Structured Grid Generation Via Constraint on Displacement of Internal Nodes

Ali Ashrafizadeh, Razieh Jalalabadi

Abstract—Structured grid generation methods have been used for many years to discretize the solution domain in fluid dynamics simulations. Various differential methods have been employed for this purpose but the traditional method, which transforms a logical grid in the computational domain to a boundary-fitted grid in the physical domain, is employing a set of poison equations. In this paper a new grid generation method is introduced in which the differences between coordinates of boundary nodes of a simple initial grid and final grid are used as the boundary condition of a set of poison equations known as grid generation equations. These Poisson equations are solved on the initial grid to obtain the displacement of nodal coordinates and construct the final grid. Two dimensional grid generation examples are finally presented and the grid qualities are compared with the results of an available differential grid generation method. The underlying ideas are clearly extendible to three-dimensional problems as well.

Index Term—Displacement of Boundary Nods, Initial Grid, Structured Grid Generation, Poison Equations.

I. INTRODUCTION

Numerical solution of equations, which describe fluids flows in practical problems, is a usual approach in Computational Fluids Dynamics (CFD). This approach requires powerful discretization methods to discrete the differential terms in equations and the physical domain. Grid generation strategies are used to discrete physical domains and in the structured or unstructured generated grids [1,2], a set of elements are generated throughout the domain regarding the boundaries shapes. Accuracy and efficiency of numerical solution of equations are strongly affected by the employed grid generation methods [3,4].

In 2D structured grid generation a physical domain is correspondent to a logical domain as shown in figure 1. The intersection of the coordinate lines \( \xi \) and \( \eta \) is known as the grid point \((i,j)\) in the physical domain (figure 1b) [5]. Traditional methods for mapping the unit square onto the physical domain are algebraic and differential grid generation methods. In algebraic grid generation methods the position of nodes in logical domain is changed to their new position in physical domain usually by using an interpolation technique [6]. But in differential grid generation methods, some constraints are used as grid generation equations and when solved, the logical grid is mapped on to the physical grid [7].

A set of widely used differential grid generation equations proposed by Thompson, Thames and Mastin (TTM) [8] is:

\[
\xi_{xx} + \xi_{yy} = P(\xi, \eta)
\]

\[
\eta_{xx} + \eta_{yy} = Q(\xi, \eta)
\]

\[ P \text{ and } Q, \text{ the control functions which are used for better control of the distribution of grid lines, need to be known at all nodal points before the solution. Several methods have been proposed to calculate these functions some of which use some 1-dimensional or multi-dimensional interpolation techniques to calculate these functions in the domain regarding to their values on the boundaries [9,10,11]. Boundary values are calculated using (1a) and (1b) and paving layers in physical domain. Paving layers are two layers of grid generated adjacent to the boundaries using a simple algebraic method as shown in Fig. 2.}

In 1D interpolation techniques, the following one-dimensional formulas can be used to calculate the internal values of the source functions using their corresponding boundary values:

\[
P(\xi, \eta) = C(\eta) P_S(\xi) + (1 - C(\eta)) P_N(\xi)
\]

\[ Q(\xi, \eta) = C(\xi) Q_S(\eta) + (1 - C(\xi)) Q_N(\eta)
\]

In multi-dimensional interpolation techniques, the internal values of the source functions are obtained in the domain using a multi-dimensional set of formulas or through the solution of a Dirichlet boundary value problem.
verning equations are presented

Although some schemes have been proposed to solve (1a) and (1b) directly [11], the usual method to solve these equations is inverting equations analytically, linearizing and then solving them numerically in the logical domain. When inverted, the system of equations to be solved is:

\[
\begin{align*}
g_{22}x_{\xi\xi} - 2g_{12}x_{\eta\xi} + g_{11}x_{\eta\eta} &= -J^2(x_{\xi}P + x_{\eta}Q) \\
g_{22}y_{\xi\xi} - 2g_{12}y_{\eta\xi} + g_{11}y_{\eta\eta} &= -J^2(y_{\xi}P + y_{\eta}Q)
\end{align*}
\]  

(3a)

(3b)

The most usual approach in the orthogonal grid generation, as another robust method in differential grid generation, is a method proposed by Ryskin and Leal [12]. Based on the assumptions of continuity and orthogonality of the coordinate lines and by imposing the orthogonality condition, \( g_{12} = x_{\xi}x_{\eta} + y_{\xi}y_{\eta} = 0 \), the following grid generation equations are obtained:

\[
\begin{align*}
\frac{\partial}{\partial \xi} \left( f \frac{\partial f}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{f} \frac{\partial f}{\partial \eta} \right) &= 0 \\
\frac{\partial}{\partial \xi} \left( \frac{1}{f} \frac{\partial \xi}{\partial \eta} \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{f} \frac{\partial \eta}{\partial \eta} \right) &= 0
\end{align*}
\]  

(4a)

(4b)

\[ f = \frac{\sqrt{g_{22}}}{\sqrt{g_{11}}} = \frac{x_{\xi}^2 + y_{\eta}^2}{\sqrt{x_{\xi}^2 + y_{\eta}^2}} \]  

(5)

Calculation of the nodal values of distortion factor in the domain is a major step in the orthogonal grid generation and is discussed in a number of publications [12, 13, 14]. Most commonly, \( f \) is calculated first at the boundaries and then interpolated into the domain.

The idea underlined in the above two classical methods, is mapping logical domain onto the physical domain by two Laplace equations. These grid generation equations are in fact constraints on the mapping functions \( x(\xi, \eta) \), \( y(\xi, \eta) \) and are solved on the logical grid.

In the new structured grid generation method presented here, two linear differential equations are used to constraint the boundary displacement of an initial grid instead of mapping functions \( x(\xi, \eta) \), \( y(\xi, \eta) \). In the multi-dimensional interpolation technique applied, two Poisson equations are solved to obtain the displacement of nodal coordinates in the physical domain.

In order to study the presented method, the underlying theory is discussed first. Then the description of the method and the governing equations are presented. The paper is followed by a section on some grid generation examples, a discussion on the results and conclusion section.

II. The Proposed Grid Generation Method

A. Grid Generation Procedure

Using simple algebraic grid generation techniques, an initial grid, with the nodal coordinates \((x_0, y_0)\), can be generated for a domain as shown in Fig.3. This initial simple grid is located in the physical \((x, y)\) space. The idea is to find a way to displace the boundaries to conform them to the boundaries of target geometry in physical domain (Fig.3b). Obviously the displacement vectors which connect the initial boundary nodes to the target boundary nodes can be calculated, so a multi-dimensional interpolation technique is employed to find the displacements of the internal nodes. Here two elliptic equations are used as grid generation equations which interpolate the boundary displacements (interpolants) into the domain to generate the physical grid.

By using simple algebraic methods, several initial grids can be generated for any domain. The shape of initial grid can affect the computational cost and the method of solving grid generation equations.

Several possible initial grids generated for physical domains are introduced next and the discretization scheme is presented afterwards.

B. The Initial Grid

The initial grid shown in Fig. 3a is a simple grid with all the simple features of the logical grid, and is an appropriate grid for a Finite Difference Method (FDM) solution to the grid generation equations. Employing the information from geometry, more appropriate grids can be generated. Some of possible initial grids for one sample geometry, here called partially adapted grids, are shown in Fig. 4. These initial grids, respect some of the features of the target geometry, but the distorted quadrilateral cells, may not be appropriate for FDM solution to the governing equations. So, a more powerful discretization method such as FEM or FVM is needed to solve the differential grid generation equations on a partially adapted initial grid.
As shown in Fig. 4, the initial grid can share only the corner points with the target geometry or both the corner points and some boundaries. An FEM or FVM discretization method can be applied to solve the differential grid generation equations for all initial grids. These schemes can also be applicable for some special geometries in which a simple, not adapted initial grid may have distorted quadrilateral cells. Such a domain is shown in Fig. 5.

Using FEM or FVM discretization method to solve the differential grid generation equations, the user can solve the grid generation problem in multi steps. In this approach an increment of boundary displacement is used as boundary condition and an intermediate target domain is generated in each step. Each intermediate grid serves as the initial grid for the next target grid and has distorted quadrilateral cells. In the final step, the target domain is the target geometry.

Several target domains for a simple geometry is shown in Fig. 6. As it will be shown in the result section, using this technique help us to avoid folding in complex geometries and generate a smoother grid.

In this paper an Element-Based Finite Volume method has been used for the solution of grid generation equations.

Algebraic methods can be used as a robust and fast method to generate simple or partially adapted grids. In order to generate simple initial grid, the corners of final geometry is connected with straight lines, but in partially adapted grid, the user choose some of the boundaries from final domain as adapted boundaries and generate other boundaries by connecting the corresponding corners with straight lines. After generating the boundaries of initial domain, the initial grid is generated with Transfinite Interpolation Technique (TFI) [6]. The TFI formulation used here for generating is as follows:

\[
\vec{R}_{i,j} = C_{i,N} \vec{R}_{i,N} + C_{j,M} \vec{R}_{M,j} + C_{i,j} \vec{R}_{i,j} + C_{i+j} \vec{R}_{M,j} + C_{i,j} \vec{R}_{M,j} + \ldots
\]

In this formula the coordinate of node \((i, j)\) is generated by using the coordinates of boundary nodes in a \((M \times N)\) grid.

C. Numerical Solution of Grid generation Equations

The equations used to interpolate the boundary displacements into the domain, in order to obtain the coordinates of internal nodal points, are two poison equations as follows:

\[
V_0(\delta x) = \frac{\partial^2 (\delta x)}{(\partial x_0)^2} + \frac{\partial^2 (\delta x)}{(\partial y_0)^2} = P(\xi, \eta)
\]

\[
V_0(\delta y) = \frac{\partial^2 (\delta y)}{(\partial x_0)^2} + \frac{\partial^2 (\delta y)}{(\partial y_0)^2} = Q(\xi, \eta)
\]

The boundary conditions for these equations are the displacements calculated on the boundaries. So for partially adapted grids (as shown in Fig. 4b, c, d), \(\delta x, \delta y \neq 0\) for boundaries which should be displaced to conform to the target geometry and \(\delta x, \delta y = 0\) for adapted boundaries.

\(P\) and \(Q\) are the functions used for better control of the distribution of grid lines like in (1a) and (1b). These
parameters are calculated using (7a) and (7b) and the paving layers generated on the boundaries in physical domain (Fig. 2), and then interpolated into the domain using TFI technique.

In the Element-Based Finite Volume method used here for the solution of (7a) and (7b), each node in the 2D physical domain is considered in a control volume as shown in Fig. 7 [15]. In this figure the solid lines are the initial grid lines and dashed lines show the control volume around node P.

Defining

\[ \vec{q}^i = \nabla (\hat{\varepsilon}^i), \quad \vec{q}^j = \nabla (\hat{\eta}^j) \]  

The integral of (7a) over the control volume with \( V_i \) and surface \( S_i \), is

\[ \int_{V_i} \nabla \cdot \vec{q}^i \, dV = \int_{S_i} \vec{q}^i \cdot dS = 0 \]  

The surface \( S_i \) consists of a number of panels, and \( i_p \) is an integration point located at the center of each panel.

These linear algebraic sets of equations can be solved by any direct or iterative linear solvers.

III. RESULTS AND DISCUSSION

A number of grid generation examples solved by TTM method and the new elliptic method proposed here are presented in this section. Four geometries are chosen here and a course \((11 \times 11)\) grid is generated in two of them in order to visualize the details of performance of all methods better. Finer \((21 \times 21)\) grids are generated in two other geometries.

The proposed grid generation equations are solved by FDM method in one step and also by FVM solver through multi steps and with two different initial grids.

In order to compare the grid quality, skewness has been chosen as a parameter to measure the quality of grids. Skewness of a cell which is between 0 and 1 measures the deviation from the orthogonality of the coordinate lines.

The computational cost in elliptic grid generation depends mostly on the cost of the solution of the grid generation equations. Both TTM equations and the proposed equations are linear poison equations and so the computational cost is similar.

Fig. 8 shows the discretized boundaries and the \((11 \times 11)\) grid generated by TTM and new proposed equations in first domain. The boundary nodes of physical domain are represented by * signs and initial grids for each solver is shown in Fig. 8d, 8g, 8j. The grid generated by TTM solver and FDM solver are similar. FVM solver with simple initial grid generates a different smoother grid but as shown in Fig. 8k and 8l, by using FVM solver in multi steps with the partially adapted initial grid shown in Fig. 8j, a smoother grid is generated. As there are 10 cells along each coordinate line in the grids, each quality measure diagram shows the relevant skewness quantity for all 100 cells on a three-dimensional plot containing \(10 \times 10\) data points.

Fig. 9 shows the second domain in which TTM method generates a smooth grid but FDM implementation of proposed equations generates a folded grid. The initial grid for FDM solver and simple initial grid for FVM solver are similar for this geometry. As it is shown in Fig. 8g to 8l, the partially adapted initial grid, help FVM solver to generate a smoother grid. Again the skewness parameter is shown for all 100 cells on a three-dimensional plot.

In Fig. 10 a finer \((21 \times 21)\) grid is generated in third sample domain. FVM solver has been used only for multi step solution for this domain and initial grid for FDM and FVM solution are similar for this geometry. As shown in three-dimensional skewness plot which contains \(20 \times 20\) data points in the domain, FVM solver generates a smoother grid compared with FDM and TTM solver.

Fig. 11 shows the \((21 \times 21)\) grids generated around an airfoil. The nodes around the airfoil are distributed uniformly and initial grids are shown around half of the airfoil and FVM solver has been used only for multi step solution. The skewness parameter is again shown for all 400 cells on a three-dimensional plot. As shown in Fig. 11d, h and l, TTM and FDM solver generate smoother grids for this geometry.
Fig. 8. The Physical domain (a), TTM grid (b), quality measures for TTM grid (c), the initial grid for FDM one step solution (d), the grid obtained by the proposed method via FDM (e), quality measures for FDM grid (f), a simple initial grid (g), the grid obtained by the proposed method via FVM for simple initial grid (h), quality measures for FVM solution of equations on simple initial grid (i), a partially adapted initial grid (j), the grid obtained by the proposed method via FVM for partially adapted initial grid (k), quality measures for FVM solution of equations on partially adapted initial grid (l).
Fig. 9. The Physical domain (a), TTM grid (b), quality measures for TTM grid (c), the initial grid for FDM one step solution (d), the grid obtained by the proposed method via FDM (e), quality measures for FDM grid (f), a simple initial grid (g), the grid obtained by the proposed method via FVM for simple initial grid (h), quality measures for FVM solution of equations on simple initial grid (i), a partially adapted initial grid (j), the grid obtained by the proposed method via FVM for partially adapted initial grid (k), quality measures for FVM solution of equations on partially adapted initial grid (l).
Fig. 10. The Physical domain (a), TTM grid (b), a larger view of a section of north boundary (c), quality measures for TTM grid (d), the initial grid for FDM one step solution(e), the grid obtained by the proposed method via FDM (f), a larger view of a section of north boundary (g), quality measures for FDM grid (h), a partially adapted initial grid (i), the grid obtained by the proposed method via FVM for partially adapted initial grid(j), a larger view of a section of north boundary (k), quality measures for FVM solution of equations on partially adapted initial grid (l).
Fig. 11. The Physical domain (a), TTM grid (b), a larger view the grid around airfoil (c), quality measures for TTM grid (d), the initial grid for FDM one step solution(e),the grid obtained by the proposed method via FDM (f), a larger view the grid around airfoil (g), quality measures for FDM grid (h), a partially adapted initial grid (i), the grid obtained by the proposed method via FVM for partially adapted initial grid(j), a larger view the grid around airfoil (k), quality measures for FVM solution of equations on partially adapted initial grid (l).

IV. CONCLUSION

In this paper, after reviewing the two most commonly used classical methods of elliptic grid generation, a new elliptic grid generation method was proposed. In this method the general idea was similar to the previous methods; solving a multi-dimensional interpolation problem, but the interpolants were different parameters. In the simple differential method presented, an initial grid was deformed to conform to the given physical boundaries and the differences between coordinates of boundary nodes of an initial grid and the final grid are used as interpolants. Two poison equations are introduced as grid generation equations and the boundary conditions are the interpolants discussed. FDM scheme in one step and FVM scheme in both one step and multi steps have been used to solve the equations and the skewness diagram is presented to study the smoothness of final grids better. FVM solver generates smoother grids with better quality especially in complex geometries and with a partially adapted grid as an initial grid. As a result, it can be mentioned that the proposed method solve grid generation problem from a different viewpoint and in general this method is computationally similar to TTM method while both methods provide grids with comparable qualities.

REFERENCES


