Estimating Criteria for Fitting B-spline Curves: Application to Data Compression

Eric Saux and Marc Daniel

Institut de Recherche en Informatique de Nantes
Nantes, France

Abstract

In this paper, we analyse the different parameter choices for fitting B-spline curves. New estimating criteria for data approximation are introduced in order to estimate the results. The definitions of norms correspond to a global analysis of the curve. Other criteria are based on a local analysis. We present a new method for data compression using fitting B-splines and compare it to usual ones.

Keywords: Data compression, estimating criteria, fitting B-splines.

1 INTRODUCTION

Data compression relative to curves is a frequent problem in many applications. Many papers have already been published. Two trends can be emphasized. The first one deals with polygonal curves for approximating data (see subsection 4.1.1). Another approach was developed by Tom Lyche and Knut Morken [18]. Their method is representative of knot removal strategies. Matthias Eck and Jan Hadenfeld's strategy [10] is complementary to Lyche and Morken's. We suggest a different technique using fitting B-splines.

In section 2, we remind the reader of the general problem of data fitting with B-splines: a general form for solving this problem and the choices of knot vector, approximating parameters and degree. The usual parameterization techniques have been taken into account as well as the Foley and Nielson methods [12] and the intrinsic Hoschek parameterization [13].

In order to estimate the results we introduce two different tools in section 3. The first one corresponds to a global analysis and a definition of norms. It estimates whether one approximation is better than another according to a tolerance criterion. These norms take into account the oscillating phenomenon which may occur. The other estimating tool yields a local analysis. The notion of local estimation is introduced to avoid dependence on parameter values.

Section 4 deals with data compression strategies. In subsection 4.1, we propose a survey of current methods. Our data compression technique using a local approach based on curve analysis is introduced in subsection 4.2. A bisection method on the number of control points can be applied: to decrease errors one has to globally increase the number of control points. Lastly, we compare our data compression technique to usual ones in section 5 according to compression rates and computation costs.

2 DATA FITTING WITH B-SPLINE CURVES

B-spline functions are often used for curve modelling. This basis is fairly well conditioned and has many other nice properties which usually lead to stable and simple algorithms. Basic properties of splines and B-splines can be found in [8, 11].

2.1 Least squares fitting

The formulation of our fitting problem is to define, for a set of $(n+1)$ different ordered points $P = (p_0, ..., p_n)$ in a space $\mathbb{R}^d$, a B-spline curve f as close as possible to data p_i. We briefly present our notations. Let

- k be the B-spline curve order (degree+1) with $k \leq m$,
- $T = (t_i)_{i=0}^{m+k}$ be the knot vector defined by a non-decreasing sequence of numbers so that $t_0 = t_1 = ... = t_{k-1} < t_k$, $t_m < t_{m+1} = t_{m+2} = ... = t_{m+k}$ and $t_i < t_{i+1}$ $(k-1 \leq i \leq m)$,
- $(\zeta_i)_{i=0}^{n}$ be $(n+1)$ parameter values,
- $(Q_i)_{i=0}^{m}$ be $(m+1)$ control points.

Associated B-spline curve f is defined by:

$$f(t) = \sum_{j=0}^{m} Q_j N_{j,k,T}(t) \qquad Q_j = (Q_{1j}, ..., Q_{dj})^T \in \mathbb{R}^d \quad (1)$$

where functions $N_{j,k,T}$ are the normalized basis functions computed with the De Boor formula. The least squares fitting problem searches for control points $(Q_j)_{j=0}^{m}$ of curve f so that $(n+1)$ points $f(\zeta_i)$ produce a least squares smoothing of the set of points p_i. The problem is to find control points Q_j so that:

$$\sum_{i=0}^{n} (f(\zeta_i) - p_i)^2 \text{ is minimum.} \quad (2)$$

We assume that $m < n$ in order to really obtain a fitting problem. Since (2) corresponds to the minimization of the sum of d independent positive quantities (one per coordinate), it is equivalent to d independent minimizations, one per coordinate. With a matrix formulation, M_q and M_p being respectively the $(m+1, d)$ and $(n+1, d)$ matrices of points Q_i and p_i, we have to find a matrix M_q which minimizes the norm of each column vector E_i $(i = 1, ..., d)$ of the matrix $A.M_q - M_p$. The problem is expressed in the matrix form (system of normal equations) by:

$$A^T.A.M_q = A^T.M_p$$

This formulation is associated with a square matrix with an often high condition number involving potential numerical difficulties or bad results. Actually, the condition number of the matrix $A^T.A$ is the square of the condition number of A (in 2-norm). Therefore, the

results are necessarily better when solving the overdetermined system $A.M_q = M_p$ (system $(n+1, m+1)$) by a least squares technique through a $Q.R$ matrix decomposition. This equivalent theoretical formulation of the initial problem guarantees far better numerical results, with far fewer computations [7].

We should now specify:

- knot vector T,
- $(n+1)$ parameter values ζ_i,
- order k.

These three components are very important for obtaining fair results in approximation and in interpolation.

2.2 Knot vector

Knot vector T can be uniform or defined from an extension of the De Boor formula for interpolation [6]:

$$\begin{cases} t_0 = ... = t_{k-1} = \zeta_0 \\ t_{m+1} = ... = t_{m+k} = \zeta_n \\ t_{i+k} = \frac{\zeta_{i+l_1} + ... + \zeta_{i+l_2}}{l_2 - l_1 + 1} \quad for \quad i = 0, ..., m-k \end{cases} \tag{3}$$

The extreme values of l_1 and l_2 are fixed by the compatibility with interpolation. If l_1 and l_2 are constants, the number of ζ_i in the summation depends on m. If this number is fixed (equal to $k-1$), this implies that l_1 and l_2 are functions of i. Choosing constant values for l_1 and l_2 is not satisfactory as soon as an important difference between n and m exists. As a matter of fact, the number of ζ_i in the summation is then high, and all the knots are located near an average value. Simple rules of variation for l_1 and l_2 must be found. The following ones are compatible with the interpolation problem:

$$l_1(i) = E(\frac{n-m}{m-k}i) + 1 \tag{4}$$

$$l_2(i) = E(\frac{n-m}{m-k}i) + k - 1 \tag{5}$$

$E(x)$ being the truncated integer (x + 0.5).

These functions give all different knots and provide a good distribution between parameters ζ_i and knots t_i. We obtain in the particular case $m = k$:

$$t_k = \frac{\zeta_1 + ... + \zeta_{m-1}}{m-1}$$

The number of control points can continuously increase from a low value up to $n+1$. Interpolation and least squares fitting have exactly the same processing formulation (3).

2.3 Parameterization

This is a difficult problem and some solutions are proposed. The poorest one is a uniform spacing. Points p_i are assumed to be different providing the definition of a strictly increasing sequence of parameters. A parameter distribution using arc lengths was tested. The possible improvement does not justify this iterative method. The chord length parameterization is commonly proposed. Lee defined a centripetal parameterization and its general formulation [17]:

$$\begin{cases} \zeta_0 = 0 \\ \zeta_i = \frac{\sum_{j=1}^{i} \|p_j - p_{j-1}\|^e}{\sum_{j=1}^{n} \|p_j - p_{j-1}\|^e} c \quad i = 1, ..., n \quad (0 \le e \le 1) \end{cases} \tag{6}$$

We obtain respectively a uniform, chord length and centripetal model, with a parameter e equal to 0, 1 and 0.5. c is a constant expanding the distribution (to reduce numerical problems).

Two spacings have been proposed by Foley and Nielson [12]. The authors call them respectively the affine invariant chord spacing and the affine invariant angle spacing. They both use a distance deduced from statistical theory. The second spacing takes into account the angles between the different line segments joining the data points. The latter is particularly efficient when important variations in the angles occur. The authors prove the interesting result that with these parameterizations, the spline interpolation method is affine invariant. This means that we obtain the same result by applying an affine transformation to a spline fitting curve and by computing the spline fitting once the affine transformation has been applied to the given points. This is of importance in a CAD system where geometric transformations often occur. It must be noticed that the parameterization defined by (6) satisfies this property except if the transformation is a non homogeneous scaling. These properties are deduced from the fact that spacings are not modified by such transformations [12] (it is obvious that translations, rotations and homogeneous scalings remain the spacings defined by (6) unchanged).

In Hoschek's paper [13], an iterative approach is proposed to find intrinsic parameter values leading to a better approximation. Hoschek's concept is to find a sequence of new parameter values $\tilde{\zeta}_i$ for which corresponding points on f are closer to data p_i than latter ζ_i. Then we start the least squares process again with the new parameter values and repeat these steps until all error vectors $p_i - f(\tilde{\zeta}_i)$ are approximately orthogonal to the approximating curve. At each step of the process correction parameters $\tilde{\zeta}_i$ are computed. But for a fixed B-spline curve, they do not correspond to the closest approximations of data p_i. We suggest replacing them by parameters $\bar{\zeta}_i$ whose values on the B-spline curve are now the closest approximations of data [24]. We can apply descent algorithms to move along the curve and reach these optimal parameters $\bar{\zeta}_i$ (see subsection 3.2). The result is a better global approximation with a faster convergence speed.

2.4 Degree

Order k can be defined with regard to:

- computation speed: the higher the degree, the slowest the computations.
- shape modelling: the higher the degree, the higher the number of shapes which could be modelled. An order 2 produces fair approximations of line segments. An order 4 (cubic B-spline curves) is commonly used for having fair approximations of parabolic sections and for producing inflection points.
- differential parameters: order 4 is the lowest for computing continuous curvatures and tangents.

B-spline order k can be changed. But, the condition number of the system increases very quickly when k is rising. This implies the distortion of the control polygon (the line segments connecting control points Q_i). As a result, k cannot increase too much for numerical reasons.

3 CRITERIA FOR ESTIMATING DATA APPROXIMATION

The main problem is now to measure the accuracy for data fitting. The difficulty is to estimate curve f between given points p_i. We define two estimating criteria for checking whether approximating curve f satisfies a tolerance criterion. More details on these new estimating criteria can be found in [24].

3.1 Definition of norms N_∞ and N_2

It is possible to compare different B-spline curves using the norms

$$N_\infty(f) = Max\Big\{Max\{|Q_{ji}|; i = 0, ..., m\}; j = 1, ..., d\Big\} \quad (7)$$

$$N_2(f) = \frac{\sqrt{\sum_{i=0}^{m} \|Q_i\|^2}}{m+1} \quad (8)$$

We first consider original polygonal curve P as an interpolating B-spline curve g of degree one, defined on arbitrary knot vector $\tilde{T}$ with parameter values ζ_i [24]. With this hypothesis, g belongs to linear space $S_{2,\tilde{T}}$. Using norms N_∞ and N_2 on $f - g$ requires that B-spline curves $f \in S_{k,T}$ and $g \in S_{2,\tilde{T}}$ belong to the same linear space. The common degree is obtained through the degree elevation process [20]. The knot vectors are unified using subdivision algorithms (Boehm, Oslo, improved Oslo).

Let $f_{[i]}$ be the approximating spline segment of line segment $[p_i, p_{i+1}]$. The extreme points of $f_{[i]}$ are $f(\zeta_i)$, $f(\zeta_{i+1})$. When $N_\infty(f - g)$ is equal to tolerance ε, all $f_{[i]}$ curve segments are included in a band. Width L of the half-band is less than (or equal to) $\sqrt{2}\varepsilon$ (Figure 1 (a)).

For our fitting problem, if the value of $N_\infty(f-g)$ is large, either we have a bad correspondence between data p_i and their approximations $f(\zeta_i)$, or the control polygon is far from the initial line. Both situations are inappropriate. The first situation can be detected by computing $Sup_{i=0,...,n} \|f(\zeta_i) - p_i\|$.

More generally, if g is not a piecewise linear curve but an arbitrary B-spline curve, if $N_\infty(f - g)$ equals ε, f is included in a band centred around g (Figure 1 (b)). More than being a band criterion for data approximation, we have a band criterion for B-spline approximation.

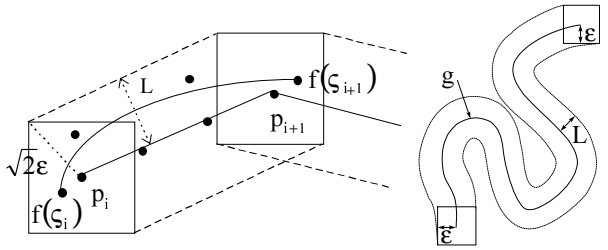

Figure 1: (a) Band criterion for data fitting (b) and B-spline fitting.
(• being control points)

If required, additional information concerning the average behaviour of curve f can be deduced from norm N_2.

3.2 Local estimation criteria

The goal of the second estimating tool is to obtain a local estimation approach independent of parameter values ζ_i.

We focus our attention on the definition of accurate measurements which guarantee that approximating B-spline curve f satisfies tolerance ε. If we call such a criterion d, a condition of validation is $d(P, f) \leq \varepsilon$.

New criterion d that we introduce is linked to geometric properties of B-spline f, and more precisely to its control points. They "roughly" represent the shape of the B-spline. The degree of accuracy between f and its control polygon depends on the number of control points in the representation. Using subdivision algorithms, the control polygon can be as closed as required of the corresponding curve.

The convex hull property can be used to predict the B-spline position according to the position of the control points. The definition of B-splines ensures local modelling on each interval $[t_i, t_{i+1}]$ $(t_i < t_{i+1})$: the B-spline curve position depends on k control points $Q_{i-k+1}, ..., Q_i$ (Figure 2).

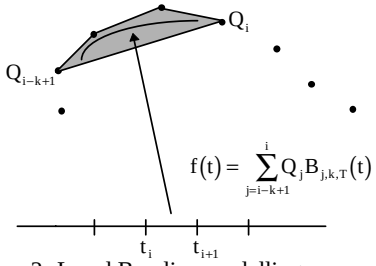

Figure 2: Local B-spline modelling: an example for an interval $[t_i, t_{i+1}]$.

Such a geometric formulation (Figure 2) does not yield sufficient accuracy. The B-spline curve on interval $[t_i, t_{i+1}]$ (or curve segment $[f(t_i), f(t_{i+1})]$) is included in the convex hull of control points $Q_{i-k+1}, ..., Q_i$. But it is a wide inclusion: curve segment $[f(t_i), f(t_{i+1})]$ is within the convex hull but is not strictly limited by it.

Our goal is to obtain a measurement between original curve P and its B-spline approximation f. An equivalent formulation is to have a measurement between each line segment $[p_i, p_{i+1}]$ and its approximating curve segment $[f(\zeta_i), f(\zeta_{i+1})]$. If the maximum of these measurements is within tolerance ε, we can claim that curve f satisfies the approximating problem.

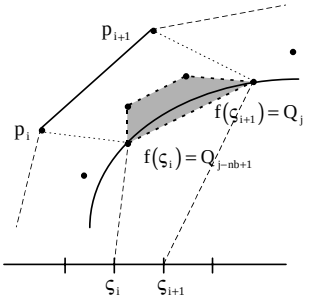

Figure 3: Convex hull of curve segment $[f(\zeta_i), f(\zeta_{i+1})]$.

Through this approach, we should obtain a control polygon whose convex hull only contains curve segment $[f(\zeta_i), f(\zeta_{i+1})]$. Such a polygon provides local estimation values for B-spline behaviour. This can be achieved by transforming each curve segment into its Bézier representation by applying subdivision algorithms, i.e. by inserting parameters $(\zeta_i)_{i=0}^{n}$ with $(k-1)$ multiplicity into knot vector T. The representation is illustrated in figure 3. We should note that it is not really a Bézier segment because there may be a knot t_j within $[\zeta_i, \zeta_{i+1}]$. The number of control points of this curve segment is nb $(nb \geq k)$.

This stage could be improved in order to obtain a more accurate estimation. Parameter value ζ_i is linked to approximating point $f(\zeta_i)$ of data p_i. Generally, $f(\zeta_i)$ does not correspond to the best approximation of data p_i [13]. We call $(\bar{\zeta}_i)_{i=0}^{n}$ optimal parameter values, whose value $f(\bar{\zeta}_i)$ is the closest approximation of data p_i. We can apply descent algorithms to move along curve f and reach these optimal parameters or apply Hoschek's technique. One should make sure of the numerical convergence of the descent method by ensuring that the new approximating point is closer to data p_i than the latter at each iteration. By replacing parameters ζ_i with parameters $\bar{\zeta}_i$ in the last stage, we obtain a more accurate estimation of data fitting on each line segment (Figure 4).

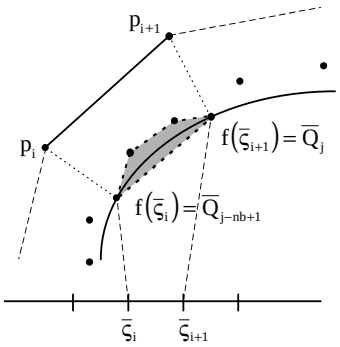

Figure 4: Convex hull of curve segment $[f(\bar{\zeta}_i), f(\bar{\zeta}_{i+1})]$.

In the last part, we apply the Hausdorff metric to polygonal curve $P^i = (p_i, p_{i+1})$ $(i = 0, ..., n-1)$ and corresponding $\bar{Q}^j = (\bar{Q}_{j-nb+1}, ..., \bar{Q}_j)$. If the maximum of the Hausdorff distance between both polygonal curves P^i and $\bar{Q}^j$ $(i = 0, ..., n-1)$ is equal to ε, then fitting B-spline curve f is at the most at distance ε from original polygonal curve P. Generally, this criterion is more accurate than the previous one defined by N_∞.

By studying the control points on each best curve segment, we can also obtain local information on the average behaviour of f.

4 DATA COMPRESSION

Let P be the polygonal curve defined by the original given vertices. The general problem of data compression is to define a curve f with a minimal number of parameters so that $d(P, f) \leq \varepsilon$ (d being a criterion for estimating data approximation). The number of parameters of f ought to be lower than the number of parameters in initial curve P. In practice, tolerance ε is often chosen so that there is no visual difference between P and f for the given representation scale.

In subsection 4.1, we present a brief overview of current data compression methods. Most of them are based on either the representation by means of a list of points or the spline representation. We introduce our data compression method using fitting B-spline curves in subsection 4.2.

4.1 Usual methods

4.1.1 Using representation by means of a list of points

The data compression problem using this representation is formulated so that the perpendicular distance of each point on curve P to the fitting line segment is within a pre-defined error tolerance. Many algorithms have been proposed in this direction. The goal of this paper is not to give a taxonomy of the different methods. Many authors tried to compare these data compression algorithms [3, 5, 15, 19]. The main conclusion is that there is no reference algorithm. The results often depend on the line morphology [4]. Nevertheless we can say that some methods aim at minimizing the number of line segments at the expense of time [2, 9, 14, 22, 27] to name a few, while others aim at minimizing time with less emphasis on the number of line segments [21, 23, 26, 28].

The drawbacks are linked to the broken line effects resulting from this representation (Figure 5). In practice, a solution often used consists of combining a simplification and a smoothing process (for example, a Douglas and Peucker algorithm [9] is followed by a cubic spline calculation). Cubic approximation seems to be very attractive for modelling complex shapes [1].

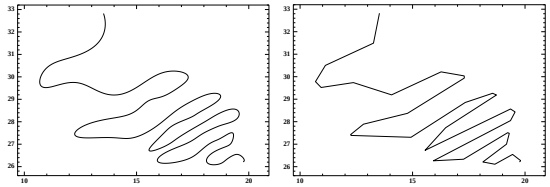

Figure 5: Jagged lines of the representation by means of a list of points (right): an example for mountain road modelling (left).

The advantage of using fitting B-splines as we suggest in the following subsection is to be able to treat both compression and smoothing. In addition, B-spline curves are able to produce more powerful processings through:

- parametric curves: possibility of combining the curve with the trajectory of an object,
- continuous curves: possibility of computing continuous curvatures and tangents,
- B-spline properties: possibility of having curve displacements through the modification of control points.

4.1.2 Using spline representation

The famous methods based on splines concerning data compression are knot removal strategies.

The purpose of these strategies is to reduce the number of knots in a given spline without perturbing the spline more than given tolerance ε. Such reduction means that we approximate the given spline in a space S, by a spline in a subset of S. In other words the number of degrees of freedom is reduced and we obtain data compression.

Some knot removal techniques have been already published. Lafranche and Le Méhauté propose an approach using a Bézier approximation of a function in $\mathbb{R}^2$ [16], while Lyche and Morken [18] as well as Eck and Hadenfeld [10] consider the problem using a B-spline representation.

Lyche and Morken's strategy

Without going into details, we can summarize Lyche and Morken's knot removal strategy. The inputs are:

- an original curve P to compress,
- a tolerance ε,
- an interpolating B-spline f of P defined on a knot vector T (several choices could be made for T, as described in subsection 2.2).

The problem is to build an approximating B-spline g, defined on a knot vector τ, which is a subset of T. τ is built with a minimal number of knots so that the difference between g and f is less than (or equal to) ε (i.e. $d(f, g) \leq \varepsilon$ where d could be the criterion introduced in subsection 3.1).

The strategy can be broken down into three main stages:

- First, we have to assign a weight ω_j to each inner knot t_j. For this we define an approximating B-spline g^i defined on T in which we remove inner knot t_i, and compute the difference with f. Weight $\omega_i = d(f, g^i)$ quantifies the significance of t_i in the representation of f.
- The second stage selects the knots to be removed on the principle that a knot can be removed if and only if its weight is less than (or equal to) the tolerance. Here we must take into account the vicinity constraint when close knots should be removed together.
- The last stage is reconstructing part of approximating curve g. It is a reconstructing step because the control points of approximating curve g are defined using the control points of f.

Eck and Hadenfeld's strategy

Eck and Hadenfeld's strategy [10] is a complementary method; it is based on the same principle. The inputs are the same. There is still an interpolating spline f and the goal is still to find a knot vector with minimal length. It is also a knot removal strategy. The three main stages are preserved:

- the weight computation part,
- the selection of knots to be removed,
- the reconstructing stage.

The difference is in the weight computation part. There are two methods for computing the control points of approximating B-spline curve g^i. We call them a "forward" computation of the control points which leads to B-spline curve g^i_I, and a "backward" computation which leads to B-spline curve g^i_{II}. These latters stem from the choice of the starting control points in the computing process: if we start computing the control points of g^i from the lowest subscript and increase it, we obtain the "forward" algorithm and conversely. The computation of the new control points corresponds to a reverse knot insertion process.

Generally, B-spline curves g^i_I and g^i_{II} are not the same. Obviously, the only exception occurs if a knot has been inserted artificially before or, in other words, if the continuity order at the respective knot is higher than it should be according to its multiplicity. Thus, the necessity of interpreting the knot removal process as an approximating process is manifest.

If we call A^I_j (respectively A^{II}_j) the control points of B-spline curve g^i_I (respectively g^i_{II}), Lyche and Morken determine g^i being the best approximation among these two B-spline curves. Eck and Hadenfeld define a new B-spline curve g^i whose each control point A_j is within a line segment whose extremities are the control points of the two methods (i.e. $A_j \in [A^I_j, A^{II}_j]$). A set of real numbers μ_j is introduced for geometric reasons. Control points A_j (points marked with • in figure 6) split the line segments from A^I_j to A^{II}_j in the ratio $\mu_j : (1 - \mu_j)$.

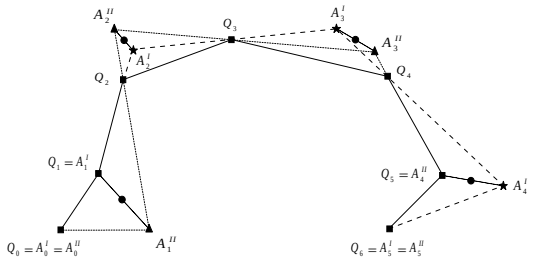

Figure 6: An example for the general construction of knot removal (k=4, $\mu_j = 1/2$).

4.2 Fitting B-spline technique

Fitting B-spline curves are suitable for data compression. Data usually come from a digitizing process. This leads to digitizing errors. We assume that these are removed by a "cleaning" process. "Cleaning" involves the removal of spurious elements such as peaks, loops, duplicates and other redundant data. Nevertheless, noise cannot be totally removed, requiring application of fitting techniques.

The quality of the approximation depends on:

- order k,
- knot vector T,
- parameter values $(\zeta_i)_{i=0}^n$.

We explained possible choices in section 2. We are going to discuss the particular choices for data compression. Data p_i is not totally independent of its neighbours p_{i-1} and p_{i+1}. This is the reason why we have to find specific parameters to adjust approximation.

We carried out a preliminary study on the influence of different parameterizations and knot vectors on the quality of the approximating curve [24]. The relationships between knot vectors and parameterizations for a large set of data have been studied. We briefly summarize the results.

Order k is an important parameter not for compression but for approximation. An order 4 is commonly used because a degree 3 generates inflection points. Thus, it is able to obtain close representations of complex shapes with only one spline segment. This order is also the lowest for having a continuous curvature along the curve (see subsection 2.4).

We now have to define knot vector T. Equation 3 provides good distribution between parameters $(\zeta_i)_{i=0}^n$ and knots $(t_i)_{i=0}^{m+k}$. This formula is generally used when data distribution is not uniform, in other words when we have an irregular density of points. A uniform knot vector is generally a poor solution in this case. We do not have such a situation. Rather than using equation 3 we advise the use of a uniform knot vector. With redundant information, non-correspondence between approximating parameters and knots is reduced. Even if the approximation is slightly less close to data p_i, a unified distribution generally gives good approximating results. The main advantage for data compression is that a uniform vector needs not be stored.

As regards parameter values, we suggest using Hoschek's intrinsic parameterization [13] starting from a centripetal one (see subsection 2.3). The oscillating phenomenon of Hoschek's parameterization which can occur between the given points when the distance between them is large, is non-existent here due to the amount of information.

A natural approach of compression is to determine the minimum number of control points so that the corresponding B-spline approximation yields an error smaller than (or equal to) the given tolerance. A reasonable assumption is that the error in the approximating process increases as the number of control points decreases. If we start by letting $(n+1)$ be the number of control points of f (the interpolating curve is assumed to be the curve of reference), the minimum number of control points can then be determined using a bisection method [25]. It may happen that a high number of points yields an initial system of non-maximum rank. In such a case, the initial number of control points is chosen slightly smaller than $(n+1)$.

5 RESULTS

The fitting strategy described in subsection 4.2 has been implemented and extensively tested. In this section, we present some significant tests and statistical summaries. The inputs of the algorithm are a curve P with a corresponding list of coordinates and tolerance ε. The output is a B-spline curve f with corresponding knot sequence T and control points $(Q_i)_{i=0}^m$ so that $d(f, P) \leq \varepsilon$. d being the new estimating criterion introduced in subsection 3.2.

The digitized curves P come from a cartography institute. They have been chosen for the diversity of their number of points and the diversity of their shapes.

In order to validate our method, the two following aspects are of importance:

- first we have to achieve high compression rates,
- in addition, our data compression algorithm should produce results with "reasonable" computation costs.

5.1 Compression rates

We have compared our fitting strategy with polygonal Douglas and Peucker's and Arge and Dæhlen's algorithms [9, 2]. The choice of these algorithms can be explained by first the interest in cartography (Douglas and Peucker) and the ability to obtain high compression rates (Arge and Dæhlen). Our strategy is also compared with knot removal strategies. It is impossible to present here all the results we obtained by applying these algorithms at many scales or tolerances. It is more difficult when the amount of initial lines is large. We present in this subsection significant samples of the results. We refer the reader to our research report [25] for more details.

Tolerance ε is first set so that there is no visual difference between the initial curve and its approximation at a fixed scale. ε=0.02mm (set by the institute) corresponds to the real case for the graphics we study in this subsection (Figures 7, 8 and 9).

The compression rates are computed using the following formulas (the data are assumed to be within a plane):

- $100 - \frac{100\times(m+1)}{(n+1)}$ for piecewise linear curves,
- $100 - \frac{100\times(2m+3)}{2(n+1)}$ for B-spline curves defined with a uniform knot vector,
- $100 - \frac{100\times(3m-k+6)}{2(n+1)}$ for B-spline curves defined with a knot vector from equation 3.

The second formula is obtained from the storage of order k (1 data), $(m+1)$ control points ($2(m+1)$ coordinates). The third needs to take knots t_i into account ($(m+k+1-2(k-1))$ data if we remove $(k-1)$ multiplicity of extreme knots t_0 and t_{m+k}).

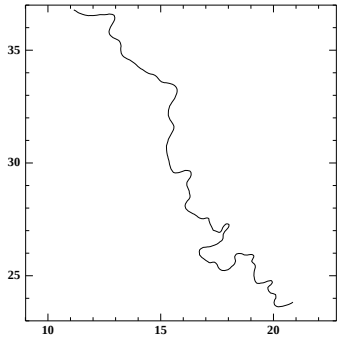

Figure 7: Initial isobathymetric line[1] (257 points).

Figure 7 depicts a section of an isobathymetric line around Brest's roadstead (France). Statistical results corresponding to this kind of lines are presented in table 1. Our fitting strategy is analysed using different knot vectors and parameterizations.

Method	Knots	Strategy	Compression
Polygonal		*Arge Dæhlen*	**68%**
		Douglas Peucker	*56%*
B-spline	*De Boor*	*Lyche Morken*	*13%*
		Eck Hadenfeld	*51%*
		Fitting (ε=0.5)	*22%*
	Uniform	*Fit.(Hos. ε=0.5)*	**61%**
		Fitting (ε=0.5)	*49%*

Table 1: Compression statistical results corresponding to a set of 20 isobathymetric lines (ε=0.02mm, k=4).

[1] line whose points correspond to the same value in depth

We can hope for higher compression rates by having higher tolerance (or lesser accuracy). On the contrary, higher accuracy yields less compact representations.

As Buttenfield indicates [4], the features of the initial line determine performance. The compression rate depends mainly on the tolerance and also on the intrinsic geometry of data. Using Buttenfield's guidelines, we have tested the method at different scales with complex and smooth lines.

Method	Knots	Strategy	$\diamond$	$\star$
Polygonal		*Arge Dæhlen*	**61%**	**74%**
		Douglas Peucker	**51%**	*65%*
B-spline	*De Boor*	*Lyche Morken*	*5%*	*24%*
		Eck Hadenfeld	*38%*	*58%*
		Fitting (ε=0.5)	*11%*	*34%*
	Uniform	*Fit.(Hos. ε=0.5)*	*49%*	**68%**
		Fitting (ε=0.5)	*40%*	*59%*

Table 2: Compression statistical results corresponding to a set of 20 complex coastlines ($\diamond$) and 20 smooth lines ($\star$) (ε=0.02mm, k=4).

Many curvature changes in the complex coastlines we compress (Figure 8) require more elements (control points) for modelling the shapes than expected for polygonal methods (Table 2). This naturally implies lower compression rates in comparison with table 1.

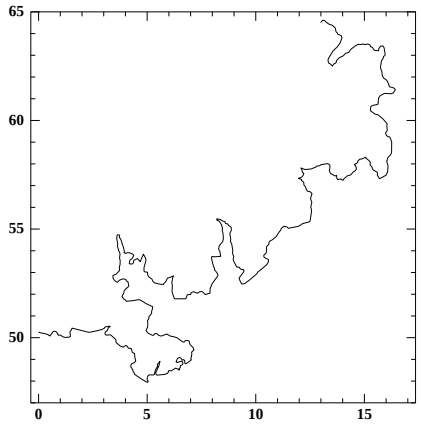

Figure 8: An example of coastline which has a complex geometry (486 points).

On the contrary, smooth shapes lead to higher data compression (Table 2). Figure 9 (mountain road) is a sample of the lines we study. The others come from waterway and railway digitalization. For modelling smooth curves, it may be better to use B-splines (Figure 11) rather than C^0 lines (Figure 10).

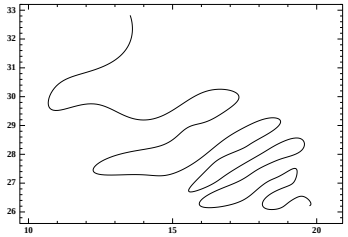

Figure 9: Initial mountain road (251 points).