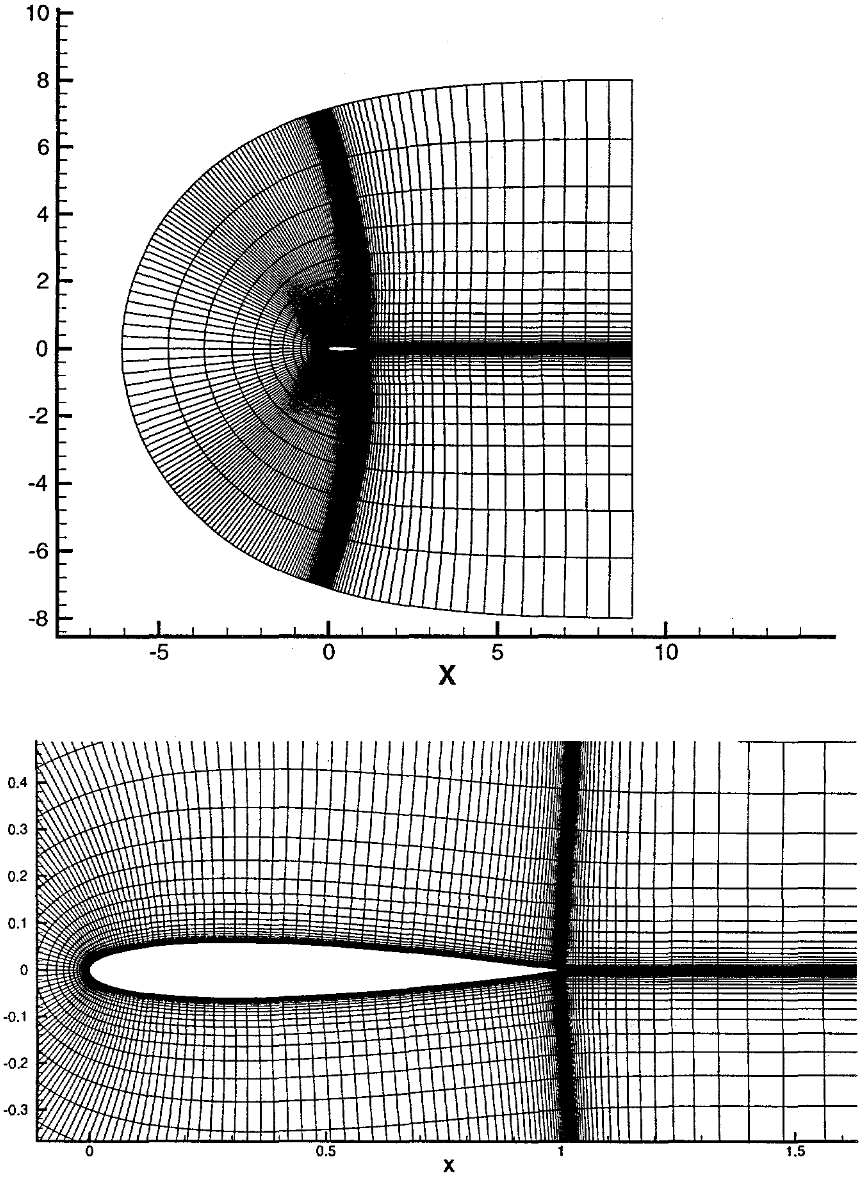

Figure 6. Effect of hyperbolic tangent stretching

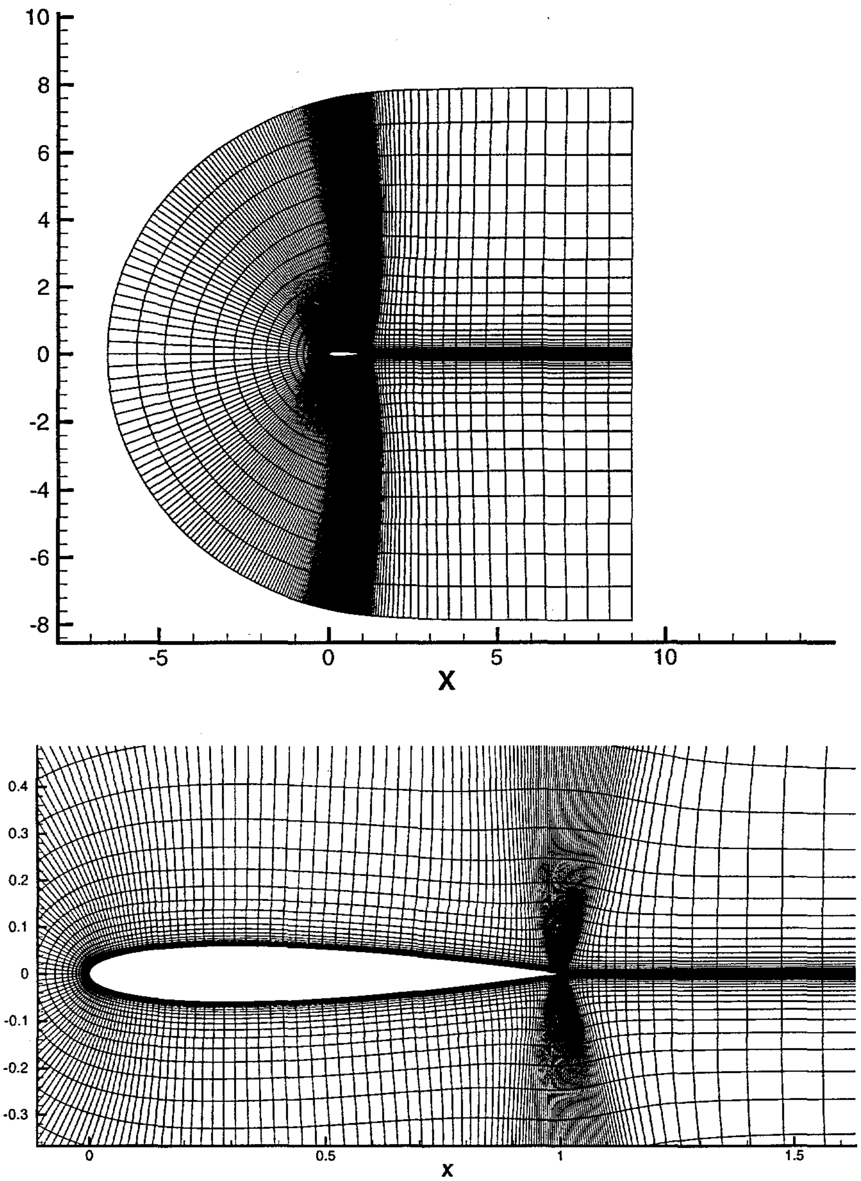

Figure 7. Effect of volume blending

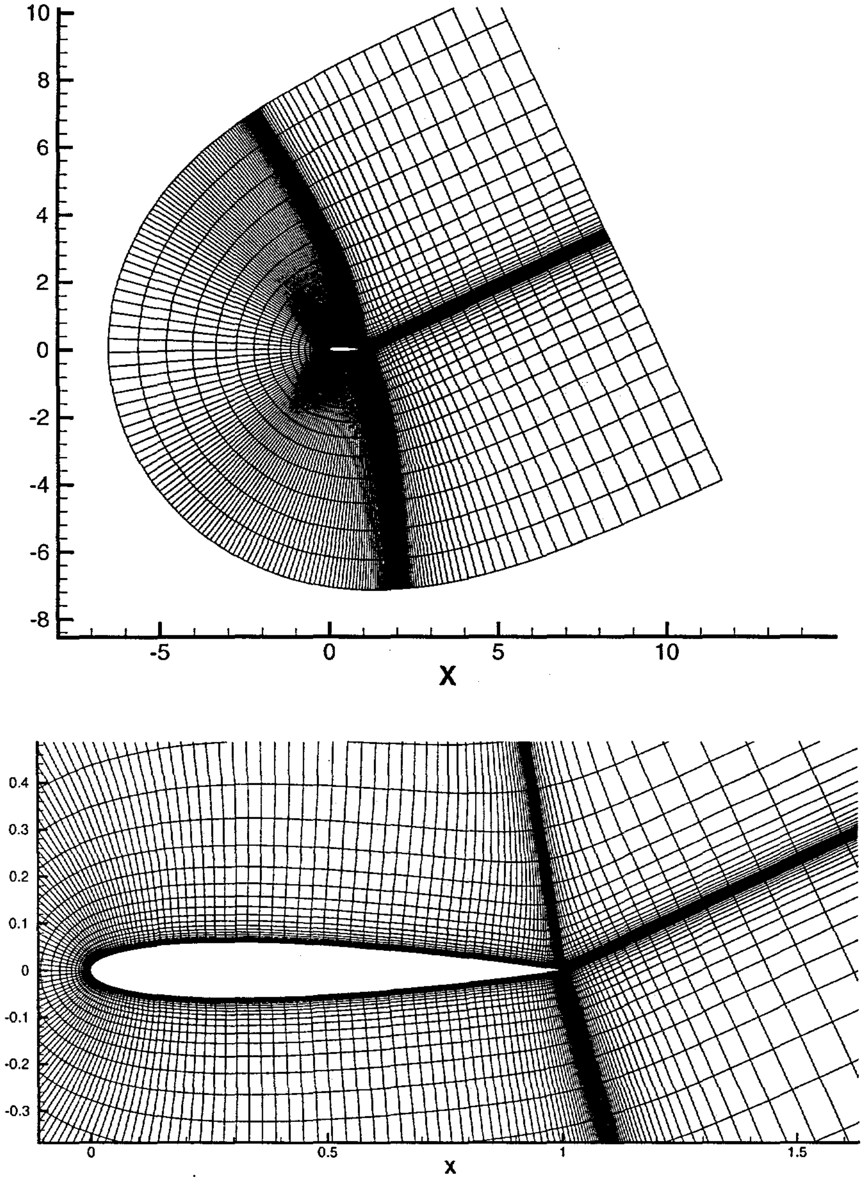

Figure 8. Allignment of wake cut with intended flow angle

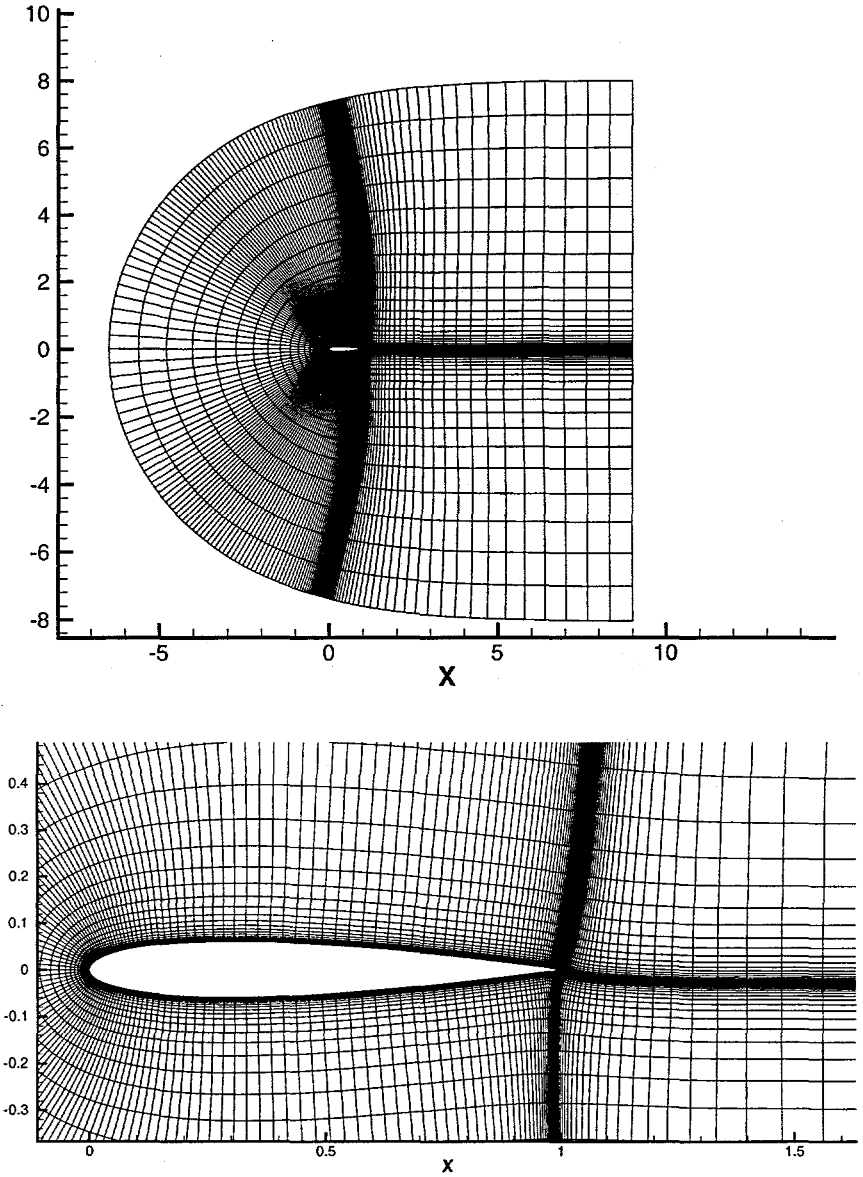

Figure 9. Effect of adjustment of wake angle near trailing edge

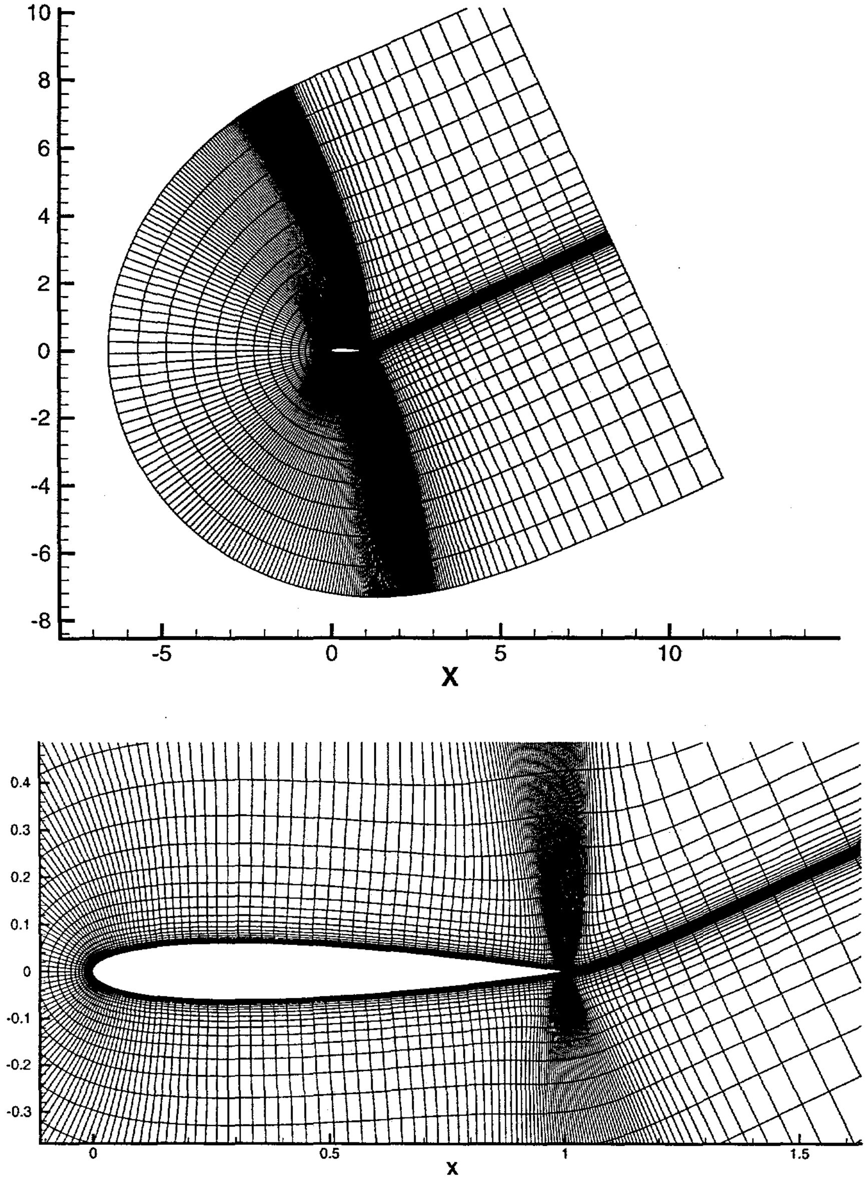

Figure 10. Final mesh, with wake allignment, adjustment of wake angle near trailing edge, and volume blending

5 O-mesh

In the following a series of O-meshes will be generated around a NACA-65-418 airfoil to demonstrate some of the possibilities of the HypGrid2D program. The meshes are generated for a demonstration purpose only, and for actual flow computations some parameters (domain height, cell height at the wall etc.) may have to be adjusted to other values.

5.1 Baseline mesh

The "gcf.dat" file used for baseline mesh is given below. The mesh has 288 cells in the ξ-direction and 48 in the η-direction, resulting in a Basis2D mesh of 6 blocks of 48 × 48 cells.

```
*********** General commands *******************************************
2            : Meshtype 1=cmesh, 2=omesh, 3=hmesh
288 48       : Number of cells in ksi, and eta (ni,nj)
8.d0         : Approximate height of domain
1.d-5        : Approximate height of first cell
1            : Stretching function tanh=1, sinh=2
.true.       : Surface representation (.true.-> spline or linear)
0.0d         : Blending of volume and mean volume
1.d0         : Numerical dissipation factor
0            : Number of smoothing sweeps
0.d0         : Smooth factor
*********** C-mesh commands *******************************************
0            : Cells in wake (c/h-mesh)
0.d0         : Approximate lenght of wake (c/h-mesh)
0.d0         : Intended flow angle
0.d0         : Wake angle at trailing edge
*********** O-mesh commands *******************************************
48 48        : Outlet part of boundary above and below wake cut
0.d0         : Wake contraction (o-mesh)
```

The "dist.dat" file used to distribute the points on the airfoil is given below.

```
# 11
0.00e00 0.0e00
7.00e-2 6.50e-3
1.20e-1 3.00e-2
2.00e-1 1.50e-1
3.50e-1 4.00e-1
5.00e-1 5.00e-1
6.50e-1 6.00e-1
8.00e-1 8.50e-1
8.80e-1 9.70e-1
9.30e-1 9.935e-1
1.00e00 1.00e00
```

The scaling of the cell height obtained by using the "high.dat" file is not used in the present example and the following simple "high.dat" file is used.

```
# 2
0.00 1.0
1.00 1.0
```

The description of the airfoil surface is given in the "prof.dat" file in clockwise direction starting from the bottom side of the trailing edge:

```
# 201
     1.000000000           .0000000000E+00
     .9997518659           .2554276944E-04
     .9990084767           .8026084106E-04
     .9977713227           .1516961202E-03
     .9960417747           .2330727875E-03
     .9938217402           .3193760931E-03
     .
     .
     .
     .9705204368           .3561293939E-02
     .9755964875           .2993082860E-02
     .9802042842           .2476949012E-02
     .9843391180           .2008545678E-02
     .9879968166           .1585443271E-02
     .9911737442           .1206855755E-02
     .9938665628           .8734043222E-03
     .9960729480           .5869282759E-03
     .9977906346           .3504792985E-03
     .9990181923           .1684939052E-03
     .9997547865           .4754499605E-04
     1.000000000           .0000000000E+00
```

The resulting mesh is shown in figure 11. The mesh possesses good orthogonality and smoothness, but it is clear that the spreading of the cells in the trailing edge region is undesirable.

With respect to Basis2D related boundary conditions the following attributes are specified. For η=NJ inlet conditions (attribute=201) are specified for the ξ=[IOUTL+1,NI-IOUTU], while outlet conditions (attribute=401) are specified for the remaining part of the η=NJ boundary. For ξ=1,ξ=[2,NJ-1] and ξ=NI η=[2,NJ-1] internal interfaces are specified (attribute=1). For η=1,ξ=[1,NI] wall conditions (attribute=101) are specified.

5.2 Wake contraction

In figure 12 the result of using the wake contraction parameter is shown. This is obtained by changing line 19 in the "gcf.dat" file into:

```
0.2d0      : Wake contraction (o-mesh)
```

The result is clearly visible both in the total and the detailed view of figure 12, where the spreadin of the mesh in the wake region is removed.

5.3 Airfoil rotation

In figure 11 and 12 it is seen that that both the baseline mesh and the reclustered mesh has some undesirable jumps in cell size near the wake cut. By rotating the airfoil, until the angle bisection line of the trailing edge is horizontal, this problem can be minimized. The parameter for intended flow angle of the HypGrid2D program can be used for this purpose. By making the following change to line 15 in the "gcf.dat" file:

```
-8.d0       : Intended flow angle (flow_angle)
```

Even though the difference may not be that great in the present example a clear difference is seen near the airfoil in the cut region, where the grid is more uniform compared to the baseline grid, see figure 13.

5.4 Final mesh

By combining the wake contraction, the rotation of the airfoil, and using the volume blending function illustrated in the C-mesh example previously we obtain the final mesh, see figure 14. The lines 8, 15 and 19 are changed into:

```
1.d-3       : Blending of volume and mean volume
  .
  .
  .
-8.d0       : Intended flow angle (flow_angle)
  .
  .
0.2d0       : Wake contraction (o-mesh) (wake_contract)
```

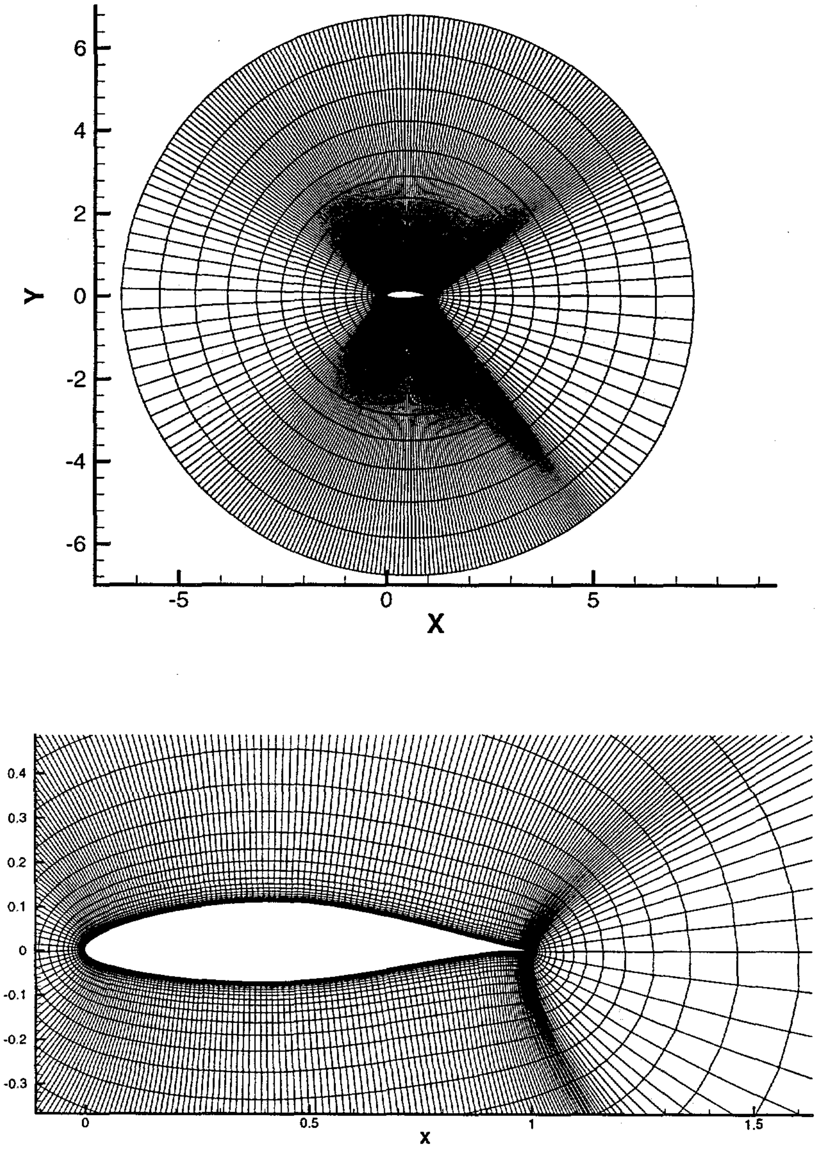

Figure 11. Baseline mesh

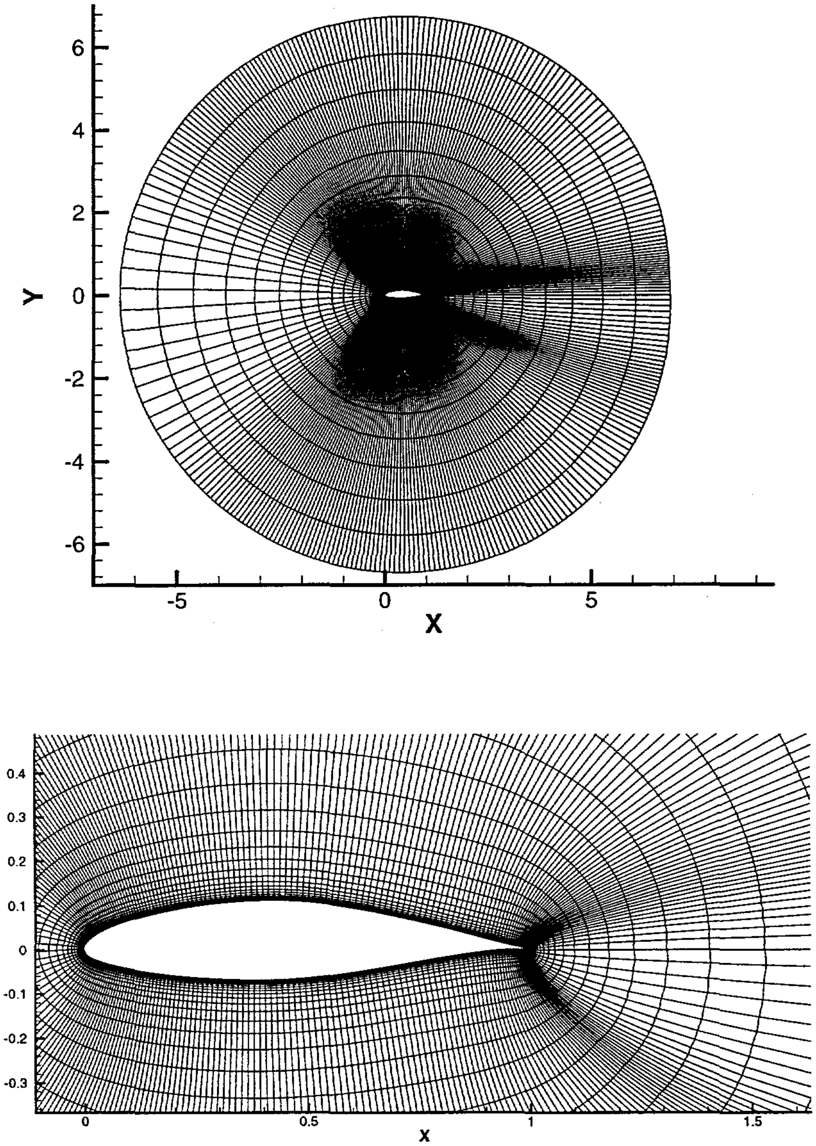

Figure 12. Effect of wake contraction

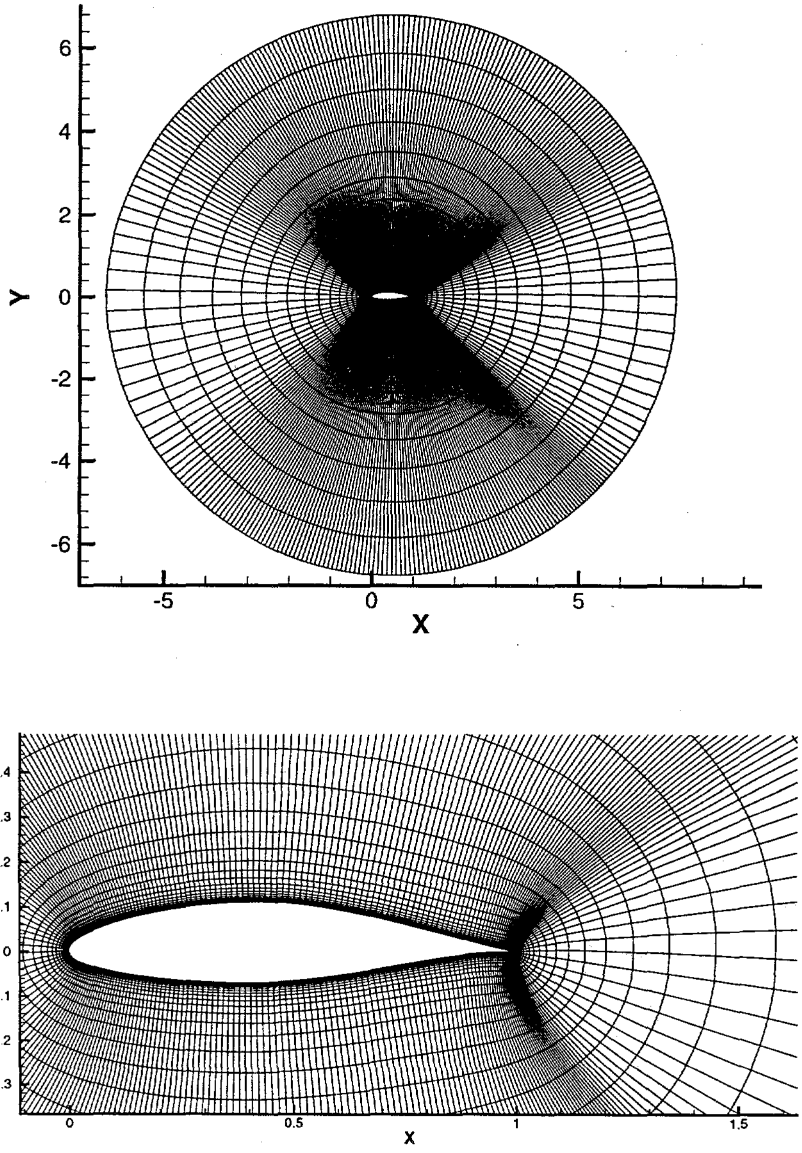

Figure 13. Effect of airfoil rotation

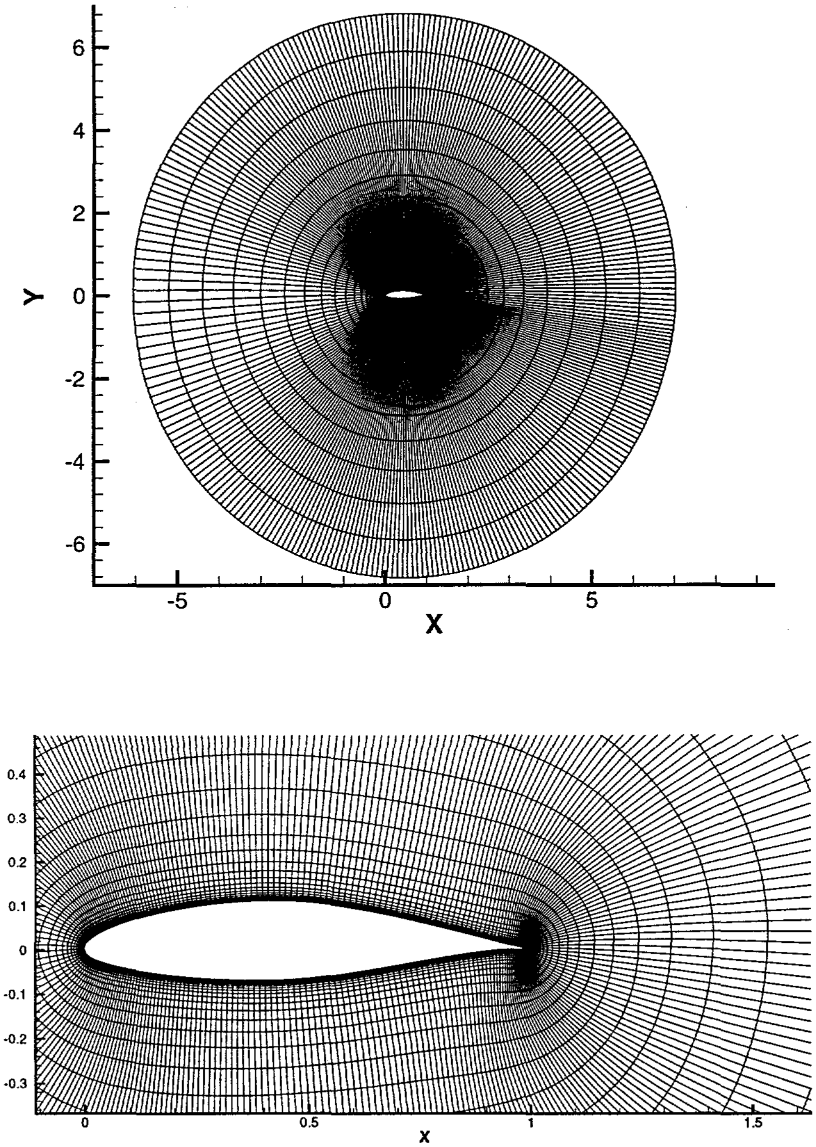

Figure 14. Final mesh with wake contraction and volume blending

6 Hill mesh

In the following two examples of meshing of "natural" terrain is given, a Gaussian hill and an escarpment. The Gaussian hill is a smooth shape, whereas the escarpment has a discontinuity at the start and end of the terrain elevation. Non of these shapes are especially challenging for the grid generator, but are shown because of the practical importance in connection with natural terrain computations.

6.1 Gaussian hill

A Gaussian hill given by the following formula, can be used as a typical example of a steep hill:

$$h(x) = h_o \exp(-(x/L)^2 \log(2)) \ ,$$

where h_o is the height of the hill (300 m), L is the half width of the hill (250 m), and x is the distance from the center of the hill.

The following grid control file 'gcf.dat' is used for the generation of the hill mesh.

```
*********** General commands *******************************************
3          : Meshtype 1=cmesh, 2=omesh, 3=hmesh
288 48     : Number of cells in ksi, and eta (ni,nj)
5.d3       : Approximate height of domain
1.d-1      : Approximate height of first cell
1          : Stretching function tanh=1, sinh=2
.true.     : Surface representation (.true.-> spline or linear)
0.d0       : Blending of volume and mean volume
1.d0       : Numerical dissipation factor
0          : Number of smoothing sweeps
0.d0       : Smooth factor
*********** C-mesh commands ******************************************
48         : Cells in wake (c/h-mesh)
6.0d3      : Approximate lenght of wake (c/h-mesh)
0.d0       : Intended flow angle
0.d0       : Wake angle at trailing edge
*********** O-mesh commands ******************************************
0 0        : Outlet part of boundary (o-mesh) (ioutu,ioutl)
0.d0       : Wake contraction (o-mesh)
```

The "dist.dat" file used to distribute the points on the part of the terrain given in the "prof.dat" file is given below:

```
# 4
0.00e00 0.0e00
0.30e00 0.4e00
0.70e00 0.6e00
1.00e00 1.00e00
```

The scaling of the cell height obtained by using the "high.dat" file is not used in the present example and the following simple "high.dat" file is used.

```
# 2
0.00 1.0
1.00 1.0
```

The Gaussian shape formula is used for generating the "prof.dat" file, 1000 m up and downstream of the summit of the hill.

```
#    200
 -.100000E+04 .457764E-02
 -.989950E+03 .571437E-02
 -.979899E+03 .711742E-02
   .
   .
   .
  .859296E+03 .833064E-01
  .869347E+03 .687069E-01
  .879397E+03 .565391E-01
  .889447E+03 .464221E-01
  .899497E+03 .380301E-01
  .909548E+03 .310855E-01
  .919598E+03 .253521E-01
  .929648E+03 .206300E-01
  .939698E+03 .167498E-01
  .949749E+03 .135690E-01
  .959799E+03 .109676E-01
  .969849E+03 .884512E-02
  .979899E+03 .711742E-02
  .989950E+03 .571437E-02
  .100000E+04 .457764E-02
```

To obtain regions up and down-stream of the hill given in the 'prof.dat' file, the number of cells in the wake is set to 48 and the wake lenght to 6000 meters, see the 'gcf.dat' file. The result can be seen in figure 15, showing both a total and a close view.

With respect to Basis2D related boundary conditions, the following attributes are specified. FOR $\eta = 1$,ξ=[1:NI] wall conditions (attribute=101) are specified. For $\eta = nj$,ξ=[1:NI] and for $\xi = 1$,η=[2;NJ-1] inlets conditions (attribute=201) are specified. For $\xi = NI$,η=[2;NJ-1] outlet conditions (attribute=401) are specified.

6.2 Escarpment

The escarpment is generated using the same input files as the gaussian hill, execpt that the "prof.dat" file is changed into the following:

```
#    2
 -200.0 0.0
  200.0 200.0
```

The result of this geometry can be seen in figure 16 where good orthogonality is observed everywhere near the surface, except at the slope discontinuities.

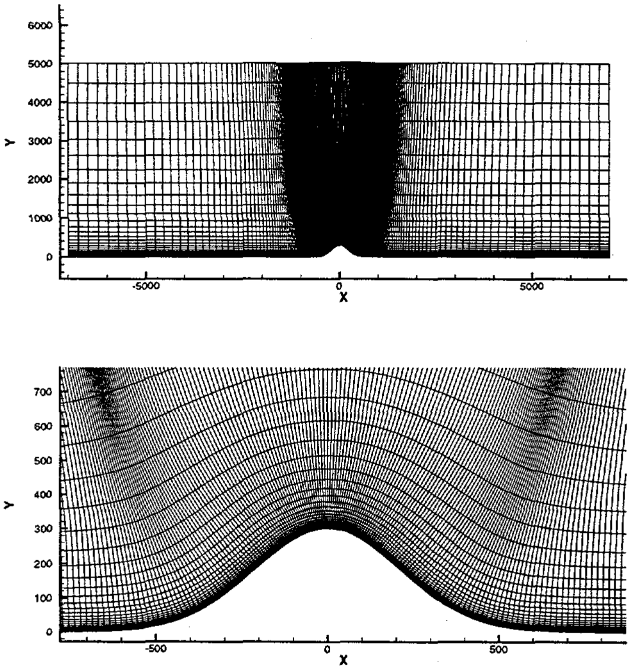

Figure 15. Total and detailed view of the "H"-mesh around the gaussian hill

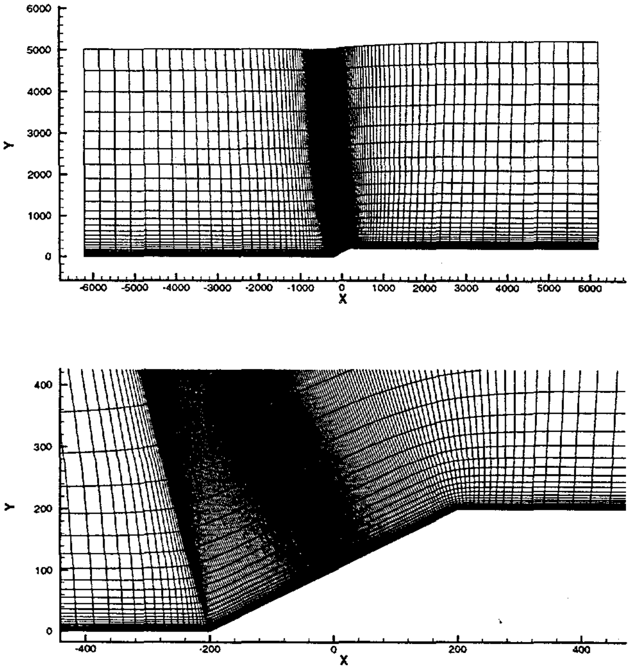

Figure 16. Total and detailed view of the "H"-mesh around the gaussian hill

7 Conclusion

The implementation of a hyperbolic mesh generation procedure, based on an equation for orthogonality and an equation for the cell face area is described. The method is fast, robust and gives meshes with good smoothness and orthogonality. The procedure is implemented in a program called HypGrid2D. The HypGrid2D program is capable of generating C-, O- and "H"-meshes for use in connection with the EllipSys2D Navier-Stokes solver.

To illustrate the capabilities of the program, some test examples are shown. First a series of C-meshes are generated around a NACA-0012 airfoil. Secondly a series of O-meshes are generated around a NACA-65-418 airfoil. Finally "H"-meshes are generated over a gaussian hill and a linear escarpment .

References

[1] Y. N. Jeng and Y. C. Liou, *Hyperbolic Equation Method of Grid Generation for Enclosed Regions*, AIAA JOURNAL, 34 (1996), pp. 1293–1295.

[2] J. A. Michelsen, *Basis3D - a Platform for Development of Multiblock PDE Solvers*, Tech. Report AFM 92-05, Technical University of Denmark, 1992.

[3] N. N Sørensen, *General Purpose Flow Solver Applied to Flow over Hills*, Risø-R- 827-(EN), Risø National Laboratory, Roskilde, Denmark, June 1995.

[4] J. L. Steger and D. S. Chaussee, *Generation of Body-Fitted Coordinates using Hyperbolic Partial Differential Equations*, SIAM J. SCI. STAT. COMPUT., 1 (1980), pp. 431–437.

[5] J. L. Steger and R. L. Sorenson, *Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates*, in Numerical Grid Generation Techniques, Nasa, 1980. NASA Conference Publication 2166.

[6] C. H. Tai, D. C. Chiang, and Y. P. Su, *Three-Dimensional Hyperbolic Grid Generation with Inherent Dissipation and Laplacian Smoothing*, AIAA JOURNAL, 34 (1996), pp. 1801–1806.

[7] C. H. Tai, S. L. Yin, and C. Y. Soong, *A Novel Hyperbolic Grid Generation Procedure with Inherent Adaptive Dissipation*, Journal of Computational Physics, 116 (1995), pp. 173–179.

[8] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation Foundations and Applications*, Elsevier Science Publishers B. V., 1985.

[9] Marcel Vinokur, *On One-Dimensional Stretching Functions for Finite-Difference Calculations*, Journal of Computational Physics, 50 (1983), pp. 215–234.

Bibliographic Data Sheet

Risø-R-1035(EN)

Title and author(s)

HypGrid2D
a 2-D Mesh Generator

Niels N. Sørensen

ISBN	ISSN
87-550-2368-1	0106-2840

Dept. or group	Date
The Test Station for Wind Turbines Dept. of Wind Energy and Atmospheric Physics	March 1998

Groups own reg. number(s)	Project/contract No.

Pages	Tables	Illustrations	References
36	1	16	7

Abstract (Max. 2000 char.)

The implementation of a hyperbolic mesh generation procedure, based on an equation for orthogonality and an equation for the cell face area is described. The method is fast, robust and gives meshes with good smoothness and orthogonality. The procedure is implemented in a program called HypGrid2D. The HypGrid2D program is capable of generating C-, O- and "H"-meshes for use in connection with the EllipSys2D Navier-Stokes solver.
To illustrate the capabilities of the program, some test examples are shown. First a series of C-meshes are generated around a NACA-0012 airfoil. Secondly a series of O-meshes are generated around a NACA-65-418 airfoil. Finally "H"-meshes are generated over a gaussian hill and a linear escarpment.

Descriptors INIS/EDB

AIRFOILS; COMPUTERIZED SIMULATION; CURVILINEAR COORDINATES; FLOW MODELS; MESH GENERATION; NAVIER-STOKES EQUATIONS; TWO-DIMENSIONAL CALCULATIONS

Available on request from:
Information Service Department, Risø National Laboratory
(Afdelingen for Informationsservice, Forskningscenter Risø)
P.O. Box 49, DK-4000 Roskilde, Denmark
Phone (+45) 4677 4004 · Fax (+45) 4677 4013