Nurbs-NG notes.

Carlo Dormeletti (onekk)

August 6, 2023

Abstract

These are mostly personal notes about Nurbs WB, mostly theoretical intro to BSplines and some notes to not pollute the code too much.

Contents

1 Introduction

https://en.wikipedia.org/wiki/Non-uniform_rational_B-spline

This post from by Crish_G: https://forum.freecad.org/viewtopic.php?p=561221#p561221

OCCT Geom.BSplineSurface documentation.

NURBS will stand for **N**on **U**niform **R**ationale **B**spline **S**urface.
Nurbs are a special case of rational B-splines, just as uniform B-splines are a special case of non-uniform B-splines. Thus, the term Nurbs encompass almost every other possible 3D shape definition.
These Surfaces are defined by some data:

Poles are also called "control points".

Knots a sequence of parameter values that determines where and how Poles affect the final surface.

Multiplicities are repetition of knots.

Weights When Weights are different from 1 we have a NURBS, ie the surface is Rational.

BSplines are special calculated curves derived from mathematical equations, so complex curves could be described and calculated using a limited set o parameter.
Descritpion below could use even Curves instead of Surface, there is not more differences in concepts, and usually most theory is referred to curves and then Generalized in the U, V parametric space.
There are n + 1 control points, $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{n+1}$. The Ni,k basis functions are of order k(degree k-1). k must be at least 2 (linear), and can be no more than n+1 (the number of control points). The order of the curve (linear, quadratic, cubic,...) is not dependent on the number of control points.
For most part of the description you could think a BSpline Surface as the result of a net of Bspline Curves in U and V parametric direction.

1.1 Knots

The knot vector can, by its definition, be any sequence of numbers provided that each one is greater than or equal to the preceding one. In simple words Knot vector should be in non decreasing order.

Number of knots = Numpoles + degree + 1.

Most often in Nurbs articles, knots arrays are written WITH knot repetition, in OpenCascade, knots are described by 2 arrays:

1. knots (without repetition)
2. multiplicities, see below.

1.2 Weights

Rational characteristic is defined in each parametric direction (U, V), in other word a NURBS could be rational in U but not in V or the other opposite.

Rational B-splines are defined simply by applying the B-spline equation to homogeneous coordinates, rather than normal 3D coordinates.

This lead that in Nurbs poles could be written in homogeneous coordinates 4D : (x, y, z, w), where 'w' is the weight associated to a pole.

This means that always: len(weights) = len(poles)

1.3 Multiplicities

Shorted in *Mults* So always: len(mults) = len(knots).

1 ¡= Mults(i) ¡= Degree

This knots vector = [0.0, 0.0, 0.0, 1.0, 2.0, 2.0, 2.0] written in Traditional way will became:

knots = [0.0, 1.0, 2.0]

mults = [3, 1, 3]

Mults could have some special cases, where the knots are regularly spaced in one parametric direction; in other word difference between two consecutive knots is constant.

Uniform : all the mults are equal to 1.

Quasi-uniform : all the mults are equal to 1, except for first and last knots, and these are equal to Degree + 1.

Piecewise Bezier : all the mults are equal to Degree except for first and last knots, which are equal to Degree + 1. Resulting surface is a concatenation of Bezier patches in PD.

2 Some considerations

In "not periodic" surface:

- bounds of knots and mults tables are: 1 ¡ knot ¡ NbKnots where NbKnots is the number of knots of the BSS in parametric direction.
- first and last mults may be Degree+1 (this is recommended if you want the curve to start and finish on the first and last pole).
- Poles.ColLength() == Sum(UMults(i)) - UDegree - 1 ¿= 2 (for U)
- Poles.RowLength() == Sum(VMults(i)) - VDegree - 1 ¿= 2 (for U)

In "periodic" surfaces:

- first and last mults must be the same.
- given k periodic knots and p periodic poles in para,etric direction:

- period is such that: period = Knot(k+1) - Knot(1),
- poles and knots tables in PD can be considered as infinite tables, such that:
  - Knot(i + k) = Knot(i) + period,
  - Pole(i + p) = Pole(i)
- Poles.ColLength() == Sum(UMults(i)) except first or last. (for U)
- Poles.RowLength() == Sum(VMults(i)) except first or last. (for V)

Note: Data structure tables for a periodic BSpline surface are more complex than those of a non-periodic one.