



HAL
open science

Multivariate parametric regression under shape constraints

François Wahl, Thibault Espinasse

► **To cite this version:**

François Wahl, Thibault Espinasse. Multivariate parametric regression under shape constraints. 2017. hal-01262601v2

HAL Id: hal-01262601

<https://hal.science/hal-01262601v2>

Preprint submitted on 27 Nov 2017 (v2), last revised 23 Apr 2018 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multivariate parametric regression under shape constraints

François Wahl, Thibault Espinasse

November 27, 2017

Abstract

We consider a multivariate regression model built as a linear combination of functions of a single variable in univariate situations, or of product of univariate functions in multivariate cases. For each variable, the univariate functions form a Chebyshev system. The regression model is defined on a bounded domain and subject to one or more shape constraints on its definition domain, with the restriction that they can be transformed in positivity constraints for the regression function itself or for its derivatives. We develop an iterative procedure, where at each step the initial shape requirement is approximated by a set of linear constraints. This procedure is shown to converge to the optimal solution in the least square sense for univariate and then for multivariate cases. Numerical studies and a real industrial example with a multivariate polynomial regression subject to shape constraints of monotony illustrate the performance of the proposed method.

Keywords. multivariate parametric regression, monotony, shape constraints, quadratic programming, Chebyshev system, simplexes

1 Introduction

Our focus in this article is on multivariate parametric regression under shape constraints and especially on monotone polynomial regression on bounded sets of \mathbb{R}^V , for $V \geq 1$. Basically, polynomials are linear combinations of elementary functions, the monomials, resulting from the product of integer power law functions in each variable. In this paper, we keep this structure: the fitting function can be decomposed in a sum of products of elementary functions in each variable, but our methodology can deal with other elementary functions than monomials, and other constraints than as long as they expressions remain linear with respect to the coefficients.

More specifically, let $(X_i, Y_i)_{i=1, I}$ be a set of I observed points. Without loss of generality, the predictors X_i belong to $[0, 1]^V$, where V is the dimension of the input space. The observed responses Y_i are real.

We assume that (X_i, Y_i) are linked through an unknown function F_α from $[0, 1]^V$ to \mathbb{R} expressed as a linear combination of J known elementary functions f_j : $F_\alpha(x) = \alpha_0 + \sum_{j=1}^J \alpha_j f_j(x)$. The responses Y_i are subject to independent and identically distributed random errors ϵ_i with bounded variance. The model we are working on can then be written:

$$Y_i = F_\alpha(X_i) + \epsilon_i \quad (1)$$

The real coefficients, stored in the vector $\alpha = {}^t(\alpha_0, \alpha_1, \dots, \alpha_J)$, are to be found out.

Additionally F_α should respect shape constraints like monotonicity or convexity with respect to one or more variables, that will be detailed in the sequel. The least square problem to be solved can then be rephrased as

$$\underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^I (Y_i - F_\alpha(X_i))^2, \text{ s.t. shape constraints.} \quad (2)$$

The solution to (2) will be called the optimal solution.

The purpose of this paper is to transform these initial shape requirements in a finite number of linear constraints on the coefficients which approach the same solution as in the problem (2). The least square problem is thus transformed in a classical convex quadratic programming problem (see [16]).

Shape constraints have been investigated since mid 1990's in the field of 'Graphic Computer Aided Design', GCAD for short, and is a central theme in this area. Our purpose in this paper is to transpose some of these ideas to regression analysis.

The theory of shape constraints in GCAD is well developed in [6] and [18] for example, around Bernstein polynomials [12], Bézier curves and splines. In this field, the concern is to follow smoothly a set of control points given a priori, in order to represent adequately what the designer has in mind. As explained in this paper, in a certain sense, in regression analysis, the goal is the opposite and is to find the control points so that the regression

curve behaves as desired. Nonetheless this paper borrows some of the ideas of this field, specifically around simplexes and corner cutting or refinement algorithms [7], [2].

We keep the structure implemented in GCAD. In the univariate case, a common hypothesis in GCAD is that the set of functions $f_1(x), \dots, f_J(x)$ form a set of Chebyshev functions [10]. This will be one of our main hypothesis and will be detailed in the next section 1.2.

When $F_\alpha(x)$ is multivariate, one possible way in GCAD is to proceed through tensor products as in [7]. This is the reason why we assume that each $f_j(x)$ can be decomposed in a product of V univariate functions:

$$f_j(x) = f_{j_1}(x_1) \cdots f_{j_V}(x_V).$$

Each of the f_{j_v} ($v \in [1, V]$) maps $[0, 1]$ to \mathbb{R} , and is at least continuous and derivable on $[0, 1]$ as many times as needed, i.e., up to the order J . As in the univariate case, we impose that each set of functions $f_{1_v}(x_v), \dots, f_{J_v}(x_v)$ for $1 \leq v \leq V$ forms a Chebyshev system. This will be explained in section 3.

Typically, F_α is a polynomial. But the interest of using Chebyshev systems is that the framework can handle more general cases as well, as sum of power functions, or sum of exponentials for example (see subsection 1.2).

This paper is organized as follows: a state of the art is first developed as a beginning and then our general framework is introduced in the subsection 1.2. The theory is exposed for monotony constraints, first for univariate functions (section 2), before extending our ideas to more variables (section 3). Practical considerations are detailed in subsection 4.1. Simulations studies are then demonstrated with an example for a one variable polynomial bounded from above (subsection 4.2) and for a polynomial in 2 variables (subsection 4.3). Eventually, one industrial case is detailed in petroleum engineering related to hydrotreatment of naphtha (subsection 4.4). Conclusions and perspectives are given in section 5. Finally, we give in Appendix (section 5) the proofs of the results.

1.1 State of the art

Compared to non parametric regression, polynomials are immediate to calculate, involving only a few multiplications and additions. Most of all, they are easy to interpret, showing very clearly the influence of each variable, and their interactions.

Their disadvantage over nonparametric regressions is that they may lack of flexibility to represent particular function behaviors, like for example nearly flat regions followed by abrupt changes. However, compared to parametric regressions, final expressions in non parametric fitting are not easy to handle and the help of the computer is unavoidable. They are also not interpretable. This is why we believe as in [9], there is still room for parametric regressions and especially for polynomial regression.

In contrast to classical least square problems, constrained extensions are generally very hard to tackle, even for low degree polynomials. Indeed, most of the time it implies complicated non linear expressions of the coefficients. On the contrary, nonparametric regressions can adapt themselves very efficiently to constrain the behavior of the resulting function. They have received considerable attention for many years, first in one dimension and more

recently in multivariate situations. Restricting ourselves to isotonic regression in more than one dimension, a few performing algorithms have been proposed, based on splines [19] [17], on kernel type [5] [13] regressors, or on GAM [22].

Taking the derivatives, studies on monotone polynomials reduce to the study of positive polynomials. These latter can be positive over the entire real line, over a semi-infinite interval, or on a compact set. In these three situations, Karlin and Studden([10]) have given a representation theorem for positive polynomials in one variable of any degree. Still the obtained expressions remain highly non linear.

Ben-Tal and Nemirovski [1] have shown how to solve the problem via Semi-Definite Programming techniques in the three above situations. Hawkins [9] has set out a method and put his results into practice for positive polynomials in one variable on \mathbb{R} . Murray [15] has proposed a few improvements over this initial framework.

For more than one variable, the problem remains largely open. This is precisely the purpose of this paper to tackle the case of multivariable situations. In our methodology, we restrict the domain of definition to be a compact interval for one variable or a product of compact interval for more variables.

As already mentioned, even if the first motivation of this work is polynomial regression, it appears that the scope of our framework can be enlarged to deal with more general functions, as long as the basis functions form a Chebyshev system.

In non parametric methodologies, very often, the initial shape constraints are transformed in linear constraints on the coefficients of the regression. Once the constraints are set, the problem of minimizing the sum of squared errors under linear conditions reduces to a standard quadratic programming optimization procedure [16] and can be solved by classical algorithms available in usual mathematical softwares. In this paper, we keep this idea of expressing the constraints linearly.

The general idea is to solve the above minimization problem with a subset of the needed constraints and then allow this subset to grow iteratively. This method shares a few aspects with the method of sieves. Sieves designate a sequence of subset of functions A_k of some space A containing the function to be estimated, with the restriction that $\bigcup A_k$ should be dense in A . The difference in our method is that the space of functions is fixed, and the constraints change from one iteration to the other (see Section 2.2). The interested reader can find more details in [17], which concerns mainly monotonic splines.

Compared to non parametric methodologies, our method depends only on the number of elementary functions in the expression of F_α and not on the number of points. A marginal important benefit of our approach is that the expected behavior will be respected everywhere in the domain and not only in the vicinity of the observed points (see [14] for a short discussion on this topic). Finally, since no tuning parameters have to be estimated, the computational difficulty of the whole procedure is reduced compared to non parametric regression.

1.2 Notations and Definitions and Basic Notions

The indices $i \in [1, I]$ will refer to the observation points, $j \in [1, J]$ to the approximation functions, $v \in [1, V]$ to the variables. The upper case letters X or Y are reserved for the

observed points, while the lower case x or x_v for $v \in [1, V]$ is used for variables. Bold upper case letters like \mathbf{T} correspond to matrices, bold lower case letters to vectors.

Regression function. As already introduced, we consider $F_\alpha(x)$ a function from $[0, 1]^V$ to \mathbb{R} in the form of a linear combination of $J + 1$ known elementary functions f_j , with $f_0(x) = 1$.

$$F_\alpha(x) = \sum_{j=0}^J \alpha_j f_j(x) = \alpha_0 + \sum_{j=1}^J \alpha_j f_j(x),$$

where $\boldsymbol{\alpha} = {}^t(\alpha_0, \alpha_1, \dots, \alpha_J)$ is the vector of coefficients.

Each $f_j(x)$ is decomposed in a product of V univariate functions:

$$f_j(x) = f_{j,1}(x_1) \cdots f_{j,V}(x_V),$$

where $\forall v \in [1, V], x_v \in [0, 1] \mapsto f_{j,v}(x_v) \in \mathbb{R}$.

For all v , $f_{0,v}(x_v) = 1$. Each of the $f_{j,v}$ is at least continuous and derivable on $[0, 1]$ as many times as needed, i.e., up to the order J_v .

When F_α is a polynomial, each $f_j(x)$ is a monomial. In the univariate case, we have $f_j(x) = x^{d_j}$, for $j \in [0, J]$, and the resulting polynomial is of degree d_J . In the multivariate case, $f_j(x)$ takes the form $f_j(x) = x_1^{d_{j1}} x_2^{d_{j2}} \cdots x_V^{d_{jV}}$ where each of the d_{jv} for $j \in [1, J]$ and $v \in [1, V]$ are integers.

The notation

$$F_\alpha^{(k)}(x) \text{ or } f_j^{(k)}(x)$$

designates the derivative of order k ($k \geq 1$) of $F_\alpha(x)$ or $f_j(x)$ with respect to x in the univariate case.

Vectorial Notations. In univariate cases, $\mathbf{f}(x)$ refers to the the column vector

$$\mathbf{f}(x) = {}^t(f_1(x), \dots, f_J(x)).$$

We define also the derivatives

$$\mathbf{f}^{(k)}(x) = {}^t(f_1^{(k)}(x) \cdots f_J^{(k)}(x)).$$

$\mathbf{f}_\bullet(x)$ incorporates the constant term:

$$\mathbf{f}_\bullet(x) = {}^t(1, f_1(x), \dots, f_J(x)).$$

These notations are extended to multivariate cases as well.

Curve C_J . Alternatively, we consider the linear function defined by:

$$Z : [0, 1]^J \rightarrow \mathbb{R}$$

$$t = (t_1, \dots, t_J) \rightarrow Z(t) = \alpha_0 + \sum_{j=1}^J \alpha_j t_j.$$

The input space of Z will be denoted \mathbb{T} instead of $[0, 1]^J$. When $(t_1, \dots, t_J) = (f_1(x), \dots, f_J(x))$, Z describes a curve if $V=1$, a manifold of dimension V in multivariate situations in non degenerate cases, and of dimension smaller than V otherwise. This curve or manifold will be denoted C_J .

\mathbb{T} is an affine space. A point in this space is written T_{index} where the index is numeric or litteral. The notation \mathbf{T} designates the matrix of a collection of points in \mathbb{T} , e.g. the vertices of a simplex or a polytope.

Osculating simplex. In the remainder of this section, we restrict ourselves to the univariate case. As it is needed in the sequel we introduce the notion of osculating k -spaces and osculating hyperplanes which are special cases of the former.

Definition 1 *An osculating k -space at the point $T_x = (f_1(x), \dots, f_J(x))$ or more shortly at x is the affine space spanned by the first k independent vectors $\mathbf{f}^{(l)}(x)$ for $l = 1, k$, and passes by T_x .*

The osculating hyperplane to C_J at T_x is the osculating $J - 1$ -space at T_x .

In Computer Aided Design ([8]), Bézier curves connecting an initial point T_0 to a final point T_J in the affine space \mathbb{T} are integrally embedded in a simplex S_J whose vertices are its control points. This simplex is called 'osculating simplex' [18] and is defined as follows (see [7]):

Definition 2 *The osculating simplex between two points T_0 and T_J is the simplex for which the vertices are T_0, T_J and T_j for $0 < j < J$. The vertices $T_j, j = 1, \dots, J - 1$ are found as the intersections of the osculating j -space at T_0 and the osculating $(J - j)$ -space at T_J .*

With our choice of functions, $f_0(x) = 1$ and the osculating simplex is contained in an affine space of dimension J .

Chebyshev system. The study of Bézier curves is intimately linked to the theory of Chebyshev systems [7], [20], [10]. In the following our definitions are restricted to the interval $[0, 1]$, but it is not mandatory: any interval between an initial point a and final point b , open or closed would work.

The following determinant is denoted

$$M \begin{pmatrix} x_0, & \dots, & x_j \\ f_0, & \dots, & f_j \end{pmatrix} := \begin{vmatrix} f_0(x_0) & \dots & f_j(x_j) \\ f_0^{(1)}(x_0) & \dots & f_j^{(1)}(x_j) \\ \dots & \dots & \dots \\ f_0^{(j-1)}(x_0) & \dots & f_j^{(j-1)}(x_j) \end{vmatrix}.$$

Definition 3 *The functions f_0, f_1, \dots, f_J are called an extended Chebyshev system of class C^{J-1} on $[0, 1]$, or ET-system if they are J times differentiable on $[0, 1]$ and if*

$$\forall x_1 \leq \dots \leq x_J \in [0, 1], M \begin{pmatrix} x_0, & \dots, & x_J \\ f_0, & \dots, & f_J \end{pmatrix} > 0.$$

Definition 4 The functions f_0, \dots, f_J are called an extended complete Chebyshev system on $[0, 1]$, or ECT-system if

$$\forall j, 1 \leq j \leq J, \forall x_0 \leq \dots \leq x_j \in [0, 1], M \begin{pmatrix} x_0, & \dots, & x_j \\ f_0, & \dots, & f_j \end{pmatrix} > 0.$$

In Definition 4, when passing from an ET to an ECT, J is replaced by j and x_0 (resp. x_1, \dots, x_j) can be repeated at most j times.

A consequence of definition 3 and 4 which will be useful for the proofs in the appendix is that all the columns vectors of the determinant $M \begin{pmatrix} x_0, & \dots, & x_j \\ f_0, & \dots, & f_j \end{pmatrix} > 0$ are linearly independent.

Two important results on ET and ECT-systems are the following. The first one is based on the notion of multiplicity of the intersection of a curve and a hyperplane at x .

Definition 5 Let H be a hyperplane in dimension J containing the point (a_1, \dots, a_J) , and spanned by $J - 1$ vectors ${}^t(\alpha_{j,1}, \dots, \alpha_{j,J})_{j=1, J-1}$. Denoting

$$E(t_1, \dots, t_J) := \begin{vmatrix} t_1 - a_1 & \dots & t_J - a_J \\ \alpha_{1,1} & \dots & \alpha_{1,J} \\ \dots & \dots & \dots \\ \alpha_{J-1,1} & \dots & \alpha_{J-1,J} \end{vmatrix},$$

the equation of H is given by

$$E(t_1, \dots, t_J) = 0.$$

The multiplicity of the intersection at x of the hyperplane H and the curve $C_J = (f_1(x), \dots, f_J(x))$ is defined as the order of the first non vanishing derivative of the determinant $E(f_1(x), \dots, f_J(x))$.

If the curve C_J intersects the hyperplane H at x and if its tangent at x is not contained in H then the multiplicity is 1.

Theorem 1 (extracted from [10], chap I Corollary 4.1) Let $f_0(x) = 1$ for all $x \in [0, 1]$. If f_0, f_1, \dots, f_J is an ET-system on $[0, 1]$, then any hyperplane in \mathbb{T} intersects the curve C_J in \mathbb{T} at most J times counting multiplicities.

The second theorem (see [10], Theorem 1.1 Chap XI) gives a characterization of an ECT-system in terms of Wronskians. The Wronskian for $1 \leq j \leq J$ is defined as

$$W_{f_0, \dots, f_j}(x) \equiv \begin{vmatrix} f_0(x) & \dots & f_j(x) \\ f_0^{(1)}(x) & \dots & f_j^{(1)}(x) \\ \dots & \dots & \dots \\ f_0^{(j-1)}(x) & \dots & f_j^{(j-1)}(x) \end{vmatrix}.$$

This notation is a shortcut for $M \begin{pmatrix} x, & \dots, & x \\ f_0, & \dots, & f_j \end{pmatrix}$.

Theorem 2 (extracted from [10]) Let f_0, f_1, \dots, f_J be of class C^J on $[0, 1]$. Then f_0, \dots, f_J is an ECT-system on $[0, 1]$ if and only if for $j = 0, \dots, J$ we have $W_{f_0, \dots, f_j}(x) > 0$ on $[0, 1]$.

Examples. Theorem 2 gives a means to easily check that a set of functions is an ECT. With a sequence of increasing positive real d_j verifying $0 < d_1 < \dots < d_j$, the following $W_{f_1^{(1)}, \dots, f_J^{(1)}}(x)$ never vanish on the interval $[a, 1]$ with $0 < a < 1$ for the first example, and on the entire $[0, 1]$ for the other two:

1. $f_j(x) = x^{d_j}$,

$$W_{f_1^{(1)}, \dots, f_J^{(1)}}(x) = \prod_{j=1}^J d_j \prod_{1 \leq j < k} (d_k - d_j) x^{\sum_{j=1}^J d_j - J(J+1)/2}.$$

2. $f_j(x) = x^j$, i.e. the functions f_j form a sequence of monomials, the previous formula simplifies to

$$W_{f_1^{(1)}, \dots, f_J^{(1)}}(x) = \prod_{j=1}^J j!$$

3. $f_j(x) = \exp(d_j x)$,

$$W_{f_1^{(1)}, \dots, f_J^{(1)}}(x) = \prod_{j=1}^J d_j \prod_{1 \leq j < k} (d_k - d_j) \exp\left(\sum_{j=1}^J d_j x\right).$$

From now on, we require additionally that:

Assumption A The functions f_0, f_1, \dots, f_J form an ECT on $[0, 1]$.

Since $f_0(x) = 1$, a direct consequence of this assumption, is that the functions $f_1^{(1)}, \dots, f_J^{(1)}$ form also an ECT on $[0, 1]$.

Hypothesis A is very common in GCAD. In fact a stronger version, total positivity [7], is most of the times put into practice in this area, but this will not be necessary for our purpose.

Assumption A is requested because we expect that C_J will be included in its osculating simplex between any beginning point and any final point chosen in $[0, 1]$. This is detailed in Theorem 3 in section 2.

All the proofs are postponed to the appendix.

2 Univariate case

In this section, we detail the linear constraints approaching (2) in one dimension ($V = 1$) which is then transformed into Problem (3) below. We proceed as follows.

In subsection 2.1, through Proposition 1 we formalize our analysis. The conditions for which this proposition is true are examined in Theorem 3.

However, Proposition 1 proposes only a set of sufficient conditions for a function $F_\alpha(x)$ to be monotone. To go beyond this first step in subsection 2.2, still under Assumption A, we detail in Theorem 4 an algorithm which is guaranteed to find the optimal solution. A discussion of the refinement schema employed in the algorithm follows. We give a comparative example to Hawkin's methodology [9].

2.1 Univariate case: Osculating simplexes

The $J + 1$ vertices of the osculating simplex are gathered in a matrix \mathbf{T} of dimension $J \times (J + 1)$, where each column is a vertex. The expression ${}^t\mathbf{T}\boldsymbol{\alpha} \geq 0$ means that each coordinate of the vector ${}^t\mathbf{T}\boldsymbol{\alpha}$ is non negative.

Because of the linearity of $Z(t)$ and because a simplex is convex, we claim the following proposition.

Proposition 1 *Assume that the curve C_J is included in its osculating simplex on $[0, 1]$. If ${}^t\mathbf{T}\boldsymbol{\alpha} \geq 0$, then $\forall x \in [0, 1]$, we have $F_\alpha(x) \geq 0$.*

At this point, the problem we have to solve is transformed in **Problem (3)**:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_{i=1}^I (Y_i - F_\alpha(X_i))^2, \text{ s.t. } {}^t\mathbf{T}\boldsymbol{\alpha} \geq 0. \quad (3)$$

The purpose of the rest of this subsection is to make explicit the conditions under which a curve C_J between T_0 and T_J is included in its osculating simplex. To prepare the algorithm of section 2.2, we require this property to be true whatever the initial point T_0 and the final point T_J taken on the curve between $x = 0$ and $x = 1$.

Theorem 3 *Let T_0 and T_J be two points on the curve C_J . Under Assumption A, the portion of the curve between T_0 and T_J is included in its osculating simplex.*

Choosing the osculating simplex to enclose the curve is natural in Bézier curves theory. As a consequence of Lemma 2 in the Appendix, this simplex shrinks to a single point when T_J tends to T_0 .

2.2 Algorithm for finding the optimal solution, univariate case

As already mentioned, the conditions of Proposition 1 for finding a monotone polynomial or more generally a monotone function fitting the observed points $(X_i, Y_i)_{i=1, I}$ are only sufficient. In this subsection, we propose an algorithm capable of finding the optimal

solution in the least square sense as soon as the functions f_j verify the conditions of Theorem 3.

Our idea is a variation on a corner cutter or refinement algorithm. These algorithms are known since the mid seventies [2] [20] and closely linked to Bézier curves [8] and B-splines [4].

In this subsection, first, the corner cutting algorithm is introduced with a simple example for a degree 2 polynomial. It is then generalized to any function $f_j(x)$. In Theorem 4 the convergence of this algorithm is stated. This subsection is concluded with a few practical considerations.

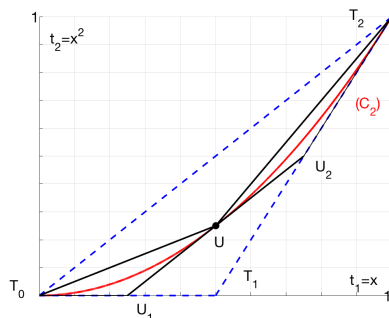


Figure 1: **corner cutting algorithm** the simplex $(T_0T_1T_2)$ is replaced by the polytope $(T_0U_1UU_2T_2)$, formed of two simplexes, (T_0U_1U) and (UU_2T_2) . The corner T_1 of the initial simplex is cut.

Example in dimension 2. For a short while, $F_\alpha(x)$ is assumed to be a polynomial of degree 2. In Section 2.1 we established that a sufficient condition for $F_\alpha(x)$ to be positive over $[0,1]$ is that the corresponding function $Z(t)$ be positive in the vertices T_0, T_1 and T_2 (see figure 1).

But we have restrained ourselves to simplexes. In fact it is easy to obtain a narrower convex polytope surrounding C_2 , if more than 3 vertices are allowed. For example, in Figure 1, the polytope P'_2 whose vertices are T_0, U_1, U_2, T_2 is included in the osculating simplex P_2 defined by the three vertices T_0, T_1, T_2 .

P'_2 is constructed by taking one of its sides on the tangent to the curve C_2 at $x = 1/2$ corresponding to the point U . After choosing the cutting point U , the two triangles (T_0, U_1, U) and (U, U_2, T_2) are uniquely determined. We obtain the polytope (P'_2) with vertices (T_0, U_1, U, U_2, T_2) .

This process of cutting can continue: each time we cut a simplex of the current polytope by a new tangent, we remove one corner and add two new vertices. What we are going to prove, is that when this step is repeated indefinitely, every point of curve C_2 can in this fashion be transformed in a vertex of a simplex and therefore in a constraint in the problem (3), so that the positivity of the polynomial is ensured everywhere on $[0,1]$.

Generalization. There is no difference if the degree J of the polynomial is greater than 2, or more generally in dimension J if the monomials are replaced by the functions $f_j(x)$. At

each step of the algorithm, the polytope surrounding the curve is composed of a succession of osculating simplexes, connected by one vertex located on the curve. Calling U this common vertex, the osculating hyperplane to the curve C_J at U is then the support of one face of the first connected simplex and of one face of the second one.

The whole process is only possible under the condition that the curve remains inside each of these osculating simplexes. This is a consequence of Assumption A and Theorem 3.

The convergence of the cutting algorithm is stated in the next theorem. We consider $P_{J,K}$ a set of nested simplexes, built so that $P_{J,K+1} \subset P_{J,K}$. For example, at step K , the initial vertex of each simplex of $P_{J,K}$ corresponds to $x = (k-1)/2^K$ and the final one to $k/2^K$ with k varying from 1 to 2^K . In the proof of 4, we need the following proposition.

Proposition 2 *Let A_J be the set of coefficients for which $\forall x \in [0, 1], F_\alpha(x) \geq 0$:*

$$A_J = \{\alpha \mid \forall x \in [0, 1], F_\alpha(x) \geq 0\}.$$

Similarly, we denote $A_{J,K}$ the set of possible coefficients at step K , that is the coefficients for which ${}^t\mathbf{T}\alpha \geq 0$ where \mathbf{T} is the matrix gathering (in columns) the vertices of $P_{J,K}$. Let $Cost_{J,K}$ be:

$$Cost_{J,K} = \min_{\alpha} \sum_{i=1}^I (Y_i - F_\alpha(X_i))^2, \text{ s.t. } {}^tT\alpha \geq 0.$$

Then,

1. $\forall K, A_{J,K} \subset A_{J,K+1} \subset A_J$.
2. A_J and all the $A_{J,K}$ are closed convex cones.
3. $\mathring{A} = \{\alpha \mid \forall x \in [0, 1], F_\alpha(x) > 0\}$.
4. *The sequence of $Cost_{J,K}$ is decreasing with K .*

The fact that the value $Cost_{J,K}$ of the cost function corresponding to the solution of Problem (3) at step K is decreasing with K gives an easy stopping criterium for the algorithm 1 below.

We define $\tilde{\alpha}_{J,K}$ the vector of coefficients of the solution of Problem (3) when the constraints match the vertices of $P_{J,K}$. The coefficients of the optimal solution to (2) are stored in a vector denoted $\tilde{\alpha}_J$.

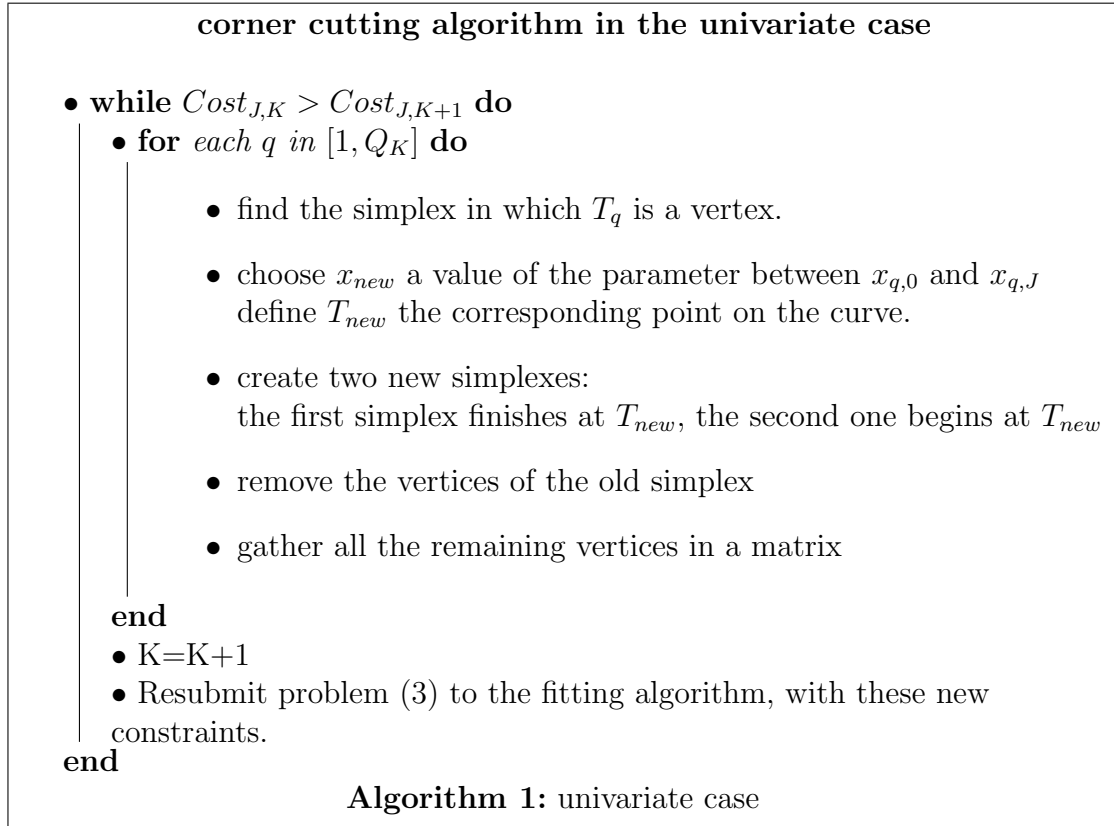
Theorem 4 *Under Assumption A, we have*

$$\lim_{K \rightarrow \infty} \tilde{\alpha}_{J,K} = \tilde{\alpha}_J.$$

The proof consists of proving that $\overline{\bigcup_{K \rightarrow \infty} A_{J,K}}$ is dense in A_J . This is an analog of sieves methods [17].

Algorithm 1. The algorithm which puts Theorem 4 into practice is presented below. As already said, at step K , the problem is solved by means of a quadratic programming algorithm. If the solution is not strictly inside the convex constrained region $A_{J,K}$ (see Proposition 2), then it is located on one constraint or on the intersection of two or more constraints. In this case, the constraints are said to be active. The active constraints indicate which region of the parameter values should be refined in the next step, since there is a one to one correspondence between the constraints, the vertices and the values of the parameter.

The set of active constraints at step K is numbered from 1 to Q_K . Each constraint $q \in [1, Q_K]$ matches a vertex T_q of one of the simplexes following the curve C_J . Let $x_{q,0}$, $x_{q,J}$ be the values of the parameter corresponding to the initial and final points of the simplex containing T_q , i.e. the two vertices of this simplex which are on the curve.



Stopping the procedure. To count the number of constraints added each time we cut a corner, we notice that at each step, we replace the old simplex by two new simplexes, which have a vertex in common. The number of vertices is finally augmented by $2 \times (J + 1) - (J + 1) - 1 = J$ at each step.

Practically, no one can afford to add an infinite number of constraints. If no improvement is seen after removing a corner and replacing it by J new vertices, or if the improvement is too small, the algorithm should stop.

5 degree polynomial and Hawkin’s values

	lower	estimated	upper	Hawkin
β_5	6.0874	11.3317	16.3324	10.99
β_4	-22.9273	-21.4133	-19.6327	-21.42
β_3	0.8264	6.8498	12.9451	7.29
β_2	20.6942	22.1779	23.3518	22.18
β_1	7.1634	8.7006	10.2383	8.59
β_0	0.6619	0.9910	1.3355	0.99

4 degree polynomial

	lower	estimated	upper
β_4	-22.664	-21.546	-19.346
β_3	17.455	19.294	19.768
β_2	20.675	22.369	23.302
β_1	5.4395	6.2205	6.9578
β_0	0.37451	0.95338	1.0414

Table 1: **estimation and confidence bands** for the coefficients of a polynomial of degree 5 fitted on Hawkin’s data on the left, and for a polynomial of degree 4 on the right. The column Hawkin gives the values estimated by Hawkin for the 5 degree polynomial.

Calculating the vertices of the osculating simplex. In the core of the algorithm, the determination of the vertices of the osculating simplex between two points T_0 and T_J on the curve taken at locations x_0 and x_J respectively is often needed. This is detailed in Lemma 1 in the Appendix, as a preliminary to Theorem 3.

Hawkin’s example. In Figure 2, we illustrate our approach with the simulation data proposed by Hawkins [9]. In this example, 50 points are drawn from the equation $y = 4x(x - 2)^2(x + 0.5)^2(x^2 + 2) + \epsilon$ with $\epsilon \sim N(0, 1)$. Neither the true underlying function is monotone on its definition domain, nor is the unconstrained least square fit with the points given by Hawkins.

In Hawkin’s methodology, the fit is over the entire real line \mathbb{R} and even degree polynomial are not permitted. We present two simulations studies, the first one with a polynomial of degree 5 in order to make comparisons with Hawkin’s results, and the second one with a polynomial of degree 4. The equation of the obtained fit is given in Table 1.

These simulations have been repeated a thousand times to give an idea of the distributions of the estimators. In Table 1 the columns ‘lower’ and ‘upper’ give the 5% and 95% percentiles.

2.3 Optimization of the split point, univariate case

So far, we have not discussed the location of the split point in Algorithm 1 when we create two new simplexes out of one. When invalidating a corner a first natural idea in Algorithm 1 is to create a new vertex on the curve for the same value of the parameter as the vertex taken out: if we remove T_k corresponding to x_k , then the coordinates of the new vertex are $(f_1(x_k), \dots, f_J(x_k))$.

However, with some extra computational work, it is possible to find the location on the curve where the volume of the initial simplex is the most reduced.

Proposition 3 *Let T_0, T, T_J be three points on the curve corresponding to $x_0 < x < x_J$. Then the function $V_{new} = V(x_0, x) + V(x, x_J)$ has a unique minimum between x_0 and x_J ,*

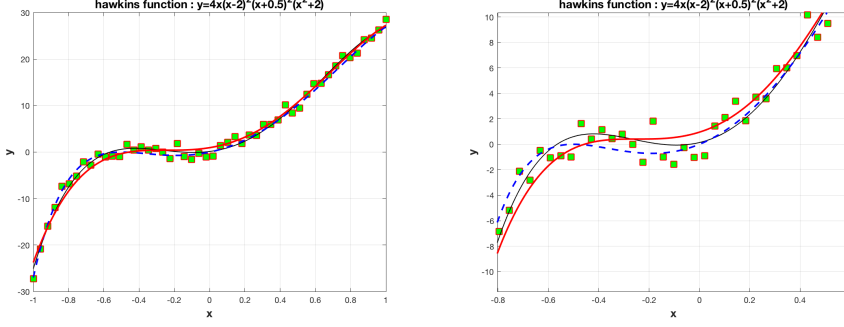


Figure 2: **Hawkins's function** In squared green, the observed points. In red, the fit. In black, the true function. In dashed blue, the least square approximation with a polynomial of degree 4. The right panel shows the resulting function on a restricted interval

where $V(x_0, x)$ (resp. $V(x, x_J)$) stands for the volume of the simplex between x_0 and x (resp. x and x_J).

This way of cutting leads to a variant of the initial Algorithm 1, where we look for the optimal cut. The drawback of this approach is that finding the minimum of V_{new} is computationally costly since calculating a volume involves the evaluation of $2J - 1$ determinants as explained in Proposition 4 below.

We need here to introduce the determinants D_j and $D_{j,j}$:

$$D_j = \left| \mathbf{f}^{(1)}(x_0) \quad \dots \quad \mathbf{f}^{(j)}(x_0) \quad \mathbf{f}^{(1)}(x_J) \quad \dots \quad \mathbf{f}^{(j-j)}(x_J) \right|.$$

$D_{j,j}$ is obtained by replacing the j -th column of D_j by $\mathbf{f}(x_J) - \mathbf{f}(x_0)$.

$$D_{j,j} = \left| \mathbf{f}^{(1)}(x_0) \quad \dots \quad \mathbf{f}^{(j-1)}(x_0) \quad \mathbf{f}(x_J) - \mathbf{f}(x_0) \quad \mathbf{f}^{(1)}(x_J) \quad \dots \quad \mathbf{f}^{(j-j)}(x_J) \right|.$$

Proposition 4 $V(x_0, x_J) = \frac{1}{J!} D_{J,J} \frac{\prod_{j=1}^{J-1} D_{j,j}}{\prod_{j=1}^{J-1} D_j}.$

Sequence of monomials. The optimization of the split point becomes extremely simple when the system of functions $f_j(x)$ is the traditional sequence of monomials: $\{x^j\}_{j=1}^J$. In this case, we prove now that the optimal parameter for the split point is $\frac{x_0 + x_J}{2}$.

As a consequence, an other way of splitting the curve in Algorithm 1 is to create a new vertex on the curve for a value of the parameter equal to $x = \frac{x_0 + x_J}{2}$, even if it is only fully justified for a sequence of monomials.

We start with the following proposition, where the symbol \propto means 'is proportional to'.

Proposition 5 $V(x_0, x_J) \propto (x_J - x_0)^{\frac{J(J+1)}{2}}.$

The final result of all these developments is that in the case of a sequence of monomials, the optimal cut point corresponds to $\frac{x_0 + x_J}{2}$ as stated in the next theorem.

Theorem 5 *Let the system of functions $f_j(x)$ be the sequence of monomials $\{x^j\}_{j=1}^J$. Then the optimal cut point between x_0 and x_J is $\frac{x_0 + x_J}{2}$.*

As a last comment, the complete procedure which involves calculating the new vertices after finding out the split point, is especially simple with the sequence of monomials $\{x^j\}_{j=1}^J$, since the vertices of the osculating simplex can be calculated analytically.

3 Multivariate case

In case of multivariate functions, we proceed in three successive steps. We begin with a simple example in order to introduce our approach. In a second step, we generalize the previous methodology of section 2 in one dimension to this new situation and conclude this subsection with Theorem 6, which transposes Proposition 1 to multivariate functions. As in the univariate case, the proposed constraints are only sufficient conditions. We then propose an algorithm in section 3.2 capable of finding the optimal solution.

3.1 Multivariate case: circumscribing simplexes

We switch to a more general situation, where $x = (x_1 \cdots x_V)$ is V -dimensional. We recall that $F_\alpha(x)$ is written as a linear combination of J functions

$$F_\alpha(x) = \alpha_0 + \sum_{j=1}^J \alpha_j f_j(x),$$

where each of the $f_j(x)$ is decomposed in a product of V univariate functions:

$$f_j(x) = f_{j_1,1}(x_1) \cdots f_{j_V,V}(x_V),$$

with j_1 (resp. j_V) taken in the set $[0, J_1]$ (resp. $[0, J_V]$), and $f_{0,v}(x_v) = 1$ for all v .

$F_\alpha(x)$ can be expressed as the scalar product

$$F_\alpha(x) = \langle \boldsymbol{\alpha}, \mathbf{f}_\bullet(x) \rangle$$

where $\mathbf{f}_\bullet(x) = (1, f_1(x), \cdots, f_J(x))$ gathers all the functional terms in $F_\alpha(x)$.

Restricting our attention to each variable x_v , the vector

$$\mathbf{f}_{v\bullet}(x_v) = (1, f_{v,1}(x_v), \cdots, f_{v,J_v}(x_v))$$

collects all the functions depending on x_v exclusively.

When x traverses its definition domain, the points $(f_1(x), \cdots, f_J(x))$ describe a manifold called C_J , which is of dimension V in non degenerate cases, and of dimension lesser than V otherwise.

Alternatively, we examine the linear function

$$Z(t) = \alpha_0 + \sum_{j=1}^J \alpha_j t_j,$$

which coincides with $F_\alpha(x)$ when $t_j = f_j(x)$.

Our problem is to determine the vector of coefficients α , so that $F_\alpha(x) \geq 0$ (or ≤ 0) in the entire domain. As in dimension 1, one way to solve this question is to enclose C_J in a convex polytope P_J and check the positivity of Z in every vertex of P_J . How to choose P_J will be explained very soon. Assuming that P_J is known and denoting T_j one of its vertices, verifying the positivity amounts to check that $Z(T_j) \geq 0$, for all $j \in [1, J]$. We bring together all the constraints in a matrix \mathbf{T} where each column corresponds to one of the T_j . The problem to solve in dimension V can be rephrased as

Problem (4)

$$\underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^I (Y_i - F_\alpha(X_i))^2, \text{ s.t. constraints } {}^t\alpha\mathbf{T} \geq 0. \quad (4)$$

which is the analog of Problem (3), the only difference being that X_i is now V -dimensional.

To extend the previous results from dimension 1 and control the number of constraints, we proceed by tensorization. Specifically, we assume that

$$\mathbf{f}_\bullet(x) = \mathbf{f}_{1\bullet}(x_1) \otimes \cdots \otimes \mathbf{f}_{V\bullet}(x_V). \quad (5)$$

This is formalized in Theorem 6 which gives a means to automatically generate the needed constraints. Theorem 6 joins together Proposition 1 and Theorem 3 and transposes their statement to multivariate situations. We suppose that each family of functions $\{f_{v,j_v}(x_v)\}_{j_v=0}^{J_v}$ verifies Assumption A for $1 \leq v \leq V$.

Let T_{v,j_v} for $j_v \in [0, J_v]$ be the vertices of the osculating simplex containing the curve $C_{J,v} = (f_{v,1}(x_v), \dots, f_{v,J_v}(x_v))$, for $x_v \in [0, 1]$. The matrix \mathbf{T}_v of dimension $J_v \times (J_v + 1)$ contains in columns the vertices T_{v,j_v} . The matrix \mathbf{T} is defined as the tensor product:

$$\mathbf{T} = \bigotimes_{v=1}^V \mathbf{T}_v.$$

Setting $J + 1 = \prod_{v=1}^V (J_v + 1)$, the dimension of \mathbf{T} is $J \times (J + 1)$.

In the space $\mathbb{T} = [0, 1]^J$, each column of \mathbf{T} corresponds to a vertex of a polytope P_J . This polytope is a simplex: the number of its vertices is $J + 1$. It contains the curve C_J as stated in the following theorem.

Theorem 6 *Under Assumption A*

1. C_J is included in P_J .
2. If ${}^t\alpha\mathbf{T} \geq 0$, then $\forall x \in [0, 1]^V$, we have $F(x) \geq 0$.

Dropping terms. Actually, a function $F(x)$ containing all the terms resulting from the tensor product $\mathbf{f}_{1\bullet}(x_1) \otimes \cdots \otimes \mathbf{f}_{V\bullet}(x_V)$ is of little practical use. If we can not drop some of these terms, these kind of functions will fail to match practical applications. For instance,

in real situations, cubic polynomials will not include necessarily all the interactions terms: it is very common to ignore interactions of more than two variables.

Dropping some terms amounts to take the corresponding coefficients (in the function $Z(t)$) equal to 0. As a result, in the matrix of constraints, the corresponding rows are merely deleted.

Number of constraints and optimality. Since for each variable, the number of vertices is $J_v + 1$, we derive the corollary 1.

Corollary 1 *The number of constraints in Theorem 6 is at most*

$$J + 1 = \prod_{v=1}^V (J_v + 1).$$

3.2 Algorithm for finding the optimal solution, multivariate case

In univariate cases, the proposed algorithm is based on the notion of osculating hyperplanes. In multivariate situations, we use instead the fact that the vertices of the polytope on which we request $F(x)$ to be positive result from the tensor product of V matrices. The columns of each of these matrices correspond to the vertices of a simplex for the matching variable. We note that the tensor product of all these matrices corresponds also to a polytope.

To extend Algorithm 2.2, in each dimension v ($v \in [1, V]$), instead of taking a simplex, we can consider a polytope with more than $J_v + 1$ vertices. This polytope is constructed by means of the algorithm exposed in the univariate case (see subsection 2.2). At step K , for each v , we can as in subsection 2.2 build a polytope $(P_{v,J_v,K})$, and for each v , gather all the vertices of the corresponding polytope in a matrix $\mathbf{T}_{v,K}$. We then generate the tensor products of all these matrices $\mathbf{T}_K = \otimes_{v=1,V} \mathbf{T}_{v,K}$ and obtain the coordinates of the vertices on which we must check the positivity of the corresponding function $Z(t_1, \dots, t_V)$.

Now, $C_J = (f_1(x_1, \dots, x_V), \dots, f_J(x_1, \dots, x_V))$ is a manifold of dimension V . When $K \rightarrow \infty$, all the points of C_J will become a vertex in \mathbf{T}_K . As previously in section 2.2 let us call $P_{J,K}$ the polytope whose vertices are the columns of \mathbf{T}_K , $F_K(x)$ the solution to the problem (4) when the constraints are formed by the vertices of $P_{J,K}$. The coefficients are collected in the vector $\tilde{\alpha}_{J,K}$. Analogously, we examine $F_J(x)$ the optimal solution to (2) and $\tilde{\alpha}_J$ its vector of constraints.

Theorem 7 *We assume that each family of functions $\{f_{v,j}(x_v)\}_{j=1}^{J_v}$ verifies Assumption A for $1 \leq v \leq V$. Then*

$$\lim_{K \rightarrow \infty} \tilde{\alpha}_{J,K} = \tilde{\alpha}_J.$$

The proof is similar to the previous one in Theorem 4 with the generalization to the tensorial product of constraints.

Algorithm. Practically, the algorithm for solving problem (4) proceeds iteratively. At each step, the vertices corresponding to the Q_K active constraints should be removed as in

a classical cutting algorithm and replaced by new ones. Since these vertices result from a tensor product, one constraint in the product refers to V constraints, one in each dimension. For each q in $1, Q_K$, we consider the vertex T_q in the product space corresponding to the constraint q . We have $T_q = \otimes_{v=1, V} T_{v,q}$. The simplex to which $T_{v,q}$ belongs begins at the value $x_{v,0}$ of the parameter x_v and finishes at x_{v,J_v} .

Each vertex $T_{v,q}$ can also be interpreted as a constraint for the variable v . The idea is then to proceed as if all the V constraints corresponding to the V vertices $T_{v,q}$ were active.

Corner cutting algorithm in the multivariate case

- **while** $Cost_{J,K} > Cost_{J,K+1}$ **do**
 - **for** each q in $[1, Q_K]$ **do**
 - for** each variable v in $[1, V]$ **do**
 - choose $x_{v,new}$ a value of the parameter in $[x_{v,0}, x_{v,J_v}]$
define $T_{v,new} = {}^t(f_{v,1}(x_{v,new}), \dots, f_{v,J_v}(x_{v,new}))$.
 - create two new simplexes:
the first simplex finishes at $T_{v,new}$,
the other one begins at $T_{v,new}$.
 - remove the vertices of the old simplex (containing $T_{v,q}$)
 - gather all the remaining vertices in a matrix
 - end**
 - end**
 - $K=K+1$
 - Form the new constraints by applying the tensor product to the new matrices;
 - Resubmit problem (4) to the fitting algorithm, with these new constraints;
- end**

Algorithm 2: multivariate case

Once again, if no improvement in the fitting criterium is seen after removing a vertex and replacing it by new ones, or if the improvement is too small, the algorithm should stop. Also, since the number of constraints will grow extremely fast in multivariate situations, the number of allowed constraints should not exceed a limit set by the user (one hundred thousand for example is possible on modern computers).

4 Examples

In this section we examine how to apply our theory. We begin by enumerating the situations where our method can be used (subsection 4.1). The case of univariate functions is illustrated in subsection 2.2 with Hawkin’s example. We add the case of a bounded function in 4.2. Next, we consider an example with a bivariate polynomial. We conclude this section with one industrial example (see 4.4).

All the figures and the examples are based on $\text{\textcircled{R}}$ Matlab routines, available upon request. These routines comprise two main parts. The first one sets up the initial constraints and is problem dependent. The second one solves the problem in the case of multivariate polynomials subject to constraints of monotony. Other examples may be obtained by altering slightly this second part.

Not reported here, we have compared our method to Murray and coauthor ([15]) algorithms. The results are quite similar. The main difference comes from the fact that the polynomials they are fitting must be monotone over the entire real line. This has the consequence to slightly degrade the fit, when compared to our algorithms which works on a bounded interval. On the other hand, the coefficients of the polynomials may be very different and the shape of our solution is not guaranteed to be monotone outside the definition interval.

In the simulations below, the examples we consider are based on polynomials which do not respect the sufficient conditions of Theorem 3. Another characteristic very different from the examples taken in [15] is that we work with very few points as it is representative of many real world situations: gathering experimental data is always a long and hard process.

4.1 Other type of constraints

A few features open up the applicability of our method to a really large panel of parametric regressions. This is discussed in more details in this section.

1. As it is well known, monotonicity requirements are not the only shape constraints that can be considered. In fact, the same method can be applied to any shape constraints as long as the corresponding constraints stay linear with respect to the coefficients of the model. This includes monotony, concavity or convexity constraints, bound constraints on the function itself, or on its derivatives and equality constraints.
2. Monotony requirements (or other constraints) can be applied simultaneously to any number of variables. The only consequence is that the number of constraints to fulfill will increase with the number of variables. Four constraints are required in the real example worked out in section 4.4.
3. Obviously, every monotone transformation of the variables x_1, \dots, x_v will not change the procedure. This includes dealing with decreasing requirements by changing the corresponding variable x_1 (for example) in $1 - x_1$, or changing the definition domain in any bounded interval in \mathbb{R} .

- An other advantage of the method is that expert knowledge can be incorporated in the polynomial to obtain the desired behavior more easily. If one expects a linear variation with respect to the first variable, while the second variable should correspond to a third degree polynomial, then the corresponding terms can be omitted in the fit to force the response to exhibit the correct shape.

4.2 A bounded function in dimension 1

We consider here the function $y = 1 - 4(x - 1/2)^2$, drawn on Figure 3, in black, on the left panel.

At $x = 1/2$, this function reaches its maximum, $y = 1$. Ten values for x are drawn uniformly on $[0, 1]$, and a random gaussian noise of standard deviation 0.1 is added to the resulting values of y . The points are shown as green squares on the Figure 3. They are fitted with a 2 degree polynomial with the additional constraint that the maximum should not exceed 1.

The resulting polynomial is drawn in red. In dashed blue, we find the original equation and in black, the least square solution. We can see that the obtained fit respects the constraint, unlike the least square fit with a maximum exceeding 1. In this case, 12 constraints were necessary.

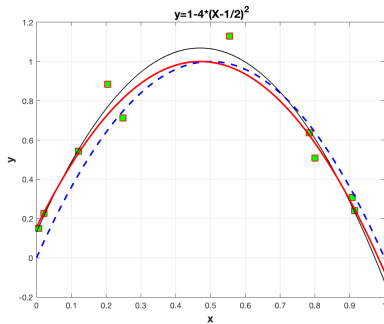


Figure 3: fit of the function $y = 1 - 4(x - 1/2)^2$. The original function is in blue, the obtained constrained function with a maximum of 1 in red. In black the least square solution.

4.3 Simulated example in dimension 2

In Figure 4, 100 points are generated with the equation $y = -6x_1^3x_2 + 10x_1^2 - 3x_1$. A gaussian noise with a standard deviation of 0.1 is again added to y .

On the left panel, the original function is plotted. On the right panel, we show the calculated regression with the constraint that y should increase with x_1 . The figures are rotated to clearly show the behavior of the original and calculated functions. It can be seen that the original function first decreases with x_1 and then increases. On the right panel, the calculated function is always increasing with respect to x_1 .

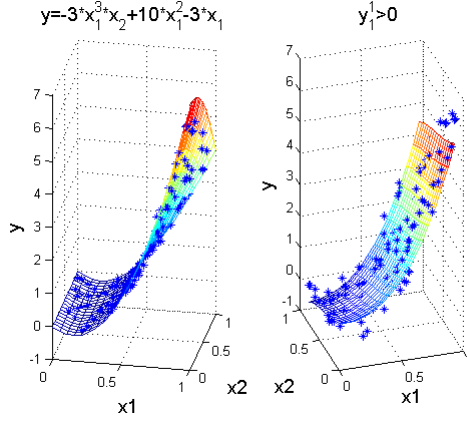


Figure 4: regression in dim 2 with $d = 3$, monotony constraints on x_1

4.4 Real example: hydrotreatment of naphtha

In petroleum process engineering, hydrotreating consists in treating a petroleum cut under hydrogen pressure in an industrial reactor. After being extracted, the crude oil has first to be refined and fractionated in different cuts before being commercialized. Specifically, in naphtha cuts, impurities (mainly sulphur) must be removed, before any further use.

A pseudo-kinetic model is commonly proposed to approximate this process and is given by the following equation :

$$\ln\left(\frac{C}{C_0}\right) = -k \frac{1}{V V H} \exp\left(-\frac{E_a}{RT}\right) P_{H_2}^J P_{H_2S}^s$$

with the following variables : C the concentration of the chemical to be removed remaining at the outlet of the reactor;

C_0 its initial concentration;

T the temperature of the process;

P_{H_2} the partial hydrogen pressure;

P_{H_2S} the partial H_2S pressure;

$V V H$ the Velocity per Volume and per Hour;

k , E , m and s are parameters and must be estimated from experimental measurements.

Taking the logarithm of each side of this formula, the equation can be easily linearized and rewritten as $y = \sum_{j=1}^4 \beta_j x_j$, where $y = \ln(-\ln(\frac{C}{C_0}))$, $x_1 = 1/T$, $x_2 = \ln(V V H)$, $x_3 = \ln(P_{H_2})$, $x_4 = \ln(P_{H_2S})$.

Unfortunately, this expression is unable to take into account the full complexity of the process, and a few empirical terms must be added. Finally, a degree 2 polynomial in the variables $x = (x_1, x_2, x_3, x_4)$ is postulated. But some constraints must be respected : the process is more efficient (which means that C decreases or equivalently y increases) when :

- the temperature T increases or x_1 decreases
- $V V H$ decreases or x_2 increases
- P_{H_2} or x_3 increases.

Figure 5 compares the results when regressing with and without constraints. The left panel exhibits the residues (y calculated - y experimental), showing only minor differences when the experimental points are predicted by both methods: RMSE=0.438 with constraints and 0.411 without. But the obtained equations are really different as shown on the right.

On the right panel, we see a kind of spider plot, showing the behavior of the response when only one variable varies at a time, starting from a given point in the domain (here: $[x^1 = 0.71, x^2 = 0.64, x^3 = 0.174, x^4 = 0.062]$). The dotted lines correspond to the regression without constraints, the solid line to the regression with constraints. The plain triangle marks the estimated response for the regression without constraints, the circle for the regression with constraints. x-axis are translated so that every curve crosses at the center of the graphic. Black lines correspond to variations along T or x^1 , red lines to variations with VVH or x^2 , blue lines to variations with P_{H_2} or x^3 . The behaviors for the regression without constraints are obviously wrong: the black dotted line is increasing instead of decreasing and the blue has a minimum.

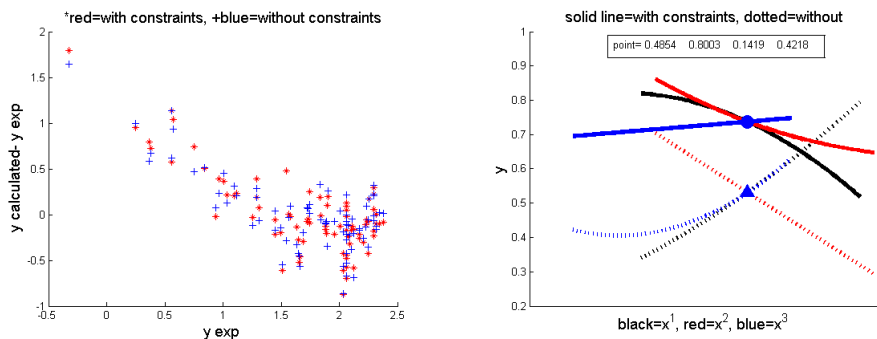


Figure 5: **polynomial fit to the data of HDS experiments** Residue diagram for the HDS data on the left panel and spider plot for a comparison of the UNconstrained and constrained multivariate regression on the right

5 Perspectives and Conclusions

The proposed procedure is very general and flexible. Moreover it can be found useful in a lot of problems. It is specially well adapted to polynomial regression, a problem occurring very often in industrial applications. Most importantly, our method will give satisfactory results in multidimensional cases even with few available experimental data.

For big data set, the limitations will come from the number of variables. To give a rough idea, if we set the limit at 500000 constraints, it gives a limit of about 80 variables for a fit with a multivariate degree 3 polynomial in every variable at the first iteration..

The proposed method will suffer from the usual flaws of linear regression, as it is based on a least squares procedure. Notably, to avoid some instabilities in the coefficients, a bit of regularization would be welcome, as considered in [21].

A second enhancement would be to optimize the parameters of the basis functions. In this paper, the parameters of the Chebyshev functions are supposed fixed. But obviously,

with extra computational work, they could be automatically chosen to fit as best as possible the observation points at hand.

Thirdly, the scope of this kind of regression could be extended to non parametric regressions. GAMs are natural good candidates as well as local polynomial regression.

Fourth, uncertainty intervals are not included in the method. Indeed, as the constraints change at each iteration, the residues can not be considered as identically distributed, so that bootstrap algorithms are not available at first sight.

The original algorithms for polynomials are developed in Matlab[®] and available upon request.

Acknowledgements

The authors would like to thank Damien Hudebine from IFPEN for the example in hydrotreatment, Pr. Fabrice Gamboa for his advices and Pr. G. Ciuperca for her careful reading. This work was supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR), and by the GdR MASCOTNUM.

Appendix: proofs

Proof of Proposition 1.

Because of the linearity of Z on \mathbb{T} and because a simplex is convex by definition, $F_J(x)$ is guaranteed to be positive on $[0, 1]$, if it is positive in every vertices of S_J .

Preliminaries to Theorem 3 .

Theorem 3 needs the three following preliminary lemmas where we prove that in a small neighborhood of a point on a curve, a smooth curve is included in its osculating simplex.

Let T_0 be a point on the curve C_J corresponding to x_0 and T_J corresponding to $x_J = x_0 + h$. We denote T_0, T_1, \dots, T_J the vertices of the osculating simplex between x_0 and $x_0 + h$.

The vectors $\mathbf{f}^{(1)}(x_0), \dots, \mathbf{f}^{(j)}(x_0)$ are all linearly independent. This results from the definition 3 of an ET. In this basis,

$$\overrightarrow{T_0 T_j} = \sum_{k=1}^j \gamma_{k,j} \mathbf{f}^{(k)}(x_0). \quad (6)$$

Similarly,

$$\overrightarrow{T_j T_J} = \sum_{k=j+1}^J \gamma_{k,j} \mathbf{f}^{(k)}(x_J).$$

Lemma 1 *Let D_j be the determinant*

$$D_j = \left| \mathbf{f}^{(1)}(x_0) \quad \dots \quad \mathbf{f}^{(j)}(x_0) \quad \mathbf{f}^{(1)}(x_J) \quad \dots \quad \mathbf{f}^{(J-j)}(x_J) \right|.$$

$D_{j,k}$ is obtained by replacing the k -th column of D_j by $\mathbf{f}(x_J) - \mathbf{f}(x_0)$. Then, that for all j , $D_j \neq 0$, and $\gamma_{k,j} = \frac{D_{j,k}}{D_j}$.

All the determinants D_j are strictly positive as a result of the definition 3 of ET systems and ([10] chapter 1).

For $0 < j < J$, T_j belongs to the osculating j -space at T_0 and simultaneously to the osculating $J - j$ -space at T_J . Thus, the vector $\overrightarrow{T_0T_j}$ is a linear combination of the first j derivatives at T_0 and similarly $\overrightarrow{T_JT_j}$ is a linear combination of the first $J - j$ derivatives at T_J . Consequently, the coordinates $\gamma_{1,j}, \dots, \gamma_{J,j}$ of T_j , as stated in this lemma, result from the Cramer's rule applied to the linear system of equations:

$$\left(\mathbf{f}^{(1)}(x_0) \cdots \mathbf{f}^{(j)}(x_0) \mathbf{f}^{(1)}(x_J) \cdots \mathbf{f}^{(J-j)}(x_J) \right) \begin{pmatrix} \gamma_{1,j} \\ \cdots \\ \gamma_{J,j} \end{pmatrix} = (\mathbf{f}(x_J) - \mathbf{f}(x_0))$$

Lemma 2 *The coefficients $\gamma_{k,j}$ can be approximated by $\gamma_{k,j} \sim \frac{h^k}{k!} + o(h^k)$.*

We start from the previous lemma 1 and the expression of $D_{j,k}$.

Since $x_J = x_0 + h$, taking the Taylor expansion of $f_1(x_0 + h) - f_1(x_0), \dots, f_J(x_0 + h) - f_J(x_0)$, it can be readily shown that the first non vanishing term in the development of the $D_{j,k}$ for $1 \leq k \leq j$ is $\frac{h^k}{k!} D_j$. This results in the statement of the lemma.

Lemma 3 *Let T_0 and T_J be two points on the curve C_J corresponding to x_0 and $x_J = x_0 + h$. Then, for h small enough, the portion of the curve C_J between x_0 and x_J is included in the cone generated by $\{\overrightarrow{T_0T_1}, \dots, \overrightarrow{T_0T_J}\}$.*

Our aim is to prove that any point $T_x = (f_1(x), \dots, f_J(x))$ verifies:

$$\overrightarrow{T_0T_x} = \sum_{j=1}^J \lambda_j(x) \overrightarrow{T_0T_j}, \text{ s.t. } \lambda_j(x) \geq 0, \forall j \in [1, j], \forall x \in [x_0, x_0 + h],$$

where, for $j = 1, J$, $\lambda_j(x)$ are real coefficients depending on x and T_j are the vertices of the osculating simplex.

Since $\overrightarrow{T_0T_j}$ belongs to the osculating j -space at T_0 for any $0 < j < J$, $\overrightarrow{T_0T_j}$ can be written: $\overrightarrow{T_0T_j} = \sum_{k=1}^j \gamma_{k,j} \mathbf{f}^{(k)}(x_0)$. That is:

$$\begin{aligned} \overrightarrow{T_0T_1} &= \gamma_{1,1} \mathbf{f}^{(1)}(x_0) \\ \overrightarrow{T_0T_2} &= \gamma_{1,2} \mathbf{f}^{(1)}(x_0) + \gamma_{2,2} \mathbf{f}^{(2)}(x_0) \\ &\dots \end{aligned}$$

Gathering all the coefficients $\gamma_{k,j}$ in a matrix $\mathbf{\Gamma}$, we obtain the system of linear equations:

$$\begin{pmatrix} \overrightarrow{T_0 T_1} \\ \overrightarrow{T_0 T_2} \\ \dots \\ \overrightarrow{T_0 T_J} \end{pmatrix} = \mathbf{\Gamma} \begin{pmatrix} {}^t \mathbf{f}^{(1)}(x_0) \\ {}^t \mathbf{f}^{(2)}(x_0) \\ \dots \\ {}^t \mathbf{f}^{(J)}(x_0) \end{pmatrix}. \quad (7)$$

The matrix $\mathbf{\Gamma}$ is lower triangular and non singular: indeed, its determinant is the product of all the $\gamma_{j,j}$, which are approximated by $\frac{h^j}{j!}$ (see Lemma 2). Therefore the determinant of $\mathbf{\Gamma}$ is different from 0 as long as h is distinct from 0.

Furthermore, a Taylor expansion of $\overrightarrow{T_0 T_x}$ gives

$$\overrightarrow{T_0 T_x} \sim \sum_{j=1}^J \frac{(x-x_0)^j}{j!} \mathbf{f}^{(j)}(x_0) + o((x-x_0)^J). \quad (8)$$

Plugging together equations (7) and (10), we obtain:

$$\begin{pmatrix} \lambda_1(x) \\ \dots \\ \lambda_J(x) \end{pmatrix} \sim {}^t \mathbf{\Gamma}^{-1} \begin{pmatrix} \frac{(x-x_0)}{1!} \\ \dots \\ \frac{(x-x_0)^J}{J!} \end{pmatrix} + o((x-x_0)^J).$$

The next step is to solve $\mathbf{\Gamma}^{-1}$. With Lemma 2, we have

$$\mathbf{\Gamma} \sim \begin{pmatrix} \frac{h}{1!} & 0 & \dots \\ \frac{h^2}{2!} & \frac{h^2}{2!} & 0 & \dots \\ \dots & \dots & \dots & \dots \\ \frac{h^J}{J!} & \dots & \dots & \frac{h^J}{J!} \end{pmatrix} + o(h).$$

We find a lower band matrix for $\mathbf{\Gamma}^{-1}$:

$$\mathbf{\Gamma}^{-1} = \begin{pmatrix} \frac{1!}{h} & 0 & 0 & \dots & 0 \\ -\frac{1!}{h} & \frac{2!}{h^2} & 0 & \dots & 0 \\ 0 & -\frac{2!}{h^2} & \frac{3!}{h^3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -\frac{(J-1)!}{h^{J-1}} & \frac{J!}{h^J} & \dots \end{pmatrix} + o(h).$$

Finally, replacing h by $x_J - x_0$, we obtain a positive approximation for $\lambda_j(x)$, $1 \leq j \leq J$:

$$\lambda_j(x) \sim \frac{(x-x_0)^j}{(x_J-x_0)^j} \left(1 - \frac{1}{j+1}(x-x_0)\right) + o(h).$$

Proof of Theorem 3.

We denote T_j , for $0 < j < J$, the vertex of the osculating simplex defined as the intersection of the osculating j -space at T_0 and the osculating $J-j$ -space at T_J . We define F_j for $0 \leq j \leq J$ as the face of the osculating simplex containing all the vertices except T_j .

Each F_j , $0 \leq j \leq J$, intersects C_J exactly J times taking into account the multiplicities. Indeed, T_0 is the only contact point between C_J and F_j since the osculating hyperplane at T_0 is the supporting hyperplane of F_j . The same holds for T_J and F_0 .

For F_j , for $0 < j < J$, by construction of the osculating simplex, T_0, T_1, \dots, T_{j-1} belong to the osculating $j - 1$ -space at T_0 . Thus the face T_0, T_1, \dots, T_{j-1} is supported by the vectorial sub-space spanned by the first $j - 1$ derivatives at T_0 . In the same way, $T_{j-1}, \dots, T_{j-1}, T_j$ is included in the vectorial sub-space spanned by the first $J - j - 1$ derivatives at T_j .

This amounts to say that the multiplicity of the contact between C_J and F_j at T_0 is j . Similarly, the multiplicity of the contact between C_J and F_j at T_j is $J - j$. Finally, C_J intersects J times F_j . Due to Theorem 1 T_0 and T_j are the only intersection points between T_0 and T_j . As a conclusion, C_J stays on one side of each of the faces F_j .

The curve C_J lies inside its osculating simplex between x_0 and x_J because since it is the case in a small neighborhood of x_0 (see Lemma 3), it is true everywhere.

Proof of Proposition 2.

item 1 Thanks to Proposition 1, $A_{J,K}$ can be seen as

$$A_{J,K} = \{\boldsymbol{\alpha} \mid \forall t \in P_{J,K}, \langle \boldsymbol{\alpha}, t \rangle \geq 0\}.$$

By construction, $P_{J,K+1} \subset P_{J,K}$. Indeed, each simplex of $P_{J,K+1}$ results from cutting in two one of the simplexes in $P_{J,K}$, as illustrated on Figure 1.

Thus, if we have $\langle T, \boldsymbol{\alpha} \rangle \geq 0$ for all the vertices T of $P_{J,K}$, then it is also true for all the vertices of $P_{J,K+1}$. This last statement means that $A_{J,K} \subset A_{J,K+1}$.

$P_{J,K}$ is a collection of successive osculating simplexes, each of them finishing at the point where the next one begins. Thus $P_{J,K}$ circumscribes the curve C_J , and this implies that if $\boldsymbol{\alpha}$ is in $A_{J,K}$ then $\forall x \in [0, 1], F_{\boldsymbol{\alpha}}(x) \geq 0$, or equivalently that $A_{J,K} \subset A_J$.

item 2 We only detail this claim for A_J , similar considerations can be applied to the $A_{J,K}$. Indeed, if $\forall x F(x) \geq 0$ for a given $\boldsymbol{\alpha}$, then it is also verified for $\lambda \boldsymbol{\alpha}$ where λ is real and positive. Thus A_J is a cone. It is convex: if $F(x) \geq 0$ for $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$, then it is also non-negative for $p\boldsymbol{\alpha}_1 + (1 - p)\boldsymbol{\alpha}_2$ for any $p \in [0, 1]$.

If A_J were not closed, then we could find a vector of coefficients $\boldsymbol{\alpha}$, limit of a sequence of coefficients $\boldsymbol{\beta}_l$ belonging to A_J , for which $\exists x \in [0, 1]$ such that $F_{\boldsymbol{\alpha}}(x) < 0$.

We rewrite $F_{\boldsymbol{\beta}_l}(x)$ as $F_{\boldsymbol{\beta}_l}(x) = F_{\boldsymbol{\beta}_l}(x) - F_{\boldsymbol{\alpha}}(x) + F_{\boldsymbol{\alpha}}(x)$.

$\boldsymbol{\alpha}$ being the limit of the sequence $\boldsymbol{\beta}_l$, we could make the difference $F_{\boldsymbol{\beta}_l}(x) - F_{\boldsymbol{\alpha}}(x)$ as small as needed. Indeed, $F_{\boldsymbol{\alpha}}(x)$ (resp. $F_{\boldsymbol{\beta}_l}(x)$) is built as the scalar product of $\boldsymbol{\alpha}$ (resp. $\boldsymbol{\beta}_l$) and $\mathbf{f}_{\bullet}(x) = {}^t(1, f_1(x), \dots, f_J(x))$. Thus,

$$F_{\boldsymbol{\beta}_l}(x) - F_{\boldsymbol{\alpha}}(x) = \langle \boldsymbol{\beta}_l - \boldsymbol{\alpha}, \mathbf{f}_{\bullet}(x) \rangle.$$

By Cauchy-Schwartz inequality, we have:

$$|\langle \boldsymbol{\beta}_l - \boldsymbol{\alpha}, \mathbf{f}_{\bullet}(x) \rangle| \leq \|\boldsymbol{\beta}_l - \boldsymbol{\alpha}\| \|\mathbf{f}_{\bullet}(x)\| \leq \|\boldsymbol{\beta}_l - \boldsymbol{\alpha}\| \sup_{x \in [0,1]} \|\mathbf{f}_{\bullet}(x)\|.$$

Thus we could determine an integer L for which:

$$\forall l > L, |\langle \beta_l - \alpha, \mathbf{f}_\bullet(x) \rangle| < |F_\alpha(x)|.$$

$F_{\beta_l}(x)$ would be negative. This is a contradiction.

item 3 Indeed, the set $B = \{\alpha \mid \forall x \in [0, 1], F_\alpha(x) > 0\}$ is open and $B \subset A_J$. We aim at proving that $\bar{B} = A_J$, which will show that $B = A_J$, the adherence of A_J .

If $\alpha \in A_J$, for all positive integer l the vector $\alpha_l = \alpha + \frac{1}{l}$ belongs to B :

$$F_{\alpha_l}(x) \geq F_\alpha(x) + \frac{1}{l} > 0.$$

The sequence formed by the vectors α_l converges to α when l approaches infinity, which permits to conclude that $\bar{B} = A_J$.

item 4 This is a direct consequence of item 1: since $A_{J,K} \subset A_{J,K+1}$, the minimum over $A_{J,K}$ is greater or equal to the minimum over $A_{J,K+1}$.

Proof of Theorem 4.

We denote $B = \overline{\bigcup_{K \rightarrow \infty} A_{J,K}}$. Our goal is to prove that $B = A_J$, or in other words that B is dense in A_J . If this is true, $\tilde{\alpha}_J$ the optimal solution to Problem (2), as an element of A_J , is the limit of a sequence of vectors $\alpha_{J,K}$, each of them taken in one $A_{J,K}$.

$\alpha_{J,K}$ is not necessarily the solution $\tilde{\alpha}_{J,K}$ to Problem (3). However, since $A_{J,K} \subset A_J$, we have

$$\sum_{i=1}^I (Y_i - F_{\alpha_{J,K}}(X_i))^2 \geq \sum_{i=1}^I (Y_i - F_{\tilde{\alpha}_{J,K}}(X_i))^2 \geq \sum_{i=1}^I (Y_i - F_{\tilde{\alpha}_J}(X_i))^2.$$

$\tilde{\alpha}_J$ being the limit of the sequence $\alpha_{J,K}$ when K tends to infinity is then the limit of $\tilde{\alpha}_{J,K}$.

The inclusion $B \subset A_J$ is immediate, as a consequence of items 1 and 2 of Lemma 2. Conversely, we have to prove that every point of A_J is attained. We choose α in A_J and want to show that $\alpha \in \overline{\bigcup_{K \rightarrow \infty} A_{J,K}}$.

Continuing with the sequence of α_l defined in item 3 of Lemma 2, since α_l belongs to the open set A_J , we can find an open ball B_l of radius r_l , centered in α_l and included in A_J . The radius r_l is chosen to be decreasing with l toward 0 as its limiting value. If we exhibit now a vector $\beta_l \in B_l$ simultaneously belonging to A_{J,K_l} for some K_l , our assertion is proved: α will be the limit of a sequence of β_l taken in A_{J,K_l} , since $\|\beta_l - \alpha\| \leq 1/l + r_l$ by the triangular inequality.

We first observe that the distance from any point of P_{J,K_l} to the curve C_J can be made as small as needed: more precisely,

$$\forall \epsilon, \exists K_l \text{ such that } \forall u \in P_{J,K_l}, \exists t \in C_J \text{ for which } \|u - t\| < \epsilon.$$

Indeed, we restrict our attention to one of the simplexes S_k composing P_{J,K_l} , the simplex containing u . S_k begins at the value $x_0 = \frac{k}{2K_l}$ and finishes at $x_0 + h$ with $h = \frac{1}{2K_l}$. The

maximum distance of two points within S_k is one of the distances between two of its vertices. By means of Equation (6) and Lemma 2, when K_l is sufficiently large, calling T_{j_1} and T_{j_2} two of these vertices, the vector $\overrightarrow{T_{j_1}T_{j_2}}$ is approximated by

$$\overrightarrow{T_{j_1}T_{j_2}} \sim \sum_{l=j_1+1}^{j_2} \frac{h^l}{l!} \mathbf{f}^{(l)}(x_0) + o(h^{j_1+1}).$$

$\|T_{j_1}T_{j_2}\|$ and then $\|u - t\|$ are bounded from above by $\frac{M_1}{2K_l}$ with $M_1 = \sum_{j=1}^J \sup_{x \in [0,1]} \|\mathbf{f}^{(j)}(x)\|$.

Now, for any $\beta_l \in B_l$ and any $u \in S_k$ we would like to prove that $\langle u, \beta_l \rangle > 0$. We start from the identity

$$\langle u, \beta_l \rangle = \langle u - t, \beta_l \rangle + \langle t, \beta_l - \alpha_l \rangle + \langle t, \alpha_l \rangle.$$

- We observe that $\langle t, \alpha_l \rangle > 1/l$.
- By Cauchy-Schwartz inequality $\langle t, \beta_l - \alpha_l \rangle \geq -r_l M_2$ with $M_2 = \sup_{x \in [0,1]} \|\mathbf{f}(x)\|$.
- Similarly, $\langle u - t, \beta_l \rangle \geq -\frac{M_1}{2K_l} \|\beta_l\|$. Decomposing β_l in $\beta_l - \alpha_l + \alpha_l - \alpha + \alpha$, we obtain $\|\beta_l\| \leq r_l + 1/l + \|\alpha\|$.

Eventually,

$$\langle u, \beta_l \rangle \geq -\frac{M_1}{2K_l} (r_l + \frac{1}{l} + \|\alpha\|) - r_l M_2 + \frac{1}{l}.$$

For a given l , r_l and K_l are chosen so that the right part of the previous inequality be positive.

Proof of Proposition 3.

When cutting the initial simplex at x the volume of the two new simplexes replacing the old one becomes: $V_{\text{new}} = V(x_0, x) + V(x, x_J)$. If $x = x_0$ or $x = x_J$ then $V_{\text{new}} = V(x_0, x_J)$ and is maximum. Due to Rolle's theorem, there exists a x for which V_{new} is minimum. This minimum is unique since by construction $V(x_0, x)$ is a strictly increasing function while $V(x, x_J)$ is strictly decreasing.

Proof of Proposition 4.

Indeed, the volume of a simplex with vertexes T_0, \dots, T_J is known to be:

$$V(x_0, x_J) = \frac{1}{J!} |\overrightarrow{T_0T_1} \ \dots \ \overrightarrow{T_0T_J}|.$$

Taking the notation of Lemma 1, for $j < J$, $\overrightarrow{T_0T_j}$ is decomposed in

$$\overrightarrow{T_0T_j} = \sum_{k=1}^j \frac{D_{j,k}}{D_j} \mathbf{f}^{(k)}(x_0).$$

Standard manipulations on determinants give the expected result.

Proof of Proposition 5.

We recall the expressions of D_j and $D_{j,j}$ that will be used abundantly in this proposition.

$$\begin{aligned} D_j &= |\mathbf{f}^{(1)}(x_0) \cdots \mathbf{f}^{(j)}(x_0) \mathbf{f}^{(1)}(x_J) \cdots \mathbf{f}^{(j)}(x_J)| \\ D_{j,j} &= |\mathbf{f}^{(1)}(x_0) \cdots \mathbf{f}(x_J) - \mathbf{f}(x_0) \mathbf{f}^{(1)}(x_J) \cdots \mathbf{f}^{(j)}(x_J)| \\ D_{j,J} &= |\mathbf{f}^{(1)}(x_0) \cdots \mathbf{f}^{(j)}(x_0) \cdots \mathbf{f}^{(j-1)}(x_0) \mathbf{f}(x_J) - \mathbf{f}(x_0)| \end{aligned}$$

The first step is to prove that $D_j \propto (x_J - x_0)^{\delta_j}$ where $\delta_j = j(J - j)$. Then we establish with the same kind of reasoning that $D_{j,j} \propto (x_J - x_0)^{\delta_j + j}$. The conclusion of the proposition follows immediately as a final step.

case D_{J-1} . We begin by $D_{J-1} = |\mathbf{f}^{(1)}(x_0) \cdots \mathbf{f}^{(J-1)}(x_0) \mathbf{f}^{(1)}(x_J)|$ to illustrate how to proceed. Only the last column of this determinant depends on x_J and is equal to $\mathbf{f}^{(1)}(x_J)$.

We consider D_{J-1} as a function of x_J . Developing along the last column we see that D_{J-1} is a polynomial of degree $J - 1$ in x_J . Besides, by construction, D_{J-1} and its first $J - 2$ derivatives with respect to x_J vanish at $x_J = x_0$.

Therefore, $D_{J-1} \propto (x_J - x_0)^{J-1}$.

case D_j . For a general j , the last $J - j$ columns depend on x_J . We face two situations depending on whether the number of columns where we find x_J is smaller or equal to the number of columns where x_0 is present ($J \leq 2j$ or not).

We assume first that $J \leq 2j$. D_j can be considered as a polynomial in x_J . The terms of highest degree come from the product of coordinates taken in the last $J - j$ columns. This results in a polynomial of degree $\delta_j = j(J - j)$.

We examine now the derivatives of D_j evaluated at $x_J = x_0$. We must take at least the j -th derivative to remove the collinearity between the derivative of the column $\mathbf{f}^{(1)}(x_J)$ and any of the $\mathbf{f}^{(k)}(x_0)$ when $k = 1, \dots, j$ and $x_J = x_0$. Then, we take the derivative of the column $\mathbf{f}^{(2)}(x)$ in D_j at least again j times. This process continues until the last column. Eventually, the derivative of D_j has been taken $j(J - j)$ times before we find the first non vanishing derivative of $D_j(x)$ taken at $x_J = x_0$: indeed, $D_j^{(j(J-j))}(x_0)$ is exactly the wronskian of the system of functions $\{f_k^{(1)}(x)\}_{k=1}^J$ evaluated at $x = x_0$, which is strictly positive since this system is assumed to be an ECT.

As a conclusion to the case $J \leq 2j$, $D_j \propto (x_J - x_0)^{\delta_j}$ is the only possibility.

The second alternative is when $J > 2j$. Here, we reverse the role of x_0 and x_J and obtain exactly the same result for symmetry reasons.

case $D_{j,j}$. The analysis is exactly the same as for D_j save for a supplementary column where x_J is present, since $\mathbf{f}^{(j)}(x_0)$ is replaced by $\mathbf{f}(x_J) - \mathbf{f}(x_0)$. This results in a polynomial of degree $\delta_{j,j} = \delta_j + j = j(J - j + 1)$.

When $J < 2(J - j)$, $D_{j,j}$ is treated as a polynomial in x_0 and a similar reasoning leads to the same conclusion.

Conclusion. Plugging all these evaluations in the formula of the volume of the osculating simplex (Proposition 4) gives the expected result when the f_j form a sequence of monomials.

Proof of Theorem 5.

Let x be the parameter of the cut point. From Lemma 3, x minimizes $V(x_0, x) + V(x, x_J)$. Applying Proposition 5, it amounts to find the minimum of $(x - x_0)^{J(J+1)/2} + (x_J - x)^{J(J+1)/2}$, which is obviously obtained when $x = \frac{x_0 + x_J}{2}$.

Proof of Theorem 6

We only have to prove item 1. Item 2 is immediate.

Given a point $T_{x^*} = {}^t(f_1(x^*), \dots, f_J(x^*))$ corresponding to the values $x^* = (x_1^*, \dots, x_V^*)$ of the variables, we have to show that $T_{x^*} \in P_J$.

By construction, each $T_{1,j_1} \otimes \dots \otimes T_{v,j_v}$ matches T_j one of the vertex of P_J .

Our aim is then to exhibit $J + 1$ non negative coefficients $\mu_j, j = 0, J$, summing to 1 such that

$$T_{x^*} = \sum_{j=0}^J \mu_j T_j.$$

For each x_v , we consider the curve $C_{J,v}$ described by $(f_{1,v}(x_v), \dots, f_{1,J_v}(x_v))$ and the point T_{v,x_v^*} corresponding to the value x_v^* of the variable x_v .

Then we may find $J_v + 1$ positive coefficients λ_{v,j_v} summing to 1 such that:

$$T_{v,x_v^*} = \sum_{j_v=0}^{J_v} \lambda_{v,j_v} T_{v,j_v}. \quad (9)$$

Stemming from Equation 5, T_{x^*} can be decomposed in the tensor product:

$$T_{x^*} = T_{1,x_1^*} \otimes \dots \otimes T_{V,x_V^*}. \quad (10)$$

The combination of equations 10 and 9 gives:

$$T_{x^*} = \sum_{j_1=0}^{J_1} \dots \sum_{j_V=0}^{J_V} (\lambda_{1,j_1} \dots \lambda_{V,j_V} T_{1,j_1} \otimes \dots \otimes T_{V,j_V})$$

Furthermore, one can observe that:

$$\sum_{j_1=0}^{J_1} \dots \sum_{j_V=0}^{J_V} \lambda_{1,j_1} \dots \lambda_{V,j_V} = \prod_{v=1}^V (\lambda_{v,0} + \dots + \lambda_{v,J_v}) = 1.$$

Thus T_{x^*} is expressed as a linear combination of the vertices of P_J , where all the coefficients are positive and sum to 1. The proof of item 1 is complete.

References

- [1] Aharon Ben-Tal and Arkadiaei Semenovich Nemirovskiaei. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.
- [2] S. M. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, pages 346–349, 1974.

- [3] S. Da Veiga and A. Marrel. Gaussian process modeling with inequality constraints. *Annales de la faculté des sciences de Toulouse Mathématiques*, 21(3):529–555, 4 2012.
- [4] Carl De Boor. *A practical guide to splines; rev. ed.* Applied mathematical sciences. Springer, Berlin, 2001.
- [5] P. Du, C.F. Parmeter, and J.S. Racine. Nonparametric kernel regression with multiple predictors and multiple shape constraints. *Statistica Sinica*, 23:1347–1371, 2013.
- [6] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design (3rd Ed.): A Practical Guide.* Academic Press Professional, Inc., San Diego, CA, USA, 1993.
- [7] M. Gasca and C.A. Micchelli. *Total Positivity and Its Applications.* Mathematics and Its Applications. Springer Netherlands, 2013.
- [8] Myung-Soo Kim Gerald E. Farin, Josef Hoschek. *Handbook of Computer Aided Geometric Design.* Elsevier, 2002.
- [9] Douglas M. Hawkins. Fitting monotonic polynomials to data. *Computational Statistics*, 9:233–247, 1994.
- [10] S. Karlin and W.J. Studden. *Tchebycheff systems: with applications in analysis and statistics.* Pure and applied mathematics. Interscience Publishers, 1966.
- [11] F. Lauer and G. Bloch. Incorporating prior knowledge in support vector regression. *Machine Learning*, 2008.
- [12] G.G. Lorentz. *Bernstein polynomials.* Mathematical expositions. University of Toronto Press, 1953.
- [13] H. Maatouk and X Bay. Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582, 2017.
- [14] M.C. Meyer. Constrained penalized splines. *Canadian Journal of Statistics*, 40(1):190–206, 2012.
- [15] K. Murray, S. Müller, and B. A. Turlach. Fast and flexible methods for monotone polynomial fitting. *Journal of Statistical Computation and Simulation*, 86(15):2946–2966, 2016.
- [16] J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer, New York, 2nd edition, 2006.
- [17] Dávid Papp and Farid Alizadeh. Shape-constrained estimation using nonnegative splines. *Journal of Computational and Graphical Statistics*, 23(1):211–231, 2014.
- [18] J.M. Peña. *Shape Preserving Representations in Computer-aided Geometric Design.* Nova Science Publishers, 1999.

- [19] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer-Verlag, New York, 2nd edition, 2005.
- [20] Larry Schumaker. *Spline Functions: Basic Theory*. Cambridge Mathematical Library. Cambridge University Press, 3 edition, 2007.
- [21] H. Trevor, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2nd edition, 2009.
- [22] S. Wood. *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2006.