LEAST SQUARES SPLINES WITH FREE KNOTS: GLOBAL OPTIMIZATION APPROACH

Gleb Beliakov

School of Computing and Mathematics,

Deakin University,

662 Blackburn Rd. Clayton, 3168

Tel. +61 3 9244 7532

Fax +61 3 9244 7460

Email: gleb@deakin.edu.au

Abstract

Splines with free knots have been extensively studied in regard to calculating the optimal knot positions. The dependence of the accuracy of approximation on the knot distribution is highly nonlinear, and optimisation techniques face a difficult problem of multiple local minima. The domain of the problem is a simplex, which adds to the complexity. We have applied a recently developed cutting angle method of deterministic global optimisation, which allows one to solve a wide class of optimisation problems on a simplex. The results of the cutting angle method are subsequently improved by local discrete gradient method. The resulting algorithm is sufficiently fast and guarantees that the global minimum has been reached. The results of numerical experiments are presented.

Key words: Least squares splines, regression splines, splines with free knots, global optimisation, cutting angle method.

1. Introduction

Least squares splines is a commonly used approach to function approximation, which provides certain advantages compared with smoothing splines, notably data reduction [8]. Spline knots are selected in advance, not necessarily at data points, and their number is usually less than data. Coefficients of the spline are computed by minimizing the least squares criterion, and since splines of a given order with fixed knots form a linear vector space, the problem is reduced to solution of a linear system of equations, typically by QR factorization. B-splines are commonly used as a basis.

The major drawback of selecting the knots in advance is that the quality of approximation critically depends on the good choice of knot positions. The simplest uniform mesh is frequently a bad choice, as evidenced by many examples [7,8,11]. To resolve this problem, several authors have proposed to allow flexibility in knot positions, and to optimize these positions with respect to the quality of approximation [7,8,11]. Splines with free knots have shown to perform significantly better than splines with fixed knots, however at a price. The resulting optimization problem is nonlinear and nonconvex, with many local minima and stationary points. Presence of stationary points on the boundary of the domain has been shown in [11]. In addition, because the knots form an increasing sequence, the domain of the optimization problem is a simplex, and hence the problem involves constraints. Penalty functions and logarithmic transformation of coordinates are used to transform it to the unconstrained problem, and local gradient-based techniques are employed for minimization [8,11,13,22,23].

The great number of local minima presents well-known difficulties in this problem of nonconvex optimization (cf. “lethargy” property [11]). The local optimization algorithms descend to the nearest minimum, and hence the quality of knot selection depends on the goodness of initial choice. Loach and Wathen [13] investigate linear splines with free knots and use coordinate descent as a less expensive alternative to Gauss-Newton method, employed in [11]. Schutze and Schwetlick [22,23] use damped Gauss-Newton method and Kaufman approximation to the Jacobian, thus avoiding the need for logarithmic coordinate transformation, although at each step of the Gauss-Newton method they have to solve a constrained linear problem. They also allow constraints on the derivatives of the spline [22]. Dierckx [8] employs Gauss-Newton method with a penalty function to handle the restrictions. Alternatives to nonlinear knots optimisation involve knot reduction [10, 14,21] and using knots optimized for piecewise polynomials (or deficient splines) [16]. The latter technique could be used to find a good initial approximation for Gauss-Newton method.

In this paper we explore recent techniques of global optimization, which do not suffer from the drawbacks of local methods. As its name implies, global optimization methods aim at finding the global minimum of the objective function. There are two broad categories: stochastic and deterministic approaches [9,24]. Stochastic techniques are more widely known: examples are random search, simulated annealing and genetic algorithms. We consider the other category: deterministic methods.

Deterministic global optimization is a difficult problem, shown to be NP-hard. Several techniques are available: cutting plane, branch and bound, DC programming, Lipschitz optimization, etc. [9,17,24]. All these methods suffer from increasing combinatorial complexity for greater number of variables, and their practical applicability is usually

limited to small size problems (from one to five variables). On the brighter side is the fact that they can determine whether the global minimum has been reached with a specified tolerance.

We consider a recently developed method of cutting angle [1,2,3,19,20]. This method constructs a saw-tooth cover of the objective function $f(\mathbf{x})$ – the max-min type auxiliary function that always underestimates $f(\mathbf{x})$. The maxima of the auxiliary function are taken at known values of $f(\mathbf{x})$. The nonlinear optimization problem is translated into a sequence of max-min type auxiliary problems, and the sequence of global minima of the auxiliary problems converges to the global minimum of $f(\mathbf{x})$.

We chose this method for two reasons. Firstly, it is formulated for optimization on a simplex, and hence is particularly suitable for knot optimization problem. Secondly, newly developed fast cutting angle algorithm allows one to handle higher dimensional problems in a reasonable time [5]. As our numerical experiments confirm, optimization of 5-7 internal knots is achieved within 2 minutes of computing time on an ordinary PC. The solution is further improved using the discrete gradient method [4].

The paper is structured as follows. The next section will formally define least squares splines and set the problem of knot optimization. Then we shall present the cutting angle method, especially its algorithmic aspects. Next we shall examine some test examples, and finally draw conclusions.

2. Least squares splines with free knots

Suppose, there is a given set of data points $\{(x_i, y_i)\}_{i=1}^{I}$ on the interval $[a,b]$, and a prescribed set of approximation knots $\{t_j\}_{j=-k}^{N+k+1}$, such that $t_0 = a$, $t_{N+1} = b$ and $t_{-k} \le ... \le t_0 < t_1 < ... < t_{N+1} \le ... \le t_{N+k+1}$. The position of the knots outside $[a,b]$ is arbitrary [8]. Let $N_j^{k+1}(x)$ denote normalized B-spline of order k+1 (degree k) with knots $t_j, ..., t_{j+k+1}$. The recursive relation for B-splines is well known [7,8]

$$N_j^{l+1}(x) = \frac{x - t_j}{t_{j+1} - t_j} N_j^l(x) + \frac{t_{j+l+1} - x}{t_{j+l+1} - t_{j+1}} N_{j+1}^l(x),$$

$$N_j^1(x) = \begin{cases} 1, \text{ if } x \in [t_j, t_{j+1}), \\ 0 \text{ otherwise.} \end{cases}$$

The least squares spline $S(x)$ is a piecewise polynomial of order k+1

$$S(x) = \sum_{j=-k}^{N} a_j N_j^{k+1}(x) \qquad (1)$$

which minimises the least squares criterion

$$\delta(a;t) = \sum_{i=1}^{I} \left(S(x_i) - y_i\right)^2 . \qquad (2)$$

When the knots $\{t_j\}$ are fixed, the problem of least square approximation is linear, and can be solved by standard methods, such as QR decomposition.

As we mentioned in the introduction, the quality of approximation depends on the adequate choice of knots, and in many cases wrongly selected knots result in unacceptable quality [7,8,11]. Therefore it has been proposed not to fix the knots, but allow some variability in their positions, and to optimise these positions with respect to the least squares criterion (2). Thus, $\delta(\mathbf{a};\mathbf{t})$ is minimised with respect to **a** and **t** simultaneously.

The problem can be split into linear and nonlinear parts [11], and each will be dealt with separately. The linear part (convex quadratic optimisation with respect to **a**) is solved by QR decomposition, as in the case of fixed knots. The nonlinear part takes the form

$$\min \delta(t),$$

$$\text{s.t. } t_i < t_{i+1}, i = 0,\ldots,N$$

$$\text{where } \delta(t) = \min_a \delta(a;t).$$

Local optimisation techniques have been previously employed to solve this nonlinear problem [8,11,13,22,23]. The derivatives with respect to **t** can be found explicitly, and fast Newton-type methods allow one to descend into the nearest local minimum. However, the problem is that $\delta(t)$ is nonconvex, and it possesses a great number of local minima and stationary points. Hence existing methods do not guarantee that the solution is the best knots distribution.

We explore the cutting angle method of global optimisation, described in the next section. First, however, we need to transform the domain of **t** to the unit simplex.

Since the knots $\{t_j\}_{j=0}^{N+1}$ form an increasing sequence (we are not interested in the knots outside [*a*,*b*], nor in multiple knots), and $t_0 = a$, $t_{N+1} = b$ are fixed, we need only N independent variables. Let us transform **t** to **p** using

$$p_i = \frac{t_i - t_{i-1}}{b-a}, i = 1,...,N+1.$$

It follows that $p_i > 0$ and $\sum_{i=1}^{N+1} p_i = 1$ and hence **p** belongs to the interior of the unit simplex.

On the other hand, from differentiability of $\delta(t)$ with respect to **t** (and hence **p**) follows that $\delta(t)$ is Lipschitz (see also a more general result in [15]). Hence we can apply the cutting angle method.

3. Cutting angle method

The idea of the cutting angle method is to construct a saw-tooth cover of the objective function – an auxiliary max-min type function whose maxima are taken at known values of the objective function $f(\mathbf{x})$. The saw-tooth cover always underestimates the objective function, and hence its minima are always below f. On the other hand, the sequence of minima of the auxiliary functions is increasing, and it converges to a global minimum of f [19]. The problem is translated into a sequence of auxiliary problems of minimization of the saw-tooth cover.

The cutting angle method is based on results in abstract convexity [19]. The cutting angle method arises, as do the Piyavskii and Mladineo methods [9,17,24], as a special case of the generalized cutting plane method described in [19]. One-dimensional Lipschitz global optimization algorithms and their multidimensional extensions, such as Piyavskii and

Mladineo methods, have been known for some time [9,17]. Cutting angle method is formulated in general multidimensional case, rather than being an extension of one-dimensional results. Formally it is described using so-called IPH functions.

Let $f(\mathbf{x})$ be a Lipschitz function defined on the unit simplex S. Consider the following problem of global optimisation

$$f(\mathbf{x}) \to \min \text{ subject to } \mathbf{x} \in S$$

It was shown in [1,19] that this problem can be reformulated as the global optimization problem of an IPH function over the unit simplex. IPH stands for Increasing Positively Homogeneous functions of degree one [19]. The class of IPH functions f defined on R_+^n is

$$\left\{ f : \forall \mathbf{x}, \mathbf{y} \in R_+^n \ \mathbf{x} \geq \mathbf{y} \text{ implies } f(\mathbf{x}) \geq f(\mathbf{y}); \forall \mathbf{x} \in R_+^n, \lambda \in R, \lambda > 0 : f(\lambda \mathbf{x}) = \lambda f(\mathbf{x}) \right\}.$$

Here, and in the remainder of this paper, vector inequality $\mathbf{x} \geq \mathbf{y}$ means dominance, i.e. $\forall i : x_i \geq y_i$. Similarly, $\mathbf{x} > \mathbf{y}$ means strict dominance, i.e. $\forall i : x_i > y_i$

Examples of IPH functions are:

1) $f(\mathbf{x}) = \mathbf{a}^t \mathbf{x}$, $a_i \geq 0$;

2) $f(\mathbf{x}) = \|\mathbf{x}\|_p$, $p > 0$;

3) $f(\mathbf{x}) = \sqrt{[\mathbf{Ax}, \mathbf{x}]}$, where $\mathbf{A}$ is a matrix with nonnegative entries;

4) $f(x) = \prod_{j \in J} x_j^{t_j}$, $J \subset I = \{1, \ldots, n\}, t_j > 0, \sum_{j \in J} t_j = 1.$

Although the cutting angle method has been formulated for IPH functions, every Lipschitz function can be transformed to a restriction of a certain IPH function to the unit simplex with the help of an additive constant [2,3,19].

Let $g : S \to \mathbf{R}$ be a Lipschitz function. Then $f(\mathbf{x}) = g(\mathbf{x}) + c$ is an IPH function on the unit simplex with

$$c \geq 2L - \min_{\mathbf{x} \in S} g(\mathbf{x})$$

where L is the least Lipschitz constant of g in the L_1-norm. Since adding a constant does not affect the location of the minima, we can effectively minimize any Lipschitz function using this transformation with an appropriate constant [19].

Thus, without loss of generality, we consider the problem of minimization of an IPH function $f(\mathbf{x})$ over n-dimensional unit simplex. Define the *support vectors*

$$\boldsymbol{l} = \left(\frac{f(\mathbf{x})}{\mathbf{x}}\right) = \left(\frac{f(\mathbf{x})}{x_1}, \frac{f(\mathbf{x})}{x_2}, \ldots, \frac{f(\mathbf{x})}{x_n}\right).$$

We will also use n unit vectors $\boldsymbol{e}^m = (0,\ldots,0,1,0,\ldots,0)$, with 1 in the m-th position, and we call the corresponding support vectors $\boldsymbol{l}^m = \left(\frac{f(\boldsymbol{e}^m)}{\boldsymbol{e}^m}\right)$, $m=1,\ldots,n$, *basis vectors*.

We consider a set of $K \geq n$ support vectors (and hence K known values of the function $f(\mathbf{x})$ at K distinct points), $\mathcal{K} = \{\boldsymbol{l}^k\}_{k=1}^K$. Let also the first n support vectors be the basis vectors (taken

at the vertices of the simplex). This choice of support vectors guarantees that the algorithm will locate all local (and hence global) minimizers of $f(\mathbf{x})$ in the interior of the unit simplex.

The auxiliary function

$$h_K(\mathbf{x}) = \max_{k \le K} \min_{i=1,..,n} l_i^k x_i$$

is the saw-tooth cover of $f(\mathbf{x})$. It always underestimates the value of $f(\mathbf{x})$, $h_K(\mathbf{x}) \le f(\mathbf{x})$.

Hence, $\lambda_K = \min_{\mathbf{x} \in S} h_K(\mathbf{x}) \le \min_{\mathbf{x} \in S} f(\mathbf{x})$. On the other hand, the sequence of its minima, $\{\lambda_K\}_{K=n}^{\infty}$ is increasing [19], and converges to the global minimum of $f(\mathbf{x})$.

We can formulate the cutting angle algorithm as follows [1,19].

Algorithm 1.

Step 0. (Initialisation)

a. Take points $\boldsymbol{e}^m, m = 1,...,n$, and construct basis vectors $\boldsymbol{l}^m = \left(\frac{f(\boldsymbol{e}^m)}{\boldsymbol{e}^m} \right), m = 1,...,n$.

b. Define the function $h_n(\mathbf{x}) = \max_{k \le n} \min_{i=1,...,n} l_i^k x_i = \max_{k \le n} l_k^k x_k$.

c. Set $K=n$.

Step 1. Find $\mathbf{x}^* = \arg\left[\min_{\mathbf{x} \in S} h_K(\mathbf{x}) \right]$.

Step 2. Set $K = K+1$ and $\mathbf{x}^K = \mathbf{x}^*$.

Step 3. Compute $\boldsymbol{l}^K = \left(\frac{f(\mathbf{x}^K)}{\mathbf{x}^K} \right)$. Define the function

$$h_K(\mathbf{x}) = \max_{k \le K} \min_{i=1,...,n} l_i^k x_i = \max\left\{ h_{K-1}(\mathbf{x}), \min_{i=1,...,n} l_i^K x_i \right\}.$$ Go to Step 1.

A more general version of this algorithm is known as the Φ-bundle method, and its convergence under very mild assumptions was proven in [18].

The crucial and most time consuming step of the Algorithm 1 is Step 1, minimization of the auxiliary function. This problem is essentially of combinatorial nature. Some properties of the auxiliary function (1) are studied in [2,3,19]. Among them we note the following.

Theorem 1 [2,19]

Let **x**>0 be a local minimizer of $h_K(\mathbf{x})$ over the relative interior of *S*, $ri\ S = \{\mathbf{x} \in S, \mathbf{x} > 0\}$. Then there exists a subset $L = \{l^{k_1}, l^{k_2}, ..., l^{k_n}\}$ of the set $\mathcal{K}$, such that

1) $\mathbf{x} = \left(\frac{d}{l_1^{k_1}}, \frac{d}{l_2^{k_2}}, ..., \frac{d}{l_n^{k_n}} \right)$ with $d = \left(\sum_i \frac{1}{l_i^{k_i}} \right)^{-1}$

2) $\max\limits_{k \le K} \min\limits_{i=1,...,n} \frac{l_i^k}{l_i^{k_i}} = 1$

3) Either $\forall i : k_i = i$, or $\exists m : k_m > n, l_i^{k_m} > l_i^{k_i} \forall i \neq m.$

We call the subset *L*, which satisfies conditions 2) and 3) of the Theorem 1, a *valid* combination of support vectors. The value of the auxiliary function at **x** is $h_K(\mathbf{x}) = d$.

In order to find the global minimum of the auxiliary function at Step 1 of the algorithm, we need to examine all its local minima, and hence all valid combinations of the support vectors. This process can be significantly accelerated (as reported in [2]) by noticing, that

$$h_K(\mathbf{x}) = \max\left\{h_{K-1}(\mathbf{x}), \min_{i=1,\dots,n} l_i^K x_i\right\}$$

Then, if we have already computed all the local minima of the auxiliary function $h_{K-1}(\mathbf{x})$ at the previous iteration, we only need to compute those minima that have been added by aggregation of the last support vector $\boldsymbol{l}^K$. This means that we need to examine only those combinations of support vectors that include vector $\boldsymbol{l}^K$ (i.e., one of $\boldsymbol{l}^{k_i} = \boldsymbol{l}^K$). The cutting angle algorithm of [2,19] works based on the above theorem, by examining all possible combinations of *n* support vectors (out of *K*).

Recently [5] it was established that valid combinations of support vectors are related to each other and can be seen as nodes of a directed graph with a single root. It is therefore possible to obtain all these combinations by considering only the nodes of the graph, and not all possible combinations of *n* vectors out of *K*. This has decreased the complexity of step 1 of the cutting angle algorithm from $O\left(\binom{K-1}{n-1}n^2\right)$ to $O(n^2|\mathcal{L}|)$, where $\binom{a}{b}$ denote binomial coefficients and $|\mathcal{L}|$ is the number of points that used to be local minima of $h_{K-1}(\mathbf{x})$ but are not local minima of $h_K(\mathbf{x})$ (indeed, with the addition of each new support vector $\boldsymbol{l}^K$ some of the minima of $h_{K-1}(\mathbf{x})$ disappear and new minima appear). Usually $|\mathcal{L}| \le n$. This improvement has allowed us to solve practical problems in 5-10 variables in very short time. The details of the new algorithm and its mathematical background are presented in [5].

Still, the cutting angle method has very slow convergence. To satisfactory solve problems with 5-10 variables, the number of iterations *K* must be of order of thousands. This is due to the fact that the problem itself is NP-hard. To obtain greater precision towards the end of the

algorithm, local descent techniques can be used. Newton-type methods would be the fastest, but the derivative free discrete gradient method [4] can also be applied. The next section illustrates the use of combination cutting angle – discrete gradient on well-known examples.

4. Examples

Example 1. Titanium heat data [7,8,11].

Titanium heat data is a classical test problem. Its difficulty for free knot spline approximation lies in the fact that the global minimum (for 5 internal knots, Fig.1) is deep and narrow. There are many suboptimal minima which despite similar error values do not provide quality approximation (Fig.2). Moreover, it looks like some suboptimal minima effectively lie in a valley (the first two internal knots can be moved without affecting the quality of approximation (Fig. 3)).

Local optimization methods give different results when started from different initial points (see [11]). Notably, when the initial point is the uniform partition of the interval, none of the methods described in [8,11,23] converges to the global minimum. The discrete gradient method alone, despite its ability to avoid shallow local minima, does not converge to the global minimum either. From other starting points it exhibits behavior similar to other methods.

Cutting angle method alone does converge to the global minimum (after 8000 iterations, which took 3092.0 sec on a Pentium III PC). This convergence is very slow to be used in practice. Combination cutting angle - discrete gradient gives a suitable alternative. A few

iterations of the cutting angle method were performed to adequately sample the search space. Then the discrete gradient method was started from the 10 best solutions provided by the cutting angle method. Only 20 iterations of the discrete gradient method were performed at this stage. Lastly, the best solution was improved using discrete gradient method with 1500 iterations.

Table 1 presents computational results. The error $\delta(\mathbf{a};\mathbf{t})$ is computed as

$$\delta(\mathbf{a};\mathbf{t}) = \left(\frac{1}{I-1} \sum_{i=1}^{I} w_i \left(y_i - S(x_i) \right)^2 \right)^{1/2}, \; w_1 = w_I = \tfrac{1}{2}, w_i = 1 \text{ otherwise.}$$

All global minima corresponding to the results in [11] have been found. Timing was done on a Pentium III 750 Mhz processor (Windows NT 4, 128 Mb of RAM, Visual C++ compiler). Plots of the resulting splines are given in [8]. Case of 5 internal knots is reproduced in Fig. 1-4. Notice that even though the spline with 24 equidistant knots (Fig.4) produces roughly the same error as the spline with 5 optimal knots, the approximation in the former case is worse.

Example 2. Pezzack's angular displacement data.

Dierckx [8] uses noisy version of Pezzack's data to fit quintic splines with free knots. We used modification of Pezzack's data by Lanshammar [12] (also noisy values). First we fitted cubic spline with free knots, and obtained results similar to Example 1 (Table 2). However, more iterations of the cutting angle method were required. Spline with 4 internal knots provides a reasonable estimate of the function and is reproduced on Fig.5.

The results for quintic splines are presented in Table 3. Case of 4 internal knots is presented on Fig. 6. In our view, this solution is superior to that presented in [8, p.73], since there are no oscillations in the rightmost part of the graph. In all the other cases (3,5,6 and 7 internal knots) the resulting knot positions are the same as in [8].

Constrained least squares splines are treated in the same way. Given monotonicity or convexity restrictions, at each iteration spline coefficients are found by solving a restricted least squares problem [6]. Since both cutting angle and discrete gradient methods require only the value of the error $\delta(\mathbf{a};\mathbf{t})$, they are completely independent of the properties of the spline and the algorithm used to compute the coefficients. The derivatives with respect to knot positions need not exist (the discrete gradient method uses a generalization of the gradient, so called subgradient, to treat nonsmooth functions).

Example 3. Approximation of $f(x) = \arctan(10x)$ on [-10,10] [22].

Following [22] we used 41 data points equidistant in [-10,10] with added pseudorandom noise uniformly distributed in [-0.075,0.075]. Unconstrained splines, even when the data contains no noise, produce undesirable oscillations, and to enforce monotonicity of the approximant monotonicity constraints have been imposed [6]. However, using monotone splines in this example requires an unacceptably large number of equidistant knots to achieve the accuracy consistent with the noise level (Table 4). The same accuracy was achieved using only 4 free internal knots. The resulting monotone spline with is depicted in Fig. 6. It was not possible to achieve any meaningful approximation with fewer knots.

5. Conclusion

Global optimization problem in several variables is notoriously difficult. If the function is Lipschitz, in principle it is possible to determine whether the global minimum has been reached (with a given tolerance) using deterministic techniques. The price is huge computational effort.

We have applied recently developed cutting angle method of global optimization to the problem of optimal knot placement for least squares splines. Cutting angle method is particularly suitable for this problem, because it works best on a simplex, which is exactly the domain of knot positions. This way we avoid introducing penalty functions. On the other hand, fast algorithmic implementation of the cutting angle method allows us to solve the problem in a reasonable time. As opposed to stochastic or local search methods, cutting angle will guarantee the best knot placement.

As several authors have pointed out, this big computational effort makes sense only if we need to approximate data with a limited number of knots [7,8]. Approximation as good, or even better, can be easily achieved by simply increasing the number of knots. Still, for a reasonably small number of knots (5-10), the problem can be solved on an ordinary PC in a matter of minutes. Whether this effort is justified, depends on the application of the resulting spline.

References

1. M. Andramonov, A. Rubinov and B. Glover, Cutting angle method in global optimization, *Appl. Math. Lett.* **12** (1999) 95-100.
2. A. Bagirov and A. Rubinov, Global minimization of increasing positively homogeneous function over the unit simplex, *Annals of Operations Res*., 2001, in press.
3. A. Bagirov and A. Rubinov, Modified versions of the cutting angle method, in: *Convex analysis and global optimization*, N. Hadjisavvas and P.M.Pardalos.(eds.), Kluwer, Dordrecht, in press.
4. A. Bagirov, Derivative-free methods for unconstrained nonsmooth optimization and its numerical analysis, *Journal Investigacao Operacional*, **19** (1999) 75-93.
5. L.M. Batten and G. Beliakov, Fast algorithm for the cutting angle method of global optimization, submitted to *J. of Global Optimization*, 2001.
6. G.Beliakov, Shape preserving approximation using least squares splines, *Approx. Theory and its Appl.*, **16** (2000), 80-98.
7. C. de Boor, A practical guide to splines, Springer, New-York, Heidelberg, 1978.
8. P. Dierckx, *Curve and surface fitting with splines*, Clarendon press, Oxford, 1995.
9. R. Horst, P.Pardalos and N.Thoai, *Introduction to global optimization*, Kluwer, Dordrecht, 1995.
10. Y. Hu, An algorithm for data reduction using splines with free knots, *IMA J. of Numer. Anal.* **13** (1993), 365-381.
11. D. Jupp, Approximation to data by splines with free knots, *SIAM J. Numer. Anal.* **15** (1978) 328-343.
12. H. Lanshammar, On practical evaluation of differentiation techniques for human gait analysis. *Journal of Biomechanics*, **5** (1982) 99-105

13. P.D. Loach and A.J. Wathen, On the best least squares approximation of continuous functions using linear splines with free knots, *IMA J. of Numer. Anal.* **11** (1991) 393-409.
14. T. Lyche and K. Morken, A data reduction strategy for splines with applications to the approximation of functions and data, *IMA J. of Numer. Anal.* **8** (1988), 185-208.
15. T. Lyche and K. Morken, The sensitivity of a spline function to perturbations of the knots, *BIT* **39** (1999), 305-322.
16. G. Meinardus, G. Nurnberger, M. Sommer and H. Strauss, Algorithms for piecewise polynomials and splines with free knots, *Math. of Comp*. **53** (1989), 235-247.
17. R. Mladineo, An algorithm for finding the global maximum of a multimodal, multivariate function, *Math. Progr*. **34** (1986) 188-200.
18. D. Pallachke and S. Rolewicz, *Foundations of mathematical optimization (Convex analysis with linearity)*, Kluwer, Dordrecht, 1997.
19. A. Rubinov, *Abstract convexity and global optimization*, Kluwer, Dordrecht, 2000
20. A. Rubinov and M. Andramonov, Lipschitz programming via increasing convex-along-rays functions, *Optimization Meth. and Software*, **10** (1999) 763-781.
21. L. Schumaker and S. Stanley, Shape-preserving knot removal, *Comp. Aided Geom. Des.* **13**, (1996) 851-872.
22. T. Schutze and H. Schwetlick, Constrained approximation by splines with free knots, BIT 37 (1997) 105-137.
23. H. Schwetlick and T. Schutze, Least squares approximation by splines with free knots, BIT 35 (1995) 361-384.
24. A.Torn and A.Zilinskas, *Global optimization*, Springer, Heidelberg, 1989.

Tables

Number of internal knots (n-1)	Number of iterations K	Error $\delta(\mathbf{a};\mathbf{t})$	Time (sec)	Number of equidistant knots required to achieve same $\delta(\mathbf{a};\mathbf{t})$
3	50	0.095	3.5	13
4	100	0.034	4	20
5	200	0.0123	5	24
6	500	0.0082	7	32
7	1000	0.0052	35	36

Table 1. Results of optimizing knot positions for the titanium heat data. After K iterations of the cutting angle method, the result was improved using discrete gradient method (1500 iterations). The fourth column gives the overall computing time.

Number of internal knots (n-1)	Number of iterations K	Error $\delta(\mathbf{a};\mathbf{t})$	Time (sec)	Number of equidistant knots required to achieve same $\delta(\mathbf{a};\mathbf{t})$
3	100	0.090	6	10
4	500	0.031	10	14
5	500	0.031	22	14
6	1000	0.0097	51	22
7	1500	0.0092	95	22

Table 2. Free knot cubic spline approximation to Pezzack's data.

Number of internal knots (n-1)	Number of iterations K	Error $\delta(\mathbf{a};\mathbf{t})$	Time (sec)	Number of equidistant knots required to achieve same $\delta(\mathbf{a};\mathbf{t})$
3	100	0.064	8	11
4	500	0.042	13	14
5	1000	0.026	42	15
6	1000	0.0098	51	19
7	1500	0.0077	105	23

Table 3. Free knot quintic spline approximation to Pezzack's data.

Number of internal knots (n-1)	Number of iterations K	Error $\delta(\mathbf{a};\mathbf{t})$	Time (sec)	Number of equidistant knots required to achieve same $\delta(\mathbf{a};\mathbf{t})$
4	200	0.057	3	33
5	200	0.057	4	33
6	200	0.056	7	33
7	200	0.056	8	33

Table 4. Free knot monotone cubic spline approximation of *arctan* noisy data.

Figures

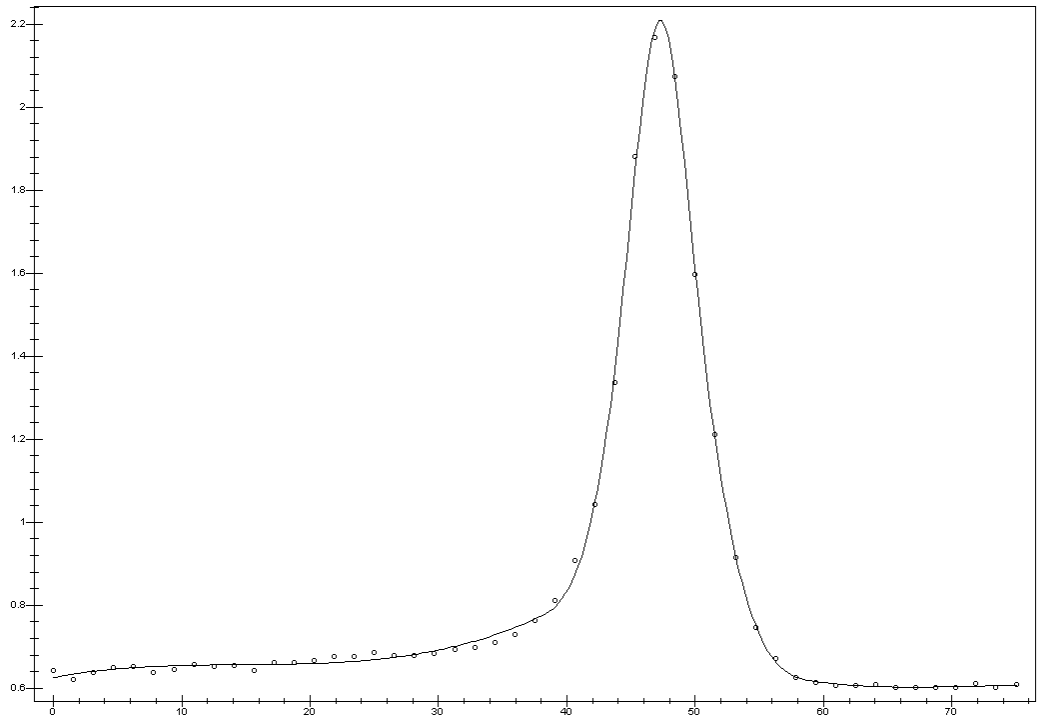

Figure 1. Cubic spline with optimal knots **t**=(37.65, 43.96, 47.36, 50.20, 59.20).

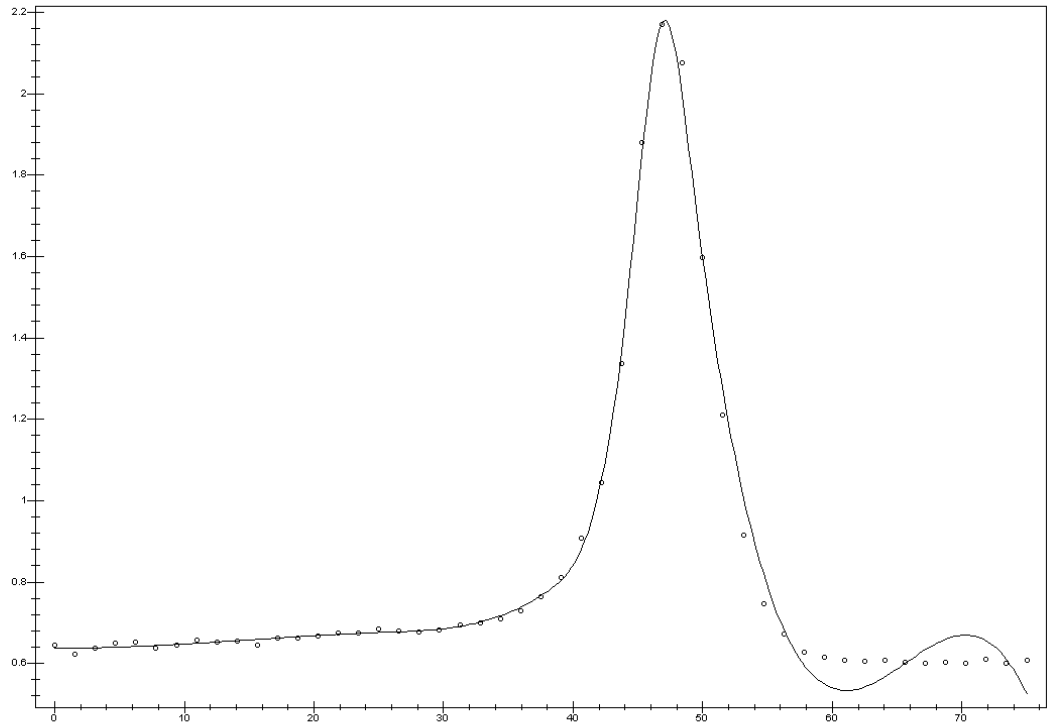

Figure 2. Cubic spline with suboptimal ($\delta(\mathbf{a};\mathbf{t})$=0.034) knots

$\mathbf{t}$=(25.04, 38.22, 44.07, 47.06, 49.25).

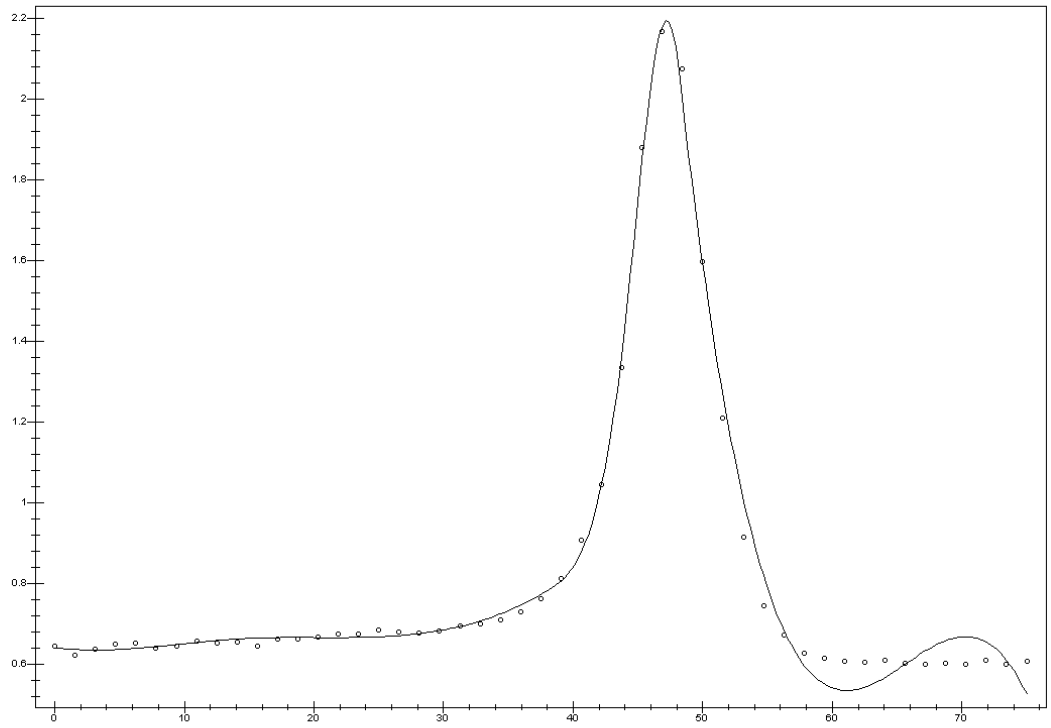

Figure 3. Cubic spline with suboptimal ($\delta(\mathbf{a};\mathbf{t})$=0.034) knots

t=(14.56, 38.20, 43.88, 47.83, 48.55). The position of the first knot is altered from that on Fig.2 without any decrease in the quality of approximation.

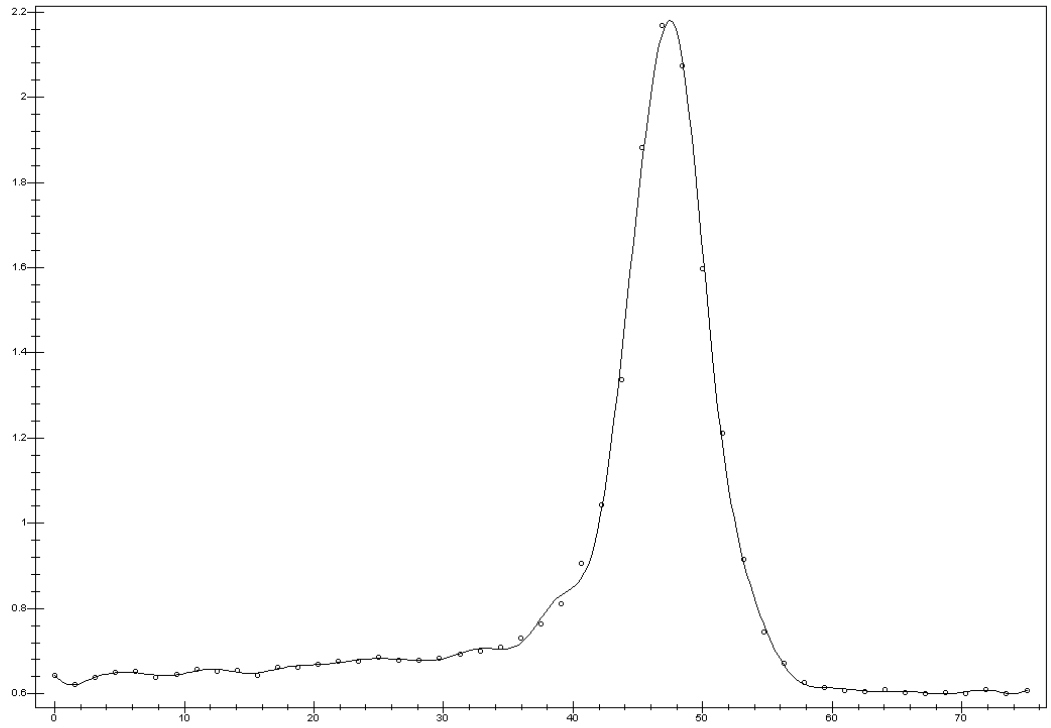

Figure 4. Cubic spline with 24 equidistant knots. The error of approximation $\delta(\mathbf{a})=0.0122$, but visually the quality is poor.

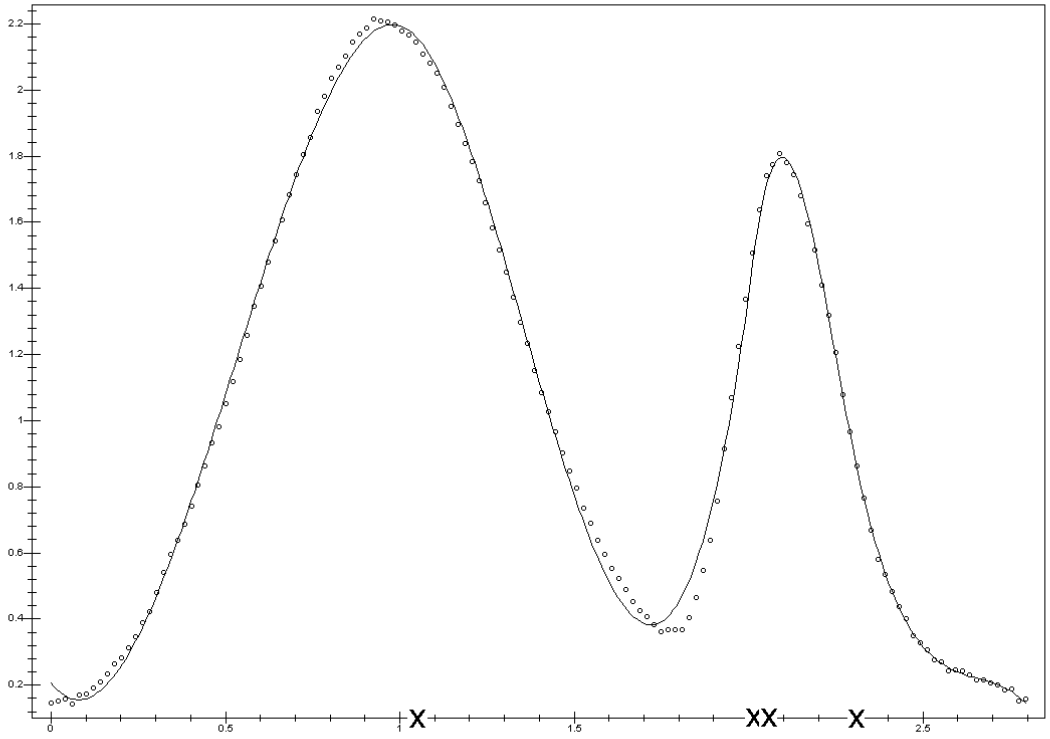

Figure 5. Optimal cubic spline approximation to noisy Pezzack's data [12] with 4 internal knots ($\delta(\mathbf{a};\mathbf{t})=0.031$).

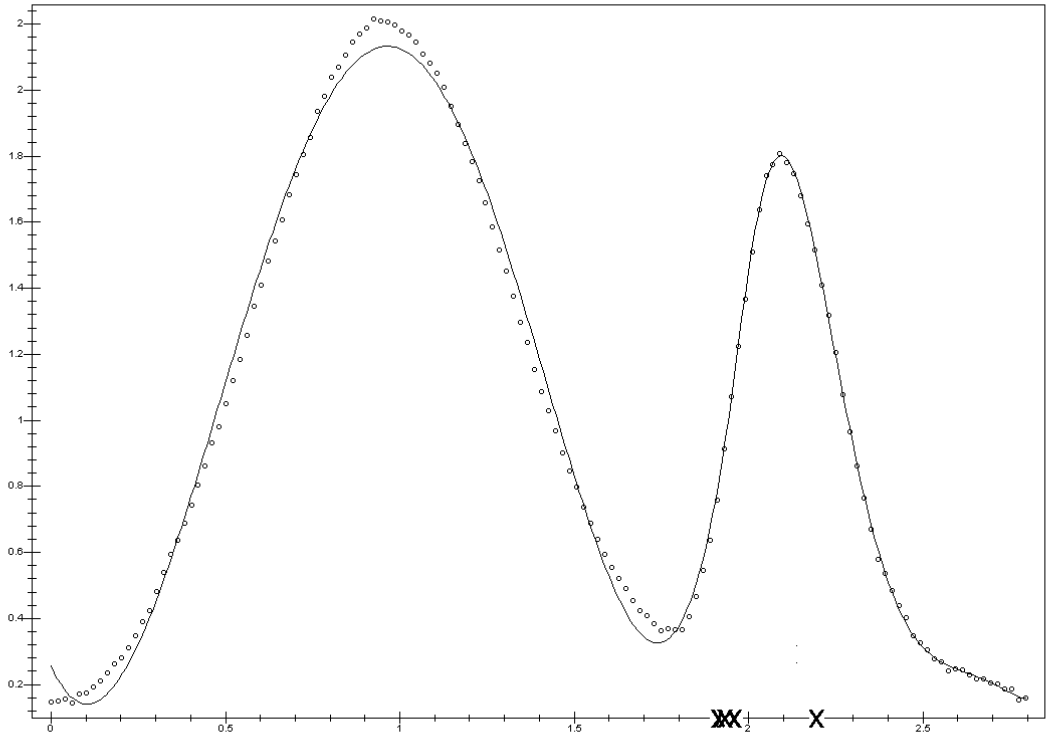

Figure 6. Optimal quintic spline approximation to noisy Pezzack data [12] with 4 internal knots ($\delta(\mathbf{a};\mathbf{t})$= 0.042). This knots placement $\mathbf{t}$=(1.92, 1.93, 1.98, 2.2) is superior to that of [8].

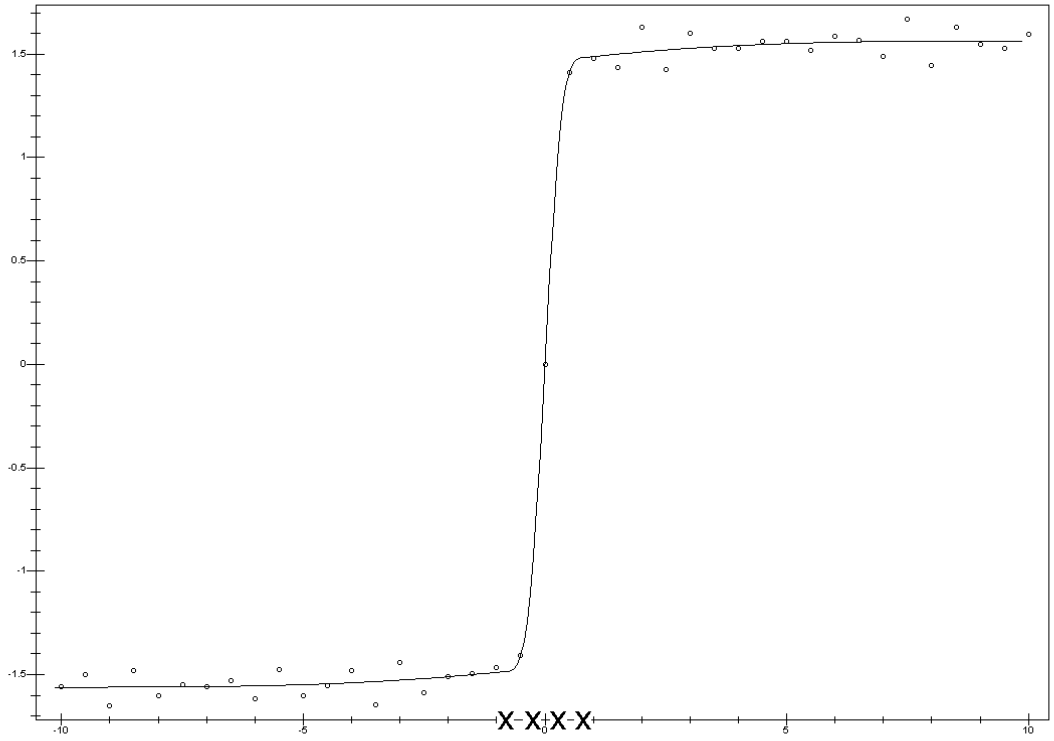

Figure 7. Optimal cubic spline approximation to noisy arctan data with 4 internal knots ($\delta(\mathbf{a};\mathbf{t})$=0.057).