Original scientific paper
Accepted 14. 12. 2016.

Szilvia B.-S. Béla
Márta Szilvási-Nagy

Adjusting Curvatures of B-spline Surfaces by Operations on Knot Vectors

Adjusting Curvatures of B-spline Surfaces by Operations on Knot Vectors

ABSTRACT

The knot vectors of a B-spline surface determine the basis functions hereby, together with the control points, the shape of the surface. Knot manipulations and their influence on the shape of curves have been investigated in several papers (see e.g. [4] and [5]). The computations can be made very efficiently, if the basis functions and the vector function of the B-spline surface are represented in matrix form (see [1] and [6]). In our latest work [2] we summarized the knot manipulation techniques and the corresponding computations in matrix form. We also developed an algorithm for a direct knot sliding, how a knot can be repositioned in one step instead of inserting a new knot value, then removing an old one from the knot vector.

In this paper we analyse the effect of varying knot intervals on the Gaussian curvature of a B-spline surface at a given point. We present an algorithm for the deformation of a B-spline surface, so that it should go through a given point with a given Gaussian curvature. The result of this deformation is, that a sphere with a given radius will fit tangential the reshaped surface at the given point with equal Gaussian curvatures. In applications the same situation arises, when a ball-end tool is pushed into a surface during processing.

In our algorithm we use only linear interpolation equations besides the repositioning of knot values, in order to get numerically stable and effective solutions.

Key words: surface representations, geometric algorithms

MSC2010: 65D18, 68D05

Podešavanje zakrivljenosti B-splajn ploha operacijama na čvor vektorima

SAŽETAK

Na ovaj način čvor vektori B-splajn ploha određuju temeljne funkcije zajedno s kontrolnim točkama te oblik plohe. U nekoliko članaka (vidi na primjer [4] i [5]) proučavale su se operacije na čvorovima i njihov utjecaj na obilk krivulja. Izračuni mogu biti izvedeni vrlo efikasno ako su temeljne funkcije i vektor funkcije B-splajn plohe prikazane u matričnom obliku (vidi [1] i [6]). U našem posljednjem radu [2] saželi smo metode operacija na čvorovima i odgovarajućih izračuna u matričnom obliku. Također, razvili smo algoritam za izravno klizanje čvorova, tj. pokazali smo kako čvor može biti premješten u jednom koraku umjesto da uvodimo novu vrijednost čvora, a zatim uklanjanjem starog iz čvor vektora.

U ovom članku analiziramo utjecaj mijenjanja intervala čvorova na Gaussovu zakrivljenost u zadanoj točki B-splajn plohe. Prikazujemo algoritam za deformaciju B-splajn plohe tako da prolazi kroz zadanu točku sa zadanom Gaussovom zakrivljenosšću. Rezultat ove deformacije kaže da kugla zadanog radijusa dira preoblikovanu plohu u zadanoj točki s jednakim Gaussovim zakrivljenostima. Ista situacija događa se u primjenama kad je alat kuglama uguran u plohu tijekom procesa.

S ciljem da postignemo numerički stabilna i efikasna rješenja, osim premještanja vrijednosti čvora, u našem algoritmu koristimo samo jednadžbe linearne interpolacije.

Ključne riječi: prikazi plohe, geometrijski algoritmi

1 Problem solution in the symmetric case

The mathematical formulation of the problem

Our task is to push a given sphere into a B-spline surface by reshaping a part of it around a given common interpolation point such that, the surface and the sphere are in a tangential position and they have equal Gaussian curvatures at this point (Fig 1, Fig 2). We only address the geometrical side of the problem, but not the mechanical aspects. Furthermore, we do not set conditions on area or volume preserving. We just focus on this geometrical design problem.

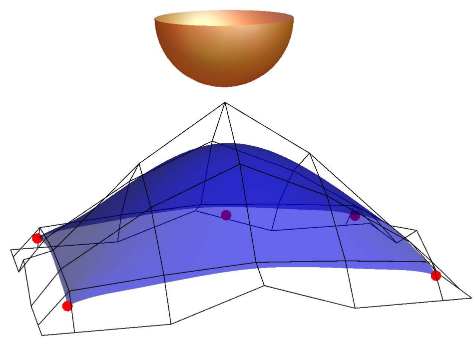

Figure 1: *The sphere will be pushed into the interpolation point given in the middle.*

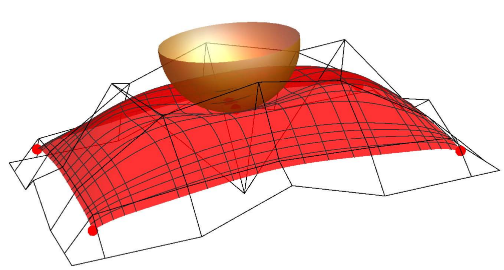

Figure 2: *The deformed surface and the sphere have equal Gaussian curvature at the common point.*

The shape of the surface can be controlled by its control points and by the parametrization of the basis functions, that means, by the knot vectors. The interpolation problem with a prescribed Gaussian curvature leads to quadratic rational expressions of the surface data, but our algorithm avoids nonlinear numerical methods by choosing appropriate variables, interpolation conditions and by applying a simple iteration method.

Let the B-spline surface of degree 3×3 be given with non-uniform periodic knot vectors $(u_1 < u_2 < \cdots < u_{n+4})$ and $(v_1 < v_2 < \cdots < v_{m+4})$, $n,m \geq 4$. The vector function representing the B-spline surface is

$$\mathbf{b}(u,v) = (u^3, u^2, u, 1) \cdot \left(\mathbf{N}_\mathbf{u}^4\right)^T \cdot \mathbf{Q} \cdot \mathbf{M}_\mathbf{v}^4 \cdot (v^3, v^2, v, 1)^T,$$

where $\mathbf{N}_\mathbf{u}^4$ and $\mathbf{M}_\mathbf{v}^4$ are the corresponding coefficient matrices of the basis functions and $\mathbf{Q}$ is the matrix of the control points $\mathbf{q}_{i,j}$, $(i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m)$.

We will restrict the computation to a region of 4×4 patches of the B-spline surface because this part is influenced by the second order curvature condition prescribed in the middle of this region. In this case $n,m \geq 7$. The other parts of the surface outside of this region remain unchanged.

The input data of the B-spline surface are the knot vectors (i.e. the parameter values) and the control points. The parameter grid with the actual region is shown in Fig 3, the control net with the generated surface is shown in Fig 4. In these examples the control net and the knot vectors are symmetrical about the midpoint of the actual region, therefore, the generated surface is also symmetrical about this point.

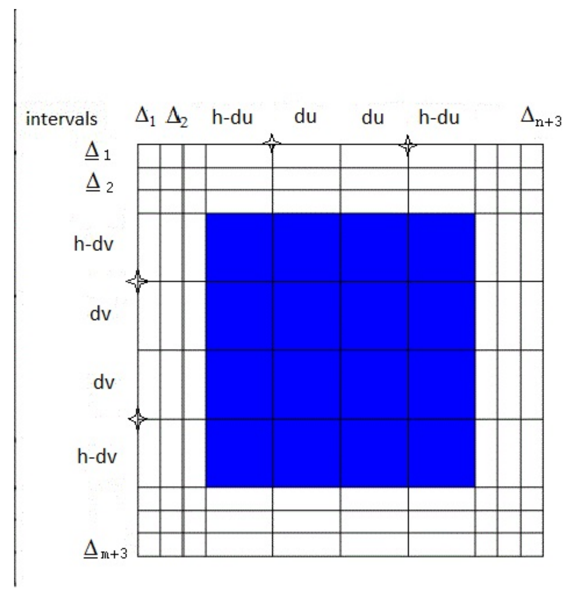

Figure 3: *Parameter domain of the surface:* 4×4 *surface patches are determined by the net of* 11×11 *knot values, (h is constant).*

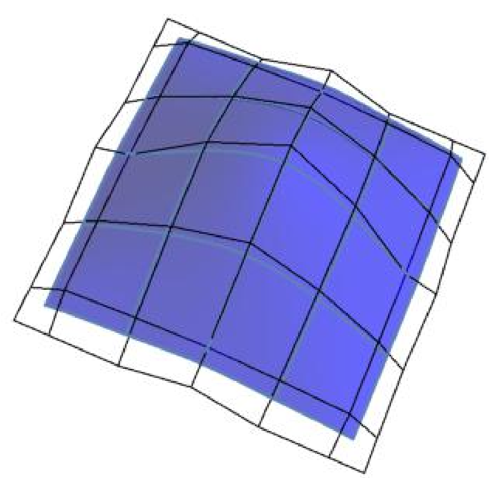

Figure 4: 7×7 *control points and the generated surface.*

First phase of the deformation: interpolation

For the deformation of the surface we prescribe 9 interpolation points and the Gaussian curvature at the interpolation point P in the middle, where the sphere with the corresponding radius will touch the reshaped surface. Four interpolation points are given in the corners of the B-spline surface. Five interpolation points, namely P and the four corner points are shown in Fig 1. The remaining four interpolation points are determined around P according to the radius of the given sphere symmetrically with respect to P (Fig 5). They are computed on the surface of the sphere. The corresponding parameter values of the required surface are estimated by the relative measurements of the surface patches and the sphere. To the nine interpolation conditions we choose nine variables, which are nine inner control points of the 7×7 control net (Fig 6).

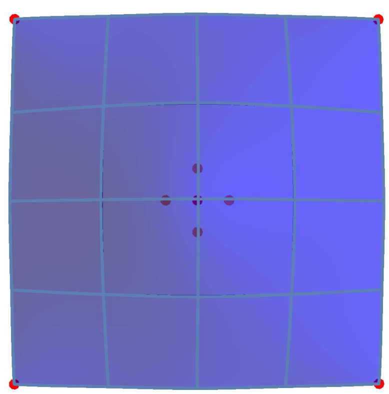

Figure 5: *The nine interpolation points*

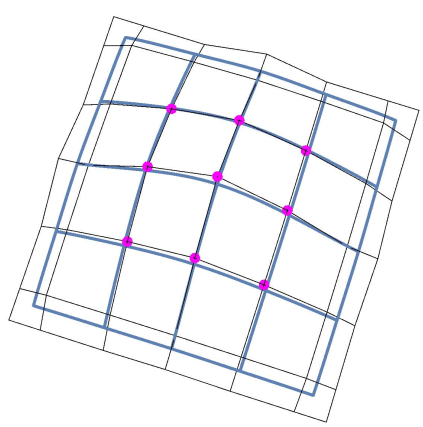

Figure 6: *Nine variable control points in the 7×7 control net*

Each of the nine interpolation conditions is a linear vector equation in the nine control points, which are the unknowns of a system of linear equations.

$$\mathbf{p}_i = \mathbf{b}(u_i, v_i),\ i = 1, \ldots, 9$$

The knot vectors are now fixed, and the coefficient matrices are computed accordingly. On the right hand side the vector function $\mathbf{b}(u,v)$ is depending on the nine unknown control points and it is evaluated at the parameter values (u_i, v_i). The unknown nine control points are included in the matrix $\mathbf{Q}$. The pairs of the parameter values (u_i, v_i) belong to the interpolation points, the position vectors of which are denoted by $\mathbf{p}_i$. The solution results in a control net of the B-spline surface interpolating the nine given points. Its Gaussian curvature at the midpoint P is now determined. How can it be equal to the given value?

Second phase of the deformation: adjusting the Gaussian curvature

Now we modify the knot vectors in order to deform the shape of the surface around the interpolation point P. In the knot vectors four knot intervals in the middle of the knot vectors will be changed by repositioning (sliding) the knot values $u_5 \in (u_4, u_6)$, $u_7 \in (u_6, u_8)$ and symmetrically $v_5 \in (v_4, v_6)$, $v_7 \in (v_6, v_8)$, respectively. These knot values are marked in Fig 3. The variables in the knot vectors are du and dv. For smaller du, dv the generated B-spline surface gets nearer to the control net, for larger du, dv it is more flat, and lies farther from the control net.

If we change the knot intervals $du = dv$ and analyse, how the Gaussian curvature of the surface (denoted by κ_G) at the point P changes, we get a monotone scalar function $\kappa_G(du)$. In Fig 7 the corresponding $radius = (\sqrt{\kappa_G})^{-1}$ of the sphere is shown depending on $du = dv$.

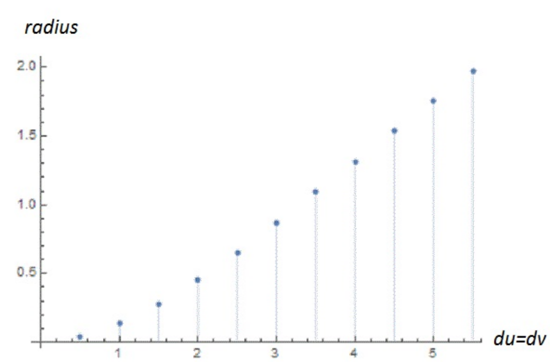

Figure 7: *The radius of the sphere as function of the length of the knot intervals $du = dv$*

The Gaussian curvature is varying in a limited interval while the chosen parameter values are repositioned in the intervals (u_4, u_6), (u_6, u_8) and (v_4, v_6), (v_6, v_8) of a fixed length h, respectively. To a given Gaussian curvature between these limits the corresponding knot intervals $du = dv$ are determined by a simple iteration from this scalar function.

Then the required surface is generated with the new knot vectors. The result of this computation is shown in Fig 2. The point P is an umbilical (a special elliptical) point of the new surface due to the symmetrical data and symmetrical change of the knot vectors. After this deformation the boundary curves of the surface consisting of 4×4 patches do not change, as it is shown in Fig 8.

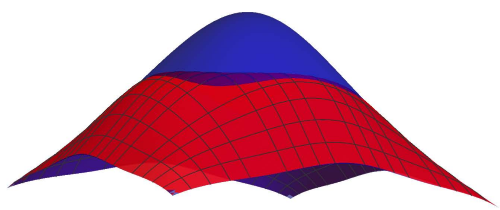

Figure 8: *The original and the deformed surfaces have the same boundary curves.*

2 The asymmetric case

In the non-symmetrical case the deformation presented above leads to an elliptic surface point P, where the main curvature values are different, though the Gaussian curvature is equal to that of the given sphere. This situation is shown in Fig 9 by pushing the sphere a bit into the deformed surface, where the intersection has an elliptical form.

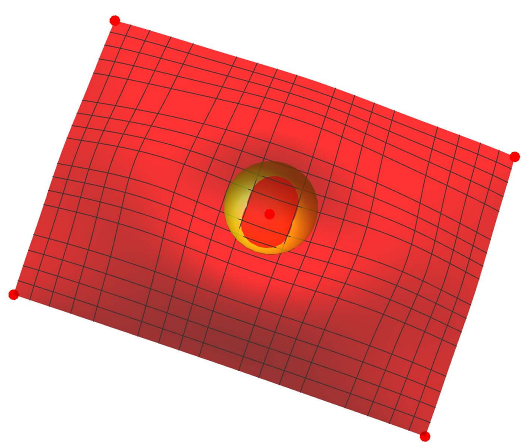

Figure 9: *Elliptical surface point at P with different main curvatures*

Now we carry out a further deformation in order to get a special elliptic, i.e. an umbilical point at P. We repeat the second phase of the deformation by changing the knot intervals, now only in one parameter direction, let us say, on the v-knot vector. For the chosen values of dv within the intervals (v_4, v_6) and (v_6, v_8) we compute the curvatures of the parameter curves at the surface point P. Assuming that the parameter net is orthogonal, the Gaussian curvature is the product of the curvatures of the u- and v-parameter curves. In Fig 10 the dependence of the curvature of the v-parameter curve, denoted by $v-curvature$, on the knot interval dv is shown. Meanwhile the values of the $u-curvatures$ are practically constant (they are not shown). We determine the knot interval dv to the $v-curvature$ which is equal to the $u-curvature$ from this monotone discrete function by simple iteration. The computed surface with this knot vector has an umbilical point at P. That is visualized by the sphere pushed slightly into the surface at the touching point (Fig 11).

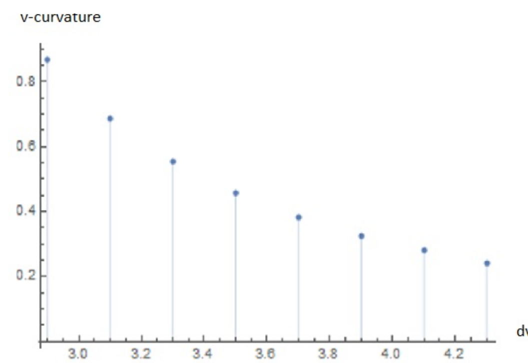

Figure 10: *The values of the $v-curvatures$ at the values of the knot interval dv*

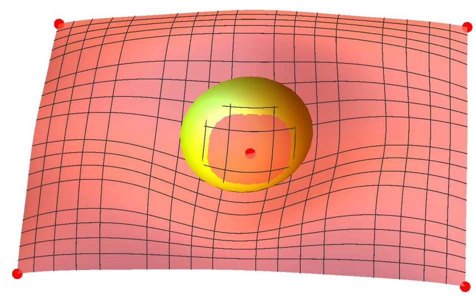

Figure 11: *Umbilical surface point at P with equal main curvatures*

We remark that by this second deformation the Gaussian curvature has been changed slightly. Analysing this variation, an appropriate correction could be carried out in a similar way, repeating the computation for modified values of du and dv, if the difference in the radii of the touching spheres computed from the resulting Gaussian curvatures is not acceptable. We have omitted this step.

We remark also that there are no equal values of u- and v-curvatures on the considered interval, if the surface is strongly asymmetric around the point P. In this case the parameter values of the four inner interpolation points have to be corrected accordingly.

3 The mathematical tools

Matrix form of the basis functions

The B-spline surface in our algorithm is presented in matrix form. This form allows to write the interpolation conditions in the form of explicit vector equations. In [1] we have given a short overview of the published papers about the matrix form of B-spline functions, and we also have given a method for generating the entries of the coefficient matrices of any degree over periodic knot vectors.

In our algorithm we have changed the position of a knot value within a given knot interval in order to analyse the change of the Gaussian curvature at a given surface point. In each step the corresponding coefficient matrices have been generated for the vector function representing the B-spline surface.

In order to represent the basis functions of non-uniform B-splines in matrix form first we describe a reformulation technique of the de Boor-Cox recursion. From this recursion formula we can generate the representation matrix of the basis functions in the not normalized Bernstein basis. Then we can apply a simple transformation from the not normalized Bernstein basis to the polynomial space spanned by the power basis. Thus with the algebraic reformulation of the B-spline recursion we gain the conversion matrices of the B-spline functions to the power basis.

First we present here a simple reformulation of the de Boor-Cox recursion. The basis functions of order k over the knot vector $\{t_1,\dots t_n\}$ are defined by the de Boor-Cox formula as:

$$\begin{aligned} N_i^1(t) &= \begin{cases} 1, & t\in[t_i,t_{i+1}) \\ 0, & \text{otherwise,} \end{cases} \\ N_i^k(t) &= \alpha(t_i,t_{i+k-1};t)\,N_i^{k-1}(t)+\alpha(t_{i+k},t_{i+1};t)\,N_{i+1}^{k-1}(t), \end{aligned}$$

where the function α is defined as

$$\alpha(A,B;t)=\frac{t-A}{B-A} \tag{1}$$

for arbitrary parameters A,B, where $A\neq B$, and for all $t\in[A,B]$.

We can generate the pieces of the basis functions restricted to one knot interval $[t_j,t_{j+1})$ by rewriting the recursion as

$$\begin{aligned} N_i^1(t) &= \begin{cases} 1, & t\in[t_i,t_{i+1}) \\ 0, & \text{otherwise,} \end{cases} \\ N_i^k(t) &= \alpha(t_j,t_{j+1};t)\Big[\alpha(t_i,t_{i+k-1};t_{j+1})\,N_i^{k-1}(t) \\ &+ \alpha(t_{i+k},t_{i+1};t_{j+1})\,N_{i+1}^{k-1}(t)\Big] \\ &+ \alpha(t_{j+1},t_j;t)\Big[\alpha(t_i,t_{i+k-1};t_j)\,N_i^{k-1}(t) \\ &\quad +\alpha(t_{i+k},t_{i+1};t_j)\,N_{i+1}^{k-1}(t)\Big], \text{ where } t\in[t_j,t_{j+1}). \end{aligned} \tag{2}$$

According to this form we transform all segments of the basis functions from the knot span $[t_j,t_{j+1})$ and represent them over the unit interval as follows:

$$\begin{aligned} N_i^1(t(u)) &= \begin{cases} 1, & i=j \\ 0, & \text{otherwise,} \end{cases} \\ N_i^k(t(u)) &= u\Big[\alpha(t_i,t_{i+k-1};t_{j+1})\,N_i^{k-1}(t(u)) \\ &+ \alpha(t_{i+k},t_{i+1};t_{j+1})\,N_{i+1}^{k-1}(t(u))\Big] \\ &+ (1-u)\Big[\alpha(t_i,t_{i+k-1};t_j)\,N_i^{k-1}(t(u)) \\ &\quad +\alpha(t_{i+k},t_{i+1};t_j)\,N_{i+1}^{k-1}(t(u))\Big], \end{aligned} \tag{3}$$

where $u\in[0,1), t\in[t_j,t_{j+1})$ and $u(t)=\alpha(t_j,t_{j+1},t)$. Over the knot spans, where $j=k,k+1,\dots,n-k$ the basis functions have k different, non-zero polynomial segments. These segments can be represented by a matrix equation in the not normalized Bernstein basis $\{u^{k-1},u^{k-2}(1-u),\dots,(1-u)^{k-1}\}$ over the unit interval:

$$\begin{pmatrix} N_1^k(t(u)) \\ N_2^k(t(u)) \\ \vdots \\ N_{n-k}^k(t(u)) \end{pmatrix} = \mathbf{C}^k\cdot\begin{pmatrix} u^{k-1} \\ u^{k-2}(1-u) \\ \vdots \\ (1-u)^{k-1} \end{pmatrix}, \quad \begin{aligned} &t\in[t_j,t_{j+1}), \\ &u\in[0,1), \end{aligned} \tag{4}$$

where $\mathbf{C}^k\in\mathbb{R}^{n-k\times k}$, and it contains the coefficients of $u^m(1-u)^{k-1-m}$ computed recursively from (4). For each $k=2,3,\dots$ this matrix contains several rows, where all elements are zeros. These rows contain the coefficients of the basis functions that are zero over the knot span $[t_j,t_{j+1})$. The non-zero rows for each j (from row $j+1-k$ to row j), where $j\geq k$, give the coefficients of the basis functions with the support containing the interval $[t_j,t_{j+1})$.

If we represent the segments of the basis functions from a given knot span $[t_j,t_{j+1})$ in the matrix equation form (4), then it is easy to transform this representation to a matrix representation in the power basis: $\{u^{k-1},u^{k-2},\dots,u,1\}$. In order to find the transformation matrix of the basis functions to the power basis, it is sufficient to find the transformation matrix $\mathbf{P}^k$ from the not normalized Bernstein basis to the power basis for polynomials of degree $k-1$:

$$\begin{pmatrix} N_1^k(t(u)) \\ N_2^k(t(u)) \\ \vdots \\ N_{n-k}^k(t(u)) \end{pmatrix} = \mathbf{C}^k\cdot\mathbf{P}^k\cdot\begin{pmatrix} u^{k-1} \\ u^{k-2} \\ \vdots \\ 1 \end{pmatrix}, \quad \begin{aligned} &t\in[t_j,t_{j+1}), \\ &u\in[0,1). \end{aligned} \tag{5}$$

The $\mathbf{P}^k$ matrix is a lower triangular matrix with the entries:

$$\mathbf{P}^k[i,l] = \begin{cases} (-1)^{i-l+1} \cdot \binom{i-1}{l-1}, & l \leq i, \\ \\ 0, & \text{otherwise,} \end{cases}$$

where l and $i = 1, \ldots, k$. This matrix can be easily derived according to the Binomial-theorem. A conversion matrix from the not normalized Bernstein basis to the power basis can be also found in the literature [3].

Interpolation and iteration

These mathematical tools are well-known basic methods in solutions of various numerical problems. In our algorithm the linear interpolation problem formulated by a system of linear vector equations has been separated from the non-linear interpolation problem, where a scalar value (i.e. the Gaussian curvature of the surface) has to be interpolated with vector variables. We have solved this problem by simple iteration on a monotone, scalar function by computing discrete values of this function in an appropriate interval. This method is fast and numerical stable. In this way we have avoided non-linear numerical problems.

The computations and the figures have been made by the symbolical algebraic program package Wolfram Mathematica.

4 Remarks

In our algorithm there are some basic assumptions. The first one is that the local deformation of a B-spline surface of degree 3×3 has been computed on a region of 4×4 patches, where the fixed point lies in the middle of this region. Though each control point and each basis function has an effect on four knot intervals, the boundary curves of this part of the surface have not changed under the used interpolation conditions.

The next assumption is that the prescription of the four inner interpolation points are computed from the data of the given sphere. The setting of the parameter values for which the surface interpolates these points has been made on the base of experimental results, as usually in the solutions of many practical problems.

A further simplification in the solution is that we have computed with surfaces on orthogonal parameter grids. In this way we have computed the Gaussian curvature as the product of the curvatures of the u- and v-parameter curves. In this way we have avoided the numerical computation of the main curvatures in each step, because this computation is not essential in our algorithm.

5 Conclusions

We have given an algorithm for the solution of a practical problem: how to press a given sphere into a B-spline surface at a prescribed position. We have shown a solution, when the surface is symmetric around the given interpolation point. Then in the non-symmetric case we have shown a further deformation of the surface in order to transform the elliptical surface point into an umbilical one. In this case the given sphere osculates the deformed surface with equal Gaussian curvatures at the given tangential point.

Acknowledgement

The research of the second author was supported in a co-operation with the Technical University Berlin.

References

[1] SZ. BÉLA, M. SZILVÁSI-NAGY, General matrix representation of B-splines and approximation of B-spline curves and surfaces with third order continuity, *Seventh Hungarian Conference on Computer Graphics and Geometry*, Budapest, 2014, 1–6.

[2] SZ. BÉLA AND M. SZILVÁSI-NAGY, Manipulating end conditions of B-spline curves using knot sliding, *Eighth Hungarian Conference on Computer Graphics and Geometry*, Budapest, 2016, 83–93.

[3] G. FARIN, *Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide*, Academic Press, Second Edition, 1990.

[4] I. JUHÁSZ, M. HOFFMANN, The effect of knot modiications on the shape of B-spline curves, *Journal for Geometry and Grapics* **5** (2001), 111–119.

[5] I. JUHÁSZ, M. HOFFMANN, Constrained shape modification on cubic B-spline curves means of knots, *Computer-Aided Design* **36** (2004), 437–445.

[6] M. SZILVÁSI-NAGY, SZ. BÉLA, Stitching B-spline curves symbolically, *KoG* **17** (2013), 3–8.

Szilvia B.-S. Béla
e-mail: belus@math.bme.hu

Márta Szilvási-Nagy
e-mail: szilvasi@math.bme.hu

Department of Geometry, Budapest University of Technology and Economics

Budafoki út 8., H-1521 Budapest, Hungary