Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 223 (2007) 571–588
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Generation of curvilinear coordinates on multiply connected regions with boundary singularities

Vianey Villamizar [a,*], Otilio Rojas [b,1], Joseph Mabey [c]

[a] Department of Mathematics, Brigham Young University, 292 Talmage Math/Computer Building, P.O. Box 26539, Provo, UT 84602, USA
[b] Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Venezuela
[c] Department of Mechanical Engineering, Brigham Young University, Provo, UT 84602, USA

Received 25 February 2006; received in revised form 24 September 2006; accepted 26 September 2006
Available online 20 November 2006

Abstract

A new algorithm to generate smooth two-dimensional boundary conforming coordinates, with grid lines control, on multiply connected regions including boundary singularities, is devised. The technique fits into the category of elliptic grid generators, since it is based on the numerical solution of Poisson equations. The physical domain $\mathscr{D}$ is transformed into a topologically equivalent connected rectangular domain $\mathscr{D}'$ by defining a branch cut inside $\mathscr{D}$. Control of the grid line spacing over the multiply connected regions is established by appropriately distributing nodal points on the branch cut, and on other boundary curves. Simple and computationally convenient expressions for the control functions, present in the Poisson system, are obtained from this initial distribution of points. Thus, a natural link between clustering properties and the control functions is created. Fixing the nodal points on the branch cut causes the convergence of the numerical method about the singular boundary points. An iterative smoothing process relocating the branch cut is described. As a result, smooth grids are obtained. Grid quality analysis shows that coordinate line orthogonality in the new meshes is superior than in similar grids obtained from a well-known commercial software. The algorithm is successfully tested over a variety of multiply connected domains. The computational advantage of this novel grids is revealed by using them to numerically model the vibrations of complexly shaped annular membranes.

Keywords: Grid generation; Control functions; Smoothing process; Annular membrane vibrations

1. Introduction

The smoothness of a mesh and its appropriate grid point spacing are of primary importance in the numerical solution of partial differential equations (PDE) modelling physical phenomena. The focus of this work is to develop an algorithm capable of producing smooth curvilinear coordinates with control of the grid line

* Corresponding author. Tel.: +1 801 422 1754; fax: +1 801 422 0504.
E-mail addresses: vianey@math.byu.edu (V. Villamizar), orojas@blues.ciens.ucv.ve (O. Rojas), josephmabey@hotmail.com (J. Mabey).
[1] Present address: Computational Sciences Research Center, SDSU, San Diego, CA 92182-1245, USA.

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.09.028

spacing on geometries with one or several holes in the presence of non-smooth boundaries. An elliptic grid generation system is adopted. More precisely, a transformation T from a computational domain $\mathscr{D}'$ with rectangular coordinates (ξ,η), to a physical domain $\mathscr{D}$ with boundary conforming curvilinear coordinates $(x(\xi,\eta),y(\xi,\eta))$ is defined (see Fig. 1). Following a standard approach [19], the new curvilinear coordinates are obtained by numerically solving a Dirichlet boundary value problem governed by the familiar quasi-linear elliptic system of partial differential equations given by

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -\alpha\phi(\xi,\eta)x_\xi - \gamma\psi(\xi,\eta)x_\eta, \tag{1}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -\alpha\phi(\xi,\eta)y_\xi - \gamma\psi(\xi,\eta)y_\eta, \tag{2}$$

with control functions ϕ and ψ to be determined. Here, α, β and γ are scale metric factors of the transformation T, defined by $\alpha = x_\eta^2 + y_\eta^2, \beta = x_\xi x_\eta + y_\xi y_\eta, \gamma = x_\xi^2 + y_\xi^2$. It has been said [15] that the secret of each "good" elliptic grid generation algorithm is the method of computing control functions ϕ and ψ which satisfies all the clustering requirements. Unfortunately, there are not universal prescriptions to specify these functions, although many approaches have been proposed [11]. In our work, special conditions will be imposed on the definitions of ϕ and ψ to factor in all the clustering requirements. These conditions will be discussed in detail in Sections 2 and 3.

Numerous elliptic grid generation algorithms have been proposed over the last 20 years. An excellent source of the historical development and current procedures is contained in a handbook of grid generation by Thompson et al. [19]. In chapter 1 of this reference, standard procedures widely accepted now are described. Regarding the domain treatment, the common practice consists of dividing the physical region into subregions and generating a structured grid called a subgrid. If the subgrids share common interfaces, the grid or mesh is called a block-structured grid. Smoothness at the block interfaces is obtained by imposing continuity of each of the curvilinear coordinate functions and of their corresponding partial derivatives at the interfaces. Thus, the location of the interface nodes is determined by the grid generation method [18]. Although complex grids are divided into simpler ones, the iterative generation process is still performed over all the subgrids together. This procedure requires an additional data indexing procedure to link the blocks across the interfaces throughout the generation process.

An alternative procedure for two-dimensional multiply connected regions with non-smooth boundaries is proposed in this work. It involves a novel domain treatment and new control function definitions. The physical domain is transformed into a single rectangular block by means of a branch cut conveniently located inside the physical region. This branch cut, indicated by C_1 and C_2 in Fig. 1, plays a key role and will be denoted by the cursive letter $\mathscr{C}$. Dirichlet boundary conditions are determined from an initial distribution of mesh points not only on the physical boundaries, but also on the branch cut $\mathscr{C}$, as described in Section 3 (see Fig. 1). Also,

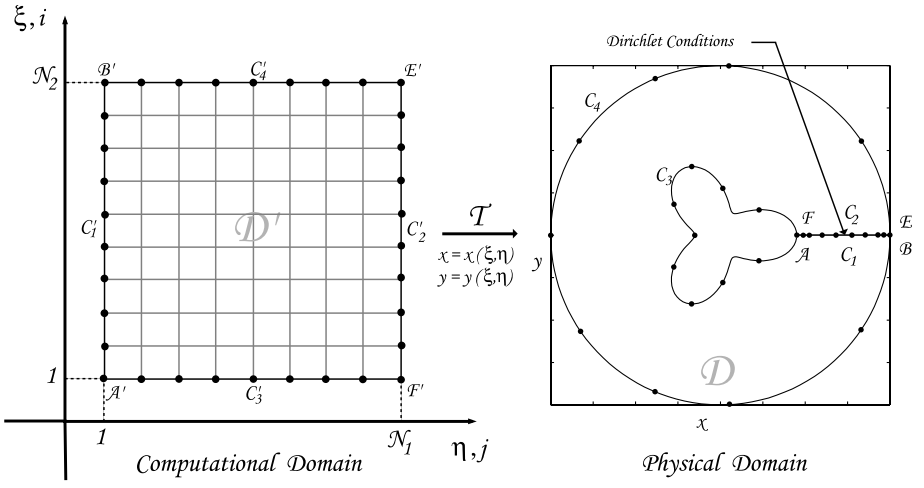

Fig. 1. Grid curves control based on distribution of grid points over a branch cut.

control functions solely dependent on the initial distribution of grid points on $\mathscr{C}$ and on the physical boundaries are defined (see Section 2). During the generation process an intermediate mesh essentially maintaining grid line distance according to the initial distance among boundary and branch cut nodes is obtained. The resulting mesh is continuous but does not necessarily have slope continuity at $\mathscr{C}$. An iterative smoothing process (described in Section 4) is implemented that consists of relocating the branch cut $\mathscr{C}$ and numerically solving the elliptic system until convergence is reached. As a result, globally smooth grids inside $\mathscr{D}$, retaining the point spacing of the initial grid on $\mathscr{C}$ and on the physical boundaries, are generated. Moreover, it is possible to specify the number of ξ-curves ($\xi = constant$) or η-curves ($\eta = constant$) that will be generated with a prescribed spacing. We call this new algorithm "Branch Cut Grid Line Control", or BCGC algorithm.

In this work, only O–type grids will be analyzed although the procedure can be easily extended to other grid types. An important feature of this procedure is the ability to generate smooth grids in the presence of boundaries with sharp corners or cusps. At these singular boundary points, our effort is directed toward the generation of conformal rather than orthogonal grids. As pointed out in [8], for highly obtuse or highly acute corners, it is often necessary to relax orthogonality at boundary points extending some distance from the corner.

In Section 5, smooth grids are generated for several domains with non-smooth boundaries. For some of these domains, the standard treatment of the branch cut $\mathscr{C}$ as an interface leads to diverging results (see Section 8.1). In Section 7, the BCGC grids are used to numerically approximate the vertical displacements of a complexly shaped annular membrane including boundary singularities. A stable periodic oscillation with a time period of approximately 17.25 units of time is attained. In contrast, numerical solutions based on grids generated from the meshing module of ANSYS, a well-known computer-aided engineering (CAE) software, become unstable after 22,000 time steps.

2. Definition of the grid control functions

The grid control functions $\phi(\xi,\eta)$ and $\psi(\xi,\eta)$ were systematically derived in [17] by imposing some geometrical constraints on the grid lines. Then, by numerically solving Eqs. (1) and (2), line control in the interior from the grid point distribution on the boundaries was achieved. However, this was successfully accomplished only for simply connected regions. In the present work, we extend the technique introduced in [17] to multiply connected domains. In the following sections, we will illustrate the method for a domain with a single hole in its interior. Extending the technique to domains with multiple holes is more laborious but straightforward.

In our approach, Dirichlet boundary conditions are specified on the two vertical segments $\eta = 1$ and $\eta = N_1$ corresponding to the branch cut $\mathscr{C}$ (see Fig. 1). As usual, they are also defined on the horizontal straight segments $\xi = 1$ (inner physical boundary) and $\xi = N_2$ (outer physical boundary) of the computational domain. We will show later that any initial spacing of grid points on the branch cut will be preserved in the interior by conveniently defining the control functions ϕ and ψ. The definition follows the one described in [17], but it is adapted to our multiply connected domains. By combining Eqs. (1) and (2) in two different ways, it is possible to eliminate $\psi(\xi,\eta)$ and $\phi(\xi,\eta)$, on the right hand side, and obtain

$$\alpha[y_\eta(x_{\xi\xi}+\phi x_\xi)-x_\eta(y_{\xi\xi}+\phi y_\xi)] = y_\eta^2\left[2\beta\left(\frac{x_\eta}{y_\eta}\right)_\xi-\gamma\left(\frac{x_\eta}{y_\eta}\right)_\eta\right], \tag{3}$$

$$\gamma[y_\xi(x_{\eta\eta}+\psi x_\eta)-x_\xi(y_{\eta\eta}+\psi y_\eta)] = y_\xi^2\left[2\beta\left(\frac{x_\xi}{y_\xi}\right)_\eta-\gamma\left(\frac{x_\xi}{y_\xi}\right)_\xi\right] \tag{4}$$

From Eq. (3), it is observed that if the right hand side vanishes on the vertical segments $\eta = 1$ and $\eta = N_1$, then $\phi(\xi,\eta) = \frac{x_\eta y_{\xi\xi}-y_\eta x_{\xi\xi}}{y_\eta x_\xi - x_\eta y_\xi}$, when $\eta = 1$ and $\eta = N_1$. Now, an easy way for the right hand side of (3) to vanish is to impose the conditions $\beta = 0$ and $\left(\frac{x_\eta}{y_\eta}\right)_\eta = 0$, on the branch cut $\mathscr{C}$. This last condition requires that the transverse coordinate curves (ξ-curves), be locally straight to $\mathscr{C}$. Also, $\beta = 0$ implies that $\frac{x_\eta}{y_\eta} = -\frac{y_\xi}{x_\xi}$, when $\eta = 1$ and $\eta = N_1$. Substitution of this expression into the above formula for ϕ leads to the elimination of all the partial derivatives in the variable η. In fact,

$$\phi(\xi,\eta)=\frac{-(x_\xi x_{\xi\xi}+y_\xi y_{\xi\xi})}{x_\xi^2+y_\xi^2},\quad \text{when } \eta=1 \quad \text{and } \eta=N_1. \tag{5}$$

A procedure completely analogous is applied to Eq. (4) along the horizontal segments $\xi=1$ and $\xi=N_2$. Then after the elimination of all the partial derivatives in the variable ξ, we are led to

$$\psi(\xi,\eta)=\frac{-(x_\eta x_{\eta\eta}+y_\eta y_{\eta\eta})}{x_\eta^2+y_\eta^2},\quad \text{when } \xi=1 \quad \text{and } \xi=N_2. \tag{6}$$

The partial definition of the control functions (5) and (6) involves derivatives with respect to one variable only. In particular, the control function ϕ only contains derivatives with respect to ξ, which is the free variable along the branch cut $\mathscr{C}$. Similarly, ψ only contains derivatives with respect to η, which is the free variable along the straight segments $\xi=1$ and $\xi=N_2$ corresponding to the physical boundaries. Therefore, the derivative terms present in the above definitions can be easily approximated from the initial distribution of grid points. As a consequence, numerical values of the control functions on the rectangular computational domain boundaries can be completely determined from the initial distribution of grid points on the physical boundaries and on the branch cut $\mathscr{C}$. In the next paragraphs, details about the previous statements are given.

To begin, we will discuss the initial distribution of points along $\mathscr{C}$ and the physical boundaries. For simplicity, the branch cut $\mathscr{C}$ will be located along the x-axis of the physical domain as shown in Fig. 1. As a consequence, $y(\xi,1)=y(\xi,N_1)=0$ for all ξ. Therefore, partial derivatives with respect to ξ of the y coordinate are all zero along the branch cut. In particular, $y_{\xi\xi}(\xi,1)=0$ on $\mathscr{C}$. If additionally, a uniform partition of nodes is specified as a boundary condition on $\mathscr{C}$, then $x(\xi,1)$ is a linear function of ξ. As a result, $x_{\xi\xi}(\xi,1)=0$. Substitution into (5) leads to $\phi=0$ on $\mathscr{C}$. Therefore, control of grid lines cannot be established through the function ϕ unless the initial distribution of grid points on the branch cut is not uniform.

Our definition for the non-uniform initial distribution of nodes on $\mathscr{C}$ consists of uniformly spaced clusters of left end and right end points, and a non-uniformly spaced set of nodes in between. To start the definition, a step size $\Delta h=l/(N_2-1)$ is computed. The symbol l represents the length of the branch cut. A certain number of points n_r are selected in the vicinity of the outer circle separated by a uniform distance $\Delta h_r=f_r\Delta h$. Similarly, a set of n_l points is selected in the vicinity of the inner boundary with a uniform spacing $\Delta h_l=f_l\Delta h$. The parameters f_l and f_r define the contraction ($0<f_l,f_r<1$) or stretching ($f_l,f_r>1$) experienced by consecutive points on the branch cut which were initially separated a distance Δh. The expression $f_l\times(n_l/f_r)\times n_r$ will be used to describe the control imposed on the grid over the branch cut $\mathscr{C}$. Similarly, the notation $N_1\times N_2$ will be used to identify the grid size.

The spacing between intermediate points on $\mathscr{C}$ is determined by an osculating polynomial of degree five that interpolates two nodes. One of them is the closest in the left cluster to the intermediate set. The other one is the closest in the right cluster to the intermediate points. Continuity of the first and second derivatives is enforced at these interpolating nodes to determine the interpolating polynomial. Two examples of this initial distribution on $\mathscr{C}$ are depicted at the left in Figs. 7 and 8. The reasons for this choice are the following: (1) The node distribution is governed by a C^2 curve. Therefore, the control functions (5),(6) are continuous at any node of the rectangular boundaries where the boundary curve is sufficiently smooth. (2) It is possible to precisely determine the number of grid lines forming clusters near the inner and near the outer boundary. (3) There is a smooth cell size transition between the cluster cells and the ones immediately adjacent located in the intermediate region (see Figs. 4 and 5). A completely analogous procedure can be implemented on the outer and inner boundaries to obtain an initial distribution of nodes with a desired spacing.

Before proceeding with the construction of the grid generation algorithm, numerical values of the grid control functions ϕ and ψ at all nodes of $\mathscr{D}'$ are needed. To begin, a uniformly spaced rectangular grid with step sizes $\Delta\xi=1$ and $\Delta\eta=1$ is defined on the computational domain $\mathscr{D}'$. The discrete values for ξ and η are represented by $\xi_i=(i-1)\Delta\xi$ and $\eta_j=(j-1)\Delta\eta$, for $i=1,\ldots,N_2$ and $j=1,\ldots,N_1$, respectively. Also, the discrete values of $x(\xi_i,\eta_j)=x(i,j)$ and $y(\xi_i,\eta_j)=y(i,j)$ are denoted by $x_{i,j}$ and $y_{i,j}$, respectively.

Similar notations for the discrete values of α, β, γ, ϕ, and ψ will be employed. As previously stated, values of ϕ on $\mathscr{C}$ can be easily approximated from a non-uniform initial distribution of points $\{(x_{i,1},y_{i,1})=(x_{i,N_1},y_{i,N_1}),i=1,\ldots,N_2\}$ on $\mathscr{C}$. In fact, by using centered differences in the variable ξ the following approximations are obtained:

$$\phi_{i,1} = -\frac{1}{\gamma_{i,1}}\left[\frac{(x_{i+1,1}-x_{i-1,1})(x_{i+1,1}-2x_{i,1}+x_{i-1,1})}{2}+\frac{(y_{i+1,1}-y_{i-1,1})(y_{i+1,1}-2y_{i,1}+y_{i-1,1})}{2}\right],$$
$$\phi_{i,N_1} = \phi_{i,1} \quad i = 2,\ldots,N_2-1. \tag{7}$$

If $i = N_2$ then, second order backward finite-difference approximations of the derivative terms present in ϕ at $(N_2, 1)$ are employed. Analogously, values of ψ at the physical boundaries can be easily approximated from an initial distribution of points using centered differences in the variable η. Finally, the values of $\phi(\xi,\eta)$ and $\psi(\xi,\eta)$ at the interior points of the computational domain $\mathscr{D}'$ are obtained by linear interpolation along the ξ-curves and the η-curves, respectively. It will be shown in the following sections that the control function ϕ carries the spacing of the initial distribution of grid points on $\mathscr{C}$ to the corresponding ξ-curves inside the physical domain $\mathscr{D}$. These two distances may differ to some degree depending on the bounding curve shapes. Similarly, it can be shown that the control function ψ is responsible for carrying the spacing of the initial distribution of grid points on the physical boundaries to the corresponding η-curves.

3. Boundary value problem and its numerical solution

Following the procedure described in the previous sections, it was possible to define the grid control functions ϕ and ψ over the entire rectangular computational domain. As a next step, we proceed to numerically solve the boundary value problem consisting of Eqs. (1) and (2), with Dirichlet boundary conditions along the sides of the rectangular computational region (see Fig. 1). This boundary value problem is solved by approximating the partial derivative terms, included in (1) and (2), using centered differences and then by applying point SOR iteration to the discrete system of equations that results. More precisely, the discrete equation to be solved iteratively for the curvilinear coordinate x is given by

$$x_{i,j} = \frac{1}{2(\alpha+\gamma)_{i,j}}\left[\alpha_{i,j}\left(\left(1+\frac{\phi_{i,j}}{2}\right)x_{i+1,j}+\left(1-\frac{\phi_{i,j}}{2}\right)x_{i-1,j}\right)+\gamma_{i,j}\left(\left(1+\frac{\psi_{i,j}}{2}\right)x_{i,j+1}+\left(1-\frac{\psi_{i,j}}{2}\right)x_{i,j-1}\right)\right.$$
$$\left.-\frac{\beta_{i,j}}{2}(x_{i+1,j+1}-x_{i+1,j-1}-x_{i-1,j+1}+x_{i-1,j-1})\right], \tag{8}$$

where

$$\alpha_{i,j} = \left((x_\eta)_{i,j}\right)^2+\left((y_\eta)_{i,j}\right)^2 \quad \beta_{i,j} = (x_\xi)_{i,j}(x_\eta)_{i,j}+(y_\xi)_{i,j}(y_\eta)_{i,j}$$
$$\gamma_{i,j} = \left((x_\xi)_{i,j}\right)^2+\left((y_\xi)_{i,j}\right)^2 \quad (x_\eta)_{i,j} = (x_{i,j+1}-x_{i,j-1})/2,$$
$$(y_\eta)_{i,j} = (y_{i,j+1}-y_{i,j-1})/2, \quad (x_\xi)_{i,j} = (x_{i+1,j}-x_{i-1,j})/2,$$
$$(y_\xi)_{i,j} = (y_{i+1,j}-y_{i-1,j})/2 \quad i = 2,\ldots,N_2-1, \quad j = 2,\ldots,N_1-1.$$

An analogous equation is obtained for the discrete values $y_{i,j}$. The convergence of the iteration is accelerated by updating the most recent value of x and y at the k-iteration with the relaxation parameter ω as follows: $x_{i,j}^k = \omega x_{i,j}^k + (1-\omega)x_{i,j}^{k-1}$ and $y_{i,j}^k = \omega y_{i,j}^k + (1-\omega)y_{i,j}^{k-1}$. The iterative process requires the definition of an initial mesh ($k = 1$) with a grid point spacing over $\mathscr{C}$, the inner, and the outer boundary in accordance with the desired spacing between neighboring grid lines in the final grid, as shown in Fig. 2(a). This initial grid consists of circular rings around the inner boundary and straight segments (rays). Each ring passes through a branch cut node and its radius is the distance from this node to the origin. The straight segments (rays) interpolate nodes between the inner and the outer boundary.

The SOR iteration stops when the maximum distance, in terms of the sup-norm, between the points of two consecutive grids is within some specified tolerance. It means $\max_{1\leqslant i\leqslant N_2, 1\leqslant j\leqslant N_1}(|x_{i,j}^k - x_{i,j}^{k-1}|, |y_{i,j}^k - y_{i,j}^{k-1}|) < \mathrm{Tol}$. For illustration, a coarse grid, obtained following the numerical procedure just described, is shown in Fig. 2(b). The domain corresponds to the three-leafed rose region with inner boundary described by the parametric equations, $x(\theta) = 0.3(2+\cos(3\theta))\cos(\theta)$, $y(\theta) = 0.3(2+\cos(3\theta))\sin(\theta)$, $0 \leqslant \theta \leqslant 2\pi$. The outer radius is $r_{\text{outer}} = 2$. The grid shown in Fig. 2(a), which has size 61×31 and has clusters of five ξ-curves at each end with spacing factors $f_l = 0.6$ and $f_r = 0.5$ ($0.6\times$ $5/0.5\times 5$), was used as an initial grid. The iteration process

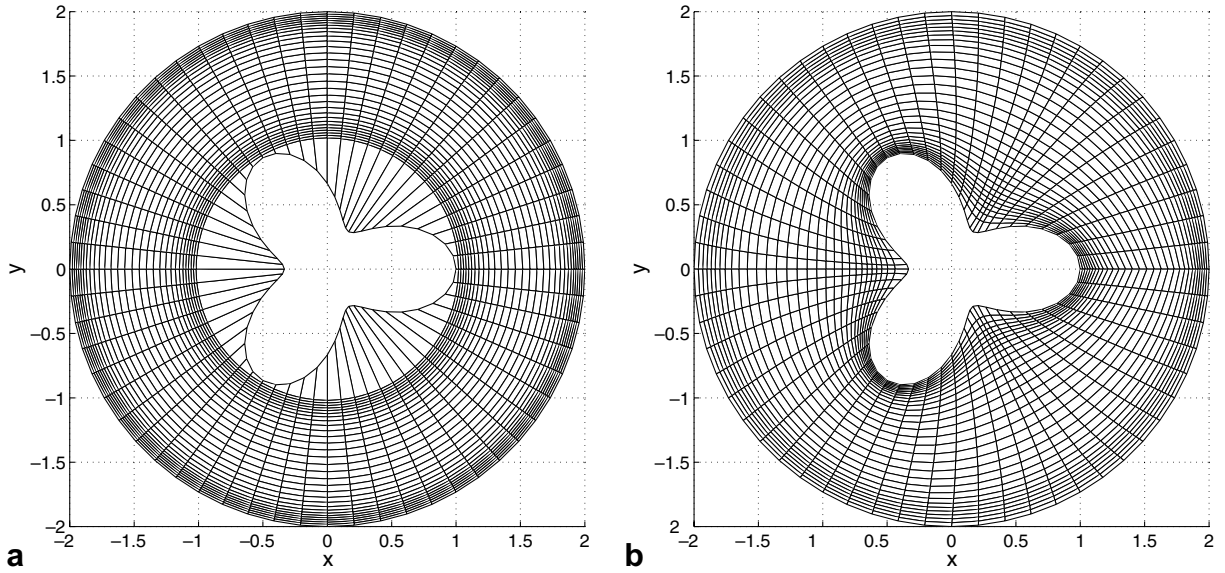

Fig. 2. (a) Initial grid G_0 ($61 \times 31 - 0.6 \times 5/0.5 \times 5$). (b) First BCGC grid G_1.

converged after 63 iterations for a relaxation parameter $\omega = 1.85$ and a tolerance $\mathrm{Tol} = 10^{-5}$. It is evident how the ξ-curves passing through initially clustered $\mathscr{C}$ nodes stay close throughout the entire domain. However, the resulting grid though continuous is non-smooth over the branch cut as shown in Fig. 2(b). This non-smoothness problem is solved by defining an iterative smoothing procedure described in the next section. As stated in the introduction, this algorithm will be called "Branch Cut Grid Line Control", or BCGC algorithm. The corresponding grid generator subroutine will be called *gengridBCGC*.

4. BCGC algorithm and the smoothing process

In this section, we summarize the steps required to generate a smooth BCGC grid with a prescribed spacing of ξ-curves near the inner boundary curve, or near the outer boundary; or with a prescribed spacing of η-curves at a specified location inside the physical domain.

Step 1 (first branch cut).

(a) Select an initial branch cut $\mathscr{C}_1$. Choose $\eta = 1$, the grid line corresponding to $\mathscr{C}_1$.
(b) Specify an initial distribution of points over $\mathscr{C}_1$ as part of a Dirichlet boundary condition (follow the procedure described in Section 2). Also, specify an initial distribution of points on the physical boundaries.
(c) Define the control functions ϕ and ψ as indicated in Section 2.
(d) Define an initial mesh G_0 for the domain $\mathscr{D}$ according to Section 2 (see Fig. 2(a)).
(e) Apply the SOR iterative numerical method described in Section 3 (subroutine *gengridBCGC*) to obtain an intermediate mesh G_1 (see Fig. 2(b)). Note the non-smoothness of this grid at the branch cut.

Step 2 (second branch cut).

(a) Select a different η-curve as a new branch cut, $\mathscr{C}_2$. For simplicity, $\mathscr{C}_2$ is chosen 180° from the previous branch cut $\left(\eta = \mathrm{int}\left(\frac{N_1+1}{2}\right)\right)$.
(b) The coordinates of the points on grid G_1 that are located over $\mathscr{C}_2$ define a Dirichlet boundary condition on $\mathscr{C}_2$ for a new BVP. This new problem is identical to the one solved in step 1 save the boundary values are over the new branch cut $\mathscr{C}_2$.

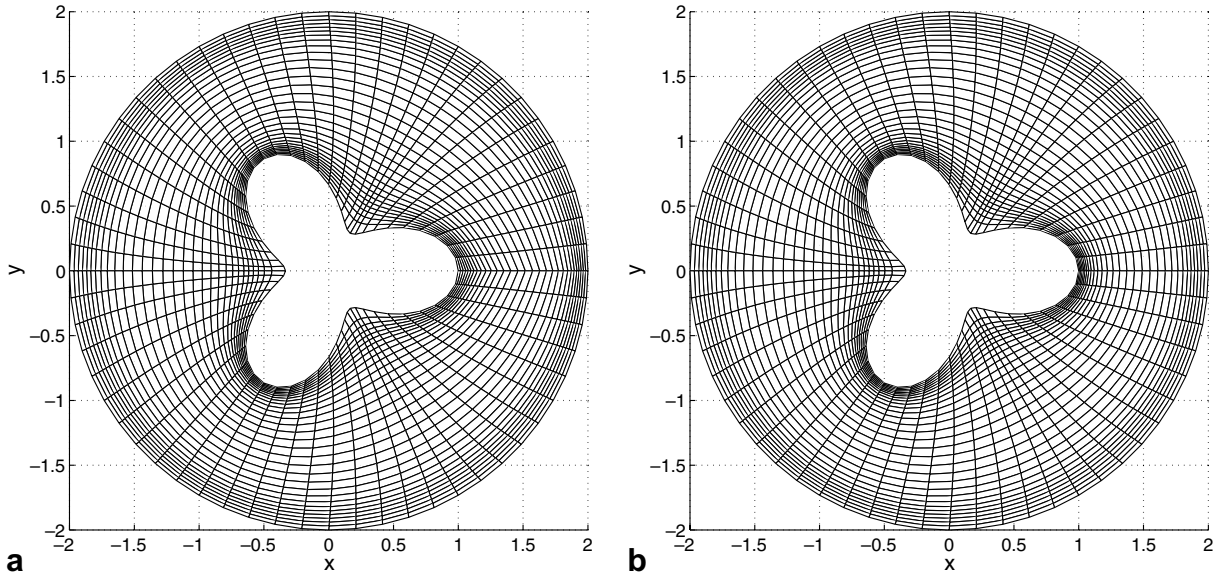

Fig. 3. (a) First BCGC grid G_1. (b) Final grid G_3.

(c) A new grid G_2 is obtained by applying *gengridBCGC* to the new BVP described in (b) that uses G_1 as an initial grid. Note that the previously broken grid at $\mathscr{C}_1$ is now smooth as seen in Fig. 3(b).

Step 3 (Stop criteria and final smooth grid).

(a) Select a new branch cut $\mathscr{C}_3$, *e.g.* $\mathscr{C}_3 = \mathscr{C}_1$.

(b) The coordinates of the points on grid G_2 that are located over the new branch cut $\mathscr{C}_3$ partially define a Dirichlet boundary condition for a new BVP.

(c) Using G_2 as the initial grid, the subroutine *gengridBCGC* is applied to the new BVP and a new grid G_3 is generated. If the number of iterations performed by *gengridBCGC* to generate G_3 is three or less, the process stop and G_3 is selected as the final grid.

(d) Otherwise, the smoothing continues as many times ($i = 4, 5, \ldots, \mathrm{M}$) as needed. During the process, new branch cut relocations C_i are implemented and new intermediate grids G_i are obtained until *gengridBCGC* requires three iteration or less to generate the final grid G_M.

During the branch cut interchange process the control functions ϕ and ψ remain unchanged. This is necessary to preserve the spacing information from the initial distribution through the generation process. The smoothing iterative process just described can be considered as an adjustment of the initial distribution of nodes over the original branch cut $\mathscr{C}$. The adjustment consists of relocating the branch cut nodes in order to achieve smoothness of the final grid on $\mathscr{C}$ without sensibly affecting the spacing of the grid lines.

It is well-known that the Winslow grid generator, which is defined by (1) and (2) with zero right hand sides, gives smooth grids. A rigorous result, Rado's theorem [12], establishes that the Winslow transformation is a diffeomorphism (one-to-one, onto and of class C^1) if the transformation from the computational to the physical domain boundary is one-to-one, onto and continuous in both directions. Even if the physical boundary has slope discontinuities, the interior grid is generally smooth if the regularity condition on the boundary transformation is verified. An example is provided in [9] for a domain identical to the "Chevron" symbol. We are not aware of a similar rigorous result for the nonhomogeneous quasi-linear elliptic system (1),(2). However, it is reasonable to expect smoothness of the solution in the interior of $\mathscr{D}$ if the control functions are continuous or more regular. This fact motivates our definition of $x(\xi,\eta)$ as a C^2 function on $\mathscr{C}$, since it implies that $\phi(\xi,\eta)$ is continuous on $\mathscr{C}$. Also, $\psi(\xi,\eta)$ is continuous everywhere on the physical boundaries if the initial distribution of points is sufficiently smooth, with the only exception of the boundary points where

Table 1
Grid convergence of BCGC algorithm applied to the three-leafed rose ($61 \times 31 - 0.6 \times 5/0.5 \times 5$)

Grids compared	Maximum distance	Average distance	Iterations
G_0–G_1	5.8e−01	1.8e−01	63
G_1–G_2	1.5e−01	6.8e−03	56
G_2–G_3	9.5e−06	9.7e−07	1

Table 2
BCGC algorithm applied to the three-leafed rose domain with grid parameters: $2 \times 7/0.3 \times 3$

Grid size	101×101	151×151	201×201	251×251	301×301
G_0–G_1	521	1082	1789	2620	3561
G_1–G_2	441	902	1470	2129	2864
G_2–G_3	167	285	398	498	587
G_3–G_4	17	32	53	78	107
G_{17}–G_{18}	–	–	–	–	3
Total Iters	1166	2345	3794	5440	7295

there is slope discontinuity. By extending the definition of the control functions to the interior of $\mathscr{D}$ using linear interpolation, we were able to obtain smoothness of the grid lines in the interior even for singular boundaries as the astroid and epicycloid (see Section 5).

The grids obtained by applying the BCGC algorithm to the rose's domain defined in Section 3 are illustrated in Figs. 2 and 3. In Table 1, the convergence process is described. "Maximum distance" refers to the maximum Euclidean distance between corresponding nodes of the two grids being compared. "Avg. dist" refers to the average of all the Euclidean distances between corresponding nodes. The final mesh G_3 shown in Fig. 3(b) is smooth (everywhere in the interior) and its grid lines are spaced according to the initial node distribution.

The smoothing process can be validated by repeating the previous experiment, for a 61×31 BCGC grid with control parameters $1 \times 1/1 \times 1$, for the three-leafed rose. As expected, the first intermediate grid obtained G_1 is broken at the branch cut nodes. The final grid G_3 which is obtained in three BCGC steps after 119 iterations is identical to the smooth grid that would be generated by applying Winslow's algorithm directly.

The performance of the BCGC algorithm on practical computational grids is reported on Table 2 for various mesh sizes. The maximum distance from the origin to the rose boundary is now two. The outer boundary is located at $r_{\text{outer}} = 10$, the mesh parameters are $2 \times 7/0.3 \times 3$, and the other parameters are defined as in the previous experiment. All the computational experiments were performed on a dual processors 2.7 GHz PPC970.

5. Application of BCGC algorithm to multiply connected domains including boundary singularities

Four additional multiply connected domains with a hole in their interiors are analyzed in this section. For comparison, the minimum distance from the inner and the outer boundary points equals three in all cases. The grids consist of 61 η-curves and 31 ξ-curves and the branch cut spacing is defined as $0.6 \times 5/0.5 \times 5$. Along the hole bounding curve and the outer boundary curve, the location of the grid points (Dirichlet boundary condition) is determined using a uniform angular step. The rest of the parameter values are identical to the previous three-leafed rose's experiment. First, a BCGC grid is obtained for an astroid shaped obstacle whose parametric equations are given by $x(\theta) = 0.5(3\cos(\theta) + \cos(3\theta))$ and $y(\theta) = 0.5(3\sin(\theta) - \sin(3\theta))$, for $0 \leqslant \theta \leqslant 2\pi$. The initial and the final grid after four steps are shown in Fig. 4. The following experiment corresponds to a domain with an epicycloid inner boundary shape and $r_{\text{outer}} = 6$, as shown in Fig. 5. Two more experiments are performed for an elliptical and pacman inner curves, respectively. The convergence process of the BCGC algorithm for these domains is summarized in Table 3. It is observed that for regions with sharp corners or cusps (astroid and epicycloid), the number of iterations is slightly higher at first. However, the total number of iterations is similar in all cases with the exception of the astroid where the number of iterations is

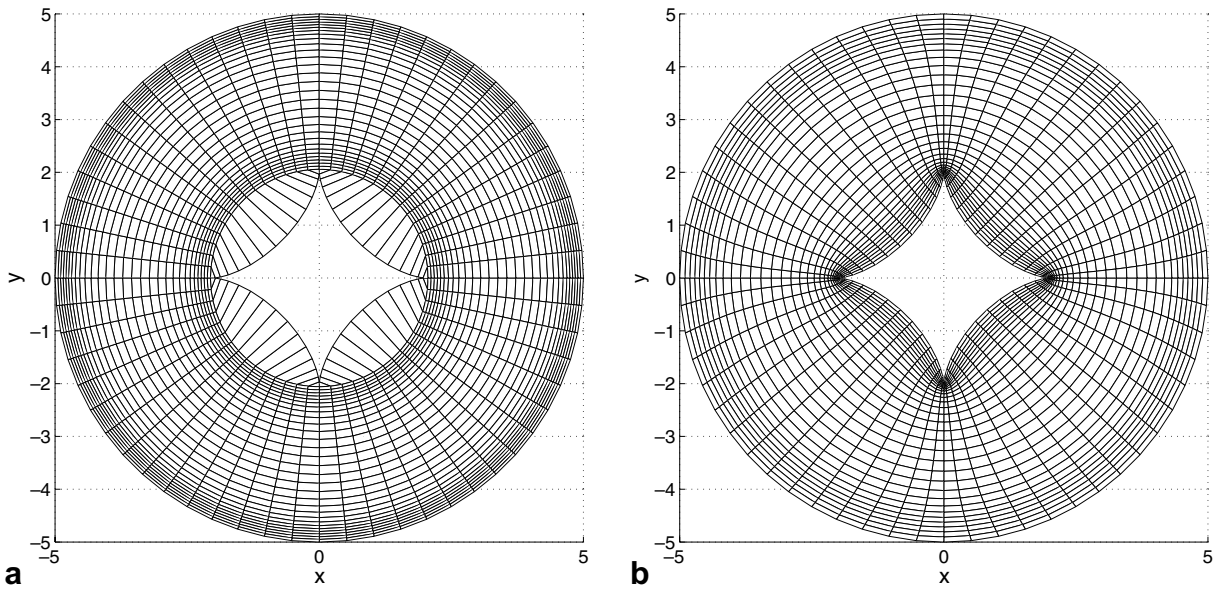

Fig. 4. (a) Astroid initial grid, G_1. (b) Astroid fourth BCGC grid, G_4.

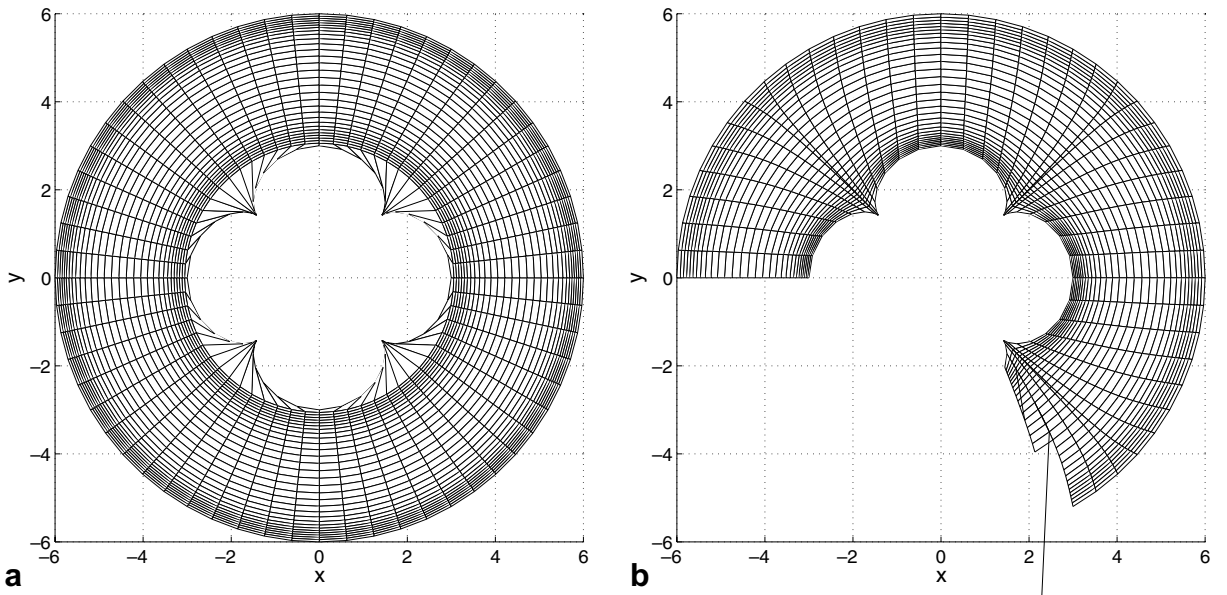

Fig. 5. (a) Epicycloid initial grid, G_1. (b) Epicycloid third BCGC grid, G_3.

Table 3
BCGC algorithm applied to arbitrary shaped domains ($61 \times 31 - 0.6 \times 5/0.5 \times 5$)

Step	Astroid	Ellipse	Pacman	Epicycloid	Rose
G_0–G_1	112	76	66	97	73
G_1–G_2	100	65	59	62	64
G_2–G_3	14	16	1	1	10
G_3–G_4	1	1	–	–	1
G_4–G_5	–	–	–	–	–
Total Iters	227	158	126	160	148

Table 4
Jacobian and deviation from orthogonality

Shape	ADO		J_{min}		Iters		J_{max}	
	101×101	301×301	101	301	101	301	101	301
Astroid	6.1°	6°	1.2e−6	7.5e−9	1240	7968	5.6e−2	6e−3
Pacman	1.4°	2°	4.7e−3	1.7e−4	1076	6674	5.9e−2	7.7e−3
Epicycloid	9°	8.8°	2e−3	6.2e−5	965	6226	5.3e−2	5.6e−3
Rose	5.2°	5°	4.7e−3	5e−4	1166	7295	5e−2	5.4e−3

Control parameters: $2 \times 7/0.3 \times 3$, $r_{outer} = 10$.

about 55% greater than the average of the others. As seen in Table 4, BCGC algorithm was able to generate much finer and practical non-self-overlapping grids with a prescribed spacing for all the domains under consideration.

6. Grid quality analysis

There are several properties that serve to measure the quality of a grid. For example, smoothness, orthogonality, non-self-overlapping, and bounded aspect ratios. These properties can be determined by the Jacobian matrix $\mathscr{J}$ of the transformation T [10]. A highly desired one is that the grid be non-self-overlapping. This is equivalent to requiring that the parametric functions $x(\xi,\eta)$ and $y(\xi,\eta)$ defining the transformation T be one-to-one, which is also equivalent to the condition that the Jacobian $J = x_\xi y_\eta - x_\eta y_\xi$ of the transformation T does not change in sign on the entire parametric domain [5]. For algebraic grid generation methods, such as transfinite interpolation (TFI) or boundary-conforming mapping (BCM), a well-known deficiency is that for very convoluted bounding curves self-overlapping often occurs [23]. This is when some quadrilateral cells overlap each other.

To test self-overlapping behavior of the BCGC algorithm, we performed several experiments for a 61×41 grid of the astroid domain, defined in the previous section, with control parameters: $0.8 \times 5/0.3 \times 5$. A self-overlapping grid can be detected by a change in sign of the Jacobian. A set of experiments depending on the ratio, ratio $= r_{outer}/\max(r_{inner})$, was considered. The smallest ratio employed was 2/1000. The maximum inner radius, $\max(r_{inner})$, was kept fixed at 2. As the radius of the outer boundary, r_{outer}, increases the number of BCGC iterations and cell sizes also increase. However, the resulting grid is self-overlapping free. As expected, the quality of the BCGC meshes degrades as ratio approaches 1. For instance, if $r_{outer} \leqslant 4$ the BCGC grids overlap at the cusps. This overlapping problem can be fixed by increasing f_l, the spacing among the branch cut nodes in the vicinity of the inner boundary. In fact for $f_l = 1$, the 2/4 ratio grid is non-self-overlapping. Therefore, the BCGC algorithm does not guarantee non-self-overlapping grids in general. However in many cases when a self-overlapping grid is generated, it is possible to make it non-self-overlapping, without changing the grid size, by adjusting the spacing on the initial grid, as mentioned above.

In Table 4, we have included a fine 301×301 non-self-overlapping grid for the astroid domain. Thus even for boundary shapes with strong singularities, it is possible to generate reasonable fine non-self-overlapping grids by applying the BCGC algorithm with appropriate control parameters.

Another important property of a grid is the orthogonality between the crossing grid lines. In this work, we adopt some parameters introduced in [6]. For instance, the maximum and mean deviation from orthogonality ADO and MDO, respectively, are defined as

$$\mathrm{ADO} = \frac{1}{(N_1-1)(N_2-2)} \sum_{i=2}^{N_2-1} \sum_{j=1}^{N_1-1} \left(\left|90^\circ - \theta_{i,j}\right|\right) \quad \theta_{i,j} = \arccos\left(\frac{\beta_{i,j}}{(\alpha_{i,j}\gamma_{i,j})^{1/2}}\right).$$

$$\mathrm{MDO} = \max\left|90^\circ - \theta_{i,j}\right|, \quad 2 \leqslant i \leqslant N_2 - 1; \quad 1 \leqslant j \leqslant N_1 - 1$$

The angle $\theta_{i,j}$ is a discrete approximation for the local distortion angle between grid lines at the point $(x_{i,j}, y_{i,j})$. Unfortunately for many interesting domains such as those discussed in this work, the orthogonality constraints lead to collapse of grid lines. It is well-known that simultaneously conditioning more than one prop-

erty of the grid potentially results in clashing demands [10]. For example in [1], it is shown that reducing the aspect ratio values may result in an increment of the mean deviation from orthogonality.

As mentioned in the introduction, our main effort is in controlling the grid line spacing. The results obtained by applying BCGC algorithm to 101×101 and 301×301 grids, with control parameters $2 \times 7/0.3 \times 3$, are reported in Table 4. The columns $J_{\min}$ and $J_{\max}$ contain the minimum and maximum absolute values of the Jacobian, respectively. These values are closely related to the area of the cells. The *Iters* column reports the total number of iterations required by the BCGC method to reach convergence. Curves describing the epicycloid and astroid experience abrupt changes of 360° at the cusps points. Neighboring grid lines also experience abrupt changes as they go near these singular points. As a consequence, the maximum deviation from orthogonality is relatively high. However, the BCGC grids obtained are not too far from orthogonality in the majority of the nodes, as revealed by the values of the parameter average deviation from orthogonality (ADO) (in degrees) in Table 4.

7. Vibration of complexly shaped annular membranes using BCGC grids

In this section, application of the BCGC grids to the numerical modelling of arbitrarily shaped vibrating annular membranes is considered. Analytical solutions can only be obtained for geometrically simple shapes. Therefore, numerical methods should be employed in general. Complexly shaped domains lead to the use of boundary conforming curvilinear coordinates $x(\xi,\eta)$ $y(\xi,\eta)$, as described in Section 1. The IBVP, written in terms of BCGC curvilinear coordinates, is given by

$$W_{tt} = \frac{c^2}{J^2}\left(\alpha W_{\xi\xi} - 2\beta W_{\xi\eta} + \gamma W_{\eta\eta} + \alpha\psi W_{\xi} + \gamma\phi W_{\eta}\right), \quad (\xi,\eta) \in \mathscr{D}', \quad t > 0, \tag{9}$$

$$W(\xi,1,t) = 0, \quad W(\xi,n_2,t) = 0, \tag{10}$$

$$W(\xi,\eta,0) = f(\xi,\eta), \quad W_t(\xi,\eta,0) = g(\xi,\eta), \quad (\xi,\eta) \in \mathscr{D}', \tag{11}$$

where $\mathscr{D}'$ is the rectangular domain in coordinates ξ and η. We choose a complex domain including cusps in its boundary to show the advantages of using BCGC grids. More precisely, the shape of the annular membrane has the astroid curve defined in Section 5 as its inner boundary, and, a circular outer boundary of radius $r_{\text{outer}} = 10$. The membrane is fixed at the boundary points. There is an initial deformation given by

$$f(\xi(x,y),\eta(x,y)) = \begin{cases} 0, & \text{if } r = \sqrt{x^2+y^2} \leqslant 5; \\ \sin\left(\frac{\pi}{5}(10-r)\right), & \text{if } 5 < r \leqslant 10. \end{cases} \tag{12}$$

The initial velocity is assumed to be zero, and the wave speed is taken as $c = 1$. The above IBVP Eqs. (9)–(12) is numerically solved using an explicit, marching in time, finite-difference method (centered finite-differences) based on the Winslow and the BCGC grids described in this work. The temporal step size Δt is chosen such that numerical stability is guaranteed.

For illustration, we choose an 81×41 coarse grid, with control parameters $2 \times 7/0.3 \times 3$ and a time step $\Delta t = 0.001$, to present our results. They are shown in Fig. 6. As expected, the vertical displacement of the membrane corresponds to a time periodic oscillation. After the transient solution dies out, a time period of approximately 17.25 units of time is attained. We ran experiments for grid sizes up to 301×301. Appropriate cell sizes are required to satisfy the CFL stability condition for the explicit finite-difference method. We found that for 141×141 Winslow grids and $\Delta t = 0.001$ the numerical solution blows out because the CFL condition is violated. In contrast, BCGC grids, with control parameters $2 \times 7/0.3 \times 3$, produces satisfactory results for $\Delta t = 0.001$ combined with mesh sizes up to 241×241.

8. Concluding remarks

We have developed an efficient numerical method (BCGC algorithm) to control the spacing of grid lines for two-dimensional multiply connected regions including severe boundary singularities. Smooth boundary-conforming coordinates have been obtained by combining a smoothing process with the numerical solution of quasi-linear elliptic systems of partial differential equations. An important feature of the BCGC algorithm

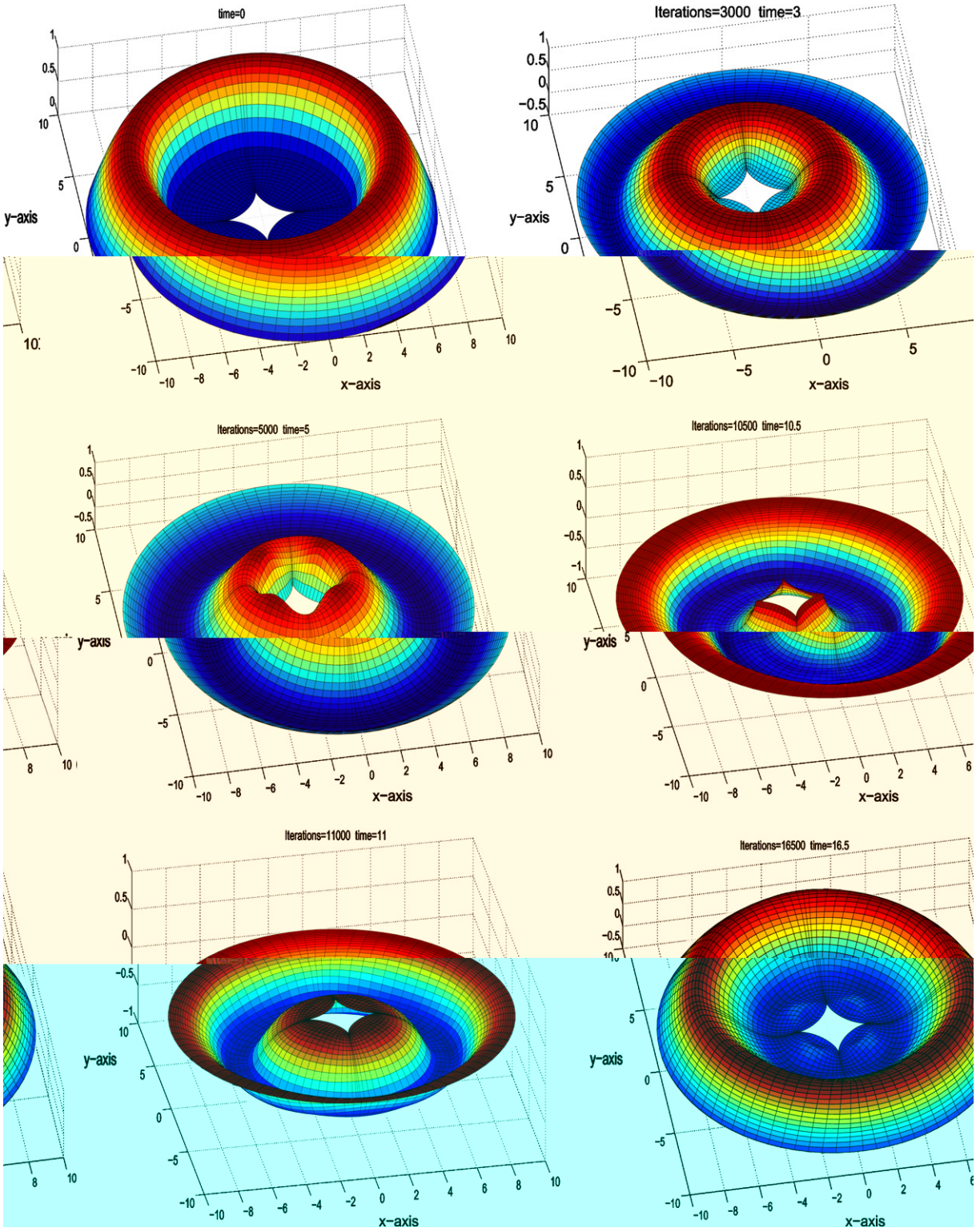

Fig. 6. Vibrating annular membrane with an astroid curve defining the inner boundary.

is its ability to produce non-self-overlapping and smooth grids (interior region) for domains exhibiting sharp corners or cusps. Results for domains bounded by non-smooth curves, such as epicycloid and astroid, are illustrated in Figs. 3–5.

Currently, we are working on the construction of BCGC grids for 3D complexly shaped domains. The results of these studies will be reported in a separate paper. Certainly, BCGC technique would have a greater impact in 3D applications than in 2D applications. However, the results presented here for 2D complex configurations will benefit the practitioners of grid generation since they constitute an important contribution to the present 2D technology regarding grid control. On the other hand, most of the techniques and applications of 3D grid generation have been, and continue to be, devised initially for two-dimensional configurations. This statement is supported by a considerable number of recent publications in 2D grid generation techniques [1–3,8,23], and also by recent publications in applied areas that use 2D grids [3,7,13,14,20,22].

8.1. Comparison with standard techniques

The BCGC algorithm includes several new aspects not considered in previously developed elliptic grid generation algorithms. First, the grid nodes on the selected branch cut are fixed (Dirichlet boundary condition) during the application of the SOR iteration to the discrete system defined in Section 3. This means that the selected branch cut $\mathscr{C}$ is treated as a physical boundary in contrast with standard procedures [18,19] where $\mathscr{C}$ is considered as an interface on which complete continuity is usually verified. This procedure is responsible for the convergence of the iterative numerical method in the presence of severe boundary singularities, as explained below. As a result, a non-smooth grid at the originally selected $\mathscr{C}$ is obtained (recall the rose experiment described in Section 3).

Second, a practical grid line control mechanism, for O-type grids, is defined from an initial node distribution on $\mathscr{C}$ (see Section 2) combined with appropriate definition of control functions ϕ and ψ. Third, a novel grid smoothing procedure is introduced (see Section 4). It consists of relocating the original branch cut and then reapplying the iterative numerical method until convergence is achieved. The relocation procedure is repeated as many times as needed, until smooth grids in the interior of $\mathscr{D}$ are obtained.

Finally, the definition of the control function ϕ on the branch cut only includes derivatives of the physical coordinates $x(\xi,\eta)$ and $y(\xi,\eta)$ with respect to the free variable ξ along $\mathscr{C}$ (see Eq. (5)). The main advantage of it is that values of the control functions are obtained directly from the initial distribution of grid nodes on $\mathscr{C}$ and on the physical boundaries. Thus, a natural link between the initial clustering properties and the control functions is established. The influence of the initial distribution of nodes on $\mathscr{C}$ onto the grid line distribution of the final BCGC grid is evident from Figs. 7 and 8, for the epicycloid domain.

In [18], control functions are defined following three different approaches: from a global smoothed initial algebraic grid, from an initial boundary point distribution, and by imposing orthogonality at boundaries with specified normal spacing. The difficulty associated with the last approach is that the definition of the control functions along the physical boundaries and $\mathscr{C}$ includes derivatives of the coordinates with respect to both independent variables ξ and η. Therefore, only one of these derivatives can be approximated from the initial boundary or branch cut point distributions, then additional conditions need to be imposed.

The second approach based on the initial distributions of nodes along the branch cut and the physical boundaries is also different than our approach. Our definition includes derivatives of the two coordinates $x(\xi,\eta)$ and $y(\xi,\eta)$ with respect to the free variable instead of derivatives of only one of the two physical coordinates as proposed in [18]. Another approach, described in detail in chapter 4 of Ref. [19], requires the introduction of a parameter space P and a definition of a "grid control map" from the computational domain $\mathscr{D}'$ to P. The definition of the grid control map is not simple, especially when generating smooth grids in the entire domain. In some cases, it is required to solve Laplace's equation for the parameters involved in the grid control map.

The non-smoothness of the grids at the originally selected branch cut could be avoided by modifying the branch cut treatment in the BCGC algorithm. In fact by assuming complete continuity on the original $\mathscr{C}$, it is possible to obtain smooth grids for domains such as the three-leafed rose of Section 3 without relocating $\mathscr{C}$. Unfortunately, this procedure does not always work for domains with severe singularities such as the astroid and the epicycloid of Section 5, as shown in Table 5.

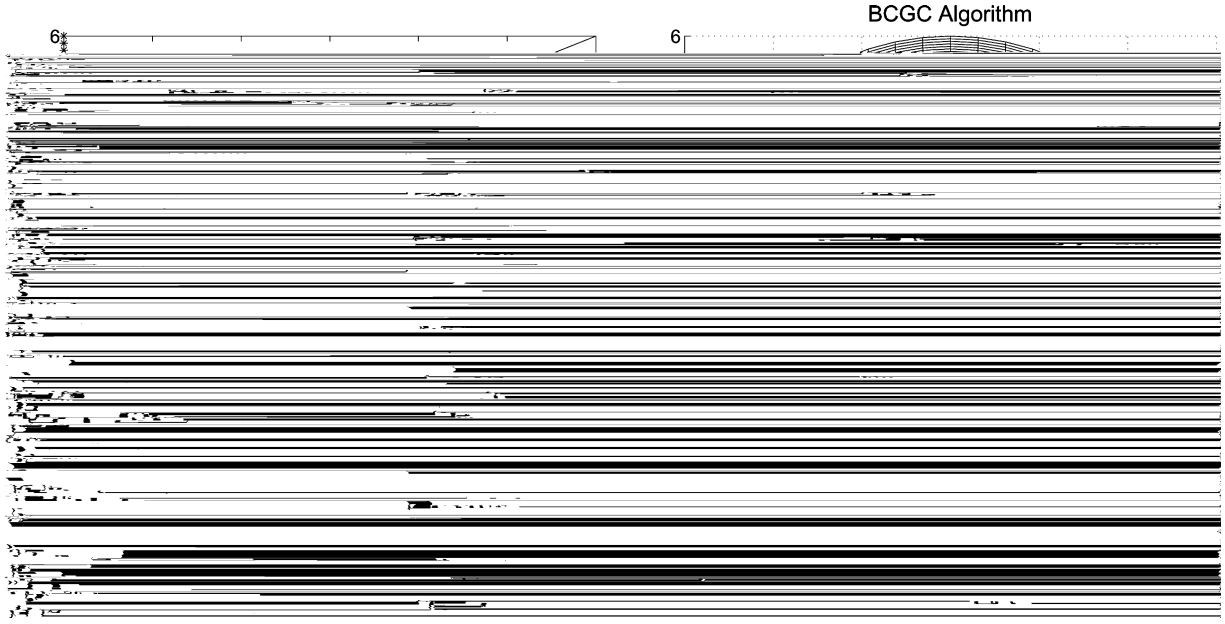

Fig. 7. Initial distribution along $\mathscr{C}$ and final BCGC grid $(61 \times 31 - 2 \times 5/0.3 \times 5)$.

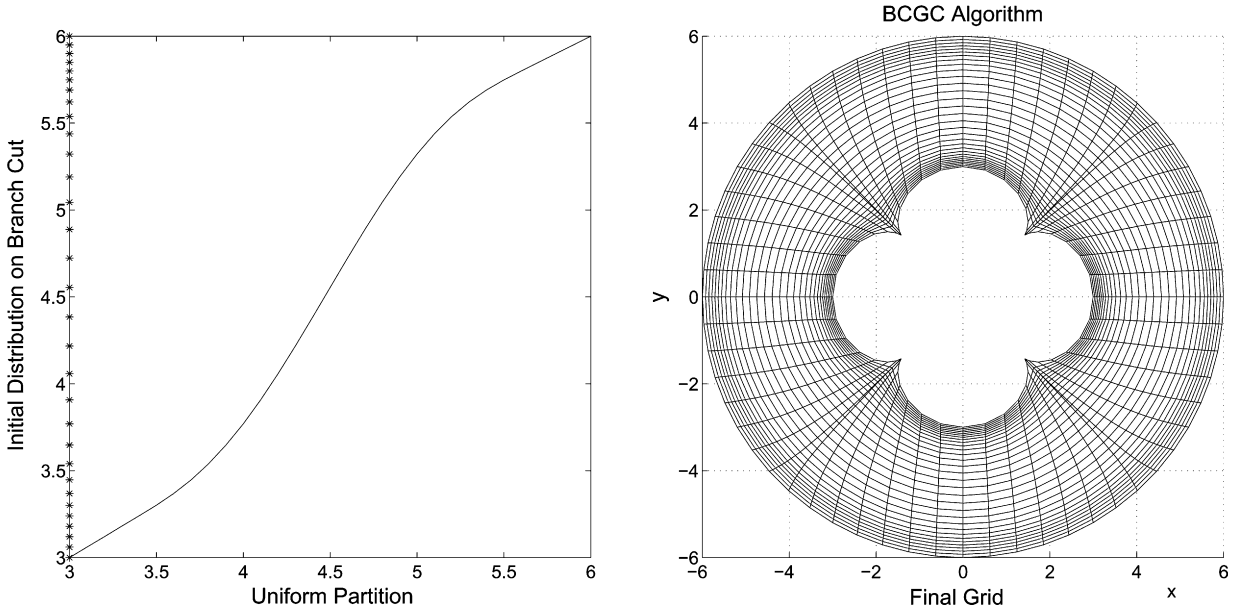

Fig. 8. Initial distribution along $\mathscr{C}$ and final BCGC grid $(61 \times 31 - 0.6 \times 5/0.3 \times 5)$.

One of the major accomplishments of the BCGC algorithm is its ability to obtain smooth grids (in the interior) for those singular domains where the standard treatment of a branch cut, as an interface, fails. In fact, fixing the node locations on the selected branch cut is a key step in the convergence of the SOR iterative numerical method.

In Table 5, several experiments performed for the astroid and epicycloid domains are reported. The grids analyzed are 61×41 with control parameters on $\mathscr{C}$ given by $f_l \times 5/0.3 \times 5$ and $r_{\text{outer}} = 6$. The left control factor

Table 5
Convergence comparison: BCGC against BCGC Standard algorithm ($61 \times 41 - f_l \times 5/0.3 \times 5$)

Inner boundary	0.6×5		0.8×5		1×5		1.2×5		1.4×5		1.6×5	
	S	B	S	B	S	B	S	B	S	B	S	B
Epicycloid-Iters	D	174	D	174	D	187	D	187	D	175	D	170
Astroid-Iters	D	227	D	210	D	190	D	194	D	186	D	187

f_l is adjusted for each experiment. A range of values for f_l, for which the BCGC algorithm converges and the BCGC Standard diverges, is identified. The symbol S stands for the BCGC Standard algorithm. The symbol B stands for the BCGC algorithm. The letter D indicates that the corresponding algorithm is diverging. Thus, it is found that BCGC always converges, while BCGC Standard always diverges, when $f_l \in (0.6, 1.6)$, for the experiments described above. We performed similar experiments for fixed control parameters $0.8 \times 5/0.3 \times 5$ and variable grid sizes. The domain characteristic and other parameters are identical to the epicycloid's experiment in Section 5. It was noticed that the standard approach failed to converge with relatively coarse grids (61×61), while BCGC algorithm converged even for much finer grids (201×201) subject to the same clustering conditions.

8.2. Comparison of BCGC and structured grids generated by the CAE software ANSYS

Structured grids for the astroid annular region, bounded by the astroid curve inside and the circle outside, were generated using the meshing module of the well-known commercial CAE software ANSYS. This module generates grids by using free meshing or mapped meshing options. Free meshing creates unstructured grids in general. ANSYS mapped meshes have regular patterns. Their cells can be quadrilateral distributed by rows. These are the type of structured grids that can be used to approximate the values of a field variable using a finite-difference method in curvilinear coordinates. Mapped meshes are usually constructed dividing the physical region into subregions and then executing the code to generate a subgrid in each subregion. For multiple connected regions as those studied in this work, mapped meshes break at the interfaces between subregions, as shown in the top left graph of Fig. 9. As a consequence, ANSYS mapped meshes are not smooth in the interior.

Control of grid lines in mapped meshes depends on the initial distribution of nodes along the subregion boundaries. However, there is not a practical procedure to control the grid line spacing in the interior. We conducted a grid quality comparison between BCGC and ANSYS meshes of various sizes. The domain chosen was the astroid which includes severe singularities in the form of cusps. Our quality analysis revealed that the, defined in reference [6], of ANSYS mapped mesh cells is about 70% more than BCGC grid cells with control parameters $1.5 \times 7/0.8 \times 3$. These results are shown in Table 6. Therefore, BCGC grids for complex domain configurations are smoother and also more orthogonal than ANSYS mapped meshes of the same size.

Finally, we repeated the experiment of Section 7 using an ANSYS mapped mesh of size 81×41. The numerical solution showed instabilities after 22,000 time steps. The deterioration of the numerical solution is depicted at the left of Fig. 9. At the right of the same figure, a numerical solution of the membrane displacement based on BCGC grids of the same size is shown. As opposed to mapped meshes, a stable periodic oscillation was observed for numerical simulations consisting of more than 100,000 iterations on BCGC grids. This experiment reveals the importance of the smoothness of the grid and the appropriate spacing between grid lines in the vicinity of the singularities in numerical simulation. These are precisely the two factors considered in this work for the construction of BCGC grids.

8.3. Impact of BCGC grids in numerical simulation

Smooth grids with higher density of coordinate curves near the physical boundaries are highly desirable in the computation of fluid properties in the presence of boundary layers [14]. In contrast, for complex geometries such as the astroid and the epicycloid, stretching the distance between neighboring grid lines may be required to avoid self-overlapping near sharp corners or cusps. Also, adjusting the distance between neighbor-

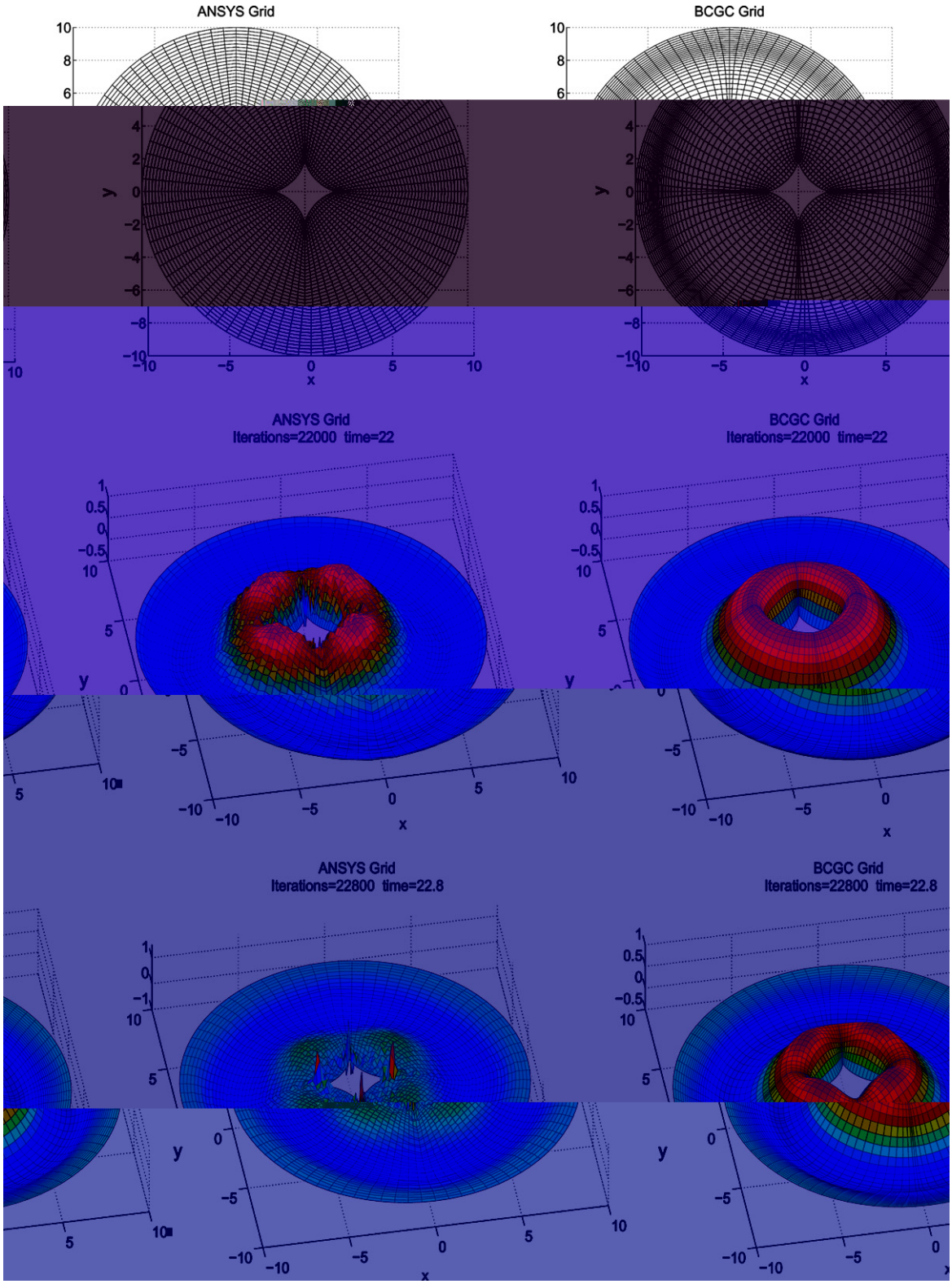

Fig. 9. Vibrating annular membrane using and 81×41 ANSYS mapped grid (left) and a 81×41 BCGC grid with control parameters $2 \times 7/0.3 \times 3$ (right).

Table 6
Comparison of average deviation from orthogonality (ADO) between ANSYS mapped and BCGC grids with control parameters $1.5 \times 7/0.8 \times 3$

Grid type	101×101	161×161	201×201	301×301
Mapped	7.77°	7.98°	8.06°	8.15°
BCGC	4.67°	4.62°	4.61°	4.58°

ing grid lines is especially important when implementing an explicit finite-difference marching in time scheme for wave propagation problems [4,21,20]. As discussed in Section 7, the cell sizes of the grids used in the computation should satisfy the CFL stability condition. By appropriately controlling the grid line spacing in the case of complexly shaped annular membranes (see Section 7), we increased the size of the grids by approximately 70% (141×141 to 241×241) keeping the time step size fixed at $\Delta t = 0.001$, and still satisfying the CFL stability condition. Therefore, the precision can be greatly improved without drastically increasing the computational cost.

Explicit marching in time numerical schemes (Finite-Difference Time-Domain) are very common in the numerical modelling of scattering phenomena [16,13]. Thus, the use of BCGC grids in wave scattering will be of importance in this field. We will discuss the advantage of applying BCGC grids in wave scattering numerical computation in a forthcoming paper.

Acknowledgements

The third author acknowledges support from the Undergraduate Mentorship Program of the Department of Mathematics of Brigham Young University (BYU). The second author is also grateful for the support received from the Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela during his visit to BYU. The authors wish to thank Daniel Villamizar for his valuable help with the figures included in this article.

References

[1] V. Akcelik, B. Jaramaz, O. Ghattas, Nearly orthogonal two-dimensional grid generation with aspect ratio control, J. Comput. Phys. 171 (2001) 805–821.
[2] X. Cai, B. Jiang, G. Liao, Adaptive grid generation based on least-squares finite element method, Computers Math. Applic. 48 (2004) 1077–1085.
[3] J. Castillo, T. McGuinness, Steady state diffusion problems on non-trivial domains: support operator method integrated with direct optimized grid generation, Appl. Numer. Math. 40 (2002) 207–218.
[4] F. Collino, T. Fouquet, P. Joly, Conservative space–time mesh refinement methods for the FDTD solution of Maxwell's equations, J. Comput. Phys. 211 (2006) 9–35.
[5] M.P. do Carmo, Differential Geometry of Curves and Surfaces, Prentice-Hall, Englewood Cliffs, NJ, 1976.
[6] L. Eca, 2D orthogonal grid generation with boundary point distribution control, J. Comput. Phys. 125 (1996) 440–453.
[7] U.K. Kaul, New boundary constraints for elliptic systems used in grid generation problems, J. Comput. phys. 189 (2003) 476–492.
[8] A. Khamayseh, G. Hansen, Quasi-orthogonal grids with impedance matching, SIAM J. Sci. Comput. 22 (2000) 1220–1237.
[9] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, 1993.
[10] P.M. Knupp, N. Robidoux, A framework for variational grid generation: conditioning the Jacobian matrix with matrix norms, SIAM J. Sci. Comput. 21 (2000) 2029–2047.
[11] S.H. Lee, B.K. Soni, The enhancement of an elliptic grid using appropriate control functions, Appl. Math. Comput. 159 (2004) 809–821.
[12] G. Liao, On harmonic maps, in: J.E. Castillo (Ed.), Mathematical Aspects of Numerical Grid Generation, SIAM, Philadelphia, 1991.
[13] E. Manoha, R. Guenanff, S. Redonnet, M. Terracol, Acoustic scattering from complex geometries, in: AIAA (Ed.), 10th AIAA/CEAS Aeroacoustics Conference, AIAA, 2004, pp. 1551–1553.
[14] S.A. Salem, Numerical simulations for the contraction flow using grid generation, J. Appl. Math. Comput. 16 (2004) 383–405.
[15] S.P. Spekreijse, Elliptic grid generation based on Laplace equations an algebraic transformations, J. Comput Phys. 118 (1995) 38–61.
[16] A. Taflove, Advances in Computational Electrodynamics. The Finite-difference Time-domain Method, Artech House, 1998.
[17] P.D. Thomas, J.F. Middlecoff, Direct control of grid point distribution in meshes generated by elliptic equation, AIAA J. 18 (1980) 652–656.

[18] J.F. Thompson, A general three-dimensional elliptic grid generation system on a composite block structure, Comput. Meth. Appl. Mech. Eng. 64 (1987) 377–411.
[19] J.F. Thompson, B.K. Soni, N.G. Weatherill, Handbook of Grid Generation, CRC Press, 1999.
[20] V. Villamizar, O. Rojas, Time-dependent numerical method with boundary-conforming curvilinear coordinates applied to wave interactions with prototypical antennas, J. Comput Phys. 177 (2002) 1–36.
[21] V. Villamizar, M. Weber, Scattering cross section of non-smooth cylindrical obstacles of arbitrary shape, in: H. Haddar, J.H. Hesthaven, (Eds.), in: Proceedings of the 7th International Conference on Mathematical and Numerical Aspects of Wave Propagation, Waves 2005, pp. 167–169, Rhode Island, 2005.
[22] M.R. Visbal, D.V. Gaitonde, On the use of higher-order finite difference schemes on curvilinear and deforming meshes, J. Comput. Phys. 181 (2002) 155–185.
[23] C.L. Wang, K. Tang, Non-self overlapping hermite interpolation mapping: a practical solution for structured quadrilateral meshing, Comput. Aided Des. 37 (2005) 271–283.