MESH GENERATION BASED ON THE PARAMETERIZATION OF NURBS SURFACE THAT CAN BE USED FOR SELF-ORGANIZED POTENTIALL BASED SHAPE DETERMINATION

Xin Li

Technical University of Clausthal (TUC)
Arnold-Sommerfeld-Str. 6
38678 Clausthal-Zellerfeld, Germany
xin.li@tu-clausthal.de

Dietmar P. F. Möller

Technical University of Clausthal (TUC)
Erzstr. 1
38678 Clausthal-Zellerfeld, Germany
dietmar.moeller@tu-clausthal.de

ABSTRACT

In this article Non-Uniform Rational B-Spline (NURBS) surface is used to build mathematical models with various forms that can be used in a study for self-organized potential based shape determination. This requires that the NURBS model can be discretized into a hexagonal mesh like a graphene structure. Thereafter, the mesh is utilized for the determination based on the self-organization process.

Keywords: Non-uniform Rational B-Spline, hexagonal mesh, self-organization process

1 INTRODUCTION

This research is a part of the PhD program "Self-Organizing Multifunctional Structures for Adaptive High Performance Lightweight Constructions". In this program new structures should be developed, so that the entire constructions can achieve high performance, based on the weight of constructions which is as light as possible. An appropriate discrete event simulation approach is important to use for prediction and verification. This approach requires a model in arbitrary surfaces in $\mathbb{E}^3$ space. In geometric design, geometry of complex surfaces is represented in terms of polynomial functions, as described in the literature, as shown in (Bezier 1986; Piegl and Tiller 1987; Prautzsch et al. 2002; Rogers 2001; Pressely 2010), the Hermit interpolation polynomials (Ciarlet and Raviart 1972), the confluent Vandermode Matrices (Respondek 2013; Respondek 2016) and others. Examples for complex geometric surfaces are:

- Bézier Surfaces: A Bézier surface is defined by a set of control points. Similar to interpolation, a key difference is that the surface does not pass through the central control points; rather, it is stretched toward them as though each were an attractive force. They are visually intuitive; and are used for many applications. A dimension count shows that the *n+1* linearly independent Bernstein polynomial B_1^n is the basis for all polynomials of degree $\leq n$. Therefore, every polynomial curve $b(u)$ of degree $\leq$ n has a unique *nth* degree Bézier representation

  $$b(u) = \sum_{i=0}^{n} c_i B_i^n(u)$$

  Since the Bernstein polynomial represents a basis, every polynomial surface $b(x)$ has a unique Bézier representation with respect to the reference simplex A. The coefficient b_i is called the Bézier point of b. They are the vertices of the Bézier net of $b(x)$ over the simplex A (Prautzsch et al. 2002).

- <u>B-spline Surface:</u> With regard to the Bézier representation of polynomial curves, it is desirable to write a spline $s(u)$ as an affine combination of some control points c_i as follows:

$$s(u) = \sum c_i N_i^n(u)$$

where the $N_i^n(u)$ are basic spline functions with minimal support and certain continuity properties. Schoenberg introduced the name B-splines for those functions (Schoenberg 1967). The B-spline can be defined by the recursion formula

$$N_i^0(u) = \begin{cases} 1 & if\ u_i \in u[a_i, a_{i+1} \\ 0 & otherwise \end{cases},$$
$$and$$
$$N_i^n(u) = a_i^{n-1} N_i^{n-1}(u) + (1 - a_{i+1}^{n-1} N_{i+1}^{n-1}(u)),$$
$$where$$
$$a_i^{n-1} \frac{(u - a_i)}{(a_{i+n} - a_i)}$$

is the local parameter with respect to the support of N_1^{n-1} (Prautzsch et al. 2002). B-spline surface permits the use of more control points in the characteristic polyhedron while retaining low order basis functions. B-spline basis functions are nonzero only over a given finite interval and enable the effect of a control point on the surface shape to be localized. Another advantage of the B-spline formulation is its ability to preserve arbitrarily high degrees of continuity over the complex surface patch. These characteristics make the B-spline surfaces popular for use in an interactive modeling environment (Woodward 1987).

- <u>Rational B-splines:</u> Standard for surface modeling in CAD and computer graphics. Any typical surface forms, such as flat planes and quadratic surfaces, e.g., cylinders, spheres, ellipsoids of revolution, as well as more complex fully sculptured surfaces, are easily and accurately represented by rational B-spline surfaces. As with rational curves, rational forms of Bézier surfaces are possible. A Cartesian product rational B-spline surface in four-dimensional homogeneous coordinate space is as follows:

$$Q(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}^h N_{i,k}(u) M_{j,l}(w)$$

where $N_{i,j}^h s$ are the four-dimensional homogeneous polygonal control vertices, $N_{i,k}(u)$ and $M_{j,l}(w)$ are the non-rational B-spline basis functions. An algorithm for a simple rational B-spline surface pseudocode is given in (Rogers 2001).

- <u>Nonuniform Rational B-splines:</u> Nonuniform rational B-splines are commonly referred to as NURBS, have become the standard for representation, design, and data exchange of geometric information processed by computers. NURBS provides a unified mathematical basis for representing analytic shapes, such as conic sections and quadratic surfaces, as well as free-form entities, such as car bodies and ship hills (Körber and Möller 2003). NURBS are generalizations of nonrational B-splines and rational and nonrational Bézier curves and surfaces (Piegl and Tiller 1997). A rational B-spline curve is the projection of a nonrational polynomial B-spline curve defined in four-dimensional homogeneous coordinate space back into the three-dimensional physical space which results in

$$P(t) = \sum_{i=1}^{n+1} B_i^h N_{i,k}(t)$$

where B_i^h are the four-dimensional homogeneous control polygon vertices for the non-rational 4D B-spline curve, and $N_{i,k}(t)$ is the non-rational B-spline basis function.

Vector valued polynomials with convenient properties are valuable for complex modeling and simulation purposes. They are mainly utilized for industrial and academic developments. A NURBS surface $S(u,v)$ can be defined as follows:

$$S(u,v)=\frac{\sum_{i=0}^{m}\sum_{j=0}^{n}P_{i,j}w_{i,j}N_{i,p}(u)N_{j,q}(v)}{\sum_{i=0}^{m}\sum_{j=0}^{n}w_{i,j}N_{i,p}(u)N_{j,q}(v)}$$

$$0\le u,v\le 1$$

with the control point matrix

$$P_{i,j}=\begin{Bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,n} \end{Bmatrix}, P_{i,j}=(x,y,z)$$

the basic functions

$$N_{i,0}(u)=\begin{Bmatrix} 1 & \text{if } u_i\le u\le u_{i+1} \\ 0 & \text{otherwise} \end{Bmatrix},$$

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

and the knot vector $U=\{u_o,\ldots,u_m\},u_i\le u_{i+1}$. The surface $S(u,v)$ has $(m+1)\times(n+1)$ control points $P_{i,j}$, and weights $w_{i,j}$.

Assuming the degrees of basic functions along the *u* and *v axes* to be *p*-1 and *q-1*, respectively, the number of knots is $(m+p+1)\times(m+q+1)$. The non-decreasing knot sequence is $t_0\le t_1\le \cdot\cdot\cdot \le tm+p$ along the *u* direction and $s_0\le s_1\le \cdot\cdot\cdot \le s_{n+l}$ along the *v* direction with the parameter domain in the range: $t_{p-1}\le u\le tm+1$ and $s_{q-1}\le v\le s_{n+1}$. If the knots have multiplicity *p* and *q* in the *u* and *v* directions, respectively, surface computation will interpolate the four corners of the boundary control points. In (Piegl and Tiller 1997), an algorithm for relatively fast computation of a vertex on a NURBS surface is discussed, exploiting redundancies and the property that most basic functions, the $N_{i,p}(u)$ and $N_{j,q}(v)$, evaluate to zero for given *u* and *v*.

The key to rendering complex objects lies in computing and rendering only those parts of the object which are visible to the viewer. This, in particular, is challenging in the case of complex surface rendering because it's impossible to perform any kind of space partitioning and utilize one of the traditional occlusion culling methods. The reason for this property is that for a given point of view, the whole surface could possibly be overlooked. Moreover, omitting only those parts of the object which don't lie in the viewing area of the viewer doesn't suffice. The remaining triangle count could be far too high. The key to surface rendering lies in the level of detail (LOD), i.e., rendering those parts of the surface which are far away from the viewer or which are rather smooth with less detail, i.e., less triangles and those parts which are close to the viewer or which are rough with more detail, which works for large objects only as small objects are always perceived in full detail (Piegl and Tiller 1997).

2 PARAMETRIZATION AND MESHING METHODS

The generation of mesh with regard to the NURBS methods is used to start at any point for surfaces of different shapes. The received mesh spread out on surfaces in 2D until a hexagonal mesh appears. The surface is regarded as a 2-manifold glued by numerous tiny pieces, which are homeomorphic locally to disks in $\mathbb{E}^2$, as shown in Figure 1. Beside the illustrated 2D computer graphic 3D digital image synthesis principles are of importance for mesh generation. The book of Andrew Glassner (1993) gives a good insight.

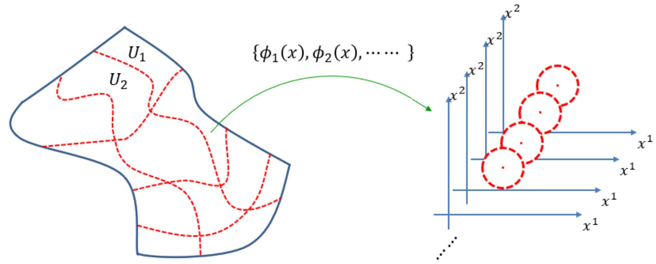

Figure 1: A manifold can be locally mapped into a Euclidean space with n dimensions

2.1.1 Parametrization by NURBS surface

A surface is parameterized by non-uniform rational B-splines surface (NURBS Surface). As described before, NURBS uses control points and weights under two non-uniform knot vectors. A curve can be established more readily and flexibly within the local support. Similarly, a surface can easily be developed following the formula:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} \boldsymbol{N}_{i,p}(u)\boldsymbol{N}_{j,q}(v)\omega_{i,j}\boldsymbol{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} \boldsymbol{N}_{i,p}(u)\boldsymbol{N}_{j,q(v)}\omega_{i,j}} \qquad 0 \le u, v \le 1 \qquad (1)$$

where: control net $\{\boldsymbol{P}_{i,j}\}$,
degree $\{p, q\}$,
weights $\{\omega_{i,j}\}$,
basic functions $\boldsymbol{N}_{i,p}(u)$ *and* $\boldsymbol{N}_{j,q}(v)$,
knot vectors $\{0,0,0, \cdots u_1, u_2, \cdots, 1,1,1\}$ *and* $\{0,0,0, \cdots v_1, v_2, \cdots, 1,1,1\}$.

We consider this parametric function as a deformation of a $[0,1] \times [0,1]$ square plane to a spatial surface with scaling along to parametric directions, shearing and bending. This map is also bijective when the NURBS surface is opened, which means the points on the plane and on the surface are one-to-one corresponding. Therefore, we can directly calculate the mesh points on the plane based on (1). Another important property is that, the surface is p or q-degree continuous ($p-1$ or $q-1$-degree continuous at knots without multiplicity). This ensures that the deformation of plane is gradual and the deformation at a point can represent the deformation in a small enough neighborhood around this point.

2.2 Meshing method

As mentioned previously, a surface is going to be approximated with tiny planes, on which new mesh points are produced based on previous mesh points. These planes are defined by the derivate of two parameters, which vary according to the previous mesh points. In order to generate a regular hexagonal mesh, a regular triangle is found at first, and then a hexagonal is embedded into a triangle (see Figure 2).

Assume that, the side length of the triangle is s, then the first step is to find a vector $\left(\frac{s}{2}, \frac{\sqrt{3}s}{2}\right)^T$ at the local coordinate system $u'Ov'$, whose basis vectors are the derivatives of two parameters and the origin locates at the previously given point, and then the endpoint of the vector is converted from the local coordinate system to the coordinate system of parametric domain by the linear transformation $[\vec{u}, \vec{v}]'$. The point, which is converted into parametric plane, corresponds to the mesh point at the surface. Meanwhile this point is served as the given point for a following calculation of mesh point. By the iteration of the method, mesh points are fabricated step by step and the whole surface is "glued" automatically.

As an example, control points (colorful points) are randomized given (see Figure 3). Meanwhile the degrees of two direction are set to 4, since the calculation of the curvature requires a high continuous parametrization. As a balance of the continuity and the calculated amount, degree 3 for continuity at each knot is a compromise. The continuity of the surface along a particular parametric line can be changed by knot insertion later.

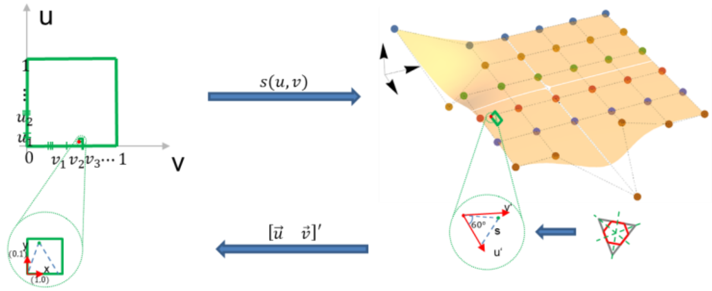

Figure 2: Meshing method

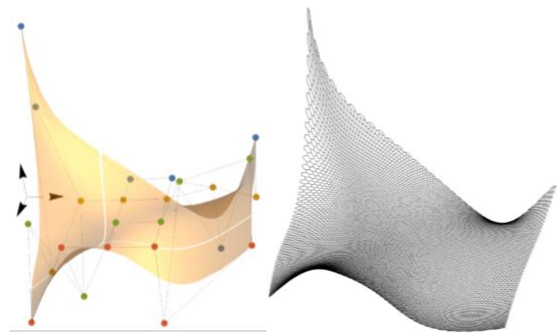

Figure 3: Left: Arbitrary spatial surface in 3D space; right: mesh model of the surface

3 SELF ORGANIZING MAPS

Self-organizing maps (SOM) is an unsupervised learning method (Kohonen 2001), which compresses the complex high-dimensional information into a simple low-dimensional pattern in 2D or 3D. Meanwhile it preserves the most important topological and/or metric relationships of the primary data elements. The aspects of visualization and abstraction, can be used to diverse fields for process analysis, dimensionality reduction, data visualization or clustering, and others. The SOM network typically consists of two layers of nodes, input layer and output layer. The nodes in input layer are directly connected to the nodes in output layer by weight ωi,. Assuming spatial coordinates and a d-dimension (d equals to the dimension of the input data) vector of weights are given to every node in initial self-organizing maps. Then: 1) at each time step t of the training, a random input vector of data is chosen, and the node having the closest (norm L2) feature vector (yellow disk) to that input vector, called the Best Matching Unit (BMU), is found; 2) the BMU is going to be updated. That means the feature vector Vt of BMU is updated so that it (Dt) is closer to the imputed data at time step t, $Vt+1 = Vt + L(t) \times (Dt - Vt)$ in which $L(t)$ is the learning rate; 3) the weight vectors Wt of neighborhoods of the BMU are updated depending on the coefficient$N(\delta, t)$ with regard to the Euclidean distance δ between the neighborhood and the BMU with $Wt+1 = Wt + N(\delta, t) \times L(t) \times (Dt - Wt)$; 4) steps 1) to 3) will be repeated until the change of SOM between time step t and $t+1$ is tiny enough. SOM provides a method that a free-formed mesh with inherent properties on each node can organize themselves under some conditions, and after a period of iteration return to a stable form. This means, that we can use SOM to simulate the self-organizing process based on mesh generated NURBS that can be the initial self-organizing map with spatial coordinates; the relationship of the mesh points (such as deformation, stress, elasticity, potential energy etc.) which can be the weight vector of the SOM node; the potential fields which can be the input data; and the total potential energy of the whole system under test which can be the criterion for updating the weight vector. Additionally, the constraints of the mesh should also be considered, like the displacement of the boundaries, the deformation, transportation etc. Different from the SOM, the choice of BMU should be started from these constraints, and iterated from the constraints to non-constraints points.

4 OUTLOOK

We previously described a methodological interpolation approach of polynomials as basis for the next step studying self-organized potential based shape determination. The most influencing aspects of a self-organized process can be identified as characteristic properties of atoms or molecules, determining boundary conditions at which the process take place. To achieve self-organization of mesh generation, questions must be answered in self-organizing multifunctional structures for adaptive high performance light-weight construction like automatic identification or clustering of singularities (Polthier and Preuss 2003), or design of path parameterization so that the mesh is seamless (Erickson and Whittlesey 2005).

ACKNOWLEDMENT
The authors appreciate the support by the Lower Saxony Ministry of Science and Culture (MWK) within the PhD program "Self-organizing multifunctional structures for adaptive high performance light-weight constructions".

REFERENCES
Bézier, P.: The Mathematical Basis of UNISURF CAD System. Butterworths Publ., 1986

Ciarlet, P., G., Raviart, P.A.: General Lagrange and Hermite Interpolation in R^N with Applications to Finite Element Methods. Springer Publ. 1972

Erickson, J., Whittlesey, K.: Greedy optimal homotopic and homology generators. In: Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2005

Glassner, A.: Principles of Digital Image Synthesis and 3D Computer Graphics: A Handbook for Artists and Designers. Academic Press, 1993

Körber, C., Möller, D. P. F.: Dynamic Depth Triangulation of Large NURBS Surfaces in Real-Time and its Application to Geoscience. In: Proceed. 4th Mathmod Conf., pp. 618-622. Eds.: I. Troch, F. Breitenecker, ARGESIM Report, Vol. 24, 2003

Piegl, L., Tiller, W.: The NURBS Book. Springer Publ. 1997

Prautzsch, H.; Boehm, W.; Paluszny, M.: Bézier and B-Spline Techniques. Springer Publ. 2002

Pressely, A.- N.: Elementary Differential Geometry. Springer Publ.. 2010

Respondek, J. S.: Recursive Numerical Recipes for the High Efficient Inversion of the Confluent Vandermonde Matrices. J. Appl. Math. And Comp. Vol. 255, pp. 718-730, 2013

Respondek, J. S.: Incremental Numerical Recipes for the High Efficient Inversion of the Confluent Vandermonde Matrices. In Computers and Mathematics, Vol. 71, pp. 489-502, 2016

Rogers, D. F.: Introduction to NURBS. Morgan Kaufmann Publ. 2001

Schoenberg, I. J.: On Spline Functions. In: Sischa, O. (Ed.) In-Equalities, pp.255-291, Acad. Press 1967

Kohonen, T.: Self-Organizing Maps. Springer Publ. 2001

Woodward, C. D.: Blends in Geometric Modeling. In. Martin, R. R. (Ed.) Mathematical Methods of Surfaces II, pp. 255-297, Oxford University Press, 1987

AUTHOR BIOGRAPHIES
Mr. Xin Li is a PhD Student at TU Clausthal in the PhD program "Self-organizing multifunctional structures for adaptive high performance light-weight constructions". His email address is: xin.li@tu-clausthal.de.

Prof. Dr. Dietmar P. F. Möller is a supervisor in the PhD program "Self-organizing multifunctional structures for adaptive high performance light-weight constructions". His email address is: dietmar.moeller@tu-clausthal.de.