



NEW NUMERICAL SOLUTION FOR TWO PARAMETRIC SURFACES INTERSECTION DRAGGING PROBLEM

RAMADHAN A. M. ALSAIDI*

*Department of Mathematics, College of Science and Arts in Gurayat, Jouf University, Gurayat 77454,
Saudi Arabia*

*Corresponding author: rsaidi@ju.edu.sa

ABSTRACT. The problem of intersecting two parametric surfaces has been one of the main technical challenges in computer-aided design, computer graphics, solid modeling, and geometrics. This paper aims at reducing and minimizing time and space required for the computations process of parametric surface intersection. To do this, a new numerically accelerating method based on continuation technique was utilized first by calculating a starting point, and second by tracing sequential points along the intersection curve following Broyden's method. Two factors have been identified as influential in controlling component jumping: initial points and step size. Test examples of intersecting two parametric surfaces demonstrated that this method was highly efficient with high-speed parametric solution. The intersection results are often given as curve's points.

1. INTRODUCTION

The intersection between two parametric surfaces has been an incessantly fascinating yet challenging topic in algebraic geometry because it has significant applications in computer graphics, modeling and computer-aided geometric design. The applications of this geometrical algebraic notion cover a wide-range of varying computerized processes including simulation of manufacturing processes, boundary evaluations, finite element of mesh generations, and contouring of scattered data. For its graphically-oriented signification, the topic

Received April 24th, 2021; accepted June 15th, 2021; published August 26th, 2021.

2010 *Mathematics Subject Classification.* 68U05.

Key words and phrases. Broyden's method; continuation method; curve tracing; initial point; parametric surface; surface intersection.

©2021 Authors retain the copyrights
of their papers, and all open access articles are distributed under the terms of the Creative Commons Attribution License.

has been of a great appeal to young and old researchers in computing arena. With the mounting interest in the field of computing designs, intersecting two surfaces in particular became the focus of several studies in the last decade of the millennium. A cataloguing survey of these studies may include, among many others, Chapter 12 of [1] Hoschek and Lasser which presented a comprehensive overview to the problem of intersecting free-form curves and surfaces; Patrikalakis's [2] review of the surface-surface intersection (SSI) methods which developed between (1988-1992); Farin's [3] SSI bibliography of 50 references, including techniques developed in the era of 1968–1990. Patrikalakis and Maekawa [4, 5] proposed a new results developed in the late 90s, containing an extensive body of SSI literature. On the other hand, special styles of surfaces such as sweep surfaces or torus were studied in [6]. A new algorithm for torus/torus intersection was introduced in [7]. Ku-JinKim [8] suggested a system to calculate all the circles in the intersection curve of two tori established on the geometric properties of the circles included in a torus. Recent studies include [9] in which an efficient and robust technique using a Bounding Volume Hierarchy (BVH) for computing the intersection curve of two freeform surfaces has been discussed. Numerous methods have been suggested to ray trace parametric surfaces, such as NURBS or Bézier surfaces. Nishita et al. [10] examined the clipping algorithm for computing the points at which a ray intersects a rational Bezier surface patch. Campagna et al. in [11] extended and optimized the Nishita et al. algorithm. To speed up Nishita's method, Wanget al. [12] combined Bezier-clipping ray coherency and Newton iterations. For computing the self-intersection and intersection curve(s) of two biquadratic Bézier surface patches, Chau, Stéphane, et al. [13] introduced three symbolic-numeric procedures. In [14], Sylvester matrix was used to construct a 9th degree polynomial coefficients of two cubic Bézier curves for getting the intersection points of the aforementioned curves through computing the real roots of this polynomial. Alsaidi, R. A. M [15] presented two methods for computing the intersection points of two parametric surfaces and one technique for computing the start point based on extended Newton method. These methods gave an acceptable accurate result for computing the intersection curve between two surfaces. Continuation technique among these methods uses extended Newton method frequently. However, Newton method has many disadvantages such as both the Jacobian matrix and its inverse actions should be calculated for each iteration which consumes quite time depending on the size of problem system. To overcome this problem, this work uses new accelerating methods known as Quasi-Newton methods or Broyden's method [16] instead of Newton method. The paper aims to solve the dragging problem of intersection parametric surfaces including biquadratic Bézier form surface intersection by optimizing Alsaidi's work [15] through minimizing the required memory and rendering the computations as simple as possible. In the next section, the problem will be stated. In section 3, Newton method and Broyden's method are recalled and compared. The developed technique is explained in section 4. Section 5 deals with the numerical problems so as to compare the develop method with Alsaidi's work [15]. Section 6 contains summary and the concluding remarks of this work.

2. PROBLEM STATEMENT

Let $S1(u, v)$ and $S2(s, t)$ be two parametric surfaces defined as:

$$S1 = \begin{bmatrix} X1(u, v) \\ Y1(u, v) \\ Z1(u, v) \end{bmatrix} \text{ and } S2 = \begin{bmatrix} X2(s, t) \\ Y2(s, t) \\ Z2(s, t) \end{bmatrix}$$

Then the problem of intersection between them involves the setting

$$S1(u, v) = S2(s, t) \Rightarrow S1(u, v) - S2(s, t) = 0$$

which can be interpreted as three nonlinear polynomial equations in four unknowns (u, v, s, t) as:

$$(2.1) \quad C(u, v, s, t) = \begin{bmatrix} X1(u, v) - X2(s, t) \\ Y1(u, v) - Y2(s, t) \\ Z1(u, v) - Z2(s, t) \end{bmatrix} = \begin{bmatrix} f_1(u, v, s, t) \\ f_2(u, v, s, t) \\ f_3(u, v, s, t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

This system can be solved by computing the set

$$P = \{(u, v, s, t) \in [0, 1]^4 \mid C(u, v, s, t) = S1(u, v) - S2(s, t) = 0\}$$

which is defined as a curve in four-dimension space $C(u, v, s, t) = (u(\alpha), v(\alpha), s(\alpha), t(\alpha))$ On the other hand, two biquadratic B'ezier surfaces $X(u, v)$ and $Y(s, t)$ are defined by

$$X(u, v) = \sum_{i=0}^2 \sum_{j=0}^2 P_{ij} B_i(u) B_j(v), \quad Y(s, t) = \sum_{i=0}^2 \sum_{j=0}^2 C_{ij} B_i(s) B_j(t),$$

where P_{ij} and C_{ij} are restricted to be in \mathbb{Q}^3 , and the quadratic Bernstein polynomials is defined as

$$B_i(u) = \binom{2}{i} u^i (1 - u)^{2-i}$$

Consequently, the intersection curve is computed by solving the following system $X(u, v) = Y(s, t)$ which is represented by three non-linear equations and four unknown variables.

Note that the intersection curve is as in $[0, 1]^4$.

However, there are many numerical techniques for solving these equations. This paper focuses on continuation method [15, 17, 18] which is efficient in both cases of parametric surface and biquadratic Bezier surface.

3. DEVELOPED METHOD

In this work, the computation system of the intersection points between two parametric surfaces consists of two different and successive phases: first, calculating the starting point, then tracing sequential points along the intersection curve [15, 19].

3.1. Computing a starting point. As mentioned above, the intersection problem can be viewed as solving the system of equation 2.1. There are many numerical methods to solve this problem such as Newton method which is used in many works such as Alsaidi's [15]. However, the Newton's method has the disadvantage that the Jacobian matrix and its inverse should be calculated at each iteration. This implies that the form of

the iterative procedure for Newton’s method takes longer time. This work, consequently, utilizes Broyden’s technique to avoid this problem. In this approach, an approximation matrix is employed instead of the Jacobian matrix. The approximation matrix can advantageously be changed at each iteration. In this way, the Broyden’s iterative expression is derived as:

$$x_{n+1} = x_n - A^{-1}C(x_n)$$

Here is A_n defined as:

$$A_n = A_{n-1} + \frac{y_n - A_{n-1}A_n s_n}{\|s_n\|_2^2} s_n^T$$

Where $y_n = C_n - C_{n-1}$ and $s_n = x_n - x_{n-1}$. For a more detailed treatment the reader is encouraged to consult [16]. For the intent of simplicity, the following algorithm summarizes the steps of the Broyden’s technique to compute the starting point:

Algorithm 1 (Broyden’s Method).

Input: The function matrix C of 2.1.

An initial point given $x_0 = (u_0, v_0, s_0, t_0)$.

A given tolerance ε .

Number of iterations M .

- (1) Evaluating $C(x_0)$.
- (2) Computing the Jacobian matrix

$$J(u, v, s, t) = \begin{bmatrix} \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial s} & \frac{\partial f_1}{\partial t} \\ \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial s} & \frac{\partial f_2}{\partial t} \\ \frac{\partial f_3}{\partial u} & \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial s} & \frac{\partial f_3}{\partial t} \end{bmatrix}$$

- (3) Since in this stage there is not enough information to calculate A_0 , set $L = 0_4$ and $A_a = J(x_0) \Rightarrow A_0^{-1} = I^{-1}(x_0)$, viz the Jacobian matrix J is computed at initial guess x_0 .
- (4) Extracting the new point by $x_1 = x_0 - A_0^{-1}C(x_0)$
- (5) Evaluating $C(x^1)$.
- (6) Calculating $y_1 = C(x_1) - C(x_0)$ and $s_1 = x_1 - x_0$.
- (7) Estimating $s_1^t A_0^{-1} y_1$.
- (8) Computing $A_1^{-1} = A_0^{-1} + \left(\frac{1}{s_1^t A_0^{-1} y_1}\right) [(s_1 - A_0^{-1} y_1) s_1^t A_0^{-1}]$
- (9) Extracting next point via $x_2 = x_1 - A_1^{-1}C(x_1), L = L + 1$
- (10) The previous process is repeated by updating initial guess ($x_0 = x_2$) in the next stage until the threshold is created (i.e. $C(x_2) - C(x_1) \leq \varepsilon$ or $L > M$). This indicates that the starting point (i.e. $x_1 = (u_1, v_1, s_1, t_1)$) is computed.

The sequential intersection points can be traced along the tangent direction by the following procedures:

- (i) Determination of the tangent unit vector to the path which is uniquely defined by $A \cdot \lambda = 0$ and represented by the solution by q .
- (ii) Selection the desired distance Δ and the tangent vector direction.
- (iii) Extraction of the initial guess via $x_0 = x_1 + (q \times \Delta)$, where Δ is the step size.
- (iv) To find next point, solve eq. 2.1 using Broyden's technique.
- (v) Evaluation of the relation error (E). If $\leq \varepsilon$, then the next point is computed and then go to Step III.
- (vi) Else, go to step (iv).

3.2. Type convergence of Broyden's Method. It should be noted that Broyden's technique converges superlinearly. This indicates that $\lim_{n \rightarrow \infty} \frac{\|x^{n+1} - p\|}{\|x^n - p\|} = 0$, where p is the solution to the system $C(x) = 0$, and x^n and x^{n+1} are successive approximations to p .

4. NUMERICAL IMPLEMENTATIONS

The develop system has been applied and its performance was measured by some representative numerical applications. All the computations were performed on a computer with 8 GB RAM and 2.50 GHz Intel Core i5 processor and the code was applied in MATLAB. The average of accuracies for every experiments was 100%.

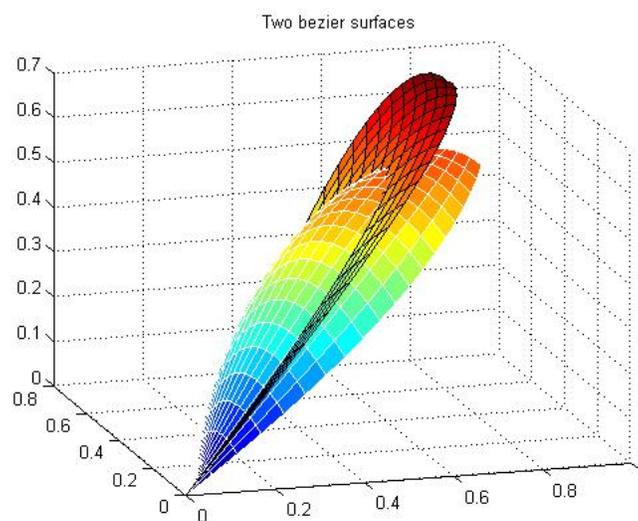


FIGURE 1. Two biquadratic Bezier surfaces B1 and B2

4.1. **First example:** Consider two biquadratic surfaces (B_1 and B_2) having the following control points:

$$B_1(u, v) = \begin{bmatrix} \left(\frac{1}{7}, 0, \frac{3}{5}\right) & \left(\frac{3}{5}, \frac{1}{5}, \frac{3}{4}\right) & \left(1, 0, \frac{7}{10}\right) \\ \left(\frac{3}{8}, \frac{4}{9}, \frac{2}{3}\right) & \left(\frac{2}{3}, \frac{3}{4}, \frac{1}{3}\right) & \left(\frac{6}{7}, \frac{3}{8}, \frac{5}{7}\right) \\ \left(\frac{1}{5}, \frac{6}{7}, \frac{4}{7}\right) & \left(\frac{3}{4}, \frac{7}{8}, \frac{3}{4}\right) & \left(\frac{7}{8}, \frac{7}{9}, \frac{5}{8}\right) \end{bmatrix}$$

$$B_2(s, t) = \begin{bmatrix} \left(\frac{2}{7}, \frac{1}{7}, \frac{2}{5}\right) & \left(\frac{3}{5}, \frac{1}{10}, \frac{2}{3}\right) & \left(1, 0, \frac{4}{5}\right) \\ \left(\frac{3}{8}, \frac{4}{9}, \frac{2}{3}\right) & \left(\frac{1}{3}, \frac{1}{2}, 1\right) & \left(\frac{5}{7}, \frac{3}{8}, \frac{2}{7}\right) \\ \left(\frac{1}{5}, \frac{6}{7}, \frac{3}{7}\right) & \left(\frac{3}{4}, \frac{7}{8}, \frac{5}{8}\right) & \left(\frac{7}{8}, \frac{4}{7}, \frac{1}{2}\right) \end{bmatrix}$$

The two Bezier's surfaces are displayed in figure 1.

It is worth pointing out here that the performance of the proposed system needs to select the appropriate initial guess which was chosen as $x_0 = (0.9994, 0.2501, 0.9994, 0.2502)$ and the step size was determined as 0.02. The resulting intersection curve between B1 and B2 as points is shown in figure 2.

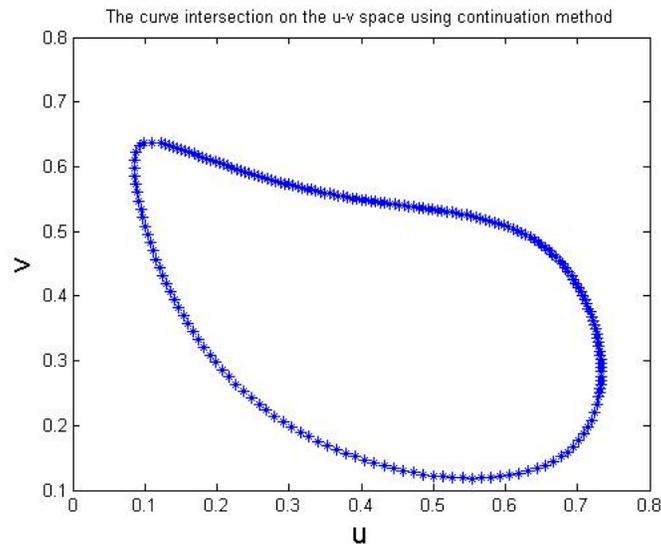


FIGURE 2. The intersection curve points between B1 and B2 on the u,v space

A Comparison between the time performance of proposed system and Alsaidi's work is done. It is provided for some points in Table 1.

K th Point	Develope's Point Coordinates	Alsaiddi's Point Coordinates	Number of iterations		Time performance in seconds	
			Developed	Alsaiddi	Developed	Alsaiddi
1 st	(0.7307, 0.2501, 0.7936, 0.3184)	(0.7274, 0.2327, 0.7849, 0.2990)	12	6	0.0479425	5.1307965
30 th	(0.5193, 0.5305, 0.6851, 0.6779)	(0.5367, 0.5272, 0.7021, 0.6699)	3	3	0.010677	0.1008304
61 ^{at}	(0.1332, 0.6325, 0.2210, 0.7251)	(0.1474, 0.6275, 0.2400, 0.7278)	3	3	0.013063	0.066004
100 ⁱⁿ	(0.2801, 0.2146, 0.2897, 0.2206)	(0.2643, 0.2274, 0.2758, 0.2336)	4	3	0.017218	0.071702
127 th	(0.6313, 0.1290, 0.6533, 0.1722)	(0.6160, 0.1248, 0.6355, 0.1653)	3	3	0.01235	0.062806
166 th	(0.5656, 0.5205, 0.7289, 0.6544)	(0.5791, 0.5164, 0.7409, 0.6460)	4	3	0.017066	0.067136
200 th	(0.1389, 0.6306, 0.2286, 0.7936)	(0.1493, 0.6268, 0.2425, 0.7281)	3	3	0.018479	0.078431

TABLE 1. A comparison between the time performance

Table 1 shows that the present system runs faster and achieves the best performance in the first example.

4.2. **Second example:** Given the two surfaces

$$S_1(u, v) = \begin{bmatrix} u \\ v \\ u^2 + v^2 \end{bmatrix} \text{ and } S_2(s, t) = \begin{bmatrix} s \\ t \\ \frac{45-s^2-t^2}{5} \end{bmatrix}.$$

Figure 3 shows the two surfaces.

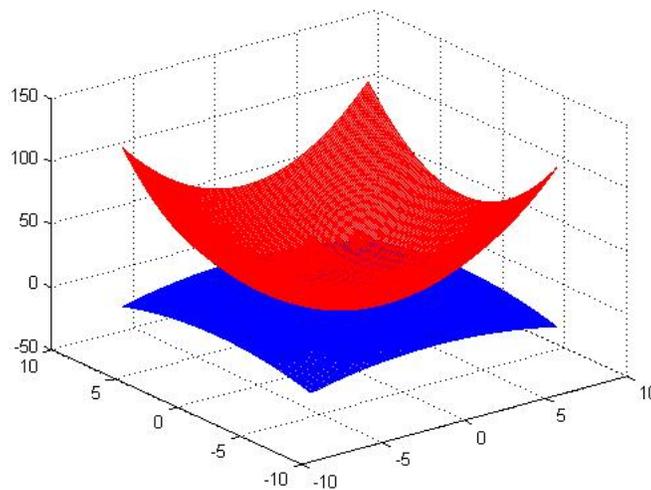


FIGURE 3. Two parametric surfaces S1(u,v) and S2(s,t)

Ideally specify an initial point $[u_0, v_0, s_0, u_0] = [1, 1, 1, 1]$. After the introducing system has been applied, the outcome intersection curve represented as points (i.e. 151 points) is illustrated in figure 4.

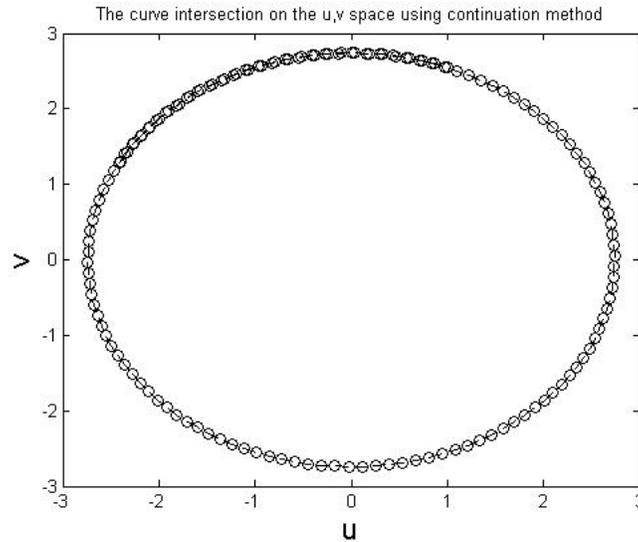


FIGURE 4. The intersection curve points between S1 and S2 on the u,v space

It is to be noted that the experiment was performed under the same environment which Alsaïdi [15] used. The comparison results indicate that the developed system consistently outperformed the Alsaïdi’s methods as shown in Table 2.

2* K th Point	Develope’s Point Coordinates	2* Alsaïdi’s Point Coordinates	Number of iterations		Time performance in seconds	
			Developed	Alsaïdi	Developed	Alsaïdi
1 st	(1, 2.5495, 1, 2.5495)	(1.936, 1.936, 1.936, 1.936)	9	6	0.0097	0.299215
25 th	(-2.092, 1.767, -2.092, 1.767)	(-1.198, 2.463, -1.198, 2.463)	4	3	0.003846	0.110219
71 st	(0.257, -2.726, 0.257, -2.726)	(-0.85, -2.603, -0.85, -2.603)	4	3	0.003947	0.126905
105 th	(2.633, 0.754, 2.633, 0.754)	(2.7148, -0.3605, 2.7148, -0.3605)	4	3	0.004042	0.118836
127 th	(0.433, 2.704, 0.433, 2.704)	(1.473, 2.31, 1.473, 2.31)	4	3	0.003772	0.129153
151 st	(-2.419, 1.284, -2.419, 1.284)	(-1.702, 2.146, -1.702, 2.146)	4	3	0.004643	0.132114

TABLE 2. Showing the results of the comparison between in example2.

Significantly observed, Alsaïdi’s work-run took an average of 0.121591624 Sec which is considerably more than the average of 0.004146306 sec of the proposed system. Hence, it can be said that the developed system offers the best quality–runtime ratio.

5. ANALYSIS

The execution of the proposed system is significantly affected by the following factors: initial guess and step lengths through tracing the control component jumping. The effect of these parameters is investigated in the previous example.

5.1. Effect of initial guess: It is well-known that the select of an initial guess is very efficient in the computation of any proposed method. However, there is no common solution that fits all concrete problems or cases. There are many works that set some criteria to determine the initial guess but they did not reach the efficiency in problem solving. Consequently, the best choice of the initial guess in this work was determined experimentally by taking into consideration that the initial guess is close to the solution (see table 3 .

Initial guess	Run-time(average)
[1, 1, 1, 1]	0.004146305960265
[0, 1, 1, 1]	0.003603891390728
[1, 0, 1, 1]	0.003603891390728
[1, 1, 0, 1]	0.004146305960265
[1, 1, 1, 0]	0.004146305960265
[1, 1, 0, 0]	0.003347392052980
[1, 0, 0, 0]	0.002750917880795
[0, 0, 0, 0]	convergence not achieved

TABLE 3. Influence of initial guess on Run-time and convergence

5.2. Effect of step size: The efficiency of the tracing in a proposed system is dependent on getting good step size to avert bisection. Nevertheless, too small step size increases the number of segments that are required for approximating the intersection curve within a given tolerance [20]. In this work, the effect was investigated through the performance of the develop system with different size of step lengths while another parameter was kept fixed. The result shows in table 4.

Size length	Run-time(average)
0.02	0.003288
0.2	0.004146306
0.6	0.003288048344371

TABLE 4. Effect of step size on Run-time and convergence

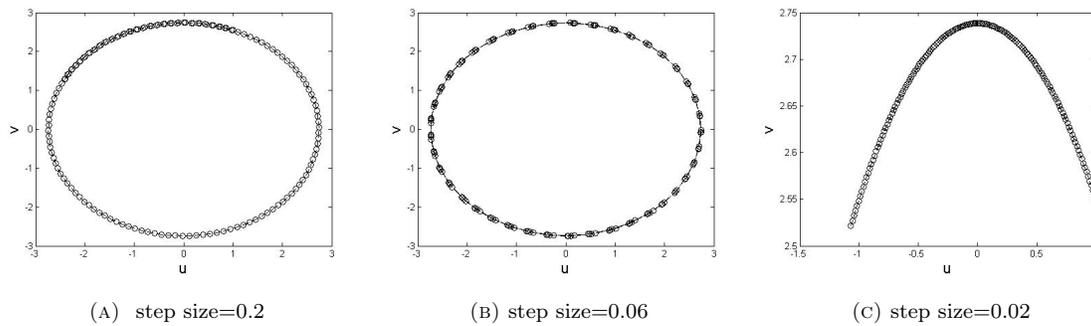


FIGURE 5. plots of effect of step size

6. CONCLUSION

This work introduces a develop continuation algorithm method based on Broyden's technique to compute points of curve yielded from an intersection between two parametric surfaces.

It has experimentally been found that the proposed algorithm can be applied to a variety of parametric surfaces efficiently. The comparisons between the developed technique and Alsaisi's work has been applied to measure out and validate the efficacy of the developed method. Moreover, the effect of factors as well as possible restrictions has been numerically examined.

The main contribution of this research work has been the reducing of the dragging intersection problem by speeding up the computation process. The developed system offers the best quality–runtime ratio and achieves the best performance in biquadratic Bezier surfaces.

It is recommended to extend the application of the current intersection technique on implicit surfaces and rational B-spline surfaces to more general and complex surfaces.

Acknowledgement: The author feels grateful to the reviewers and editors for their suggestions and constructively insightful comments that refine and improve the paper. This work has been supported by Jouf University, KSA.

Conflicts of Interest: The author(s) declare that there are no conflicts of interest regarding the publication of this paper.

REFERENCES

- [1] J. Hoschek, D. Lasser, *Fundamentals of computer aided geometric design*, AK Peters, Ltd., Wellesley, MA, 1993.
- [2] N. M. Patrikalakis, Surface-to-surface intersections, *IEEE Computer Graph. Appl.* 13 (1) (1993), 89–95.
- [3] R. E. Barnhill, *Geometry processing for design and manufacturing*, SIAM, Philadelphia, 1992.
- [4] N. M. Patrikalakis, T. Maekawa, *Shape interrogation for computer aided design and manufacturing*, Vol. 15, Springer, 2002.
- [5] N. M. Patrikalakis, T. Maekawa, Intersection problems, in: *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer, 2010, pp. 109–160.

-
- [6] J.-K. Seong, K.-J. Kim, M.-S. Kim, G. Elber, R. R. Martin, Intersecting a freeform surface with a general swept surface, *Computer-Aided Design*, 37 (5) (2005), 473–483.
- [7] X.-M. Liu, C.-Y. Liu, J.-H. Yong, J.-C. Paul, Torus/torus intersection, *Computer-Aided Design Appl.* 8 (3) (2011), 465–477.
- [8] K.-J. Kim, Circles in torus–torus intersections, *J. Comput. Appl. Math.* 236 (9) (2012), 2387–2397.
- [9] Y. Park, S.-H. Son, M.-S. Kim, G. Elber, Surface–surface-intersection computation using a bounding volume hierarchy with osculating toroidal patches in the leaf nodes, *Computer-Aided Design*, 127 (2020), 102866.
- [10] T. Nishita, T. W. Sederberg, M. Kakimoto, Ray tracing trimmed rational surface patches, in: *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, 1990, pp. 337–345.
- [11] S. Campagna, P. Slusallek, H. P. Seidel, Ray tracing of parametric surfaces: Bezier clipping, chebyshev boxing and bounding volume hierarchy. a critical comparison with new results, *Computer Graphics Group, University of Erlangen, Germany* (1997).
- [12] S.-W. Wang, Z.-C. Shih, R.-C. Chang, et al., An efficient and stable ray tracing algorithm for parametric surfaces, *J. Inform. Sci. Eng.* 18 (4) (2002), 541–561.
- [13] S. Chau, M. Oberneder, A. Galligo, B. Jüttler, Intersecting biquadratic bézier surface patches, in: *Geometric Modeling and Algebraic Geometry*, Springer, 2008, pp. 161–180.
- [14] B. Bily, The method of finding points of intersection of two cubic bezier curves using the sylvester matrix, *Silesian J. Pure Appl. Math.* 6 (2016), 155–176.
- [15] R. A. M. Alsaidi, A. Musleh, Two methods for surface/surface intersection problem comparative study, *Int. J. Computer Appl.* 92 (2014), 1-8.
- [16] R. Burden, J. Faires, A. Burden, Numerical solutions of nonlinear systems of equations, In: *Numerical analysis*. Cengage Learning, Boston, (1997) pp 544–576.
- [17] W. C. Rheinboldt, *Numerical analysis of parametrized nonlinear equations*, Wiley-Interscience, 1986.
- [18] H. B. Keller, *Lectures on Numerical Methods in Bifurcation Problems*, Springer Verlag, Heidelberg, Germany, 1987.
- [19] K. Abdel-Malek, H.-J. Yeh, Determining intersection curves between surfaces of two solids, *Computer-Aided Design* 28 (6-7) (1996), 539–549.
- [20] T. Dokken, *Aspects of intersection algorithms and approximation*, PhD Thesis, University of Oslo, (1997).