Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2008) 7977–7997

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Boundary treatment for 2D elliptic mesh generation in complex geometries

Yaoxin Zhang [*], Yafei Jia, Sam S.Y. Wang, H.C. Chan

National Center for Computational Hydroscience and Engineering, The University of Mississippi, 102 Carrier Hall, University, MS 38677, United States

Received 14 November 2007; received in revised form 18 May 2008; accepted 19 May 2008
Available online 25 May 2008

Abstract

This paper presents a boundary treatment method for 2D elliptic mesh generation in complex geometries. Corresponding to Neumann–Dirichlet boundary conditions (sliding boundary conditions), the proposed method aims at achieving orthogonal and smooth nodal distribution along irregular boundaries. In this method, a three-lined auxiliary mesh is constructed which is composed of the boundary line, an auxiliary line generated on its inner side, and the reflection line of this auxiliary line. The movements of the boundary nodes are determined by solving this auxiliary mesh. The boundary nodes are further corrected and adjusted by a second-order parabolic interpolation method and a weighting parameter depending on the curvature of the boundary. The proposed method was demonstrated through examples, and applied to field cases with complex geometries. It has been shown that this method is stable and effective in producing high quality meshes.

Keywords: Neumann–Dirichlet; Mesh generation; Complex geometry

1. Introduction

To achieve a mapping between the physical coordinates and the computational coordinates, elliptic mesh generation solves a set of partial differential equations (PDE) in a bounded domain with initial conditions and boundary conditions. In this initial-boundary value problem, three kinds of boundary conditions are available: (1) Dirichlet boundary conditions: the boundary nodal distribution is specified and fixed; (2) Neumann–Dirichlet boundary conditions: the boundary nodes move along boundaries (Dirichlet) to satisfy the orthogonal condition (Neumann) and, (3) Neumann boundary conditions. Since purely Neumann boundary conditions cannot generate boundary-fitted meshes [4], the first two types of boundary conditions are widely used in practice.

The TTM generation system developed by Thompson et al. [9], the RL generation system proposed by Ryskin and Leal [6], and the conformal mapping system [10] are the most famous and widely used elliptic mesh generation systems. When generating orthogonal mesh in geometrically complex domains with weak

[*] Corresponding author. Tel.: +1 662 915 3960; fax: +1 662 915 7796.
E-mail address: yzhang@ncche.olemiss.edu (Y. Zhang).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.05.008

constraints (specified boundary distribution) corresponding to the Dirichlet boundary conditions, if inappropriate nodal distribution is imposed along boundaries, the mesh quality cannot be guaranteed using the above systems. For examples, Eça [2], and Zhang et al. [11–13] pointed out that serious skewness or distortion and overlapping may occur using the RL system due to the local strong orthogonal conditions and, the conformal mapping may produce folded meshes due to the strong smoothness constraint—equal scale factors in all directions. According to the numerical tests of Eça [2], favorable boundary nodal distribution can greatly improve mesh quality. However, it is difficult to predict such an appropriate initial nodal distribution (usually non-uniform) along boundaries. Instead, in practice the most direct and natural boundary nodal distribution is equal-spaced or uniform which may work well in simple domains but not in complex ones. In order to control the mesh near the boundaries with Dirichlet boundary conditions, Sorenson [7] used a second-order elliptic equations with inhomogeneous source terms (control functions) to simulate the Neumann boundary conditions. Thomas and Middlecoff [8] also derived a set of orthogonal control functions which are evaluated at the boundaries and interpolated for the interior nodes so that orthogonality at the boundaries can be achieved with boundary nodes fixed.

Compared with the rigidness of the Dirichlet boundary conditions, the Neumann–Dirichlet boundary conditions is more flexible and can significantly improve mesh quality if appropriately applied, otherwise it may even cause failures of mesh generation. For examples, Duraiswami and Prosperetti [1] successfully employed this type of boundary conditions in the RL system; Jeng and Chen [3] used the least-square method to derive a functional defined by the deviation from Beltrami equation so that the boundary nodes can slide along boundaries to minimize this functional and, Eça [2] also tested the Neumann–Dirichlet boundary conditions in the RL system and observed numerical divergence when evaluating the distortion function using its original definition. Although the Neumann–Dirichlet boundary conditions enjoyed some successes in certain applications, difficulties still remain in applying them to highly irregular boundaries large amount of which exist in fields in computational fluids dynamics (CFD) analysis. One difficulty lies in the control of the nodal movements along boundaries. To prevent boundary nodes from overtaking or collapsing to its neighbors to satisfy the orthogonal condition, "guards" must be used to limit the nodal movements [5]. It is difficult to define such a "guard" appropriate for each boundary node. Another difficulty is to keep the shape of the boundaries unchanged as much as possible. Thus, special treatments, such as curvature corrections [3], need to be used.

In this paper, a method of applying the Neumann–Dirichlet boundary conditions to complex geometries has been developed. This method is capable of producing orthogonal and smooth nodal distribution along boundaries, and an auxiliary mesh based on the boundary line and its neighboring mesh line is necessary. The nodal movements along boundaries are controlled by solving this auxiliary mesh using an elliptic mesh generation system. In this method, a second-order boundary curvature correction method is used to minimize the curvature changes caused by the movements of the boundary nodes, and further adjustments of the boundary nodes are achieved by a weighting parameter related to the curvature of the boundaries. Some academic examples and applications for natural rivers are used to illustrate the proposed method.

2. Current study

2.1. Conventional method

In 2D elliptic mesh generation, a domain is bounded by at least four boundaries and a boundary is composed of at least three boundary nodes and both starts and ends with a corner. Two types of boundary conditions, namely, the Dirichlet boundary condition and the Neumann–Dirichlet boundary condition are generally used. Boundary nodes are fixed and cannot move in the Dirichlet boundary conditions; while in the Neumann–Dirichlet boundary condition, boundary nodes can slide along boundaries (Dirichlet) to satisfy the orthogonal conditions (Neumann), so it is also called the sliding boundary condition. The Dirichlet boundary condition is simple and easy to use, but it is rigid and the mesh quality depends largely on the initial nodal distribution along boundaries. Numerical tests by Eça [2] and Zhang et al. [11–13] showed that inappropriate nodal distribution may cause serious skewness interiorly, although orthogonality at the boundaries can be achieved. Compared to the Dirichlet boundary conditions, the Neumann–Dirichlet boundary condition is more difficult to use especially for cases with complex boundaries.

Considering a mapping between the physical coordinates ($x^i(\equiv x, y)$, $i = 1,2$) and the computational coordinates (ξ^i ($\equiv \xi, \eta$), $i = 1,2$), a metric tensor g_{ij} can be defined as follows:

$$g = \begin{Bmatrix} (x_\xi^2 + y_\xi^2) & (x_\xi x_\eta + y_\xi y_\eta) \\ (x_\xi x_\eta + y_\xi y_\eta) & (x_\eta^2 + y_\eta^2) \end{Bmatrix}, \tag{1}$$

where $x_\xi = \partial x/\partial\xi$ and so forth.

For one boundary node (x_i^b, y_i^b), as shown in Fig. 1, the movements of this node $(\Delta x_i^b, \Delta y_i^b)$ to satisfy the orthogonal condition ($g_{12} = g_{21} = 0$) (Neumann) can be expressed in central difference discretization as follows:

$$\Delta x_i^b = \frac{x_\xi^b \cdot x_\eta^b + y_\xi^b \cdot y_\eta^b}{(x_\eta^b)^2 + (y_\eta^b)^2} \cdot x_\eta^b \cdot r_b, \tag{2}$$

$$\Delta y_i^b = \frac{x_\xi^b \cdot x_\eta^b + y_\xi^b \cdot y_\eta^b}{(x_\eta^b)^2 + (y_\eta^b)^2} \cdot y_\eta^b \cdot r_b, \tag{3}$$

$$x_\xi^b = x_i^b - x_i^n, \tag{4a}$$

$$y_\xi^b = y_i^b - y_i^n, \tag{4b}$$

$$x_\eta^b = (x_{i+1}^b - x_{i-1}^b)/2, \tag{4c}$$

$$y_\eta^b = (y_{i+1}^b - y_{i-1}^b)/2, \tag{4d}$$

where the superscripts “b” and “n” denote the boundary node and the neighboring interior node, respectively and, r_b (0–1) is an empirical parameter to control the movements.

In Eqs. (2) and (3), r_b is usually in a range of $[0, 0.5]$ and used to prevent the boundary node from moving out of the boundary segment it belongs to [5]. Although varied distribution of r_b is more reasonable and desired, constant value is often used in practice, because it is difficult to specify an appropriate distribution of r_b for all boundary nodes.

To satisfy the Dirichlet boundary condition (sliding along boundaries), a certain second-order or high-order interpolation is usually employed to further adjust the movements of the boundary node (called boundary curvature correction) to make sure the boundary nodes moves only “along” boundaries.

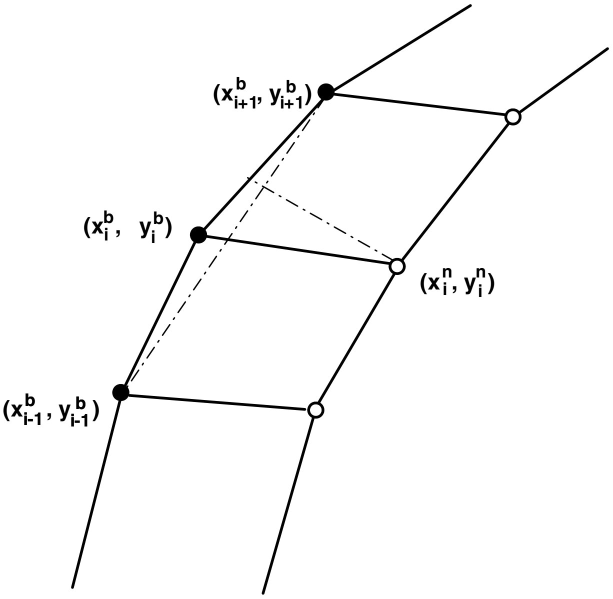


Fig. 1. Conventional method.

2.2. Auxiliary mesh

For the conventional method [5], the movements of the boundary nodes are driven only by the orthogonal conditions at the boundary. The success of this method depends largely on the appropriate selection of empirical parameter r_b. Too large value of r_b may lead to collapse of boundary nodes, while too small value may result in insufficient movements of the boundary nodes.

In general, an elliptic mesh generation system is able to produce smooth and orthogonal interior mesh with the Dirichlet boundary conditions. This favorable characteristic inspires an idea that the movements of the boundary nodes are controlled by an elliptic generation system. For this purpose, an auxiliary mesh needs to be constructed for the selected boundary, and it must satisfy the following two basic requirements:

(1) the selected boundary serves as an interior mesh line of it and,
(2) it should contain information from the interior domain.

In current study, this auxiliary mesh is constructed based on the boundary line and its immediate interior neighboring mesh line. The size of the auxiliary mesh is $3 \times np$ (np is the number of boundary nodes).

Let $F_b(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np})$ denote a selected boundary and $F_{nb}(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np})$ denote its interior neighboring mesh line, respectively (see Fig. 2(a)), then the auxiliary mesh for this boundary can be constructed as follows:

An auxiliary mesh line F_1 is directly obtained from the boundary line F_b and its neighboring line F_{nb}:

$$F_1(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np}) = F_{nb}(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np}) + r_m \cdot [F_b(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np}) - F_{nb}(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np})], \tag{5}$$

where r_m is a parameter in a range of $[0, 1]$.

Another auxiliary line F_3 outside F_b is necessary for constructing the auxiliary mesh which can be determined in three steps:

(a) Starting from a node except the first and the last on the first auxiliary line F_1, a line is drawn perpendicularly to intersect the boundary line F_b so that an intermediate line $F_{\text{in}}(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np})$ composed of the interception points can be obtained.
(b) The two ending nodes (x_1, y_1) and (x_{np}, y_{np}) of F_1, as shown in Fig. 2(b), move along F_1 to (x'_1, y'_1) and (x'_{np}, y'_{np}) where a perpendicular line can be drawn to intersect F_b.
(c) The auxiliary mesh line F_3 is a reflection of the first auxiliary line F_1 with equal distance to the intermediate line F_{in}:

$$F_3(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np}) = F_1(x'_1, y'_1, x_2, y_2, \ldots, x'_{np}, y'_{np}) + 2 \cdot [F_{\text{in}}(x_1, y_1, x_2, y_2, \ldots, x_{np}, y_{np}) - F_1(x'_1, y'_1, x_2, y_2, \ldots, x'_{np}, y'_{np})]. \tag{6}$$

In [5], a reflected neighboring mesh line with equal distance across the boundary was proposed to evaluate the orthogonal control functions at the boundary developed by Thomas and Middlecoff [8]. Eq. (6) describes a different reflection: the first mesh line F_1 and the reflected line (the third mesh line) F_3 are in symmetry to the boundary line F_b, the auxiliary mesh therefore can be solved with the Dirichlet boundary conditions to achieve orthogonality along the boundary line F_b.

The width of the auxiliary mesh is controlled by the parameter r_m in Eq. (5). The smaller it is, the narrower would be the auxiliary mesh, and vice versa. With r_m close to 0, the three mesh lines will collapse into one line and the boundary nodes will become frozen. Although r_m is a scale parameter with similar function to r_b defined in Eqs. (2) and (3), in the conventional method r_b can directly control the movements of the boundary nodes and is evaluated completely empirically, while in the current study, r_m has indirect influences on the nodal movements along boundaries, and the evaluation of r_m can be guided by the following rule: an appropriate value of the parameter r_m will produce a valid auxiliary mesh without mesh lines across each other. An invalid auxiliary mesh indicates a big value of the parameter r_m which must be reduced. In the current study, according to numerical tests, a value of 0.5 is found optimal for r_m.

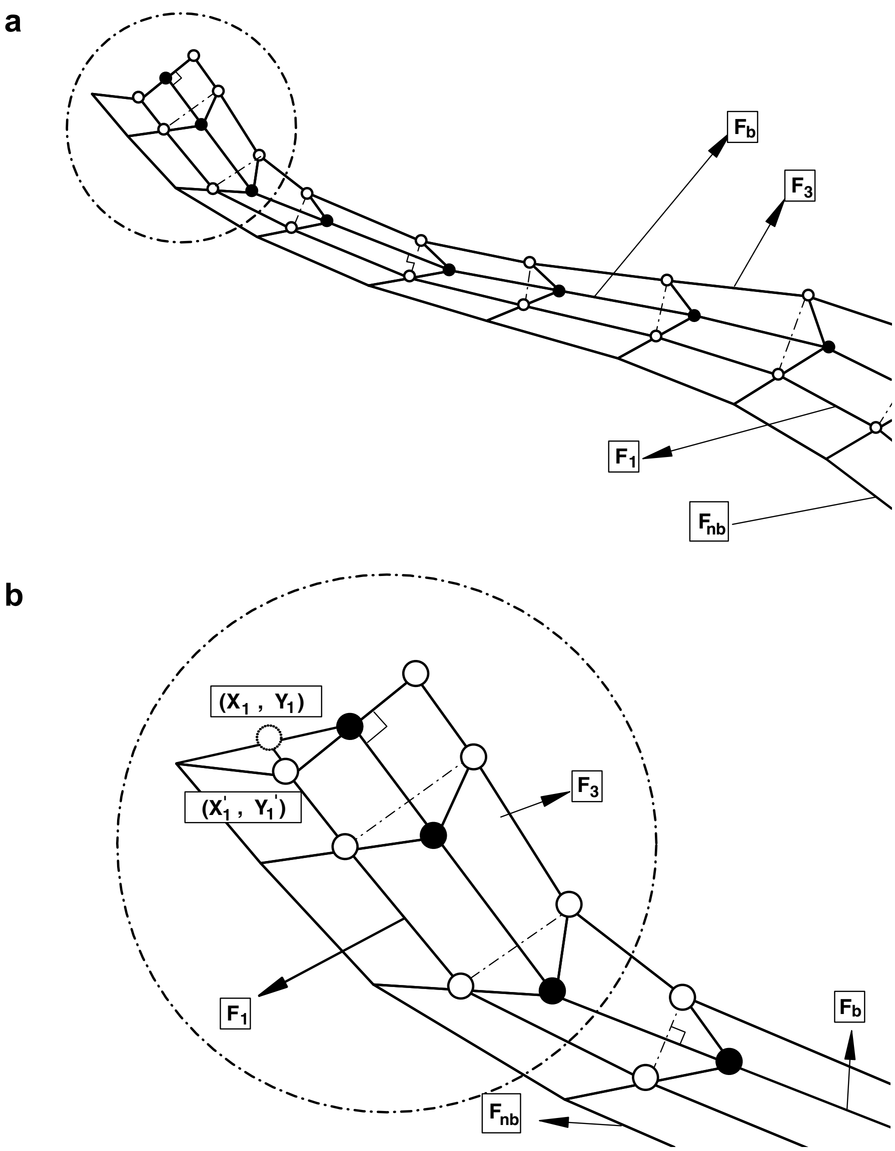


Fig. 2. Auxiliary mesh.

The auxiliary mesh has only three mesh lines and the boundary conditions (F_1 and F_3) have direct and strong influences on the interior mesh (F_b), so the convergence process normally will be fast. Secondly, in the outer iteration cycle, it is not necessary to have a converged solution since it will be used next as boundary condition for the inner cycle. Therefore, in current study, one iteration cycle is recommended to solve the auxiliary mesh. With this approach, both the interior and the boundaries are solved by the elliptic generation system.

2.3. Boundary curvature correction

The movements of the boundary nodes must be restricted along the exact boundaries to preserve the original boundary value problem. The boundary can be represented by either local or global polynomial functions, such as cubic Spline and B-Spline. For complicated boundary which may not satisfy the continuous conditions, special cares must be taken for the cubic Spline line, while the B-Spline does not pass through all the shape control points of a boundary. Compared to the global functions, the local polynomial function is more stable, flexible and easy to use. In current study, a second-order polynomial is employed to fit the local boundary curve and correct the boundary curvature.

For one typical boundary node (x_b, y_b), the new location of this node obtained by solving the auxiliary mesh is denoted by (x'_b, y'_b) in which only one coordinate is independent because it should be on the local boundary curve. In other words, one coordinate needs to be adjusted while the other one is fixed. Assuming (x_k, y_k), (x_{k+1}, y_{k+1}), and (x_{k+2}, y_{k+2}) are three closest points on the original boundary curve, then one can obtain the corrected x''_b and y''_b as follows:

$$x''_b = \sum_{i=k}^{k+2} \left(\prod_{\substack{j=k \\ j \neq i}}^{k+2} \frac{y'_b - y_j}{y_i - y_j} \right) \cdot x_i \quad \text{if } |x_{k+2} - x_k| \leqslant |y_{k+2} - y_k|, \tag{7}$$

$$y''_b = \sum_{i=k}^{k+2} \left(\prod_{\substack{j=k \\ j \neq i}}^{k+2} \frac{x'_b - x_j}{x_i - x_j} \right) \cdot y_i \quad \text{if } |x_{k+2} - x_k| \geqslant |y_{k+2} - y_k|. \tag{8}$$

Eqs. (7) and (8) calculate the differences of y and x coordinates for the three closest points. In case two points are on one vertical or horizontal line, these equations will not be valid, and the users need to switch between them. Therefore, either Eq. (7) or (8) is used to adjust the boundary node to make it fit the original boundary curve.

In the case that the original boundary curve can be represented by a known function, this function will be used to correct the location of the boundary node. Note that no matter how accurate the curvature correction is or even the exact boundary curve function is known, there will be changes for the boundary curves when applying the sliding boundary condition. In other words, the curvature changes are inevitable, but the curvature correction can help to minimize the changes to some extent, and it can make the new locations of the boundary nodes as close to the original curve as possible, which is very important in mesh generation.

2.4. Weighted boundary

Different boundaries with different curvatures and shapes have different influences on the mesh quality. In general, boundaries with simpler shapes (low curvatures) will have less influence than those with complex shapes (high curvatures). In other words, boundaries with complex shapes will cause more difficulties for mesh generation and thus they demand higher priorities to be treated.

On the other hand, with the boundary nodes moving, the boundary curve, represented by finite number of the piece-wise line segments, will change accordingly even with curvature corrections which can only minimize the changes to some extent. When the mesh is not fine enough, this drawback will become more significant. So in principle, the Neumann–Dirichlet boundary conditions should be only applied to where it is necessary. Based on this point of view, in the current study a weighting parameter is proposed to each boundary according to its curvature:

$$w_b = \frac{C_b}{\max(C_b)}, \tag{9}$$

where C_b represents the curvature of boundary F_b and is defined as follows:

$$C_b = \sum_{1}^{np} \theta_i, \tag{10a}$$

$$\theta_i = \beta_i / \pi, \tag{10b}$$

where β_i is the angle at the node i between its two neighboring nodes on boundary F_b.

As can be seen in Eqs. (9) and (10), the boundary with maximum curvature will have the highest weight 1, and the boundary without curvature (straight line) will be assigned a weight of 0. In case that all boundaries are straight lines, the above equations become invalid.

Eq. (9) is used for final tuning of the movements of the boundary nodes. For one typical boundary node (x_b, y_b), its new location will be

$$x_b^{\text{new}} = x_b \cdot (1 - w_b) + x_b'' \cdot w_b, \tag{11a}$$

$$y_b^{\text{new}} = y_b \cdot (1 - w_b) + y_b'' \cdot w_b, \tag{11b}$$

where (x_b'', y_b'') is the location after curvature correction.

From Eq. (11), obviously the boundary nodes on straight line will be kept fixed and the Dirichlet boundary condition will be applied to this boundary.

2.5. Measures of boundary curvature preservation

The measurements of the boundary curvature preservation can be obtained by comparing the new boundary curve F_{new} with the original boundary curve F_b. In this study, instead of directly comparing F_{new} with F_b, C_{new} and $C_b(p_{b,0}, p_{b,1}, \ldots, p_{b,n})$, the new representations of F_{new} and F_b using B-Spline with the same degree (=3) and the same number of points n, are used for comparisons. To make a fair comparisons, a set of points $(p_{\text{new},0}, p_{\text{new},1}, \ldots, p_{\text{new},n})$are interpolated from C_{new} so that each pair of points ($p_{\text{new},i}$ and $p_{b,i}$) satisfies either of the following conditions: (1) $x_{\text{new},i} = x_{b,i}$ and, (2) $y_{\text{new},i} = y_{b,i}$.

Three metrics are defined to compare C_{new} and C_b using the abovementioned two sets of points: maximum curve difference (MCD), maximum angle difference (MAD), and maximum average angle difference (AAD):

$$\text{MCD} = \max |p_{\text{new},i} - p_{b,i}| / L_b, \tag{12a}$$

$$\text{MAD} = \max |\delta_{\text{new},i} - \delta_{b,i}|, \tag{12b}$$

$$\text{AAD} = \max \left(\frac{1}{n-2} \sum_{2}^{n-1} |\delta_{\text{new},i} - \delta_{b,i}| \right), \tag{12c}$$

where $|p_{\text{new},i} - p_{b,i}|$ represents the distance between $p_{\text{new},i}$ and $p_{b,i}$; L_b is the total length of the boundary and, δ_i (in degree) is the angle at p_i between p_{i-1} and p_{i+1}.

2.6. Elliptic mesh generation system

For a valid auxiliary mesh, the movements of boundary nodes on F_b are driven and controlled by an elliptic mesh generation system. Generally speaking, different elliptic generation systems will produce different meshes in the same domain with the same boundary conditions. For examples, the RL system is more effective in orthogonality, while the conformal mapping yields smoother meshes. Note that the proposed algorithm for boundary treatments is generally suitable for other elliptic mesh generation systems. In current study, an improved RL system with smoothness controls proposed by Zhang et al. [11] is selected due to its stable performance in complex geometries.

This improved RL system can be described as follows:

$$\frac{\partial}{\partial \xi}\left(h_\xi^\alpha \cdot f \frac{\partial x}{\partial \xi} \right) + \frac{\partial}{\partial \eta}\left(h_\eta^\alpha \cdot \frac{1}{f} \frac{\partial x}{\partial \eta} \right) = 0, \tag{13a}$$

$$\frac{\partial}{\partial \xi}\left(h_\xi^\alpha \cdot f \frac{\partial y}{\partial \xi} \right) + \frac{\partial}{\partial \eta}\left(h_\eta^\alpha \cdot \frac{1}{f} \frac{\partial y}{\partial \eta} \right) = 0, \tag{13b}$$

where the distortion factor f is defined as the ratio of the scale factors in ξ- and η-directions, i.e.,

$$f = \frac{h_\eta}{h_\xi} = \left(\frac{x_\eta^2 + y_\eta^2}{x_\xi^2 + y_\xi^2} \right)^{1/2} \tag{14}$$

and $h_\xi = g_{11}^{1/2}$, $h_\eta = g_{22}^{1/2}$.

One can obtain the discretization of Eq. (13) at one typical mesh node (i,j) using central difference scheme:

$$F_{i,j}x_{i,j} = f_{i+1/2,j}c_{i+1,j}x_{i+1,j} + f_{i-1/2,j}c_{i-1,j}x_{i-1,j} + \frac{c_{i,j+1}}{f_{i,j+1/2}}x_{i,j+1} + \frac{c_{i,j-1}}{f_{i,j-1/2}}x_{i,j-1}, \tag{15a}$$

$$F_{i,j}y_{i,j} = f_{i+1/2,j}c_{i+1,j}y_{i+1,j} + f_{i-1/2,j}c_{i-1,j}y_{i-1,j} + \frac{c_{i,j+1}}{f_{i,j+1/2}}y_{i,j+1} + \frac{c_{i,j-1}}{f_{i,j-1/2}}y_{i,j-1}, \tag{15b}$$

where $F_{i,j} = f_{i+1/2,j}c_{i+1,j} + f_{i-1/2,j}c_{i-1,j} + \frac{c_{i,j+1}}{f_{i,j+1/2}} + \frac{c_{i,j-1}}{f_{i,j-1/2}}$ and $c_{i,j}$ is the contribution factor of mesh node (*i*,*j*) defined as follows:

$$c_{i+1,j} = [(h_\xi)_{i+1/2,j}]^\alpha = \left(\sqrt{(x_{i,j} - x_{i+1,j})^2 + (y_{i,j} - y_{i+1,j})^2}\right)^\alpha, \tag{16a}$$

$$c_{i-1,j} = [(h_\xi)_{i-1/2,j}]^\alpha = \left(\sqrt{(x_{i,j} - x_{i-1,j})^2 + (y_{i,j} - y_{i-1,j})^2}\right)^\alpha, \tag{16b}$$

$$c_{i,j+1} = [(h_\xi)_{i,j+1/2}]^\alpha = \left(\sqrt{(x_{i,j} - x_{i,j+1})^2 + (y_{i,j} - y_{i,j+1})^2}\right)^\alpha, \tag{16c}$$

$$c_{i,j-1} = [(h_\xi)_{i,j-1/2}]^\alpha = \left(\sqrt{(x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2}\right)^\alpha, \tag{16d}$$

where $c_{i+1,j}$ is the distance between point (*i*,*j*) and point $(i+1,j)$ and $\alpha \in [0,1]$ is a adjustable parameter.

When $\alpha \to 0$, this improved RL system (Eq. (13)) approaches to the original RL system developed by Ryskin and Leal [6]:

$$\frac{\partial}{\partial \xi}\left(f\frac{\partial x}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{f}\frac{\partial x}{\partial \eta}\right) = 0, \tag{17a}$$

$$\frac{\partial}{\partial \xi}\left(f\frac{\partial y}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{f}\frac{\partial y}{\partial \eta}\right) = 0. \tag{17b}$$

2.7. Solution procedure

The linearized system defined by Eq. (15) is solved using an iterative algorithm. In this study, the following algorithm is proposed for boundary treatments in complex geometries using the improved RL system defined in Eq. (13):

1. Define the boundaries of the domain and use an algebraic method to generate an initial mesh.
2. Identify each boundary and calculate its curvatures to assign the boundary weights w_b using Eq. (9). Note that the original boundaries identified in this step will be used for boundary curvature corrections later.
3. Specify the parameter α defined in Eq. (16).
4. Calculate the distortion function *f* from Eq. (14).
5. Calculate the contribution factors using Eq. (16).
6. Solve Eq. (15) using fixed *f* obtained from step 4 and the contribution factors from step 5 with the Dirichlet boundary condition for all boundaries.
7. Construct a valid auxiliary mesh for each boundary (see previous section for details) and solve this mesh using the improved RL system with one iteration cycle to adjust the locations of the boundary nodes.
8. Perform boundary curvature corrections using Eqs. (7) or (8).
9. Update the location of the boundary nodes using Eq. (11).
10. Update the mesh and check if the convergence condition is satisfied. If not, repeat steps from 4 to 9.

Two convergence criterions are used and the satisfaction of either one will stop the computation. These two criterions are defined as the maximum difference between the grid coordinates and the maximum relative difference of the distortion function *f* between consecutive iterations, respectively:

$$\max\left(\sqrt{(x^n_{i,j}-x^{n-1}_{i,j})^2+(y^n_{i,j}-y^{n-1}_{i,j})^2}\right)<10^{-6}, \tag{18}$$

$$\max\left(\frac{f^n-f^{n-1}}{f^n}\right)<10^{-6}, \tag{19}$$

where n is the iteration number.

Obviously, the construction and the solving process of the auxiliary mesh need more computation and will slow down the whole computation, which is a drawback of the present algorithm.

3. Examples

Four academic examples widely used in the literatures ([1–2] and [11–13]) are selected to illustrate the present method and these four domains are defined as follows:

(1) Domain A (with concave boundary) is bounded by $x=0$, $x=1$, $y=0$, and $y=0.75+0.25\sin(\pi(0.5+2x))$.
(2) Domain B is a unit square with one half-circle on each side.
(3) Domain C is bounded by $x=0$, $y=0$, $y=1$, and $x=\frac{1}{2}+\frac{1}{6}\cos(\pi y)$.
(4) Domain D is bounded by two-half-circles and x-axis. The radius of the small circle is one-third of that of the big one.

For these four domains, initial uniform nodal distribution is applied to all boundaries, and the exact curve functions will be used for sliding boundary conditions. Two standard academic criterions, namely, orthogonality and smoothness, are used to evaluate the mesh quality, which are represented by the following indicators, maximum deviation orthogonality (MDO), averaged deviation from orthogonality (ADO), maximum grid aspect ratio (MAR), and averaged grid aspect ratio (AAR):

$$\mathrm{MDO}=\max(\theta_{i,j}-90), \tag{20a}$$

$$\mathrm{ADO}=\frac{1}{(N_i-2)}\frac{1}{(N_j-2)}\sum_{2}^{N_i-1}\sum_{2}^{N_j-1}\max(\theta_{i,j}-90), \tag{20b}$$

$$\mathrm{MAR}=\max\left[\max\left(f_{i,j},\frac{1}{f_{i,j}}\right)\right], \tag{21a}$$

$$\mathrm{AAR}=\frac{1}{(N_i-2)}\frac{1}{(N_j-2)}\sum_{2}^{N_i-1}\sum_{2}^{N_j-1}\max\left[\max\left(f_{i,j},\frac{1}{f_{i,j}}\right)\right], \tag{21b}$$

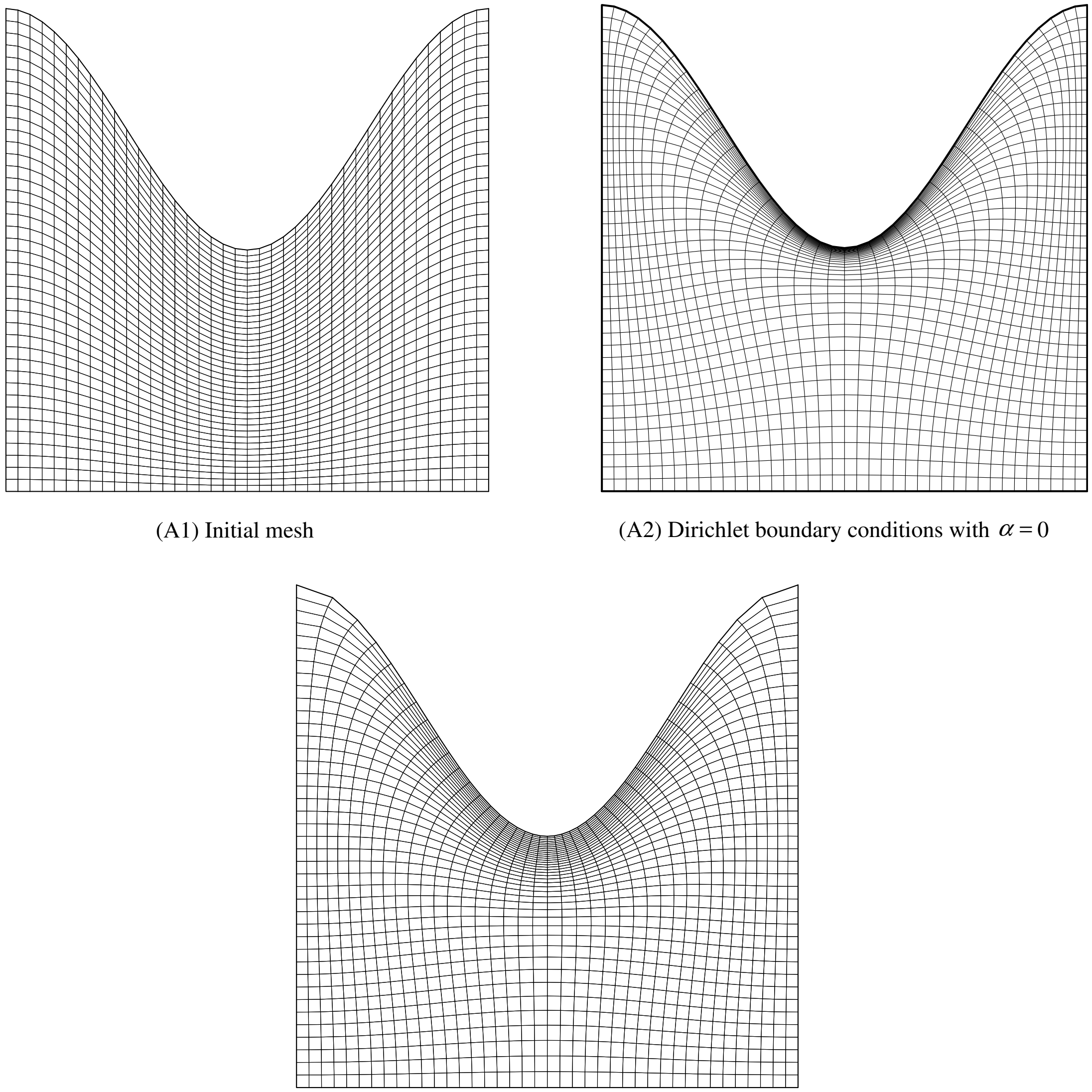
where N_i and N_j are the maximum number of mesh lines in ξ- and η-directions, respectively; and θ is defined as

$$\theta_{i,j}=\arccos\left(\frac{g_{12}}{h_\xi h_\eta}\right)_{i,j}. \tag{22}$$

3.1. Symmetric domains

Figs. 3 and 4 show the meshes generated using different boundary conditions, and Table 1 summarizes the mesh evaluations. According to the weights of the boundaries, the sliding boundary conditions were applied to the top concaved boundary in domain A and all four boundaries in domain B, respectively.

As can be seen, in both domains, both boundary conditions successfully produced orthogonal meshes. As for mesh smoothness, with the Dirichlet boundary conditions, due to the local strong orthogonal conditions [11–13], meshes (A2 and B2) squeeze at the concave boundary in domain A and at the center of domain B and, with the Neumann–Dirichlet boundary conditions, using the present method, the mesh quality (A3) has been improved significantly both in orthogonality and smoothness in domain A; while in domain B, mesh (B3) still squeeze to the four singular corners, although interior mesh was improved greatly. The four singular corners



(A1) Initial mesh

(A2) Dirichlet boundary conditions with $\alpha = 0$

(A3) Sliding boundary conditions with $\alpha = 0$ and $r_m = 0.5$

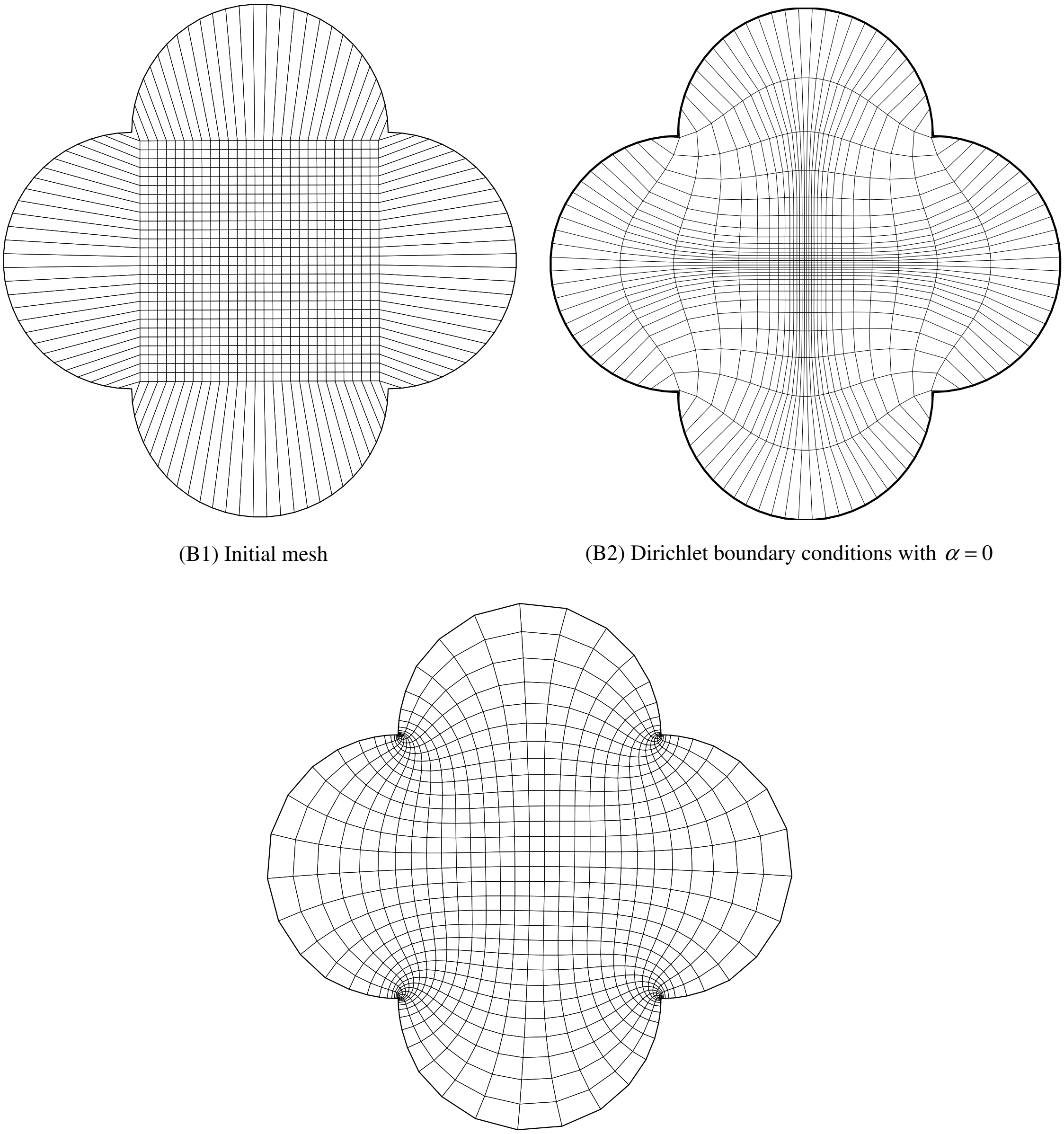
Fig. 3. Meshes in domain A.

in domain B resulted in grid around them with high deviation from orthogonality (MDO) for both boundary conditions.

For the resulting meshes in A3 and B3, boundary curvature differences can be observed both quantitatively and visually, but the boundary nodes are guaranteed to be on the exact original curves. As one expected, in both cases the maximum curvature difference occurred at the coarse part of the boundaries (around two corners of top boundary in A3 and the crests of half-circles in B3, respectively). If the selected boundary with sliding boundary conditions corresponds to a physical boundary, i.e., a wing of an aircraft, the above curvature differences may not be tolerable and hence the resulting meshes (A3 and B3) cannot be accepted [5].

3.2. Asymmetric domains

As shown in Figs. 5, 6 and Table 2, similar observations on the resulting meshes in domains C and D can be obtained. With Eq. (15), the right curved boundary of domain C and the two half-circled boundaries of domain D were selected to use the Neumann–Dirichlet boundary conditions.



(B1) Initial mesh

(B2) Dirichlet boundary conditions with $\alpha = 0$

(B3) Sliding boundary conditions with $\alpha = 0$ and $r_m = 0.5$

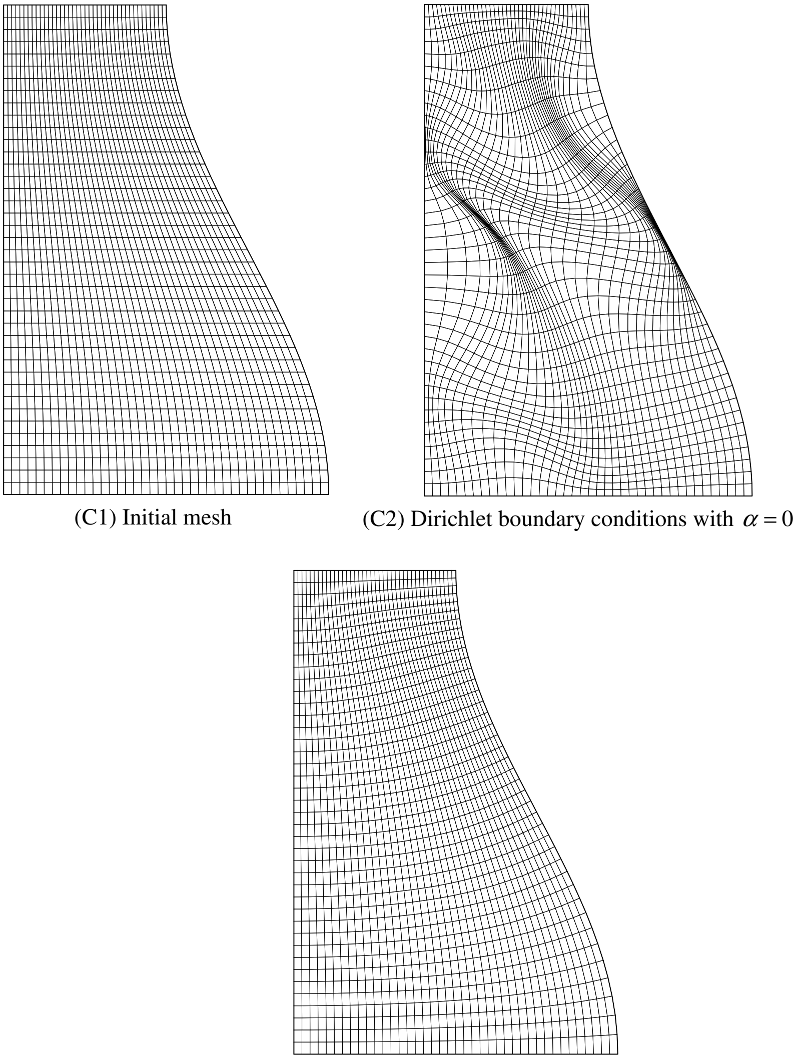
Fig. 4. Meshes in domain B.

For both domains, orthogonality was achieved by both boundary conditions. As can be seen, with the Dirichlet boundary conditions, mesh distortion or skewness (C2 and D2) tends to occur in the asymmetric domains in order to satisfy the local orthogonal conditions. However, when using the present method, the above phenomena disappeared and both mesh orthogonality and smoothness were dramatically improved (C3 and D3).

Similarly, with the Neumann–Dirichlet boundary conditions, slight curvature differences presented in both cases. The maximum difference appeared at the bottom part of the right curved boundary in C3 and around the right corner of the top half-circled boundary in D3.

Table 1
Evaluation of meshes in domains A and B

Case	Size	ADO	MDO	AAR	MAR	α	r_m	MCD	MAD	AAD
A2	41 × 41	0.21	4.54	4.43	33.9	0	–	–	–	–
A3	41 × 41	0.09	1.57	2.72	8.42	0	0.5	0.007	2.43	0.190
A4	41 × 41	0.11	1.32	3.42	19.47	0	0.05	0.006	2.43	0.189
A5	41 × 41	0.09	1.23	2.99	13.91	0	0.1	0.007	2.43	0.191
A6	41 × 41	0.09	1.37	2.74	8.74	0	0.25	0.007	2.43	0.192
A7	41 × 41	0.13	1.87	2.86	11.35	0	0.75	0.006	2.43	0.185
A8	41 × 41	0.15	1.84	2.79	12.43	0	1.0	0.005	2.43	0.181
B2	30 × 30	0.13	24.3	4.22	13.0	0	–	–	–	–
B3	30 × 30	0.8	30.2	1.24	4.34	0	0.5	0.01	1.38	0.41

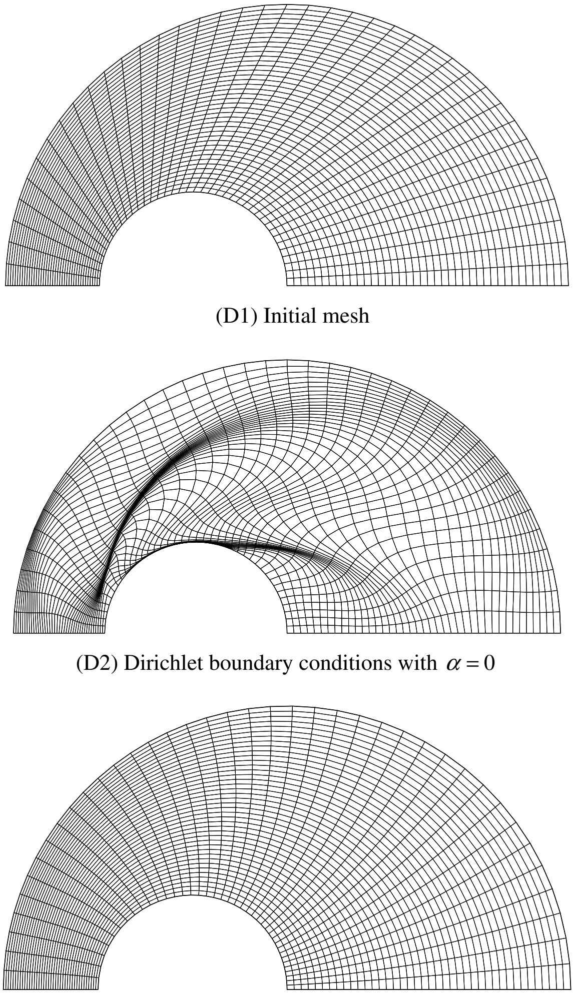


(C1) Initial mesh

(C2) Dirichlet boundary conditions with $\alpha = 0$

(C3) Sliding boundary conditions with $\alpha = 0$ and $r_m = 0.5$

Fig. 5. Meshes in domain C.



(D1) Initial mesh

(D2) Dirichlet boundary conditions with $\alpha = 0$

(D3) Sliding boundary conditions with $\alpha = 0$ and $r_m = 0.5$

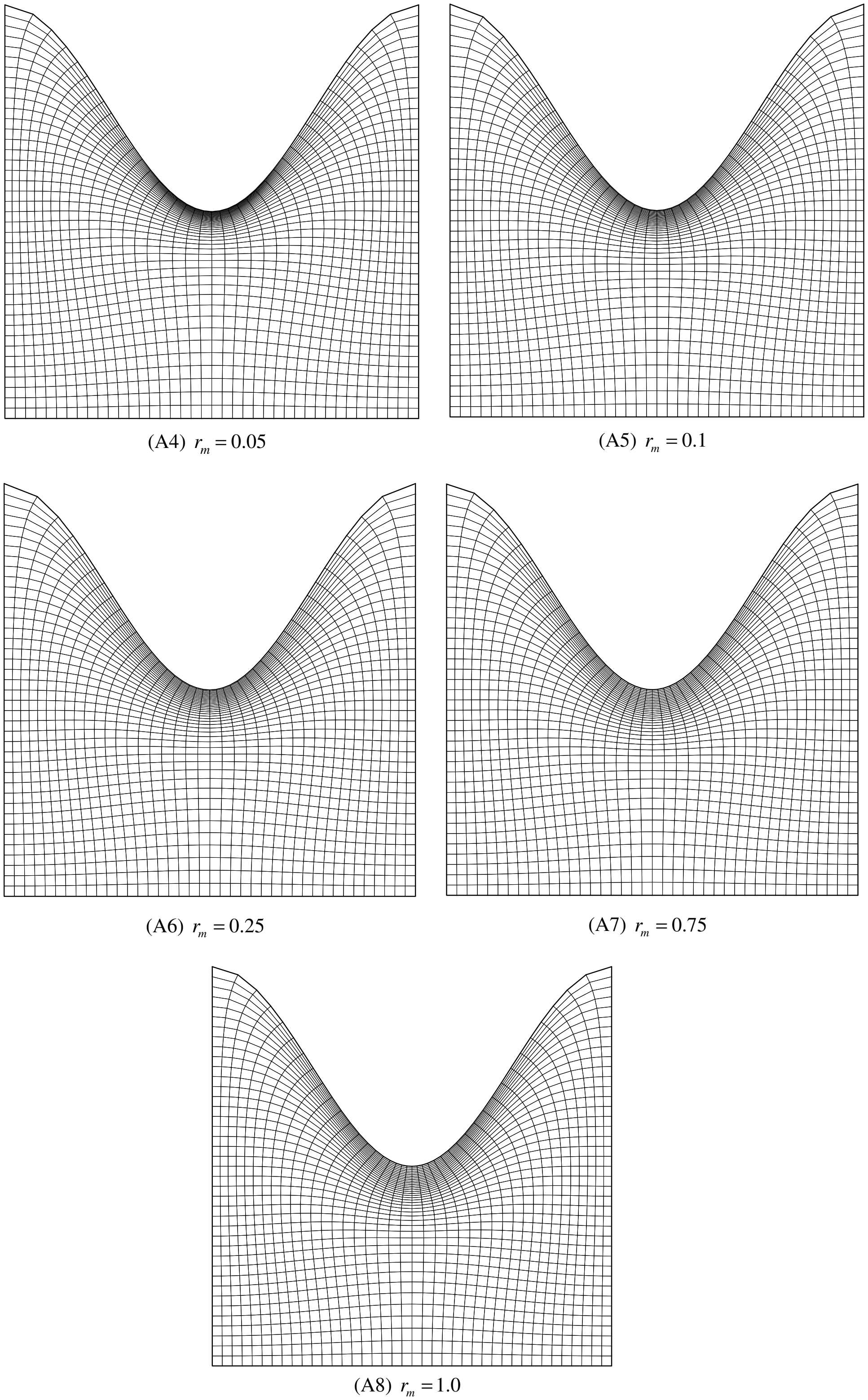
Fig. 6. Meshes in domain D.

Table 2
Evaluation of meshes in domains C and D

Case	Size	ADO	MDO	AAR	MAR	α	r_m	MCD	MAD	AAD
C2	41 × 41	0.37	1.11	3.98	46.1	0	–	–	–	–
C3	41 × 41	0.08	0.16	2.22	2.99	0	0.5	0.006	0.294	0.025
C4	41 × 41	0.07	0.19	2.25	3.25	0	0.5	0.006	0.291	0.024
D2	41 × 41	0.62	3.70	147	5373	0	–	–	–	–
D3	41 × 41	0.08	0.28	3.56	8.01	0	0.5	0.006	0.841	0.054
D4	41 × 41	0.12	0.34	3.53	7.14	0	0.5	0.006	0.852	0.055
D6	21 × 21	1.33	8.11	11. 5	77.3	0	–	–	–	–
D7	21 × 21	0.21	0.8	3.44	6.37	0	0.5	0.007	1.03	0.11
D9	11 × 11	2.47	8.41	5.49	25.4	0	–	–	–	–
D10	11 × 11	0.52	1.96	3.32	6.56	0	0.5	0.013	2.12	0.44

3.3. Sensitivity analysis

Domain A is selected to perform the sensitivity tests of the parameter r_m. Meshes generated with different values of r_m from 0 to 1 were plotted in Fig. 7 and their evaluations are listed in Table 1 and, Fig. 8 shows the



(A4) $r_m = 0.05$

(A5) $r_m = 0.1$

(A6) $r_m = 0.25$

(A7) $r_m = 0.75$

(A8) $r_m = 1.0$

Fig. 7. Meshes with different parameter r_m.

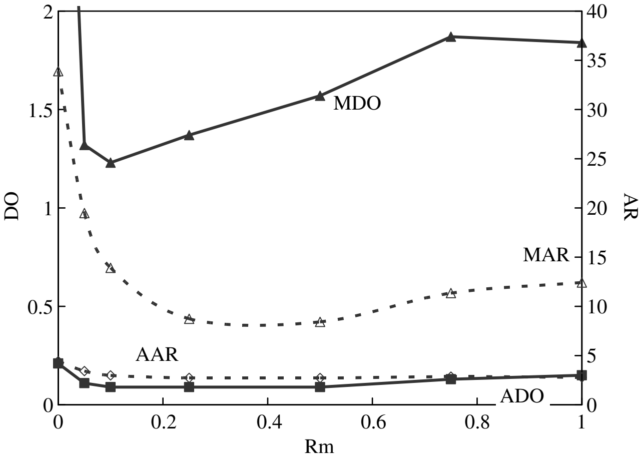
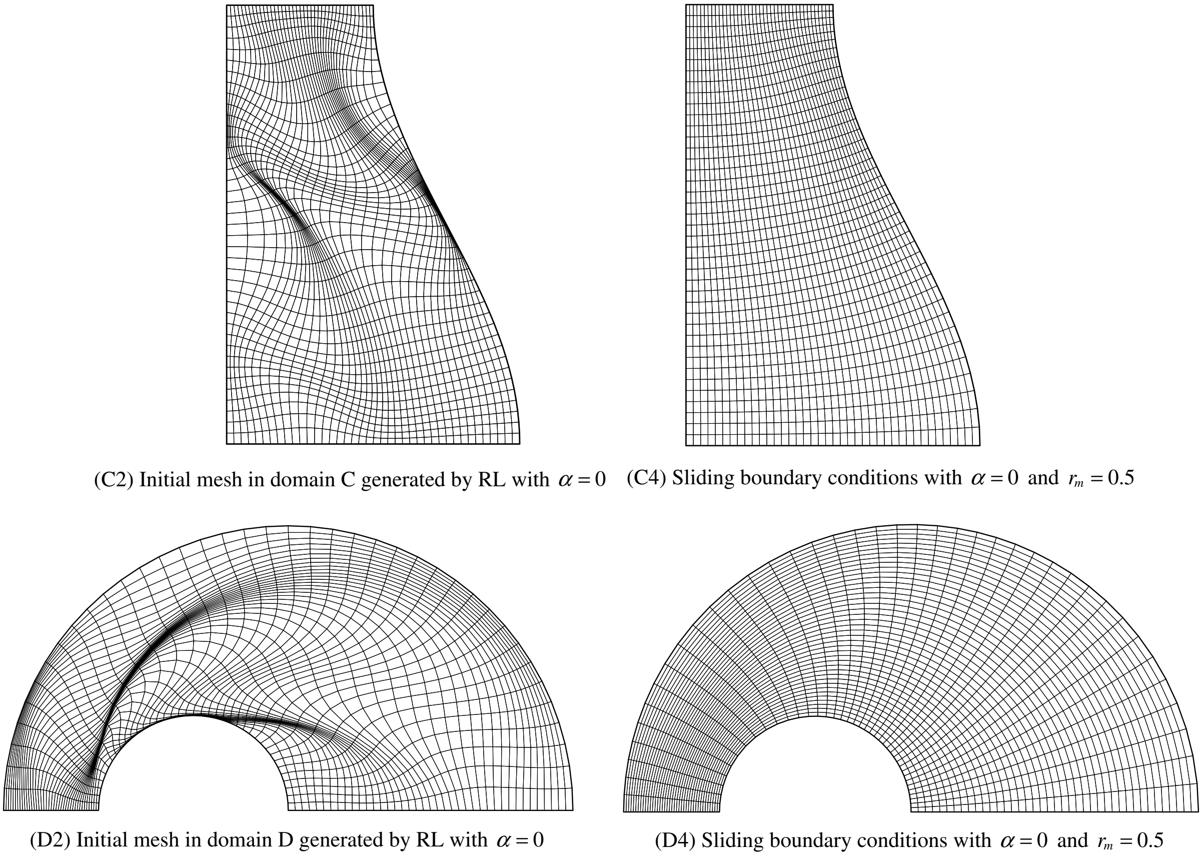


Fig. 8. Effects of parameter r_m.

changes of the indicators ADO, MDO, AAR and MAR with the parameter r_m. From these figures, it is found that mesh orthogonality (ADO and MDO) changes more mildly than mesh smoothness (AAR and MAR), which means r_m has little effects on mesh orthogonality. As for mesh smoothness, basically, the smaller r_m is, the more the top concaved boundary tends to freeze and the closer the mesh lines move to the concaved



(C2) Initial mesh in domain C generated by RL with $\alpha = 0$ (C4) Sliding boundary conditions with $\alpha = 0$ and $r_m = 0.5$

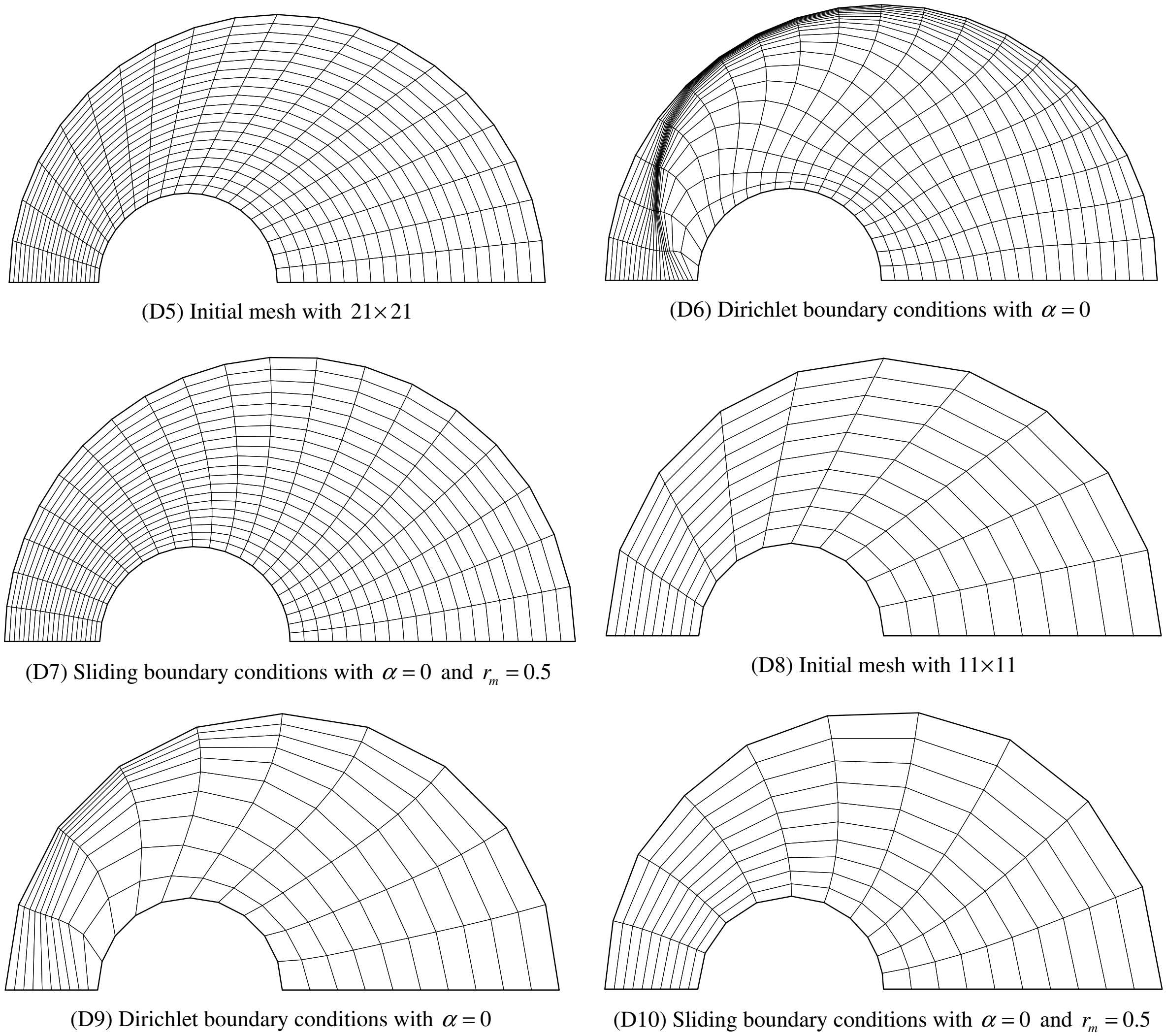
(D2) Initial mesh in domain D generated by RL with $\alpha = 0$ (D4) Sliding boundary conditions with $\alpha = 0$ and $r_m = 0.5$

Fig. 9. Effects of initial conditions.

boundary, and vice versa. The boundary curvature changes (MCD, MAD, and AAD) are almost the same for all the cases.

Domains C and D are selected to test the effects of the initial conditions on the present method. The final meshes with mesh distortion and overlapping (C2 and D2) generated with the Dirichlet boundary conditions are used as initial meshes. As shown in Fig. 9 and Table 2, the resulting meshes (C4 and D4) in both domains are very similar to their counterparts (C3 and D3), which indicates that the present method is not sensitive to the initial conditions.

Domain D is also used to analyze the effects of the mesh density. Three different algebraic meshes (41 × 41, 21 × 21, and 11 × 11) were generated as initial meshes. As can be seen in Fig. 10 and Table 2, with the mesh density decreasing, when using the Dirichlet boundary conditions, the mesh distortion and mesh overlapping (D6) become less serious and can even be neglected (D8). With the sliding boundary conditions, the boundary changes (D3, D7, and D9) become more significant with the mesh density decreasing and, as expected, better meshes both in orthogonality and smoothness were produced. In general, mesh orthogonality degenerate with the mesh density decreasing.



(D5) Initial mesh with 21×21

(D6) Dirichlet boundary conditions with $\alpha = 0$

(D7) Sliding boundary conditions with $\alpha = 0$ and $r_m = 0.5$

(D8) Initial mesh with 11×11

(D9) Dirichlet boundary conditions with $\alpha = 0$

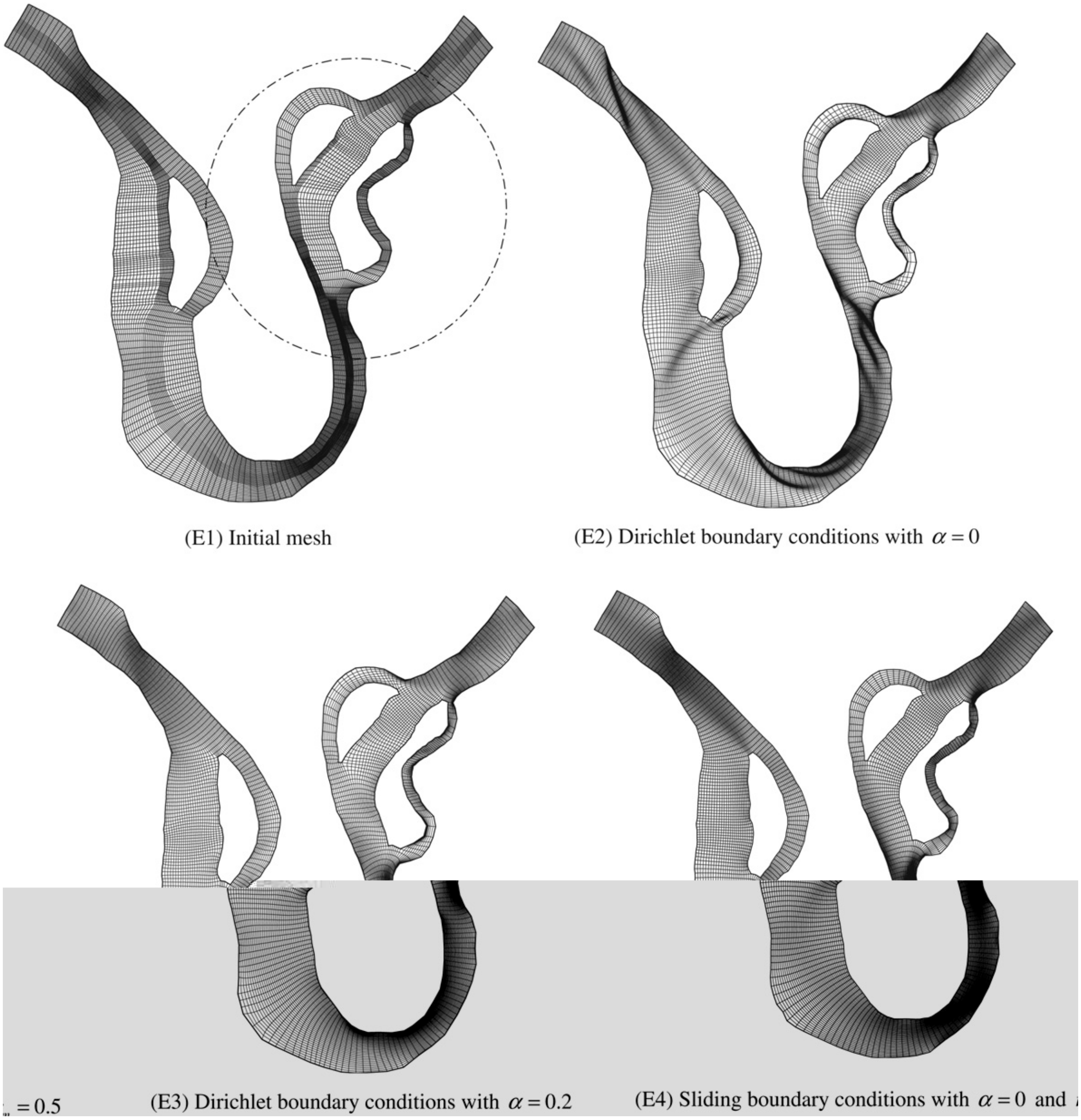
(D10) Sliding boundary conditions with $\alpha = 0$ and $r_m = 0.5$

Fig. 10. Effects of mesh density.

With the sensitivity analysis regarding to the mesh density, it can be concluded that for a boundary with a fixed number of nodes, the boundary curvature correction can only minimize the curvature changes to some extent but cannot avoid curvature changes. High-ordered boundary curvature correction method has limited effects on the curvature preservation, but using more nodes to represent the boundary (mesh refinement) can significantly improve the curvature preservation.

4. Applications

The present method was applied to two natural rivers (domains E and F) with complex boundaries. There are three islands in domain E and very irregular boundaries in domain F, which will cause big difficulties for mesh generation. For both domains, the initial meshes with nearly uniform nodal distribution along boundaries were generated by an algebraic method.


(E1) Initial mesh

(E2) Dirichlet boundary conditions with $\alpha = 0$

$_m = 0.5$

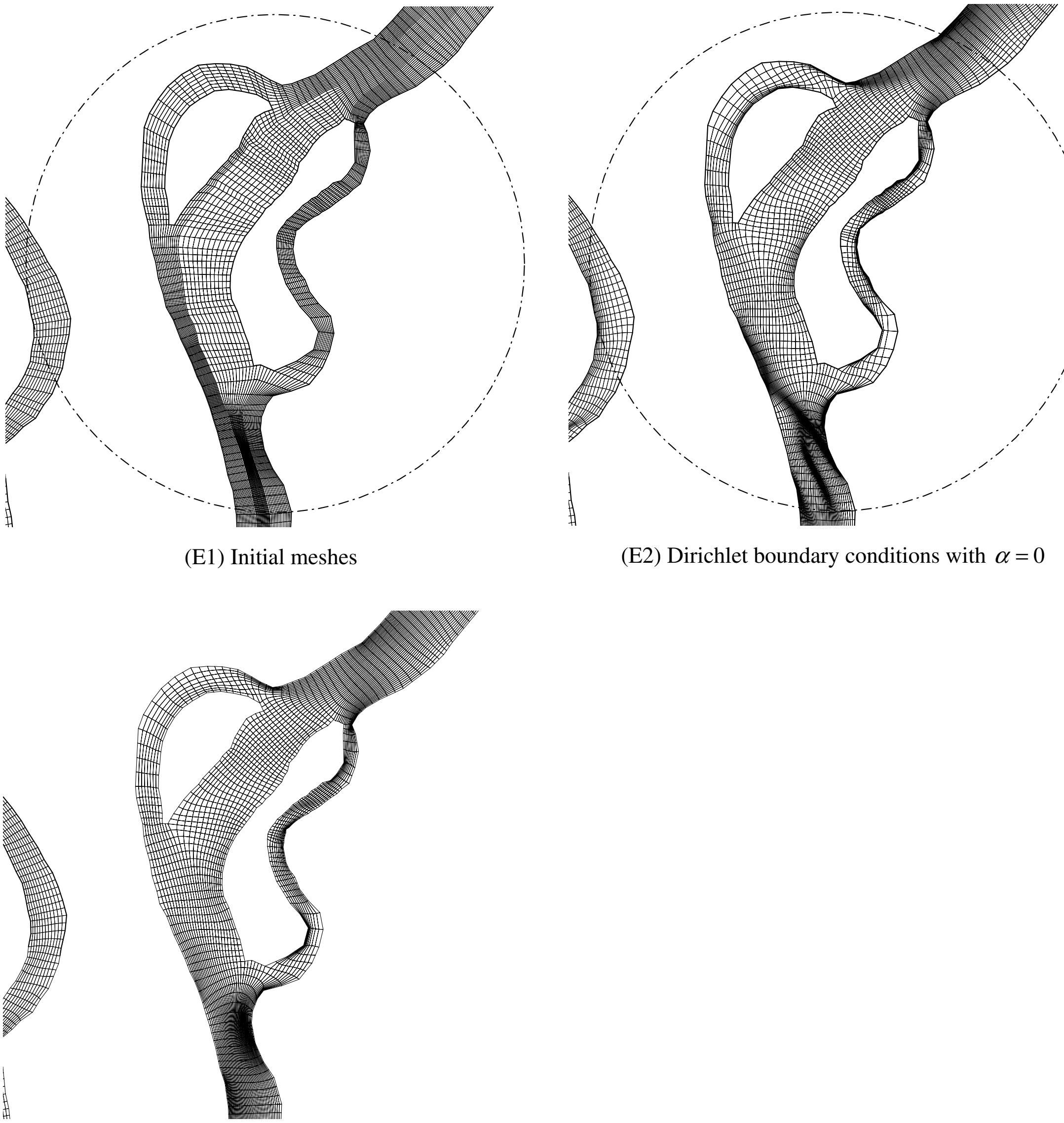
(E3) Dirichlet boundary conditions with $\alpha = 0.2$

(E4) Sliding boundary conditions with $\alpha = 0$ and

Fig. 11. Meshes in domain E.

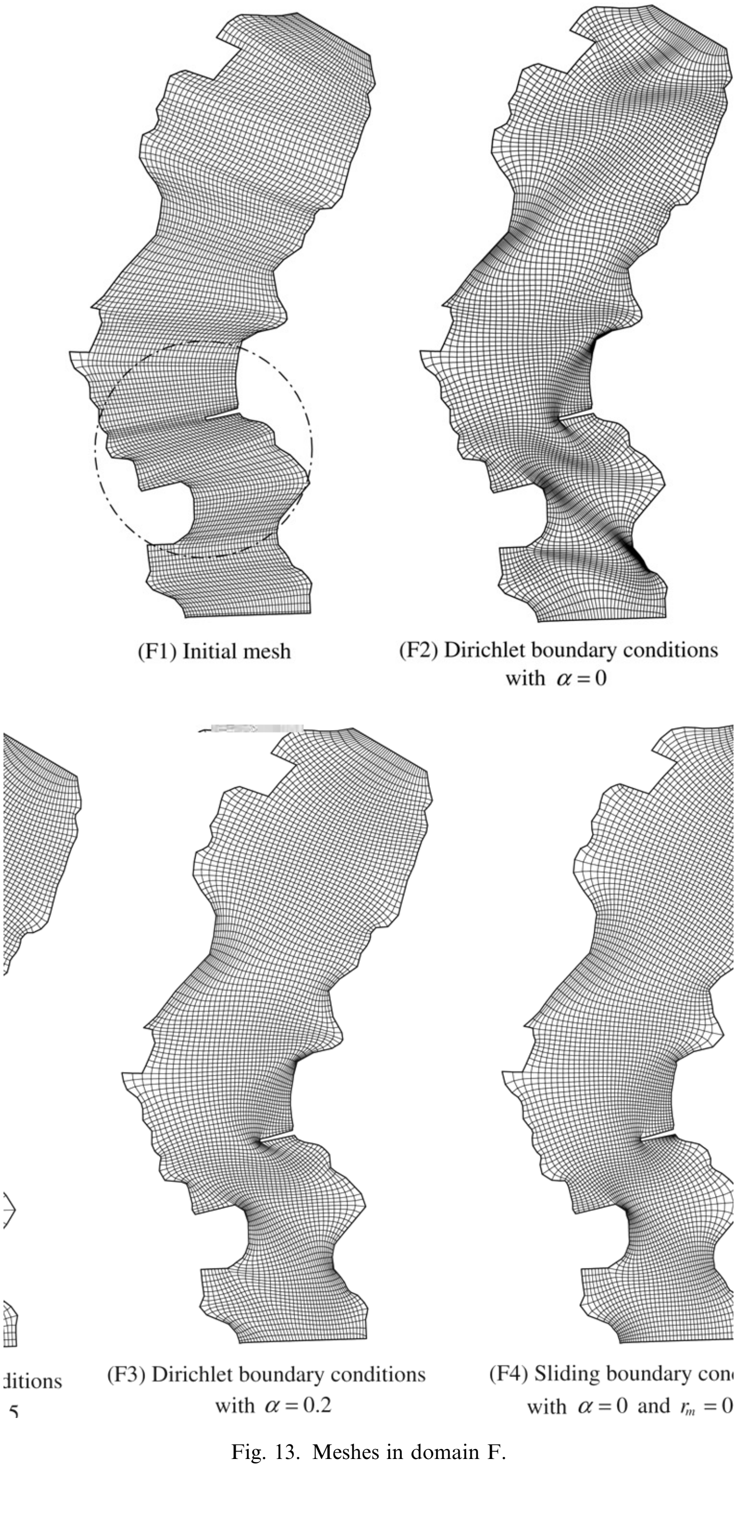
Figs. 11–14 and Table 3 shows the resulting meshes and the evaluation report, respectively. As expected, with the Dirichlet boundary conditions, skewed and squeezed meshes occurred in both domains, and the RL system without smoothness controls ($\alpha = 0$) failed to generate acceptable meshes (E2 and F2), while with considering smoothness ($\alpha = 0.2$), mesh smoothness (E3 and F3) was dramatically improved especially in domain E at the cost of a little orthogonality. When using the present method to apply the Neumann–Dirichlet boundary conditions to both domains, even without considering smoothness ($\alpha = 0$), high quality meshes (E4 and F4) both in orthogonality and smoothness were produced.

However, boundary curvature changes can still be sensed at some sharp crests or troughs of the curved boundaries. This problem could be alleviated greatly by using more nodes to represent those boundaries at the boundary definition stage, but increasing the accuracy of the curvature correction method is generally less effective.



(E1) Initial meshes

(E2) Dirichlet boundary conditions with $\alpha = 0$



(F1) Initial mesh

(F2) Dirichlet boundary conditions with $\alpha = 0$

ditions
5

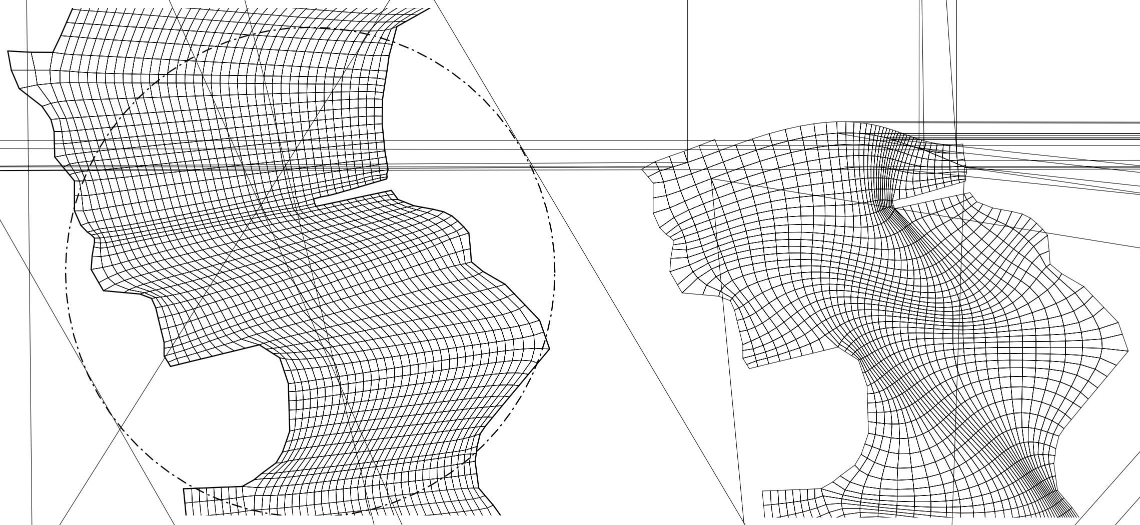
(F3) Dirichlet boundary conditions with $\alpha = 0.2$

(F4) Sliding boundary con
with $\alpha = 0$ and $r_m = 0$

Fig. 13. Meshes in domain F.

5. Conclusions

In CFD analysis, complex geometries remain a challenge not only in numerical simulation but also in mesh generation. In this study, a boundary treatment method for applying the Neumann–Dirichlet boundary



(F1) Initial mesh

conditions is proposed for 2D elliptic mesh generation in geometrically complex domains. The present method consists of three components: (1) the auxiliary mesh constructed with the boundary line and its neighboring mesh line; (2) the boundary curvature corrections using a second-order parabolic interpolation; and (3) the weighting of the boundaries according to the boundary curvatures. First, the boundary nodes are driven by solving the auxiliary mesh, then they are corrected by the parabolic interpolation and, finally, they are adjusted by the weights of the boundary.

According to the numerical tests, the width of the auxiliary mesh controlled by the parameter r_m has more effects on mesh smoothness rather than mesh orthogonality and, the present method is not sensitive to the initial conditions. Examples and applications have demonstrated that the present method can significantly improve mesh quality both in mesh orthogonality and smoothness. With the Neumann–Dirichlet boundary conditions, the boundary curvature changes are unavoidable and can be minimized to limited degrees by boundary curvature corrections despite of its accuracy. The drawback of the present method is that more computation efforts are needed which will slow down the whole computation.

Table 3
Evaluation of meshes in domains E and F

Case	Size	ADO	MDO	AAR	MAR	α	r_m	MCD	MAD	AAD
E2	43 × 250	1.98	23.6	12.5	16197	0	–	–	–	–
E3	43 × 250	4.84	30.9	4.48	60.8	0.2	–	–	–	–
E4	43 × 250	0.13	8.98	4.45	14.32	0	0.5	0.008	6.84	0.56
F2	41 × 137	1.13	15.6	2.32	85.32	0	–	–	–	–
F3	41 × 137	3.43	26.8	1.61	10.02	0.2	–	–	–	–
F4	41 × 137	0.52	14.9	1.57	11.52	0	0.5	0.012	7.93	1.03

In current study, a value of 0.5 is suggested for parameter r_m, and only one iteration cycle is used to solve the auxiliary mesh. Further researches on the evaluation of r_m and speeding up the computation are still needed.

Acknowledgement

This work is a result of research sponsored by the USDA Agriculture Research Service under Specific Research Agreement No. 58-6408-2-0062 (monitored by the USDA-ARS National Sedimentation Laboratory) and The University of Mississippi.

References

[1] R. Duraiswami, A. Prosperetti, Orthogonal mapping in two dimensions, J. Comput. Phys. 98 (1992) 254–268.

[2] L. Eça, 2D orthogonal grid generation with boundary point distribution control, J. Comput. Phys. 125 (1996) 440–453.

[3] Y.N. Jeng, C.-T. Chen, Two-dimensional orthogonal grid generation with floating boundary points, Numer. Heat Transfer, Part B: Fundam. 36 (2) (1999) 207–232.

[4] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, 1994.

[5] A. Khamayseh, A. Kuprat, C.W. Mastin, Boundary orthogonality in elliptic grid generation, Hand Book of Grid Generation, CRC Press, New York, 1999 (Chapter 6).

[6] G. Ryskin, L.G. Leal, Orthogonal mapping, J. Comput. Phys. 50 (1983) 71–100.

[7] R.L. Sorenson, A computer program to generate two-dimensional grids about airfoils and other shapes by the use of Poisson's equations, NASA TM 81198, NASA Ames Research Center, 1980.

[8] P.D. Thomas, J.F. Middlecoff, Direct control of the mesh node distribution in meshes generated by elliptic systems, AIAA J. 18 (1980) 652–657.

[9] J.F. Thompson, F.C. Thames, C.W. Mastin, TOMCAT—a code for numerical generation of boundary-fitted curvilinear coordinate system on fields containing any number of arbitrary two-dimensional bodies, J. Comput. Phys. 24 (1977) 274–302.

[10] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, Numerical Grid Generation: Foundation and Application, North-Holland, New York, 1985.

[11] Y.X. Zhang, Y.F. Jia, S.a.m.S.Y. Wang, 2D nearly orthogonal mesh generation, Int. J. Numer. Methods Fluids 46 (2004) 685–707.

[12] Y.X. Zhang, Y.F. Jia, Sam S.Y. Wang, Structured mesh generation with smoothness controls, Int. J. Numer. Methods Fluids 51 (2006) 1255–1276.

[13] Y.X. Zhang, Y.F. Jia, Sam S.Y. Wang, 2D nearly orthogonal mesh generation with controls on distortion functions, J. Comput. Phys. 218 (2) (2006) 549–571.