COMPUTER-AIDED DESIGN & APPLICATIONS, 2017
VOL. 14, NO. 1, 95–106
http://dx.doi.org/10.1080/16864360.2016.1199759

G1 continuous bifurcating and multi-bifurcating surface generation with B-splines

Varun Asthana and Amba D. Bhatt

Motilal Nehru National Institute of Technology, India

ABSTRACT

As an extension to the work done by various authors, this paper proposes a new methodology to construct bifurcating surface by using only a single open B-spline surface equation, which is contrary to conventional approach of stitching surface patches. The paper emphasizes on G1 continuity at the junction of two branches as well as on overall G1 continuity of the generated surface, by exploiting the concept of disjoint surface through knot value repetition. The proposed method allows the flexibility to have irregular non-uniform cross-sections in the generated surface. The ends of the dichotomous surface generated can also be so structured that when the algorithm if used in iteration, will produce multi-level of bifurcation with the overall surface to be G1 continuous, at the sub-division(s) of a branch into further branches.

KEYWORDS

Dichotomous branching; bifurcation; multi-bifurcation; surface generation

1. Introduction

Surface construction from cross sectional data, either scanned or manually determined, is a designing process used in various applications like automobile parts (manifold of engine's cylinder head), neuron tree construction [2], human vascular system, human bronchial tree [4], tree skeletal structure [7]; and many more. Most of the constructions cited above involve branching at either single level or multilevel. As in case of manifold design of an engine, bifurcation suffices the design requirements while bronchial tree construction involves bifurcation and sub branching. This work put forwards an approach to construct bifurcating shapes by use of only a single open B-spline surface equation which offers the flexibility to have irregular non-uniform cross-sections with asymmetrical branching in the generated surface. In this work dichotomous branching surface and bifurcating surface are used interchangeably.

1.1. Previous related work

In past many authors have addressed the problem of modeling of branched surfaces with varying techniques. The methods used by other authors for the generation of bifurcating surface includes method of triangulation [6], [8], stitching of right circular cylindrical surface patches [3], [11], blending of half tubular Bezier patches [12], skinning and trimming of surface [5], and few others. Each had its own limitations which has been countered by the algorithm proposed in this work. With method of triangulation, continuity requirements could not be met easily while the stitching of surface patches require an additional step of aligning tangential vectors of the corresponding stitched patches. The model proposed in this work is neither limited to circular cross-section nor uniform cross section, and also B-spline surface offers better control than the Bezier surface. This work subdues the complexities confronted in the modeling of dichotomous branching shapes by using disjoint open B-spline surface, along with maintaining G1 continuity at the junction of two branches. Bhatt et al. [1] used a similar technique of disjoint surface, used in this work for generation of bifurcating surfaces using single B-spline surface equation but was limited to order 3 in one of the two defining directions of B-spline surface. The algorithm used in this work uses a different arrangement of control points and a different technique for disjoining the surface than used in [1], which led to a model capable of handling any order of B-spline in both of the defining directions; along with an additional benefit of using the algorithm in iteration to produce multiple bifurcating surface. Authors have also used T-splines [10] to efficiently join two or more B-spline surfaces together, having different knot vectors, without any gap at the boundary by inserting control

CONTACT Varun Asthana varunasthana92@gmail.com

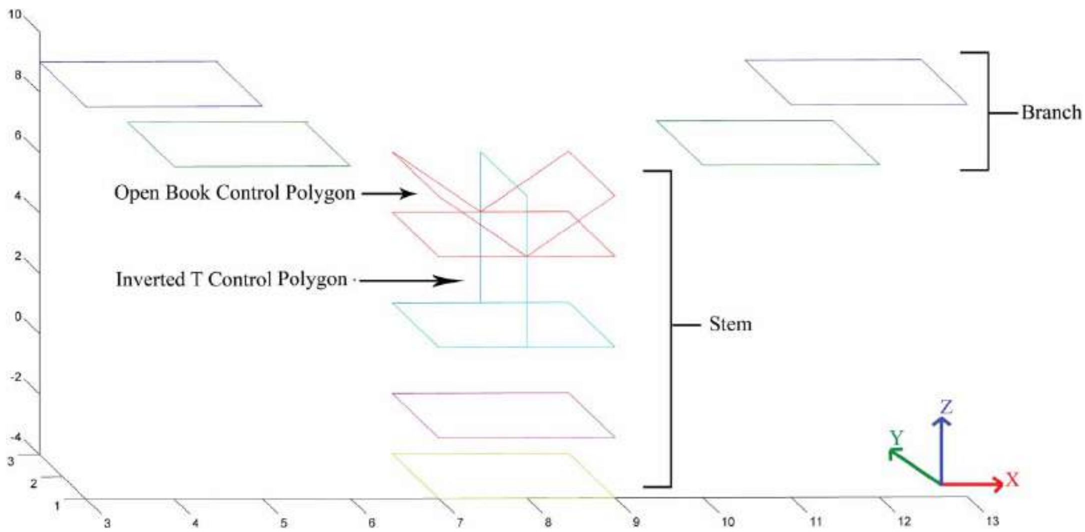

Figure 1. General Layout of Control Polyhedron.

points to form T-junctions. As we have not used surface patches in this work, hence joining of B-spline surfaces is not required (but methodology of T-splines may be used to produce similar results using surface patches with G1 continuity).

1.2. Overview of the paper

The work presented here is related to surface generation of bifurcating duct by the use of a disjoint open B-spline surface from the available data sets. The data sets are presumed to be a set of coordinates which acts as control points for the generated B-spline surface, along with a set of coordinates which gives the information about control points where a disjoint in the surface is required.

The control point layout, forming control polyhedron, used for surface generation is shown in Fig. 1. The geometrical arrangement of control points in parallel horizontal planes along the z axis, forming square shapes (either one or in a pair of two) is referred to as a control polygon for that plane. Two special shaped control polygons are also used- one shaped like an open book structure and one shaped like an inverted T structure, as shown in Fig. 1. The horizontal levels having two square shaped polygons form the part of the surface termed as branches, while the horizontal levels with single square polygon form the part of the surface termed as stem. Open book shaped control polygon along with inverted T shaped control polygon marks the junction of two branches as well as the top of the stem.

In the subsequent sections and sub-sections, methodology is explained in details with conceptual figures.

2. Methodology

In this work it is assumed that the control polyhedron is already available. In real world problems, where the surface has to be interpolated on real data set, the control polyhedron set can be formed by using methods described in [9].

The equation to generate B-Spline curve from the given set of control points is-

$$C(u) = \sum_{i=0}^{n} N_{i,k}(u) P_i \tag{2.1}$$

where,

$C(u)$ = Point on B-spline curve
$n+1$ = No. of control points in u direction
P_i = i^{th} control point
$N_{i,k}(u)$ = Basis function in u direction

$$N_{i,k}(u) = (u - u_i)\frac{N_{i,k-1}(u)}{u_{i+k-1} - u_i} + (u_{i+k} - u)\frac{N_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}} \tag{2.2}$$

Where,

$$N_{i,1} = \begin{cases} 1, & u_i \le u \le u_{i+1} \\ 0, & otherwise \end{cases}$$

And u_i are called the parametric knots or knot values. These values form a sequence of non-decreasing integers called knot vectors. For an open B-spine curve, u_i are given by:

$$u_j = \begin{cases} 0, & j < k \\ j - k + 1, & k \le j \le n \\ n - k + 2, & j > n \end{cases} \tag{2.3}$$

where,

$0 \le j \le n + k$
k = Order of the curve
k-1 = Degree of the curve

Open B-spline surface is generated by

$$S(u, v) = \sum_{i=0}^{n} N_{i,k}(u) \sum_{j=0}^{m} N_{j,l}(v) P_{ij} \tag{2.4}$$

where,

S(u,v) = Point on B-spline surface
m+1 = No. of control points in v direction
$N_{j,l}(v)$ = Basis function in v direction

and other variables hold the same meaning as before. P_{ij} simply contains the control points in defined order.

2.1. Disjoint B-spline curve

The surface generated in this work by the use of a single B-spline surface equation is made to disjoin both in u and v directions. Disjoint in B-spline can be obtained by knot value repetition. Here, disjoint curve is being discussed, the approach of it is then applied to the B-spline surface both in u and v directions as per the requirements so as to have branched surface.

For a curve of degree 3 and number of control points = 12,

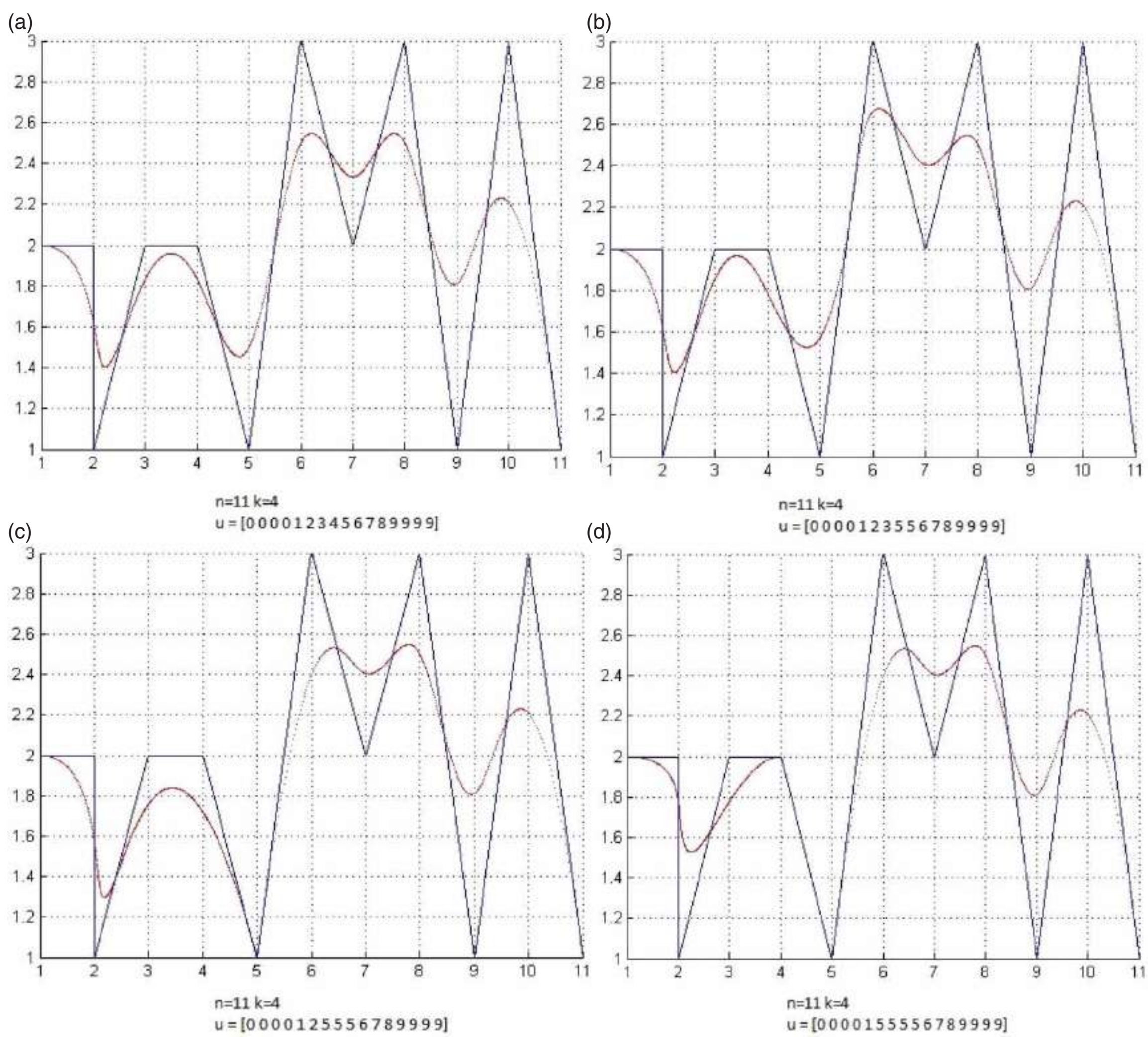

Figure 2. B-Spline Curve with: (a) Standard knot values, (b) Multiplicity 2 of knot value 5, (c) Multiplicity 3 of knot value 5, and (d) Multiplicity 4 of knot value 5.

n = 11
k = 4
u = [0 0 0 0 1 2 3 4 5 6 7 8 9 9 9 9] using Eqn. (2.3)

Of the four figures shown above, Fig. 2(a), Fig. 2(c), and Fig. 2(d) are important for us. It can be seen in Fig. 2(c) that when a knot value is repeated for "k-1" times, i.e. 3 times in this example, the original single curve of Fig. 2(a) ceases at an intermediate control point and a new curve originates from the same control point, with the two curves having C0 continuity. If the knot value is repeated for "k" times, i.e. 4 times in this example as in Fig. 2(d), the original single curve divides into two curves by skipping an intermediate control point. In this manner, we can make two or more disjoint curves without having C0 continuity from single B-spline equation. One of the noteworthy outcome of this disjoining of the curve is that in Fig. 2(c) and Fig. 2(d) the second curves are same but the first curves of both the figures are not same. Also for the very first disjoint in the curve at n^{th} control point, we need to have at least "k-1" control points before it, as the first k knot values of the knot vector have to be zero.

2.2. Algorithm for disjoint B-spline surface generation

The parameters *u* and *v* used, define the direction of the surface in horizontal plane and vertical plane respectively. The technique described in the sub-section [2.1] to generate disjoint curve is applied to generate disjoint surface. The information about the control points at which the surface will be disjoined both in *u* and *v* directions is made available as input data in matrices Mu and Mv respectively. Along with knot value repetition, control point repetition is also done at the point of disjoint. Requirement for control point repetition is explained in sub-section [3.4].

Input-

- Order of surface in *u* direction as 'k'; and in *v* direction as 'l'.
- x, y and z coordinates of control points as Px, Py and Pz respectively.
- Disjoint curve information along *u* and *v* directions in Mu and Mv respectively.

Algorithm-

- Introducing control points multiplicity in Px, Py and Pz using Mu and Mv.
- Calculating knot vectors U and V in *u* and *v* directions respectively using Eqn. (2.3).
- Knot value repetition in U using Mu.
- Knot value repetition in V using Mv.
- Calculating basis functions $N_{i,k}(u)$ and $N_{j,l}(v)$.
- Calculating surface points using open B-spline surface equation.

In this work all experiments are done using k = 4 and l = 3 but the proposed model allows to have flexibility in the degree of surface both in *u* and *v* directions. The results for bi-cubic surface (i.e. k = 4, l = 4) and bi-quartic surface (i.e. k = 5, l = 5) are also shown in Fig. 10(a) and Fig. 10(b). Various approaches used to reach the final model to generate the bifurcating surface are described in the subsequent sections along with conceptual figures.

3. Discussion of experiments

The final surface generated in this work is the result of a series of experiments and improvements in each one of them. To understand the logic of the proposed method, sequential experiments are discussed.

3.1. Control polyhedron representation

Fig. 3 shows how the control points of various layers of control polygon (forming the control polyhedron) joins with the control points of adjacent layer of control polygon. Here control points are labeled as P(i,j) where "i" refers to the control polygon and "j" refers to the control point in the i^{th} polygon. A control point labeled more than once indicates that it has been traversed more than once. In the *u* direction, control points are traversed in the numerical sequence (incorporating disjoints at appropriate points) while in the *v* direction, control point labeled "1" in a control polygon will join with control point labeled "1" in all control polygons traversing from top to bottom. The direction of traversing as well as disjoints in the surface can also be understood by following the arrows shown in Fig. 3. Arrow head defines the direction of traversing and number of arrows define the number of times a particular section is being traversed in each control polygon.

3.2. First model

Total number of control points in each control polygon, while defining a B-spline surface, have to be same. In this model, the control polygons of the branch region have a total of 18 control points with 9 control points on each arm of the branch, used in a rectangular arrangement as shown in Fig. 3. Control polygons in the stem region are defined differently, as it needs to have same number of defining points i.e. 18 points. G1 continuity at the junction of the branch is theorized by utilizing the property of the open B-spline that it passes through the first and the last control point, and is tangential to the first and the last segment of the control polygon.

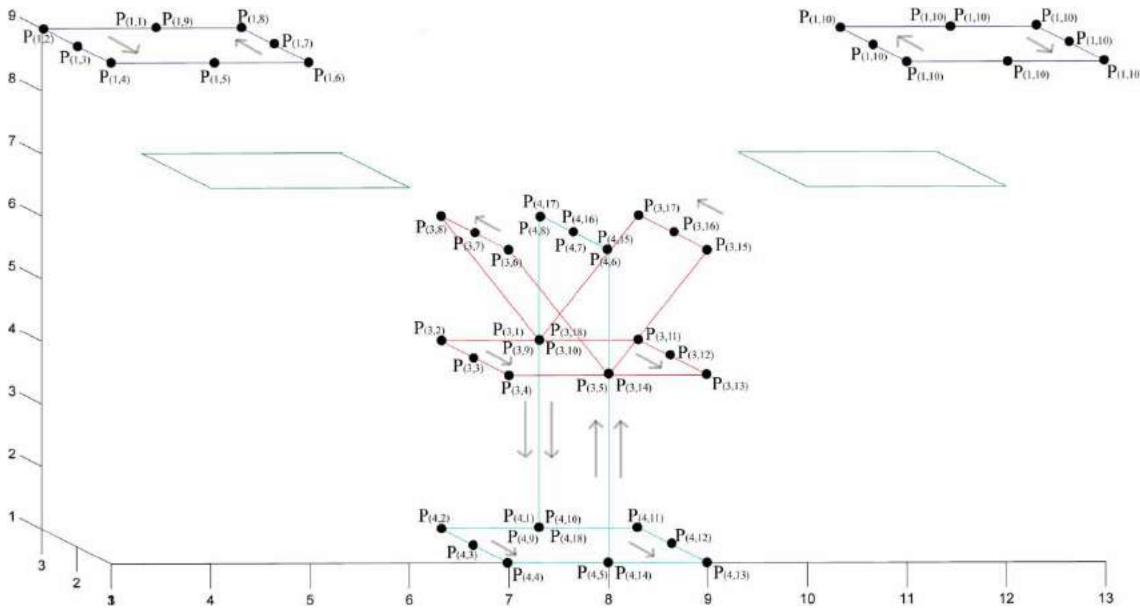

Figure 3. Shows the control polyhedron and the traversing order of control points for the First Model.

In the first control polygon of the stem (open book control polygon) control points labeled P(3,5), P(3,9), P(3,10) and P(3,14) marks the sudden change in the *u* direction of the B-spline while in the branch region a disjoint is required between control points P(1,9) and P(1,10). To have appropriate shape of the generated surface, B-splines in the *u* direction are terminated at control points P(i,5) and P(i,14) while a disjoint between control point P(i,9) and P(i,10) is introduced.

Learnings from the generated surface of first model-

- There are many open spaces. These spaces are present at the places where the spline (either in *u* direction or *v* direction) reaches the terminating control point/polygon; and at the point of disjoint in the spline. This can be seen in Fig. 4(b) where stem fails to reach the lower most layer of the last control polygon.
- If the above problem was not in existence, even then the two arms of the branch are not overlapping at the junction, as visible in Fig. 4(a) and Fig. 4(b).
- The disjoint curve became predominant in the entire stem region, hence a disjoint of spline along the *v* direction is required.
- A partition line is present in the surface where the contour suddenly changes its direction after terminating at the control points P(i,5) and P(i,14). This part of the surface will only have C0 continuity in *u* direction if the sets of control points {P(i,4), P(i,5) & P(i,6)} and {P(i,13), P(i,14) & P(i,15)} are not individually aligned in a straight line.

3.3. Second model

In this model we have addressed the issue of dominance of the disjoint in the entire stem region of the generated surface of the first model. Hence, few new layers in the control polyhedron are added in the stem region, below the inverted T control polygon. Fig. 5(a) shows the new control polyhedron. All the new layers of control polygon added have similar geometrical arrangement with those of a single arm of the branch (i.e. square shape) but have different scheme of control point traversing since the number of control points are twice in the stem compared to number of control points in a single arm of the branch. The arrangement of control points for all other control polygons is same as that of the First Model.

In Fig. 5(a) the control polygon of stem in inverted T shape i.e. P(4,j) is overlapped by a control polygon P(5,j), though it's not visible in the figure, which is identical to control polygon 6, P(6,j). The inverted T polygon is used to disjoint the spline in *v* direction and begin the second part of the spline from the overlapped control polygon P(5,j). The first four control polygons of stem region (seen from top to bottom i.e. control polygons P(3,j), P(4,j), P(5,j) and P(6,j)) should be aligned in a straight line or projected straight line (in similar way as shown in Fig. 14(b) and Fig. 14(c)) to have G1 continuity in *v* direction. But with a disjoint introduced in *v* direction, open space appears at the disjoint (same as in First Model).

As shown in Fig. 6, clockwise and counter-clockwise traversing of the control polygon in the stem region (as shown in Fig. 5(a)) did not produce the identically

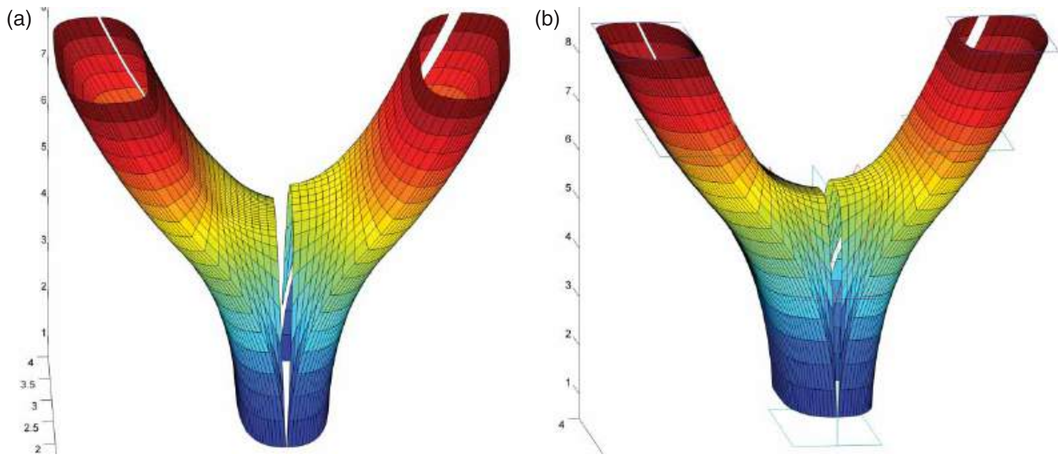

Figure 4. Interpolated surface of First Model: (a) Without control polyhedron, and (b) With control polyhedron.

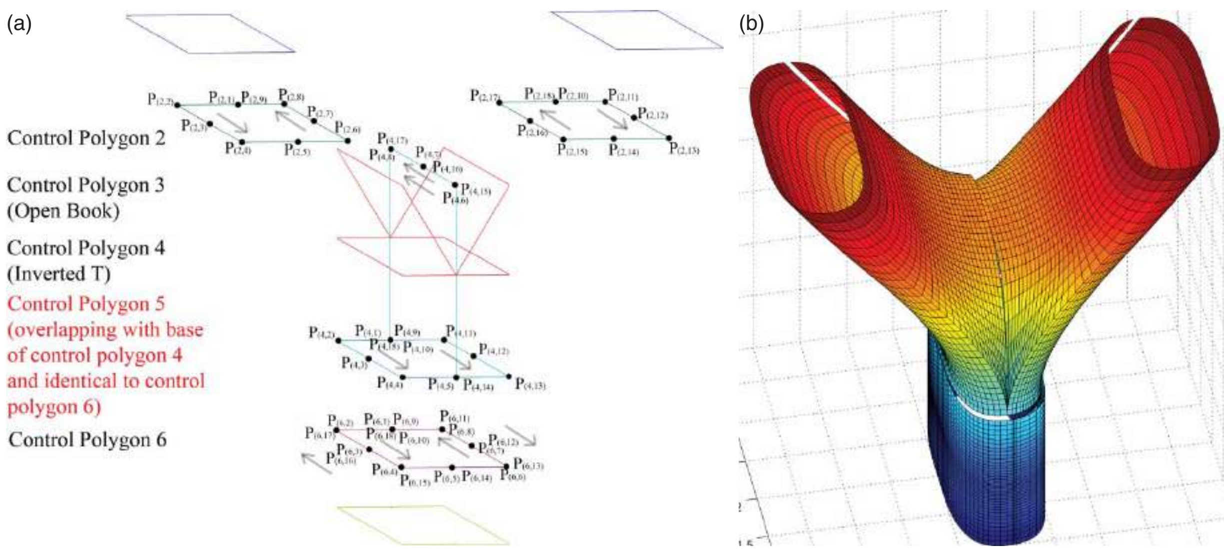

Figure 5. Second Model and interpolated surface results: (a) Shows arrangement of control polyhedron, and (b) Shows the interpolated surface.

overlapping contours. Hence a new arrangement of control points is required which could maintain the same directional traversing for segments of surface which are traversed twice so as to have overlapping contours.

3.4. Final model

In this model, three problems of previous models have been addressed. Firstly, to attain same directional traversing for segments of surface which are traversed twice so as to have overlapping contours. Secondly, to close the open spaces where the surface terminates and/or disjoins both in u and v directions. Thirdly, to attain G1 continuity at the junction of the branch by exact overlapping.

To maintain the same directional traversing of the repeated segments of control polyhedron in stem region, a new order of traversing of the control points in the control polyhedron is used. The new arrangement is shown in Fig. 7(a). The generated surface contours in u direction may be interpreted as 4 contours- contour 1 from control point P(i,1) to P(i,5); contour 2 from control point P(i,6) to P(i,10) by incorporating a disjoint at control point P(i,5); contour 3 from control point P(i,11) to P(i,15) by incorporating a disjoint at control point P(i,10); and contour 4 from control point P(i,16) to P(i,20) by incorporating a disjoint at control point P(i,15).

The reason for open spaces in the surface at various disjoints (Fig. 5(b)) i.e. control point P(i,9) in u direction and control polygon P(4,j) in v direction, and at boundary conditions i.e. control point P(i,18) in u direction and control polygon P(8,j) in v direction, is by virtue of step wise increment of knot vector U in the code used to generate the surface. For example, when u is changed

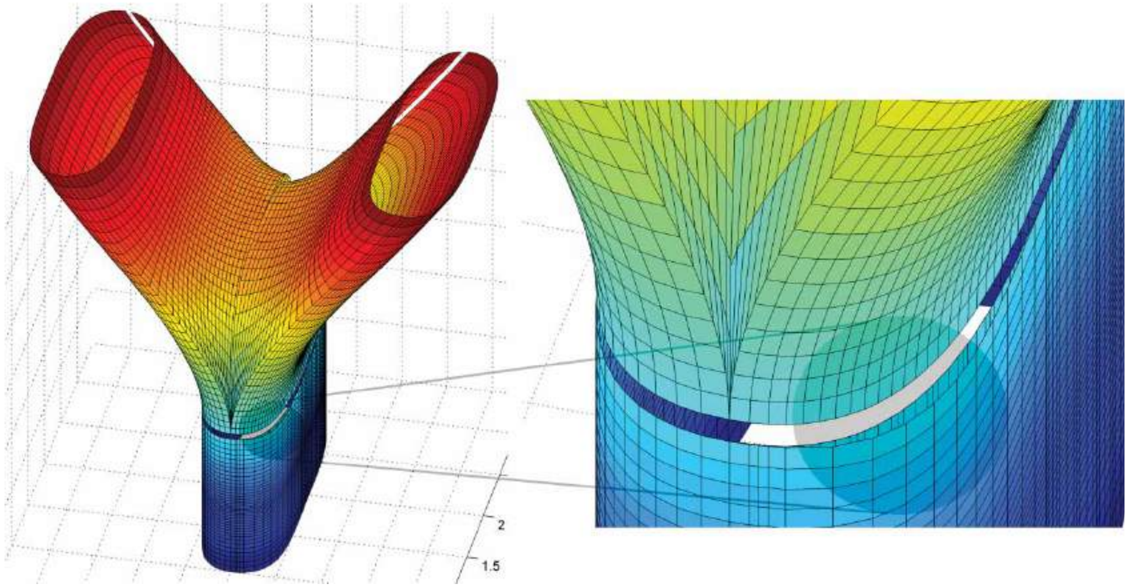

Figure 6. Shows the non-overlapping contours with clockwise and anti-clockwise traversing of same control polygon.

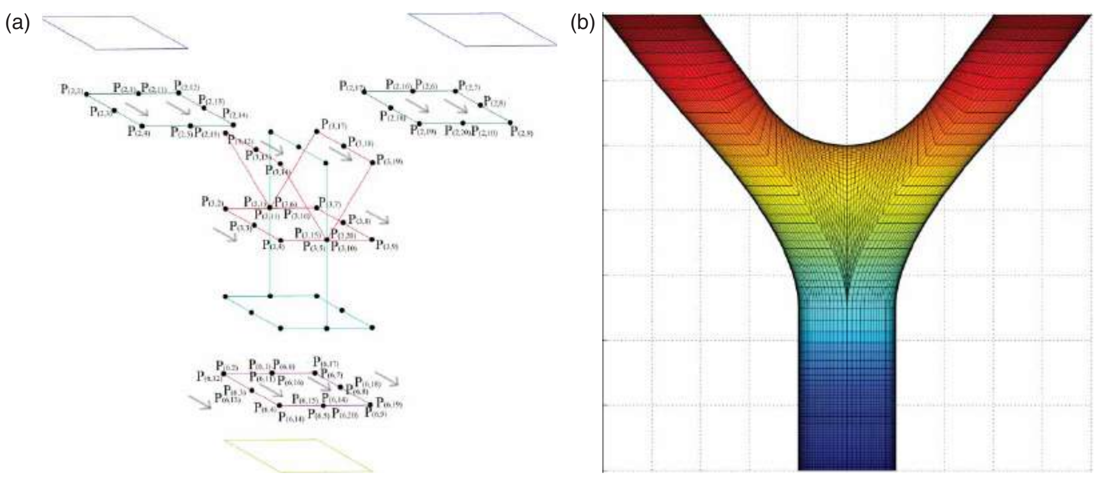

Figure 7. Final Model and interpolated surface results: (a) Shows arrangement of control polyhedron, and (b) Shows the interpolated surface.

from 1 to 2 in 10 steps then the open space will be more than when u is incremented in 20 steps. Thus even if u is incremented in a very large number of steps then also some open space will still be present, which may not be visible to naked eyes. In order to close the surface, contours are first terminated at the control points in u direction and at the control polygons in v directions where a disjoint is required. After terminating the spline at the control points/polygons the algorithm for disjoining the surface is applied. Also the contours are terminated at the boundary condition, both in u and v directions.

In Second Model (Fig. 5), the inner side of the left arm was constructed by traversing the control points P(i,5) to P(i,9) and the inner side of right arm was constructed by traversing the control points P(i,14) to P(i,18). The difference in shapes of inner left and inner right arm of the branch at junction, though they have geometrically identical control polygons, is because of the fact that the surface contour in u direction is disjoined at the control point P(i,9). As shown in Fig. 2(c) and Fig. 2(d), after having a disjoint in the spline, the first part of the curve changes its shape while the second part remains same. Thus the shape of inner side of the left arm, which acts as the first part of the surface contour disjoined at control point P(i,9), is changed. In order to have same shape for the inner side of left and right arm at the junction, both of these should either behave as the 1st part of spline after disjoint or both of these should behave as the 2nd part of spline after disjoint. This has been achieved by the above mentioned new order of traversing of the control points (Fig. 7(a)).

To incorporate the termination of spline at the boundary conditions as well as at the locations where a disjoint is required, multiplicity of control points/polygons and knot values is used. A contour can be terminated at a control point/polygon by knot value repetition for "k-1" times, and a disjoint in the spline can be achieved by knot value repetition for "k" times (as explained in subsection [2.1]). At the control points/polygons of disjoint, since the splines have to be terminated first before they are made to disjoin, control points'/polygons' multiplicity is used so as to eliminate the ambiguity of knot values responsible for terminating the spline with the knot values responsible for the disjoint. For example, let k = 4 and n = 12 the knot vector will be-

k = 4
n = 12 (P1, P2, P3, P4, P5, P6, P7, P8, P9 . . . . . .)
U = [0 0 0 0 1 2 3 4 5 6 7 8 9 10 10 10 10] using Eqn. (2.3)

To terminate and restart the spline at control point 5 (P5), the knot vector will become-

U = [0 0 0 0 1 4 4 4 5 6 7 8 9 10 10 10 10]

by repeating the knot value "4" for "k-1" times i.e. 3 times in this example. Instead if we want to have a disjoint in the spline at control point 5 (P5) and again begin the spline from control point 6 (P6), then the knot vector will become-

U = [0 0 0 0 1 5 5 5 5 6 7 8 9 10 10 10 10]

i.e. by repeating the knot value "5" for "k" times (4 times in this example).

Now if we want to first terminate the spline at control point 5 and also disjoin it between control points 5 and 6, it will lead to an ambiguous situation for deciding the knot vector.

Thus the multiplicity of control point 5 is increased to "k" times. Now,

k = 4
n = 15 (P1, P2, P3, P4, P5, P5, P5, P5, P6, P7 . . . . . .)
U = [0 0 0 0 1 2 3 4 5 6 7 8 9 10 11 12 13 13 13 13] using Eqn. (2.3)
(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10 . . . . . .) = (P1, P2, P3, P4, P5, P5, P5, P5, P6, P7 . . . . . .)

Now to terminate the spline at P5 as well as to have a disjoint between P5 and P6, we will terminate the spline at X5 and disjoin the spline between X8 and X9. The knot vector will now be-

U = [0 0 0 0 1 4 4 4 8 8 8 8 9 10 11 12 13 13 13 13]

This resolved the ambiguity encountered before and it also does not affect the actual curve, as after termination, the new spline from X5 to X8 is just a point at P5, which is eventually the terminating point of the previous segment of the spline.

This technique is then implemented for surface generation both in u and v directions. The generated surfaces for two similar layouts of control polyhedron are shown in Fig. 7(b) and Fig. 8(b).

3.4.1. G1 continuity achieved

The above model gave us a successful result for generating a bifurcating surface using only a single B-spline surface equation. The next step is to incorporate G1 continuity at all the critical points/polygons of the surface and at the junction so as to have a better surface, not just a surface with C0 continuity as can be seen in Fig. 9(a) (last problem of the First Model). *These critical points/polygons are those control points/polygons of the control polyhedron where the spline is terminated; and started again*. Also, since in the above model we have terminated the spline before it is made to disjoin, hence the control points/polygons where a disjoint is present are also included in the critical points/polygons.

This G1 continuity of B-spline is achieved by aligning the control polygon segments on either side of a critical point in a straight line. This is done by adding one control point/polygon very close to the critical point/polygon on either side of it, with all three points/polygons aligned in a straight line. This resulted in a smoother G1 continuous surface, as shown in Fig. 9(b).

4. Advantages of final model

The model presented above offers a wide range of advantages in the designing or reconstruction of surface in terms of shape and also in terms of parameters used. All the experiments conducted above used the degree of 3 in u direction and degree of 2 in v direction (k = 4, l = 3); but it is not necessary. The model can handle any degree both in u and v directions. The results for bi-cubic (k = 4, l = 4) and bi-quartic (k = 5, l = 5) surfaces with similar control polyhedron arrangements are shown in Fig. 10(a) and Fig. 10(b) respectively.

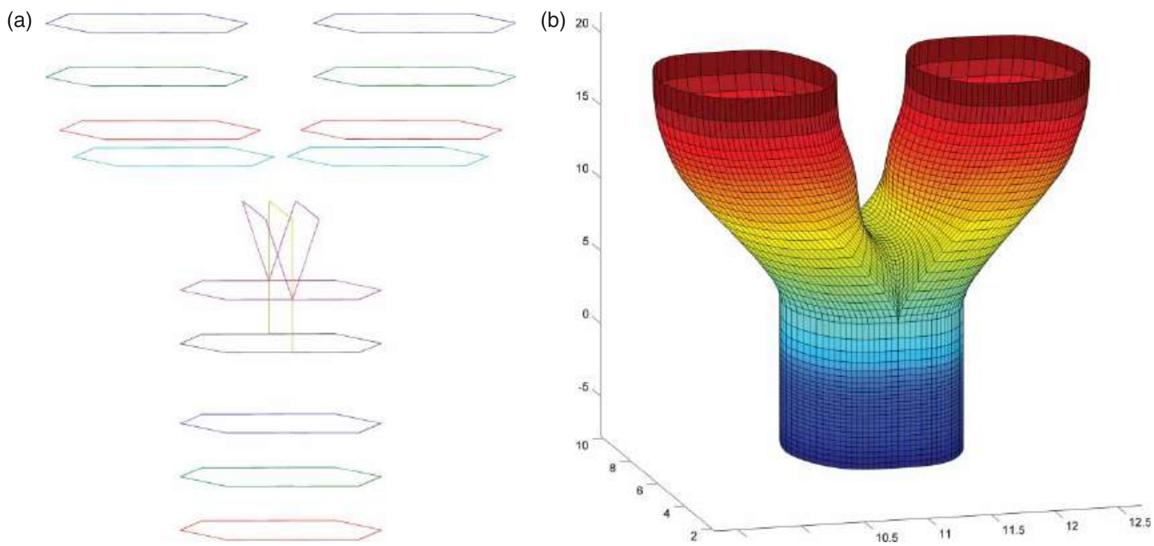

Figure 8. Final Model and interpolated surface results: (a) Shows another arrangement of control polyhedron, and (b) Shows the interpolated surface.

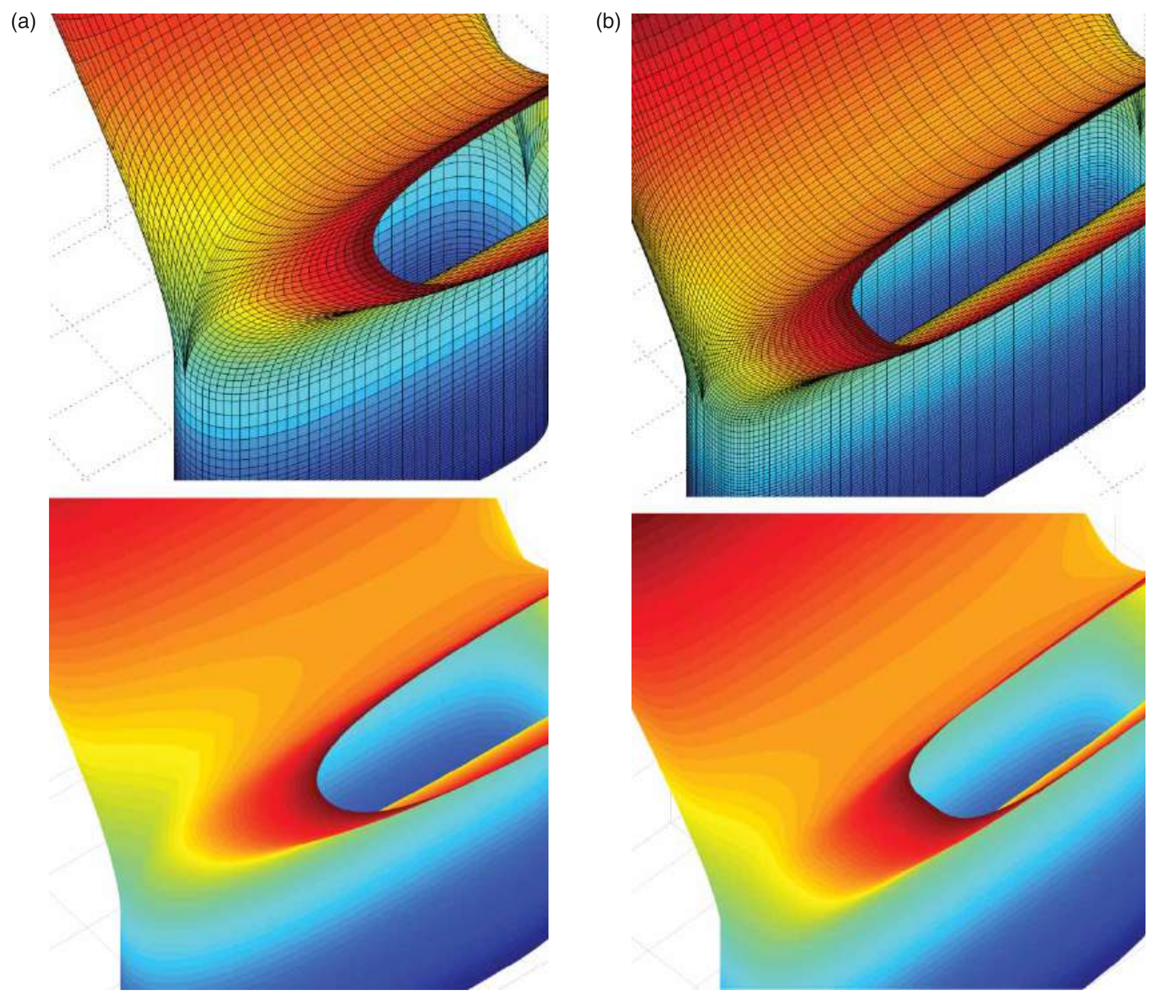

Figure 9. Shows the junction of two arms of a branch with: (a) G1 continuity only at upper part, and (b) G1 continuity at the upper part and sides.

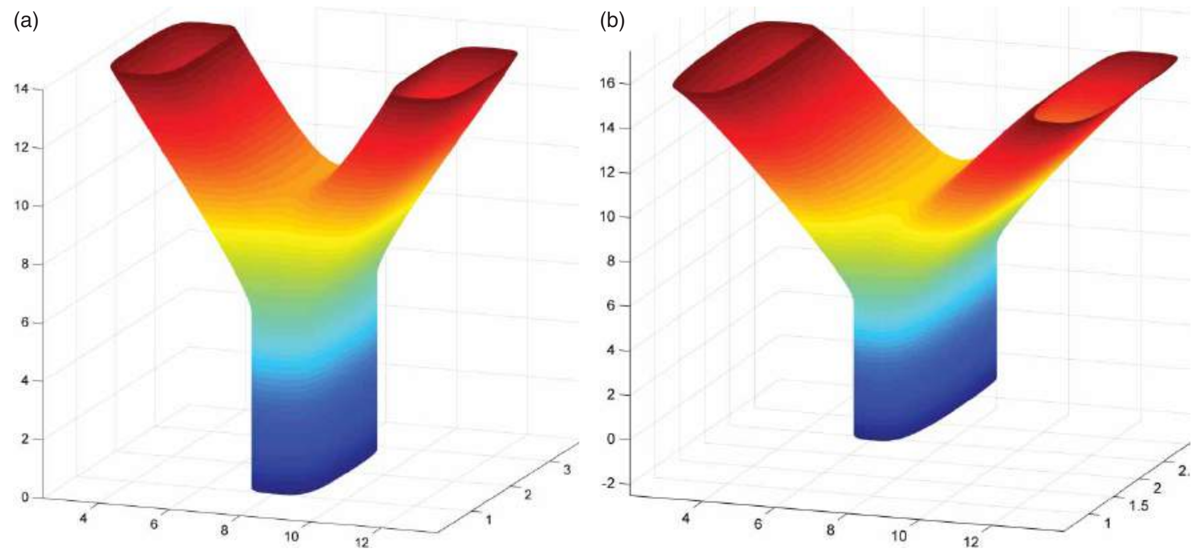

Figure 10. Shows the interpolated surface with G1 continuity: (a) Bi-Cubic Surface (order 4 by 4), and (b) Bi-Quartic Surface (order 5 by 5).

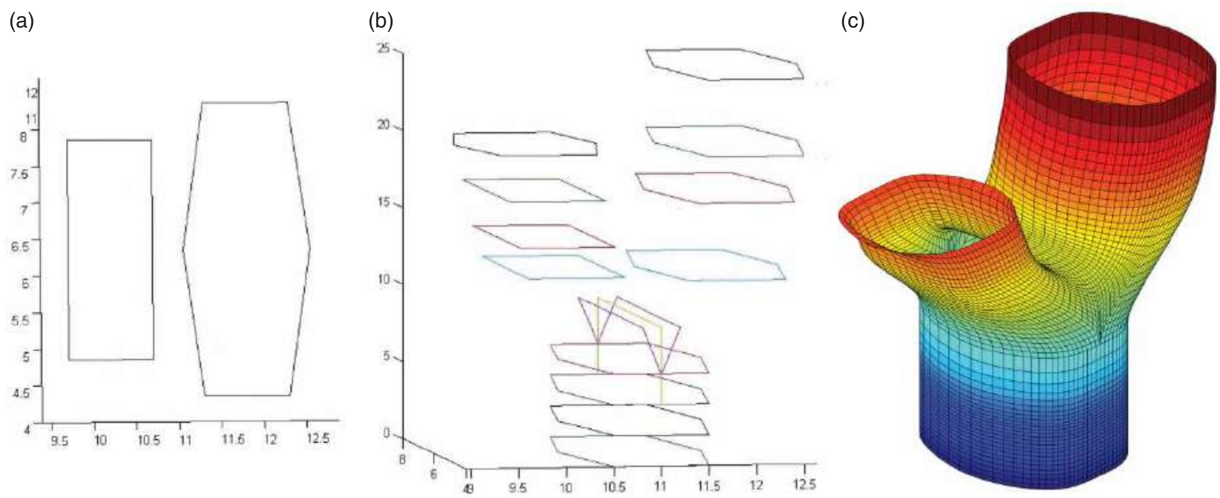

Figure 11. Asymmetric Branching: (a) Top view, (b) Control polyhedron, and (c) Generated surface.

All the above illustrated images of the generated surface were taken to be regular and symmetric for ease of understanding. With this model we can have asymmetric branches as well as irregular non-uniform cross-sectional shapes along the length as shown in Fig. 11 and Fig. 12 respectively. We have assumed that the control points data set is available in horizontal slices, but this is not mandatory. If the control points provided are arranged in non-parallel slices but conforming to a valid shape, a surface can still be generated incorporating twist and turns along the length (as shown in Fig. 12). Thus the proposed model provides a way to construct complex shapes which can handle a gamut of patterns of input data, still conforming to the continuity requirements.

If this algorithm is used in iteration, we can generate G1 continuous sub branches, as shown in Fig. 13, by aligning the last two control polygons of a stem (of the sub-branch, marked red and yellow in Fig. 14(a)) and the first two control polygons of the branch (of parent surface, marked yellow and green in Fig. 14(a)) in a straight line, as shown in Fig. 14(b), or in a projected straight line, as shown in 14(c)). Last polygon of stem (of sub-branch) and first polygon of branch (of parent surface) have to be geometrically identical and overlapping (marked yellow in Fig. 14). This methodology of using the final model in iteration may be useful for the 3-D construction of bronchial tree structures, vascular systems, etc.

In [1], authors developed multiple bifurcations by using single equation without the iteration of algorithm for single bifurcation, but they had to redefine the control polyhedron (with each sub-division number of control points in each control polygon of surface increased).

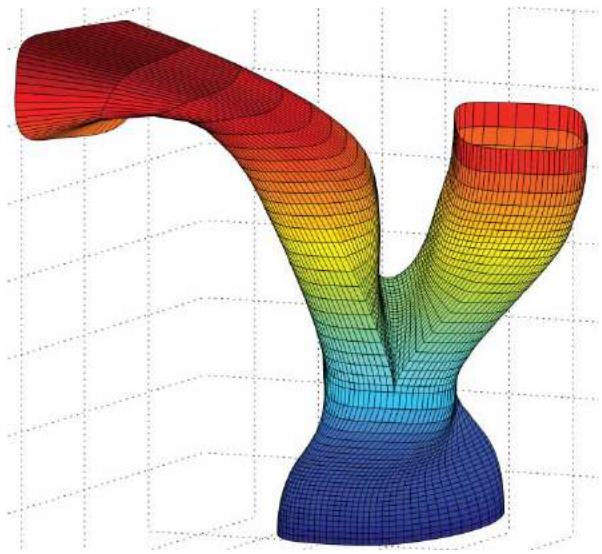

Figure 12. Shows irregular interpolated surface.

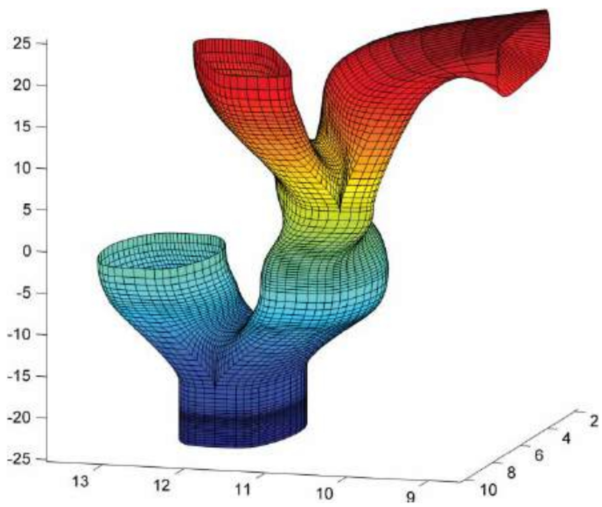

Figure 13. Shows multiple branched surface.

Thus pre-information about the number of sub-branches would therefore be required in their model so as to define the control polyhedron accordingly. If their model is used in iteration, certain modifications in defining the control polyhedron might be required to have atleast G1 continuity at the junction of sub-branches with respective parent surface, by virtue of difference in direction of traversing of control polygons in branch region and in stem region, which may lead to non-overlapping of contours, even if control polygons are geometrically identical.

With the iteration of algorithm used in the work presented here, number of control points in each control polygon remains constant i.e. same as used to define a single bifurcation. The general control polyhedron arrangement, as shown in Fig. 7(a), could be used for sub-branches as well, along with various other advantages offered in this model. Thus, no pre-information about the number of sub-branching is required.

5. Conclusion

The method discussed in this work provides an easy and effective way of generating dichotomous branched surface using only a single open B-spline surface equation. It allows the generated surface to have at least G1 continuity both parametrically and geometrically. This method differs from the conventional approach of stitching various surface patches together, which sometimes can be a complex step both computationally and logically for achieving continuity, to generate the final surface. Moreover, the flexibility to have irregular non-uniform cross sections in the surface along with asymmetric branches allows the generated surface to be more realistic. Thus it is a tool which can handle varying topological and geometrical

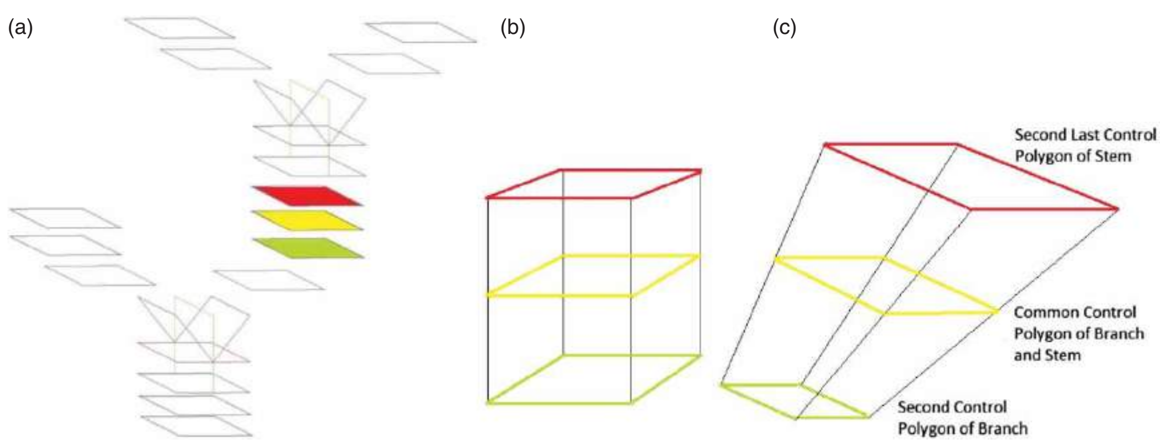

Figure 14. Shows: (a) General layout of control polyhedron for multiple-branching, (b) Shows arrangement of control polygons at the junction of parent surface and sub-branch in straight line, and (c) In projected straight line.

complexities while constructing the surface. Some applications of this can be reconstruction of human bronchial tree, engine's manifold design, and Y & T frames for automobiles.

6. Scope for future work

The work presented here gave a satisfactory result conforming to various topological and geometrical complexities. Final model is based on the assumption that control polyhedron is already available. We have to realize a method suitable for this model to construct the control polyhedron slices, automatically estimate the branching position and then insert the open book and the inverted T slices accordingly. With these automations clubbed with surface fitting, we may regenerate surfaces that are geometrically very close to the real world scanned objects.

ORCiD

Varun Asthana http://orcid.org/0000-0002-2012-5025

References

[1] Bhatt, A.; Goel. A.; Gupta, U.: Awasthi, S.: Reconstruction of Branched Surfaces: Experiments with Disjoint B-spline Surface, Computer-Aided Design and Applications, 12(1), 2015, 76–85. http://dx.doi.org/10.1080/16864360.2014.949577

[2] Capowski, J. J.: Computer-aided reconstruction of neuron trees from several serial sections, Computers and Biomedical Research, 10(6), 1977, 617–629. http://dx.doi.org/10.1016/0010-4809(77)90017-9

[3] Felkel, P.; Wegenkittl, R.; Buhler, K.: Surface models of tube trees, Computer Graphics International, 2004, 70–77, http://dx.doi.org/10.1109/CGI.2004.1309194

[4] Fetita, C. L.; Preteux, F. J.: Three-dimensional reconstruction of human bronchial tree in HRCT, Nonlinear Image Processing X, 3646, 1999, 281, http://dx.doi.org/10.1117/12.341094

[5] Gabrielides, N. C.; Ginnis, A. I.; Kaklis, P. D.; Karavelas, M. I.: G1-smooth branching surface construction from cross sections, Computer-Aided Design, 39(8), 2007, 639–651. http://dx.doi.org/10.1016/j.cad.2007.05.004

[6] Klein, R.; Schilling, A.; Strasser, W.: Reconstruction and simplification of surfaces from contours, Computer Graphics and Applications, 1999, 198–207, 325. http://dx.doi.org/10.1109/PCCGA.1999.803363

[7] Livny, Y.; Yan, F.; Olson, M.; Chen, B.; Zhang, H.; El-Sana, J.: Automatic reconstruction of tree skeletal structures from point clouds, ACM Transactions on Graphics (TOG), 29(6), 2010, http://dx.doi.org/10.1145/1882261.1866177

[8] Meyers, D.; Skinner, S.; Sloan, K.: Surfaces from contours, ACM Transactions on Graphics (TOG), 11(3), 1992, 228–258, http://dx.doi.org/10.1145/130881.131213

[9] Rogers, D. F.; Adams, J. A.: Mathematical Elements for Computer Graphics, Chapter-6, Tata McGraw-Hill, New Delhi, 2002, 445–458.

[10] Sederberg, T. W.; Zheng, J.; Bakenov, A.; Nasri, A.: T-splines and T-nurccs. ACM Transactions on Graphics (TOG), 22(3), 2003, 477–484, http://dx.doi.org/10.1145/882262.882295

[11] Spencer, R. M.; Schroeter, J. D.; Martonen, T. B.: Computer simulations of lung airway structures using data-driven surface modeling techniques, Computers in Biology and Medicine, 31(6), 2001, 499–511, http://dx.doi.org/10.1016/S0010-4825(01)00020-8

[12] Ye, X.; Cai, Y. Y.; Chui, C.; Anderson, J. H.: Constructive modeling of G1 bifurcation, Computer Aided Geometric Design, 19(7), 2002, 513–531. http://dx.doi.org/10.1016/S0167-8396(02)00131-0