Subdivision

1 B-splines

Subdivision can be used to draw the B-spline curves in a way similar to Bezier curves. The B-spline subdivision can be obtained by superposing two kinds of operations:

Doubling which maps a sequence of control points

$$P_0, P_1, P_2 \ldots, P_n$$

into the (twice as long) sequence

$$P_0, P_0, P_1, P_1, P_2, P_2, \ldots, P_n, P_n$$

and

Averaging which maps

$$P_0, P_1, P_2, \ldots, P_n$$

into the sequence of averages of each pair,

$$\frac{P_0 + P_1}{2}, \frac{P_1 + P_2}{2}, \frac{P_2 + P_3}{2}, \ldots, \frac{P_{n-1} + P_n}{2}.$$

More precisely, in the case of B-splines of degree k, one needs to perform one doubling operation and follow it by k averaging steps. The resulting sequence of control points defines the same curve as the initial one. Moreover, iterating the subdivision produces sequences of points converging to the B-spline curve at a very fast rate; thus, by performing a few subdivision steps and then just joining the consecutive points one can get an excellent approximation of the B-spline. Notice that, just as in the case of Bezier curves, both of the above operations are very cheap computationally (the averaging involves only addition of floats and division by two, which can be computed e.g. by decrementing the exponent and therefore is much cheaper than a general division).

Example. Let's consider subdivision for B-splines of degree 1. Let the control points be

$$P_0, P_1, P_2 \ldots, P_n.$$

The doubling stage yields

$$P_0, P_0, P_1, P_1, P_2, P_2, \ldots, P_n, P_n.$$

As a result of averaging, we get

$$P_0, \frac{P_0 + P_1}{2}, P_1, \frac{P_1 + P_2}{2}, P_2, \frac{P_2 + P_3}{2}, \ldots, \frac{P_{n-1} + P_n}{2}, P_n.$$

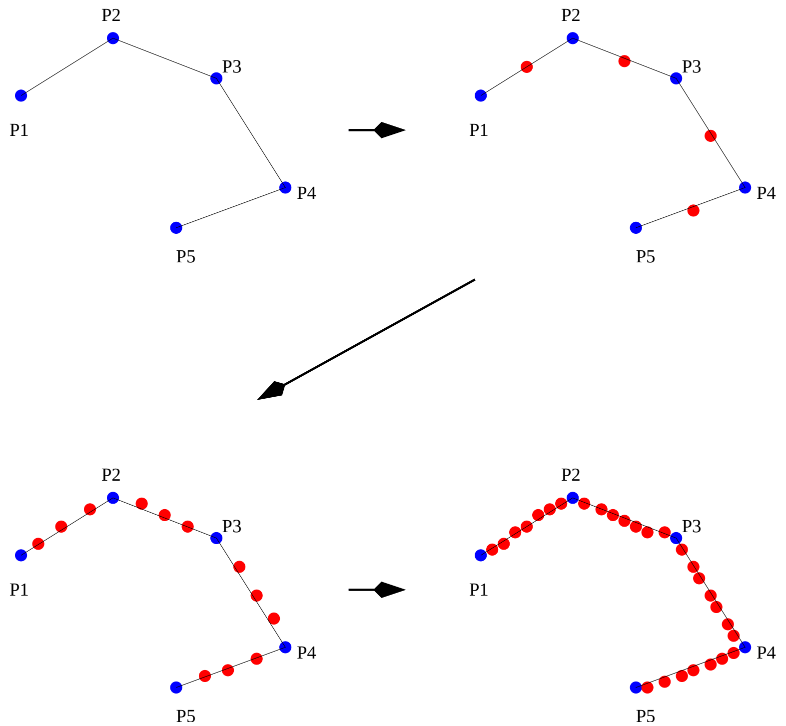

Figure 1: Subdivision for degree-1 B-splines. At each steps it adds all the midpoints (I tried really hard to put them at midpoints). After a few iterations, all the points are densely stretched along the B-spline curve (here: piecewise linear curve which joins the consecutive control points).

Thus, between each pair of consecutive control points, the subdivision inserts their average. Figure 1 explains how it works.

Another example. Now, let's take a closer look at subdivision procedure for cubic B-spline curves (most common in applications). As before, let's start from control point sequence

$$P_0, P_1, P_2 \ldots, P_n.$$

The doubling stage yields

$$P_0, P_0, P_1, P_1, P_2, P_2, \ldots, P_n, P_n.$$

Now, we need to do *three* subdivision steps. They produce the following sequences:

$$P_0, \frac{P_0 + P_1}{2}, P_1, \frac{P_1 + P_2}{2}, P_2, \frac{P_2 + P_3}{2}, \ldots, \frac{P_{n-1} + P_n}{2}, P_n,$$

then

$$\frac{3P_0+P_1}{4}, \frac{P_0+3P_1}{4}, \frac{3P_1+P_2}{4}, \frac{P_1+3P_2}{4}, \ldots, \frac{3P_{n-1}+P_n}{2}, \frac{P_{n-1}+3P_n}{2},$$

and finally

$$\frac{P_0+P_1}{2}, \frac{P_0+6P_1+P_2}{8}, \frac{P_1+P_2}{2}, \frac{P_1+6P_2+P_3}{8}, \ldots, \frac{P_{n-2}+6P_{n-1}+P_n}{8}, \frac{P_{n-1}+P_n}{2}.$$

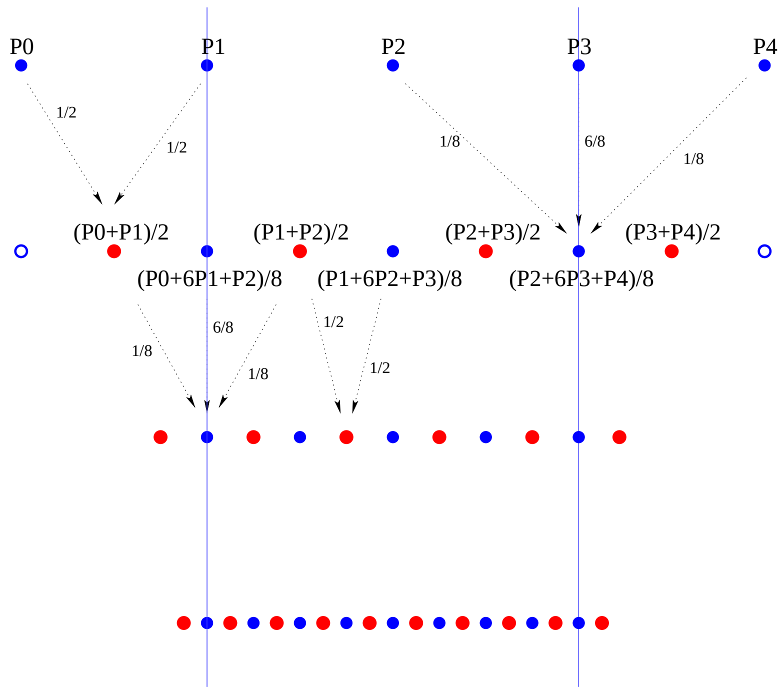

Figure 2: The Cubic B-spline Subdivision. One can think of one step as a procedure which adds a midpoint between each two on the previous level (all midpoints are shown in red above) and moves each of the existing points a bit (displaced 'old' points are shown blue). The new locations of the points are computed as weighted averages with weights of 1/8,6/8 and 1/8. Notice that we have insufficient data to compute the new locations of the first and last vertices, so they are simply removed (blue circles shown on the second level from top). As we keep subdividing, the points cover densely the space between the points corresponding to P1 and P3 (i.e. second and second last of the initial control points). Thus, the B-spline curve doesn't start at P0, but rather somewhere close to P1; more precisely, at a point which results from P1 after moving it in each of the subdivision steps.

2 Four point rule

Sometimes it is useful to have a nice smooth curve passing (unlike the B-splines) exactly through the specified control points. A particularly simple way to achieve this is by interpolating subdivision schemes. Interpolating subdivision does not move the existing control points. It only inserts one (defined by special rules) in between each pair. In the case of the four-point rule, the point inserted between P_i and P_{i+1} is given by

$$-\frac{1}{16}P_{i-1} + \frac{9}{16}P_i + \frac{9}{16}P_{i+1} - \frac{1}{16}P_{i+2}.$$

Thus, if the initial control points are

$$P_0, P_1, P_2 \ldots, P_n$$

then the points resulting from subdivision are

$$P_1, \frac{-P_0 + 9P_1 + 9P_2 - P_3}{16}, P_2, \frac{-P_1 + 9P_2 + 9P_3 - P_4}{16}, P_3, \ldots, \\ P_{n-2}, \frac{-P_{n-3} + 9P_{n-2} + 9P_{n-1} - P_n}{16}, P_{n-1}.$$

Notice that, since P_{-1} is not available, we can't compute the point to be inserted just before P_1; this is why the sequence starts with P_1. The 4-point rule with the above control point sequence starts at P_2, ends at P_{n-2} and passes through all control points in between the two. The process is schematically shown in Figure 3.

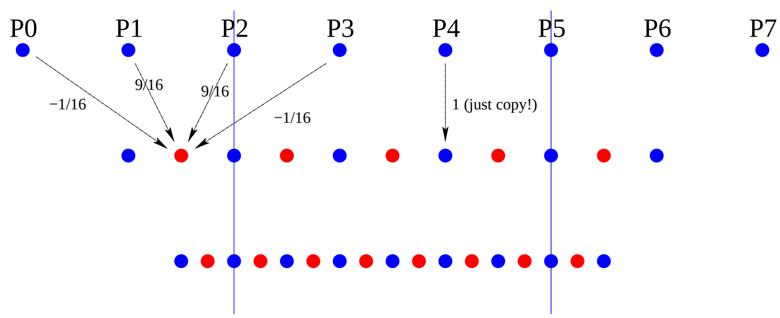

Figure 3: Four point subdivision; the points which existed on the previous level (blue) stay in place. Red points are computed by combining the neighboring four blue points on the previous level with weights $-\frac{1}{16}$, $\frac{9}{16}$, $\frac{9}{16}$ and $-\frac{1}{16}$. The limit curve starts at the third (P_2) and ends at the third from the end (in this case, P_5).

3 Subdivision for surfaces: Loop scheme

Subdivision is also possible in the surface setting. The Loop Subdivision Scheme allows to turn a triangulated mesh into a much smoother-looking one by applying a process similar to those discussed above for curves. Mathematically speaking, it produces a sequence of triangulated surfaces which converges to a smooth (having a tangent plane everywhere) surface. See Figure 4 for an example.

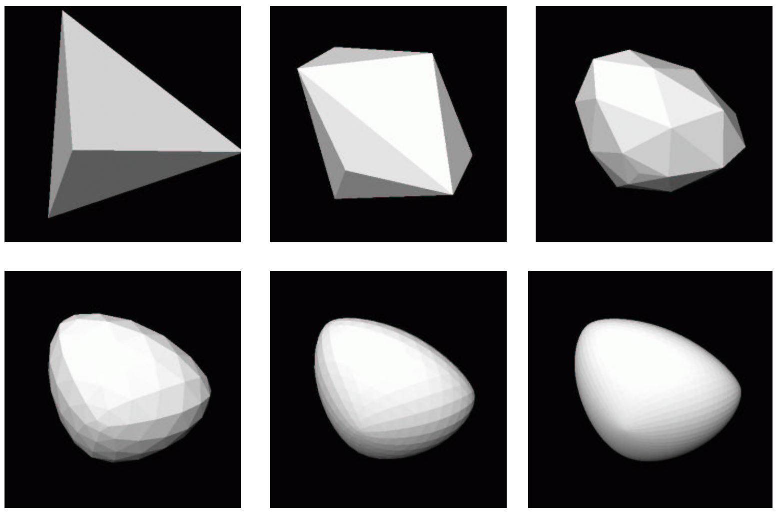

Figure 4: Loop subdivision applied to a tetrahedron (top left). Each next picture shows the result of applying subdivision to the previous one. Notice how smooth the surface get after a few subdivision steps.

The schematic picture of a subdivision is shown in Figure 5.

Now, we'll specify precisely how the red points and the new locations for the black points are computed. All the calculations are based on the immeditely neighboring points before subdivision; the resulting points are their weighted combinations. The weights are shown in Figure 6.

For the example mesh shown in Figure 7, the new vertex correswponding to the edge joining P2 and P9 is given by

$$Q = \frac{3}{8}P_2 + \frac{3}{8}P_9 + \frac{1}{8}P_1 + \frac{1}{8}P_3.$$

The new location of the vertex P_9 is given by

$$\text{New}P_9 = \frac{5}{8}P_9 + \frac{3}{8*8}(P_1 + P_2 + \ldots + P_8).$$

Final remarks on Loop Subdivision. In order for the computation to be possible, we need a *manifold mesh.* In a manifold mesh, each edge has exactly

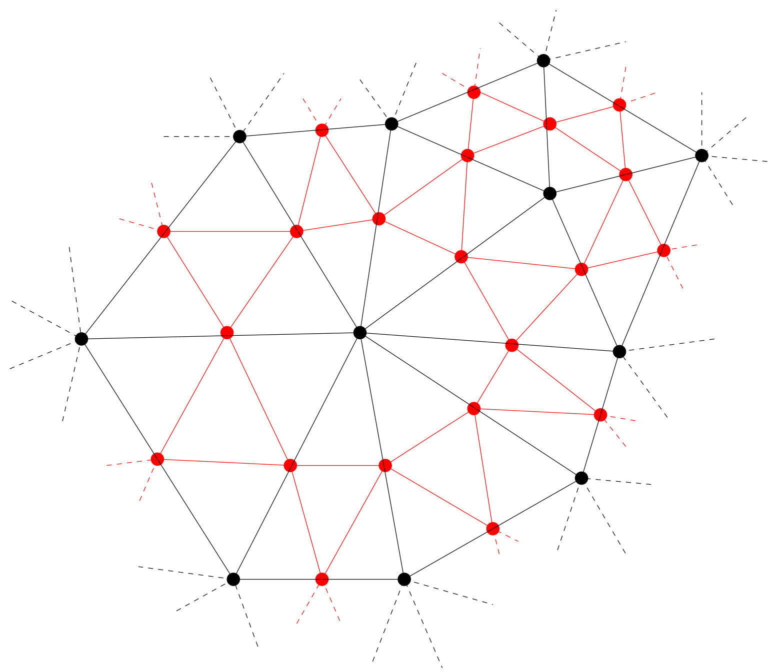

Figure 5: Loop subdivision adds 'edge vertices' (red points), one per edge. Each of the triangles of the input mesh gets split into four. In the figure, the black lines show the initial mesh and, shown in red, are vertices and edges added by the subdivision procedure. Note that this figure is meant to show the logical relationship between the points before and after subdivision. The red points are typically not at all on the edges joining the black points (but they correspond to edges: one per edge is added). Also, the black points do *not* stay in place, but each of them has a corresponding vertex in the mesh before subdivision.

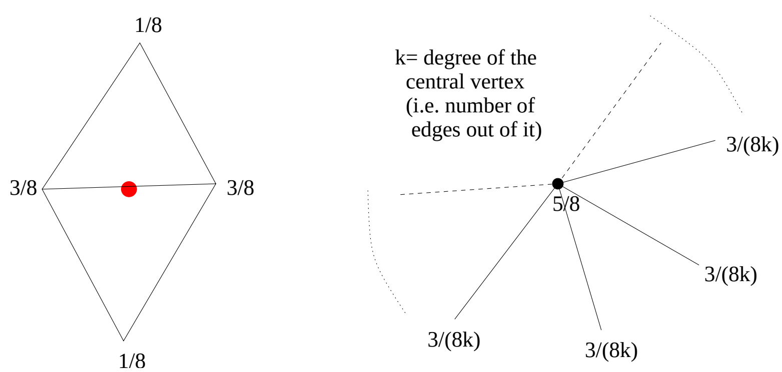

Figure 6: The weights for the Loop subdivision scheme.

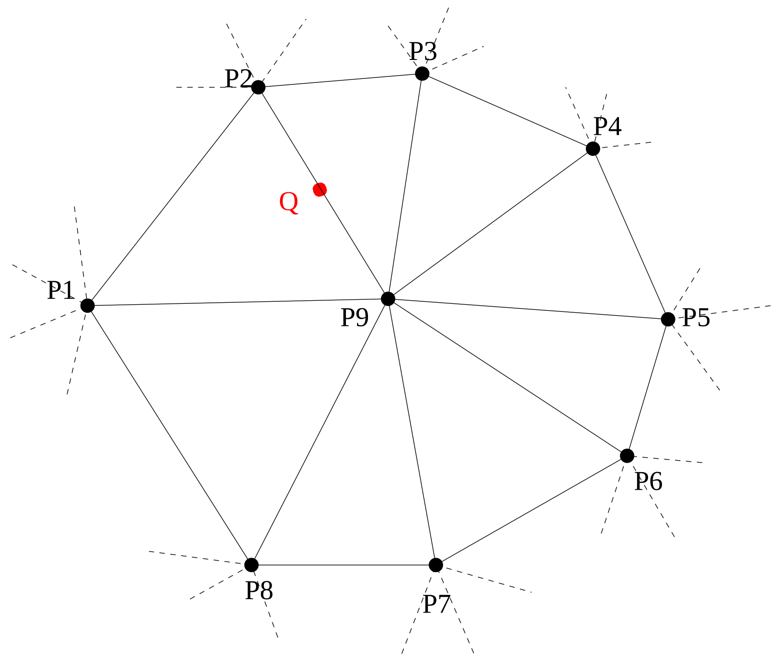

Figure 7: An example.

two incident trialgles. Also, each vertex needs to have a closed fan of triangles around it. A manifold mesh as described above cannot have boundary. It should be intuitively clear that it is possible to obtain a nice manifold triangulation of a surface of any 3D solid. One only needs to ensure that the triangles fit tightly together.

For vertices of degree 3, weights of 7/16 for the central vertex and 3/16 for the three neighboring ones work better than the ones given above.