Computer-Aided Design and Applications

http://www.cadanda.com

An Algorithm for Adding Draft Angle to B-spline Surfaces

Kristiaan Schreve[1]

[1]Stellenbosch University, South Africa, kschreve@sun.ac.za

ABSTRACT

An algorithm for adding draft angles to B-spline surfaces is described. The algorithm modifies the control net of the surface in order to satisfy the draft angle conditions. The draft angle conditions are based on simple vector products performed on the quadrilaterals of the control net. The algorithm minimizes the total displacement of the control points in such a way that the draft angle conditions will be satisfied everywhere on the surface. The algorithm is demonstrated with some examples.

Keywords: B-spline surface, draft angle, least squares minimization
DOI: 10.3722/cadaps.2008.452-460

1. INTRODUCTION

Draft angles are very important in the mould and die industry. The draft angle allows the part to be extracted from the mould or die. The exact angle will depend on the material and manufacturing process. Draft angles can also vary for different parts of the object, e.g. internal surfaces may have a larger draft angle than external surfaces. Just like any other part parameter, the designer may wish to change the draft angle at any stage.

Most CAD packages have some tool to add draft angles. This works mostly for geometric primitives. Adding draft angle to complex free form surfaces are much more difficult and, to the knowledge of the author, there are no commercial CAD package that can add draft angle to general B-spline surfaces.

Unfortunately there are not many publications addressing this issue. Tokuyama and Bae [8] attempted using isocline curves and isocline surfaces. They first find an isocline curve on a free-form surface and then generate an isocline surface, which is essentially a ruled surface between the isocline surface and the intersection line in the draft direction on the parting surface. One of the major challenges of their method is finding the isocline curve.

Yan and Tan [9] present an algorithm for adding draft angles to constant radius blending surfaces, but not free-form surfaces in general. Their algorithm results in gaps between the constant radius blend surfaces. They solve this by adding blend surfaces to fill the gaps. Their work is specific to the problem often found in CAD modeling of adding draft angles to rolling ball fillet radii.

This author developed a method to modify a B-spline curve so that the curve will satisfy the draft angle condition [5],[6]. This was used successfully in a few situations to construct free-form surfaces from generator curves modified by this method. However, there are obvious limitations to working with curves only.

Other work on draft angles mostly focus on identifying undercuts on molded parts and finding the optimum parting configuration [10,2]. These studies do not attempt to change the surface so that draft conditions will be satisfied.

The approach in this paper differs from both [8] and [9] in that the B-spline surface is modified to satisfy the draft condition. No additional surfaces or curves are added.

The draft conditions are presented in the next section after which an algorithm is described that was used to add draft angle to the surfaces presented in the examples.

2. DRAFT ANGLE CONDITIONS

In another paper [7], sufficient conditions for draft angles on general B-splines were derived. Essentially, the conditions are derived from subdivision of the control net. A tangent plane to the surface can be found by sufficiently subdividing the control net. Since the subdivision is a linear interpolation process, it implies that if the control net satisfies the draft angle conditions everywhere, any subdivision will also and therefore a subdivision that gives the tangent plane will also satisfy the draft conditions. If the tangent plane satisfies the draft condition, then the surface also satisfies it. For a detailed proof, please see [7].

The conditions are presented in Eqn. (1) and (2). In these equations, $\mathbf{P}_{i,j}$ are the control points of the B-spline surface, $\mathbf{D}$ is the unit vector indicating the direction in which the draft conditions must be satisfied, hereafter called the *pull direction* and ϕ is the draft angle in radians. The $\otimes$ symbol indicates the vector product and $\cdot$ indicates the scalar product.

$$\frac{\left(\mathbf{P}_{i+1,j}-\mathbf{P}_{i,j}\right)\otimes\left(\mathbf{P}_{i,j+1}-\mathbf{P}_{i,j}\right)\cdot\mathbf{D}}{\left\|\left(\mathbf{P}_{i+1,j}-\mathbf{P}_{i,j}\right)\otimes\left(\mathbf{P}_{i,j+1}-\mathbf{P}_{i,j}\right)\right\|} > \cos\left(\frac{\pi}{2}-\phi\right) \tag{1}$$

$$\frac{\left(\mathbf{P}_{i+1,j}-\mathbf{P}_{i+1,j+1}\right)\otimes\left(\mathbf{P}_{i,j+1}-\mathbf{P}_{i+1,j+1}\right)\cdot\mathbf{D}}{\left\|\left(\mathbf{P}_{i+1,j}-\mathbf{P}_{i+1,j+1}\right)\otimes\left(\mathbf{P}_{i,j+1}-\mathbf{P}_{i+1,j+1}\right)\right\|} > \cos\left(\frac{\pi}{2}-\phi\right) \tag{2}$$

What these equations essentially mean, is that the unit vector formed by the vector product of the two edges meeting at $\mathbf{P}_{i,j}$ and its scalar product with the pull direction must be larger than the constant on the right hand side of the inequalities above. The same must be true for the control point on the opposite side of the quadrilateral belonging to $\mathbf{P}_{i,j}$, i.e. $\mathbf{P}_{i+1,j+1}$. This is illustrated in the following figure. It shows one quadrilateral of a control net of a B-spline surface in solid lines. The vectors are indicated with arrows. $\mathbf{D}$ is the pull direction. The dashed lines indicate the continuation of the control net.

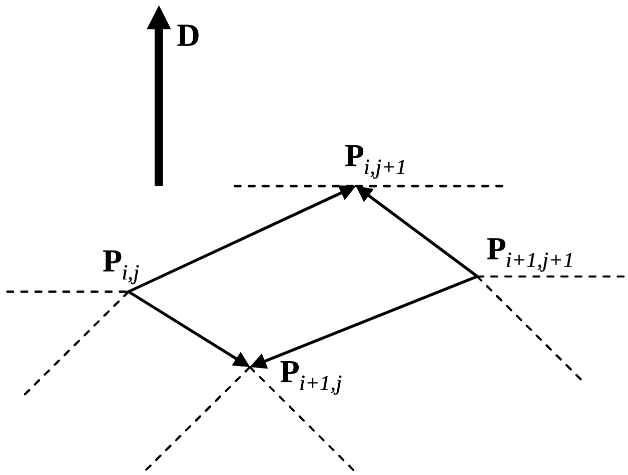

Fig. 1: Vectors used to check draft condition of a control net quadrilateral.

These inequalities give a quick way to evaluate whether a B-spline surface satisfies the draft conditions. Note that it was only proven that these are sufficient conditions. They hold for any B-spline surface and are independent of its degree. It is faster and easier to calculate than finding the derivatives of the B-spline surface. However, using them to adjust the control points of a surface that do not satisfy the conditions is not easy. Control points can be moved in any number of ways that still satisfy the conditions. In the next section, an algorithm will be described that uses these inequalities to adjust a B-spline surface so that it satisfies the draft conditions everywhere.

3. ALGORITHM

When changing a surface to satisfy the draft conditions, one would want the minimum distortion to the surface so that the original design will be intact. Therefore it was decided to minimize the total displacement of the control points of the surface. In essence, the algorithm identifies quadrilaterals in the control polygon that do not satisfy the draft conditions. It then finds a plane, from here on called the *draft plane*, on which the control points of that quadrilateral can be projected so that the draft conditions are satisfied. It is over conservative to project all four points on the same plane since there are two separate inequalities for the opposite corners of the quadrilateral. This, however, simplifies the algorithm significantly. The distance between the control points and the draft planes is minimized using a least squares approximation.

3.1 Least Squares Minimization

The distance from a point to a plane is given by Eqn. 3 where $\mathbf{a}_j$ is the normal vector of the j'th draft plane, $\mathbf{P}_i$ is a control point of the B-spline surface and $\mathbf{Q}_j$ is some point on the draft plane. Note that d_{ij} is a signed distance.

$$d_{ij} = \mathbf{a}_j \cdot \left(\mathbf{P}_i - \mathbf{Q}_j\right) \tag{3}$$

This defines a system of 4n linear equations, where n is the number of quadrilaterals that do not satisfy the draft conditions. It is 4n since there are four control points per quadrilateral. The draft plane for the j'th quadrilateral is therefore defined by the point on the plane, $\mathbf{Q}_j$, and its unit normal, $\mathbf{a}_j$. In this equation all the $\mathbf{a}_j$'s are predetermined (discussed in the next section), so the $\mathbf{Q}_j$ must be found.

This system of 4n linear equations can easily be rewritten in the following form (Eqn. 5), where $\mathbf{A}$ is a $4n\times n$ matrix which is a function of the components of $\mathbf{a}_j$ and $\mathbf{P}$ is a $4n\times 1$ column vector which is a function of the scalar product of $\mathbf{a}_j$ and $\mathbf{P}_i$. $\mathbf{Q}$ is the column vector that will contain the origins of the draft plane for each quadrilateral that needs adjustment and its scalar product with $\mathbf{a}_j$.

$$\mathbf{AQ}=\mathbf{P} \tag{4}$$

This is a standard set of linear equations. This system of equations can be solved in the least squares sense [1]. The least squares solution is found by solving the following system.

$$\mathbf{A}^{\mathrm{T}}\mathbf{AQ} = \mathbf{A}^{\mathrm{T}}\mathbf{P} \tag{5}$$

Since a plane is defined by its normal vector and *any* point on the plane, it is only necessary to fix one coordinate of that point. Therefore, Eqn. 5 can be simplified by setting the x and y coordinates to zero and solving only for the z coordinate. This is of course only possible when the normal vector is not perpendicular to the z-axis. If it is perpendicular, simply fix any two other coordinates of the point on the plane.

3.2 Finding the Normal Vector of Draft Planes

The unit normal vector of each draft plane, i.e. $\mathbf{a}_j$ in Eqn. 3, must be determined before Eqn. 4 can be set up. This is a local optimization step that only concerns the four control points associated with the quadrilateral that must be adjusted. In this case the objective function that must be minimized is the following equation, where $\mathbf{Q}$ is a point on the plane, $\mathbf{a}$ is the unit normal vector of the draft plane, $\mathbf{P}_i$ are the four control points associated with the quadrilateral and d is the distance from the control points to the draft plane.

$$d^2 = \sum_{i=1}^{4} \left\| \mathbf{a} \cdot \left(\mathbf{P}_i - \mathbf{Q}\right) \right\|^2 \tag{6}$$

This is illustrated in Fig. 2. The draft plane, that must be determined, is shown with the dashed lines. The original quadrilateral is shown with solid lines and its four control points $\mathbf{P}_i$. The pull direction is again shown as $\mathbf{D}$. $\mathbf{Q}$ is determined beforehand as

$$\mathbf{Q} = \frac{1}{4}\sum_{i=1}^{4} \mathbf{P}_i \tag{7}$$

The unit normal vector, $\mathbf{a}$, of the draft plane must now be determined by minimizing Eqn. 6. In this case the signed distance is not used when determining the unit normal vector because a unique solution is not possible; in fact a solution may be found that is completely counter intuitive to the problem at hand.

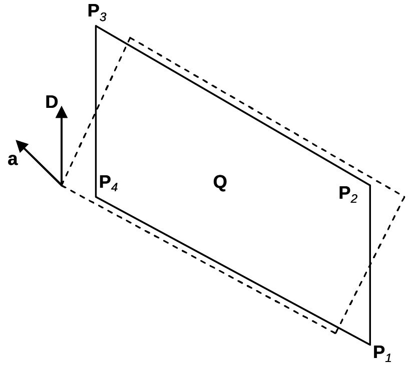

Fig. 2: Draft plane (dashed plane) fitting.

However, **a** must satisfy the following two constraints:

$$\mathbf{a}\cdot\mathbf{D}=\cos\left(\frac{\pi}{2}-\phi\right) \tag{8}$$

$$\|\mathbf{a}\|=1 \tag{9}$$

Using the above equations, **a** can be written in terms of one parameter (in Eqn. 6 all three components of **a** must be solved). This is illustrated in Fig. 3. In this figure $||\mathbf{a}||=1$ and $\gamma=\pi/2-\phi$. The pull direction is **D**, but now $||\mathbf{D}||=\cos(\gamma)$. It also follows that $||\mathbf{b}||=||\mathbf{c}||=||\mathbf{d}||=\sin(\gamma)$, therefore:

$$\mathbf{D}\cdot\mathbf{b}=0 \tag{10}$$

The vectors **b**, **c** and **d** can be determined in the following way. Vector **b** is found from Eqn. 10 and the fact that $||\mathbf{b}||=\sin(\gamma)$. One more equation is needed to fix **b**. Since the direction of **b** is not important, except that it is perpendicular to **D**, simply set the *x* components of **D** and **b** equal. This gives three equations to solve **b**. Now, **d** can be solved from the fact that **b** and **d** are chosen perpendicular to each other.

$$\mathbf{d}=\frac{\mathbf{D}\otimes\mathbf{b}}{\|\mathbf{D}\otimes\mathbf{b}\|}\sin(\theta) \tag{11}$$

Then, **c** can be solved.

$$\mathbf{c}=\mathbf{b}\cos(\theta)+\mathbf{d}\sin(\theta) \tag{12}$$

Lastly,

$$\mathbf{a}=\mathbf{D}+\mathbf{c} \tag{13}$$

Now, **a** is only a function of θ. So Eqn. 6 can now be minimized for θ. In this case, Brent's minimization method is used [3].

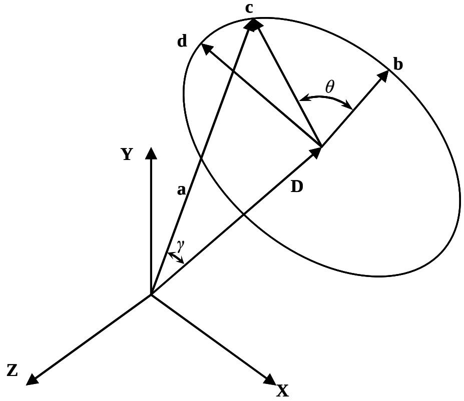

Fig. 3: Finding **a** in Terms of θ.

3.3 Reducing the Linear System

The system of equations (Eqn. 5) can be reduced further because quadrilaterals sharing either an edge or a control point must be fixed relative to each other. Say the j'th and k'th quadrilateral share control point $\mathbf{P}_i$. The normal vectors of each quadrilateral will be determined as described in the previous section. If $\mathbf{Q}_j$ is a point on the draft plane associated with the j'th quadrilateral then $\mathbf{Q}_k$, a point on the draft plane of the k'th quadrilateral, can be determined by virtue of the fact that the projection of $\mathbf{P}_i$ on the j'th and k'th draft planes must be the same point. The easiest way is to take $\mathbf{Q}_k$ to be the projection of $\mathbf{P}_i$ on the j'th draft plane.

The j'th draft plane is therefore defined as (where $\mathbf{T}$ is any point on the draft plane):

$$\mathbf{a}_j \cdot (\mathbf{Q}_j - \mathbf{T}) = 0 \tag{14}$$

The projection of $\mathbf{P}_i$ on this plane can be written as [4]:

$$\mathbf{P}_i' = \mathbf{P}_i - \mathbf{a}_j \left(\mathbf{a}_j \cdot (\mathbf{P}_i - \mathbf{Q}_j)\right) \tag{15}$$

Now set

$$\mathbf{Q}_k = \mathbf{P}_i' \tag{16}$$

Now $\mathbf{Q}_k$ is known in terms of $\mathbf{Q}_j$ and this reduces the dimensions of the linear system in Eqn. 5.

The reduction can significantly reduce the order of the linear system. Most surfaces in practical mechanical engineering examples will have one or two regions that need adjustment to satisfy the draft conditions. This means that there will typically be one or two sets of connected quadrilaterals that must be adjusted. Since all connected quadrilaterals are related according to Eqn. 15, this means that only one point per set of quadrilaterals must solved. In the examples of the following section, Eqn. 5 reduced to 2 equations that can easily be solved with Gauss elimination. The algorithm is therefore fairly efficient.

Finding the connectivity between the quadrilaterals in a region is more difficult. The current implementation of the algorithm does this only for a specific arrangement of the quadrilaterals. Further research is required for a general arrangement.

3.4 Projecting the Control Points onto the Draft Plane

The last step is to project the control points of the affected quadrilaterals onto their respective draft planes. The back projection is done with Eqn. 15.

3.5 Algorithm Outline

In summary, the algorithm can be outlined as in the following figure. It starts out by checking each quadrilateral. If it does not satisfy the draft angle conditions, it is marked. The draft plane is then calculated for each marked quadrilateral and these quadrilaterals are grouped in regions of connected quadrilaterals. The linear system is then set up to recalculate the draft plane origins. While this is done, the connectivity constraint (Eqn. 15) is incorporated. The system is then solved. Lastly, the control points of the marked quadrilaterals are projected back onto the new draft planes.

4. LIMITATIONS

The algorithm has a number of limitations. Firstly it does not check for self intersections of the control net after adjustment. In certain cases this can happen with the result that it may cause self intersections in the surface itself.

For some surfaces, one adjustment of the control net may cause quadrilaterals that originally satisfied the draft conditions, to violate the conditions after adjustment. The current implementation of the algorithm does not check for such violations.

5. EXAMPLES

Three examples are given to demonstrate the algorithm. In the first example, draft is added to a general swept surface. The next example shows a lofted surface. The last example is a simple extruded surface and is included to compare this algorithm with the method of adding isocline surfaces proposed by [8]. In all the examples the pull direction is the Z-axis and the draft angle is 5°.

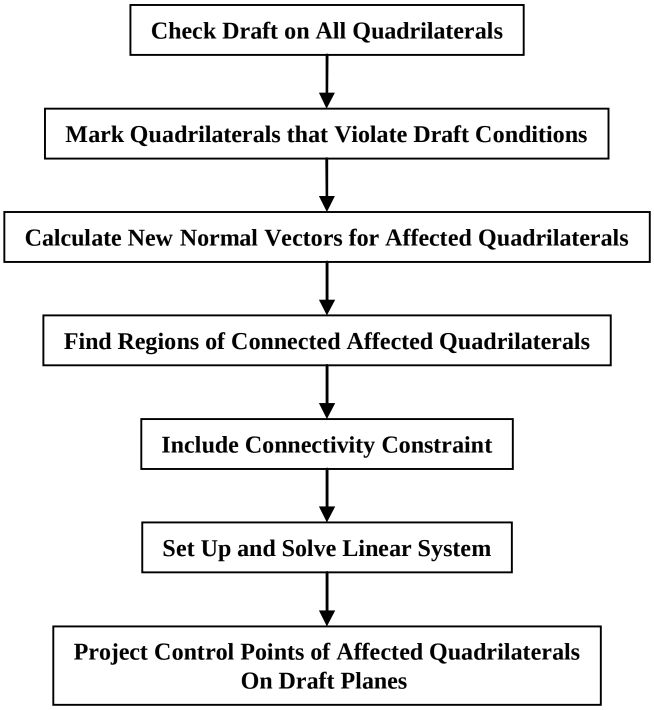

Fig. 4: Outline of draft angle algorithm.

5.1 Swept Surface

Fig. 5 shows a swept surface (on the left) with its control polygon. The dark areas close to the bottom edge of the surface indicate that the draft angle condition is violated there. Notice also that the control polygon also violates the draft condition (the quadrilaterals on the left and right sides are inclined "inward"). On the right in Fig. 4 the same surface is shown after it was modified with the draft angle algorithm. Notice that the dark areas at the bottom have disappeared indicating that the surface satisfies the draft condition everywhere. Notice also that the control polygon now also satisfies the draft condition (the quadrilaterals on the sides are now inclined "outward"). The draft angle algorithm therefore successfully modified the surface so that it now has a positive draft angle everywhere.

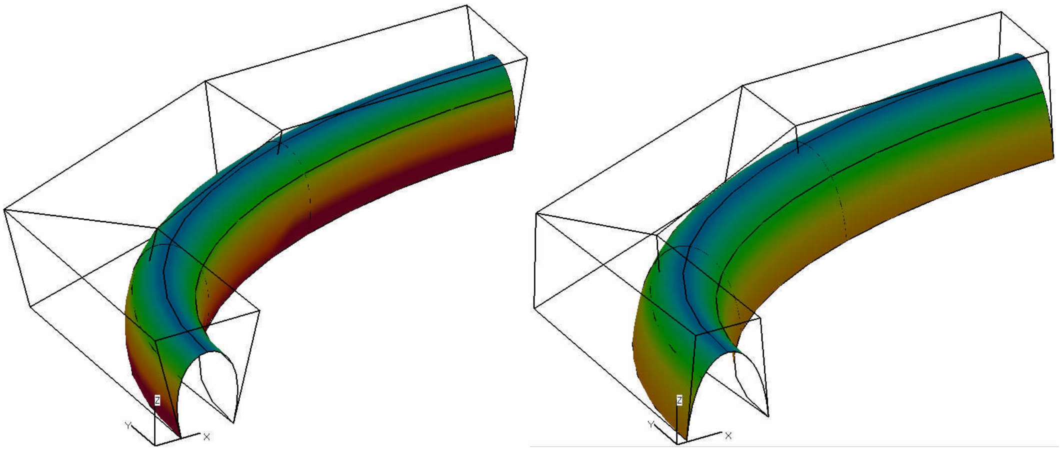

Fig. 5: Swept surface with control polygon: without draft (left) and with draft (right).

5.2 Lofted Surface

Fig. 6 shows a lofted surface (on the left) with its control polygon. The same observations made for the swept surface (Fig. 5) can be made for the lofted surface shown in Fig. 6. The original surface is symmetrical about its centre plane. However, the modified surface (Fig. 6, right) is no longer so. Fig. 7 shows the base splines of the lofted surface before (dashed) and after (solid) the surface was modified. When viewing the end points, the discrepancy can be seen clearly. Detailed checking of the algorithm revealed that the error lies with the calculation of the draft plane normal vectors (see Section 3.2). The algorithm uses a minimization method to find the angle θ in equation 11. A small error in θ (in the third decimal) causes a small error in the value of the normal vector.

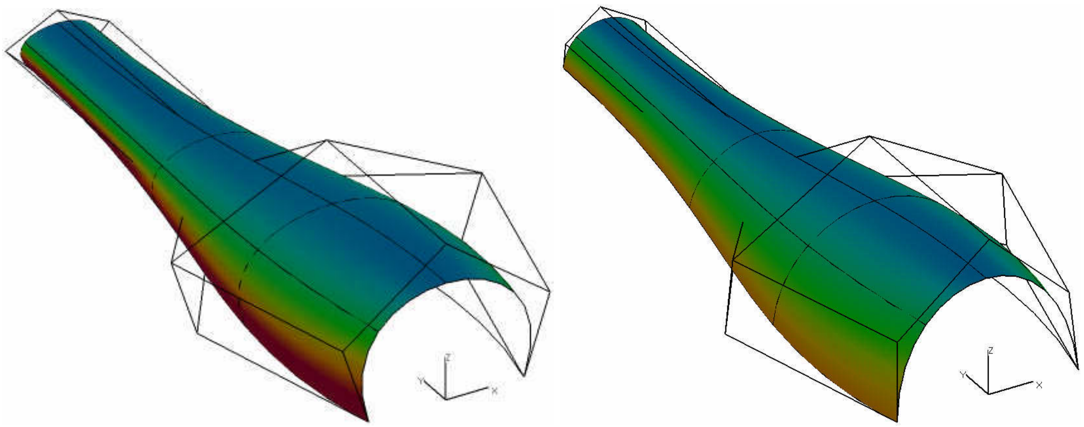

Fig. 6: Lofted surface with control polygon: without draft (left) and with draft (right).

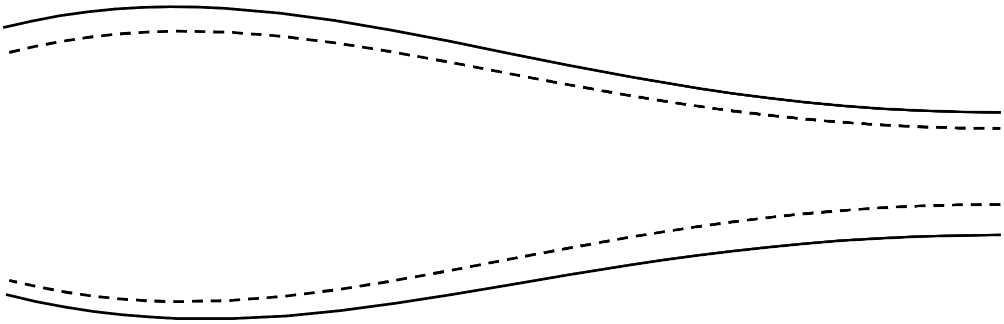

Fig. 7: Comparison of base splines of lofted surface; dashed splines are from the original surface, solid splines are from the modified surface.

5.3 Comparison of the Algorithm with the Isocline Surface Method

Fig. 8 shows two pictures of an extruded surface with draft angle already added. The surface on the left was modified with the algorithm proposed here and the surface on the right was modified by adding an isocline surface. The example is simple, but serves to illustrate a few important points. There is no visually noticeable difference between the two surfaces. However, an overlay of the two surfaces (Fig. 9) points out the difference. The figure shows an overlay of three surfaces. The original surface (without draft) is indicated with longer dashes. The isocline surfaces are indicated with shorter dashes. The surface after modification with the algorithm proposed in this paper is shown with the solid line. The horizontal line shows the isocline points.

Notice that the proposed algorithm changed the entire surface. In real design work, this is probably not desirable. A possible solution is to insert knots close to the isocline points. Due to the local modification property of B-splines, this will reduce the effect on the rest of the surface. However, it is not possible to completely eradicate this effect because of the way that the least squares minimization is done (see Section 3.1).

This algorithm preserves the continuity of the surface. The isocline surfaces are only G1 continuous at the isocline lines. There is also no deed to trim the surfaces and add new surfaces such as is the case with adding isocline surfaces.

One can also see that this algorithm is conservative. Notice in Fig. 9 that the end points of the solid lines are further apart than the end points of the isocline surfaces. This shows that the algorithm is conservative in this example.

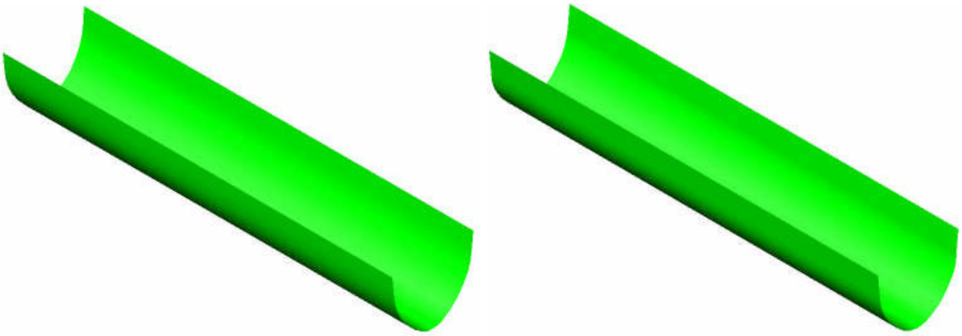

Fig. 8: Comparison of extruded surfaces with draft angle added with the developed algorithm (left) and using an isocline surface (right).

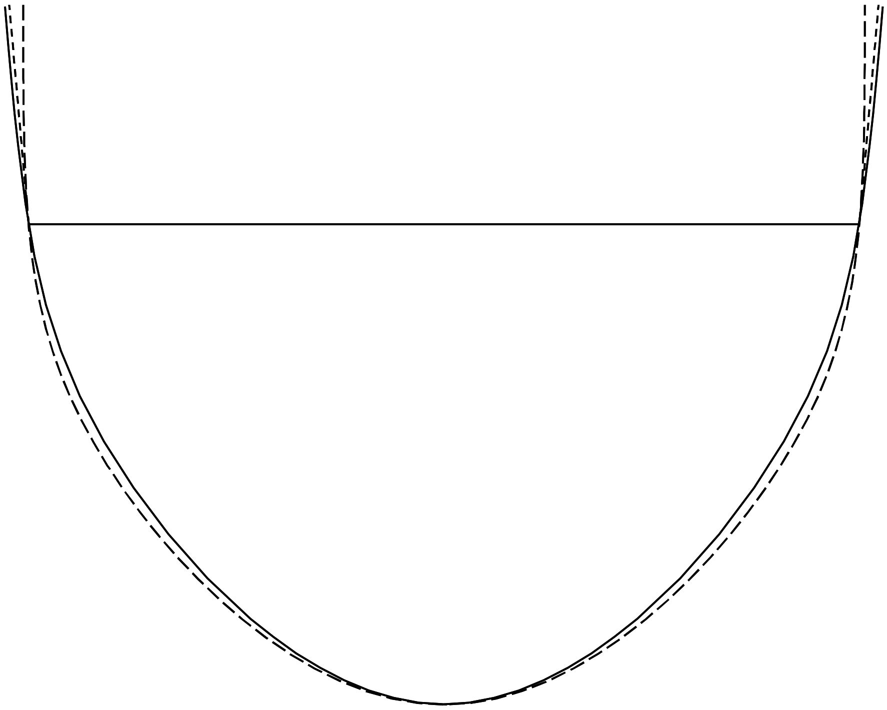

Fig. 9: Front end view (overlay) of the extruded surfaces. Short dashed surfaces are the isocline surfaces. The longer dashed surface is the original surface. The solid surface is the surface modified with the algorithm proposed here. The horizontal line indicates where the isocline points are.

6. CONCLUSIONS

An algorithm was presented to add draft angle to general B-spline surfaces. It was implemented and demonstrated with three examples. Although a reasonably complex minimization problem, the system was reduced significantly by incorporating constraints that maintains the connectivity between the quadrilaterals of the control net. The draft angle conditions given at the beginning of the paper are easy to evaluate and valid for any degree of the B-spline surface. It is therefore possible to add the draft angle without having to calculate derivatives on the surface.

The algorithm will have some limitations, e.g. deep narrow holes and intersecting surfaces. However, these are not practical examples. Therefore, for real engineering examples, as long at it is possible to add the draft angle on the real object, it should also be possible to add it using the algorithm.

The algorithm is conservative. Firstly because it projects all four points of the control polygon on the same draft plane. It will not always be necessary to adjust all four points of a quadrilateral. This will happen if one of the two draft conditions holds. The second reason is inherent to the draft conditions. By adjusting the entire quadrilateral, a larger section of the surface than may actually be necessary is changed. However, it is not clear yet how conservative the conditions are.

7. REFERENCES

[1] Golub, G. H.; Van Loan, C. F.: Matrix Computations, 3rd Edition, Johns Hopkins University Press, Baltimore, 1996.

[2] Khardekar, R.; Burton, G.; McMains, S.: Finding Feasible Mold Parting Directions Using Graphics Hardware, Computer-Aided Design, 38(4), 2006, 327-341.

[3] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P.: Numerical Recipies in C, 2nd Edition, Cambridge University Press, Cambridge, 1992.

[4] Schreve, K.: Edge Scanning and Sweep Surface Approximation in Reverse Engineering, Ph.D. Thesis, University of Stellenbosch, Stellenbosch, South Africa, 2002.

[5] Schreve, K.: Fitting B-splines with Draft Angle to Scanned Data, In: Proceedings of the 2nd Int Conf on Advanced Research in Virtual and Rapid Prototyping, 2005 September 28 –October 1, Leiria, Portugal, 269-275.

[6] Schreve, K.: Fitting 2D B-splines with Draft Angle to Point Data, Computer-Aided Design, 38(8), 2006, 836-844.

[7] Schreve, K.: Sufficient Conditions for Draft Angles on General B-spline Surfaces, Computer-Aided Design, 2008, submitted for publication.

[8] Tokuyama, Y.; Bae, S.: An Approximate Method for Generating Draft on a Free-form Surface, The Visual Computer, 15, 1999, 1-8.

[9] Yan, Y.; Tan, S. T.: Adding Draft Angles on Mechanical Components Containing Constant Radius Blending Surfaces, Computer-Aided Design, 36(7), 2004; 565-580.

[10] Yin, Z.-P.; Ding, H.; Li, H.-X.; Xiong, Y.-L.: Geometric Mouldability Analysis by Geometric Reasoning and Fuzzy Decision Making, Computer-Aided Design, 36(1), 2004, 37-50.