B-spline Surfaces

A B-spline surface is taking a bi-directional net of control points, two knot vectors, and products of the univariate B-spline basis functions:

\[ S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{ij} \]

with
\[
U = \{0, \ldots, 0, u_{p+1}, \ldots, u_{r-p-1}, 1, \ldots, 1\}_{p+1}
\]

\[
V = \{0, \ldots, 0, v_{q+1}, \ldots, v_{s-q-1}, 1, \ldots, 1\}_{q+1}
\]
\( U \) has \( r + 1 \) knots, \( V \) has \( s + 1 \), so the following relationships hold:

\[
  r = n + p + 1 \quad \text{and} \quad s = m + q + 1
\]

Five steps are required to compute a point on a B-spline surface at fixed \((u, v)\) parameter values:

1. Find the knot span in which \( u \) lies, say \( u \in [u_i, u_{i+1}) \) (Alg. A2.1).
2. Compute the nonzero basis functions $N_{i-p,p}(u)$, ..., $N_{i,p}(u)$ (A2.2).

3. Find the knot span in which $v$ lies, say $v \in [v_j, v_{j+1})$ (Alg. A2.1).

4. Compute the nonzero basis functions $N_{j-q,q}(u)$, ..., $N_{j,q}(u)$ (A2.2).

5. Multiply the values of the nonzero basis functions with the corresponding control points.
The last step takes the form:

\[ S(u, v) = [N_{k,p}(u)]^T [P_{k,l}] [N_{l,q}(v)] \]

with,

\[ i - p \leq k \leq i \quad \text{and} \quad j - q \leq l \leq j \]
Example:

Let $p = q = 2$ and

$$
\sum_{i=0}^{4} \sum_{j=0}^{5} N_{i,2}(u) N_{j,2}(v) P_{ij} \quad \text{with}
$$

$U = \{0, 0, 0, 0.4, 0.6, 1, 1, 1\}$ and $V = \{0, 0, 0, 0.2, 0.5, 0.8, 1, 1, 1\}$. Suppose we want to compute $S(0.2, 0.6)$. Then $0.2 \in [u_2, u_3)$ and $0.6 \in [v_4, v_5)$, and,
\[ S(0.2, 0.6) = \]
\[ = \begin{bmatrix}
N_{0,2}(0.2) & N_{1,2}(0.2) & N_{1,3}(0.2)
\end{bmatrix}
\]
\[ \cdot \begin{bmatrix}
P_{02} & P_{03} & P_{04} \\
P_{12} & P_{13} & P_{14} \\
P_{22} & P_{23} & P_{24}
\end{bmatrix}
\begin{bmatrix}
N_{2,2}(0.6) \\
N_{3,2}(0.6) \\
N_{4,2}(0.6)
\end{bmatrix} \]
See algorithm A3.5

For efficiency, Algorithm A3.5 uses a local array, `temp[]`, to store the intermediate vector/matrix product.
The properties of the tensor product basis functions follow from those from the corresponding univariate basis functions, i.e., non-negativity, partition of unity, local influence, Bernstein basis as a special case, one maximum, differentiability.

From these, properties of B-spline surfaces follow, i.e., interpolation of corner points, affine invariance, convex hull property, locality, continuity, and differentiability.
Note that there is no variation diminishing property for B-spline surfaces!

In addition, iso-parametric curves on $S(u, v)$ are obtained in a manner analogous to that for Bezier surfaces.
Derivatives of a B-spline Surface

Generally, we are interested in computing all partial derivatives of \( S(u, v) \) up to and including order \( d \):

\[
\frac{\partial^{k+l}}{\partial^k u \partial^l v} S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_i^{(k)}(u) N_j^{(l)}(v) P_{ij}
\]

with \( 0 \leq k + l \leq d \).
Algorithm A3.6 computes the point on a B-spline surface and all partial derivatives up to and including order $d$ ($d > p, q$ is allowed). Like algorithm A3.5, this is a five step process, with the last step being vector/matrix/vector multiplications of the form:
\[
\frac{\partial^{k+l}}{\partial^k u \partial^l v} S(u, v) = \\
= \left[ N^{(k)}_{k, p}(u) \right]^T \left[ P_{k, l} \right] \left[ N^{(l)}_{l, q}(v) \right] \\
\text{with,} \\
0 \leq k + l \leq d \\
uspan - p \leq r \leq uspan \quad \text{and} \\
vspan - q \leq s \leq vspan
\]
Output is the array, $\text{SKL[]}[]$, where $\text{SKL}[k][l]$ is the derivative of $S(u, v)$ with respect to $u$ $k$-times, and $v$ $l$-times.

Arrays $\text{Nu[]}[]$ and $\text{Nv[]}[]$ are used to store the derivatives of the basis functions.