Simplex Regression: Multivariable parametric regression under shape constraints

François Wahl, Thibault Espinasse

Statistics and Computing manuscript No.
(will be inserted by the editor)

Simplex Regression: Multivariable Parametric Regression under Shape Constraints

François Wahl · Thibault Espinasse

Received: date / Accepted: date

Abstract We consider a multivariable regression model under shape constraints (monotonicity, convexity, positivity,...) built as a linear combination of product of functions of a single variable. For each variable, the functions form a Chebyshev system. We develop an iterative procedure, where at each step the initial shape requirement is approximated by a set of linear constraints. The main result of this paper is that this procedure is shown to converge to the optimal solution in the least square sense. The theory is first established in the single variable case and then extended to the multivariable framework by means of tensor products. Numerical studies and a real industrial example with a multivariable polynomial regression subject to shape constraints of monotony illustrate the performance of the proposed method.

Keywords monotony · quadratic programming · Chebyshev system · simplexes

1 Introduction

The focus in this article is on multivariable parametric regression under shape constraints on bounded intervals of sets of $\mathbb{R}$ in the case of a single variable or on a product of V intervals with V variables. Shape constraints refer to monotonicity, concavity or bounded constraints for the function or for its derivatives.

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

This work was supported by the LABEX MILYON (ANR-10-LABX-0070), and by the GdR MASCOT-NUM.

François Wahl
Camille Jordan Institute, Université Claude Bernard Lyon 1, F-69622 Villeurbanne, France
E-mail: Francois.Wahl@univ-lyon1.fr
IFP Energies nouvelles, Rond-point de l'échangeur de Solaize, BP 3, 69360 Solaize
E-mail: Francois.Wahl@ifpen.fr

Thibault Espinasse
Camille Jordan Institute, Université Claude Bernard Lyon 1, F-69622 Villeurbanne, France
E-mail: : Thibault.Espinasse@math.univ-lyon1.fr

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 $\mathbb{R}$, which copies the structure of traditional polynomials: F_α is expressed as a linear combination of $J+1$ known elementary functions f_j, with $f_0(x) = 1$:

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

where $\boldsymbol{\alpha}$ is the vector of coefficients, and each $f_j(x)$ is decomposed in a product of V functions of a single variable:

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

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

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 be written:

$$Y_i = F_\alpha(X_i) + \epsilon_i \tag{2}$$

The real coefficients stored in the vector $\boldsymbol{\alpha}$ 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 Problem 3:

$$\arg\min_{\alpha} \sum_{i=1}^{I} (Y_i - F_\alpha(X_i))^2, \text{ s.t. shape constraints.} \tag{3}$$

The solution to Problem 3 will be called the optimal solution.

Shape constraints have been investigated since mid 1990's in the field of 'Computer Graphic Aided Design', CAGD for short, and is a central theme in this area. The theory of shape constraints in CAGD is well developed in Farin (1993) and Peña (1999) for example. This paper borrows some of the ideas of this field, specifically around Chebyshev system of functions, simplexes and corner cutting or refinment algorithms (Gasca and Micchelli, 2013), (Chaikin, 1974).

A common hypothesis in CAGD is that the set of functions $\{f_j(x)\}_{j=0}^{J}$ when x is one dimensional forms an Extended Complete Chebyshev system of functions called ECT system in short (Karlin and Studden, 1966). This will be one of our main hypotheses and will be explicited in the next section 1.2.

For polynomials of more than one variable, Problem 3 remains largely open. This is precisely the purpose of this paper and its main result to tackle the case of multivariable polynomials and more generally of Chebyshev systems. Indeed, with only one variable, methods like Semi-Definite Programming (Ben-Tal and Nemirovski, 2001) (Papp and Alizadeh, 2014) are able to find the optimal estimator in shape constraints problems when F_α is polynomial. However, as stated by Ben-Tal Ben-Tal and Nemirovski (2001), these methods can not describe all the nonnegative polynomials in multivariable cases.

The idea of this paper is to transform the initial non linear shape requirements of Problem 3 in a finite number of linear constraints on the coefficients which approach the same solution. The least square problem is thus transformed in:

$$\arg\min_{\alpha}\sum_{i=1}^{I}(Y_i - F_\alpha(X_i))^2, \text{ s.t. linear constraints} \tag{4}$$

which is a classical convex quadratic programming problem (Nocedal and Wright, 2006).

We proceed iteratively: at step K the set of constraints attached to the previous problem at step $K-1$ is augmented by a finite number of linear new constraints, chosen so that the sequence of solutions of Problem 4 tends to the solution of Problem 3 when the number of steps increases toward infinity. This paper is organized as follows: a state of the art is first developed as a beginning. Notations and reminders of Chebyshev systems theory are introduced in Subsection 1.2. The theory is exposed for monotony constraints, first for functions of only one variable (Section 2), where we prove the convergence of our procedure, detail the subsequent algorithm and discuss its implementation. We then extend our ideas to the multivariable cases (Section 3). Practical considerations are considered in Section 4, where we detail also one industrial case in petroleum engineering related to hydrotreatment of naphta. Conclusions and perspectives are given in Section 5. Additionally, one can find in Appendix A a few properties of Chebyshev systems useful for the proofs. All the proofs are postponed to Appendix B.

1.1 State of the art

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 multivariable situations. Restricting ourselves to monotone regression in more than one dimension, a few performing algorithms have been proposed, based on splines (Ramsay and Silverman, 2005) (Papp and Alizadeh, 2014), on kernel type (Du et al., 2013) regressors, on Generalized Additive Models or GAM (Wood, 2006), or very recently on kriging approximations (Maatouk and Bay, 2017).

However, compared to nonparametric regression, parametric functions are immediate to calculate. They are easier to interpret, showing very clearly the influence of each variable, and their interactions. They depend 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 these parametric approaches is that the expected behavior will be respected everywhere in the domain and not only in the vicinity of the observed points (see Meyer (2012) 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. This is why we believe as in Hawkins (1994), there is still room for parametric regressions and especially for polynomial regression.

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. In contrast to classical least square problems, constrained extensions are also generally very hard to tackle. Even for low degree polynomials, it implies complicated non linear expressions of the coefficients.

Studies on constrained parametric regression have focused on polynomial regression. Taking the derivatives, studies on monotone polynomials reduce to the study of positive polynomials. Polynomials in one variable can be positive first over the entire real line, secondly over a semi-infinite interval, or thirdly on a compact set. In these three situations, Karlin and Studden (Karlin and Studden, 1966) have given a representation theorem. Still the obtained expressions remain highly non linear.

Ben-Tal and Nemirovski (Ben-Tal and Nemirovski, 2001) have shown how to solve the problem via Semi-Definite Programming techniques in the three above situations. Hawkins (Hawkins, 1994) has set out a method based on the observation that if a polynomial has to be monotone on the entire real line, if its first derivative is zero at some x^* then necessarily its second derivative at x^* is also equal to 0. His method is restricted to odd degree polynomials. Murray et al. (2016) have implemented Karlin's three alternatives in the R 'Monopoly' package. By carefully choosing the parametric form of the polynomials and the numerical schema of the calculations, the evaluation of bootstrap confidence intervals for the estimated coefficients are made possible.

To our knowledge however none of these methods can handle multivariable situations. Moreover, they are restricted to polynomials and not extended to Chebyshev system of functions.

1.2 Notations, Definitions and Basic Notions

The upper case letters X_i or Y_i where $i \in [1, I]$ are reserved for the observations. The lower case x or x_v for $v \in [1, V]$ is used for variables. The approximation functions f_j are numbered from 0 to J. Bold upper case letters like $\mathbf{T}$ correspond to matrices, bold lower case letters to vectors.

Regression function. We add here a few complements to the definition of the regression function in (1). For all v, $f_{v,0}(x_v) = 1$. Without $f_{v,0}(x_v)$, we have J_v elementary functions depending solely on x_v. Furthermore each $f_{v,j}(x_v)$ is at least continuous and derivable on $[0, 1]$ as many times as needed, i.e., up to the order J_v.

In the case of a single variable, the notation $F_\alpha^{(k)}(x)$ 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.

Vectorial Notations. In one variable cases, $\mathbf{f}(x)$ refers to the the column vector $\mathbf{f}(x) = {}^t(f_1(x), \cdots, f_J(x))$. 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:

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

These notations are extended to multivariable cases as well, with $\mathbf{f}_{v\bullet}$.

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

$$Z : [0, 1]^J \to \mathbb{R},\ t = (t_1, \cdots, t_J) \to Z(t) = \alpha_0 + \sum_{j=1}^{J} \alpha_j t_j.$$

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

Osculating simplex. In the remainder of this section, we restrict ourselves to the case of one variable only. As it is needed in the sequel we introduce the notion of osculating k-spaces (Peña, 1999) and osculating hyperplanes which are special cases of the former.

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

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

In Computer Aided Design (Farin et al., 2002), 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' (Peña, 1999) and is defined as follows (Gasca and Micchelli, 2013):

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, $j = 1, \cdots, J-1$ are found as the intersections of the osculating j-space at T_0 and the osculating $(J-j)$-space at T_J.*

Two examples of osculating simplexes are shown on the figure 1 below.

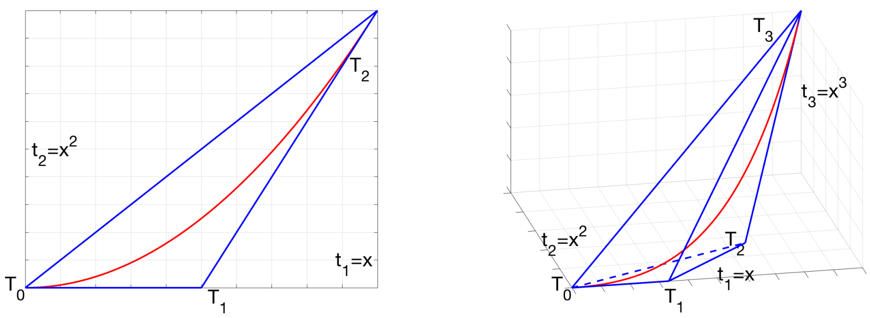

Fig. 1 examples of osculating simplexes Osculating simplex of the curve (x, x^2) on the left panel, and of the curve (x, x^2, x^3) on the right.

Chebyshev system. The study of Bézier curves is intimately linked to the theory of Chebyshev systems of functions (Gasca and Micchelli, 2013), (Schumaker, 2007), (Karlin and Studden, 1966). In the following, the proofs need a particular version called Extended Chebyshev systems referred as ET in Karlin and Studden (1966).

In Theorem 1 in Section 2, the use of ET systems guarantees that C_J will be included in its osculating simplex between any beginning point and any final point chosen in $[0,1]$. This is the heart of our construction, as will be seen in Subsection 2.2.

Because they are more easily characterized than ET systems, as can be seen in Theorem A2, instead of ET-systems, we use a more restricted form called Extended Complete Chebyshev systems, defined in Appendix A and referred as ECT. From now on, we require additionally that:

Assumption 1 *The systems of functions* $\{f_{v,j}(x_v)\}_{j=0}^{J_v}$*, for each* v *in* $[1, V]$*, form an ECT on* $[0, 1]$.

More detailed considerations about Chebyshev systems can be found in Appendix A.

2 Univariate case

In the case of one variable ($V = 1$), we explicit the form of Problem 4. We proceed as follows.

In Subsection 2.1, through Proposition 1 we formalize our analysis. The conditions for which this proposition holds are examined in Theorem 1.

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 1, we detail in Theorem 2 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 (Hawkins, 1994) later in Subsection 4.2.

2.1 Univariate case: Osculating simplexes

We consider a curve C_J and its osculating simplex S_J on $[0, 1]$. The $J + 1$ vertices of S_J are gathered in a matrix $\mathbf{T}$ of dimension $J \times (J + 1)$, where each column is a vertex. To take into account the constant term in the expression of F_α, we then define the squared matrix of constraints $\mathbf{T}_\bullet$ of dimension $(J + 1) \times (J + 1)$ as:

$$\mathbf{T}_\bullet := \begin{pmatrix} {}^t\mathbf{1} \\ \mathbf{T} \end{pmatrix},$$

where $\mathbf{1}$ is a vector of 1. The expression ${}^t\mathbf{T}_\bullet \boldsymbol{\alpha} \geq 0$ means that each coordinate of the vector ${}^t\mathbf{T}_\bullet \boldsymbol{\alpha}$ is non negative.

As a simplex, every point of S_J can be expressed as a linear combination of the vertices with positive coefficients. We thus claim the following proposition.

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

At this point, our aim is to solve the much simpler **Problem 5**, where the non linear constraints of Problem 3 have been replaced by linear constraints.

$$\arg\min_\alpha \sum_{i=1}^{I} (Y_i - F_\alpha(X_i))^2, \text{ s.t. } {}^t\mathbf{T}_\bullet \boldsymbol{\alpha} \geq 0. \tag{5}$$

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 1 *Let T_0 and T_J be two points on the curve C_J. Under Assumption 1, the portion of the curve between T_0 and T_J is included in its osculating simplex.*

We note that choosing the osculating simplex to enclose the curve is a mere continuation of the theory of Bézier curves.

2.2 Algorithm for finding the optimal solution, one variable

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. We propose here 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 1.

Our idea is a variation on a corner cutter or refinement algorithm. These algorithms are known since the mid seventies (Chaikin, 1974) (Schumaker, 2007) and closely linked to Bézier curves (Farin et al., 2002) and B-splines (De Boor, 2001).

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 2 the convergence of the algorithm is stated. This subsection is concluded with a few practical considerations.

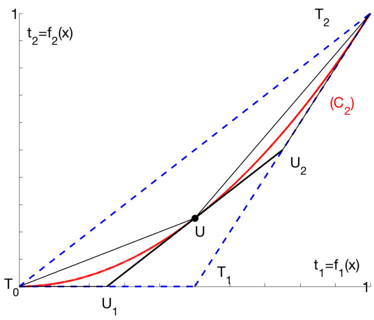

Fig. 2 corner cutting algorithm the simplex $(T_0T_1T_2)$ is replaced by the polytope $(T_0U_1UU_2T_2)$, formed of two simplexes, (T_0U_1U) and (UU_2T_2). The corner T_1 of the initial simplex is cut.

Example in dimension 2. For a short while, we take $J = 2$. In Proposition 1 we established that a 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 2).

But we have restrained ourselves to simplexes. In fact it is easy to obtain a narrower polytope surrounding C_2, if more than 3 vertices are allowed. For example, in Figure 2, the polytope P_2' whose vertices are T_0, U_1, U, 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 confounded with the tangent line to the curve C_2 at 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.

This process of cutting can continue: at each step we split a simplex in two new simplexes, and build a chain of simplexes containing the curve. At each time we cut one of the simplex by a new tangent, remove one corner and add two new vertices.

To speak informally, what we are going to prove, is that when this step is repeated indefinitely, every point of the curve C_2 is transformed in a vertex of a simplex and therefore in a constraint in the problem 5, so that the positivity of the polynomial is ensured everywhere on [0,1].

Generalization: simplex regression. Generalizing this cutting principle to J functions is straightforward. 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. This is the reason of the name given to this method, simplex regression.

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 1 and Theorem 1. The convergence of the cutting algorithm is proved in Theorem 2 which is stated after introducing some necessary notations and proving a preliminary Proposition 1.

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.

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

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

Similarly, we denote $A_{J,K}$ the set of possible coefficients at step K, that is the coefficients for which ${}^t\mathbf{T}_{\bullet K}\boldsymbol{\alpha} \geq 0$ where $\mathbf{T}_{\bullet K}$ is the matrix of constraints: its first row is composed of ones, the rest of the matrix gathers (in columns) the vertices of $P_{J,K}$.

$\tilde{\boldsymbol{\alpha}}_{J,K}$ is the vector of coefficients of the solution to Problem 5 when the constraints match the vertices of $P_{J,K}$. The coefficients of the optimal solution to 3 are stored in a vector denoted $\tilde{\boldsymbol{\alpha}}_J$.

Let $\text{cost}(\boldsymbol{\alpha})$ be defined as $\text{cost}(\boldsymbol{\alpha}) := \sum_{i=1}^{I}(Y_i - F_\alpha(X_i))^2$. We have:

$$\text{cost}(\tilde{\boldsymbol{\alpha}}_{J,K}) = \min_\alpha(\text{cost}(\boldsymbol{\alpha})), \text{ s.t. } {}^t\mathbf{T}_{\bullet K}\boldsymbol{\alpha} \geq 0.$$

In the course of Theorem 2 and in Algorithm 1 below, we make use of the following proposition.:

Proposition 2 *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. The sequence of cost($\tilde{\boldsymbol{\alpha}}_{J,K}$*) is decreasing with* K.

Theorem 2 *Under Assumption 1, we have* $\lim_{K\to\infty} \tilde{\boldsymbol{\alpha}}_{J,K} = \tilde{\boldsymbol{\alpha}}_J$.

The proof consists of observing that $\bigcup_{K\in\mathbb{N}} A_{J,K}$ is dense in A_J.

Algorithm 1. The algorithm which puts Theorem 2 into practice is presented below. As already said, at step K, the problem is solved by means of a quadratic programming algorithm. It is well known that if the solution is not strictly inside the convex constrained region $A_{J,K}$ (see Proposition 1), 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 variable definition domain 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 variables.

The fact that $\text{cost}(\tilde{\boldsymbol{\alpha}}_{J,K})$ is decreasing with K gives an easy stopping criterium for Algorithm 1 which should terminate if the difference in the cost function at steps K and $K+1$ is lighter than c a small constant chosen a priori.

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.

corner cutting algorithm in the univariate case

- **while** $cost(\tilde{\boldsymbol{\alpha}}_{J,K}) - cost(\tilde{\boldsymbol{\alpha}}_{J,K+1}) > c$ **do**
  - **for** *each* q *in* $[1, Q_K]$ **do**
    - find the simplex in which T_q is a vertex;
    - choose x_{new} a value of the variable between $X_{q,0}$ and $X_{q,J}$; define T_{new} the corresponding point on the curve;
    - create two new simplexes: the first simplex finishes at T_{new}, the second one begins at T_{new};
    - remove the vertices of the old simplex;
    - gather all the remaining vertices in a matrix;

    end
  - $K = K + 1$
  - Resubmit problem 5 to the fitting algorithm, with these new constraints.

end

Algorithm 1: univariate case

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 needed repeatedly. This is detailed in Lemma 1 in Appendix B, as a preliminary to Theorem 1 in the general case of ET systems. We also note that with the sequence of monomials $\{x^j\}_{j=1}^J$, the vertices of the osculating simplex can be calculated analytically.

Number of constraints. Counting the number of constraints added each time we cut a corner gives an idea of the effort required by the algorithm.

At each step, we replace the old simplex by two new simplexes, which have a vertex in common. The number of vertices is thus augmented by $2 \times (J+1) - (J+1) - 1 = J$ at each step.

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 $(f_1(x_k), \cdots, f_J(x_k))$.

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, 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 in Proposition 3 below.

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

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

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

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

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

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.

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

Theorem 3 *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 $\dfrac{x_0 + x_J}{2}$.*

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

3 Multivariable case

In case of multivariable functions, we proceed in two successive steps. First, we generalize the previous methodology of Section 2 in one dimension to this new situation and conclude this subsection with Theorem 4, which transposes Proposition 1 to multivariable functions. As with a single variable, the proposed constraints are only sufficient conditions. In the second step, we propose an algorithm in Section 3.2 capable of finding the optimal solution. Its convergence is proved in Theorem 5

3.1 Multivariable case: circumscribing simplexes

We switch to a more general situation, where $x = (x_1 \cdots x_V)$ is V-dimensional.

Our problem is to determine the vector of coefficients $\boldsymbol{\alpha}$, 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 of $F_\alpha(x)$ amounts to checking that $Z(T_j) \geq 0$, for all $j \in [1, J]$. We bring together all the vertices in a matrix $\mathbf{T}$ and compose the matrix of constraints $\mathbf{T}_\bullet$ by adding to $\mathbf{T}$ a first row of 1 to include the coefficient α_0 in the set of constraints. The problem to solve in dimension V can be rephrased as **Problem 6**:

$$\arg\min_{\alpha} \sum_{i=1}^{I} (Y_i - F_\alpha(X_i))^2, \text{s.t. constraints } \ {}^t\boldsymbol{\alpha}\mathbf{T}_\bullet \geq 0. \tag{6}$$

which is the analog of Problem 5, the only difference being that X_i is now V-dimensional.

To extend the previous results from dimension 1 to V dimensions and control the number of constraints, we proceed by means of tensor products. Specifically, recalling that $F_\alpha(x)$ can be written $F_\alpha(x) = \langle \alpha, \mathbf{f}_\bullet(x) \rangle$ we assume that:

$$\mathbf{f}_\bullet(x) = \mathbf{f}_{1\bullet}(x_1) \otimes \cdots \otimes \mathbf{f}_{V\bullet}(x_V). \tag{7}$$

This first requirement for $\mathbf{f}_\bullet(x)$ will be softened later on.

Products of tensors are applied as well to the matrix of constraints.

Let T_{v,j_v} for $j_v = [0, J_v]$ be the vertices of the osculating simplex containing the curve $C_{J,v} = (f_{v,1}(x_v), \cdots, f_{v,J_v}(x_v))$, where $x_v \in [x_{v,0}, x_{v,1}]$. The matrix $\mathbf{T}_v$ of dimension $J_v \times (J_v + 1)$ contains in columns the vertices T_{v,j_v}. Adding a first row of 1 to each of the $\mathbf{T}_v$, we obtain the matrices of constraints $\mathbf{T}_{v\bullet}$ of dimension $(J_v + 1) \times (J_v + 1)$ for each variable. The matrix of constraints $\mathbf{T}_\bullet$ on the domain $D = [x_{1,0}, x_{1,1}] \times \cdots, \times [x_{V,0}, x_{V,1}]$ is defined as the tensor product:

$$\mathbf{T}_\bullet := \bigotimes_{v=1}^{V} \mathbf{T}_{v\bullet}.$$

Setting $J + 1 = \prod_{v=1}^{V} (J_v + 1)$, the dimension of $\mathbf{T}_\bullet$ is $(J + 1) \times (J + 1)$. We quote also that the first row of $\mathbf{T}_\bullet$ is composed of 1. The J remaining rows form a matrix denoted $\mathbf{T}$.

Each column of $\mathbf{T}$ corresponds to a point T_j in the space $\mathbb{T} = [0,1]^J$. We define the polytope P_J as the convex hull of the set of vertices T_j. With $J+1$ vertices, this polytope is a simplex and contains the part of the manifold C_J corresponding to the domain D as stated in the following theorem 4 .

Theorem 4 joins together Proposition 1 and Theorem 1, transposes their statement to multivariable situations and gives a means to automatically generate the needed constraints.

Theorem 4 *Under Assumption 1*

1. *When x traverses D, the corresponding portion of C_J is included in P_J.*
2. *If ${}^t\boldsymbol{\alpha}\mathbf{T}_{\bullet} \geq 0$, then $\forall x \in D$, we have $F(x) \geq 0$.*

Dropping terms. Actually, a function $F(x)$ containing all the terms resulting from the tensor product $f_{1\bullet}(x_1) \otimes \cdots \otimes f_{V\bullet}(x_V)$ is of little practical use. If it is not possible to 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.

However, dropping some terms amounts to taking 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.

3.2 Algorithm for finding the optimal solution, multivariable case

In case of one variable, the proposed algorithm is based on the notion of osculating hyperplanes. In multivariable 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 $\mathbf{T}_{v\bullet,K}$ (see below). The columns of each of these matrices $\mathbf{T}_{v\bullet,K}$ correspond to the vertices of a simplex for the matching variable. We note that the resulting tensor product corresponds also to a polytope.

When $v = 1$, in Algorithm 1, we have replaced the initial simplex by a chain of simplexes (see figure 2). We keep the same procedure when $v > 1$, except that now we create a mesh of simplexes rather than a chain. This point will be detailed when developing Algorithm 2 below.

Let $C_{J_v} = (f_{v,1}(x_v), \cdots, f_{v,J_v}(x_v))$ be the curve corresponding to the variable v. At step K, for each v, we build a chain $P_{v,J_v,K}$ of simplexes containing C_{J_v}, gather all the vertices of $P_{v,J_v,K}$ in a matrix $\mathbf{T}_{v,K}$ and form $\mathbf{T}_{v\bullet,K}$ the matrix of constraints for C_{J_v} at step K by adding a row of 1.

We then generate the tensor products of all these matrices

$$\mathbf{T}_{\bullet K} = \bigotimes_{v=1,V} \mathbf{T}_{v\bullet,K}.$$

Excluding the first row, we obtain the matrix $\mathbf{T}_K$ containing the coordinates of the vertices on which we must check the positivity of the corresponding function $Z(t_1, \cdots, t_V)$.

As previously in Section 2.2, let $P_{J,K}$ be the polytope whose vertices are the columns of $\mathbf{T}_K$, and $\tilde{\boldsymbol{\alpha}}_{J,K}$ be the solution of Problem 6 when the constraints are issued from the vertices of $P_{J,K}$. That is:

$$\tilde{\boldsymbol{\alpha}}_{J,K} = \arg\min_{\alpha} \sum_{i=1}^{I} (Y_i - F_\alpha(X_i))^2, \text{s.t. constraints } \ {}^t\boldsymbol{\alpha}\mathbf{T}_{\bullet K} \geq 0.$$

Analogously to Theorem 2, we examine $F_J(x)$ the optimal solution to 3 and $\tilde{\boldsymbol{\alpha}}_J$ its vector of coefficients. Our aim is the following theorem:

Theorem 5 *Under Assumption 1,* $\lim_{K\to\infty} \tilde{\boldsymbol{\alpha}}_{J,K} = \tilde{\boldsymbol{\alpha}}_J$.

The proof is similar to the previous one in Theorem 2 with the generalization to the tensorial product of constraints.

Algorithm 2. We illustrate the refinement schema of Algorithm 2 in two dimensions before giving a general formulation.
Refinement schema with 2 variables. The key to Algorithm 2 is Theorem 4. The manifold C_J represents the function $\mathbf{f}_\bullet(x) = \mathbf{f}_{1\bullet}(x_1) \otimes \mathbf{f}_{2\bullet}(x_2)$ on $D = [0,1]\times[0,1]$. Choosing two arbitrary values x_1^* and x_2^* for the variables, we can refine D in 2^2 subdomains: $D_1 = [0, x_1^*]\times[0, x_2^*]$, $D_2 = [0, x_1^*]\times[x_2^*, 1]$, $D_3 = [x_1^*, 1]\times[0, x_2^*]$ and $D_4 = [x_1^*, 1] \times [x_2^*, 1]$.

On each of these subdomains, using Theorem 4, we know how to build a simplex including a portion of C_J. Obviously the four simplexes taken together include the whole manifold C_J, and any of these subdomains can be subdivided independently of the others.
Solving. The generalization of the previous refinement schema to any number of variables is straightforward. The algorithm for solving problem 6 is an extension of Algorithm 1 to more than one variable. The only difference is that when subdividing one simplex, we create 2^V new simplexes instead of two when $V = 1$.

In Algorithm 2, at each step K, Q_K constraints are supposed to be active. If the constraint q is one of them, it should be removed at the next step. To do this, since this constraint matches a vertex T_q of one of the simplexes containing C_J, we simply identify the subdomain containing T_q, split it in 2^V new hypercubes, and create a simplex in each of these hypercubes.

corner cutting algorithm in the multivariable case

- **while** *cost*$(\tilde{\boldsymbol{\alpha}}_{J,K}) -$ *cost*$(\tilde{\boldsymbol{\alpha}}_{J,K+1}) > c$ **do**
  - **for** *each q in* $[1, Q_K]$ **do**
    - find the simplex S_q in which T_q is a vertex;
    - choose x_{new} a new value in the domain corresponding to S_q; define T_{new} the corresponding point on the curve;
    - create 2^V new simplexes connected at T_{new};
    - remove the vertices of the old simplex;
    - gather all the remaining vertices in a matrix;

    end
  - $K = K + 1$
  - Resubmit problem 6 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.

Number of constraints The number of constraints corresponding to one of the simplexes containing C_J is its number of vertices:

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

This leads to the following proposition.

Proposition 5 *When creating a new simplex by subdividing an existing one, the number of constraints is augmented by*

$$3^V - 2^{2V} + (J + 1) * (2^V - 1).$$

When $V = 1$, Proposition 4 gives the result already detailed in the single variable case.

4 Examples

In this section we begin by enumerating the situations where our method od simplex regression can be used (Subsection 4.1). The case of functions of a single variable is illustrated in Subsection 4.2 with Hawkin's example, and with a sum of exponentials. We continue with one industrial example in multivariable settings (see 4.3).

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. 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.
3. Obviously, every monotone transformation of the variables $x_1, \cdots, x_v$ will not change the procedure.

4.2 Two examples with a single variable

In Figure 3, we illustrate our approach with the simulation data proposed by Hawkins (Hawkins, 1994). 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

5 degree polynomial and Hawkin's values

	lower	est.	upper	Hawkin
β_5	6.087	11.332	16.332	10.99
β_4	-22.927	-21.413	-19.633	-21.42
β_3	0.8264	6.850	12.945	7.29
β_2	20.694	22.178	23.352	22.18
β_1	7.163	8.701	10.238	8.59
β_0	0.662	0.991	1.336	0.99

4 degree polynomial

	lower	est.	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.

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 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 with different draws of ϵ to give an idea of the distributions of the estimators. In Table 1 the columns 'lower' and 'upper' give the 5% and 95% percentiles.

Not reported here because the results are very similar, we have compared our method of simplex regression to Murray and coauthor' algorithms (Murray et al., 2016) who have trained their method on the same data set.

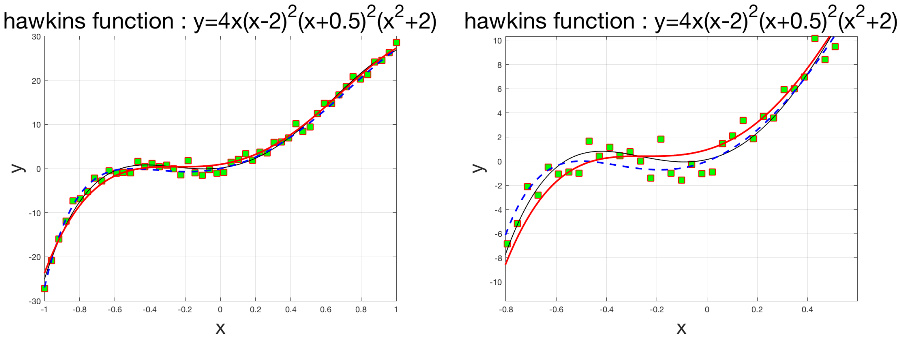

Fig. 3 Hawkins's function In squared green, the observed points. In red, the fit. In dashed black, the least square approximation with a polynomial of degree 4. The right panel shows the resulting function on a restricted interval

We continue with an example which makes use of exponential functions, compared to a polynomial of degree 5. The observed points, exactly the same in the left and right figures, are random and show a shape similar to a sigmoid. The exponents in the exponentials are completely arbitrary. In both cases, the unconstrained fit exhibits a non monotone behavior around the origin.

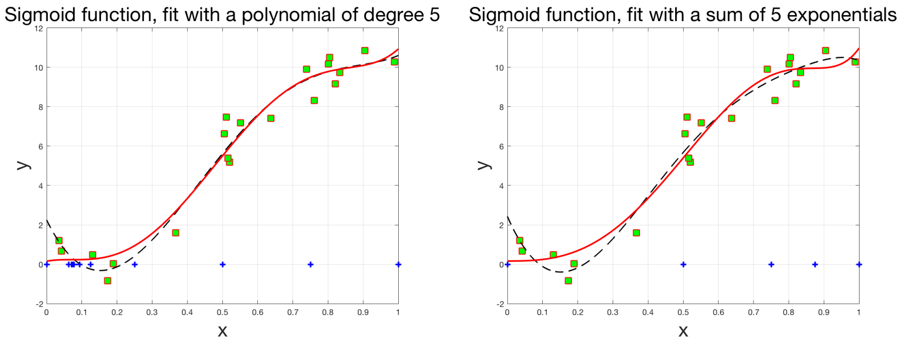

Fig. 4 sigmoid function In squared green, the observed points. In red, the fit. In dashed black, the non restricted least square approximation. The blue crosses indicate the limits on the x axis of each simplex. On the left panel, we use a 5 degree polynomial. On the right, it is a sum of 5 arbitrarily chosen exponentials, $\exp(0.5x), \exp(1.2x), \exp(2x), \exp(2.1x), \exp(2.5x)$.

4.3 Real example: hydrotreatment of naphta

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.

Finally, a degree 2 polynomial of 4 variables is proposed to approximate this process, where:
the response is $y = \log(-\log(\frac{C}{C_0}))$, with C the concentration of the chemical to be removed remaining at the outlet of the reactor and C_0 its initial concentration;
$x_1 = 1/T$, with T the temperature of the process;
$x_2 = \log(VVH)$, VVH being the Velocity per Volume and per Hour;
$x_3 = \log(P_{H_2})$, where P_{H_2} is the partial hydrogen pressure;
$x_4 = \log(P_{H_2S})$, with P_{H_2S} the partial H_2S pressure.

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
- VVH decreases or x_2 increases
- P_{H_2} or x_3 increases.
- P_{H_2S} or x_4 increases.

Figure 5 compares the results when regressing with and without constraints. The left panel exhibits the residues (y calculated - y experimental), showing minor differences when the experimental points are predicted by both methods: the root mean squared errors is $RMSE = 0.485$ with constraints and $RMSE = 0.411$ without. But the obtained equations are really different as shown on the right.

On the right panel, the plot shows the behavior of the response when only one variable varies at a time, starting from a given point in the domain which can be read on the figure. 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 all the curves meet at the center of the graphic. Black lines correspond to variations along T or x_1, red lines to VVH or x_2, blue lines to P_{H_2} or x_3, green ones to P_{H_2S} or x_4. The behaviors for the regression without constraints are obviously wrong: the blue dotted line is decreasing instead of increasing and the green has a maximum.

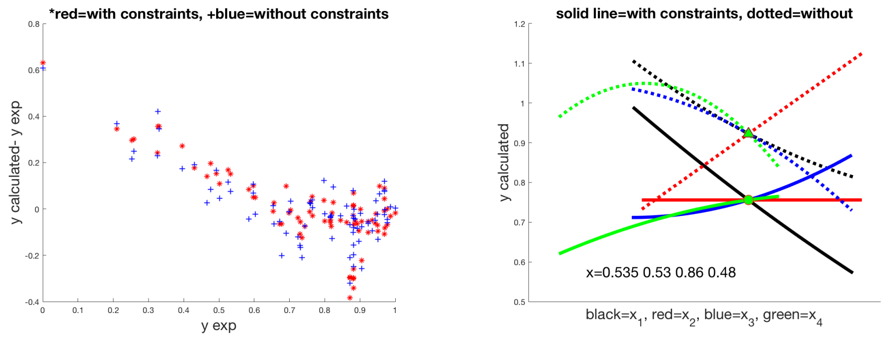

Fig. 5 polynomial fit to the data of HDS experiments Residue diagram for the HDS data on the left panel. On the right the plot compares the UNconstrained and constrained regressions.

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. It is also valid with any other ECT Chebyshev systems of functions. Most importantly, our method will give satisfactory results in multidimensional cases even with few available experimental data.

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 Trevor et al. (2009).

A second enhancement would be to find a way for limiting the number of the constraints in multivariable situations. Indeed, their number grows exponentially with the number of variables. This certainly is a bottleneck of the method.

Thirdly, the scope of this kind of regression could be extended to nonparametric regressions. GAMs are natural good candidates as well as local polynomial regression (Fan and Gijbels, 1996).

Fourth, uncertainty intervals are certainly an issue for this method. Indeed, as the constraints change at each iteration, the residues can not be considered as identically distributed, so that bootstrap algorithms are not adequate at first sight.

The original algorithms are developed in Matlab® and available upon request.

Appendix A: Chebyshev system

The study of Bézier curves is intimately linked to the theory of Chebyshev systems (Gasca and Micchelli, 2013), (Schumaker, 2007), (Karlin and Studden, 1966). 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 traditionaly denoted

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

Definition A1 *The functions $f_0, f_1, \cdots, f_J$ 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*

$$\forall x_0 \le \cdots \le x_J \in [0, 1], M\begin{pmatrix} x_0, \cdots, x_J \\ f_0, \cdots, f_J \end{pmatrix} > 0.$$

Definition A2 *The functions $f_0, \cdots, f_J$ are called an extended complete Chebyshev system on $[0, 1]$, or ECT-system if*

$$\forall 1 \le j \le J, \forall x_0 \le \cdots \le x_J \in [0, 1],\ M\begin{pmatrix} x_0, \cdots, x_j \\ f_0, \cdots, f_j \end{pmatrix} > 0.$$

In Definition A2, when passing from an ET to an ECT, J is replaced by j and x_0 (resp. $\cdots x_j$) can be repeated at most j times.

A consequence of definitions A1 and A2 which will be useful for the proofs in Appendix B is that all the columns vectors of the determinant $M\begin{pmatrix} x_1, \cdots, x_J \\ f_1, \cdots, f_J \end{pmatrix} > 0$ 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 A3 *Let H be a hyperplane in dimension J containing the point $(a_1, \cdots, a_J)$, and spanned by $J-1$ vectors ${}^t(\alpha_{j,1}, \cdots, \alpha_{j,J})_{j=1,J-1}$. Denoting*

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

the equation of H is given by

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

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

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 A1 *(extracted from Karlin and Studden (1966), chap I first part of Theorem 4.3) Let $Z(F_\alpha)$ be the number of zeros of F_α on $[0,1]$ counting multiplicities. If one at least of the coordinates of $\boldsymbol{\alpha}$ is different from zero, then $Z(F_\alpha) \leq J$.*

The following specialization is well known in GCAD.

Corollary A1 *Let $f_0(x) = 1$ for all $x \in [0,1]$. If $f_0, f_1, \cdots, f_J$ is an ET-system on $[0,1]$, then any hyperplane in $\mathbb{T}$ intersects the curve C_J on $[0,1]$ at most J times counting multiplicities.*

The second theorem (see Karlin and Studden (1966),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

$$W_{f_0,\ldots,f_j}(x) := \begin{vmatrix} f_0(x) & \cdots & f_j(x) \\ f_0^{(1)}(x) & \cdots & f_j^{(1)}(x) \\ & \cdots & \\ f_0^{(j-1)}(x) & \cdots & f_j^{(j-1)}(x) \end{vmatrix}.$$

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

Theorem A2 *(extracted from Karlin and Studden (1966)) Let $f_0, f_1, \cdots, f_J$ be of class C^J on $[0,1]$. Then $f_0, \cdots, f_J$ is an ECT-system on $[0,1]$ if and only if for $j = 0, \cdots, J$ we have $W_{f_0,\ldots,f_j}(x) > 0$ on $[0,1]$.*

Examples. Theorem A2 gives a means to easily check that a set of functions is an ECT.

Since in our notation $f_0(x) = 1$, a direct consequence of $\{f_j(x)\}_{j=0}^J$ being an ECT, is that the functions $f_1^{(1)}(x), \cdots, f_J^{(1)}(x)$ form also an ECT on $[0,1]$.

The following systems of functions are easily proved to be ECT:

1. $f_j(x) = x^{d_j}$, with a sequence of increasing positive real d_j verifying $d_1 = 1$ and $d_1 < \cdots < d_j$.
2. $f_j(x) = x^j$, i.e. the functions f_j form a sequence of monomials.
3. $f_j(x) = \exp(d_j x)$.

Appendix B: proofs

Preliminaries to Theorem 1

Theorem 1 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 $T_0, T_1, \cdots, T_J$ the vertices of the osculating simplex between x_0 and $x_0 + h$ (see Definition 2).

The vectors $\mathbf{f}^{(1)}(x_0), \cdots, \mathbf{f}^{(j)}(x_0)$ are all linearly independent. This results from the definition A1 of an Extended Chebyshev system. In this basis,

$$\overrightarrow{T_0T_j} = \sum_{k=1}^{j} \gamma_{k,j}\mathbf{f}^{(k)}(x_0). \tag{8}$$

Similarly,

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

Lemma 1 *Let D_j be the determinant*

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

$D_{j,k}$ 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 $\gamma_{k,j} = \dfrac{D_{j,k}}{D_j}$.

All the determinants D_j are strictly positive as a result of the definition A1 of ET systems.

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 $\gamma_{1,j}, \cdots, \gamma_{J,j}$ of T_j, as stated in this lemma, result from the Cramer'rule applied to the linear system of equations:

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

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

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 $f_1(x_0+h) - f_1(x_0)$,...,$f_J(x_0+h) - f_J(x_0)$, 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 $\dfrac{h^k}{k!}D_j$. 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 $x_J > x_0$. Then, in the neighborhood of x_0, for h small enough, the portion of the curve C_J corresponding to x varying in $(x_0, x_0 + h)$ is strictly included in the cone generated by $\{\overrightarrow{T_0T_1}, \cdots, \overrightarrow{T_0T_J}\}$.*

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

$$\overrightarrow{T_0T_x} = \sum_{j=1}^{J} \lambda_j(x)\overrightarrow{T_0T_j}, \text{ s.t. } \lambda_j(x) \geq 0, \forall j \in [1, j], \forall x \in [x_0, x_0 + h], \tag{9}$$

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

For every $0 < j < J$, $\overrightarrow{T_0T_j}$ belongs to the osculating j-space at T_0, and $\overrightarrow{T_0T_j}$ can be written: $\overrightarrow{T_0Tj} = \sum_{k=1}^{j} \gamma_{k,j} \mathbf{f}^{(k)}(x_0)$. Gathering the coefficients $\gamma_{k,j}$ for $k \leq j$ in a matrix $\mathbf{\Gamma}_h$, we obtain the system of linear equations:

$$\begin{pmatrix} {}^t\overrightarrow{T_0T_1} \\ {}^t\overrightarrow{T_0T_2} \\ \cdots \\ {}^t\overrightarrow{T_0T_J} \end{pmatrix} = \mathbf{\Gamma}_h \begin{pmatrix} {}^t\mathbf{f}^{(1)}(x_0) \\ {}^t\mathbf{f}^{(2)}(x_0) \\ \cdots \\ {}^t\mathbf{f}^{(J)}(x_0) \end{pmatrix}. \tag{10}$$

Furthermore, a Taylor expansion of $\overrightarrow{T_0T_x}$ gives

$$\overrightarrow{T_0T_x} = \sum_{j=1}^{J} \frac{(x-x_0)^j}{j!} \mathbf{f}^{(j)}(x_0) + o((x-x_0)^J). \tag{11}$$

Plugging together equations (9), (10) and (11), we obtain:

$$ {}^t\mathbf{\Gamma}_h \begin{pmatrix} \lambda_1(x) \\ \cdots \\ \lambda_J(x) \end{pmatrix} = \begin{pmatrix} \frac{(x-x_0)}{1!} + o((x-x_0)) \\ \cdots \\ \frac{(x-x_0)^J}{J!} + o((x-x_0)^J) \end{pmatrix}. \tag{12}$$

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

$$\mathbf{\Gamma}_h = \begin{pmatrix} \frac{h}{1!} + o(h) & 0 & & \cdots \\ \frac{h^2}{2!} + o(h^2) & \frac{h^2}{2!} + o(h^2) & 0 & \cdots \\ \cdots & & & \\ \frac{h^J}{J!} + o(h^J) & \cdots & \cdots & \frac{h^J}{J!} + o(h^J) \end{pmatrix}. \tag{13}$$

Equation (13) is rewritten:

$$\mathbf{\Gamma}_h \sim \mathbf{N}_h \mathbf{\Gamma}'(1 + o(1)). \tag{14}$$

The matrix $\mathbf{N}_h = \begin{pmatrix} \frac{h}{1!} & & \\ & \ddots & \\ & & \frac{h^J}{J!} \end{pmatrix}$ is diagonal and $\mathbf{\Gamma}' = \begin{pmatrix} 1 & 0 & \cdots & \cdots \\ 1 & 1 & 0 & \cdots \\ & & \ddots & 0 \\ 1 & 1 & \cdots & 1 \end{pmatrix}$ is lower triangular and does not depend on h.

Assembling equations (12) and (14), solving $\mathbf{\Gamma}_h$ when h is small enough but not equal to zero, and finally simplifying gives:

$$\begin{pmatrix} \lambda_1(x) \\ \cdots \\ \lambda_J(x) \end{pmatrix} = \begin{pmatrix} \frac{1!}{h} & -\frac{1!}{h} & 0 & \cdots & 0 \\ 0 & \frac{2!}{h^2} & -\frac{2!}{h^2} & \cdots & 0 \\ \cdots & & & & -\frac{(J-1)!}{h^{J-1}} \\ 0 & \cdots & & 0 & \frac{J!}{h^J} \end{pmatrix} \begin{pmatrix} \frac{(x-x_0)}{1!} + o((x-x_0)) \\ \cdots \\ \frac{(x-x_0)^J}{J!} + o((x-x_0)^J) \end{pmatrix}.$$

Eventually, we obtain a positive approximation for $\lambda_j(x)$, $1 \leq j \leq J$, in the vicinity of x_0:

$$\lambda_j(x) = \frac{(x-x_0)^j}{(x_J - x_0)^j}(1 - \frac{1}{j+1}(x-x_0)) + o((x-x_0)^j).$$

Theorem 1

Proof 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 (see Definition 2). We define Face_j for $0 \le j \le J$ as the face of the osculating simplex containing all the vertices except T_j.

For $0 < j < J$, by construction of the osculating simplex, the face Face_j is supported by the vectorial sub-space spanned by the first $j-1$ derivatives at T_0 and the $J-j-1$ derivatives at T_J. For $j=0$ or $j=J$, Face_j is the osculating hyperplane at T_J and T_0 respectively.

This amounts to saying that the multiplicity of the contact between C_J and any Face_j at T_0 is j. Similarly, the multiplicity of the contact between C_J and Face_j at T_J is $J-j$. Finally, C_J intersects Face_j J times. Due to Corollary A1, T_0 and T_J are the only intersection points between C_J and Face_j.

As a conclusion, between T_0 and T_J, C_J stays on one side of each of the faces Face_j for $0 \le j \le J$.

S_J can be viewed as the cone generated by $\{\overrightarrow{T_0T_1}, \cdots, \overrightarrow{T_0T_J}\}$ sectioned by the face Face_0. From Lemma 3, when x_J is fixed, in a small neighborhood of x_0, we know that C_J is inside the cone. Since C_J never crosses one of the face $Face_j$ except in T_0 and T_J, C_J remains inside this cone.

Preliminaries to Theorem 2

Proposition 1

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

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

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 2.

Thus, if we have $\langle T, \boldsymbol{\alpha}\rangle \ge 0$ 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 $\forall x \in [0,1], F_\alpha(x) \ge 0$, 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 $\forall x\ F(x) \ge 0$ 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 $F(x) \ge 0$ 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 $p \in [0,1]$.

The set $B = \{\boldsymbol{\alpha} \mid \forall x \in [0,1], F_\alpha(x) \ge 0\}$ is closed: we consider the application g_x defined as $\boldsymbol{\alpha} \in \mathbb{R}^{J+1} \mapsto g_x(\boldsymbol{\alpha}) \in \mathbb{R}$, $g_x(\boldsymbol{\alpha}) = \langle \mathbf{f}_\bullet(x), \boldsymbol{\alpha}\rangle$. g_x is continuous. The inverse image of the open set $\mathbb{R}^{-*} = (-\infty, 0)$, $g_x^-(\mathbb{R}^{-*})$, is then open and $C = \bigcup_{x\in[0,1]} g_x^-(\mathbb{R}^{-*})$ is also open. C being the complement of B in $\mathbb{R}^J$ is closed.

item 3 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 $A_{J,K+1}$.

We restrict our attention to the sequence $P_{J,K}$ built as a chain of simplexes S_k starting at $x = (k-1)/2^K$ and finishing at $x = k/2^K$ with k varying from 1 to 2^K. We first observe that the distance from any point of $P_{J,K}$ to the curve C_J can be made as small as needed: more precisely,

Lemma 4 $\forall \epsilon > 0, \exists K \in \mathbb{N}$ *such that* $\forall u \in P_{J,K},\ \exists t \in C_J$ *for which* $\|u - t\| < \epsilon$.

Proof To prove this claim, we choose u in $P_{J,K}$, and we restrict our attention to the simplex S_k containing u. The maximum distance of two points within S_k is one of the distances between two of its vertices. By means of Equation (8) and Lemma 2, when K is sufficiently large, calling T_{j_1} and T_{j_2} two of the vertices of S_k, 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 $\dfrac{M}{2^K} = \sum_{j=1}^{J} \sup\limits_{x\in[0,1]} \|\mathbf{f}^{(j)}(x)\|$.

Theorem 2

Proof We denote $B = \overline{\bigcup\limits_{K\in\mathbb{N}} A_{J,K}}$. Our goal is first to prove that $B = A_J$, or in other words that the sequence of sets $\bigcup\limits_{K\in\mathbb{N}} A_{J,k}$ is dense in A_J.

The inclusion $B \subset A_J$ is immediate, as a consequence of items 1 and 2 of Proposition 1. Conversely, we have to prove that every point of A_J is attained. We choose $\boldsymbol{\alpha}$ in A_J and want to show that $\boldsymbol{\alpha} \in \overline{\bigcup\limits_{K\in\mathbb{N}} A_{J,K}}$.

Starting from the vector $\boldsymbol{\alpha} \in A_J$, $\boldsymbol{\alpha} = {}^t(\alpha_0, \alpha_1, \cdots, \alpha_J)$, for any positive integer l we define $\boldsymbol{\alpha}_l$ as $\boldsymbol{\alpha}_l = {}^t(\alpha_0 + \dfrac{1}{l}, \alpha_1, \cdots, \alpha_J)$. $\boldsymbol{\alpha}_l$ belongs to A_J:

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

If we exhibit now an index K_l for which $\boldsymbol{\alpha}_l$ simultaneously belongs to A_{J,K_l}, our assertion is proved: $\boldsymbol{\alpha}$ will be the limit of a sequence of $\boldsymbol{\alpha}_l$ each of them taken in one A_{J,K_l}.

The way to achieve this goal is to consider the sequence P_{J,K_l} of Lemma 4. P_{J,K_l} is built as a chain of simplexes S_k for k varying from 1 to 2^{K_l}. Picking a point u in P_{J,K_l} we examine now what is the condition for which $\langle u, \boldsymbol{\alpha}_l\rangle > 0$.

We start from the identity

$$\langle u, \boldsymbol{\alpha}_l\rangle = \langle u - t, \boldsymbol{\alpha}_l\rangle + \langle t, \boldsymbol{\alpha}_l\rangle.$$

- We observe that $\langle t, \boldsymbol{\alpha}_l\rangle > 1/l$.
- By Cauchy-Schwartz inequality, using Lemma 4, $\langle u - t, \boldsymbol{\alpha}_l\rangle \geq -\dfrac{M}{2^K}\|\boldsymbol{\alpha}_l\|$.
- By the triangular inequality, $\|\boldsymbol{\alpha}_l\| \leq \dfrac{1}{l} + \|\boldsymbol{\alpha}\|$.

Eventually,

$$\langle u, \boldsymbol{\alpha}_l\rangle \geq -\frac{M}{2^{K_l}}(\frac{1}{l} + \|\boldsymbol{\alpha}\|) + \frac{1}{l}.$$

For a given l, K_l is chosen so that the right part of the previous inequality be positive. Since it is true for any $u \in S_k$ and for any k, we have $\boldsymbol{\alpha}_l \in A_{J,K}$, which permits to conclude that $B = A_J$.

Thus, $\tilde{\boldsymbol{\alpha}}_J$ the optimal solution to Problem 3, as an element of A_J, is the limit of a sequence of vectors $\boldsymbol{\alpha}_{J,K}$, each of them taken in one $A_{J,K}$. The second step is to extend this first result to the sequence of $\tilde{\boldsymbol{\alpha}}_{J,K}$, the solutions to Problem 5.

As $A_{J,K} \subset A_J$,we have

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

This proves that $\text{cost}(\boldsymbol{\alpha}_{J,K})$ converges toward $\text{cost}(\tilde{\boldsymbol{\alpha}}_J)$.

The function cost is convex. We call $\boldsymbol{X}$ the matrix of the model, $\boldsymbol{X} = \begin{pmatrix} \mathbf{f}_\bullet(X_1) \\ \cdots \\ \mathbf{f}_\bullet(X_n) \end{pmatrix}$.
The hessian matrix of the function cost is ${}^t\boldsymbol{X}\boldsymbol{X}$. Assuming that ${}^t\boldsymbol{X}\boldsymbol{X}$ is definite positive, which is the usual assumption in regression problems, we can infer that $\tilde{\boldsymbol{\alpha}}_J$ is also the limit of the sequence of the solutions $\tilde{\boldsymbol{\alpha}}_{J,K}$ of Problem 5.

Proposition 2

Proof 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 strictly increasing while $V(x, x_J)$ is strictly decreasing.

Proposition 3

Proof Indeed, the volume of a simplex with vertices $(T_j)_{j=1,J}$ is known to be:

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

Taking the notation of Lemma 1, for $j < J$, $\overrightarrow{T_0Tj}$ is decomposed in

$$\overrightarrow{T_0Tj} = \sum_{k=1}^{j} \frac{D_{j,k}}{D_j}\mathbf{f}^{(k)}(x_0).$$

Standard manipulations on determinants give the expected result.

Preliminary to Theorem 3 We start by showing the following lemma, where the symbol $\propto$ means 'is proportional to'.

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

Proof Restarting from Equation (8), when $f_j(x) = x^j$, the coefficients $\gamma_{k,j}$ of Lemma 2 become exactly $\gamma_{k,j} = \dfrac{(x_J - x_0)^k}{k!}$.

Recalling that

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

we see that $V(x_0, x_J) \propto \prod_{j=1}^{J} \gamma_{j,j}$, which gives the expected result.

Theorem 3

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

Theorem 4

Proof We only have to prove item 1. Item 2 is immediate since P_J is convex by construction.

We recall that $F_\alpha(x)$ can be expressed by $F_\alpha(x) = \langle \boldsymbol{\alpha}, \mathbf{f}_\bullet(x)\rangle$, where $\mathbf{f}_\bullet(x)$ results from the tensor product

$$\mathbf{f}_\bullet(x) = \mathbf{f}_{1\bullet}(x_1) \otimes \cdots \otimes \mathbf{f}_{V\bullet}(x_V). \tag{15}$$

This tensor product gives $J + 1$ terms. We rewrite $\mathbf{f}_\bullet(x)$ as

$$\mathbf{f}_\bullet(x) = {}^t\left(f_0(x), f_1(x), \cdots, f_J(x)\right).$$

C_J is described by $C_J = (f_1(x), \cdots, f_J(x))$, when x traverses $[0, 1]^V$.

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

We call T_j the vertices of P_J. By construction, each vertex T_j can be extracted from the column number j of the matrix of constraints $\mathbf{T}_\bullet$ after removing the first coordinate, equal to 1.

Our aim is to exhibit $J+1$ non negative coefficients μ_j, for $j = 0, J$, summing to 1, such that

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

This equation can be extended to the columns of $\mathbf{T}_\bullet$, and is equivalent to

$$\begin{pmatrix}1\\ T_{x^*}\end{pmatrix} = \sum_{j=0}^{J} \mu_j \begin{pmatrix}1\\ T_j\end{pmatrix}.$$

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

C_{v,J_v} is included in its osculating simplex. Then we can find $J_v + 1$ positive coefficients λ_{v,j_v} summing to 1 such that:

$$\begin{pmatrix}1\\ T_{v,x_v^*}\end{pmatrix} = \sum_{j_v=0}^{J_v} \lambda_{v,j_v} \begin{pmatrix}1\\ T_{v,j_v}\end{pmatrix}. \tag{16}$$

Stemming from Equation 15, by means of the tensor product, we have:

$$\begin{pmatrix} 1 \\ T_{x^*} \end{pmatrix} = \begin{pmatrix} 1 \\ T_{1,x_1^*} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ T_{V,x_V^*} \end{pmatrix}. \tag{17}$$

The combination of equations 16 and 17 gives:

$$\begin{pmatrix} 1 \\ T_{x^*} \end{pmatrix} = \sum_{j_1=0}^{J_1} \cdots \sum_{j_V=0}^{J_V} \lambda_{1,j_1} \cdots \lambda_{V,j_V} \begin{pmatrix} 1 \\ T_{1,j_1} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ T_{v,j_V} \end{pmatrix},$$

which leads to the desired expression for T_{x^*} after removing the first row.

Furthermore, one can observe that:

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

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.

Theorem 5

Proof Theorem 5 is the analog of Theorem 2 for a single variable. Looking closely to the proof of Theorem 2, we can see that it can be readily generalized without any change to the case of more than one variable, except for the maximum distance between two vertices of any simplex.

Indeed, calling T_{j_1} and T_{j_2} two of the vertices of one of the simplexes in the univariate case, the vector $\overrightarrow{T_{j_1}T_{j_2}}$ has been shown to be bounded by $\frac{M}{2^{K_l}}$ with $M = \sum_{j=1}^{J} \underset{x\in[0,1]}{sup} \|\mathbf{f}^{(j)}(x)\|$. To generalize to the multivariate case, due to the tensorial product, $\frac{M}{2^{K_l}}$ must be replaced by $\prod_{v=1}^{V} \frac{M_v}{2^{K_l}}$, where each M_v is taken to be

$$M_v = \sum_{j_v=1}^{J_v} \underset{x_v\in[0,1]}{sup} \|\mathbf{f_v}^{(j_v)}(x_v)\|.$$

Proposition 4

Proof When adding a point in the center of an initial domain (see figure 6)

- we replace the vertices on the external border: we add 3^V vertices and remove 2^V old ones.
- for the 2^V new simplexes, we add $2^V(J+1-2^V)$ interior points and remove $J+1-2^V$ points corresponding to the interior vertices of the old simplex.

This gives the expected result.

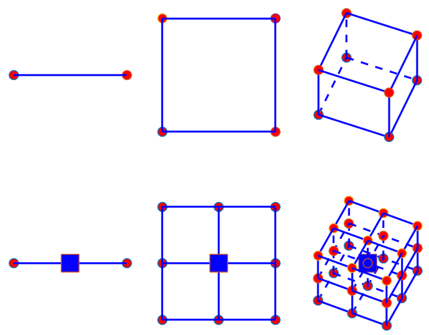

Fig. 6 number of constraints the upper row gives the limits of an initial domain ($[0, 1]^V$ for example) when $V = 1, 2, 3$. The lower row gives the new definition domain when a point drawn as a square is added in the previous lattice.

References

Ben-Tal, A. and A. Nemirovski (2001). *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Chaikin, S. (1974). An algorithm for high speed curve generation. *Computer Graphics and Image Processing 3*(4), 346–349.

De Boor, C. (2001). *A practical guide to splines*. Applied mathematical sciences. Berlin: Springer.

Du, P., C. Parmeter, and J. Racine (2013). Nonparametric kernel regression with multiple predictors and multiple shape constraints. *Statistica Sinica 23*, 1347–1371.

Fan, J. and I. Gijbels (1996). *Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66*. Chapman & Hall Ltd. Taylor & Francis.

Farin, G. (1993). *Curves and Surfaces for Computer Aided Geometric Design (3rd Ed.): A Practical Guide*. San Diego, CA, USA: Academic Press Professional, Inc.

Farin, G., H. Hoschek, and M. Kim (2002). *Handbook of Computer Aided Geometric Design*. Elsevier.

Gasca, M. and C. Micchelli (2013). *Total Positivity and Its Applications*. Mathematics and Its Applications. Springer Netherlands.

Hawkins, D. (1994). Fitting monotonic polynomials to data. *Computational Statistics 9*, 233–247.

Karlin, S. and W. Studden (1966). *Tchebycheff systems: with applications in analysis and statistics*. Pure and applied mathematics. Interscience Publishers.

Maatouk, H. and X. Bay (2017). Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences 49*(5), 557–582.

Meyer, M. (2012). Constrained penalized splines. *Canadian Journal of Statistics 40*(1), 190–206.

Murray, S., S. Müller, and B. Turlach (2016). Fast and flexible methods for monotone polynomial fitting. *Journal of Statistical Computation and Simulation 86*(15), 2946–2966.

Nocedal, J. and S. Wright (2006). *Numerical Optimization* (2nd ed.). New York: Springer.

Papp, D. and F. Alizadeh (2014). Shape-constrained estimation using nonnegative splines. *Journal of Computational and Graphical Statistics 23*(1), 211–231.

Peña, J. (1999). *Shape Preserving Representations in Computer-aided Geometric Design*. Nova Science Publishers.

Ramsay, J. and B. Silverman (2005). *Functional Data Analysis* (2nd ed.). New York: Springer-Verlag.

Schumaker, L. (2007). *Spline Functions: Basic Theory* (3 ed.). Cambridge Mathematical Library. Cambridge University Press.

Trevor, H., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning* (2nd ed.). New York: Springer-Verlag.

Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis.