INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **32**: 605–618

An algebraic boundary orthogonalization procedure for structured grids

Reijo Lehtimäki*

Laboratory of Aerodynamics, Helsinki University of Technology, PO Box 4400, FIN-02015 HUT, Espoo, Finland

SUMMARY

An algebraic procedure for grid orthogonalization has been developed. It is often difficult to include both grid clustering and orthogonalization in a grid generation method. Often the degree and extent of orthogonality are hard to control when orthogonalization is included in a complicated grid generation method. Fortunately, grid orthogonalization can be performed independently of grid generation. The orthogonalization method developed is simple and includes invertibility control. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: computational fluid dynamics; grid generation; structured

1. INTRODUCTION

In addition to grid regularity, i.e. non-invertibility, there are basically three qualities required of a structured grid. These are grid clustering to a specified region, grid orthogonality, at least to the physical boundaries, and smoothness. There is a vast spectrum of methods developed for grid generation. In algebraic methods and methods originating from partial differential equations (PDEs), grid orthogonality at the boundary is generally either inherent or achievable via some parameters. In variational methods, different grid qualities, e.g. orthogonality, are weighted against each other when forming the functional to be optimized. Incorporating all the grid qualities in a single method often makes the method quite fragile. In algebraic methods, boundary orthogonality can often be quite naturally arranged, however, grid folding is possible. Transfinite interpolation with Hermite polynomials, for example, provides inherent orthogonality properties. Transfinite interpolation can also be generalized to interpolate the derivatives together with the co-ordinate values [1]. Other algebraic methods include grid generation based on a homotopy procedure in which the othogonalization is based on local perturbations of a homotopy parameter [2]. In all these methods grid orthogonalization is

* Correspondence to: Laboratory of Aerodynamics, Helsinki University of Technology, PO Box 4400, FIN-02015 HUT, Espoo, Finland.

CCC 0271–2091/2000/050605–14$17.50

Received November 1998
Revised January 1999

embedded in the method and performed simultaneously with grid generation. The control point form [3] of transfinite interpolation allows grid orthogonalization after the grid generation via movement of the control points.

The Poisson system also makes it possible to achieve orthogonality by adding an outer iteration loop in which the control terms are adjusted [4,5]. But then a different Poisson system is solved on each inner iteration loop and the control terms depend on the result of the previous outer iteration cycle. This construction hardly increases the stability of the method and certainly increases the computing time. However, a lot of time has been devoted to the development of the determination of the control terms in the Poisson system. Control terms can be determined at the boundary from the boundary data assuming orthogonality conditions and stating a desired first spacing off the boundary [4]. The control terms can then be interpolated into the field. Another approach is to determine the control terms according to a combined map of the computational domain to the clustered computational domain and from thereof to the physical domain; the inverse of the latter satisfying the Laplace equation [6]. In this approach, orthogonality at the boundary is achieved by solving the Laplace system on the physical domain one additional time, now with Dirichlet and Neumann boundary conditions imposed on different boundary segments. This is to modify the clustered computational domain so that the actual solution of the grid generation system will then produce a grid orthogonal at boundary. If a continuous representation of the boundary is available then one possibility is to apply Neumann boundary conditions to achieve boundary orthogonality [7]. Other elliptic systems include the biharmonic system, which allows both Dirichlet and Neumann boundary conditions to be applied at the same time [8]. In all of these methods, the orthogonalization is coupled with the grid generation. The biharmonic method does not suffer from nested iteration loops, where the control terms would be determined according to a rule not belonging to the equation itself. However, a biharmonic equation does not possess a maximum principle either [9].

In fact, incorporating all the grid qualities in a single method makes the grid generation problem overdetermined. Consider a grid of N interior grid points and assume that the boundary points are pre-determined. The unknowns of the problem are then the Cartesian co-ordinates of the interior points. The number of unknowns is now $2N$ in the planar two-dimensional and $3N$ in the three-dimensional case. The ratio of the grid segment lengths in the upward and downward direction in some curvilinear co-ordinate direction determines the clustering in that direction at that point. Determining the clustering in all the curvilinear directions at all the interior points results in $2N$ equations in the two-dimensional case and $3N$ equations in the three-dimensional case. The number of equations is the same as the number of variables and the problem is already fully determined. Thus, additional requirements, like grid orthogonality, imply compromising grid clustering to some degree. It is, however, important for the grid to be at least nearly orthogonal to the physical boundaries, especially when solving the thin-layer approximation for the Navier–Stokes equations. Fortunately, the inclusion of the orthogonalization in the grid generation method is not necessary; in fact the grid orthogonalization can be separated from the grid generation method altogether. This paper presents a simple algebraic orthogonalization routine that can be used to orthogonalize the grid at the physical boundary segments. A regularity check is also applied in order to discontinue the orthogonalization before the grid becomes folded.

2. THE GENERATION OF THE CLUSTERED COMPUTATIONAL DOMAIN

Many grid generation methods are based on mapping the physical domain to a uniform unit square or unit cube called the computational domain. The points in the computational domain can be redistributed to form a non-uniform computational domain called the clustered computational domain. The components of the redistribution map are called stretching functions. Stretching functions can be included in almost any method of grid generation to achieve grid clustering. It is fairly easy to incorporate stretching functions in transfinite interpolation [10]. There is a way to apply stretching functions also to the Laplace system [11]. There is a straightforward way to include stretching functions in direct discrete grid generation as well. As an example of this, weight functions can be used to generalize the segment length functional of Castillo [12] to allow grid clustering [13]. A natural choice for the weight functions is to use segment lengths of the clustered computational domain. In all of these methods, the clustering of the grid to be generated on the physical domain will follow the clustering of the clustered computational domain. In the present study, the clustered computational domain is applied by forming weights for damping the orthogonalization process.

Since an existing grid is to be orthogonalized, the boundary points of the physical domain are pre-determined. In the two-dimensional case, the boundary point distribution of the clustered computational domain can be easily determined according to the normalized curve lengths between the grid points on the boundary of the physical domain. The interior of the computational domain can then be interpolated choosing the grid lines as straight lines. Thus, an interior point $\bar{\rho}_{i,j}$ of the clustered computational domain is at the intersection of the lines connecting $\bar{\rho}_{0,j}$ to $\bar{\rho}_{L,j}$ and $\bar{\rho}_{i,0}$ to $\bar{\rho}_{i,M}$ (Figure 1). In the three-dimensional case, the edges and faces of the computational domain can be generated as in the two-dimensional case, the edges according to the relative curve length of the physical domain and the faces by intersection of straight lines.

Three lines in three space dimensions do not generally intersect, which makes it impossible to apply the method of intersections in the calculation of the interior points of the volume. The three-dimensional clustered computational space can be generated by minimizing the functional F for each interior point

$$F(\bar{\rho}_{i,j,k}) = |\bar{\rho}_{i,j,k} - \bar{\rho}_{0,j,k}| + |\bar{\rho}_{i,j,k} - \bar{\rho}_{L,j,k}| + |\bar{\rho}_{i,j,k} - \bar{\rho}_{i,0,k}| + |\bar{\rho}_{i,j,k} - \bar{\rho}_{i,M,k}| + |\bar{\rho}_{i,j,k} - \bar{\rho}_{i,j,0}| + |\bar{\rho}_{i,j,k} - \bar{\rho}_{i,j,N}| \quad (1)$$

where $\bar{\rho} = (\xi, \eta, \zeta)^T$. The minimization of the functional F then yields the interior point $\bar{\rho}_{i,j,k}$. The six terms in functional F are the distances to the endpoints of the three grid lines passing through the interior point $\bar{\rho}_{i,j,k}$. Note that a different functional is minimized for each interior grid point. The two-dimensional analog of Equation (1), the sum of distances to the four endpoints of the two grid lines, yields the straight grid lines mentioned above. Thus, Equation (1) can be considered as a generalization of the two-dimensional method. Note that, in the two-dimensional case, the minimizational approach need not be used since the intersection of two straight lines can be calculated from a simple algebraic formula of basic analytical

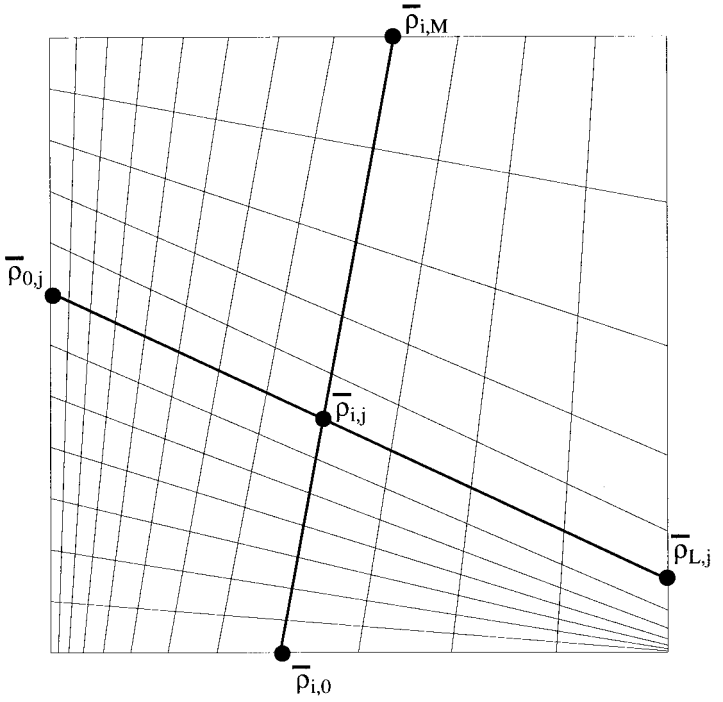

Figure 1. The interior point is at the intersection of two straight lines.

geometry. Note also that in the three-dimensional case, a strict straightness of the grid lines is not generally achievable.

The minimization of F can be effectively done by setting the gradient to zero and iterating for the solution of the system of the PDEs

$$\frac{\partial F(\bar{\rho}_{i,j,k})}{\partial \xi_{i,j,k}} = 0$$

$$\frac{\partial F(\bar{\rho}_{i,j,k})}{\partial \eta_{i,j,k}} = 0$$

$$\frac{\partial F(\bar{\rho}_{i,j,k})}{\partial \zeta_{i,j,k}} = 0 \tag{2}$$

The first of these equations, for example, yields the non-linear difference equation for the interior point $(\xi_{i,j,k}, \eta_{i,j,k}, \zeta_{i,j,k})$,

$$\frac{\xi_{i,j,k}-\xi_{0,j,k}}{|\bar{\rho}_{i,j,k}-\bar{\rho}_{0,j,k}|}+\frac{\xi_{i,j,k}-\xi_{L,j,k}}{|\bar{\rho}_{i,j,k}-\bar{\rho}_{L,j,k}|}+\frac{\xi_{i,j,k}-\xi_{0,j,k}}{|\bar{\rho}_{i,j,k}-\bar{\rho}_{0,j,k}|}+\frac{\xi_{i,j,k}-\xi_{i,0,k}}{|\bar{\rho}_{i,j,k}-\bar{\rho}_{i,0,k}|}+\frac{\xi_{i,j,k}-\xi_{i,M,k}}{|\bar{\rho}_{i,j,k}-\bar{\rho}_{i,M,k}|}$$
$$+\frac{\xi_{i,j,k}-\xi_{i,j,0}}{|\bar{\rho}_{i,j,k}-\bar{\rho}_{i,j,0}|}+\frac{\xi_{i,j,k}-\xi_{i,j,N}}{|\bar{\rho}_{i,j,k}-\bar{\rho}_{i,j,N}|}=0 \quad (3)$$

where $|\bar{\rho}_{i,j,k}-\bar{\rho}_{0,j,k}|=\sqrt{(\xi_{i,j,k}-\xi_{0,j,k})^2+(\eta_{i,j,k}-\eta_{0,j,k})^2+(\zeta_{i,j,k}-\zeta_{0,j,k})^2}$, for example. The three non-linear difference equations obtained from Equations (2) can be linearized for iteration by lagging $\xi_{i,j,k}$, $\eta_{i,j,k}$, and $\zeta_{i,j,k}$ in the denominator, i.e. taking these values from the previous iteration cycle.

3. ALGEBRAIC GRID ORTHOGONALIZATION

The grid lines emanating from the boundary are orthogonalized by moving each of the grid points on the grid line towards its orthogonal projection on the normal of the boundary. Consider the orthogonalization of a grid line of constant i to the boundary line on which $j=0$ (Figure 2). The original grid line is drawn with a solid line and the orthogonalized line is dashed. The projections of the grid points on the normal n are also marked. The magnitude of the transformation is controlled by a weight parameter w. The weight parameter may assume any value between 0 and 1. The value $w=0$ implies no transformation condition and the value $w=1$ implies undamped transformation of the point to the normal. The new location $\bar{r}'_{i,j}$ of a point $\bar{r}_{i,j}$ is calculated as

$$\bar{r}'_{i,j}=(1-w_{i,j})\bar{r}_{i,j}+w_{i,j}\bar{r}''_{i,j} \quad (4)$$

where $\bar{r}''_{i,j}$ is the orthogonal projection of $\bar{r}_{i,j}$ on the normal n.

The reason for introducing the weight parameter and damping of the orthogonalization is that, in general, the opposing boundary line and/or the boundary lines emanating from the boundary line to which the grid is to be orthogonalized are predetermined. Full orthogonalization throughout the field would then contradict these boundary conditions. Thus, the

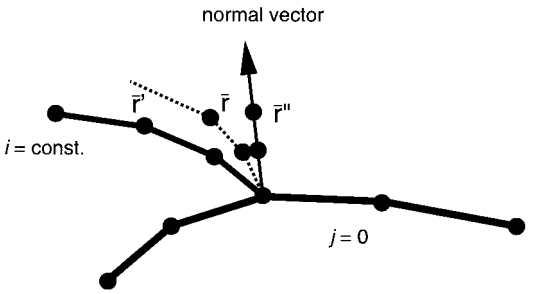

Figure 2. The orthogonalization of the grid line of constant i to the boundary line on which $j=0$.

orthogonalization is damped towards the interior of the grid as well as towards the endpoints of the boundary line of $j=0$. The weight parameter w is determined from the stretching functions, i.e. the co-ordinates ξ and η in the clustered computational domain, which reflect the actual physical distances but do not change during the orthogonalization. In the two-dimensional case, the form of the weight parameter $0 \leq w_{i,j} \leq 1$ is

$$w_{i,j} = \exp\left\{ -\alpha_1 \left[\left(1 - \frac{\tilde{\eta}_{i,j}}{\tilde{\eta}_{i,j_{\max}}} \right)^{-2} - 1 \right] \right\} \cdot [4\xi_{i,0}(1-\xi_{i,0})]^{\alpha_2} \tag{5}$$

where $\tilde{\eta}_{i,j} = \sqrt{(\xi_{i,j} - \xi_{i,0})^2 + (\eta_{i,j} - \eta_{i,0})^2}$. The first term handles the damping to the interior of the grid and the second term the damping to the endpoints of the boundary line. The $j = j_{\max}$ is the first grid line on which points are not moved at all. The α_1 and α_2 are user-defined parameters. The values of both these parameters should be positive. The greater the value of α_1, the faster the orthogonalization is damped when advancing inwards to the region. A high value of α_2 implies that only the very central region of the boundary will be orthogonalized. Also, note that the transition to the rest of the grid, which has not been orthogonalized, is smooth (C^∞).

The orthogonalization is always damped to some degree. It is thus possible to apply the orthogonalization many times in succession. Each repetition then orthogonalizes the grid further.

Equation (4) is applicable in both the two- and the three-dimensional case. In the three-dimensional case, the two directions on the boundary surface can be handled one after the other. Instead of using the normal of the surface, the first grid line segment in the out-of-surface direction is orthogonalized to the two tangential grid line directions to obtain two quasi-normals for the surface. These quasi-normals are then used to calculate the $\bar{r}''$ for the two directions. This separation of directions allows more control of the orthogonalization and also makes it possible to orthogonalize grids in one direction only, if wanted. The weight parameter $0 \leq w_{i,j,k} \leq 1$ in the three-dimensional case for the surface of $j=0$ is written as

$$\begin{aligned} w_{i,j,k} = \exp\left\{ -\alpha_1 \left[\left(1 - \frac{\tilde{\eta}_{i,j,k}}{\tilde{\eta}_{i,j_{\max},k}} \right)^{-2} - 1 \right] \right\} \cdot \left[4 \frac{\tilde{\xi}_{i,0,k}}{\tilde{\xi}_{i_{\max},0,k}} \left(1 - \frac{\tilde{\xi}_{i,0,k}}{\tilde{\xi}_{i_{\max},0,k}} \right) \right]^{\alpha_2} \\ \cdot \left[4 \frac{\tilde{\zeta}_{i,0,k}}{\tilde{\zeta}_{i,0,k_{\max}}} \left(1 - \frac{\zeta_{i,0,k}}{\tilde{\zeta}_{i,0,k_{\max}}} \right) \right]^{\alpha_3} \end{aligned} \tag{6}$$

where $\tilde{\xi}_{i,j,k} = |(\xi, \eta, \zeta)_{i,j,k} - (\xi, \eta, \zeta)_{0,j,k})|$. The $\tilde{\eta}_{i,j,k}$ and $\tilde{\zeta}_{i,j,k}$ are defined analogously. Thus, $\tilde{\xi}$ is the Euclidean distance of a point to the beginning of the i-line, on which it is located. Likewise, $\tilde{\eta}$ is the distance to the beginning of the j-line and $\tilde{\zeta}$ is the distance to the beginning of the k-line. The grid lines in the interior of the clustered computational domain not being strictly straight, the distance $\tilde{\eta}$ in Equation (6) should actually be calculated by integrating the curve length. However, the application of the Euclidean distances has not produced any problems so far.

4. GRID REGULARITY

At a complicated boundary, the orthogonalization will eventually after a sufficient number of repetitions invert the grid. The grid regularity is, therefore, verified after each application of the routine. The verification is performed by calculating a normalized cell volume, a so-called skewness value, for each of the cells affected by the orthogonalization. The grid is accepted if and only if the skewness values of all the cells are positive. If negative cells are detected, the grid before the repetition in question is returned. The orthogonalizer is not a grid resolver. Thus, if the grid is inverted to begin with, no orthogonalization is performed. Note also that the use of the positivity of the cell volume as an indicator of grid regularity is questionable. Even for a cell with a positive volume, the boundary of the cell may intersect itself as in a two-dimensional bowtie (⋈). Furthermore, is it clear that if all the cells are regular then the grid must be regular as well?

Consider the two-dimensional cell $\{\bar{r}_{i,j}, \bar{r}_{i+1,j}, \bar{r}_{i,j+1}, \bar{r}_{i+1,j+1}\}$. The signed area A is given by half of the cross-product of the diagonals $A = \frac{1}{2}(\bar{r}_{i+1,j+1} - \bar{r}_{i,j}) \times (\bar{r}_{i,j+1} - \bar{r}_{i+1,j})$. The skewness value σ is defined as the area divided by the quantities e_1 and e_2, which are the maxima of the lengths of the cell edges in the two curvilinear co-ordinate directions

$$\sigma = \frac{A}{e_1 e_2} \tag{7}$$

where

$$e_1 = \max\{|\bar{r}_{i+1,j} - \bar{r}_{i,j}|, |\bar{r}_{i+1,j+1} - \bar{r}_{i,j+1}|\}, \qquad e_2 = \max\{|\bar{r}_{i,j+1} - \bar{r}_{i,j}|, |\bar{r}_{i+1,j+1} - \bar{r}_{i+1,j}|\} \tag{8}$$

For a three-dimensional cell, the signed cell volume is normalized analogously by dividing it by three quantities, each of which is the maximum of the four edge lengths in a curvilinear co-ordinate direction. In order for the grid to be useful for flow calculations, the volume of the cell should be calculated with the same formula that the flow solver applies. The calculation of the cell volume is a problem of its own and not considered here. This problem has been considered by, among others, Kordulla and Vinokur [14].

The signature of the skewness value strictly follows the signature of the cell area or volume. Thus, a positive skewness value indicates a regular cell and a negative value indicates an inverted cell. A right-angled cell has the skewness value of $\sigma = 1$. For a degenerate cell, the skewness value vanishes, $\sigma = 0$. The more the cell deviates from being right-angled, the smaller the skewness value becomes. Thus, as its name indicates, the skewness value also reflects the skewness of the cell. Thus, it is possible to impose higher demands on the grid than plain regularity, if desired.

5. GRID EXAMPLES

Some grid examples are provided to illustrate the orthogonalization method. The original grid is also shown when appropriate.

5.1. A double-throat nozzle

A grid for a double-throat nozzle is orthogonalized to both the nozzle surface and the symmetry axis (Figure 3). Both orthogonalizations extend throughout the field and all the user-defined parameters are set to 1, the number of repetitions performed being five. In Figure 4, close-up views of the inlet of the original grid (above) as well as of the orthogonalized grid (below) are shown. The degree of orthogonalization seems quite remarkable.

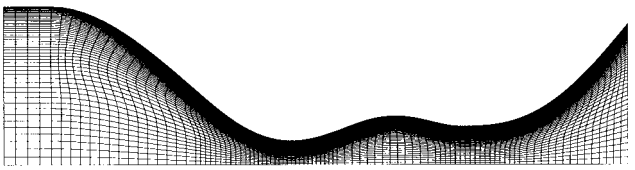

Figure 3. An orthogonalized grid for a double-throat nozzle. The grid is orthogonalized both at the nozzle surface and the symmetry axis.

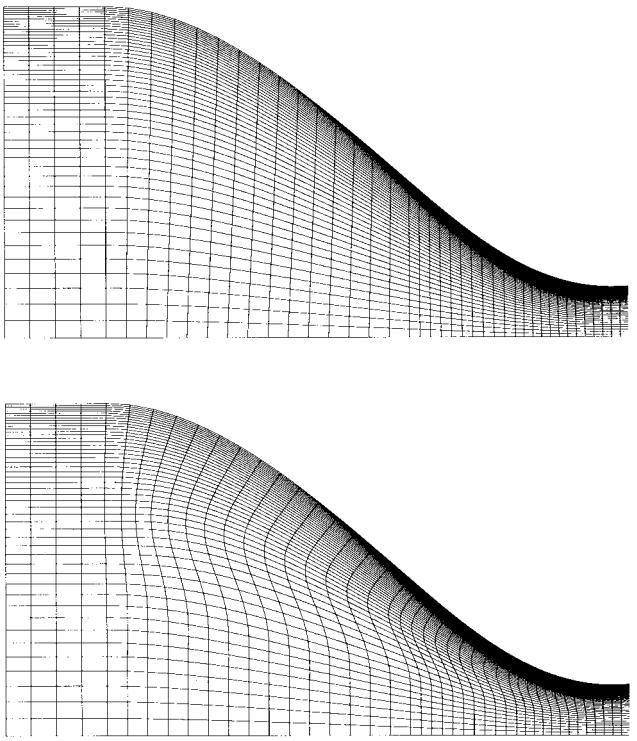

Figure 4. A close-up view of the double-throat nozzle grid at the inlet. The original grid is shown above and the orthogonalized grid below.

5.2. A cascade

A two-dimensional grid for a cascade is orthogonalized to both the lower and upper boundaries (Figure 5). Two repetitions were performed, at both boundaries. The third repetition resulted in an inverted grid and its result was thus rejected. The orthogonalization region extended 19 grid lines into the interior, the total number of grid cells in the circumferential direction being 64. The user-defined parameter controlling the damping of the orthogonalization as advancing into the interior was set as $\alpha_1 = 10$, at both boundaries. Such a high value was used to damp the orthogonalization early on, since the leading edge, and especially the trailing edge, of the blade are difficult to orthogonalize. The user-defined parameter controlling damping along the boundary was set equal to 1 at both boundaries. The trailing edge is the first region where grid inversion will occur if too much orthogonalization is applied. A closer view of this region is provided in Figure 6, in which the original grid is shown above the orthogonalized grid to allow for comparison. As can be noted, as few as two repetitions of orthogonalization clusters the grid unnecessarily in the axial direction at the trailing edge. Therefore, it is not evident that the orthogonalized grid is better than the original one for flow calculations. It might be wiser to apply orthogonalization only to the other parts of the boundary and leave the trailing edge as it is.

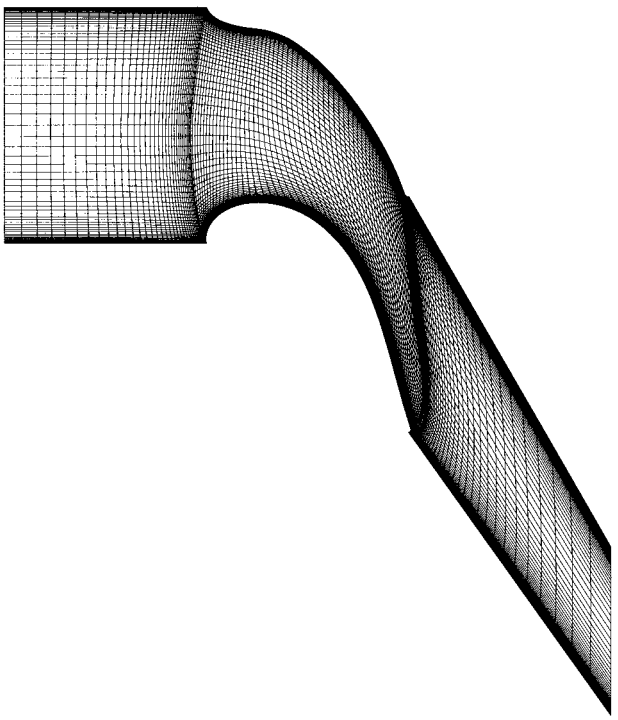

Figure 5. An orthogonalized grid for a cascade.

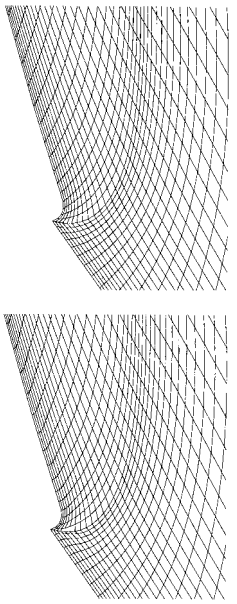

Figure 6. A close-up view at the trailing edge of the blade (lower boundary of the region). The original grid is shown above and the orthogonalized grid below.

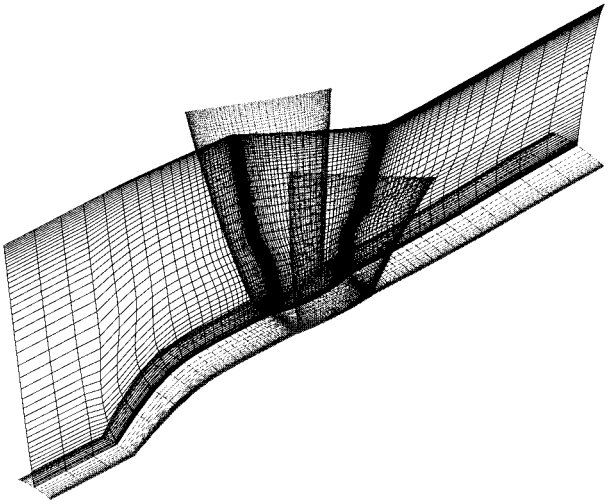

Figure 7. The surface in the middle of the blade passage of an orthogonalized grid for the NASA 55C fan rotor.

5.3. NASA 55C fan rotor

The orthogonalization method was applied for the NASA 55C fan rotor at the hub. The grid for the blade passage consists of $64 \times 48 \times 128$ cells, the numbers corresponding to the circumferential, radial and axial directions respectively. Two repetitions of orthogonalization were performed. The extent of orthogonalization was chosen as 29 grid lines into the field. The parameter controlling damping in the circumferential direction was set a value of 2, to allow a larger transition region to the block boundaries. The two other user-defined parameters were set equal to 1. The surface in the middle of the blade passage is shown in Figure 7. The most radical changes appear in the curved section at the leading edge of the hub. The grid has changed elsewhere as well, which can be noted by viewing the clustered regions at the leading and trailing edges of the blade. Figure 8 shows that the grid has become a bit unsmooth at the leading edge of the hub. Thus, some additional smoothing could be applied. The orthogonalization in the circumferential direction can be viewed in Figure 9, in which a cross-section near the leading edge of the hub is shown.

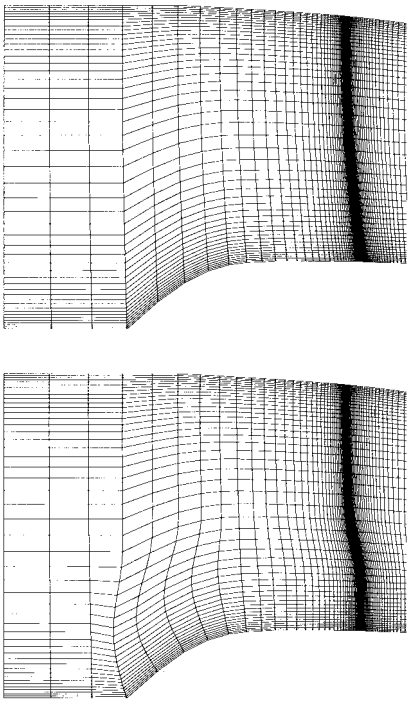

Figure 8. A close-up of the surface in the middle of the blade passage. The original grid is shown above and the orthogonalized grid below.

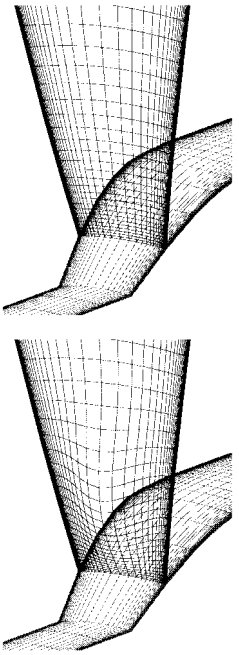

Figure 9. A cross-section of an orthogonalized grid for the NASA 55C fan rotor.

Actually, at the leading edge of the hub, where the strongest orthogonalization appears, the boundaries in the circumferential direction are inter-block boundaries and their forms are not pre-determined (as long as the form is the same for both of them). There is no limitation hindering the use of the volumetric orthogonalization procedure for surface grids. Since no attempt is made to keep the grid points on the surface, the form of the surface probably alters. This is no obstacle in this case. Consequently, the three-dimensional grid orthogonalizer was applied, in the longitudinal direction only, to the inter-block boundary upstream of the blade. The extent of orthogonalization was again chosen as 29 grid lines into the field and the user-defined parameters were set the value one. After the orthogonalization of the surface, the volume grid upstream of the blades was transfinitely interpolated. The result is depicted in Figures 10 and 11, showing the surface in the middle of the blade passage and the cross-section near the leading edge of the hub respectively. The main difference from the grid obtained earlier is that now the orthogonality carries throughout the circumferential direction, which is desirable.

6. CONCLUSIONS

Often the inclusion of orthogonality requirement to a grid generation method tends to make the method more fragile. It is also nice to be able to further orthogonalize an already generated

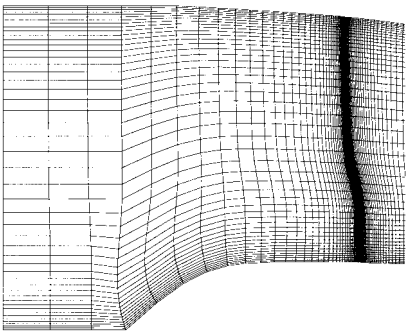

Figure 10. A close-up of the surface in the middle of the blade passage. The inter-block boundaries are orthogonalized and the volume interpolated transfinitely.

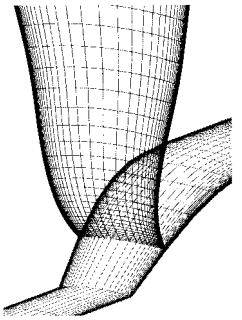

Figure 11. A cross-section of the NASA 55C fan rotor grid. The inter-block boundaries are orthogonalized and the volume interpolated transfinitely.

grid, if the orthogonality of the grid is found insufficient, without the need to regenerate it. Therefore, it is interesting to study grid orthogonalization in separation from the actual grid generation.

A simple algebraic grid orthogonalization method has been developed. The magnitude of the orthogonalization can be controlled by user-defined parameters. The orthogonalization is always damped to some degree and can be applied several times in succession. The method also refuses to invert the grid. The grid regularity is checked after each application of the orthogonalization and a prior result is returned, if the orthogonalization were to result in grid inversion.

REFERENCES

1. Smith RE, Eriksson L-E. Algebraic grid generation. *Computer Methods in Applied Mechanics and Engineering* 1987; **64**: 285–300.

2. Moitra A. Two- and three-dimensional grid generation by an algebraic homotopy procedure. *AIAA Journal* 1992; **30**(5): 1433–1434.
3. Eiseman PR. A control point form of algebraic grid generation. *International Journal for Numerical Methods in Fluids* 1988; **8**(10): 1165–1181.
4. Thompson JF. A general three-dimensional elliptic grid generation system on a composite block structure. *Computer Methods in Applied Mechanics and Engineering* 1987; **64**: 377–411.
5. Sonar T. *Grid generation using elliptic partial differential equations*. Technical Report DFVLR-FB 89-15, DFVLR: Braunschweig, 1989.
6. Spekreijse SP. Elliptic grid generation based on Laplace equations and algebraic transformations. *Journal of Computational Physics* 1995; **118**: 38–61.
7. Khamayseh A, Kuprat A, Mastin CW. Boundary orthogonality in elliptic grid generation. In *Handbook of Grid Generation*, Thompson JF, Soni BK, Weatherill NP (eds). CRC Press: Boca Raton, FL, 1999; 6-1–6-26.
8. Findling A, Herrmann U. Development of an efficient and robust solver for elliptic grid generation. In *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Arcilla AS, Häuser J, Eiseman PR, Thompson JF (eds). North-Holland: Amsterdam, 1991; 781–792.
9. Knupp P, Steinberg S. *Fundamentals of Grid Generation*. CRC Press: Boca Raton, FL, 1993; 100–101.
10. Tysell LG, Hedman SG. *Towards a general three-dimensional grid generation system*. ICAS-88-4.7.4, 1988; 1048–1058.
11. Lehtimäki R. A stretching function approach to harmonic grid generation. In *Computational Fluid Dynamics '94. Proceedings of the Second European Computational Fluid Dynamics Conference*. John Wiley: New York, 1994; 250–257.
12. Castillo JE. Discrete variational grid generation. In *Mathematical Aspects of Numerical Grid Generation*, Castillo JE (ed). SIAM: New York, 1991; 35–58.
13. Knupp P, Steinberg S. *Fundamentals of Grid Generation*. CRC Press: Boca Raton, FL, 1993; 130–131.
14. Kordulla W, Vinokur M. Efficient computation of volume in flow predictions. *AIAA Journal* 1983; **21**: 917–918.

