Approximation by NURBS with free knots

M. Randrianarivony G. Brunnett

Technical University of Chemnitz, Faculty of Computer Science
Computer Graphics and Visualization
Straße der Nationen 62, 09107 Chemnitz, Germany
Email: maharavo@informatik.tu-chemnitz.de

Abstract

This paper deals with approximating noisy samples by NURBS with special emphasis on free knots. We consider the knots as unknown parameters so as to find their optimal positions. The three unknowns (the weights, the control points and the knots) are therefore mixed together in a nonlinear way. After recalling briefly the fixed knot problem, we show a way to discard the first two unknowns. The problem becomes then simpler and we show how to solve the resulting problem with only one parameter which is the free knots. Modification of the algorithm is also treated so as to avoid undesired knot position such as non-increasing knot sequence. Implementation aspect and numerical results are given at the end to support the idea.

1 Introduction

NURBS settings are appreciated in many theoretical analyses because they allow flexible description of both free form surfaces and usual geometries such as conic sections. Indeed, the set of rational functions is much larger than that of polynomial functions, so NURBS give mainly better approximation than their B-spline counterparts do. Another reason for the appreciation of NURBS is that it is supported by many softwares. For instance, OpenGL and ACIS ([11], [4]) have built-in commands for drawing NURBS by only giving the required parameters.

Approximations with NURBS have already been treated in many documents (see among others [3], [5]). In [3], the author uses an iterative segment determination in order to establish the positions of the knots. In [5], ... In the context of B-splines, the use of free knots has already been investigated by several authors ([6], [7], [8]).

The purpose of this paper is the description of NURBS approximation with free knots. That means, we aim particularly at finding the optimal knot position while determining the other parameters which are the control points and the weights. One of the main difficulties of NURBS over B-spline is the establishment of the weights which should be positive. On the other hand, since we deal with rational functions, many of the formulations lead to nonlinear problems.

In section 2, we formulate in detail the meaning of free knots. We recall also briefly the method used for fixed knots ([3]). The main idea for solving the free knot problem is given in section 3. Numerical results are given in the last section. In this document, we treat only curves, the case of surfaces will be done in a future paper.

2 Problem setting and notations

2.1 NURBS

A NURBS (nonuniform rational B-splines) with weights $w_0, ..., w_n \in \mathbf{R}_+$ and control points $\mathbf{d}_0, ..., \mathbf{d}_n \in \mathbf{R}^3$ is given by:

$$\mathbf{X}(t) := \frac{\sum_{i=0}^{n} w_i \mathbf{d}_i N_i^k(t)}{\sum_{i=0}^{n} w_i N_i^k(t)}, \qquad (1)$$

where N_i^k is the usual ([1]) B-spline basis defined on a knot sequence:

$$\theta_0 = ... = \theta_{k-1}, \theta_k, ..., \theta_n, \theta_{n+1} = ... = \theta_{n+k}.$$

Remark 1 For our case, we deal only with the case

$$\theta_0 = ... = \theta_{k-1} = 0$$
$$\theta_{n+1} = ... = \theta_{n+k} = 1$$

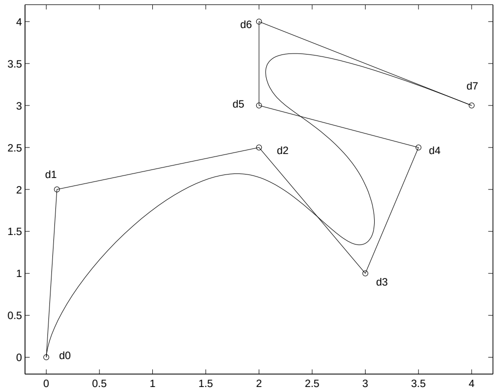

Figure 1: NURBS with $n = 7$, $k = 4$ $\mathbf{T} = (0, 0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1, 1)$

In the sequel, we will denote:

$$\mathbf{W} := (w_0, ..., w_n) \tag{2}$$
$$\mathbf{D} := (\mathbf{d}_0, ...\mathbf{d}_n) \tag{3}$$
$$\mathbf{T} := (\theta_k, ..., \theta_n). \tag{4}$$

2.2 Free knot problem

Suppose we are given noisy samples $(t_i, \mathbf{M}_i)$ $i = 0, ..., M$. M is supposed to be very large. Like in other fitting, we want to find the NURBS curve $\mathbf{X}(t) = \mathbf{X}_{\mathbf{W},\mathbf{D},\mathbf{T}}(t)$ which best fits these data. Because we want to find the optimal positions of the knots, we put them as variables. That means, we have the following problem:

$$\min_{\mathbf{W},\mathbf{D},\mathbf{T}} \sum_{i=0}^{M} \|\mathbf{X}_{\mathbf{W},\mathbf{D},\mathbf{T}}(t_i) - \mathbf{M}_i\|^2 . \tag{5}$$

This problem is very difficult because the 3 unknowns $(\mathbf{W}, \mathbf{D}, \mathbf{T})$ are all mixed in $\mathbf{X}_{\mathbf{W},\mathbf{D},\mathbf{T}}$ non-linearly. Furthermore, we need to add some constraints about the positivity of the weights. In the next section, we will show how to simplify this problem.

2.3 Brief recall of the fixed knot problem

If we are given a knot sequence $\mathbf{T}$, then the problem

$$\min_{\mathbf{W},\mathbf{D}} \sum_{i=0}^{M} \|\mathbf{X}_{\mathbf{W},\mathbf{D},\mathbf{T}}(t_i) - \mathbf{M}_i\|^2 .$$

will be refered to as fixed knot problem. It is investigated in [3] where it is shown to be equivalent to solving a linear system:

$$(A + \lambda B)\mathbf{y} = \lambda \mathbf{r} , \tag{6}$$

where A and B are given in block structure:

$$A := \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \; B := \begin{bmatrix} 0 & 0 \\ 0 & \tilde{B} \end{bmatrix}$$

$$A_{rs} := \begin{bmatrix} \Sigma\overline{N}^i_{00}\mathbf{Z}^{rs}_i & \cdots & \Sigma\overline{N}^i_{0n}\mathbf{Z}^{rs}_i \\ \vdots & & \vdots \\ \Sigma\overline{N}^i_{n0}\mathbf{Z}^{rs}_i & \cdots & \Sigma\overline{N}^i_{nn}\mathbf{Z}^{rs}_i \end{bmatrix} \tag{7}$$

$$\mathbf{Z}^{(1,1)}_i := \mathbf{I} - c_i\mathbf{M}_i\mathbf{M}_i^T \tag{8}$$
$$\mathbf{Z}^{(1,2)}_i := c_i\mathbf{M}_i \tag{9}$$
$$\mathbf{Z}^{(2,1)}_i := c_i\mathbf{M}_i^T \tag{10}$$
$$\mathbf{Z}^{(2,2)}_i := 1 - c_i \tag{11}$$

$$\tilde{B} := \begin{bmatrix} \Sigma\overline{N}^i_{00} & \cdots & \Sigma\overline{N}^i_{0n} \\ \vdots & & \vdots \\ \Sigma\overline{N}^i_{n0} & \cdots & \Sigma\overline{N}^i_{nn} \end{bmatrix} \tag{12}$$

$$\begin{aligned} \mathbf{y} &:= [\bar{\mathbf{d}}_0, ..., \bar{\mathbf{d}}_n, w_0, ..., w_n]^T \\ \mathbf{r} &:= [\mathbf{0}, ..., \mathbf{0}, \Sigma N^k_0(t_i), ..., \Sigma N^k_n(t_i)]^T \\ \mathbf{I} &:= \text{identity matrix of order } 3 \\ \bar{\mathbf{d}}_i &:= [w_i d_{ix}, w_i d_{iy}, w_i d_{iz}] \\ \overline{N}^i_{pq} &:= N^k_p(t_i) N^k_q(t_i) \\ c_i &:= 1/(1 + \mathbf{M}^2_i). \end{aligned}$$

In all these expressions, Σ is understood to be $\sum_{i=0}^{M}$ and λ is a positive constant which should be chosen large enough (see [3]) in order to ensure positivity of the weights.

3 Free knots

3.1 Simplification of the problem

The position of the knots is very important in NURBS fitting. A bad placement of knots may lead to a remarkable distortion of the geometry. One needs therefore a way to find the best position of the knots. In this section, we intend to simplify the problem (5) which has three parameters. First, we can reduce it into a problem with one parameter only. Indeed, according to section 2.3, for a given knot $\mathbf{T}$, we can solve the subproblem (6) in order to determine the corresponding weights $\mathbf{W}$ and the control points $\mathbf{D}$. In other words $\mathbf{W}$ and $\mathbf{D}$ are functions of $\mathbf{T}$ i.e.

$$(\mathbf{W}, \mathbf{D}) = (\mathbf{W}(\mathbf{T}), \mathbf{D}(\mathbf{T})) .$$

Problem (5) is therefore simplified into:

$$\min_{\mathbf{T}} \sum_{i=0}^{M} \|\mathbf{X}_{\mathbf{W}(\mathbf{T}),\mathbf{D}(\mathbf{T}),\mathbf{T}}(t_i) - \mathbf{M}_i\|^2 . \qquad (13)$$

3.2 Penalization of undesired knot position

From now on, we will write only $\mathbf{X}(t_i)$ instead of $\mathbf{X}_{\mathbf{W}(\mathbf{T}),\mathbf{D}(\mathbf{T}),\mathbf{T}}(t_i)$ in order to simplify the notation. We have then

$$\min_{\mathbf{T}} \sum_{i=0}^{M} \|\mathbf{X}(t_i) - \mathbf{M}_i\|^2 . \qquad (14)$$

This problem still allows the presence of the situation where θ_i is not increasing. In this section, we will modify this problem so that only knots with $\theta_k \leq \theta_{k+1} \leq ... \leq \theta_n$ may happen. By denoting:

$$\mathbf{X}(t) = (X_x(t), X_y(t), X_z(t)) \text{ and}$$
$$\mathbf{P}_i = (P_{ix}, P_{iy}, P_{iz}) ,$$

we have

$$\min_{\mathbf{T}} \sum_{i=0}^{M} (X_x(t_i) - P_{ix})^2 + (X_y(t_i) - P_{iy})^2 + (X_z(t_i) - P_{iz})^2 .$$

And by defining

$$\begin{cases} S_{3i} & := & X_x(t_i) - P_{ix} \\ S_{3i+1} & := & X_y(t_i) - P_{iy} \\ S_{3i+2} & := & X_z(t_i) - P_{iz} , \end{cases}$$

$$\min_{\mathbf{T}} \sum_{i=0}^{3M+2} S_i^2 . \qquad (15)$$

Note that (15) is nothing else but a scalar version of problem (13) which involves vector valued expressions. We introduce now the function

$$R(x) := \begin{cases} 0 & \text{if} \quad x > 0 \\ (-x)^3 & \text{if} \quad x \leq 0. \end{cases}$$

And we define

$$\begin{aligned} \text{Tail}(\mathbf{T}) &:= R(\theta_{k+1} - \theta_k) + \\ & R(\theta_{k+2} - \theta_{k+1}) + ... + R(\theta_n - \theta_{n-1}) . \end{aligned} \qquad (16)$$

We modify then the problem (15) into

$$\min_{\mathbf{T}} \sum_{i=0}^{3M+2} [S_i + \alpha \text{Tail}(\mathbf{T})]^2 , \qquad (17)$$

where α is a very large positive number. Let us see the relation between (15) and (17) by remarking the following two properties of equation (17).

- If we have $\theta_k \leq \theta_{k+1} \leq ... \leq \theta_n$, then $\theta_{k+r} - \theta_{k+r-1} \geq 0$ for all $r = 1, ..., n-k$ and therefore $\text{Tail}(\mathbf{T}) = 0$. Thus,

$$\sum_{i=0}^{3M+2} [S_i + \alpha \text{Tail}(\mathbf{T})]^2 = \sum_{i=0}^{3M+2} S_i^2 .$$

- If there is some r such that $\theta_{k+r} < \theta_{k+r-1}$, then $R(\theta_{k+r} - \theta_{k+r-1}) > 0$ and so $\text{Tail}(\mathbf{T})$ is nonzero. Because of our assumption that α is a very large number, we can expect that $\sum_{i=0}^{3M+2} [S_i + \alpha \text{Tail}(\mathbf{T})]^2$ is also very large.

Since we are searching for the minimum of $\sum_{i=0}^{3M+2} [S_i + \alpha \text{Tail}(\mathbf{T})]^2$, the preceeding two points show that a $\mathbf{T}$ with $\theta_{k+r} < \theta_{k+r-1}$ can never realize this minimum. That means that the integration of the trailing term in (17) penalizes those $\mathbf{T}$ with $\theta_{k+r} < \theta_{k+r-1}$.

Remark 2 Sometimes, it is also desirable to have $|\theta_i - \theta_{i+1}| > \varepsilon$ for $i = k, ..., n-1$. In this case, we need only to replace (16) into

$$\begin{aligned} \text{Tail}(\mathbf{T}) &:= R(\theta_{k+1} - \theta_k - \varepsilon) + \\ & R(\theta_{k+2} - \theta_{k+1} - \varepsilon) + ... + R(\theta_n - \theta_{n-1} - \varepsilon) . \end{aligned} \qquad (18)$$

3.3 Implementation

By denoting $r(i, \mathbf{T}) := S_i + \alpha \text{Tail}(\mathbf{T})$ and $K = 3M + 2$, the problem is then a usual nonlinear least square problem which is

$$\min_{\mathbf{T}} \sum_{i=0}^{K} [r_i(i, \mathbf{T})]^2 \, .$$

Such a problem can be solved by nonlinear least square solvers like Levenberg-Marquardt and Gauss-Newton (see [9], [10]). Note that for each evaluation of the function $r(i, \mathbf{T})$, we need to solve the subproblem (6) in order to know the corresponding $\mathbf{W}(\mathbf{T})$, $\mathbf{D}(\mathbf{T})$. We remark that the order of the linear system (6) is small. It does not depend on the number of data points. It depends exclusively on the degree n of the NURBS. Furthermore, taking into account that the support of N_i^k is $[\theta_i, \theta_{i+k})$, we conclude that the matrices in (7) are banded. More precisely, we have:

$$\forall \beta_i \quad \sum_{i=0}^{M} \overline{N}_{pq} \beta_i = 0 \quad \text{for } |p - q| \geq k.$$

As a consequence, we need only to compute a few entries and the matrices are sparse. On the other hand, we must note that the computation of one entry of this system involves all data points. The remedy to that problem is to assemble the matrix and vector

$$F := \begin{bmatrix} I - c_0 \mathbf{M}_0 \mathbf{M}_0^T \\ I - c_1 \mathbf{M}_1 \mathbf{M}_1^T \\ \cdots \\ I - c_m \mathbf{M}_m \mathbf{M}_m^T \end{bmatrix}, \ \mathbf{c} := \begin{bmatrix} c_0 \\ c_1 \\ \cdots \\ c_m \end{bmatrix} \tag{19}$$

only once and store them in arraies so that they can be read and need not be recomputed in subsequent computations. Note that F and $\mathbf{c}$ are independent of $\mathbf{T}$. They depend only on the initial data points. The only things which need to be updated in each iteration of the nonlinear least square are the values of $N_l^j(t_i)$ which are mostly zero except for some few values. They can be computed with a fast algorithm (see [2]).

4 Numerical results

The numerical results in this paper were all done with the Levenberg-Marquardt algorithm. Note that this algorithm is iterative and so it needs some initial guess. The initial guess that we have taken here is equidistant knots. The first numerical test that we perform is the reconstruction of W-form curve. We have 75 data which were added with random noise of amplitude 0.1. The curve is reconstructed with the help of the formerly described algorithm. The overall time for the reconstruction is 12 seconds. In Figure 2 we see graphical illustration of the data, the initial curve and the reconstructed curve. The second test is a free-form curve. The time needed for the reconstruction is like in the first test. A graphical illustration can be found in Figure 3.

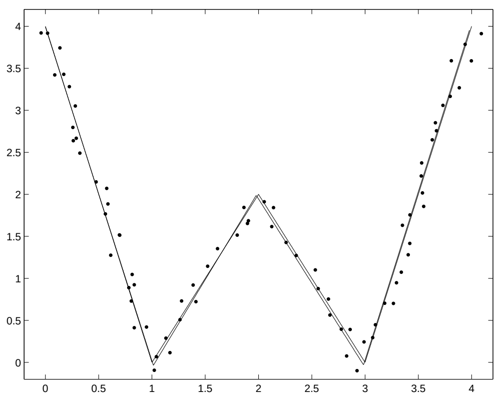

Figure 2: Initial curve, reconstructed curve and 75 samples with noise amplitude=0.1

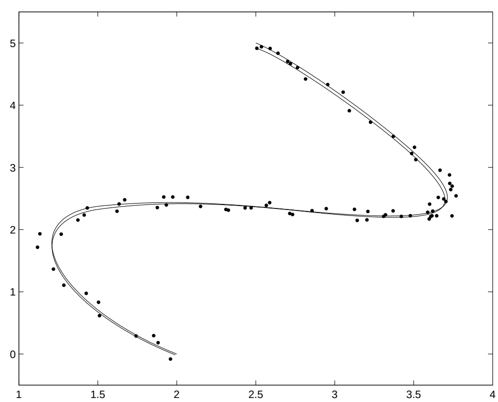

Figure 3: Initial curve, reconstructed curve and 75 samples with noise amplitude=0.1

5 Future work

All the computations that we have performed shows that the most expensive part of this algorithm is the assembly of the matrices in (6). The assembly takes in general more than 95% of the whole computational work. On the other hand, we can see that those matrices can be assembled in parallel because they consist only of sums of some terms which can be distributed on each processor. The first future work will deal with the parallelization of this algorithm in which each processor will have almost the same number of data points so as to ensure load balancing. The second future work will consist in extending this algorithm for surfaces.

References

[1] G. Farin, "*Curves and surfaces for computer aided geometric design*", Academic Press, Boston, 2. ed., 1990.

[2] C. de Boor, "*A practical guide to splines*", Springer, New York, 1978.

[3] B. Elsässer, "Approximation mit rationalen B-Spline Kurven und Flächen", *Ph.D. thesis*, Darmstadt, 1998.

[4] J. Corney, "*3D modeling with ACIS kernel and toolkit*", John Wiley & sons, Chichester, 1997.

[5] P. Laurent-Gengoux, and M. Mekhilef, "Optimization of a NURBS representation", *Computer Aided Design*, pp. 699-710, vol 25, No. 11, 1993.

[6] H. Schwetlick, and T. Schütze, "Least squares approximation by splines with free knots", *BIT*, pp. 361-384, vol 35, No. 3, 1995.

[7] D. Jupp, "Approximation to data by splines with free knots", *SIAM J. Numer. Anal.*, pp. 328-343, vol 15, No. 2, 1978.

[8] T. Schütze, "Diskrete Quadratmittelapproximation durch Splines mit freien Knoten", *Ph.D. thesis*, Dresden, 1998.

[9] J. Nocedal, and S. Wright "*Numerical Optimization*", Springer Series in Operation Research, New York, 1999.

[10] R. Fletcher, "*Practical methods of optimization*", John Wiley & Sons, Chichester, 2. ed., 1987.

[11] J. Neider, T. Davis, and M. Woo "*OpenGL programming guide*", Addison-Wesley publishing company, Reading, 1994.