6

Alternative Methods for Generating Elliptic Grids in Finite Volume Applications

A. Ashrafizadeh, M. Ebrahim and R. Jalalabadi
K. N. Toosi University of Technology
Iran

1. Introduction

Numerical solution of an engineering problem via finite volume method (FVM) requires the discretization of the solution domain and computational grid generation. While both structured and unstructured grids can be used, elliptic structured grid generation methods, when applicable, have favorable features in terms of both accuracy and computational cost.

Among the elliptic grid generation (EGG) methods, the most well known and widely used are the algebraic transfinite interpolation and differential methods which employ Poisson equations. In this chapter classical EGG methods are reviewed. It is then proposed that these methods can be classified based on the parameters being interpolated (i.e. interpolants), the interpolation method used and the grid generation equations being employed. The proposed unified view provides a framework for the development of new grid generation methods; some of which are introduced here for the first time.

Another major task in this chapter is to show that finite volume method, which employs the computational grid, can itself be used in the numerical grid generation process. In other words, FVM can be used for two different tasks; discretization of the differential equations which govern the coordinates of the computational grid points and discretization of the differential equations which govern the physical process of interest.

A typical 2D structured grid in the physical domain is shown in Fig. 1a and the corresponding logical or computational grid is shown in Fig. 1b. The classical structured grid generation methods provide equations which define, directly or indirectly, the mapping functions which describe the curvilinear coordinate lines in the physical domain, i.e. $\xi(x,y)$ and $\eta(x,y)$ curves. The grid point (i,j) in the physical domain is defined at the intersection of the curvilinear coordinate lines ξ_i and η_j as shown in Fig. 1a.

The so called algebraic grid generation methods directly specify the formulas used to calculate the physical coordinates (x,y) in terms of the logical coordinates (ξ,η) (Eiseman, 1979; Eiseman et al., 1992; Lehitmaki, 2000; Zhou, 1998). For example, in the Trans-Finite Interpolation (TFI) method (Eiseman et al., 1992), the generating equations employ the boundary nodal coordinates and some derivative terms to calculate the nodal coordinates throughout the solution domain. This method is often described as a Boolean sum of one dimensional interpolation functions U and V as follows:

$$\vec{R}(\xi,\eta) = (x,y) = U \oplus V = U + V - UV \tag{1}$$

$$U(\xi,\eta) = \sum_{i=1}^{L}\sum_{n=0}^{P} \alpha_i^n(\xi)\frac{\partial^n \vec{R}(\xi_i,\eta)}{\partial \xi^n} \tag{2}$$

$$V(\xi,\eta) = \sum_{j=1}^{M}\sum_{m=0}^{Q} \beta_j^m(\eta)\frac{\partial^m \vec{R}(\xi,\eta_j)}{\partial \eta^m} \tag{3}$$

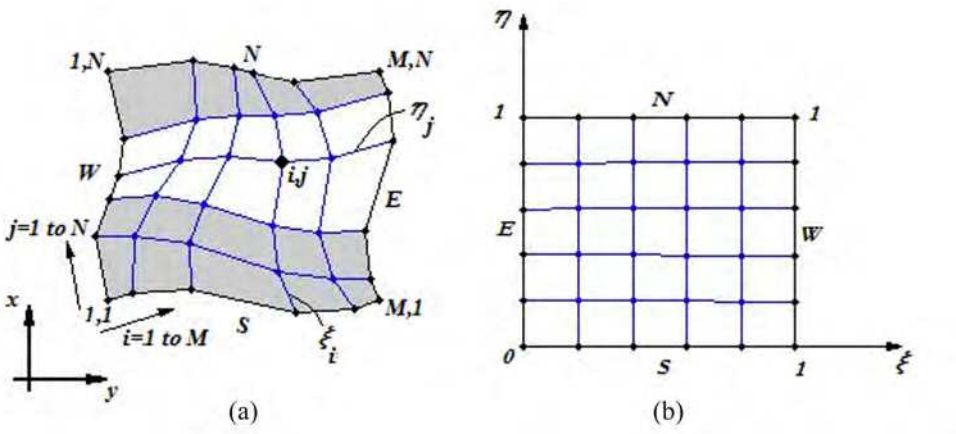


Fig. 1. (a) A physical grid, (b) the corresponding logical grid.

The highest orders of one-dimensional interpolation formulas in Eqs. (2) and (3) are specified by P and Q respectively, and L and M specify the number of auxiliary nodes used in these interpolations. For example, a zero order TFI computational molecule for generating a (M×N) grid is as follows:

$$\begin{aligned}\vec{R}_{i,j} = {} & C_{i,N}\vec{R}_{i,N} + C_{i,1}\vec{R}_{i,1} + C_{1,j}\vec{R}_{1,j} + C_{M,j}\vec{R}_{M,j} + \\ & C_{1,N}\vec{R}_{1,N} + C_{M,N}\vec{R}_{M,N} + C_{1,1}\vec{R}_{1,1} + C_{M,1}\vec{R}_{M,1}\end{aligned} \tag{4}$$

Coefficients $C_{i,N}$, $C_{i,1}$, …… in this nine-point computational molecule for the calculation of the coordinates of the nodal point (i,j) can be linear or nonlinear functions of the logical coordinates of this point, i.e. ξ_i, η_j.

In contrast to the TFI, the mapping functions are not explicitly provided in the so called differential grid generators. For example, the following differential constraints on the unknown mapping functions $\xi(x,y)$ and $\eta(x,y)$ are proposed by Thompson, Thames and Mastin (TTM) (Thompson, et al., 1974):

$$\xi_{xx} + \xi_{yy} = P(\xi,\eta) \tag{5}$$

$$\eta_{xx} + \eta_{yy} = Q(\xi,\eta) \tag{6}$$

Equations (5) and (6) are often analytically inverted and the calculations are carried out in the logical domain. The non-linear inverted equations are as follows:

$$g_{11}\, x_{\xi\xi} - 2\, g_{12}\, x_{\xi\eta} + g_{22}\, x_{\eta\eta} = -\, J^2 (P\, x_{\xi} + Q\, x_{\eta}) \tag{7}$$

$$g_{11}\, y_{\xi\xi} - 2\, g_{12}\, y_{\xi\eta} + g_{22}\, y_{\eta\eta} = -\, J^2 (P\, y_{\xi} + Q\, y_{\eta}) \tag{8}$$

In Eqs. (7) and (8), $g_{11} = x_n^2 + y_n^2$, $g_{22} = x_{\xi}^2 + y_{\xi}^2$, $g_{12} = x_{\xi}\, x_{\eta} + y_{\xi}\, y_{\eta}$ and J is the Jacobian of the transformation ($J = x_{\xi}\, y_{\eta} - y_{\xi}\, x_{\eta}$).

Ashrafizadeh and Raithby (Ashrafizadeh & Raithby, 2006) have shown that the TTM grid generation equations, i.e. Eqs. (5) and (6), can be discretized and solved in the physical domain. To apply the finite volume method to the solution of Eqs. (5) and (6) in the physical domain, an initial algebraic grid is generated first. Then, a control volume is associated with each node of the initial grid. Defining

$$\vec{q}^{\,\xi} = \vec{\nabla}\xi \; ; \qquad \vec{q}^{\,\eta} = \vec{\nabla}\eta \tag{9}$$

the integral of Eq. (5) over a control volume associated with node *i*, with volume V_i and surface S_i , is

$$\int_{V_i} \vec{\nabla} \bullet \vec{q}^{\,\xi}\, dV = \int_{S_i} \bullet\, d\vec{S} = \int_{V_i} P\, dV \tag{10}$$

The surface S_i consists of a number of panels, with an integration point *ip* located at the centre of each panel. The panel containing *ip* has area $\vec{S}_{ip}$. The integrals in Eq. (10) are approximated as follows

$$\sum_{ip} \vec{q}_{ip}^{\,\xi} \bullet \vec{S}_{ip} \equiv \sum_{ip} F_{ip}^{\xi} = (PV)_i \tag{11}$$

where F_{ip}^{ξ} can be thought of as a generalized "flow" across the panel *ip* driven by $\vec{\nabla}\xi$.

The final algebraic equation is obtained by approximating each term in Eq. (11) by an equation that involves nodal values of ξ , x , and y . This provides one constraint for ξ_i , x_i , and y_i . Applying a similar procedure, Eq. (6) leads to another algebraic equation relating η_i , x_i , and y_i for each interior node. But the values of ξ_i and η_i are known for all interior nodes, so that these two algebraic equations provide the necessary constraints for computing (x_i , y_i). The nodal values of ξ_i , η_i x_i , and y_i are all prescribed for boundary nodes, so the set of equations is closed throughout the solution domain and its boundary. This is called the Direct Design Method for solving the elliptic grid generation problem because no inversion of equations is required and the unknown nodal values of (x_i , y_i) appear explicitly (i.e. "directly") as dependent variables in the finite volume equations in the physical domain.

Calculation of the source or control functions at the right hand sides of Eqs. (5) and (6) is an important part of any method which uses this set of equations to generate the grid. In addition to the elementary method, proposed in (Thompson, et al., 1974), many researchers

have proposed methods for the automatic calculation of the boundary values of control functions (Thomas & Middlecoff, 1980; Spekreijse, 1995; Steger & Sorenson, 1997; Kaul, 2003; Lee & Soni, 2004; Ashrafizadeh & Raithby, 2006; Kaul, 2010). Assuming that the P values are known at $(\xi_i, \eta = 1)$ and $(\xi_i, \eta = 0)$ boundaries in Fig. 1a, the P values at internal nodes can be obtained through the following one dimensional interpolation formula (Thomas & Middlecoff, 1980):

$$P(\xi,\eta) = C(\eta)\ P(\xi,0) + \left(1 - C(\eta)\right)\ P(\xi,1) \tag{12}$$

The Q values at internal nodes can also be calculated similarly:

$$Q(\xi,\eta) = C(\xi)\ Q(0,\eta) + \left(1 - C(\xi)\right)\ Q(1,\eta) \tag{13}$$

Coefficients $C(\xi)$ and $C(\eta)$ in Eqs. (12) and (13) can be linear or non-linear functions of the corresponding logical coordinates.

Another noticeable classical Grid Generation method, which is known as the Orthogonal Grid Generation (OGG) method and is elliptic in certain situations, is based on the assumptions of continuity and orthogonality of the coordinate lines. The final forms of the grid generation equations in the OGG method are as follows (Ryskin & Leal, 1983):

$$\frac{\partial}{\partial \xi}\left(f \frac{\partial x}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{f} \frac{\partial x}{\partial \eta}\right) = 0 \tag{14}$$

$$\frac{\partial}{\partial \xi}\left(f \frac{\partial y}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{f} \frac{\partial y}{\partial \eta}\right) = 0 \tag{15}$$

The orthogonality condition, $g_{12} = x_\xi x_\eta + y_\xi y_\eta = 0$, is implied in Eqs. (14) and (15). The scale factor, f , is defined based on the transformation metrics relevant to the magnification effects of the mapping in different logical directions as follows:

$$f \equiv \frac{\sqrt{g_{22}}}{\sqrt{g_{11}}} = \frac{\sqrt{\left(x_\eta^2 + y_\eta^2\right)}}{\sqrt{\left(x_\xi^2 + y_\xi^2\right)}} \tag{16}$$

Calculation of the scale factor near the boundaries and throughout the solution domain is a major step in the orthogonal grid generation and is discussed in a number of publications (Ryskin & Leal, 1983; Kang & Leal, 1992; Eca, 1996; A. Bourchetin & L. Bourchetin, 2006). Most commonly, boundary values of f are calculated first and then linear or non-linear interpolation techniques are used to obtain the internal values.

Imposition of the orthogonality constraint in some problems may be difficult or even impossible. Therefore, modifications on the OGG have also been proposed to generate nearly orthogonal grids (Akcelik et al., 2001; Zhang et al., 2004, Zhang et. Al, 2006a, Zhang et. Al, 2006b).

Based on the above brief review of some of the classical EGG methods, it can be argued that in each one of these methods a set of grid generation equations is developed to calculate the

physical coordinates of the nodal points. The set of equations may directly introduce the mapping functions which transform the logical grid to the physical grid, e. g. the TFI, or they may provide differential constraints on the mapping functions and indirectly describe them, e. g. the TTM and the OGG methods. However, it is important to note that the governing equations in grid generation are radically different from equations which govern physical processes. While experimental observations provide a basis for the development of the physical governing equations, the grid generation equations are not expressions of natural phenomena and are developed based on analogy or mathematical considerations. An example of the use of physical analogy in determining the control functions in Eq. (5) and (6) is provided by Kaul (Kaul, 2003). Considering the arbitrariness in the development of elliptic grid generation equations, it is very desirable to have a clear, simple and systematic approach for proposing the governing equations in the context of structured grid generation.

In this paper, we propose a unifying rationale for the development of elliptic grid generation methods. Based on the proposed unifying view point, all existing EGG methods can be viewed as multi-dimensional geometrical interpolation techniques which employ different interpolants, interpolation methods and grid generation equations. Once the grid generation equations, the interpolants and interpolation techniques are chosen, there are various numerical solution methods to solve the algebraic equations and to calculate the nodal coordinates.

To explain the proposed framework, a number of applicable interpolants in structured grid generation are first introduced in the next section. Then, different applicable interpolation techniques are presented. Afterwards, the rationale behind the development of grid generation equations is discussed. Finally, a number of alternative EGG methods are introduced and examples of elliptic grid generation via the classical and proposed alternative methods are presented.

For the sake of simplicity and brevity, the grid generation examples in this chapter are limited to two dimensional solution domains, but the underlying ideas are clearly applicable in three-dimensional problems as well.

2. Interpolants

The logical grid, shown in Fig. 1b, is the simplest possible two-dimensional grid. The boundaries are straight lines and the nodes are distributed uniformly. This simplicity makes the logical grid generation trivial. In contrast, boundaries of the physical grid may be complex curves and the boundary nodes in this case can be distributed non-uniformly. For example, the N, S, W and E boundaries shown in Fig. 1a are different curves with different non-uniform distributions of nodes. It is exactly this complexity that makes the grid generation in the physical domain a rather difficult task as compared to the logical domain. The shape of the boundary of the domain and the distribution of the nodes along the boundary are the most important information which needs to be taken into consideration in the elliptic grid generation process. Quantities which provide information regarding the shape of the boundary coordinate lines, expected shape of the crossing coordinate lines near the boundaries and the distribution of nodes along the boundaries are here called the boundary data. Some or all of these data are the inputs required in an elliptic grid generation method.

The interpolants in an EGG problem are quantities related to the boundary data. The simplest, and most obvious, quantities which describe the boundary coordinate lines are the x and y coordinates of boundary nodal points. These are the interpolants used in many simple algebraic grid generators and we call them here the zero order boundary data.

It is possible to use higher order boundary data as interpolants as well. Quantities such as nodal values of x_ξ and y_ξ along the north boundary in Fig. 1a are tangential slope-related first order data. Similarly, $x_{\xi\xi}$ and $y_{\xi\xi}$ along the south boundary are tangential curvature-related second order data. These data provide information regarding the stretching of nodes along a boundary coordinate line.

By generating paving layers near boundaries or by making assumptions regarding the coordinate lines which cross the boundaries, it is also possible to generate normal first, second,……, and nth order boundary data. For example, using the shaded paving layer near the north boundary in Fig. 1a, it is possible to generate x_η and y_η data at the north boundary. Similarly, $x_{\eta\eta}$ and $y_{\eta\eta}$ can be generated near the south boundary using the data obtained from the two shaded paving layers near the south boundary in Fig. 1a. Therefore, using the paving layers, it is possible to generate boundary data which actually describe the boundary cell geometries or the shape of coordinate lines crossing the boundary. A simple algebraic method for generating high quality paving layers is introduced in (Ashrafizadeh & Raithby, 2006).

The first and second order boundary data can also be defined with the physical coordinates as the independent variables. For example quantities such as ξ_x, ξ_{xx}, ξ_y and ξ_{yy} fall in this category. However, in contrast to the logical coordinates, the physical coordinates of nodes are not univariate variables and the denominator in the discrete form of a quantity such as ξ_x can be zero, resulting in computational difficulties.

Boundary data of different orders, just described, can also be combined to provide more information regarding the boundary nodes and cells. Such combinations of the boundary data can be employed as the interpolants in the formulation of EGG problems. For example, in the TTM method, boundary values of $(\xi_{xx}+\xi_{yy})$, called P, and $(\eta_{xx}+\eta_{yy})$, called Q, are used as the interpolants. Figure 2 shows how the source term P provides information regarding the shape of the cells and the distribution of the nodes along the boundary coordinate line (ξ_i, η_N). In Fig. 2a, the boundary is a straight line and the nodes are distributed uniformly. The source term P is identically zero everywhere along the boundary in this case. Figure 2b shows a straight boundary with non-uniform distribution of nodes. It is seen that the source term P is not zero anymore at locations with contraction or expansion of the grid. Figures 2c and 2d show curved boundaries with uniform and non-uniform distribution of nodes respectively. It is clear that the source term P varies along the η_N coordinate line and carries some information regarding the boundary geometry in the latter two cases as well. Therefore, the source terms P and Q can be used as interpolants in an elliptic grid generation problem. The source values in these examples, which are Laplacians of the logical coordinates, have been calculated using the finite volume method as described in (Ashrafizadeh et al., 2002; Ashrafizadeh et al., 2003).

The scale factor f is used as the interpolant in the classical OGG method. The shape of the boundary and the distribution of nodes in Figs. 3a to 3d have been chosen similar to Figs. 2a to 2d respectively to study the effect of the boundary geometry on the boundary values of

f .For the straight uniform paving layer shown in Fig. 3a, $f = 1$ everywhere along the boundary. In all other cases in Fig. 3, the curved boundary and/or non-uniform distribution of nodes result in a corresponding change in the nodal f values. Therefore, the scale factor f can also be used as the interpolant in an elliptic grid generation problem.

Other combinations of the boundary data can also be used as the interpolants in an EGG problem and one can check the sensitivity of a chosen interpolant to the boundary specifications before actually using them in an elliptic grid generation algorithm as just explained. Since these computations are done on distorted and/or non-uniform grids in the physical domain, finite volume method is a suitable numerical solution choice as explained before.

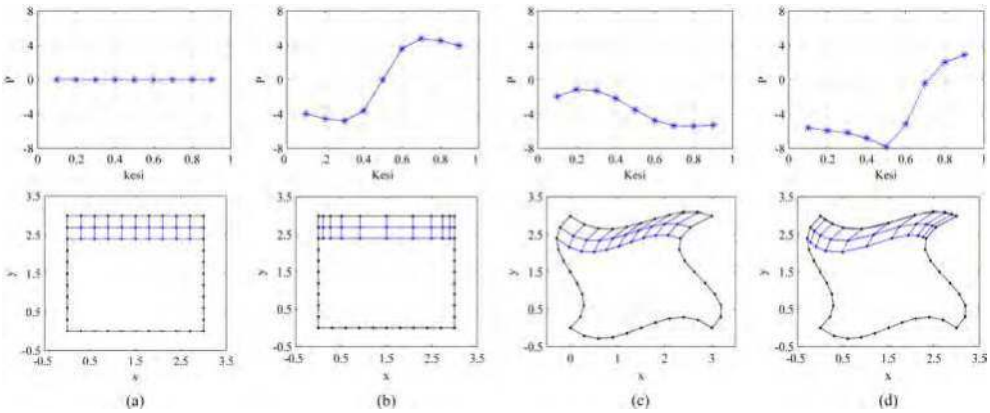


Fig. 2. Sensitivity of the source term, P, with respect to the boundary geometry and nodal distribution.

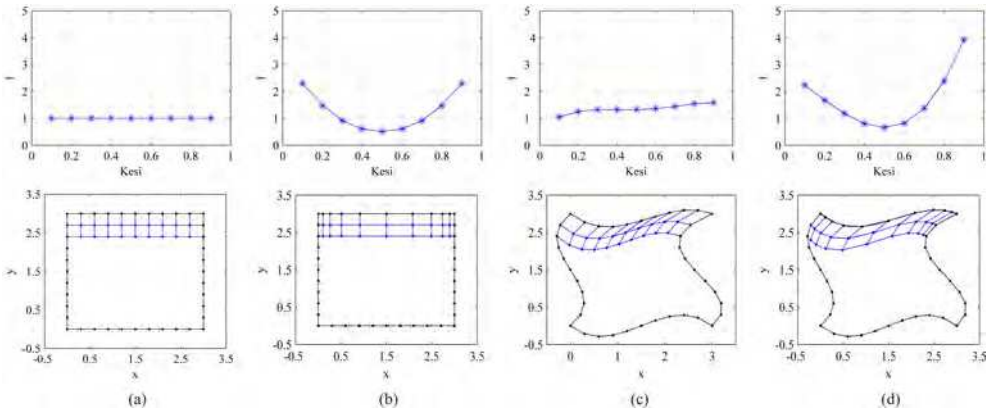


Fig. 3. Sensitivity of the scale factor, f, with respect to the boundary geometry and nodal distribution.

3. Interpolation techniques

Having chosen the interpolants, an interpolation technique is required to find the corresponding values at internal nodes. The idea is that the boundary data should be used to determine the geometrical properties of the internal cells and coordinate lines via interpolation techniques. There are three geometrical interpolation techniques that can be used in a multi-dimensional problem as follows.

3.1 One-dimensional interpolation

Univariate stretching functions provide the relations for the one dimensional interpolation. Equations (12) and (13) are examples of one dimensional interpolation formulas used in a 2D problem.

3.2 Quasi multi-dimensional interpolation

Quasi multi-dimensional interpolation techniques such as TFI, which employs the Boolean sum of 1D interpolations, can also be used to interpolate the chosen interpolants in an elliptic grid generation process. The interpolation coefficients in these algebraic methods can be constant, linear or non-linear functions of the logical or physical coordinates. Use of the physical coordinates as the independent variables in the interpolation coefficients worsens the nonlinearity of the interpolation and, therefore, this option has not gained any popularity. As an example, it will be shown later that the boundary values of the control functions in the TTM method can also be interpolated via the TFI.

3.3 Multi-dimensional interpolation

The interpolants in the EGG methods can also be interpolated by truly multi-dimensional methods, i.e. by the solution of boundary value problems. For example, it will be shown that the boundary values of the control functions in the TTM method can also be interpolated through the solution of Dirichlet boundary value problems to obtain the corresponding values for the internal nodes.

4. Grid generation equations

By viewing the elliptic grid generation problem as a multi-dimensional interpolation problem, the focus is obviously on the selection of interpolants and the interpolation techniques. However, depending on the chosen interpolants, it may also be necessary to develop grid generation equations, i.e. equations which ultimately provide the nodal coordinates throughout the domain.

The process of interpolation may actually play the role of the grid generation equations. In an algebraic grid generation method such as the zero order TFI, the interpolants are the nodal coordinates. By carrying out the interpolation, nodal coordinates are obtained throughout the domain and no additional grid generation equation is required. In other words, the interpolation equations in this case are themselves the grid generation equations.

The expressions used to define the interpolants may also be used to develop the grid generation equations. For example, in the classical TTM method, in which Eqs. (5) and (6)

are used to define the interpolants, the same equations are also inverted to obtain the grid generation equations in the logical domain. In the method proposed by Ashrafizadeh and Raithby (Ashrafizdeh & Raithby, 2006), Eqs. (5) and (6) are used to obtain the grid generation equations in the physical domain.

The classical orthogonal grid generation method is a good representative example of the cases in which neither the interpolation formulas nor the definition of the interpolants can be used for the grid generation. In contrast to the TTM, the interpolated values of the scale factor are not directly used to calculate the coordinates of internal nodes in the OGG. First order differential interpolants, such as the scale factor f, are not appropriate for the calculation of the nodal coordinates. Therefore, second order differential equations are developed by imposing the continuity constraints on the mapping functions. Consequently, Eqs. (14) and (15) are obtained and used as the grid generation equations.

Now that the commonly used grid generation methods are explained in the framework of the proposed unifying view, a number of alternative elliptic grid generation methods are introduced in the next section. The objective is to show that how new elliptic grid generation methods can be developed in the context of the suggested vantage point.

5. Alternative grid generation methods

Based on the proposed view point, there are many possible alternatives for the development of elliptic grid generation methods. By focusing on the interpolants and the interpolation techniques, EGG methods can be divided into the following four categories:

- Algebraic interpolation of Algebraic interpolants (AA methods).
- Algebraic interpolation of Differential interpolants (AD methods).
- Differential interpolation of Algebraic interpolants (DA methods).
- Differential interpolation of Differential interpolants (DD methods).

As mentioned before, the selection of appropriate grid generation equations provides another degree of freedom in the development of elliptic grid generation methods. Here we present two alternative grid generation methods in each category. It is clear that there are other possibilities and one may develop new grid generation methods in the proposed context.

5.1 AA methods

5.1.1 The AA1 method

This method works with algebraic boundary data and employs an algebraic interpolation formula with constant coefficients. Grid generation equations here are algebraic interpolation formulas similar to Eq. (4) applied to a sub-domain close to the nodal point, shown in Fig. 4, as follows:

$$x_P = 0.5\left(x_W + x_E + x_S + x_N\right) - 0.25\left(x_{NW} + x_{NE} + x_{SW} + x_{SE}\right) \tag{17}$$

$$y_P = 0.5\left(y_W + y_E + y_S + y_N\right) - 0.25\left(y_{NW} + y_{NE} + y_{SW} + y_{SE}\right) \tag{18}$$

Note that here algebraic formulas are used to ultimately interpolate the coordinates of boundary nodes (the interpolants). The nine-point computational molecules provide two sets of simultaneous equations which have to be solved to obtain the coordinates of grid points. In the context of the classical EGG methods it is hard to call this method an algebraic grid generator. We prefer to avoid the confusion by simply associate the method to the selected interpolants and the mathematical nature of the interpolation technique.

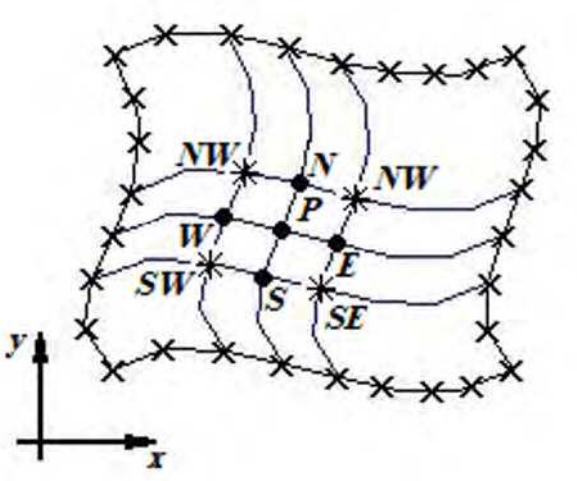


Fig. 4. Contributing nodes in the AA1, DA1 and DD1 methods.

5.1.2 The AA2 method

This method provides a combined local/global interpolation formula. Coordinates of some adjacent and neighbor boundary nodes are interpolated in a TFI-like interpolation procedure. To obtain the nodal coordinates, the following procedure is carried out:

- Coordinates of node P are calculated using zero order TFI in the shaded area in Fig. 5a. The contributing nodes are shown by solid dots in Fig. 5a. The calculated coordinates at this stage are called $(x_1, y_1)_P$.
- A similar procedure is carried out using the TFI in shaded areas shown in Figs 5b, 5c and 5d to obtain new coordinates $(x_2, y_2)_P$, $(x_3, y_3)_P$ and $(x_4, y_4)_P$.
- The final coordinates of node P, i. e. $(x, y)_P$, are obtained as follows:

$$(x,y)_P = C_1(x_1,y_1)_P + C_2(x_2,y_2)_P + C_3(x_3,y_3)_P + C_4(x_4,y_4)_P \tag{19}$$

The simplest choice for the weight factors would be $C_1 = C_2 = C_3 = C_4 = 0.25$. The weight factors may also be chosen taking into consideration the logical coordinates of point P.

Note that formula for $(x_1, y_1)_P$ includes coordinates of some boundary points as well as points P and P_1. Similarly, the $(x_2, y_2)_P$, $(x_3, y_3)_P$ and $(x_4, y_4)_P$ terms bring the coordinates of points P_2, P_3 and P_4 in the mix. Therefore, as compared to the traditional

zero order TFI method, which takes the coordinates of 8 boundary nodes to calculate the coordinates of an internal node P , this method employs the information at 16 boundary nodes as well as 4 neighbor nodes to construct a computational molecule for nodal point P . Considering the fact that the coordinates of boundary nodes are known, Eq. (14) can be re-written as 5-point computational molecule for the coordinates of node P as follows:

$$(x,y)_P = C_{P_1}(x,y)_{P_1} + C_{P_2}(x,y)_{P_2} + C_{P_3}(x,y)_{P_3} + C_{P_4}(x,y)_{P_4} + C_P \tag{20}$$

Coefficient C_P in Eq. (20) includes the effects of the above mentioned 16 boundary nodes. In contrast to the classical TFI, a simultaneous set of equations needs to be solved to obtain the coordinates of internal nodes. If only two shaded areas shown in Figs. 5a and 5b are used to develop the interpolation formulas, the coordinates of internal nodes can be obtained in a marching calculation process starting from the south east corner of the domain.

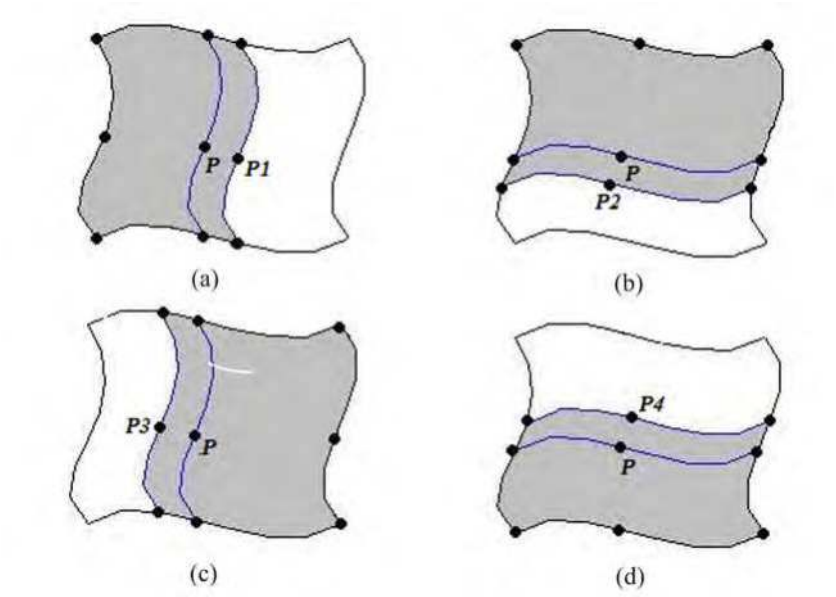


Fig. 5. Four sub-domains used to develop the computational molecule in the AA2 method.

5.2 AD methods

5.2.1 The AD1 method

In this method the interpolants are some differential boundary data interpolated by algebraic interpolation formulas. The contributing boundary nodes in each computational molecule are shown in Fig. 6 by × signs. Second order derivatives of boundary coordinates are interpolated by the algebraic TFI method:

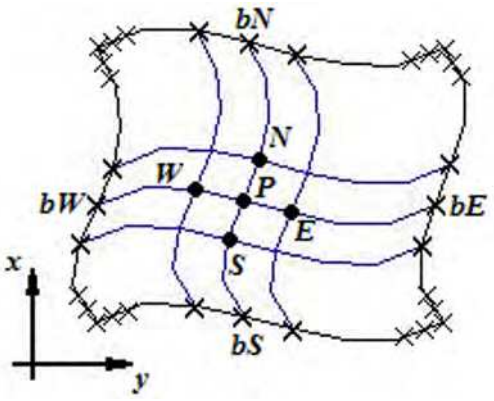


Fig. 6. Contributing nodes in the AD1 method.

$$\begin{aligned}\left(x_{\eta\eta},y_{\eta\eta}\right)_P &= \left(1-\xi_P\right)\left(x_{\eta\eta},y_{\eta\eta}\right)_{bW}+\left(\xi_P\right)\left(x_{\eta\eta},y_{\eta\eta}\right)_{bE}+\left(1-\eta_p\right)\left(x_{\eta\eta},y_{\eta\eta}\right)_{bS}\\ &+\left(\eta_P\right)\left(x_{\eta\eta},y_{\eta\eta}\right)_{bN}-\left(1-\xi_P\right)\left(1-\eta_p\right)\left(x_{\eta\eta},y_{\eta\eta}\right)_{P\text{-}SW}\\ &-\left(1-\xi_P\right)\left(\eta_P\right)\left(x_{\eta\eta},y_{\eta\eta}\right)_{P\text{-}NW}-\left(\xi_P\right)\left(1-\eta_p\right)\left(x_{\eta\eta},y_{\eta\eta}\right)_{P\text{-}SE}\\ &-\left(\xi_P\right)\left(\eta_P\right)\left(x_{\eta\eta},y_{\eta\eta}\right)_{P\text{-}NE}\end{aligned} \tag{21}$$

$$\begin{aligned}\left(x_{\xi\xi},y_{\xi\xi}\right)_P &= \left(1-\xi_P\right)\left(x_{\xi\xi},y_{\xi\xi}\right)_{bW}+\left(\xi_P\right)\left(x_{\xi\xi},y_{\xi\xi}\right)_{bE}+\left(1-\eta_p\right)\left(x_{\xi\xi},y_{\xi\xi}\right)_{bS}\\ &+\left(\eta_P\right)\left(x_{\xi\xi},y_{\xi\xi}\right)_{bN}-\left(1-\xi_P\right)\left(1-\eta_p\right)\left(x_{\xi\xi},y_{\xi\xi}\right)_{P\text{-}SW}\\ &-\left(1-\xi_P\right)\left(\eta_P\right)\left(x_{\xi\xi},y_{\xi\xi}\right)_{P\text{-}NW}-\left(\xi_P\right)\left(1-\eta_p\right)\left(x_{\xi\xi},y_{\xi\xi}\right)_{P\text{-}SE}\\ &-\left(\xi_P\right)\left(\eta_P\right)\left(x_{\xi\xi},y_{\xi\xi}\right)_{P\text{-}NE}\end{aligned} \tag{22}$$

Discrete forms of Eqs (21) and (22) result in the following formulas for the coordinates of node P:

$$\left(x_1,y_1\right)_P = C_{1S}\left(x,y\right)_{1S}+C_{1N}\left(x,y\right)_{1N}+C_{1P} \tag{23}$$

$$\left(x_2,y_2\right)_P = C_{2E}\left(x,y\right)_{2E}+C_{2W}\left(x,y\right)_{2W}+C_{2P} \tag{24}$$

Coefficients C_{1P} and C_{2P} include the contribution of boundary nodes corresponding to the nodal point P as shown in Fig. 6. The final computational molecule is obtained as follows:

$$\left(x,y\right)_P = \beta_{1P}\left(x_1,y_1\right)_P+\beta_{2P}\left(x_2,y_2\right)_P \tag{25}$$

Again the simplest choice for the weight factors would be $\beta_{1P}=\beta_{2P}=0.5$, however, these factors may also be chosen taking into consideration the logical coordinates of point P.

Therefore, in this method the coordinates of each node are constrained directly by the coordinates of 4 neighbor nodes and indirectly by 32 boundary nodes. Once again all boundary nodes contribute to the calculation of the coordinates of node P through the solution of a set of algebraic equations similar to Eq. (25).

5.2.2 The AD2 method

In this method the interpolants are the source functions P and Q, defined in Eqs. (5) and (6). The zero-order TFI is used for the interpolation of the interpolants as follows:

$$\begin{aligned}(P)_P = &(1-\xi_P)(P)_{bW} + (\xi_P)(P)_{bE} + (1-\eta_p)(P)_{bS} + (\eta_P)(P)_{bN} \\ &-(1-\xi_P)(1-\eta_p)(P)_{P\text{-}SW} - (1-\xi_P)(\eta_P)(P)_{P\text{-}NW} \\ &-(\xi_P)(1-\eta_p)(P)_{P\text{-}SE} - (\xi_P)(\eta_P)(P)_{P\text{-}NE}\end{aligned} \tag{26}$$

$$\begin{aligned}(Q)_P = &(1-\xi_P)(Q)_{bW} + (\xi_P)(Q)_{bE} + (1-\eta_p)(Q)_{bS} + (\eta_P)(Q)_{bN} \\ &-(1\quad \xi_P)(1-\eta_p)(Q)_{P\ \ SW} - (1-\xi_P)(\eta_P)(Q)_{P\text{-}NW} \\ &-(\xi_P)(1-\eta_p)(Q)_{P\text{-}SE} - (\xi_P)(\eta_P)(Q)_{P\text{-}NE}\end{aligned} \tag{27}$$

Equations (5) and (6) are used as the grid generation equations.

5.3 DA methods

5.3.1 The DA1 method

In this case Dirichlet boundary value problems are solved to interpolate algebraic boundary data, i.e. coordinates of boundary nodes. All boundary nodes, shown by × signs in Fig. 4, indirectly contribute to the calculation of coordinates of each internal node. The interpolation formulas, which are actually the grid generation equations, are mathematical expressions which imply the smoothness of functions $x(\xi,\eta)$ and $y(\xi,\eta)$ as follows:

$$x_{\xi\xi} + x_{\eta\eta} = 0 \tag{28}$$

$$y_{\xi\xi} + y_{\eta\eta} = 0 \tag{29}$$

Nodal points which contribute to the computational molecule for the calculation of the coordinates of node P are shown by • signs in Fig. 4. It is interesting to note that Eqs. (28) and (29) correspond also to the conformal mapping. These equations are also obtained by setting $f = 1$ in Eqs. (14) and (15).

5.3.2 The DA2 method

In this case an initial grid in a simple domain, (x_0, y_0), is used to generate the grid in the physical domain. The interpolants are algebraic quantities $\delta x = x - x_0$ and $\delta y = y - y_0$, in which (x, y) are the corresponding boundary nodes of the target geometry. The target domain and the initial grid are both shown in Fig. 7. The interpolation formulas, which are

actually the grid generation equations, are mathematical expressions for a two-dimensional interpolation of the boundary values of δx and δy as follows:

$$\nabla_0^2(\delta x) = \frac{\partial^2(\delta x)}{(\partial x_0)^2} + \frac{\partial^2(\delta x)}{(\partial y_0)^2} = 0 \tag{30}$$

$$\nabla_0^2(\delta y) = \frac{\partial^2(\delta y)}{(\partial x_0)^2} + \frac{\partial^2(\delta y)}{(\partial y_0)^2} = 0 \tag{31}$$

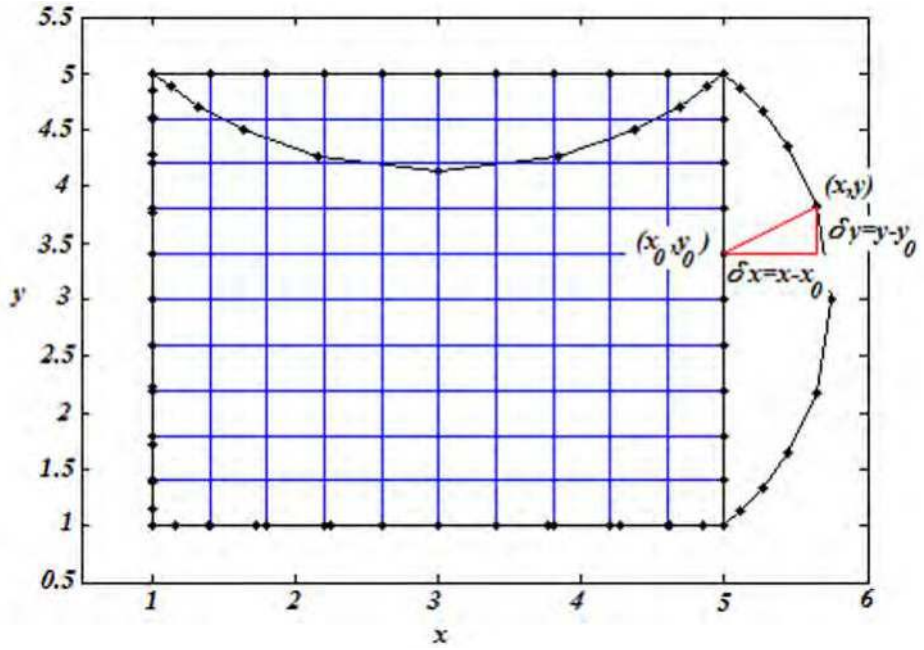


Fig. 7. Grid generation by interpolating the nodal boundary displacements (the DA2 method).

Both finite difference and finite volume methods can be used to numerically solve these equations. This method can also be used to re-mesh the computational domain in a moving boundary problem. More discussion on this method can be found in (Ashrafizadeh et al., 2009).

5.4 The DD methods

5.4.1 The DD1 method

In this example of a DD method, two differential quantities, $P = x_{\xi\xi} + x_{\eta\eta}$ and $Q = y_{\xi\xi} + y_{\eta\eta}$, are calculated at all nodes adjacent to the boundary using the paving layers as explained in (Ashrafizadeh & Raithby, 2006). These boundary data are then interpolated differentially as follows:

$$P_{\xi\xi} + P_{\eta\eta} = 0 \tag{32}$$

$$Q_{\xi\xi} + Q_{\eta\eta} = 0 \tag{33}$$

With the P and Q terms available at all internal nodes, the following grid generation equations are solved to obtain the internal nodal coordinates:

$$x_{\xi\xi} + x_{\eta\eta} = P \tag{34}$$

$$y_{\xi\xi} + y_{\eta\eta} = Q \tag{35}$$

Here again all boundary nodes, depicted by × signs in Fig. 4, contribute through the implementation of boundary conditions in the interpolation procedure for each internal node.

5.4.2 The DD2 method

Another alternative for the description of the boundary information is to use $P = \xi_{xx} + \xi_{yy}$ and $Q = \eta_{xx} + \eta_{yy}$ interpolants. Boundary values of these source functions are interpolated by a multi-dimensional interpolation technique, i.e. Eqs. (32) and (33), and the coordinates are generated by solving Eqs. (5) and (6). This method is similar to the TTM as employed in (Thomas & Middlecoff, 1980) except that a multi-dimensional interpolation method is used to calculate internal values of the source functions.

6. A brief discussion

It is worthwhile to mention few points here for further clarification:

1. The grid generation methods, just introduced, are few examples of many methods that can be developed based on the three main choices in the proposed unifying view, i. e. the choice of the interpolants, the choice of the interpolation technique and the choice of the grid generation equations. For example, a family of new methods, not discussed here, have also been developed by the authors which employ the transformation metrics at or near the boundary as interpolants. Such methods may be viewed as a continuation, and generalization, of the orthogonal grid generation method.
2. The smooth distribution of the source terms in the TTM method is a sign of grid smoothness. By properly choosing the interpolants, the interpolation technique and the grid generation equations, boundary coordinate lines are smoothly interpolated into the domain and a smooth distribution of source terms is obtained.
3. The possibility of folding exists in nearly all of the commonly used elliptic grid generation methods except for the TTM with $P = Q = 0$ on sufficiently fine grids. Therefore, the above new grid generation methods may result in folded grids for some geometrically complex domains. However, the objective in the development of new elliptic grid generation techniques is to obtain methods which generate high quality grids and are more resilient to folding as compared to the classical methods.
4. Many of the alternative grid generation methods presented here are executed much faster than the classical methods. They may also be used to generate the background or initial grids for other more expensive grid generators.

7. Grid generation examples

The performances of the proposed grid generation methods are now studied by solving various grid generation problems. In this section two geometries, often used to test the elliptic grid generators and here called the test cases, are chosen to examine some of the proposed methods and to also compare them with the classical elliptic and algebraic grid generation methods. The first test case is a quadrilateral domain, for which all four boundaries are distorted. The second test case is also a quadrilateral domain for which only two of the neighboring boundaries are distorted. A (11×11) grid is generated in all test cases. Finer, and nicer, grids can obviously be generated but we have chosen a rather coarse grid to be able to visualize the details of the performance of the methods.

The EGG methods can be compared in terms of the computational cost and the grid quality measures. Considering the fact that the grid generation cost depends mostly on the cost of the solution of the grid generation equations, the comparison in terms of the computational cost seems a trivial task in nearly all cases. As a general guideline, the solution of a nonlinear set of equations is computationally more expensive than the solution of a linear set. Regarding the grid quality, two parameters, i.e. the skewness and the aspect ratio, are chosen as the quality measures in this study. Skewness of a cell varies between 0 and 1 and measures the deviation from the orthogonality of the coordinate lines. Aspect ratio of a cell is defined as the ratio of the longest edge length to the shortest one and measures the deviation from a square cell. Cells in the logical space have zero skewness and aspect ratio equal to one.

Figure 8 shows the grids generated by the classical methods in the test domains. The generated grids by the zero-order TFI are shown in Figs. 8a and 8b. Figures 8c and 8d show the grids obtained from the TTM and Figs. 8e and 8f are orthogonal grids generated by the OGG. It is seen that the OGG method results in folded grids in both test cases. The corresponding grid quality measures are shown in Figs. 9a and 9b, 9c and 9d, and 9e and 9f respectively. Note that there are 10 cells along each coordinate line in the test grids. Each quality measure diagram shows the relevant quality measures for all 100 cells on a three-dimensional plot containing 10×10 data points.

Figures 10, 11, 12 and 13 show the grids in the test geometries obtained via the proposed new methods. It can be seen that all of the methods provide unfolded grids comparable to the grids obtained by the classical methods. Furthermore, and as expected, it is obvious that methods which employ differential interpolation techniques result in smoother grids.

The spatial distribution of the source functions $P=\xi_{xx}+\xi_{yy}$ and $Q=\eta_{xx}+\eta_{yy}$ using three different interpolation techniques are shown in Fig. 14. One dimensional interpolation is used to interpolate the control functions shown in Figs. 14a and 14b. The grids corresponding to these control functions are shown in Figs. 8c and 8d. Figures 14c and 14d show the distributions of control functions, which are obtained through a quasi-two dimensional method, i.e. the TFI. The grids corresponding to these control functions are shown in Figs. 11c and 11d. Finally, Figs. 14e and 14f show the interpolated control functions via a truly two-dimensional interpolation method, i.e. the solution of Dirichlet boundary value problems. The grids corresponding to these control functions are shown in

Figs. 13c and 13d. As expected, it is seen that the grids corresponding to the control functions shown in Figs. 14e and 14f are smoother than the other grids. A stretched grid with higher number of nodes is shown in Fig. 15 to show the applicability of AD1 in more complex domains. Similar results can be obtained via other proposed methods.

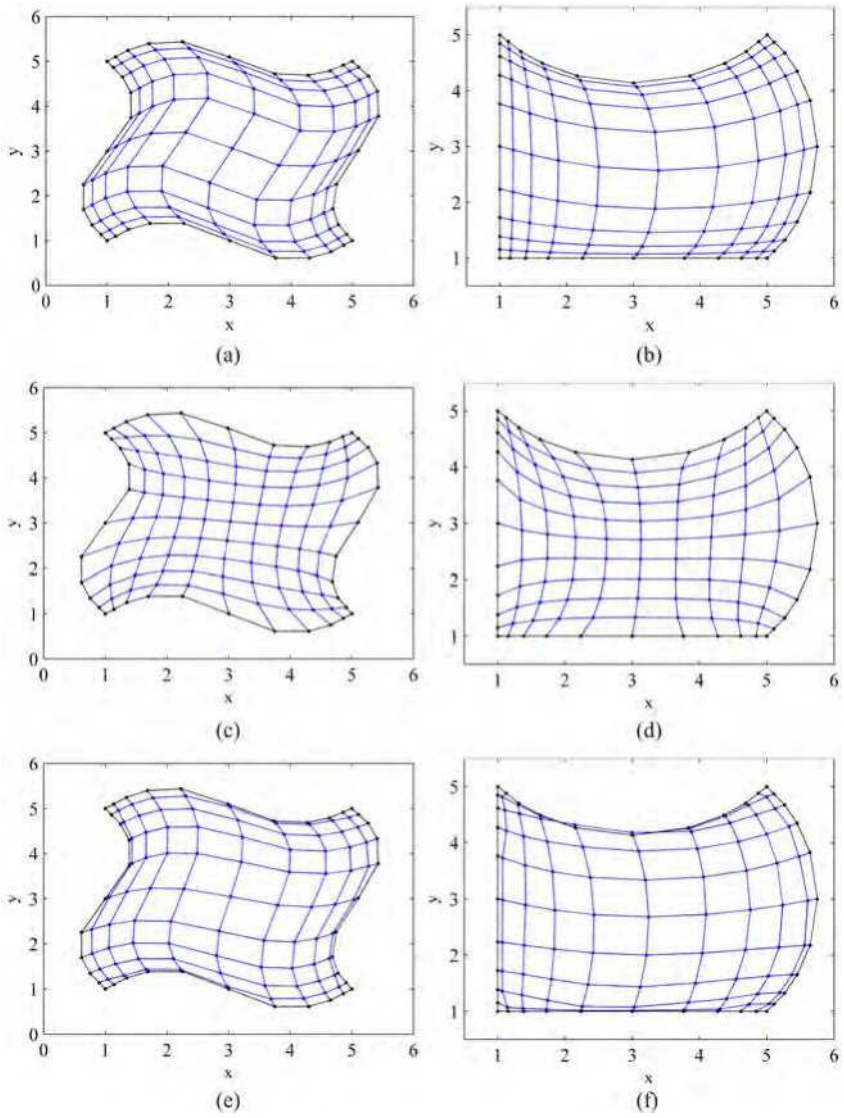


Fig. 8. Generated grids by TFI (a,b), TTM (c,d) and OGG (e,f).

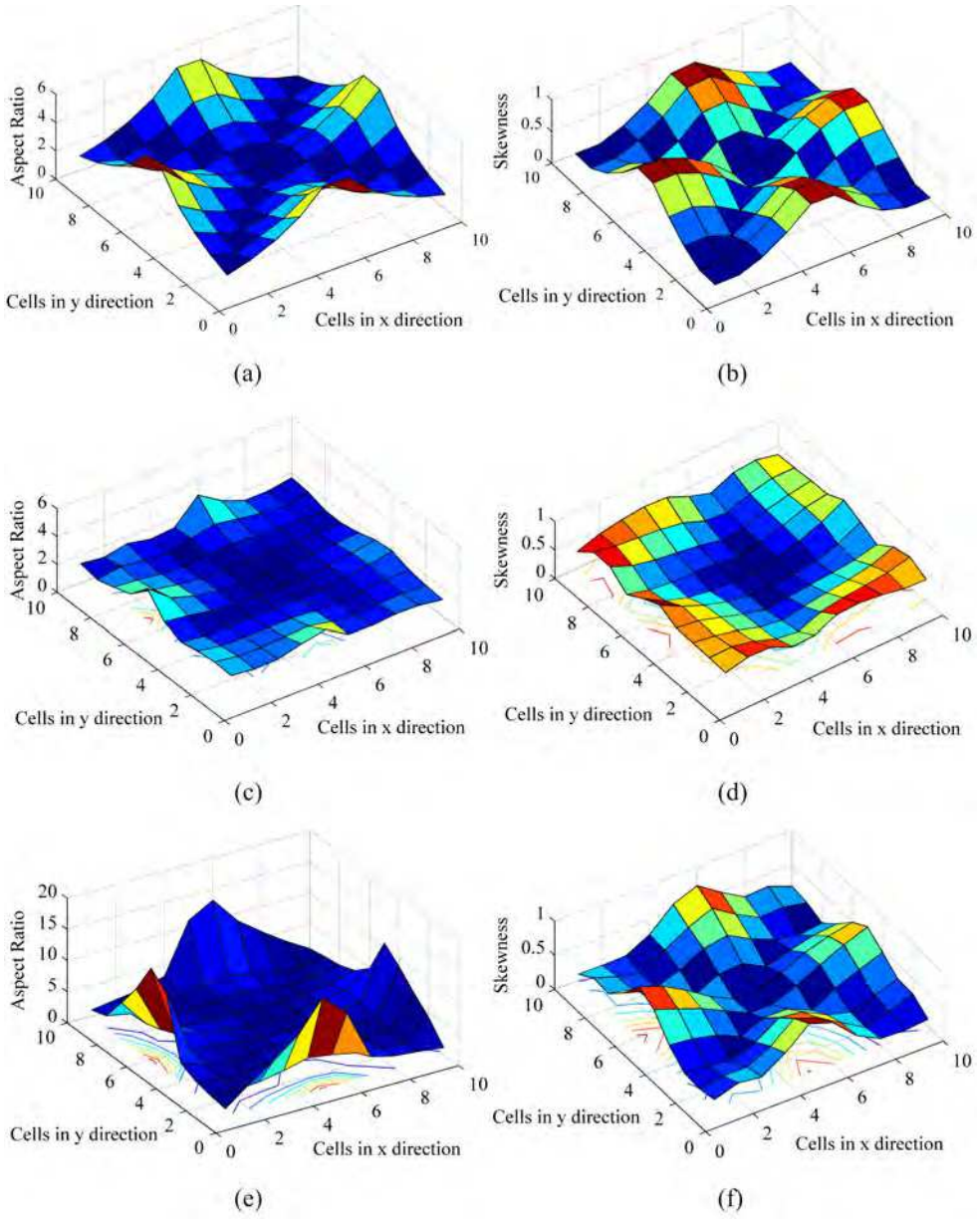


Fig. 9. Quality measures for the grids generated by TFI (a,b), TTM (c,d) and OGG (e,f).

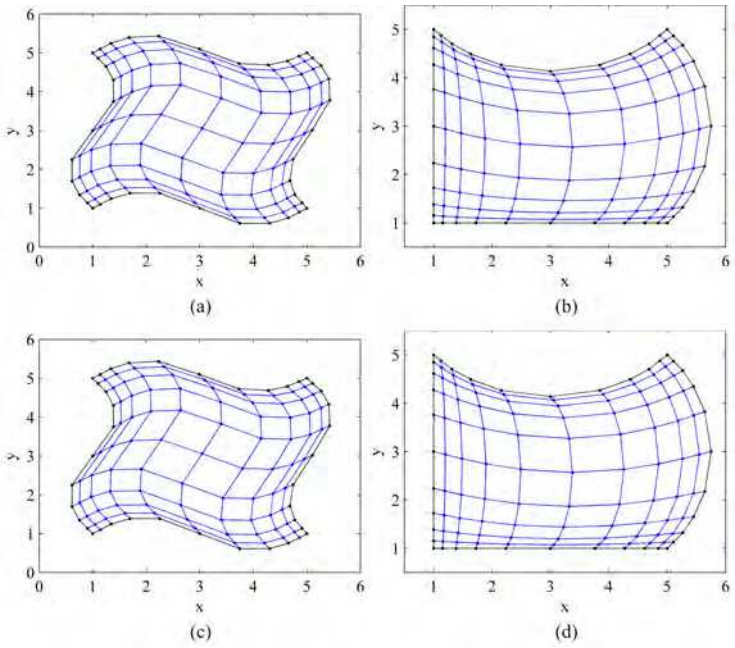


Fig. 10. Generated grids by AA1 (a, b), AA2 (c, d).

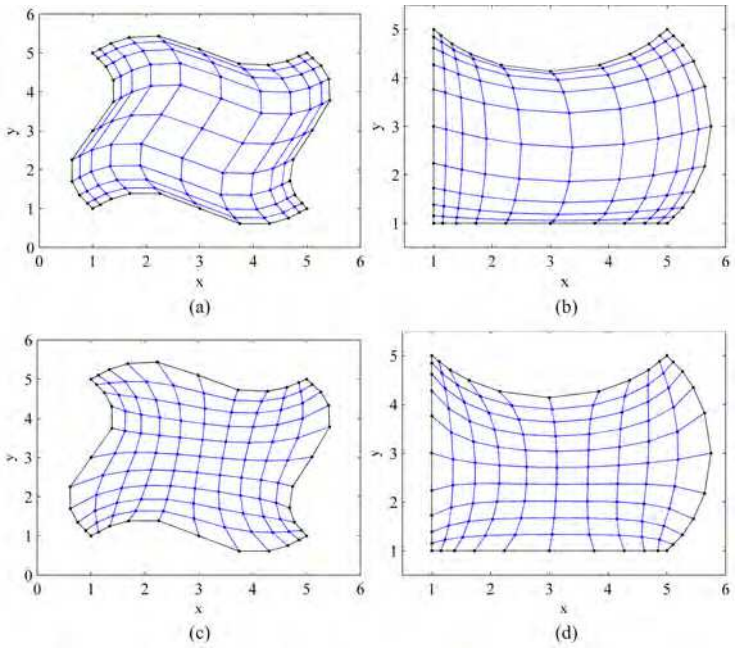


Fig. 11. Generated grids by AD1 (a, b) and AD2 (c, d).

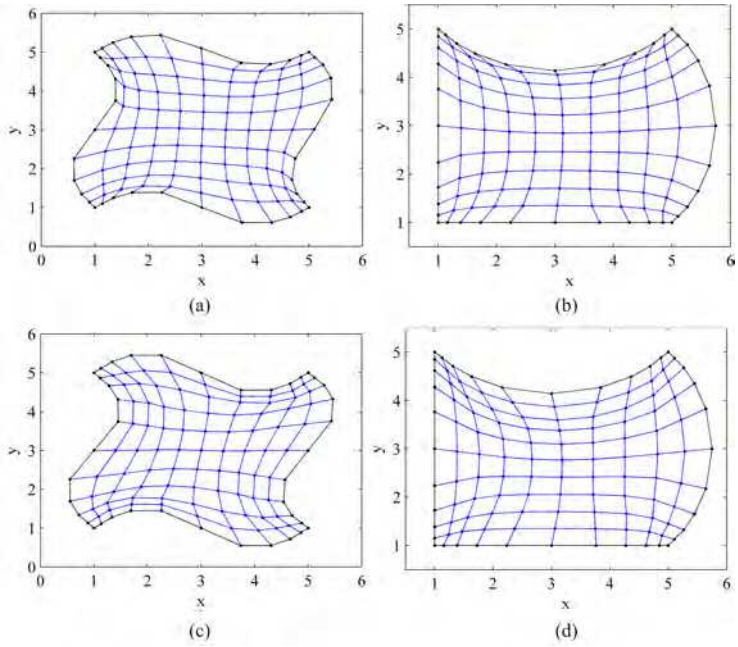


Fig. 12. Generated grids by DA1 (a, b) and DA2 (c, d).

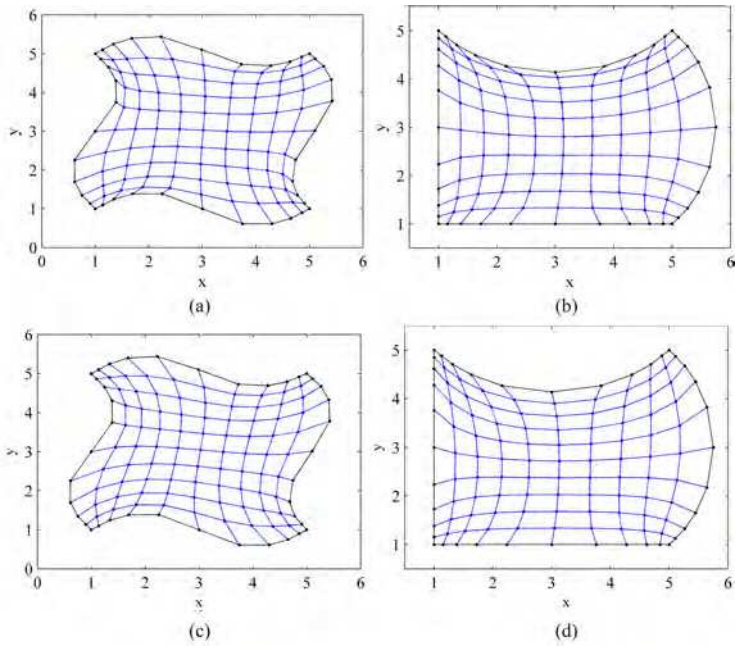


Fig. 13. Generated grids by DD1 (a, b) and DD2 (c, d).

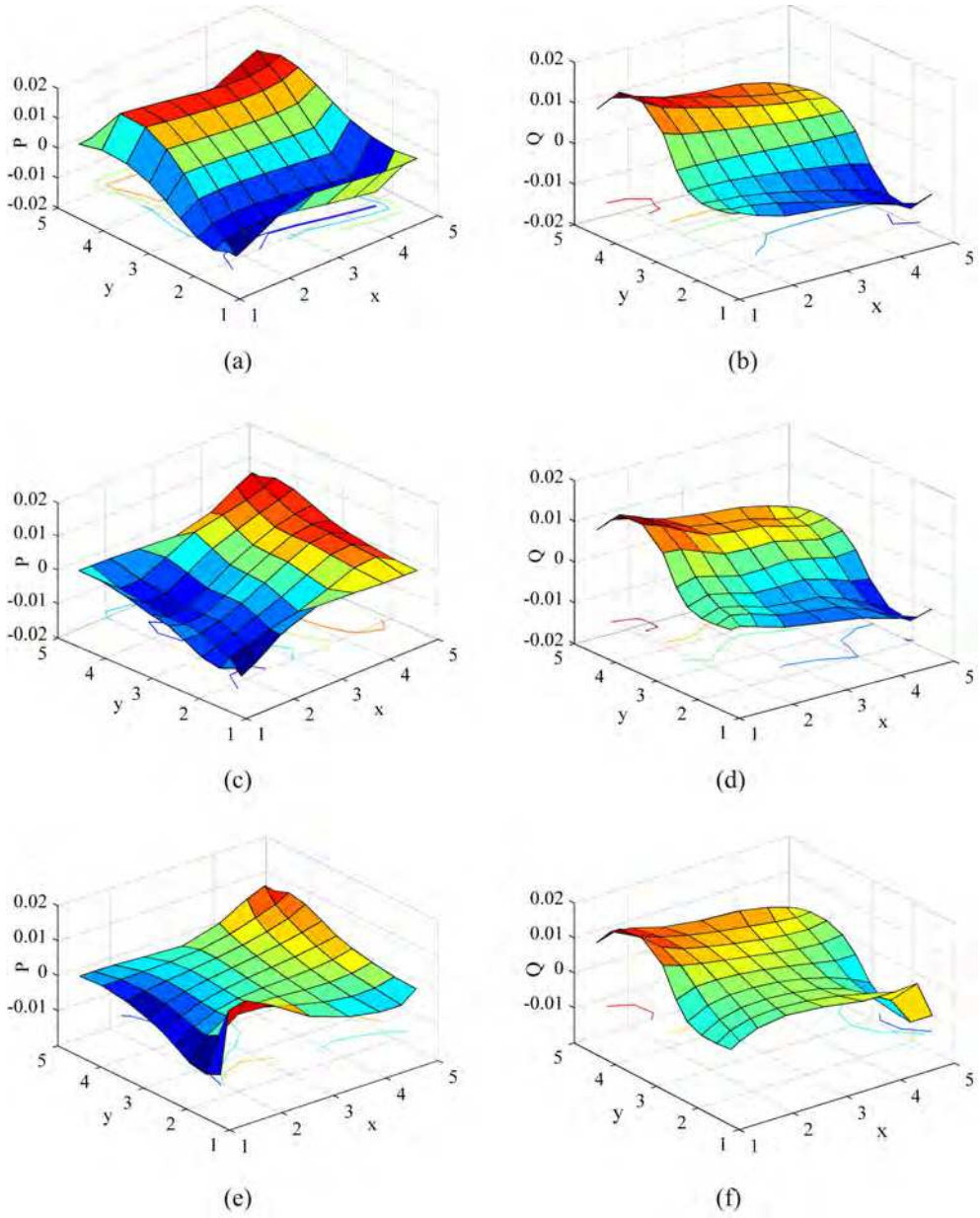


Fig. 14. Calculated P and Q by 1D interpolation (a, b), AD2 (c, d) and DD2 (e, f).

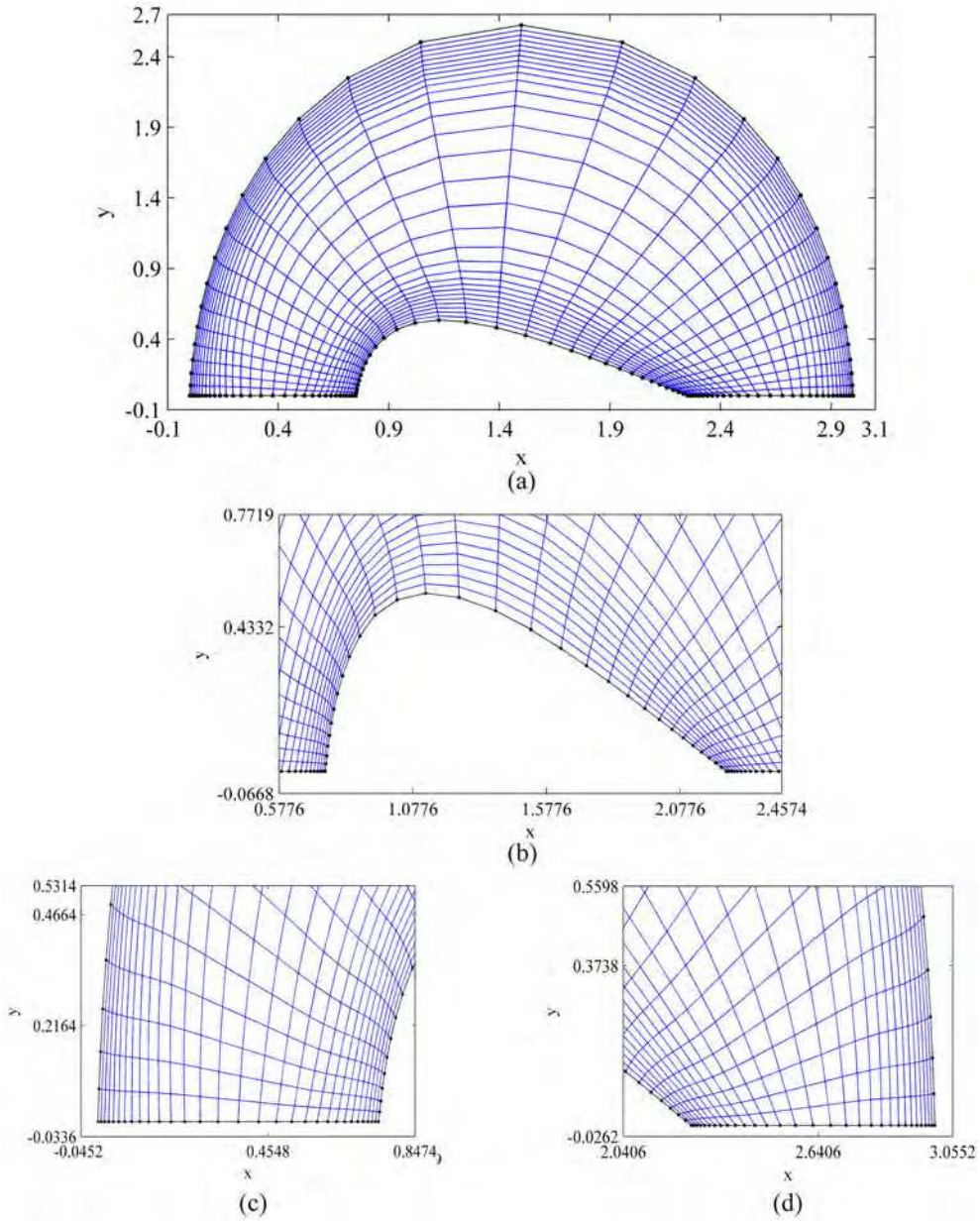


Fig. 15. A sample grid generated by AD1 (a), and a larger view of sections of the grid (b, c and d).

8. Conclusion

A unified view of the elliptic grid generation is proposed in this Chapter. It is argued that elliptic grid generation techniques are actually methods for multi-dimensional geometrical interpolation and can be described in terms of the interpolants, interpolation technique, and grid generation equations. Interpolants are used to describe the boundary shape and nodal distribution, interpolation technique is used to bring the boundary data into the domain, and grid generation equations are used to calculate the internal nodal coordinates. The most commonly used classical elliptic grid generation methods are explained in the context of the proposed unified view and new grid generation methods are also presented in the same context. A number of grid generation examples are chosen to show the applicability of the proposed methods. Authors believe that the proposed unified view provides a systematic and comprehensible approach to explain and develop a large class of elliptic grid generation methods. Some of these methods are computationally cheaper than the existing methods, yet provide grids with comparable qualities.

9. References

Akcelik V., Jaramaz B.and Ghattas O. (2001). Nearly Orthogonal Two Dimensional Grid Generation with Aspect Ratio Control. *Journal of Computational Physics,* Vol. 171, Issue 2, (August 2001), pp. 805-821, ISBN No. 0021-9991.

Ashrafizadeh, A.; Raithby, G. D. & Stubley, G. D. (2002). Direct Design of Shape. *Numerical Heat Transfer*, Part B, Vol. 41, (June 2002), pp. 501-520, ISSN: 1040-7790 print/ 1521-0626 online.

Ashrafizadeh, A.; Raithby, G. D. & Stubley, G. D. (2003). Direct Design of Ducts. *Journal of Fluids Engineering,* Transactions of ASME, Vol. 125, (January 2003), pp. 158-165, ISSN: 0098-2202 print/ 1528-901X online.

Ashrafizadeh, A.; & Raithby, G. D. (2006). Direct Design Solution of the Elliptic Grid Generation Equations. *Numerical Heat Transfer*, Part B, Vol. 50, (September 2006), pp. 217-230, ISSN: 1040-7790 print/ 1521-0626 online.

Ashrafizadeh, A.; Jalalabadi, R. & Bazargan, M. (2009). A New Structured Grid Generation Method. *Proceedings of 11th ISGG Conference,* Ecole Polytechnique: Montreal, Canada, May, 2009.

Bourchtein A. & Bourchtein. L. (2006). On Generation of Orthogonal Grids. *Journal of Applied Mathematics and Computation,* Volume 173, Issue 2, (February 2006), pp. 767-781, ISSN: 0096-3003.

Duraiswami R. & Prosperetti. A. (1992). Orthogonal Mapping in Two Dimensions. *Journal of Computational Physics*, Vol. 98, Issue 2, (February 1992), pp. 254-268, ISBN No. 0021-9991.

Eca. L. (1996). Two Dimensional Orthogonal Grid Generation with Boundary Point Distribution Control. *Journal of Computational Physics,* Vol. 125, Issue 2, (May 1996), pp. 440-453, ISBN No. 0021-9991.

Eiseman, P. R.; Choo, Y. K. & Smith, R. E. (1992). Algebraic. Grid Generation with Control Points. *Finite Elements in Fluids,* Vol. 8, (1992), pp. 97-116, ISBN No. 0-89116-850-8.

Eiseman, P. R. (1979). A Multi-Surface Method of Coordinate Generation. *Journal of Computational Physics,* Vol. 33, Issue 1, (October 1979), pp. 118-150, ISBN No. 0021-9991.

Kang. I.S. & Leal. L.G. (1992). Orthogonal Grid Generation in a 2D Domain via the Boundary Integral Technique. *Journal of Computational Physics,*Vol. 102, Issue 1, (September 1992), pp. 78-87, ISBN No. 0021-9991.

Kaul. U. K. (2003). New Boundary Constraints for Elliptic systems used in Grid Generation Problems. *Journal of Computational Physics,* Vol. 189, Issue 2, (August 2003), pp. 476-492, ISBN No. 0021-9991.

Kaul. U. K. (2010). Three-Dimensional Elliptic Grid Generation with Fully Automatic Boundary Constraints. *Journal of Computational Physics,* Vol. 229, Issue 17, (August 2010), pp. 5966-5979, ISBN No. 0021-9991.

Lee, S.H. & Soni, B.K. (2004). The Enhancement of an Elliptic Grid Using Appropriate Control Functions, *Journal of Applied Mathematics and Computation,* Vol. 159, Issue 3, (December 2004), pp. 809-821, ISSN: 0096-3003.

Lehtimaki. R. (2000). An Algebraic Boundary Orthogonalization Procedure for Structured Grids. *International Journal for Numerical Methods in Fluids,* Vol. 32, Issue 5, (March 2000), pp. 605–618, Online ISSN: 1097-0363.

Ryskin G. & Leal. L.G. (1983). Orthogonal Mapping. *Journal of Computational Physics,* Vol. 50, Issue 1, (April 1983), pp. 71-100, ISBN No. 0021-9991.

Spekreijse, S. P. (1995). Elliptic Grid Generation Based on Laplace Equations and Algebraic Transformations. *Journal of Computational Physics*, Vol. 118, Issue 1, (April 1995), pp. 38-61, ISBN No. 0021-9991.

Steger. J. L. & Sorensos. R. L. (1997). Automatic Mesh-Point Clustering Near Boundary in Grid Generation with Elliptic Partial Differential Equations. *Journal of Computational Physics,* Vol. 33, Issue 3, (December 1979), pp. 405-410, ISBN No. 0021-9991.

Thomas, P.D. & Middlecoff, J. F. (1980). Direct Control of the Grid Points Distribution in Meshes Generated by Elliptic Equations. *AIAA Journal,* (1980), Vol. 18, no.6, pp. 652-656, ISSN: 0001-1452 print/ 1533-385X.

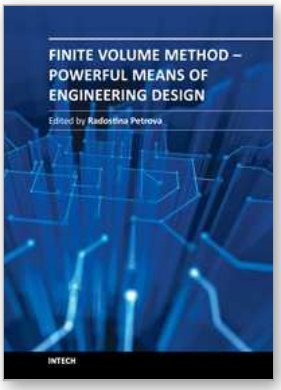
Thompson, J. F.; Thames, F. C. & Mastin, C. W. (1974). Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Fields Containing Any Number of Arbitrary Two-Dimentional Bodies. *Journal of Computational Physics,* Vol. 15, Issue 3, (July 1974), pp. 299-319, ISBN No. 0021-9991.

Zhang Y.; Jia Y. & Wang S. S. Y. (2004). 2D Nearly Orthogonal Mesh Generation. *International Journal for Numerical Methods in Fluids,* Vol. 46, Issue 7, (November 2004), pp. 685–707, Online ISSN: 1097-0363.

Zhang Y.; Jia Y. & Wang S. S. Y. (2006). 2D Nearly Orthogonal Mesh Generation with Controls of Distortion Function. *Journal of Computational Physics*, Vol. 218, Issue 2, (November 2006), pp. 549-571, ISBN No. 0021-9991.

Zhang Y.; Jia Y. & Wang S. S. Y. (2006). Structured Mesh Generation with Smoothness Controls. *International Journal for Numerical Methods in Fluids,* Vol. 51, Issue 11, (August 20060, pp. 1255–1276, Online ISSN: 1097-0363.

Zhou. Q. (1998) A Simple Grid Generation Method. *International Journal for Numerical Methods in Fluids,* Volume 26, Issue 6, (March 1998), pp. 713–724, Online ISSN: 1097-0363.



Finite Volume Method - Powerful Means of Engineering Design
Edited by PhD. Radostina Petrova

ISBN 978-953-51-0445-2
Hard cover, 370 pages
Publisher InTech
Published online 28, March, 2012
Published in print edition March, 2012

We hope that among these chapters you will find a topic which will raise your interest and engage you to further investigate a problem and build on the presented work. This book could serve either as a textbook or as a practical guide. It includes a wide variety of concepts in FVM, result of the efforts of scientists from all over the world. However, just to help you, all book chapters are systemized in three general groups: New techniques and algorithms in FVM; Solution of particular problems through FVM and Application of FVM in medicine and engineering. This book is for everyone who wants to grow, to improve and to investigate.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

A. Ashrafizadeh, M. Ebrahim and R. Jalalabadi (2012). Alternative Methods for Generating Elliptic Grids in Finite Volume Applications, Finite Volume Method - Powerful Means of Engineering Design, PhD. Radostina Petrova (Ed.), ISBN: 978-953-51-0445-2, InTech, Available from: http://www.intechopen.com/books/finite-volume-method-powerful-means-of-engineering-design/alternative-methods-for-generating-elliptic-grids-in-finite-volume-applications

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821