

GENERATION OF 2-D PARAMETRIC SURFACES WITH HIGHLY IRREGULAR BOUNDARIES

*Subhajit Sarkar and Partha Pratim Dey**

Department of Mechanical Engineering, Bengal Engineering and Science University, Shibpur, India

Abstract

The conventional methods of boundary-conformed 2D surfaces generation usually yield some problems. This paper deals with two boundary-conformed 2D surface generation methods, one conventional approach, the linear Coons method, and a new method, boundary-conformed interpolation. In this new method, unidirectional 2D surface has been generated using some of the geometric properties of the given boundary curves. A method of simultaneous displacement of the interpolated curves from the opposite boundaries has been adopted. The geometric properties considered for displacements include weighted combination of angle bisector and linear displacement vectors at all the data-points of the two opposite generating curves. The algorithm has one adjustable parameter that controls the characteristics of transformation of one set of curves from its parents. This unidirectional process has been extended to bi-directional parameterization by superimposing two sets of unidirectional curves generated from both boundary pairs. Case studies show that this algorithm gives reasonably smooth transformation of the boundaries. This algorithm is more robust than the linear Coons method and capable of resolving the 2D boundary-conformed parameterization problems.

Keywords: 2D surface, boundary-conformed surface, linear Coons method;

1. Introduction

Surfaces and their descriptions play an important role in design and manufacturing. In CAD/CAM, the mathematical models of surfaces are usually represented as parametric patches. In a variety of science and technological problems, 2D surfaces need to be parameterized according to the predefined boundaries. A number of different parametric representations of free-form surfaces have been intensively studied and developed by many researchers so far [1-11, 13-15]. The most common method of free-form surface generation has been boundary interpolation [6, 7]. In this case, the available conventional technique is the linear Coons method [2]. Application of this existing method to highly irregular boundaries yields some shortcomings. In the earlier work by Yang D.C.H. et. al. [14], the anomalies of Coons method were addressed and a solution was proposed by using normal offset vector at

* Corresponding author:

Tel: +91-9330965313

Fax: +91-33-26682916

Email: ppdey2000@yahoo.com

every data point on the curves to avoid overcrossing and a curvature-averaging function to control the spacing between interpolated curves and crowdedness. The Hermite Interpolation Mapping (HIM) scheme has been discussed by Wang & Tang [11] for generation of quadrilateral grid in a given 2D region. The conventional HIM gives the problem of self-overlapping grids when boundaries are concave and convoluted. This problem has been resolved by adjusting the tangent functions and the twist vectors through a constrained functional optimization scheme in three stages. Wang & Tang [10] discussed a structured grid generation method for a 2D region bounded by any number (n-sided) of parametric boundary curves with C^1 continuity. The initial algebraic grids have been constructed through Gregory patch mapping and the self-overlapping of resulting grids has been eliminated by functional optimization. Wang & Tang, in their work [9], introduced adaptive non-trivial blending functions in Coons patch, which have been determined by functional optimization method to avoid the self-overlapping of Coons patch.

In the present work, a study on the parameterization of 2D patches with highly irregular boundaries has been presented in terms of both the conventional and new methods. A new approach called *boundary-conformed interpolation* method has been introduced. Applicability and drawbacks of these two methods have been examined and a comparative study has been carried out on some typical cases. The conventional method for 2D surface parameterization has been discussed in sub section 1.1. The new boundary-conformed interpolation method has been discussed in detail in sections 2 & 3 and some case studies have been shown in section 4.

1.1 The linear Coons method

In linear Coons method, a surface patch $S(u,v)$ enclosed by four known boundary curves as shown in Fig. 1, where $u, v \in [0,1]$. Let $\mathbf{B}(u,0)$, $\mathbf{B}(u,1)$, $\mathbf{B}(0,v)$ and $\mathbf{B}(1,v)$ are the four boundary curves and $\mathbf{P}(0,0)$, $\mathbf{P}(1,0)$, $\mathbf{P}(0,1)$, and $\mathbf{P}(1,1)$ be the position vectors of the four corner points. A bilinear blending function is used for the interior of the surface patch. The bilinear Coons surface representation [2, 4, 10, 11] of $S(u,v)$ in terms of the four boundary curves and the four corner points is:

$$\begin{aligned} \mathbf{S}(u,v) = & [(1-u) \quad u] \begin{bmatrix} \mathbf{B}(0,v) \\ \mathbf{B}(1,v) \end{bmatrix} + [\mathbf{B}(u,0) \quad \mathbf{B}(u,1)] \begin{bmatrix} (1-v) \\ v \end{bmatrix} \\ & - [(1-u) \quad u] \begin{bmatrix} \mathbf{P}(0,0) & \mathbf{P}(0,1) \\ \mathbf{P}(1,0) & \mathbf{P}(1,1) \end{bmatrix} \begin{bmatrix} (1-v) \\ v \end{bmatrix} \end{aligned} \quad (1)$$

The linear Coons method can generate reasonably good 2D surface for relatively simple boundaries, as shown in Fig. 2.(a), but, this method gives anomalies like skew, crowdedness and overcrossing of the interpolated curves if the irregularities of given 2D boundaries are relatively high, as

shown in Fig. 2.(b).

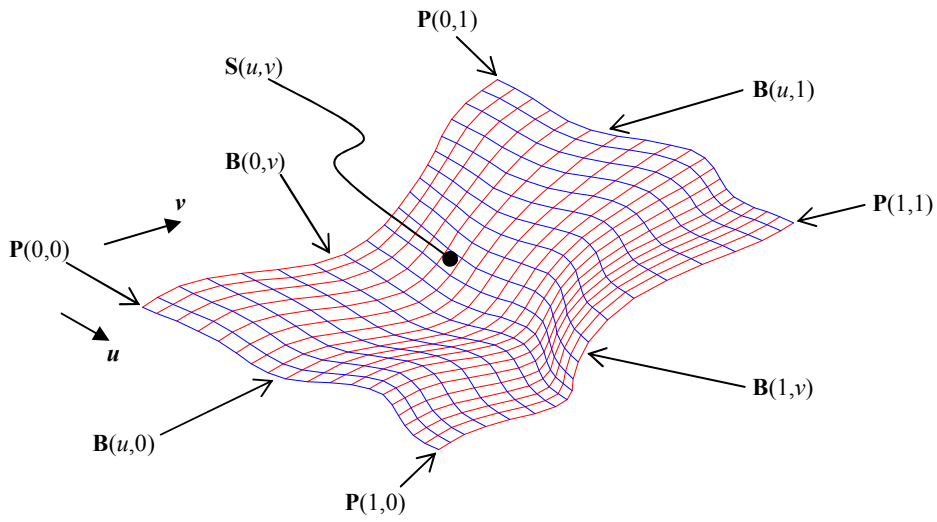


Fig. 1. Representation of a Bi-linear Coons surface patch

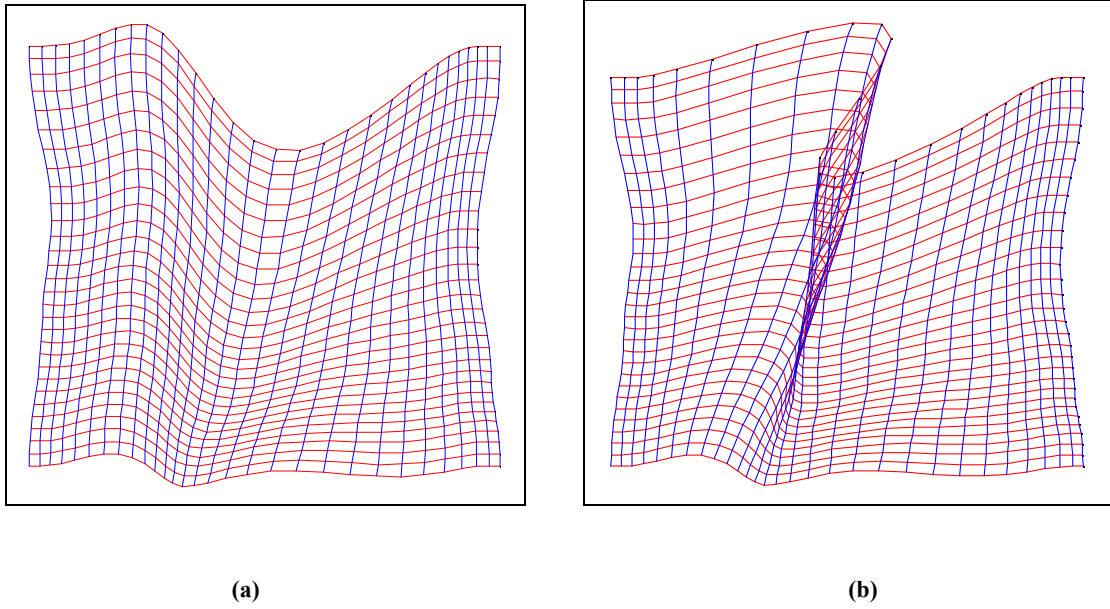


Fig. 2. Bi-linear Coons patch (a) with simple boundaries (b) with irregular boundaries

2. Boundary-Conformed Interpolation Method

This is the new method introduced here to resolve the anomalies faced in the linear Coons method as stated in section 1.1. The intermediate curves have been generated by considering the position

and angle bisector vectors at all the data-points of the two opposite generating curves. In the following sections the detail of developing the algorithm for this new method has been discussed.

2.1 Unidirectional displacement

The boundary-conformed interpolation method will now be presented for unidirectional cases. In case of unidirectional surfaces, the intermediate curves are interpolated by only one pair of opposite boundary curves and the directions of displacement of the interpolated curves are defined as the direction from one of the opposite boundaries to the other. In linear Coons method, a linear blending function is used to interpolate the opposite boundary pair. Because of linear blending function, every time the linear Coons method calculates an intermediate point (between the opposite boundaries) by interpolating only a pair of points each on one of the two opposite boundaries and hence it is insensitive to the shape of the boundary curves. This is the main reason that causes parameterization anomalies when the irregularity of the boundaries is high in linear Coons method. Hence, in boundary-conformed interpolation algorithm the angle bisector vectors of both boundaries have been considered in addition to position vectors as discussed in the following sections.

Before detailing the new algorithm an overview is presented to look deeply into the plotting mechanism of any curve. A curve is always plotted by joining a finite number of data-points, as calculated by the concerned algorithm, with straight lines between each pair of successive data-points as shown in Fig. 3. So, on a curve every set of three consecutive data-points constructs an angle at the mid-point of the set and this angle can always be bisected to form an angle bisector vector.

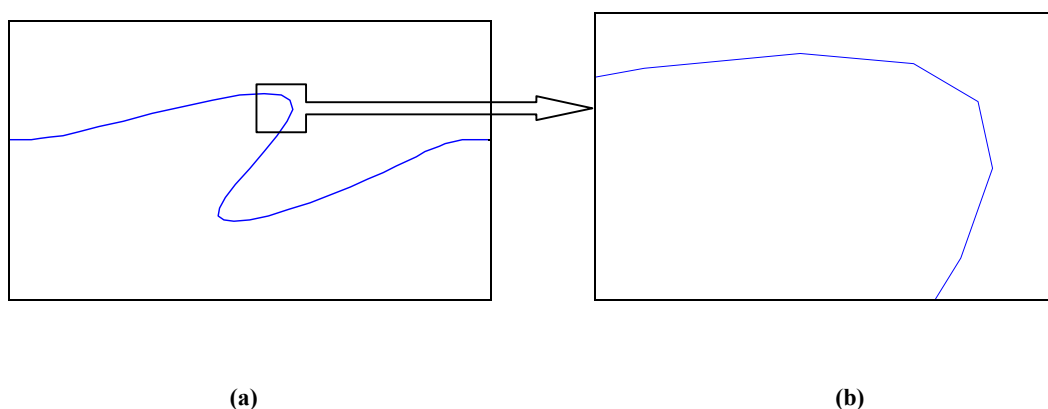


Fig. 3. A Bézier curve (a) the entire curve visually smooth (b) magnification shows the curve is composed of multiple straight lines

In light of the above phenomenon the interpolated curves can be represented as functions of four

geometric parameters,

$$\mathbf{C}_j = f(\mathbf{B}_0(u), \mathbf{B}_1(u), \mathbf{b}_0(u), \mathbf{b}_1(u)) \quad (2)$$

Here, \mathbf{B}_0 and \mathbf{B}_1 is a pair of opposite boundary curves or generating curves, and \mathbf{b}_0 and \mathbf{b}_1 are the angle bi-sectors at all the intermediate data-points of curves \mathbf{B}_0 and \mathbf{B}_1 respectively. At any of the two end data-points of a curve, it has been assumed that the curve is infinitesimally extended with the same slope as that of the end segment of the curve and hence the angle bisectors at the end data-points are normal to the curve at the respective points. The new sets of curves are generated by simultaneous interpolation from the two opposite boundary curves. This simultaneous displacement process continues till the entire surface is covered. So the entire surface is treated as if divided into two halves. Therefore, basic algorithm is denoted as

$$\mathbf{C}_j(u) = \mathbf{C}_{j-1}(u) + \mathbf{d}_j(u) \quad (3a)$$

$$\bar{\mathbf{C}}_j(u) = \bar{\mathbf{C}}_{j-1}(u) + \bar{\mathbf{d}}_j(u) \quad (3b)$$

where $\mathbf{d}_j(u)$ and $\bar{\mathbf{d}}_j(u)$ are displacement vectors; $j = 1 \dots n$, and n is the total number of interpolated isoparametric curves on each half of the surface.

$\mathbf{C}_j(u)$ and $\bar{\mathbf{C}}_j(u)$ are the interpolated curves in 'v' direction.

Let m (which is user defined) be the total number of curves including the boundaries, to be generated on the entire surface, then,

$$n = (m/2) - 1 \quad \text{when } m \text{ is even} \quad (4a)$$

$$n = ((m+1)/2) - 1 \quad \text{when } m \text{ is odd} \quad (4b)$$

The generalized form of the interpolated isoparametric curves can be given by,

$$\mathbf{C}_j(u) = \mathbf{P}(u, v_j); \quad \text{where } u \in [0,1] \quad \& \quad v_j = j/(m-1) \quad (5a)$$

$$\text{and } \bar{\mathbf{C}}_j(u) = \mathbf{P}(u, \bar{v}_j); \quad \text{where } u \in [0,1] \quad \& \quad \bar{v}_j = 1 - j/(m-1) \quad (5b)$$

2.2 Design of the displacement vector

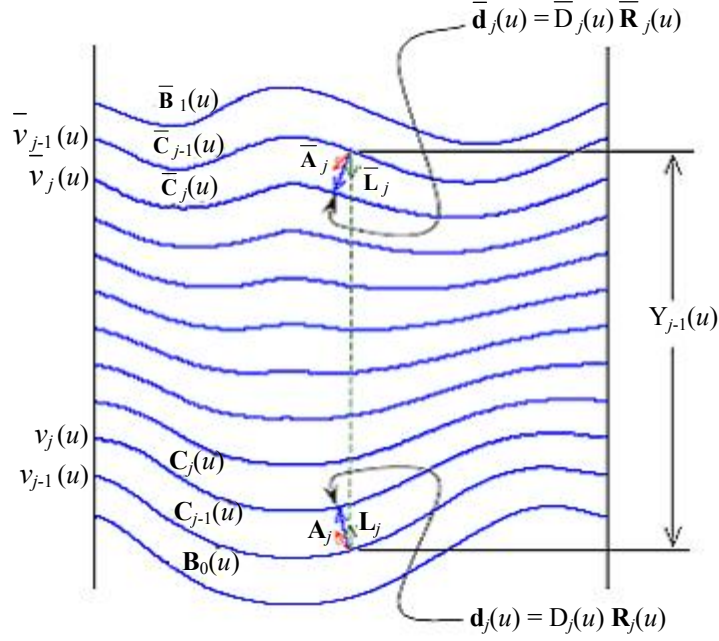


Fig. 4. Displacement vector and its constituents

Referring to Fig. 4, $\mathbf{B}_0(u)$ and $\mathbf{B}_1(u)$ are the two boundary curves and between them the intermediate curves are to be interpolated. From $\mathbf{B}_0(u)$ and $\mathbf{B}_1(u)$ the new sets of interpolated curves $\mathbf{C}_j(u)$ and $\bar{\mathbf{C}}_j(u)$ will be generated, where $j = 1 \dots n$, by moving their respective displacement vectors towards each other. Since this displacement process is symmetrical, as depicted in Eq. (3), the formulation of displacement vectors $\mathbf{d}_j(u)$ and $\bar{\mathbf{d}}_j(u)$ will be same. Therefore, the formulation of $\mathbf{d}_j(u)$ only will be detailed.

Let

$$\begin{aligned} \bar{\mathbf{d}}_j(u) &= \bar{D}_j(u) \bar{\mathbf{R}}_j(u) \\ \mathbf{d}_j(u) &= D_j(u) \mathbf{R}_j(u) \end{aligned} \quad (6)$$

where $D_j(u)$ and $\mathbf{R}_j(u)$ represent the displacement magnitude and the unit displacement direction of $\mathbf{d}_j(u)$ respectively. The explanation of design and development of the displacement vector $\mathbf{d}_j(u)$ have been presented in the following subsections.

2.2.1 Determining the displacement magnitude

Referring to Fig. 4, the displacement magnitude can be calculated as

$$D_j(u) = \frac{Y_{j-1}(u)}{2(n-j)+2} \quad \text{for odd number of curves } m \quad (7a)$$

$$D_j(u) = \frac{Y_{j-1}(u)}{2(n-j)+3} \quad \text{for even number of curves } m \quad (7b)$$

where $Y_{j-1}(u)$ is the distance between $C_{j-1}(u)$ and $\bar{C}_{j-1}(u)$

2.2.2 Designing a linear displacement direction

Now if the displacement direction is considered as a linear vector, then,

$$\mathbf{R}_j(u) = \frac{\bar{C}_{j-1}(u) - C_{j-1}(u)}{\|\bar{C}_{j-1}(u) - C_{j-1}(u)\|} = \mathbf{L}_j(u) \quad (8)$$

where $\mathbf{L}_j(u)$ represents the unit linear displacement vector. This algorithm has been applied in two sample cases as shown in Fig. 6.(a), which has relatively simple boundaries whereas boundaries in Fig. 6.(b) are much more irregular. The results are quite good for simple boundaries as shown in Fig. 6.(a)-I, but when the boundaries are more irregular this algorithm fails due to the anomalies like crowdedness and over-crossing (Fig. 6.(b)-I.)

2.2.3 Considering an angle bisector displacement direction

To resolve the problem of over-crossing of the curves, the displacement direction could be defined by angle bisector vectors at all the data-points (except endpoints where normal vectors are to be taken as discussed in 2.1) of the two opposite generating curves.

Referring to Fig. 3.(b) it is evident that on a curve every set of three consecutive data-points can make a triangle and a vector directing towards the incentre of the triangle from the mid-data-point of the concerned set of three data-points constructs the angle bisector vector of the included angle at the mid-data-point. If three consecutive data-points lie on the same straight line then the angle bisector vector becomes the normal to the straight line at the mid-data-point and hence normal vector is to be calculated at that point. The directions of these angle bisectors have to be checked and changed by rotating the

vectors 180 degree so that these vectors should always be directed towards opposite boundary curve. Direction checking criteria would be whether the value of the included angle, as shown in Fig. 5, at the mid-point of a set of three consecutive data-points is greater than 180 degree; e.g., for the upper half of the surface each intermediate curve should move downwards that is towards the lower boundary, hence if the included angle is greater than 180 degree then the direction of angle bisector vector is to be changed through 180 degree. Converse is true for the lower half.

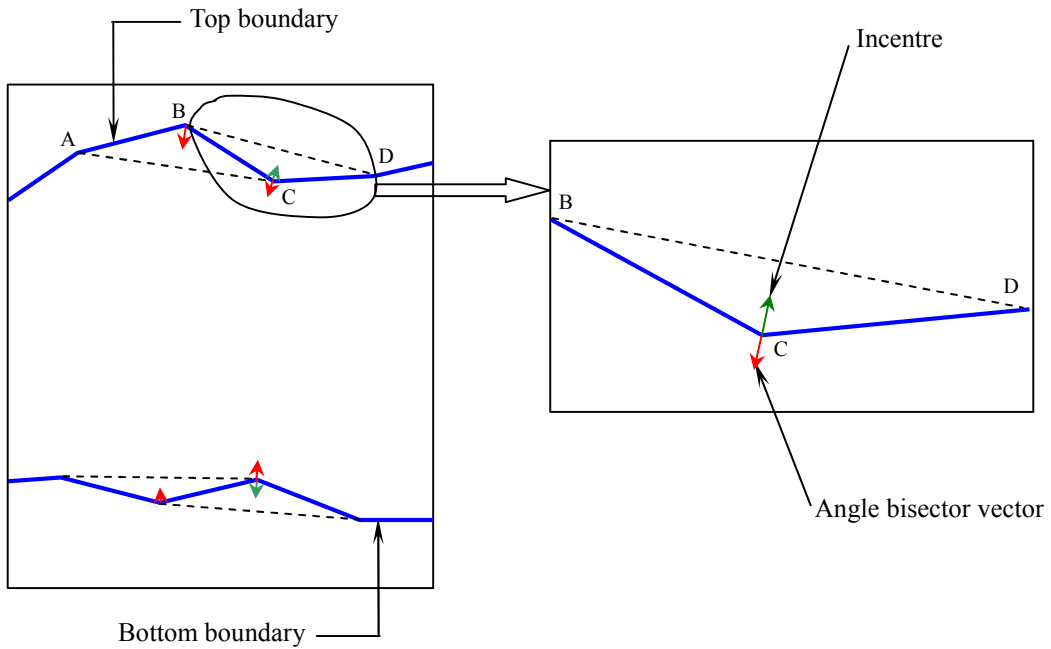


Fig. 5. Included angle and angle bisector vectors (magnified)

So with this consideration let the displacement direction be based solely on the angle bisector directions of the parent curves, i.e.,

$$\mathbf{R}_j(u) = \mathbf{A}_j(u) \quad (9)$$

where $\mathbf{A}_j(u)$ represents the unit angle bisector vector. This new algorithm has been applied to the same sample cases as plotted in Fig. 6.(a)-II & 6.(b)-II. For simple boundaries this algorithm provides the complete solutions but for irregular boundaries, though this algorithm gives the solution for the overcrossing problem, but does not give the smooth blending of the curves approaching each other at the middle.

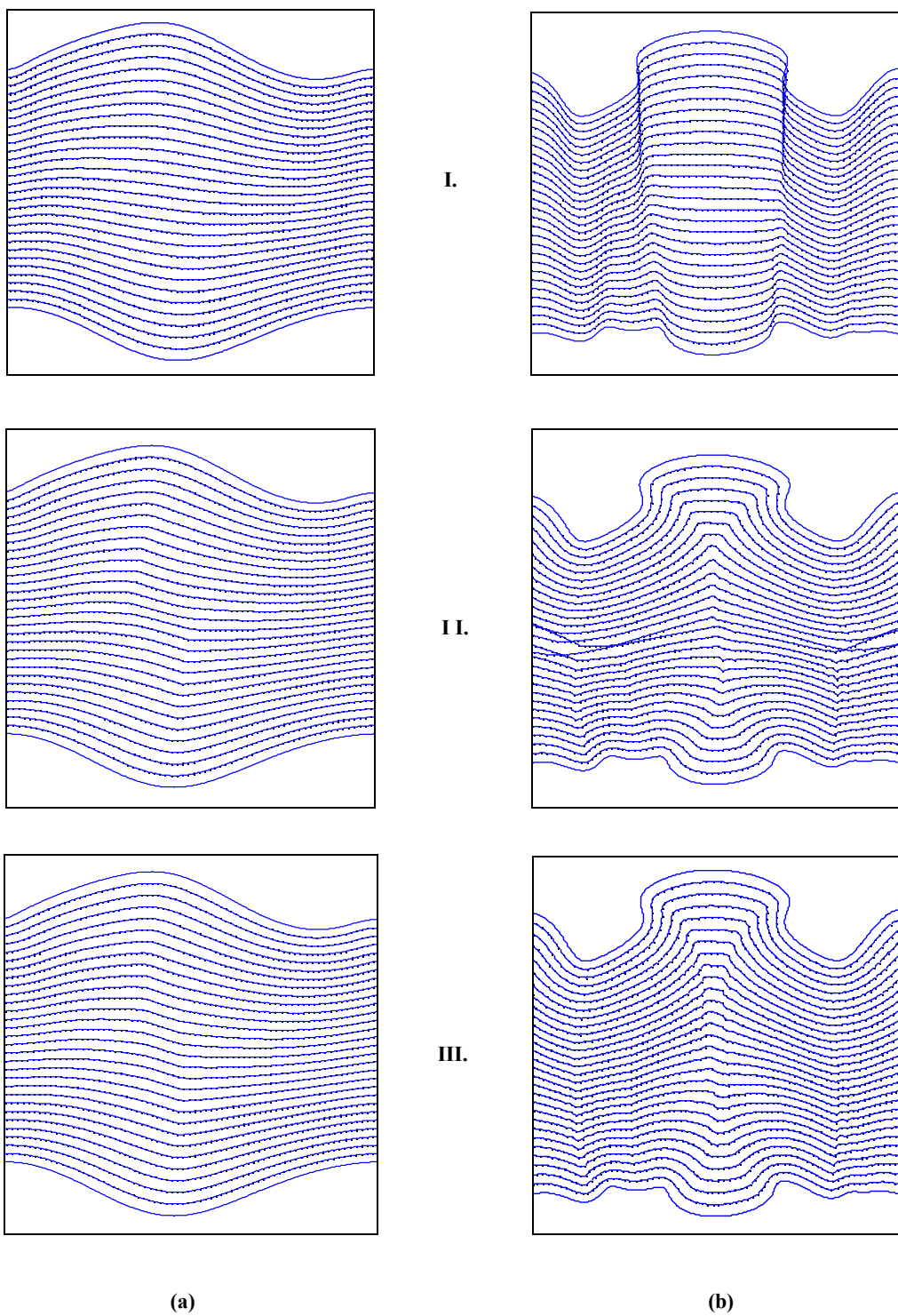


Fig. 6. Two examples for the development of displacement vector (a) with simple boundaries (b) with highly irregular boundaries

2.2.4 Combining the displacement directions

The total displacement vector may be expressed as the resultant of linear and angle bisector vectors multiplied each by a weight function. The weight functions ensure that as the interpolated curves move towards middle of the surface, the effect of angle bisectors on the resultant vector reduces and the influence of linear displacement vectors increases. This confirms the smooth blending of two facing curves at the middle.

Thus the unit displacement vector with weighted combination of linear and angle displacement vectors is designed as

$$\mathbf{R}_j(u) = \frac{(1-f) \mathbf{A}_j(u) + f \mathbf{L}_j(u)}{\|(1-f) \mathbf{A}_j(u) + f \mathbf{L}_j(u)\|} \quad (10)$$

where,

$$f = \left(\frac{j}{n} \right) \left(\left(1 - \left(\frac{j}{n} \right) \right)^j \right)^{(p/100)} \quad (11)$$

Here, f is a blending function. The weight functions $(1-f)$ & f have been constructed in such a way that the emphasis on angle bisector vector will be more near the boundaries and gradually diminishes towards the middle and consequently the linear vector will be emphasized less near boundaries and more at the middle.

The characteristics of $(1-f)$ and f have been plotted in Fig. 7. The user defined parameter p can be set to different values (*a real number* ≥ 0) to control the speed of shifting of weight from angle bisector to linear vector in Eq. (10). The larger the value of p faster the shifting will be. The value of p may be taken smaller or larger as it suits to the surface. Now as shown in Fig. 6.(b)-III., where $p=0.5$ has been taken, the earlier problems have completely been resolved by this weighted combination of angle bisector and linear vectors.

The effects of different values of p on the interpolated curves for the same boundary pairs have been tested with yet another example as shown in Fig. 8.

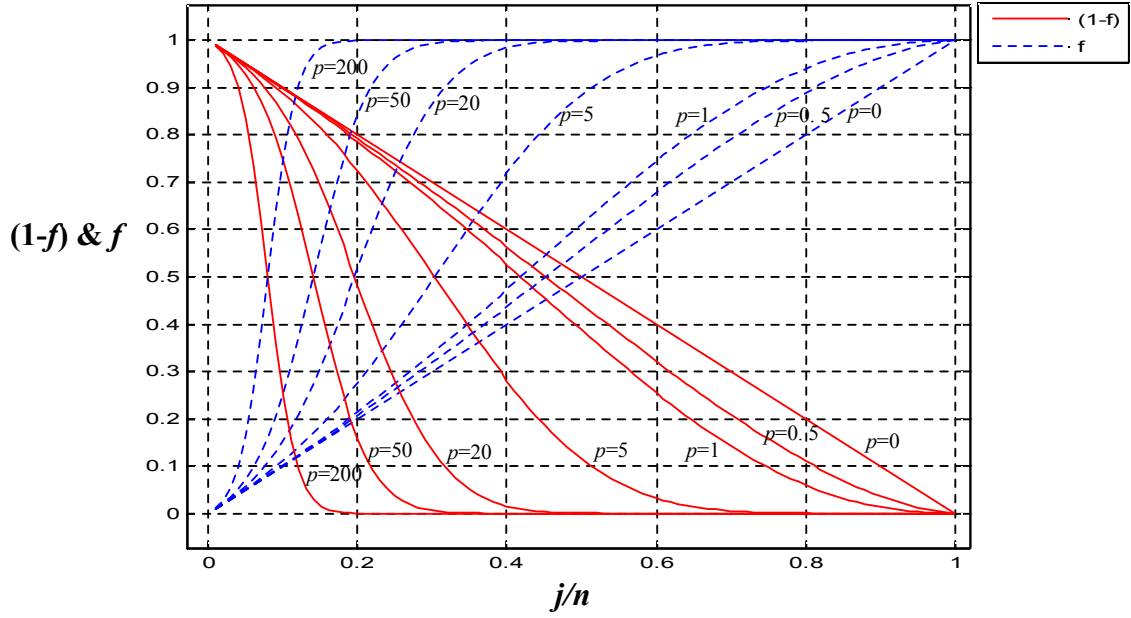


Fig. 7. Shifting of weights between angle bisector & linear vectors with the position of interpolated curve for different values of p

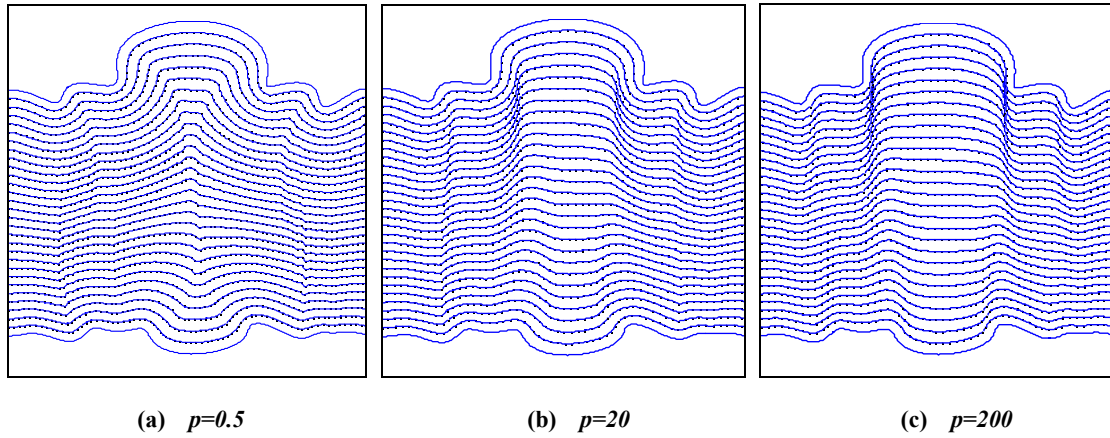


Fig. 8. Effect of different values of user defined parameter p

2.3 The complete algorithm of boundary-conformed interpolation

The complete algorithm for boundary-conformed interpolation method can now be summarized as:

$$\mathbf{C}_j(u) = \mathbf{C}_{j-1}(u) + \mathbf{d}_j(u) \quad (12)$$

$$\mathbf{d}_j(u) = \mathbf{D}_j(u) \mathbf{R}_j(u) \quad (13)$$

where,

$$D_j(u) = \frac{Y_{j-1}(u)}{2(n-j)+2} \quad (14)$$

$$n = ((m+1)/2)-1 \quad (15)$$

$$D_j(u) = \frac{Y_{j-1}(u)}{2(n-j)+3} \quad (16)$$

$$n = (m/2)-1 \quad (17)$$

$$\mathbf{R}_j(u) = \frac{(1-f) \mathbf{A}_j(u) + f \mathbf{L}_j(u)}{\|(1-f) \mathbf{A}_j(u) + f \mathbf{L}_j(u)\|} \quad (18)$$

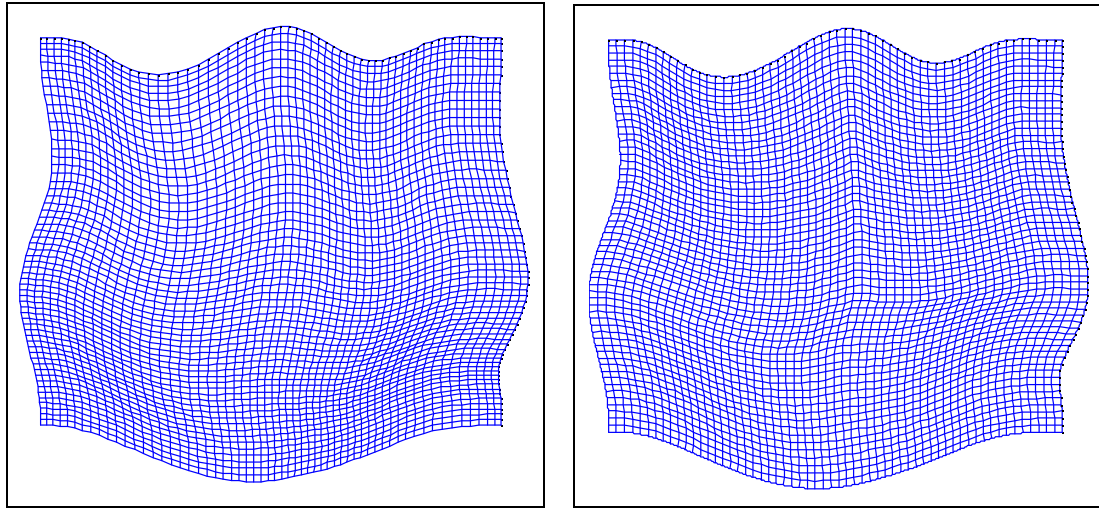
$$f = \left(\frac{j}{n} \right) \left(\left(1 - \left(\frac{j}{n} \right) \right)^j \right)^{(p/100)} \quad (19)$$

3. Bi-directional surface

The bi-directional surface can be generated using the same unidirectional mechanism. Each pair of opposite boundaries can be treated separately as a unidirectional problem. A bi-directional 2D surface is nothing but the algebraic summation of two unidirectional surfaces minus the duplicating surface, as in case of conventional method.

4. Case studies — results and comparisons

Five more cases have been presented in this section to examine and compare the two 2D parameterization methods presented in this paper. Both the methods have been written in *MATLAB7*. In Fig. 9, 10, 11 12 & 13 the results have been given side by side for comparing the capabilities of these two methods. From Case-I to Case-V, the irregularities of the boundary curves have been increased gradually to show the shortcomings of linear Coons method.

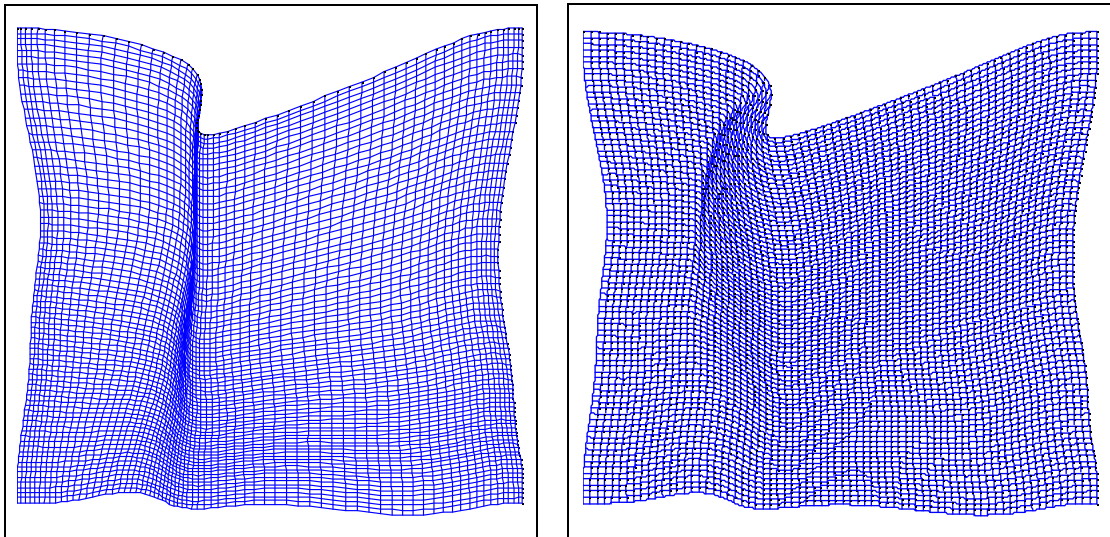


(a) Linear Coons method

(b) Boundary-conformed interpolation method

Fig. 9. Case – I

Case-I is composed of simple boundaries and the outcomes of both the linear Coons and boundary conformed interpolation methods are satisfactory as evident from Fig. 9. It has 60x60 grids and for both horizontal and vertical direction, p equals to 1.



(a) Linear Coons method

(b) Boundary-conformed interpolation method

Fig. 10. Case – II

Case-II, as shown in Fig. 10., is a surface of 72x80 grids and the values of parameter p for both horizontal and vertical directions have been made equal to 0. Here only the top boundary is irregular and all the other three boundaries are comparatively simple. As a result, linear Coons method shows the problem of overcrossing, though the boundary-conformed interpolation method gives a satisfactory result by resolving the overcrossing problem.

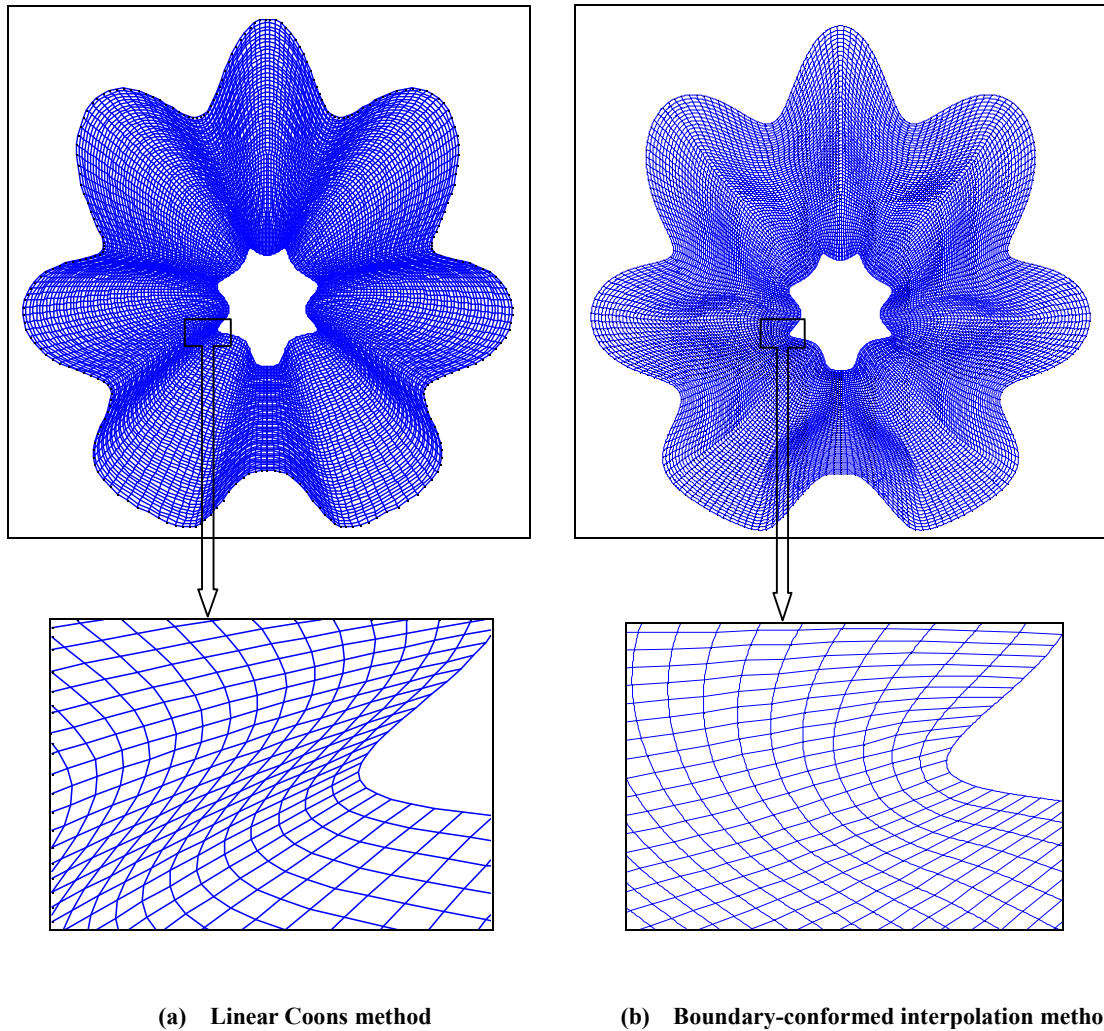


Fig. 11. Case – III

Case-III in Fig. 11., is a clear example of versatility of boundary-conformed interpolation method. It has 250x60 grids and the values of parameter p for horizontal and vertical directions are 0 and 1 respectively. Here two extremely irregular boundaries have been used. It is clear from the results the linear Coons method produce severe crowdedness problem at the portions where irregularities of boundaries are too high, but boundary-conformed interpolation method gives well-distributed grids

throughout the surface, which is evident from the enlarged sectional views.

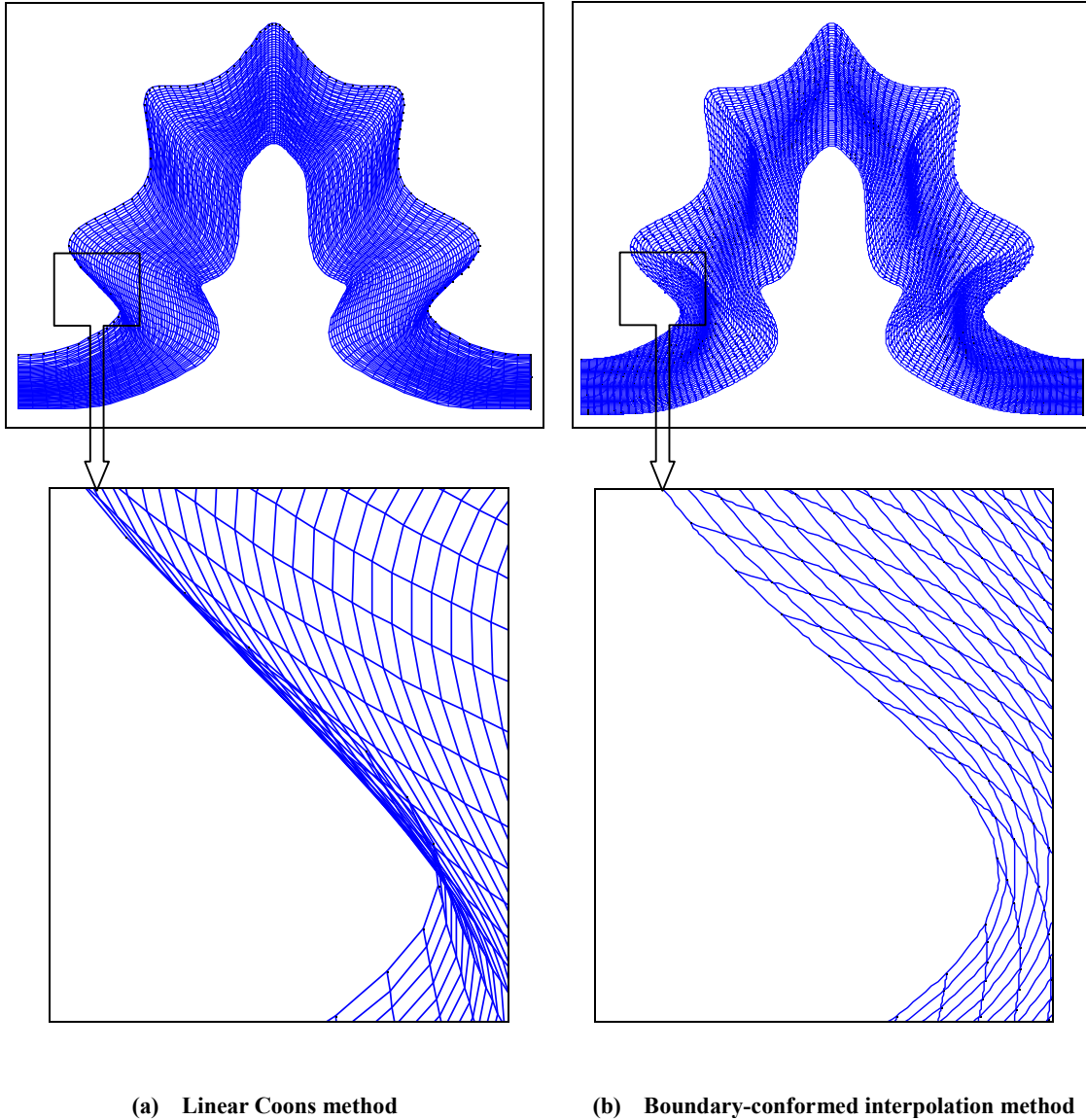
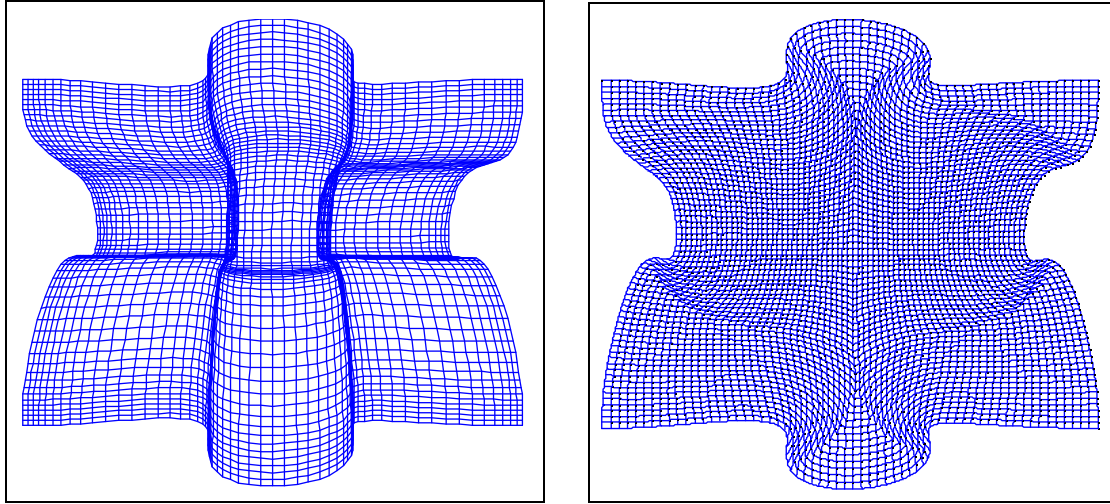


Fig. 12. Case - IV

Case-IV, as shown in Fig. 12., is a surface of 130x60 grids. The value of parameter p for horizontal direction is 100 and vertical direction is 1. Here the top & the bottom boundaries are highly irregular and the other two boundaries are simple. It is clear from the enlarged views of the part of the corresponding surfaces, the linear Coons method shows the problem of self-overcrossing and the intermediate mesh goes outside the specified boundary, though the boundary-conformed interpolation method gives a satisfactory result by resolving both the problems.



(a) Linear Coons method

(b) Boundary-conformed interpolation method

Fig. 13. Case - V

Case-V is a surface composed of all the four highly complex boundaries as shown in Fig. 13. The differences in results of the two methods are visibly clear. Linear Coons method yields the problems of severe overcrossing and crowdedness whereas the boundary-conformed interpolation method provides a very good solution of all these problems. The value of parameter p for horizontal direction is 2 and vertical direction is 0. It has 70x69 grids, which show the consistency of the algorithm of boundary-conformed interpolation method for both odd and even number of grids generations.

The above case studies show that boundary-conformed interpolation method is capable of resolving the shortcomings, which encountered by linear Coons method, in generating 2D surfaces with high degree of irregular boundaries to a very satisfactory level.

5. Conclusions

In this work, a conventional approach to generate 2D surfaces viz. the linear Coons method, has been reviewed. A new method, called *boundary-conformed interpolation* has been developed to resolve the problem of overcrossing, skewness and crowdedness, which are observed in linear Coons method. In this method, simultaneous displacement of the interpolated curves from the opposite boundaries has been adopted to ensure smooth transition of the geometric properties from the boundary curves. The geometric properties considered for displacements include algebraic summation of weighted linear and angle bisector vectors at all the data-points of the two opposite generating curves. The algorithm has one

adjustable (user defined) parameter ' p ' that controls the characteristics of transformation of one set of curves from its parents.

The new algorithm involves combination of two different vectors; one is the linear displacement vector as used in Coons method and another is the angle bisector vector. The linear displacement vector does not guarantee the non-overcrossing (i.e. an interpolated curve may intersect the others) of the interpolated curves but the angle bisector vector ensures that there will be no overcrossing i.e. an interpolated curve will not intersect the others though there is possibilities of self-intersection (an interpolated curve may intersect itself). Here *Boundary-conformed interpolation* method involves some heuristics in determining the displacement magnitude $D_f(u)$ to avoid self-intersection. Displacement magnitude is nothing but the distance between the two consecutive curves on the patch. Therefore $D_f(u)$ can be controlled by varying the number of intermediate curves to be generated. So more the number of intermediate curves less will be the magnitude of $D_f(u)$ and hence eliminating the chances of self-intersection of interpolated curves. Therefore the mesh size (i.e. the number of curves including boundaries in each direction on the patch, which is user defined) should be determined heuristically. From the experiences gathered through different case studies it has been noticed that, when there is small curvature (i.e. very sharp irregularity) involved at any point on any one of a set of opposite boundaries then the number of interpolated curves should be made large enough to avoid self-intersection in that particular direction. This will take care of the fact that the magnitude of displacement $D_f(u)$ should be smaller than the radius of curvature of the given curve. The angle bisector vector solely does not ensure a smooth blending of the two sets of interpolated curves approaching each other at the middle of the surface, which on the other hand, linear displacement vector can ensure. So the weighted combination of these two vectors has been used. Now by giving a suitable value of the user defined parameter ' p ', the crowdedness of the resultant grid can be controlled.

Different case studies show that this algorithm gives reasonably smooth transformation of the intermediate curves from irregular boundary curves. This new method can be relied upon for resolving the 2D boundary-conformed parameterization problems.

References

1. Bézier P. E. (1970), Examination of an Existing System in the Motor Industry: The Unisurf System, *Proc. Roy. Soc. (London)*, Vol. A321, pp. 270-218.
2. Coons S. A. (1964), Surfaces for Computer-Aided Design of Space Figures, *MIT ESL 9442-M-139*.
3. Farin Gerald (1988), *Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide*, Academic Press, Inc.
4. Mortenson Michael E. (1985), *Geometric Modeling*, John Wiley & Sons, Inc.
5. Peter G. D. (1973), *Interactive Computer Graphics Application of the Bicubic Parametric Surface to Engineering Design Problems*, McDonnell Douglas Automation Company, St. Louis, Missouri, presented at SIAM 1973 National Meeting, Hampton, Va., 18-21.
6. Smith Robert E. (1983), *Three-Dimensional Algebraic Grid Generation*, AIAA 6th Computational Fluid Dynamics Conference, Danvers, Massachusetts.
7. Soni Bharat K. (2000), Grid Generation: Past, present, and future, *Appl. Numer. Math.*, pp 361–369.
8. Szilva'si-Nagy M. and Szabo I. (2006), Generalization of Coons' construction, *Computers & Graphics, Elsevier*, pp 588–597.
9. Wang C. C. L. and Tang K. (2004), Algebraic grid generation on trimmed parametric surface using non-self-overlapping Coons patch mapping, *International Journal for Numerical Methods in Engineering*, vol.60, no.7, pp.1259-1286.
10. Wang C. C. L. and Tang K. (2004), Non-self-overlapping structured grid generation on an n-sided surface, *International Journal for Numerical Methods in Fluids*, vol.46, no.9, pp.961-982.
11. Wang C. C. L. and Tang K. (2005), Non-self-overlapping Hermite interpolation mapping: a practical solution for structured quadrilateral meshing, *Computer-Aided Design*, vol.37, no.2, pp.271-283.
12. Westgaard G. and Nowacki H. (2001), Construction of Fair Surfaces Over Irregular Meshes, *ASME, J. Mech. Des.*, Vol. 1, pp 376-384.
13. Yamaguchi Fujio (1988), *Curves and Surfaces in Computer Aided Geometric Design*, Springer-Verlag.
14. Yang D. C. H., Chuang J. J. and Oulee T. H. (2003), Boundary conformed toolpath generation for trimmed free-form surface, *Computer-Aided Design*, vol.35, no.2, pp.127-139.
15. Zhang J. J. and You L. (2004), Analytical C2 smooth blending surfaces, *Future Generation Computer Systems, Elsevier*, pp 1317–1326.