de Casteljau

We can apply the de Casteljau algorithm on surfaces to compute a point on the surface. Suppose that we have a control net with \(m + 1 \times n + 1\) array of control points. We wish to compute the point \(p(\bar{u}, \bar{w})\).

The algorithm first applies the de Casteljau algorithm to the control points in each row \(p_{i0}, \ldots, p_{in}\). For each such row it computes the point \(p_is\). One can observe that \(p_is\) is generated on the Bézier curve generated with the control points \(p_{i0}, \ldots, p_{in}\).

The definition of \(p_is\) is given as follows:

For every \(i\) with \(0 \leq i \leq m\), we first compute the points \(p_{is,j,k}\), where \(p_{is,j} = p_{ij}\) and

\[
p_{is,j,k}^j = (1 - \bar{u})p_{is,k}^{j-1} + \bar{u}p_{is,k+1}^{j-1}
\]

with \(1 \leq j \leq n\) and \(0 \leq k \leq n - j\), and we let \(p_is = p_{is,0}^n\).

Next we apply de Casteljau algorithm on the points \(p_{0s}, \ldots, p_{ms}\). For this we compute the points \(p_{is}^j\), where \(p_{is}^0 = p_{is}\), and

\[
p_{is}^j = (1 - \bar{w})p_{is}^{j-1} + \bar{w}p_{is+1}^{j-1}
\]

with \(1 \leq j \leq m\) and \(0 \leq i \leq m - j\).

Finally, we will have \(p(\bar{u}, \bar{w}) = p_{0s}^m\).

Alternatively, we could have computed \(p_{s0}, \ldots, p_{sn}\), first where \(p_{sj}\) is obtained by de Casteljau algorithm on the control points \(p_{0j}, \ldots, p_{mj}\) and then compute \(p_{0s}^m\) by applying the de Casteljau to the control points \(p_{s0}, \ldots, p_{sn}\). We will have \(p_{0s}^m = p_{s0}\).

\[
\text{CasteljauSurf}(P) \\
\text{for } i := 0 \text{ to } m \text{ do} \\
\quad \text{for } j := 0 \text{ to } n \text{ do} \\
\quad\quad p_{is,j}^0 := p_{ij} \\
\quad\quad \text{endfor} \\
\quad \text{for } j := 1 \text{ to } n \text{ do} \\
\quad\quad \text{for } k := 0 \text{ to } n - j \text{ do} \\
\quad\quad\quad p_{is,k}^j := (1 - \bar{w})p_{is,k}^{j-1} + \bar{w}p_{is,k+1}^{j-1} \\
\quad\quad\quad \text{endfor} \\
\quad\quad p_{is} := p_{is,0}^n \\
\quad \text{endfor} \\
\text{for } i := 0 \text{ to } m \text{ do} \\
\quad p_{0s}^0 := p_{is} \\
\quad \text{endfor} \\
\text{for } j := 1 \text{ to } m \text{ do} \\
\quad \text{for } i := 0 \text{ to } m - j \text{ do} \\
\quad\quad p_{is}^j := (1 - \bar{u})p_{is}^{j-1} + \bar{u}p_{is+1}^{j-1} \\
\quad\quad \text{endfor} \\
\quad \text{endfor}
\]

\[1\text{ Note by Tamal K. Dey, Ohio State U.}\]
endfor
Return $p^m_0$
end

The complexity of the algorithm as written above is $O(mn^2 + m^2)$. One can switch the roles of $m$ and $n$ and obtain an algorithm with complexity $O(nm^2 + n^2)$ (Write the algorithm). Observe that taking $A = mn^2 + m^2$ and $B = nm^2 + n^2$, we get $A - B = (n - m)(mn - m - n) = mn(n - m)(1 - 1/n - 1/m)$. The right hand side is greater than zero if $n > m$ and $n, m > 2$. Therefore, it is better to switch the role of $m$ and $n$ in the above algorithm if $n > m$. 