Proceedings of the 5th World Congress on Mechanical, Chemical, and Material Engineering (MCM'19)
Lisbon, Portugal – August, 2019
Paper No. HTFF 101
DOI: 10.11159/htff19.101

On a Grid Generation Method Based Upon Inhomogeneous Elliptic Partial Differential Equations

Kevin E. de Conde[1], Ezio C. Garcia[1], João Luiz F. de Azevedo[2]
[1]Instituto Tecnológico de Aeronáutica (ITA)
Praça Marechal-do-Ar Eduardo Gomes, 50 - Vila das Acácias, São José dos Campos, CEP 12228-900, São Paulo, Brazil
kevin@ita.br; ezio@ita.br
[2]Instituto de Aeronáutica e Espaço (IAE)
Praça Marechal-do-Ar Eduardo Gomes, 50 - Vila das Acácias, São José dos Campos, CEP 12228-904, São Paulo, Brazil
joaoluiz.azevedo@gmail.com

Abstract – A nearly-orthogonal boundary-conforming grid generation technique based upon inhomogeneous elliptic partial differential equations is presented. The method has been designed to generate grids for multiply-connected domains in a block-structure fashion. It also allows for control of point distribution along boundaries, as well as control of grid spacing inside the domain by using a slightly modified version of the renowned Thompson's control functions, which is reported to work well. Although not fully-automatic, the process is automatized in order to make it more usable and speed-up computational time, which is usually a demotivating factor in elliptic grid generation techniques. This is also the reason why the implementation has been performed in an objected-oriented fashion using C++ and as relaxation method the Alternating Direction Implicit (ADI) with α-sequence, generating appealing results. The method presented can be fully extended to 3D. Some results are presented and discussed.

***Keywords*:** Boundary-conforming coordinate systems, Inhomogeneous elliptic partial differential equations, Nearly-orthogonal grid.

1. Introduction

Grid generation can be understood as the process to obtain an adequate collection of points at which a system of equations is to be discretized and solved. In this context, an "optimum" point collection is desirable such that it minimizes the error in the solution. One possible way to obtain an optimized arrangement of points is by the use of boundary conforming coordinate systems, which is the reason why they have become very popular. Boundary-conforming coordinate systems are achieved by determining the values of the coordinates inside the physical region from the values and/or slopes of such coordinate lines on the boundaries. Classical approaches, such as algebraic interpolation and conformal mapping, are potential candidates, especially considering that such methods usually lead to great simplicity of implementation and low computational cost. Regarding algebraic interpolation methods, it has been pointed out that such methods may penalize applications for complex geometries, for example, collapsing lines of the same curvilinear coordinate [1]; "transfinite interpolation methods" have become popular as initial condition for more sophisticated generators, as reported in [2] and [3], among others. The foundations of algebraic interpolation methods in grid generation may be found in [4] and [5]. Conformal mapping has also been widely used as a grid generation technique in early aircraft design, and references may be found in [6]. As pointed by Ryskin and Leal [1], conformal mappings are ill-conditioned in the sense that very small changes in the shape of the domain can dramatically alter the position of mapped boundary points. On the other hand, in conjunction with a partial differential equation, such problem configures a boundary value problem, and obtaining the grid points as the solution of such a problem is a viable alternative [7]. In fact, the smoothness inherent in harmonic functions, and the uniqueness of the solution guaranteed by the maximum principles, tend to force that discontinuities and/or strong slopes present in the boundaries do not propagate inside the domain, what is desired for diminishing truncation errors in finite difference approximation for derivatives, as shown by Thompson et al. [8]. The authors still emphasize that, although the extremum principles may be weakened, or may even not be achieved by using more general elliptic systems (inhomogeneous systems), the conditions necessary for the extremum principles consist on a set of sufficient, but not necessary, conditions for a one-to-one mapping. Thus, as systems in the absence of control functions will originate grids that are not in general orthogonal to the boundaries, as well as with no control over grid spacing, it becomes more convenient to use inhomogeneous elliptic generators, such as

$$\nabla^2 \boldsymbol{\zeta} = \boldsymbol{Z}(\boldsymbol{\zeta}), \tag{1}$$

where $\boldsymbol{\zeta} = (\xi, \eta)^t$ and $\boldsymbol{Z} = (P, Q)^t$. Equation (1) corresponds to the following equation in the computational domain

$$a_{22}\boldsymbol{r}_{\xi\xi} - 2a_{12}\boldsymbol{r}_{\xi\eta} + a_{11}\boldsymbol{r}_{\eta\eta} + J^2\big(P\boldsymbol{r}_{\xi} + Q\boldsymbol{r}_{\eta}\big) = 0, \tag{2}$$

where $\boldsymbol{r} = (x, y)^t$. The coefficients a_{11}, a_{12} and a_{22} in Eq. (2) are called the covariant metric coefficients, and are defined by $a_{11} = \boldsymbol{r}_{\xi} \cdot \boldsymbol{r}_{\xi}$, $a_{12} = \boldsymbol{r}_{\xi} \cdot \boldsymbol{r}_{\eta}$, and $a_{12} = \boldsymbol{r}_{\eta} \cdot \boldsymbol{r}_{\eta}$; J is the Jacobian of the coordinate transformation, and naturally corresponds to $J^2 = a_{11}a_{22} - (a_{12})^2$. Such system was originally proposed by Thompson, Thames and Mastin [9], reason why it is known as TTM method. Finally, it is important to emphasize that, although some might consider grid generation a solved issue, this is not really true for some specialized applications [10]. An example is the case of multiply-connected domains with heterogeneous media; this is accentuated by fact that recent works have been addressing the subject, such as [10] and [11]. This is the primary reason for the interest in the present work. As pointed by Gordon and Hall [5], since mesh generation usually involves both geometry and physics, and "automatic" grid generation methods consider only geometric aspects, we refrain from calling "automatic" such methods since the actual final mesh will ultimately be resolved by the analyst who is cognizant of its specific application.

2. Methodology

2.1. Contour Orthogonality

In the solution of problems of interest in computational fluid dynamics, orthogonality or near-orthogonality next to boundaries are usually desirable, since according to [11], the accuracy on the discretization and boundary condition imposition is greatly improved. In case of a boundary corresponding to a wall, orthogonality and reduced grid spacing next to the wall are necessary in order to diminish the errors in the calculation of the large gradient of properties present in the boundary layer. Moreover, orthogonal or near-orthogonal grids allow the implementation of algebraic turbulence models to become more reliable, once the information normal to the boundary is usually needed in such models. According to [12], boundary orthogonality can be effected in two ways. In Dirichlet-Neumann orthogonality, control functions are not necessary. In this case, the points along the boundary are allowed to slide until boundary orthogonality is achieved, and the elliptic system has iterated to convergence. This approach is typically recommended for non-physical contours, such as far field boundaries, once it usually modifies the grid spacing near the boundary. In Dirichlet orthogonality, control functions are used to promote orthogonality next to the boundaries while the initial distribution of points along the boundary is kept constant. Zhang et al. [10] explain that, compared to Dirichlet orthogonality, Dirichlet-Neumann orthogonality is more difficult to use especially for cases with complex boundaries. According to [13], strictly boundary orthogonal grids, obtained from imposition of Dirichlet-Neumann orthogonality, can only be generated with partial control over the mesh spacing, while nearly-orthogonal grids allow for total control of the mesh spacing; these are obtained using Dirichlet orthogonality.

The iterative procedure necessary in elliptic grid generation methods require an initial grid to start with. Although algebraic generators are usually chosen for the initial grid, due to its low computational cost and ease of implementation, such methods may incur in grids with overlap of lines of the same curvilinear coordinate, which may not be problematic for elliptic generator of Laplace type, but may characterize a source of problems for an inhomogeneous generator since, as pointed by [14] and [15], in this case the range of initial conditions that would lead to convergence is substantially reduced. Parabolic grid present characteristics which combine smoothness and mesh generation capacity with all contours specified, inherent in elliptical generators, with the efficiency of hyperbolic generators. Thus, the method chosen for obtaining the initial grid is the parabolic grid generation technique proposed by Noack and Anderson [16], due to its advantages, such as smoothness and no grid folding, at an encouraging computational cost. On the other hand, an intrinsic characteristic of the initial parabolic grids is not allowing for control of coordinate spacing in all directions. As a consequence, the method proposed by Khaymaseh and Mastin [17] for calculating the orthogonal functions during the iterative process had to be modified, once it keeps the line spacing of the initial grid. The full technique will be presented in a forthcoming article.

2.2. Relaxation Procedure, Implementation and Internal Functions

Computational cost for elliptic grid generation is a decisive issue that has been also reported by [10] and [13]. In the present work, in order to address computational cost issues, the following aspects have been emphasized: 1) The implementation has been performed in an object-oriented fashion using C++; 2) The Alternating Direction Implicit (ADI) method, proposed in [18] and [19], has been used as relaxation method; and 3) α-sequence, as described in [20], have been used in order to accelerate convergence. Moreover, the internal function calculation process is somewhat automatized in order to make it more usable and speed-up computational time. A slightly modified version of the renowned Thompson's control functions is used [19]. Furthermore, the method presented can be fully extended to 3-D, in which case the computational costs become even more important. About the use of object-oriented programming, the standard approach (which is well documented) has been used, and no further observations are necessary. With regard to ADI method, it has been implemented as proposed in the cited series of articles by Ballhaus et al. [18], [19]. Complementary information, such as a Von-Neumann stability analysis, a discussion regarding the use of the α-sequence, as well as relevant usages may be found in [20] and [21]. The internal function calculation process is addressed in the following section.

2.3. Internal Functions Calculation

As pointed by [12], refined grids may be necessary to solve with precision the large gradients present in boundary layers and, although the elliptic generator system with orthogonality on boundaries provides for good characteristics in terms of smoothness and orthogonality, it may be still necessary to control the grid spacing. Thompson et al. [22] proposed exponential control functions that have been reported to provide efficient control over the grid spacing. As argued by Thompson et al. [8], the exponential control functions (the internal functions) may have to present absolute values significantly large to produce expressive effects on the grid. Larger amplitudes of the control functions may impact negatively on convergence. This is explained in [14], which also explains that large amplitudes may restrict possible initial conditions that lead to convergence. The internal control functions used here are a slightly modified version of those proposed in [8]. The basic idea is to work around the problem at hand by "normalizing" the author's functions. To understand the methodology, consider the following inhomogeneous elliptic system which, similarly to Eq. (1), has also been widely used for grid generation [15],

$$\begin{aligned} \nabla^2 \xi &= \frac{a_{22}}{J^2}\, P(\xi,\eta); \\ \nabla^2 \eta &= \frac{a_{11}}{J^2}\, Q(\xi,\eta). \end{aligned} \tag{3}$$

According to [8], the control functions for this system are allowed to assume values orders of magnitude smaller than the ones in system Eq. (1) for similar effects. Comparing Eq. (1) with Eq. (3) makes it obvious that multiplying the functions P and Q in Eq. (1) respectively by a_{22}/J^2 and $a_{11} = J^2$ allows the coefficients inputted by the user (the decay parameters) to be the same as if the system Eq. (3) was being used. The only drawback is that the computational cost for calculating the internal control functions is slightly increased. Thus, the functions used here are of the form

$$\boldsymbol{Z_I}(\xi,\eta) = -\boldsymbol{A}\left\{\sum_{i=1}^{n} \mathbf{a}_i \operatorname{sgn}(\boldsymbol{\zeta}-\boldsymbol{\zeta}_i)\exp(-\mathbf{c}_i|\boldsymbol{\zeta}-\boldsymbol{\zeta}_i|) + \sum_{i=1}^{n} \mathbf{b}_j \operatorname{sgn}(\boldsymbol{\zeta}-\boldsymbol{\zeta}_j)\exp\left[-\mathbf{d}_j\sqrt{(\xi-\xi_j)^2+(\eta-\eta_j)^2}\right]\right\}, \tag{4}$$

where $\boldsymbol{A} = (a_{22}/J^2, a_{11}/J^2)$. The decay parameter inputted by the user are $\mathbf{a}_i = (\mathrm{a}_i^1, \mathrm{a}_i^2)^t$, $\mathbf{b}_j = (\mathrm{b}_j^1, b_j^2)^t$, $\mathbf{c}_i = (\mathrm{c}_i^1, c_i^2)^t$ and $\mathbf{d}_j = (\mathrm{d}_j^1, d_j^2)^t$. Additionally,

$$\operatorname{sgn}(x)=\begin{cases}1, & \text{if } x>0;\\ 0, & \text{if } x=0;\\ -1, & \text{if } x=0.\end{cases} \tag{5}$$

It should be noted that the modification does not change the way $\boldsymbol{Z_I}$ works and how it affects the grid, which is well described in [22]. Thompson et al. [2] provide for a particularly good description of how to use the internal control functions to control the grid, which is summarized here. For P_I, the first term in Eq. (4) attracts ξ-lines to the ones specified by the user, ξ_i, with amplitudes determined by a_i^1. The effect of the term containing b_j^1 is to attract ξ-lines to the point (ξ,η), inputted by the analyst; the amplitudes are determined by the coefficients b_j^1, also selected by the analyst. It should be noted that the attraction to a point consists of attraction to the line (in this case, the ξ_i-line) where the point lies on. As such, the gravitation occurs perpendicularly to the refereed line on the point. There is no attraction to an η-line through P_I. In each case, the extension of the region affected by the gravitation is determined by the decay factors c_i^1 and d_j^1 , respectively. sgn is necessary so that the pulling effect occurs in both sides of the respective line or point. If sgn is not used, the gravitating effect occurs on the side of increasing ξ, while the exact opposite occurs in the other side. Additionally, this function causes P_I and Q_I to be discontinuous functions. Consequently, in regions under the effects of $\boldsymbol{Z_I}$, coordinate lines and their first derivatives are continuous, but second-order derivatives are not necessarily continuous. Besides, decay parameters with negative values occasion the reversion of effects, that is, attraction becomes repulsion and vice-versa. The effect of Q_I to η-lines is analogous. It should also be mentioned that P_I and Q_I from Eq. (4) are also compatible with the methodology proposed by [7].

The methodology proposed by Fletcher (1996) for imposing the internal control functions after the system has converged with $\boldsymbol{Z_I}=0$ has been automated as follows. The user is required to input the number of times the system will be iterated until it reaches convergence, called EI for external iterations. For each external iteration, $\boldsymbol{Z_I}$ have its decay parameters $\mathbf{a}_i$ and $\mathbf{b}_j$ multiplied by $(ne-\ 1)/EI$ (ne is the current external iteration number), and the system with the current $\boldsymbol{Z_I}$ is iterated until convergence is obtained; the converged solution is used as initial condition for the next external iteration. The process is repeated from $ne\ =\ 1$ until $ne\ =\ EI+1$, in order that for the first external iteration, the system is iterated with $\boldsymbol{Z_I}=0$, and for the last external iteration $EI\ +\ 1$, the system is iterated with the coefficients for $\boldsymbol{Z_I}$ inputted by the user, necessary to obtain the final grouping of points and/or lines. Finally, the merge of internal and orthogonal control functions will be presented in a forthcoming article, with the full methodology.

3. Results

The grids are evaluated based upon two standard academic criteria, namely, orthogonality and smoothness, represented respectively by the following indicators: deviation from orthogonality and grid aspect ratio, as proposed in [10]. Figure 1 presents the grid obtained with the present method for a geometry corresponding to fins in a multimode heat transfer calculation problem. The domain is defined by $0.4-0.2[1-\ 4.5y(1.5-\ y)]\leq\ x\leq\ 0.6+0.2[1-4.5y(1.5-y)]$ and $0\leq y\leq 0.46+\ x(1.0-\ x)$. Uniform distribution of boundary points was adopted. For the present case, the grid size is $500\times 50\ cells$, with a maximum aspect ratio of 7.1 and maximum deviation from orthogonality of 48 deg; in this case, average deviation from orthogonality is of 28 deg, and average aspect ratio is of 2.0. Figure 2 illustrates orthogonality by means of color field representations for the mesh of Fig. 1. Grid lines are omitted in favor of clarity. As may be seen in Fig. 2, deviation from orthogonality at the boundaries is minimum, except close to the right and left bottom corner regions, which is delicate due to the geometry. Naturally, deviation in such regions is expected due to the conformability requirement. In such regions, the internal functions have been used combined with the smoothing functions to avoid the strong spreading tendency reported for example by [10]. Although the mesh quality is lower in such regions compared to the rest of the domain, this does not raise concern, since for the current application, it corresponds to a region symmetry with low gradients of properties expected. Figure 3 illustrates aspect ratio by means of color field representations for the mesh of Fig. 1. As expected, boundary curvature differences are not present in grids obtained with the current method, this is, the boundaries are perfectly fitted. It should be emphasized that the parabolic initial grid presented no grid folding.

In all the cases presented, the convergence criterion adopted is that the L^{∞}-norm of the difference between current and past iterations values of the coordinates in the physical space is smaller than 10^{-11}, that is $\left||\boldsymbol{r}^{n}-\boldsymbol{r}^{n-1}|\right|_{\infty} \leq 10^{-11}$.

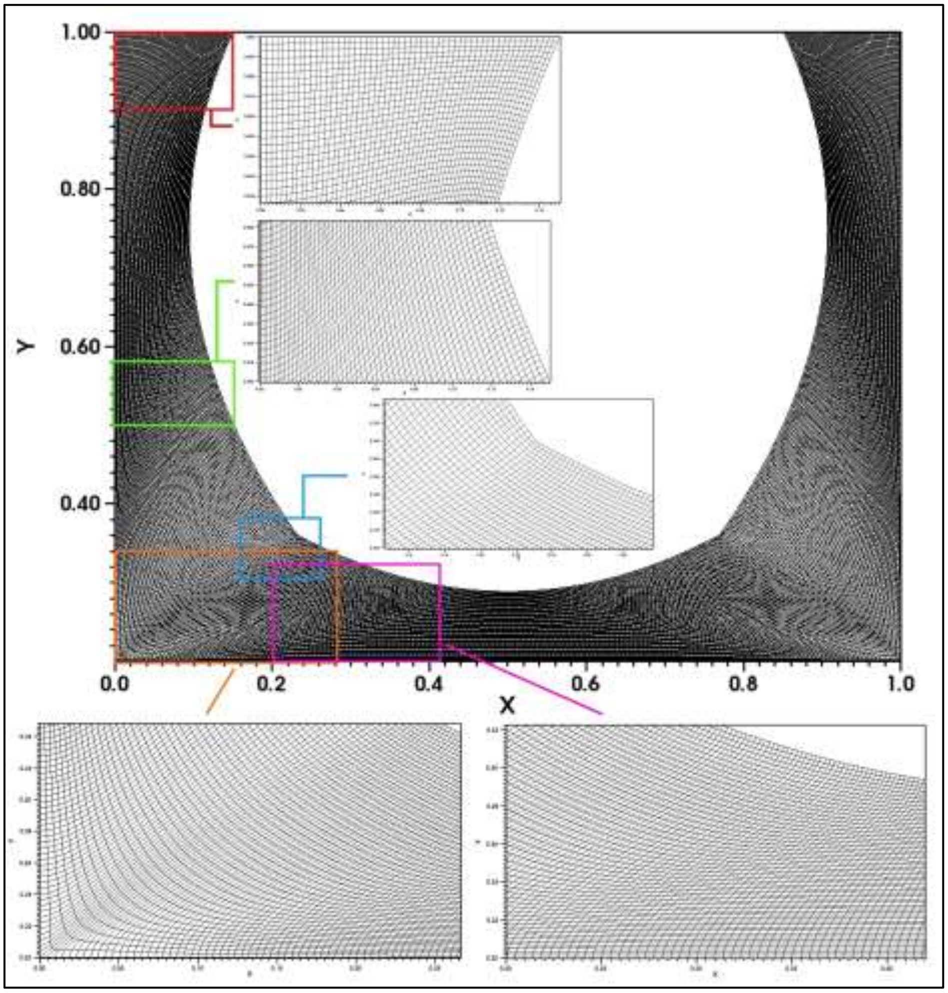

Fig. 1: Nearly orthogonal elliptic grid for fins.

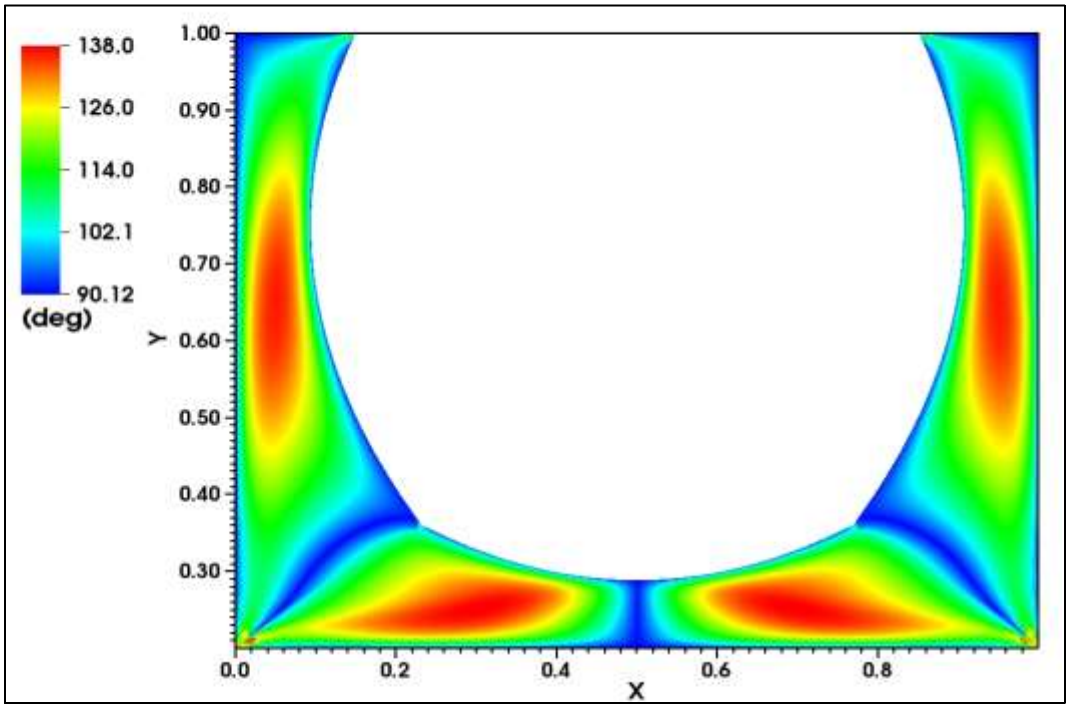

Fig. 2: Color field representation of orthogonality for grid of Fig. 1.

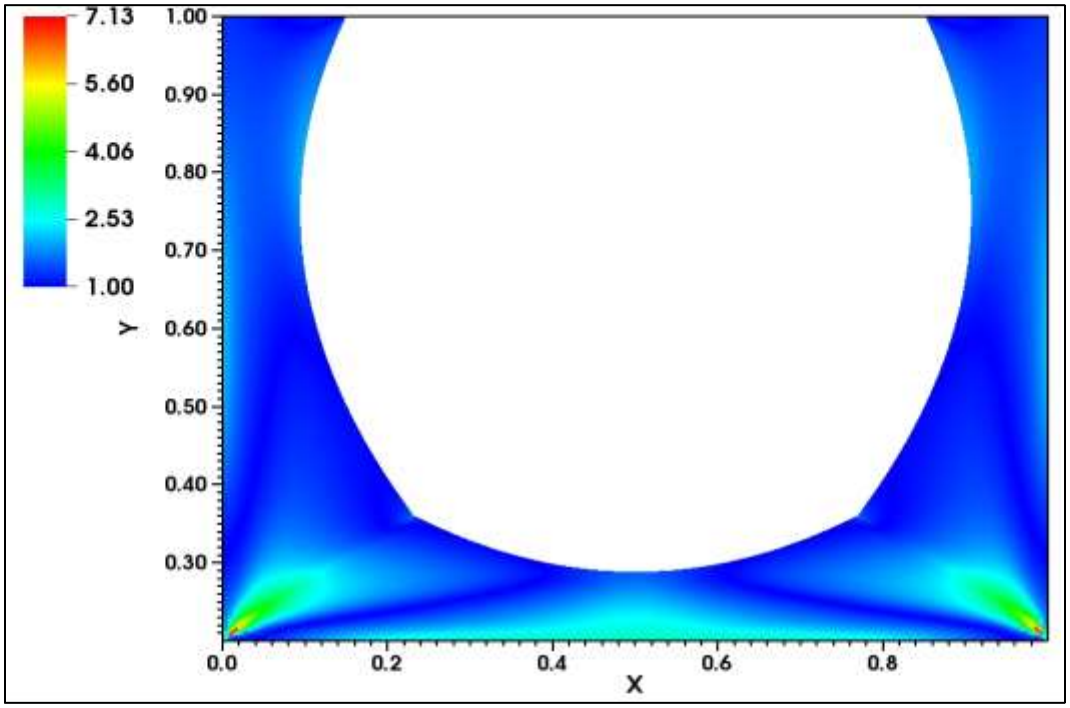

Fig. 3: Color field representation of aspect ratio for grid of Fig. 1.

4. Conclusions

In CFD analysis and mesh generation, complex geometries remain a challenge. In this study, that is the mainly concern. The mapping problem regarded as the most difficult in the literature, corresponding to the construction of a coordinate system fitted to a boundary of given shape, with a prescribed distribution of coordinate nodes along this boundary is addressed. The method proposed consists of an inhomogeneous elliptic system designed envisioning flexibility and reliability. The mapping technique proposed provide a method for the construction of a boundary-fitted coordinate system, nearly-orthogonal to the boundaries, in two dimensions; the method can be extended to 3D. The inhomogeneous terms are used in two ways: first, to obtain the near-orthogonality along the boundaries and two, to allow grid clustering inside the domain. In the last case, a simple modification on Thompson's control functions confirmed the expectations, simplifying the choice of the decay parameters by the user. As a drawback, the meshes obtained are not in general orthogonal over the entire domain, which may be desired in some applications. On the other hand, the flexibility obtained with the method is certainly an interesting asset, especially for complex geometries. Also, grids with Dirichlet boundary conditions are obtained, allowing the boundary point distribution to be controlled by the analyst. As commented, inhomogeneous elliptic generation systems are sensible to initial conditions, reason why a parabolic generation method has been chosen, which has proved to work fairly well. With regard to computational cost, the Alternating Direction Implicit scheme, with the appropriate α-sequence, has shown to be an interesting choice, in special given the necessity of internal control functions for grid clustering, and mostly in case of a 3D application. Although not fully-automatic, the method has the advantage of allowing the analyst, who is cognizant with the specific application, to use its best judgment in order to obtain an optimized grid to the problem. The results have shown that boundary-nearly-orthogonal grids, with control over point distribution along boundaries, as well as control of point and line spacing inside the domain, can be obtained with the method proposed at a reasonable computational time.

Acknowledgements

The authors gratefully acknowledge the support for the present research provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Research Grants No. 309985/2013-7 and No. 305269/2016-0. The work is also supported by Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under the Research Grant No. 2013/07375-0. The support provided by Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, through a doctoral scholarship under the Research Grant No. 88882.180821/2018-01 and through a sandwich doctoral scholarship under the Research Grant No. 88881.131976/2016-01 for the first author, is also greatly appreciated.

Nomenclature

A	Diagonal matrix with non-zero terms a_{22}/J^2 and a_{11}/J^2
a_{ij}	Covariant metric tensor; components of the covariant metric tensor
$\mathbf{a}_i$	Decay parameter for the internal control functions
$\mathbf{b}_j$	Decay parameter for the internal control functions
$\mathbf{c}_i$	Decay parameter for the internal control functions
d_j	Decay parameter for the internal control functions
i, j	Used as subscripts for the discrete coordinates ξ and η respectively
J	Jacobian of the coordinate transformation
n	Used as superscript to denote current iteration
P	Control function P
Q	Control function Q
x, y	Cartesian coordinates
$\boldsymbol{Z} = (P, Q)^t$	Vector of control functions

ξ, η	General curvilinear coordinates of the computational domain
$\boldsymbol{\zeta} = (\xi, \eta)^t$	Vector of coordinates in the computational domain

References

[1] G. Ryskin and L. G. Leal, “Orthogonal mapping,” *Journal of Computational Physics*, vol. 50, no. 1, pp. 71-100, 1983.

[2] J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, “Boundary-fitted coordinate system for numerical solution of partial differential equations - a review,” *Journal of Computational Physics*, vol. 47, no. 1, pp. 1–108, 1982.

[3] L. Eça, “2D orthogonal grid generation with boundary point distribution control,” *Journal of Computational Physics*, vol. 125, no. 2, pp. 440–453, 1996.

[4] W. J. Gordon, “Blending-function methods of bivariate and multivariate interpolation and approximation,” *SIAM Journal on Numerical Analysis*, vol. 8, no. 1, pp. 158–177, 1971.

[5] W. J. Gordon. and C. A. Hall, “Construction of curvilinear coordinate systems and applications to mesh generation,” *International Journal for Numerical Methods in Engineering*, vol. 7, no. 4, pp. 461–477, 1973.

[6] B. Fornberg, “A numerical method for conformal mapping,” *SIAM Journal on Scientific and Statistical Computing*, vol. 1, no. 3, pp. 386–400, 1980.

[7] J. L. Steger and R. L. Sorenson, “Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations,” *Journal of Computational Physics*, vol. 33, no. 3, pp. 405–410, 1979.

[8] J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, *Numerical Grid Generation: Foundations and Applications*. Elsevier-Science Publishing Co., New York, 1st edition, 1985.

[9] J. F. Thompson, F. C. Thames and C. W. Mastin, “Automatic numerical generation of body-fitted coordinate system for field containing any number of arbitrary two-dimensional bodies,” *Journal of Computational Physics*, vol. 15, no. 3, pp. 299–319, 1974.

[10] Y. Zhang, J. Yafei, S. S. Y. Wang and H. C. Chan, “Boundary treatment for 2D elliptic mesh generation in complex geometries,” *Journal of Computational Physics*, vol. 227, no. 16, pp. 7977–7997, 2008.

[11] M. Wiesenberger, M. Held and L. Einkemmer, “Streamline integration as a method for two-dimensional elliptic grid generation,” *Journal of Computational Physics*, vol. 340, no. 1, pp. 435–450, 2017.

[12] A. Khaymaseh, A. Kuprat and C. W. Mastin, “Boundary orthogonality in elliptic grid generation,” in J. F. Thompson, B. K. Soni and N. P. Weatherill, eds., *Handbook of Grid Generation*. CRC Press, Boca Raton, chap. 6, pp. 198–223, 1999.

[13] U. K. Kaul, “New boundary constraints for elliptic systems used in grid generation problems,” *Journal of Computational Physics*, vol. 189, no. 2, pp. 476–492, 2003.

[14] C. A. J. Fletcher, “Computational Techniques for Fluid Dynamics 1: Fundamental and General Techniques,” Vol. 1 of Springer series in *Computational Physics*. Springer-Verlag, New York, 2nd edition, 1996.

[15] S. P. Spekreijse, “Elliptic generation systems,” In J. F. Thompson, B. K. Soni and N. P. Weatherill, eds., *Handbook of Grid Generation*. CRC Press, Boca Raton, chap. 4, pp. 123–171, 1999.

[16] R. W. Noack and D. A. Anderson, “Solution adaptive grid generation using parabolic partial differential equations,” *AIAA Journal*, vol. 28, no. 6, pp. 1016–1023, 1990.

[17] A. Khaymaseh and C. W. Mastin, “Computational conformal mapping for surface grid generation,” *Journal of Computational Physics*, vol. 123, no. 2, pp. 394–401, 1996.

[18] W. F. Ballhaus, A. Jameson and J. Albert, “Implicit approximate-factorization schemes for the efficient solution of steady transonic flow problems,” *AIAA Journal*, vol. 16, no. 6, pp. 573–580, 1977.

[19] W. F. Ballhaus, A. Jameson and J. Albert, “Implicit approximate-factorization schemes for the efficient solution of steady transonic flow problems,” *Technical Report, National Aeronautics and Space Administration (NASA)*, Washington DC, 1977.

[20] T. Holst, “On approximate factorization schemes for solving the full potential equation,” in *Proceedings of the Advances in Computational Transonic Conference*. Pineridge Press, vol. 2, 1985, pp. 59–82.

[21] T. Holst, “On approximate-factorization schemes for solving the full potential equation,” *Technical report, National Aeronautics and Space Administration (NASA)*, Washington DC, 1997.

[22] J. F. Thompson, F. C. Thames and C. W. Mastin, “Boundary-fitted curvilinear coordinate systems for solution of partial differential equations on field containing any number of abitrary two-dimensional bodies,” *Technical report, National Aeronautics and Space Administration (NASA)*, Washington DC, 1977.