# Least Squares Approximation by B-spline Curves and Surfaces

Yong Ming Li

March, 1997

## 1 Approximation by B-spline Curve

### 1.1 Mathematics

Given a set of points $\mathbf{q}_l$ $(l = 0, 1, \cdots, L)$ with associated parameter values $\bar{t}_l$, we want to construct a B-spline curve of degree $n$ (or order $k$) with $M + 1$ control points $\mathbf{p}_i$, i.e.,

$$\mathbf{r}(t) = \sum_{i=0}^{M} N_{i,k}(t)\mathbf{p}_i$$

such that

- $\mathbf{r}(t_0) = \mathbf{q}_0$ and $\mathbf{r}(t_M) = \mathbf{q}_L$,
- Remaining $\mathbf{q}_l$ are approximated in the least squares sense, i.e.,

$$\phi(\mathbf{P}) = \sum_{l=1}^{L-1}(\mathbf{r}(\bar{t}_l) - \mathbf{q}_l)^2 \rightarrow \min,$$

  where $\mathbf{P}$ denotes a collection of $M + 1$ control points $\mathbf{p}_i$.

If the multiplicity of knots at both end points is equal to $k$, then $\mathbf{r}(t_0) = \mathbf{p}_0$ and $\mathbf{r}(t_M) = \mathbf{p}_M$. In this case, we have $\mathbf{p}_0 = \mathbf{q}_0$ and $\mathbf{p}_M = \mathbf{q}_L$. Let

$$\mathbf{Q}_l = \mathbf{q}_l - N_{0,k}(\bar{t}_l)\mathbf{q}_0 - N_{M,k}(\bar{t}_l)\mathbf{q}_L.$$

We can then write $\phi(\mathbf{P})$ in a simpler form as

$$\phi(\mathbf{P}) = \sum_{l=1}^{L-1}\left(\sum_{i=1}^{M-1} N_{i,k}(\bar{t}_l)\mathbf{p}_i - \mathbf{Q}_l\right)^2. \tag{1.1}$$

Differentiating $\phi$ with respect to $\mathbf{p}_j$ gives

$$\frac{\delta\phi(\mathbf{P})}{\delta\mathbf{p}_j} = 2\sum_{l=1}^{L-1}\left(\sum_{i=1}^{M-1} N_{i,k}(\bar{t}_l)\mathbf{p}_i - \mathbf{Q}_l\right)N_{j,k}(\bar{t}_l).$$

Accordingly, the relation $\phi(\mathbf{P}) \to \min$ is equivalent to solving the following system of $M-1$ linear equations:

$$\sum_{l=1}^{L-1}\left(\sum_{i=1}^{M-1} N_{i,k}(\bar{t}_l)\mathbf{p}_i - \mathbf{Q}_l\right) N_{1,k}(\bar{t}_l) = 0$$

$$\sum_{l=1}^{L-1}\left(\sum_{i=1}^{M-1} N_{i,k}(\bar{t}_l)\mathbf{p}_i - \mathbf{Q}_l\right) N_{2,k}(\bar{t}_l) = 0$$

$$\vdots$$

$$\sum_{l=1}^{L-1}\left(\sum_{i=1}^{M-1} N_{i,k}(\bar{t}_l)\mathbf{p}_i - \mathbf{Q}_l\right) N_{M-1,k}(\bar{t}_l) = 0$$

The above system of linear equations may also be written as

$$\sum_{i=1}^{M-1} \mathbf{p}_i \left(\sum_{l=1}^{L-1} N_{i,k}(\bar{t}_l)N_{j,k}(\bar{t}_l)\right) = \sum_{l=1}^{L-1} N_{j,k}(\bar{t}_l)\mathbf{Q}_l \quad (j = 1, 2, \cdots, M-1)$$

Let $\mathbf{N}$ be the following $(L-1) \times (M-1)$ matrix

$$\mathbf{N} = \begin{pmatrix} N_{1,k}(\bar{t}_1) & \cdots & N_{M-1,k}(\bar{t}_1) \\ \vdots & \vdots & \vdots \\ N_{1,k}(\bar{t}_{L-1}) & \cdots & N_{M-1,k}(\bar{t}_{L-1}) \end{pmatrix}.$$

Then, the system of linear equations is given by

$$(\mathbf{N}^T\mathbf{N})\mathbf{P} = \mathbf{R} \tag{1.2}$$

where, $\mathbf{P} = \begin{pmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_M \end{pmatrix}^T$ and $\mathbf{R} = N^T\mathbf{Q}$, i.e.,

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{M-1} \end{pmatrix} = \begin{pmatrix} N_{1,k}(\bar{t}_1)\mathbf{Q}_1 + \cdots + N_{1,k}(\bar{t}_{L-1})\mathbf{Q}_{L-1} \\ \vdots \\ N_{M-1,k}(\bar{t}_1)\mathbf{Q}_1 + \cdots + N_{M-1,k}(\bar{t}_{L-1})\mathbf{Q}_{L-1} \end{pmatrix}$$

There are still some issues that need to be answered before we can solve linear system (1.2), namely

- Determining parametrization of $\bar{t}_l$ which reflects the distribution of $\mathbf{Q}_l$.
- Determining parametrization of $t$. Ideally, the knot sequence should be chosen such that there is at least one data point within the span $[t_i, t_{i+1}]$. Accordingly, $(N^TN)$ would be a positive definite matrix that can be solved efficiently without the use Singular Value Decomposition (SVD) method. One of the well-known methods to solve a linear system with positive definite matrix is the so-called Cholesky decomposition.

We shall consider these aspects in the subsequent sections.

## 1.2 Curve approximation based on ordered pints

If the given points $\mathbf{q}_l$ are ordered, the algorithm to create a least squares approximation B-spline curve is simple. In this section, we shall discuss the choice of associated parameters $\bar{t}_l$, the knot sequence, computation of $\mathbf{N}$ matrix, and multiplication of matrix. These are the required conditions to setup a linear system to determine the desired control vertices of a B-spline curve.

## 1.3 Choice of parametrization

It is known that a least square approximation requires knowledge of $\bar{t}_l$ (the knots associated with $\mathbf{q}_l$) and $t_i$ (the knot sequence of a B-spline curve). Therefore, we need to discuss the choices of parametrization for both $\bar{t}_l$ and $t_i$.

We first consider parametrizing the given set of points $\mathbf{q}_l$, i.e., the choice of parametrization for $\mathbf{q}_l$.

**Method 1** : equally spaced parametrization, i.e.,

$$\bar{t}_l = \frac{l}{M} \qquad (l = 0, 1, \cdots, M).$$

This method is simple however may produce unwanted shapes (e.g., loops) when the data is unevenly spaced.

**Method 2** : chordal parametrization. Let $d = \sum_{l=1}^{M} \|\mathbf{q}_l - \mathbf{q}_{l-1}\|$ and $\bar{t}_0 = 0$. Then, we have

$$\bar{t}_l = \bar{t}_{l-1} + \frac{\|\mathbf{q}_l - \mathbf{q}_{l-1}\|}{d}, \quad (l = 1, 2, \cdots, M)$$

This is one of the most widely used methods since it gives an approximation to the arc length parametrization.

**Method 3** : centripetal parametrization. Let $\bar{t}_l = 0$ and $d = \sum_{l=1}^{M} \sqrt{\|\mathbf{q}_l - \mathbf{q}_{l-1}\|}$. Then,

$$\bar{t}_l = \bar{t}_{l-1} + \frac{\sqrt{\|\mathbf{q}_l - \mathbf{q}_{l-1}\|}}{d}, \quad (l = 1, 2, \cdots, M)$$

This is a relatively new method [Lee'1989] that gives better results than the chordal parametrization when the data takes very sharp turns.

There are other parametrization methods as well [Hoschek and Lasser'1993]. With all possible parametrization methods, we prefer to using the centripetal parametrization.

We now consider the choice of parametrization of $t$. It is noted that $\mathbf{r}(t) = \sum_{i=0}^{M} N_{i,k}(t)\mathbf{p}_i$ is defined over the knot sequence $\{t_i\}_{i=0}^{K}$, where $K = M + k$. A simple parametrization of $t$ is again the equally spaced parametrization. Since there are $M - n$ internal knots (or $M - n + 1$ internal knot spans), the equally spaced parametrization is given by

$$t_0 = t_1 = \cdots = t_n = 0, \quad t_{K-n} = t_{K-n+1} = \cdots = t_K = 1$$

$$t_{n+i} = \frac{i}{M - n + 1} \quad (i = 1, 2, \cdots, M - n)$$

Such simple parametrisation did not take the distribution of the $\{\bar{t}_l\}$ into consideration. A better parametrisation was suggested by de Boor [de Boor'1978]. Let

$$d = \frac{L + 1}{M - n + 1}.$$

Then, the internal knots are defined by

$$i = \mathrm{int}(jd), \qquad \alpha = jd - i \quad (j = 1, 2, \cdots, M - n)$$

$$t_{n+j} = (1 - \alpha)\bar{t}_{i-1} + \alpha \bar{t}_i.$$

If $d > 2.0$, then this parametrisation guarantees that every knot span contains at least one $\bar{t}_l$ and, under this condition, the matrix $(N^T N)$ in (3) is positive definite and well-conditioned. Therefore, the system of linear equations can be solved by Gaussian elimination without pivoting.

### 1.4 Curve approximation based on random points

Since $\mathbf{q}_l$ are randomly sampled, the methods of computing associated parameters $\bar{t}_l$ and knot sequence discussed in the previous sections cannot be applied here. If we already have a B-spline curve and want to refine it to get better approximation to the given points, then the associated parameters can be computed by calling minimum distance routine. If we do not have any B-spline curve to start with, we may compute the least squares line (or a cubic Bezier curve) as a seed to approximate these discrete points. By representing the seed curve in B-spline form and adding enough knots into the curve to gain flexibility, we can refine the curve such that it approximate the curve within the specified tolerance.

It should be pointed out that solutions of 1.1 or 1.2 minimizes the distance between $\mathbf{r}(\bar{t}_l)$ and $\mathbf{q}_l$, which does not necessarily give the shortest distance to the resulting curve. This is because all the associated parameters $\bar{t}_l$ were obtained before refining operation is down. Accordingly, $\mathbf{q}_l$ is not orthogonal to the resulting curve $\mathbf{r}(\bar{t}_l)$. To obtain an optimized result, we may repeat the refining process for a few times to ensure $\mathbf{q}_l$ is almost orthogonal to the final curve $\mathbf{r}(\bar{t}_l)$. Such process is known as the parametrization correction.

## 2 Approximation by B-spline Surface

### 2.1 Mathematics

Given a set of points $\mathbf{q}_l$ and associated knots $(u_l, v_l)$ with $l = 0, 1, \cdots, L$, we want to construct a non-rational B-spline surface

$$\mathbf{S}(u,v) = \sum_{i=0}^{N}\sum_{j=0}^{M} N_{i,k_u}(u)N_{j,k_v}(v)\mathbf{p}_{i,j}$$

that fits $\mathbf{q}_l$ using a least squares approximation method, i.e.,

$$\phi(\mathbf{P}) = \sum_{l=0}^{L}\left(\sum_{i=0}^{N}\sum_{j=0}^{M} N_{i,k_u}(u_l)N_{j,k_v}(v_l)\mathbf{p}_{i,j} - \mathbf{q}_l\right)^2 \to \min, \tag{2.1}$$

where $\mathbf{P}$ is a collection of $(N+1)\times(M+1)$ control points $\mathbf{p}_{i,j}$. Mathematically, this is equivalent to solving the following $(N+1)\times(M+1)$ linear equations:

$$\sum_{l=0}^{L}\left(\sum_{i=0}^{N}\sum_{j=0}^{M} N_{i,k_u}(u_l)N_{j,k_v}(v_l)\mathbf{p}_{i,j} - \mathbf{q}_l\right) N_{r,k_u}(u_l)N_{s,k_v}(v_l) = 0 \quad (r = 0,\cdots,N; s = 0,\cdots,M).$$

Alternatively, we can write the above system of linear equations as follows:

$$\sum_{l=0}^{L}\sum_{i=0}^{N}\sum_{j=0}^{M} N_{i,k_u}(u_l)N_{j,k_v}(v_l)N_{r,k_u}(u_l)N_{s,k_v}(v_l)\mathbf{p}_{i,j} = \sum_{l=0}^{L} N_{r,k_u}(u_l)N_{s,k_v}(v_l)\mathbf{q}_l \tag{2.2}$$

for $r = 0, \cdots, N; s = 0, \cdots, M$.

As it is seen, the above system involves four matrices manipulation and multiplication. It is neither trivial nor efficient to setup the final matrix based on the above representation. In the subsequent sections we shall derive two simpler methods to solve the approximation issue, depending on how the points $\mathbf{q}_l$ are sampled.

## 2.2 Surface approximation based on grid points

Given a set of grid points $\mathbf{Q}_{p,q}$ and associated knots $(\bar{u}_p, \bar{v}_q)$ with $p = 0, 1, \cdots, r$ and $q = 0, 1, \cdots, s$, we want to construct a non-rational B-spline surface

$$\mathbf{S}(u, v) = \sum_{i=0}^{N} \sum_{j=0}^{M} N_{i,k_u}(u) N_{j,k_v}(v) \mathbf{P}_{i,j}$$

that fits $\mathbf{Q}_{p,q}$ using a least square surface approximation method. A classical solution to this problem is to employ the so called discrete least square method of Gauss. In this section, however, we present a much simpler least square surface approximation method that relys upon only the least square curve approximation algorithm and thus can make the best use of curve routines.

We start our discussion by considering an interpolation problem. A B-spline surface that interpolates the given set of points $\mathbf{Q}_{p,q}$ at $(\bar{u}_p, \bar{v}_q)$ may be represented as

$$\mathbf{S}(\bar{u}, v) = \sum_{i=0}^{s} \sum_{j=0}^{r} N_{i,k_u}(u) N_{j,k_v}(v) \bar{\mathbf{P}}_{i,j},$$

which satisfies the property $\mathbf{Q}_{p,q} = \bar{\mathbf{S}}(\bar{u}_p, \bar{v}_q)$. Alternatively, we may write the above B-spline surface as

$$\bar{\mathbf{S}}(u, v) = \sum_{j=0}^{s} N_{j,k_v}(v) \bar{\mathbf{P}}_j(u) \tag{2.3}$$

where

$$\bar{\mathbf{P}}_j(u) = \sum_{i=0}^{r} N_{i,k_u}(u) \bar{\mathbf{P}}_{i,j}$$

are B-spline curves that interpolate $\mathbf{Q}_{p,j}$ at $u = \bar{u}_p$. Instead of interpolating $\mathbf{Q}_{p,j}$ by $\bar{\mathbf{P}}_j(u)$, we may use a least square curve approximation method to construct

$$\mathbf{P}_j(u) = \sum_{i=0}^{N} N_{i,k_u}(u) \check{\mathbf{P}}_{i,j}$$

that fit $\mathbf{Q}_{p,j}$ $(p = 0, 1, \cdots, r)$. Replacing $\bar{\mathbf{P}}_j(u)$ in 2.3 by $\mathbf{P}_j(u)$ gives

$$\bar{\mathbf{S}}(u, v) \approx \sum_{j=0}^{s} N_{j,k_v}(v) \mathbf{P}_j(u) = \sum_{i=0}^{N} N_{i,k_u}(u) \left[ \sum_{j=0}^{s} N_{j,k_v}(v) \check{\mathbf{P}}_{i,j} \right].$$

Let $\check{\mathbf{P}}_i(v) = \sum_{j=0}^{s} N_{j,k_v}(v) \check{\mathbf{P}}_{i,j}$. Then,

$$\bar{\mathbf{S}}(u, v) \approx \sum_{i=0}^{N} N_{i,k_u}(u) \check{\mathbf{P}}_i(v). \tag{2.4}$$

Analogously, we may use a least square curve approximation method to construct

$$\mathbf{P}_i(v) = \sum_{j=0}^{M} N_{j,k_v}(v) \mathbf{P}_{i,j}$$

which fit $\check{\mathbf{P}}_{i,j}$ $(j = 0, 1, \cdots, s)$. Replacing $\check{\mathbf{P}}_i(v)$ in 2.4 by $\mathbf{P}_i(v)$ gives

$$\bar{\mathbf{S}}(u, v) \approx \sum_{i=0}^{N} N_{i,k_u}(u) \sum_{j=0}^{M} N_{j,k_v}(v)\mathbf{P}_{i,j} = \sum_{i=0}^{N}\sum_{j=0}^{M} N_{i,k_u}(u)N_{j,k_v}(v)\mathbf{P}_{i,j} = \mathbf{S}(u, v).$$

As it is seen, we fit across the data points first in $u$ direction, then $v$ direction. One may, of course, fit across the data points first in $v$, then $u$ direction. In general, the resulting surfaces are not the same. To our knowledge, there is no criterion to decide in advance which approach will yield a better solution.

Finally, we should point out that the construction of all $\mathbf{P}_j(u)$ uses the same $u$ knot sequence. Therefore, we need to compute only once the coefficient matrices $\mathbf{N}$ and $\mathbf{N}^T\mathbf{N}$ (see equation 1.2) for constructing $\mathbf{P}_j(u)$. Furthermore, the LU decomposition of $\mathbf{N}^T\mathbf{N}$ is only done once for the $u$ direction fit. A similar rule applies to the $v$ direction fit as well.

## 2.3 Surface approximation based on random points

Since $\mathbf{q}_l$ are randomly sampled, the method discussed in the previous section cannot be applied here. Instead, we have to solve the so-called least squares approximation to discrete points. To avoid manipulating and multiplying four matrices in (2.2), we need to derive a simpler representation. For easy understanding, we start by writing a bi-quadratic B-spline surface explicitly as follows:

$$\begin{aligned}\mathbf{S}(u, v) &= [\mathbf{p}_{00}N_0(u) + \mathbf{p}_{10}N_1(u) + \mathbf{p}_{20}N_2(u)]N_0(v)\\ &+ [\mathbf{p}_{01}N_0(u) + \mathbf{p}_{11}N_1(u) + \mathbf{p}_{21}N_2(u)]N_1(v)\\ &+ [\mathbf{p}_{02}N_0(u) + \mathbf{p}_{12}N_1(u) + \mathbf{p}_{22}N_2(u)]N_2(v).\end{aligned}$$

If we store our control points in one-dimensional array as $\mathbf{p}_{00}, \mathbf{p}_{10}, \mathbf{p}_{20}, \mathbf{p}_{01}, \mathbf{p}_{11}, \cdots, \mathbf{p}_{22}$ with index ranges from $k = 0$ to $k = 8$, we can then write the surface as

$$\mathbf{S}(u, v) = \sum_{k=0}^{8} \mathbf{p}_k B_k(u, v),$$

where $B_{j\times 3+i} = N_i(u)N_j(v)$. It is noted that a B-spline surface is represented similarly as a B-spline curve. We now expand the bi-quadratic B-spline surface to a generic B-spline surface that has $(N + 1) \times (M + 1)$ control points. Denoting $K = (N + 1) \times (M + 1) - 1$ we have

$$\mathbf{S}(u, v) = \sum_{k=0}^{K} \mathbf{p}_k B_k(u, v),$$

where $B_{j\times(N+1)+i} = N_i(u)N_j(v)$. Accordingly, the least squares problem shown in (2.1) becomes

$$\phi(\mathbf{P}) = \sum_{l=0}^{L}\left(\sum_{k=0}^{K} \mathbf{p}_k B_k(u_l, v_l) - \mathbf{q}_l\right)^2 \rightarrow min. \tag{2.5}$$

The above equation is similar to equation(1.1). Therefore, the control points of B-spline surface can be computed similarly as we did for a curve.

Let $\mathbf{B}$ be the following $(L+1) \times (K+1)$ matrix

$$\mathbf{B} = \begin{pmatrix} B_0(u_0, v_0) & B_1(u_0, v_0) & \cdots & B_K(u_0, v_0) \\ B_0(u_1, v_1) & B_1(u_1, v_1) & \cdots & B_K(u_1, v_1) \\ \vdots & \vdots & \vdots & \\ B_0(u_L, v_L) & B_1(u_L, v_L) & \cdots & B_K(u_L, v_L) \end{pmatrix}.$$

Then, the $(K+1) = (N+1) \times (M+1)$ control vertices $\mathbf{p}_k$ can be determined by solving the following $(K+1) \times (K+1)$ linear system

$$(\mathbf{B}^T\mathbf{B})\mathbf{P} = \mathbf{Q}. \tag{2.6}$$

We now discuss how to determine $(u_l, v_l)$ associated with sample points $\mathbf{q}_l$. In some applications, we already have a B-spline surface (the seed surface) that approximate the given $(L+1)$ sampling points but want to refine the surface to get a better approximation. In this case, the $(u_l, v_l)$ can be obtained by calling a minimum distance routine to get the minimum distance point on the surface and associated parameters. For other applications, we may not have any seed surface. In this case, we can start with the least squares plane and represent it in a B-spline form. We then elevate the degrees of the surface in $u$ and $v$ to obtain some flexibility. In order to avoid oscillations commonly seen in high degree B-spline surface, it is recommended to limit the surface to cubic. By solving (2.6), we should get a better approximation to the given points. If the accuracy does not meet the specified tolerance, we will add more knots into the $u$ and $v$ knots sequence to obtain more flexibility to manipulate the shape of the surface. By repeating the process, we may eventually reach the accuracy requirement. It should be pointed, however, that we do not want to repeat the process too many times for the following reasons:

- It takes significant time to compute minimum distance parameters, $\mathbf{B}$ matrix, and large scale linear system when the surface has considerable number of control points. Assume that we are given 5,000 sampling points $\mathbf{q}_l$ to refine a surface that has 100 control vertices (a very moderate surface). We first need to compute 5,000 $(u_l, v_l)$ parameters based on a call to minimum distance routine. We then compute $\mathbf{B}_{5{,}000\times 100}$ matrix. Next, we need to solve a linear system whose matrix is $100 \times 100$.

- To ensure the system is positive definite, we need to have at least one sampling point at each knot interval. With the increase of control vertices, we may encounter a singular case. Although singular system can be solved using SVD method, it is not desirable in practice because of performance and oscillation considerations. Some researchers suggest to introduce energy minimization constraints into the system to ensure it is positive definite. However, it degrades the approximation accuracy in general.

It should be pointed out that solutions of 2.5 or 2.6 minimizes the distance between $\mathbf{S}(u_l, v_l)$ and $\mathbf{q}_l$, which does not necessarily give the shortest distance to the resulting surface. This is because all the associated parameters $(u_l, v_l)$ were obtained before refining operation is down. Accordingly, $\mathbf{q}_l$ is not orthogonal to the resulting surface $\mathbf{S}(u_l, v_l)$. To obtain an optimized result, we may repeat the refining process for a few times to ensure $\mathbf{q}_l$ is almost orthogonal to the final surface $\mathbf{S}(u_l, v_l)$. Such process is known as the parametrization correction.

# 3 Applications

In this section we discuss how the least squares approximation method can be used to compute a single B-spline surface that approximates a plate consisting of multiple B-spline surfaces.

Referring to Figure 1, this plate has six trimmed B-spline surfaces. For certain applications users may wish to approximate these surfaces by a single B-spline surface to simplify downstream operations. Since these surfaces are usually trimmed, we may not be able to merge them. In this case, we can use the least squares approximation method to derive a single B-spline surface that approximates the given surfaces with respect to the specified tolerance.

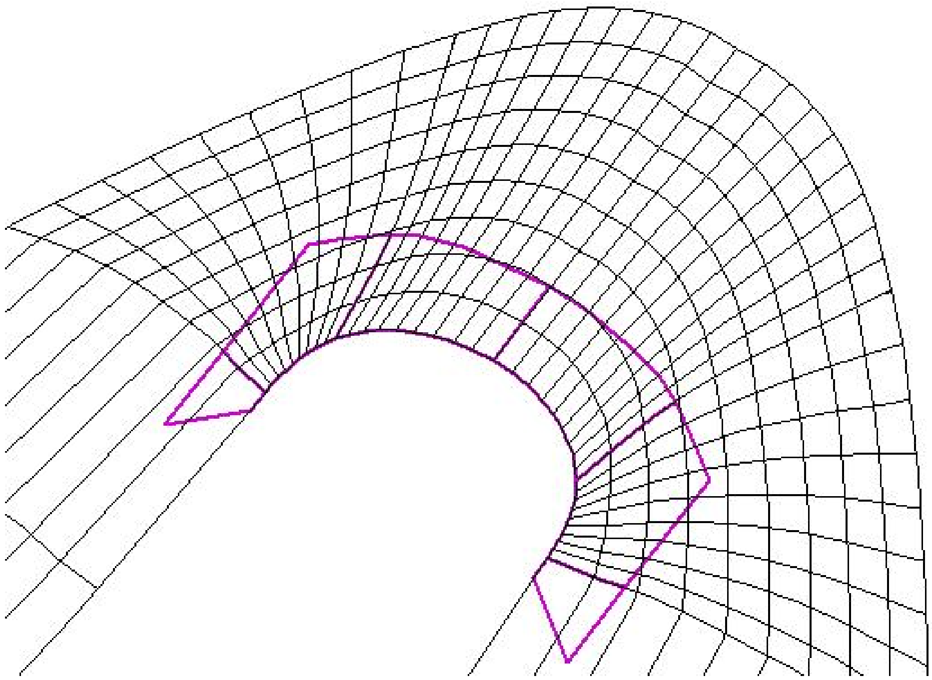

Figure 1: A plate consists of 6 B-spline surfaces

The procedure is outlined below:

1. Sample enough points from each trimmed surfaces.
2. Compute the least squares plane and represent it in B-spline form.
3. Elevate the degree of the least squares plane surface to obtain flexibility. We recommend to elevate the plane to a bi-cubic surface.
4. Refine the surface by solving equation 2.6. If approximation accuracy is within the specified tolerance, terminate. Otherwise, go to next step.
5. Add more knots into the $u$ and $v$ knots sequences of the resulting surface and go to step 4.

It is noted that the plate has a horse-shoe like shape; while the resulting surface has usually rectangular shape. This means that we will have large amount of points missing at the crescent part. As a result, we may either see oscillations in the final surface or encounter a singular system. Since the first refined surface is a bi-cubic surface, it is unlikely to have severe oscillations. To minimize the oscillations and avoid solving SVD in repeated processes, we can sample sufficient points outside the boundary from the first refined surface and add them into our collections of sampling points. These additional points will result in an acceptable surface inside and outside the boundary as shown in Figure 2.

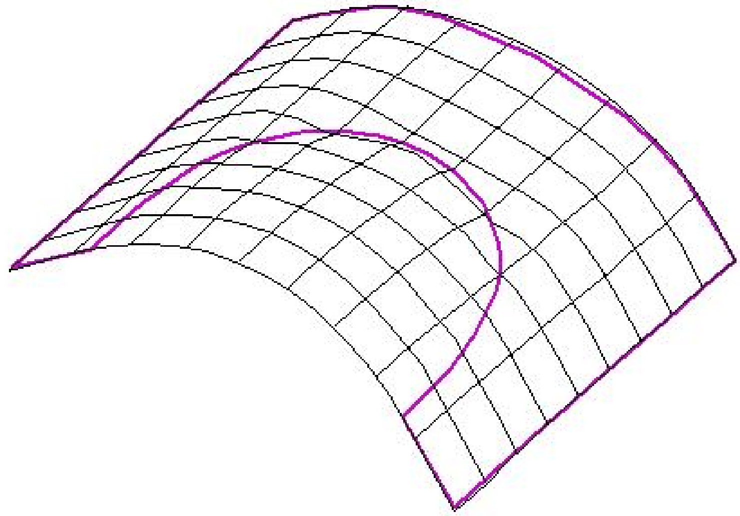

Figure 2: The "meregd" B-spline surface

# References

[1] De Boor., C., *A Practical Guide to Splines* (New York, Springer-Verlag, 1978).

[2] Hoschek, J. and Lasser, D., *Computer Aided Geometric Design* (A K Peter, Ltd., 1993).

[3] Piegl, L. and Tiller, W.. *The NURBS Book, 2nd Ed.* (Springer, 1997).