



**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. 2016. hal-01262601v1

**HAL Id: hal-01262601**

**<https://hal.science/hal-01262601v1>**

Preprint submitted on 27 Jan 2016 (v1), 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

January 25, 2016

## Abstract

This paper first shows how to calculate a polynomial regression of any degree and of any number of variables under shape constraints. This framework is readily extended to linear combinations of basis functions as long as they respect a few key properties. It is proved that the procedure developed here is optimal in a certain sense. Theoretical explanations are first introduced for monotony constraints and then applied to simulated examples to show the behavior of the proposed algorithm. Two real industrial cases are then detailed and solved.

**Keywords.** multivariate parametric regression, monotony, constraints, shape constraints, quadratic programming

# 1 Introduction

Fitting a multivariate regression function to a set of observed points is a common industrial problem. In this context, polynomial regression is a very classic tool. Its ease of use makes it a natural candidate to engineers. It is effective even with a low number of experimental points. The resulting function depends on a limited number of coefficients and is computed very efficiently, since its terms involve only powers of the variables. It is solved in most cases by means of standard least squares procedure, available in every statistical software. It avoids the use of extra computational parameters, usually referred as hyperparameters, often hard to determine [21], particularly when using non parametric regression.

Adding some constraints is a current demand of practitioners, who would like to incorporate their knowledge and their expertise in the behavior of the resulting function. But, in contrast to the previous classical polynomial regression problem, these natural extensions are generally very hard to tackle, even in low dimensions, and for low degree polynomials. Indeed, most of the time it implies complicated non linear expressions of the coefficients.

In this paper, we first address the problem of multivariate polynomial regression under monotonicity constraints. But we extend very rapidly this initial framework to much more general situations as long as the function to fit can be expressed as a linear combination of basis functions, verifying a few properties detailed in the following sections. Among others we are then able to deal with sum of power law functions and sum of exponentials.

The key idea of this article is to transform the initial shape constraints into a limited number of linear constraints. In this way, the regression problem is reduced to a standard quadratic programming optimization procedure that can be solved by classical algorithms available in usual mathematical softwares. Our methodology ensures that the number of linear constraints added to the initial problem is minimal and that these constraints are optimal. To our knowledge, these are new results.

A marginal important benefit of this approach is that the expected behavior will be respected everywhere in the domain and not only in the vicinity of the observed points (see [13] 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.

This paper is organized as follows : after a short bibliography in section 2, the theory is exposed for monotony constraints, first in dimension 1 (section 3.1), before extending the idea to more dimensions (section 4). Simulations studies are then demonstrated with ad hoc examples (section 6.1 and 6.2). Finally, two industrial cases are detailed: one in petroleum engineering related to hydrotreatment of naphtha (section 6.4), and one in nuclear experiments (section 6.5).

## 2 Selected bibliography

Imposing shape restrictions is a very usual demand in regression analysis, and is still a very active domain of research. Shape restrictions include equality constraints and prior knowledge on particular points for which values are cer-

tain, like intercept, maximum or minimum values or inequality constraints like monotony requirements or positivity constraints on the function and its derivative, concavity or convexity (see Lauer [10]).

In univariate settings, one can say that each regression method has its own extension taking into account shape restrictions. The Pool Adjacent Violators Algorithm (PAVA) from Barlow [2] for solving monotonic regression problems results in step functions. Ramsay [17] introduced the use of regression splines for monotone regression functions. Constrained penalized splines are studied in [13].

Another type of method for regression subject to monotonicity constraints is kernel-type estimators (Hall and Huang 2001 [9], Dette et al. [6]). Local polynomial is the base of the work of Marron [12]. Generalized Additive Models (GAM) are theoretically explored in the work of Wood [21]. Antoniadis and coauthors [1] propose a constrained regression function using penalized wavelet regression techniques. Extensive review on these subjects can be found in Mammen [11].

Until recently, relatively to the univariate case, only a few works existed in multivariate settings, and we focus in this section to extensions of the isotonic regression, kernel regression, SVR and GAM.

The authors in [3] extend the PAVA procedure via graph theory to higher dimensions. Numerical experiments show that GPAV algorithms enjoy both low computational burden and high accuracy. It can be run with large data sets and several variables. But the solution is not guaranteed to be  $C^2$ , and may exhibit a staircase behavior, with large regions of constant shape followed by an abrupt step to the next level.

A variety of methods ([6], [16], [10], [4]) exist nowadays in kernel regressions, for which constraints are defined locally. A very common way of doing things is to define a grid and apply the needed constraints on every node. Obviously, the number of conditions grows exponentially with the dimension of the input space and this way of proceeding is only possible in low dimension problems. Besides, there is no guarantee that between each node, constraints are still valid. Moreover, each prediction on a new point requests to solve a new complex problem if one does not interpolate between the points of the grid.

Dette and Scheder [6] start by proposing a procedure of isotonization in one dimension to obtain a strictly monotone function. The algorithm is then applied successively to each dimension in case of multivariate regression.

Racine and Parmeter [16] suggest a weighted kernel regression where the weights have to be adjusted to satisfy the monotonicity constraints. This method has been extended recently to more than two dimensions [7].

Da Veiga and Marrel ([4]) use a kriging estimator and incorporate shape restrictions by means of conditional expectations of the truncated multinormal distribution.

In SVR, the coefficients are found by solving a quadratic programming optimization problem (see Lauer and Bloch [10]). In case of additional linear constraints (with respect to the coefficients), only the number of conditions is increased, the solving mechanism remains the same.

Another family of multivariate regression functions is based on the GAM theory. They may be applied to very large data sets as investigated by Wood and coauthors in [22]. GAM models are sum of one dimensional functions, and each of them is estimated by means of a non parametric regression. Like kernel regressions, they can adapt themselves to a wide variety of function behaviors,

for example to nearly flat regions followed by abrupt changes. When splines are chosen, these univariate regressions can be constrained to respect imposed shape requirements through Ramsay’s method [17], or De Boor [5] inequalities. Practical implementations in R and refinements are detailed in [20], [19] or in [15].

However, a GAM model may be difficult to estimate, the resulting function is not analytic and interactions of variables are not easy to handle. Moreover, in practical applications, fitting as accurately as possible a set of given points is not necessarily the main objective of the regression. Very often, obtaining a function approximating the experimental points and respecting adequately the expected behavior anticipated by the experts is sufficient. In this case, polynomials-like functions are very good natural candidate.

### 3 Univariate case

Our approach is first investigated on a simple example, in order to introduce the main concepts of our methodology: we fit a polynomial  $P(x) = \sum_{j=0,d} \alpha_j x^j$  to a set of given points under a constraint of monotonicity. This first study is rapidly extended to tackle the case of sum of power law functions  $P(x) = \sum_{j=0,m} \alpha_j x^{d_j}$ , where the exponents  $d_j$  are positive real numbers, in increasing order. Finally, we set out a general framework to handle the case of a sum of nested functions,  $P(x) = \sum_{j=0,m} \alpha_j f_j(x)$ , where the functions  $f_j$  share a few elementary properties.

#### 3.1 First approach in dimension 1

Let us begin with a very simple example, in dimension  $v = 1$  where we try to adjust a third-degree polynomial ( $d = 3$ ) expressed as  $P(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$  to a set of  $n$  given points  $(x_{(i)}, y_{(i)})_{i=1,n}$ , with the constraint that the resulting solution should be monotonically increasing within the domain of definition of  $x$ , taken to be  $[0, 1]$  without loss of generality.

The derivative  $P'(x) = \beta_1 + 2\beta_2 x + 3\beta_3 x^2$  is linear with respect to the coefficients  $\beta_1$ ,  $\beta_2$  and  $\beta_3$ . To empathize this, we rewrite  $P'(x)$  as  $P'(x) = Z(t_1, t_2) = \beta_1 + 2\beta_2 t_1 + 3\beta_3 t_2$  taking  $t_1 = x$  and  $t_2 = x^2$ . Since  $Z$  is linear with respect to the coefficients and since its domain of definition,  $[0, 1]^2$ , is convex, if  $Z(t_1, t_2)$  is positive in every four corners of the square  $[0, 1]^2$ , then  $Z(t_1, t_2)$  will be positive everywhere in  $[0, 1]^2$ , and so will  $P'(x)$  for all  $x \in [0, 1]$ .

In fact all these four inequalities are not necessary: all the possible values for  $[t_1, t_2]$  are included in the triangle defined by the vertices  $[0,0]$ ,  $[1,0]$ ,  $[1,1]$ , by convexity of the function  $t \rightarrow t^2$  for  $t \in [0, 1]$ . Consequently, to be sure of the sign of the derivative, it is only necessary to check the three linear following inequalities, instead of the four initial one in every corner:

$$\beta_1 \geq 0, \beta_1 + 2\beta_2 \geq 0, \beta_1 + 2\beta_2 + 3\beta_3 \geq 0 \tag{1}$$

corresponding to the equation of  $Z(t_1, t_2)$  in the three corners  $[0,0]$ ,  $[1,0]$  and  $[1,1]$ .

Mathematically, the least square problem to be solved can be expressed as  $\underset{\beta}{\operatorname{argmin}} \sum_{i=1,n} (y_{(i)} - P(x_{(i)}))^2$ , s.t. constraints (1), which is a classical convex

quadratic programming problem (see [14]).

This example is illustrated on the following figure, with the function

$$y = 1.5x + \frac{3}{4\pi} \sin(4\pi x),$$

which is approached by a polynomial regression of degree 3. Ten values for  $x$  are randomly taken in the interval  $[0,1]$ , and the corresponding  $y$  are calculated. A random normal noise of standard deviation  $\sigma = 0.1$  is added to each  $y$ . The green squares indicate the chosen points. In the case of the constrained regression, three linear conditions as indicated above are added to the initial optimization problem.

Three curves are drawn on the figure 1: in plain red, the calculated constrained regression; in dotted blue, the non constrained standard multivariate polynomial regression; in plain black, the true function. As can be seen on the graphic, the regression without any shape constraints is not monotone.

We use the root mean square error defined as  $RMSE = \frac{1}{n} \sqrt{(\sum_{i=1}^n (\hat{y}_{(i)} - y_{(i)})^2)}$  as an indicator of the quality of the regression, where  $n$  is the number of points, and  $\hat{y}_{(i)}$  the calculated  $i$ -th value. In this case, without constraints, the calculated RMSE is  $RMSE = 0.1060$ ; with constraints, the same indicator is only slightly worse,  $RMSE = 0.1081$ .

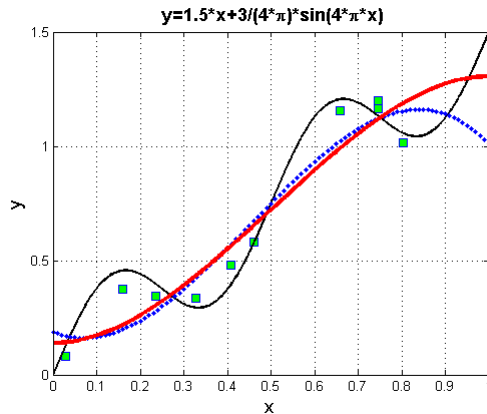


Figure 1: an example with a degree 3 polynomial

In a more general setting, still in one dimension, if the polynomial to fit is of degree  $d$ , we are going to prove in the next proposition (1) that the number of constraints needed is  $d$ . Let us quote that if the constraints were to be applied in every corner of the hypercube  $[0, 1]^d$ , we would have to take care of  $2^d$  constraints, instead of  $d$ . If this is still manageable in the univariate case, it would become rapidly intractable in multivariate situations.

The constraints will be applied to the derivative  $P'(x)$  which is a polynomial of degree  $d - 1$ .  $P'(x)$  can be in turn transformed in a linear function  $Z(t) = Z(t_1, t_2, \dots, t_{d-1})$  with  $t_1 = x, \dots, t_{d-1} = x^{d-1}$ , with values of  $t$  taken in the hypercube  $[0, 1]^d$  contained in a  $\mathbb{T}$  space.

For  $x$  is in  $[0, 1]$ , a point of coordinate  $t = (x, x^2, \dots, x^{d-1})$  is always inscribed in the convex polytope  $(P_d)$  delimited by the  $d + 1$  vertices  $T_j$  defined as  $0 \leq j \leq d$ ,  $T_j = (1_{i \in [0, j]})_{i=0, d}$ , where the first  $j + 1$  coordinates are equal to 1 and the others equal to 0. Note that  $T_0 = (1, 0, \dots, 0)$  and  $T_d = (1, \dots, 1)$ . This leads to write exactly  $d$  constraints corresponding to the  $d$  edges of  $(P_d)$  as in the proposition (1) below.

**Proposition 1.** *If all the following constraints  $(\chi_0, \chi_1, \dots, \chi_d)$  are verified:*

$$\forall 0 \leq j \leq d, \chi_j := \sum_{i=0}^j \alpha_i \geq 0$$

then,

$$\forall x \in [0, 1], P(x) := \sum_{i=0}^d \alpha_i x^i \geq 0.$$

The mathematical demonstration is omitted here, and delayed to the appendix 7.

### 3.2 Theorems in dimension 1 : the case of power law functions

In fact, this first polytope  $(P_d)$  can be reduced, using the tangents as edges. Numerically this is highly desirable: the larger the region where the constrained are to be respected the more difficult the problem is to solve.

In dimension  $d = 2$ , the natural idea is to look at the intersection of the tangent at the point  $T_2 = (1, 1)$  with the  $t_1$  axis (see figure (2), second panel from the left). We thus define  $T_1 = (1/2, 0)$ . In dimension 3, we keep  $T_0$  and  $T_1$ . And we look at the intersection of the tangent to the curve at  $T_3 = (1, 1, 1)$  with the plane  $(t_1, t_2)$ . It gives  $T_2 = (2/3, 1/3, 0)$  (see figure (2), fourth panel from the left). We continue this process until dimension  $d$  and we obtain the following  $T_j$ : we still have  $T_d = (1, \dots, 1)$  but for the other vertices  $T_j$  when  $0 \leq j < d$ , the first  $j$  coordinates are equal to  $\frac{j+1-i}{j+1}$  with  $0 \leq i \leq j$ .

The following figure 2 illustrates the previous statement when the degree  $d$  is 2 or 3. Each panel consists of one figure with the first proposed polytope in black, and the second one with the tightened up polytope in blue, to clearly emphasize the differences.

In the following theorem (1), we establish the set of linear constraints corresponding to this last situation. But beforehand, we introduce an improvement to the original idea which enlarges considerably the applicability of the upcoming theorem.

As a matter of fact, the exponents in  $P(x)$  need not to be integers. The next proposition works as well if the exponents are positive real numbers. In this case, of course,  $P(x)$  can not be stated any longer as a polynomial, but rather

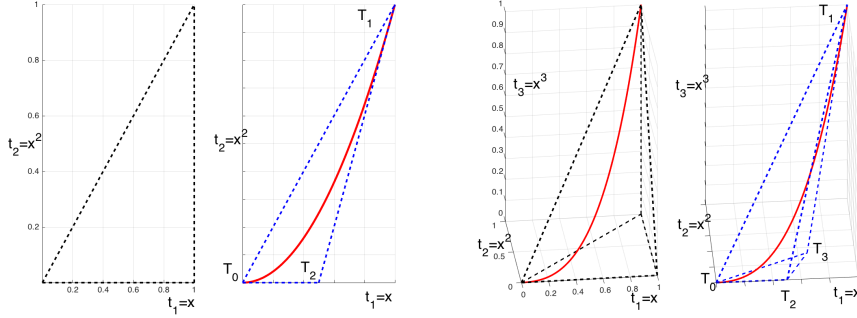


Figure 2: **single variable polynomial** Curve followed by  $t_1 = x$ ,  $t_2 = x^2$  on the left panel and by  $t_1 = x$ ,  $t_2 = x^2$ ,  $t_3 = x^3$  on the right, when  $x \in [0, 1]$

as a sum of  $m$  power law functions and  $P(x) = \sum_{j=0,m} \alpha_j x^{d_j}$ , where the  $d_j$  are positive real numbers, increasing with  $j$ .

**Theorem 1.** *if all the following constraints  $(\chi_0, \chi_1, \dots, \chi_m)$  are verified:*

$$\forall j, 0 \leq j \leq m-1, \chi_j := \sum_{i=0}^j \frac{d_{j+1} - d_i}{d_{j+1}} \alpha_i \geq 0$$

$$\chi_m := \sum_{i=0}^m \alpha_i \geq 0$$

then,

$$\forall x \in [0, 1], P(x) := \sum_{j=0}^m \alpha_j x^{d_j} \geq 0.$$

If  $P(x)$  is a classical polynomial,  $P(x) = \sum_{j=0,m} \alpha_j x^j$ , then  $\frac{d_{j+1} - d_i}{d_{j+1}}$  becomes  $\frac{j+1-i}{j+1}$ .

The idea of the proof is given as follows (the details can be found in Appendix 7). In the space  $(t_1, \dots, t_m)$ , we consider the points  $T_j$  corresponding to the vertices of the examined polytope  $(P_m)$ . If the constraints are verified in every  $T_j$ , then for any  $t$  inside  $(P_m)$ ,  $Z(t)$  will be positive. The proof consists in verifying that the curve given by  $(t_1 = x^{d_1}, \dots, t_m = x^{d_m})$  is included in the convex polytope  $(P_m)$  delimited by the vertices  $(T_j)_{j=0,m}$ .

The next theorem states that the number of points of the polytope of theorem (1) is minimal. Indeed,  $(P_m)$  is a simplex or a convex polytope in a dimension  $m$  space defined by  $m+1$  vertices which are not in the same hyperplane. Moreover it can be said that this last polytope is optimal as expressed in the next theorem (2).

**Theorem 2.** *The simplex  $(P_m)$  of theorem (1) is optimal in the sense that any convex polytope with  $m+1$  points, including the curve  $(t_1 = x^{d_1}, \dots, t_m = x^{d_m})$ , contains also  $(P_m)$ .*



For example, the polytope of proposition (1) contains the polytope of theorem (1). This last theorem (2) has two meanings. First the number of constraints is minimum. As we shall see this is particularly important with functions of two or more variables, especially with polynomials. Secondly, these constraints can not be improved.

### 3.3 Generalization in the univariate case : sum of exponentials

In theorem (2), we abundantly use the fact that the tangents to the curve  $(x^{d_j}, x^{d_k})$  for  $d_k$  greater than  $d_j$  are on the x-axis. We wish here to deal with more general functions than power law functions, as for example exponentials on a bounded interval, for which the derivatives at the origin are not zero. In this section, we first generalize our theorem (1) to a series of functions and then extend our approach to the case of tangents with a non-zero slope at  $x = 0$ .

Thus we consider a set of functions defined on  $[0, 1]$ . In the following,  $c_j = f'_j(0)$  and  $d_j = f'_j(1)$ , the value of the derivative of  $f_j$  at the points  $x = 0$  and  $x = 1$ . For power functions, the  $d_j$  correspond to the exponents. By convention,  $f_0(x) = 1$  on  $[0, 1]$ .

In the next propositions, we make use of some of these conditions:

1.  $\forall j, 1 \leq j \leq m, f_j(0) = 0, f_j(1) = 1,$
2.  $\forall j, 1 \leq j \leq m - 1, 0 \leq f_{j+1}(x) \leq f_j(x) \leq 1,$
3.  $\forall j, 1 \leq j \leq m, f'_j(0) = 0,$
4.  $\forall j, 1 \leq j \leq m, f'_j(1) = \infty.$

Then still going a step further, the preceding power law functions can be replaced by such a series of functions which satisfies some of the previous elementary properties as long as they respect the constraints of theorem (1). At this point,  $P(x)$  is expressed as a sum of  $m$  nested functions on  $[0, 1]$  and  $P(x) = \sum_{j=0, m} \alpha_j f_j(x).$

**Theorem 3.** *Under conditions 1, 2 and 3, and the complementary fourth condition*

$$\forall j, 0 \leq j \leq m - 2, \frac{f_j(x) - f_{j+1}(x)}{d_{j+1} - d_j} - \frac{f_{j+1}(x) - f_{j+2}(x)}{d_{j+2} - d_{j+1}} \geq 0,$$

*if all the following constraints  $(\chi_0, \chi_1, \dots, \chi_m)$  are verified:*

$$\forall j, 0 \leq j \leq m - 1, \chi_j := \sum_{i=0}^j \frac{d_{j+1} - d_i}{d_{j+1}} \alpha_i \geq 0$$

$$\chi_m := \sum_{i=0}^m \alpha_i \geq 0$$

*then,*

$$\forall x \in [0, 1], P(x) := \sum_{j=0}^m \alpha_j f_j(x) \geq 0.$$

The proof is similar to the previous one in theorem (1). One can check easily that the fourth condition is fulfilled for power law functions.

Many functions, not necessarily very smooth, verify this fourth item. Let us take  $g_j(x) = \frac{x^{d_j} - x^{d_{j+1}}}{d_{j+1} - d_j}$  for  $0 \leq j \leq m-1$ . Then a sufficient condition for the fourth item to be accepted is that  $g_{j-1}(x) \geq \frac{f_j(x) - f_{j+1}(x)}{d_{j+1} - d_j} \geq g_j(x)$  for all  $x$  in  $[0,1]$ .

The complete generalization of this series of propositions would be a theorem which encompasses all the cases we have discussed up to now: the derivatives would take any value between 0 and 1 at the origin, and any value greater than one at the point  $(f_1(1), \dots, f_m(1))$ . Unfortunately, our recursive construction of the polytope enclosing the curve  $(f_1(x), \dots, f_m(x))$  would not work anymore: adding a dimension to the curve when passing from  $(f_1(x), \dots, f_{m-1}(x))$  to  $(f_1(x), \dots, f_m(x))$  changes all the vertices.

However with theorem (3) we have a tool to relax the third condition concerning the tangents at the origin. We take now  $h_j(x) = \frac{f_j(x) - c_j x}{1 - c_j}$  and  $P(x)$  becomes  $P(x) = \sum_{j=0, m} \alpha_j (1 - c_j) h_j(x) + x \sum_{j=0, m} \alpha_j c_j$ . It is clear that  $h_j(x) \geq 0$ ,  $h_j(0) = 0$  and  $h_j(1) = 1$ .

Thus we apply the theorem (3) to  $\sum_{j=0, m} \alpha_j (1 - c_j) h_j(x)$  and we add the condition  $\sum_{j=0, m} \alpha_j c_j \geq 0$  to ensure the positivity of  $P(x)$  everywhere on  $[0,1]$ .

$$\text{Here } h'_j(1) = \frac{d_j - c_j}{1 - c_j}.$$

**Corollary 1.** *Under conditions 1 and 2 and if*

- $\forall j, 1 \leq j \leq m-1, h_j(x) = \frac{f_j(x) - c_j x}{1 - c_j} \geq h_{j+1}(x) = \frac{f_{j+1}(x) - c_{j+1} x}{1 - c_{j+1}}$
- $\forall j, 0 \leq j \leq m-2,$

$$\frac{h_j(x) - h_{j+1}(x)}{h'_{j+1}(1) - h'_j(1)} - \frac{h_{j+1}(x) - h_{j+2}(x)}{h'_{j+2}(1) - h'_{j+1}(1)} \geq 0,$$

*if all the following constraints  $(\chi_0, \chi_1, \dots, \chi_m)$  are verified:*

$$\forall j, 0 \leq j \leq m-1, \chi_j := \alpha_0 + \sum_{i=1}^j \left( (1 - c_i) - (1 - c_{j+1}) \frac{d_i - c_i}{d_{j+1} - c_{j+1}} \right) \alpha_i \geq 0$$

$$\chi_m := \sum_{j=0}^m \alpha_j \geq \sum_{j=0}^m \alpha_j c_j \geq 0$$

*then,*

$$\forall x \in [0, 1], P(x) := \sum_{j=0}^m \alpha_j f_j(x) \geq 0.$$

Finally, for symmetry reasons, before ending this section, we now take into account the situation of infinite tangents at  $(1, \dots, 1)$ . In this situation, the slope of the tangents at the origin can take any value between 0 and 1.

**Proposition 2.** Under conditions 1, 2 and 4,

$$\text{if } \forall j, 1 \leq j \leq m-2, \frac{f_j(x) - f_{j+1}(x)}{c_j - c_{j+1}} - \frac{f_{j+1}(x) - f_{j+2}(x)}{c_{j+1} - c_{j+2}} \geq 0,$$

if all the following constraints  $(\chi_0, \chi_1, \dots, \chi_m)$  are verified:

$$\begin{aligned} \chi_0 &:= \alpha_0 \geq 0 \\ \forall j, 1 \leq j \leq m-1, \chi_j &:= \sum_{i=0}^j \alpha_i + \sum_{i=j+1}^m \frac{c_i}{c_j} \alpha_i \geq 0 \\ \chi_m &:= \sum_{i=0}^m \alpha_i \geq 0 \end{aligned}$$

then,

$$\forall x \in [0, 1], P(x) := \sum_{j=0}^m \alpha_j f_j(x) \geq 0.$$

As in theorem (2), and proved in the appendix, the previous simplices are also optimal.

**Corollary 2.** The simplices of theorem (3), corollary (2) and proposition (2) are also optimal.

Here are some examples of functions satisfying the conditions of corollary (2) or proposition (2), depending on one (or more parameters) parameter  $k_j$  (sometimes  $k_{1_j}, k_{2_j}$ ) strictly positive and increasing with  $j$ .

- $\frac{\exp(k_j x) - 1}{\exp(k_j) - 1}$
- $\frac{(x+1)^{k_j} - 1}{2^{k_j} - 1}$
- $\frac{\cosh(k_j x) - 1}{\cosh(k_j) - 1}$
- $\frac{1}{2}(x^{k_{1_j}} + x^{k_{2_j}})$ , for adequate  $k_{1_j}$  and  $k_{2_j}$
- $\frac{\sin(\pi/2 k_j * (x-1))}{\sin(\pi/2 k_j)} + 1$ , when  $k_j \geq 1$  for all  $j$
- $1 - (1-x)^{1/k_j}$  for  $k_j > 1$ .

## 4 Multivariate case

We switch to a more general situation, where  $x = (x_1 \cdots x_v)$  is v-dimensional, with a monotony constraint required for the first coordinate  $x_1$ :  $P(x)$  should monotonically increase or decrease with  $x_1$ , when all the variables remain unchanged. To introduce the next proposition, we take the case of a polynomial

$P(x)$  composed of  $m + 1$  monomials. Each of these monomials can be written  $x_1^{i_1} \cdots x_v^{i_v}$ , where  $i_1 \cdots i_v$  are integer degrees.

As in the preceding section, we rewrite  $P'_1(x)$ , the derivative of  $P$  with respect to  $x_1$ , as:

$$P'_1(x) = Z(t) = Z(t_1, t_2, \dots, t_m) = \sum_{k=1 \cdots m} \beta_k t_k,$$

where each  $t_k$  for  $k = 1, m$  is one of the  $m$  monomials in the expression of  $P'_1$ , with corresponding coefficient  $\beta_k$ .  $t$  and  $\beta$  are the vectors containing all the  $t_k$  and the  $\beta_k$  respectively.

To check monotonicity in every point of the hypercube covered by  $x_1, x_2, \dots, x_v$ , we examine  $P'_1(x)$ , and we have to verify that  $P'_1(x) \geq 0$  (or  $\leq 0$ ) in the entire domain.

Indeed, as in dimension 1, our first claim is that one way to be sure  $P$  is monotone with respect to  $x_1$  is to impose that  $Z$  should be positive (or negative) in every corner of the corresponding hypercube. Denoting  $C_i$  one of these corners, it amounts to check that  $Z(C_i) \geq 0$  for all  $i$ . Bringing together all the constraints in a matrix form we get  $C\beta \geq 0$ , and the problem to solve in dimension  $v$  can be rephrased as:

$$\underset{\beta}{\operatorname{argmin}} \sum_{i=1, n} (y_{(i)} - P(x_{(i)}))^2, \text{ s.t. constraints } C\beta \geq 0. \quad (2)$$

This is a classical quadratic optimization problem with linear inequality constraints. It is nowadays easily solvable by usual available mathematical software, save for the number of constraints. If the principle is simple, the realization is much more tedious since the number of constraints,  $2^m$ , will increase exponentially with the number of necessary monomials, and thus with the number of variables  $v$  and the degrees of the corresponding monomials.

Similarly to dimension 1, our second claim is that we can reduce drastically the number of constraints. We first explain on a simple example how to extend the previous results from dimension 1. Then we introduce a general proposition which gives a means to automatically generate the needed constraints.

We consider first a polynomial of two variables  $P(x_1, x_2)$ . resulting from the tensor product of two polynomials  $P_1(x_1)$  and  $P_2(x_2)$ : if the variable  $x_1$  appears in some monomial with the degree  $d_{i_1}$  in  $P_1$  and the variable  $x_2$  in some other monomial in  $P_2$  with degree  $d_{i_2}$  then  $P(x_1, x_2)$  incorporates a monomial  $x_1^{d_{i_1}} x_2^{d_{i_2}}$ . For example, if  $(1, x_1)$  and  $(1, x_2, x_2^3)$  are the powers of  $x_1$  and  $x_2$  in  $P(x_1, x_2)$  then all the following 6 monomials are present in  $P(x_1, x_2)$ :  $(1, x_1, x_2, x_1 x_2, x_2^3, x_1 x_2^3)$ .

In this example, the curve  $(1, x_1)$  is contained in the segment  $T_{1,0} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ,  $T_{1,1} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ . While for the curve  $(1, x_2, x_2^3)$ , we consider the triangle

$$T_{2,0} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, T_{2,1} = \begin{pmatrix} 1 \\ \frac{2}{3} \\ 0 \end{pmatrix}, T_{2,2} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

When  $x_1$  and  $x_2$  browse  $[0, 1]^2$ , the surface generated by all the possible values of  $(1, x_1, x_2, x_1 x_2, x_2^3, x_1 x_2^3)$  will be inside the convex delimited by the 6 vertices

$$\begin{array}{cccccc}
T_0 & T_1 & T_2 & T_3 & T_4 & T_5 \\
\left( \begin{array}{cccccc}
1 & 1 & 1 & 1 & 1 & 1 \\
0 & 2/3 & 1 & 0 & 2/3 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 2/3 & 1 \\
0 & 0 & 0 & 0 & 0 & 1
\end{array} \right)
\end{array}$$

These points result from the intersections of the tangent hyperplanes at  $T_0$  and  $T_5$  with the other axis. They can also be arranged in a matrix  $T$  defined as the tensor product of  $(T_{1,0}, T_{1,1}) \otimes (T_{2,0}, T_{2,1}, T_{2,2})$ . This last statement is the basis of the proof of the next proposition, which is similar to theorem (1), but with matrix notations.

Here, all the coefficients of the polynomial are gathered in a vector  ${}^t\alpha = (\alpha_{0\dots 0}, \dots, \alpha_{i_1\dots i_v}, \dots, \alpha_{d_1\dots d_v})$ . The indices  $i_1$  (respectively  $i_2, \dots, i_v$ ) take their value in a subset of integers between 0 and  $d_1$ .  $d_1$  is the maximum degree for  $x_1$  and is necessarily included in the indices  $i_1$ . The notation  ${}^t\alpha T \geq 0$  simply implies that all the coordinates of the vector  ${}^t\alpha T$  are positive. Consequently, if  ${}^t\alpha T \geq 0$ , then  $P(x) \geq 0$  for all  $x \in [0, 1]$ .

As in the univariate case, a few additional remarks are worth mentioning before giving a generalized version of the corresponding proposition.

1.  $P(x)$  can take a much more general form than a polynomial. For two variables, we need to define two series of functions,  $f_{1,1}(x_1), \dots, f_{1,m_1}(x_1)$  and  $f_{2,1}(x_2), \dots, f_{2,m_2}(x_2)$ . Each series verify individually the conditions of theorem (3).

We then express  $P(x)$  as  $P(x) = \sum_{j=1,m} \alpha_j g_j(x)$ , where  $m$  is the total number of terms in  $P(x)$ . Each  $g_j(x)$  is the product of a function from the first series by a function from the second one. Obviously, this process can be generalized to as many variables as needed.

2. Actually, a function  $P(x)$  containing all the terms resulting from the tensor product  $(1, f_{1,1}, \dots, f_{1,m_1}) \otimes \dots \otimes (1, f_{2,1}, \dots, f_{2,m_2})$  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 polynomial will not include necessarily all interactions terms: it is very common to ignore interactions of more than two variables in this situation.

In the  $\mathbb{T}$  space, each  $g_j$  is interpreted as a coordinate  $t_k$  and we consider the  $d$ -dimensional variety defined by  $(t_j = g_j(x))_{i=1,m}$ . Dropping some term amounts to projecting the polytope resulting from the tensor product on a subspace where the corresponding coordinates are omitted. As it happens, this projection does not create any new vertex. As a result, in the matrix of constraints, the corresponding rows are merely deleted.

**Theorem 4.** *If  ${}^t\alpha T \geq 0$  then,*

$$\forall x_1, \dots, x_v \in [0, 1]^v, P(x_1, \dots, x_v) \geq 0.$$

**Corollary 3.** *The number of constraints in theorem (4) is at most  $m + 1 = \prod_{j=1,v} (m_j + 1)$ .*

In the previous example where  $P$  is a linear combination of  $(1, x_1, x_2, x_1x_2, x_2^3, x_1x_2^3)$  the number of variables in  $P'_1(x)$  is 2,  $m_1 = 1$  and  $m_2 = 2$ . Therefore, the number of constraints is at most 6. The set of constraints in this case has been already given. The proof of corollary (3) is to be found in the appendix.

To give an idea of how much it reduces the number of constraints, we anticipate a little bit one of our industrial example: in section 6.5, a degree 3 polynomial with 6 variables is needed. This polynomial includes 84 monomials. If all of them are kept, a single monotony requirement will give rise to  $3^6 = 729$  constraints with our theorem (4) and this is easily manageable in most numerical softwares. As explained in section (6.5), since 6 monotony constraints are required, we need  $6 * 3^6 = 4374$  linear inequalities.

As in the preceding univariate case, this convex is optimal : first the number of constraints is minimal. Secondly, these constraints can not be tightened up. We consider a convex with the same number of vertices, which includes the algebraic variety  $(V)$  whose terms are given by the terms of  $P(x)$ .

**Theorem 5.** *The convex  $(P_m)$  of theorem (4) is optimal in the sense that any convex polytope with  $m + 1$  points, including the previous variety, contains also  $(P_m)$ .*

## 5 Practical considerations

### 5.1 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. Monotony requirements 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. Two real examples are worked out in sections 6.4 and 6.5.
2. Of course, every monotone transformation of the variables  $x_1, \dots, x_v$  will not change the procedure. If  $P$  is monotone with respect to  $x_1$ , then through the transformation  $x_1 = h(u)$ ,  $P$  will also be monotone with respect to  $u$  as long as the derivative  $\frac{dh(u)}{du}$  is strictly increasing or decreasing and is bounded on its domain of definition.
3. Up to now, we have only considered increasing polynomials with respect to one variable. It is easy to deal with decreasing requirements by changing the corresponding variable  $x_1$  (for example) in  $1 - x_1$  and continue with the previous procedure.
4. Until now, we forced the variables to be included in the interval  $[0,1]$ . This limitation can also be easily relaxed to any bounded interval in  $\Re$ .

5. As in [10] or [4], 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 constraints;
- concavity or convexity constraints as they result on an upper or lower zero bound on the second derivative, which remains linear with respect to the coefficients;
- bound constraints on the function itself, or on its derivatives;
- equality constraints;
- any kind of linear constraint on the coefficients.

6. 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.

## 5.2 implementation

Two steps are needed in practical implementations. In the first one we set up the matrix of constraints and in the second we solve the corresponding optimization problem. For this second step any standard algorithm in QP problems will fit our need, for example active-set algorithms.

The proposed method has been implemented in a general form for monotony constraints applied on polynomials and power law functions with Matlab®, and tried on ad hoc examples for other general cases (exponential functions or bound constraints for example).

In this implementation, for monotony constraints on polynomials or power law functions, we need to take the derivative of the initial function and to perform some symbolic manipulations as follows: the considered polynomial is coded by means of one matrix and a vector of coefficients. Each line of the matrix represents a monomial and each column a variable. The values stored in this matrix are the powers of each variable in the monomials. By convention, the first column corresponds to the constant term, which is considered as a variable elevated to the power 0. For example, with three variables, the row (0, 1, 2, 3) will represent  $x_1 x_2^2 x_3^3$ .

In this way, to establish the derivative of a polynomial formula with respect to one variable means only subtracting one to the elements of the corresponding column and multiplying the coefficients by the previous values of this column.

With this structure, the various propositions presented previously can be coded without major difficulties for polynomial or power law functions.

## 6 Examples

### 6.1 Simulated example in dimension 1

In this example, 10 points are generated from the equation  $y = -6x^3 + 10x^2 - 3x$ , on the interval  $[0, 1]$ . A random gaussian noise of standard deviation 0.1 is added to  $y$ . Our request here is to fit a monotone increasing function to these points.

On the left panel of the following figure, we choose a third degree polynomial as the fitting function and the constraints come from theorem (1). On the right panel, we take  $Q(x) = \alpha_0 + \int_0^x \sum_{j=1,4} \alpha_j f_j(x) dx$  as the fitting function; the derivative  $P(x) = \sum_{j=1,4} \alpha_j f_j(x)$  should be positive everywhere on  $[0,1]$  according

to corollary (2). In this latter case, the  $f_j$  take the form  $f_j(x) = \frac{\exp(k_j x) - 1}{\exp(k_j) - 1}$ , and the  $k_j$  are randomly chosen between 0 and 5 (on the figure,  $k_1 \simeq 0.21, k_2 \simeq 0.95, k_3 \simeq 1.41, k_4 \simeq 3.18$ ).

The resulting function is drawn on the following figure (3). The dashed blue line is the original regression function. The plain red line is the obtained regression function. One can see only minor differences between the two panels, maybe more apparent at the beginning and at the end of the curves.

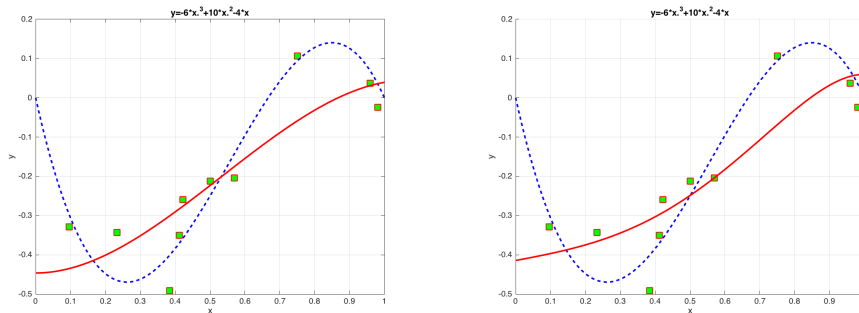


Figure 3: 10 points from  $y = -6x^3 + 10x^2 - 3x$  plus some noise. On the left, the fit is based on a third degree polynomial, on the right on a sum of 4 exponential functions.

### 6.2 a bounded function in dimension 1

We consider here the function  $y = 1 - 4(x - 1/2)^2$ , drawn on figure (4), in black, on the left panel. At  $x = 1/2$ , this function reaches its maximum,  $y = 1$ . Twenty 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 square on the figure 4. They are fitted with a 2 degree polynomial with the additional constraint that the maximum should be 1. The resulting polynomial is drawn in red.

We can see that the obtained fit respects the constraint, but is obviously not what is expected: constraints seem too stringent. The problem comes from



the fact that the original function does not respect our sufficient conditions of theorem (1).

A much better fit is obtained if we try a sum of four power law functions,  $[x^{0.5}x^{0.75}x^4x^5]$  (see right panel of figure (4)). The reason to this is that near  $x = 0$ , the tangent to the original  $y$  is very steep, as it is at  $x = 1$ .

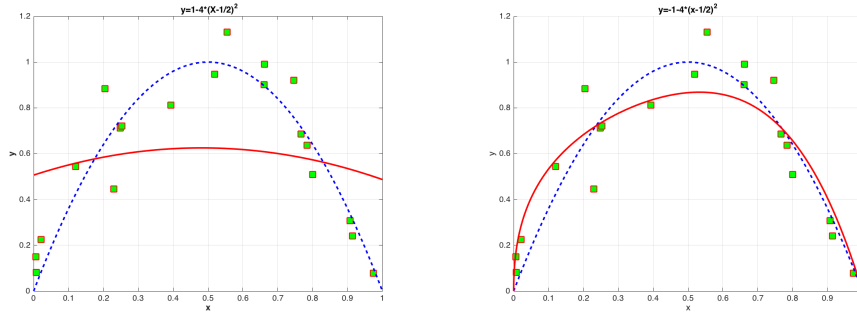


Figure 4: fit of the function  $y = 1 - 4(x - 1/2)^2$ . The original function is in blue, the obtained constrained function with a maximum of 1 in red. On the left panel, the regression function is a polynomial of degree 2. On the right, it is a sum of power law functions.

### 6.3 Simulated example in dimension 2

In figure (5) 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  $y$ .

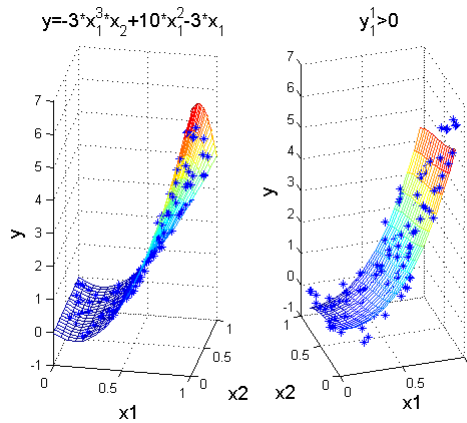


Figure 5: regression in dim 2 with  $d = 3$ , monotony constraints on  $x_1$

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  $x_1$ .

## 6.4 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 commercialised. 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 :

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

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;  $V\dot{V}H$  the Velocity per Volume and per Hour, proportional to the inverse of the time;  $k$ ,  $E$ ,  $m$  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  $y = \sum_{i=1,4} \beta_i x^i$ , where  $y = \ln(-\ln(\frac{C}{C_0}))$ ,  $x^1 = 1/T$ ,  $x^2 = \ln(V\dot{V}H)$ ,  $x^3 = \ln(P_{H_2})$ ,  $x^4 = \ln(P_{H_2S})$ .

Unfortunately, this expression is unable to take into account the full complexity of the process, and 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
- $V\dot{V}H$  decreases or  $x^2$  increases
- $P_{H_2}$  or  $x^3$  increases.

The following figure (6) 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 : Root Mean Square Error is 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  $V\dot{V}H$  or  $x^2$ , blue lines to variations with  $P_{H_2}$  or  $x^3$ . Behaviors for the regressions without constraints are obviously wrong: the black dotted line is increasing instead of decreasing and the blue has a minimum.

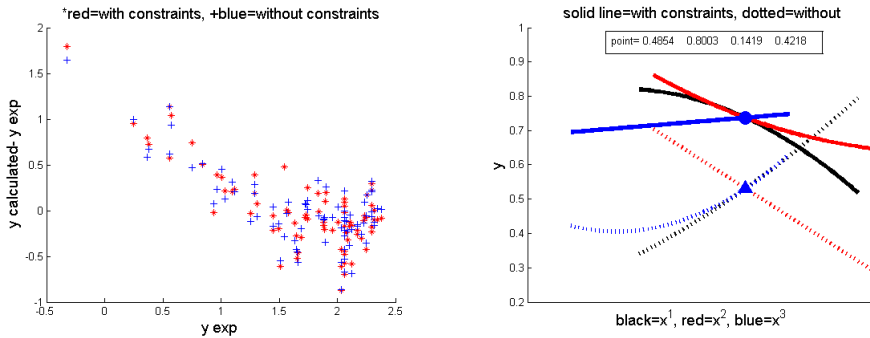


Figure 6: **polynomial fit to the data of HDS experiments** Residue diagram for the HDS data on the left panel and spider plot for a comparison of the UNconstrained and constrained multivariate regression on the right

## 6.5 Real example: radiative shock experiments

This work is described in the reference [8]. The response  $y$  is named 'Radiative Accretion shocks' and characterizes laser experiments pertinent in astrophysics. Six dimensionless variables characterizing the radiative flow should influence the answer. For clarity in this paper, these variables are named  $x_1$  to  $x_6$  but the interested reader should refer to the original paper for details. 2000 experimental observations are available for fitting the model, from which 200 were extracted by latin hypercube sampling.

Physical reasons indicate that the response is monotonically increasing with the first 3 variables and decreasing with the other 3. The response is fit with a polynomial of degree 3 in six variables, which includes 84 terms. The number of constraints is 4374, exactly the maximum possible number stated in corollary (3) of theorem (4): each function  $f_j$  is a degree 2 polynomial with  $m_j = 2$  monomials. Thus the number of constraints is at most  $\prod_{j=1,v} (m_j + 1) = 3^6 = 729$ , for each monotony requirement, giving finally  $4374=6*729$  constraints.

The following figure (7) shows the results. The lines corresponds to the conditional mean of the response with respect to the indicated variable. The plain lines on the left panel correspond to the proposed methodology and the dotted lines on the right to a multivariate linear regression on the same data. Obviously while the general behaviours of the curves are very similar, we can see that the magenta curve for  $x_4$  on the right panel is not monotone.

In this case, the  $RMSE$  changes from  $RMSE = 0.006$  for the unconstrained case to  $RMSE = 0.014$ , roughly a multiplication by 2.

## 7 Perspectives and Conclusions

The proposed procedure is very general and flexible and 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 the 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 constraints.

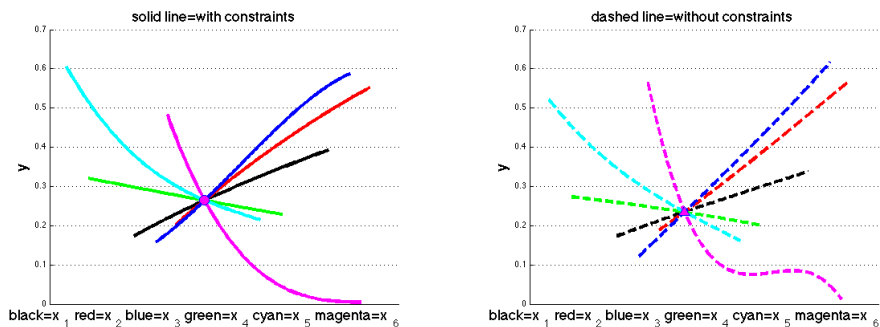


Figure 7: **polynomial fit to the data of radiative shock experiments** Spider plot for the constrained regression on the left panel and spider plot for the UNconstrained multivariate regression on the right

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.

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 in [18].

A second enhancement would be to optimize if necessary the parameters of the basis functions. In this paper, the parameters are fixed. But of course, 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.

The original algorithms for polynomials are developed in Matlab<sup>®</sup> and available upon request.

## Acknowledgements

The author would like to thank Jean Giorla from CEA-DAM-DIF for giving access to the data of the radiative shock experiments and Damien Hudebine from IFPEN for the example in hydrotreatment, and Pr. Fabrice Gamboa for his careful advices.

## Appendix: proofs

### Proposition (1).

*Proof.* In the  $\mathbb{T}$  space  $(t_1, \dots, t_d)$ , we consider the points  $T_j$  when  $0 \leq j \leq d$ , where only the first  $j$  coordinates are different from zero and equal to 1. These points define a convex polytope  $(P_d)$ .

We note immediately that the constraints can be rewritten:  $\forall 0 \leq j \leq d, Z(T_j) \geq 0$ . If the constraints are verified in every vertices  $T_j$  of  $(P_d)$ , then for any point  $T$  inside  $(P_d)$ ,  $Z(T)$  will be positive, because  $Z$  is linear. The proof

consists in verifying that the curve  $(C_d)$  given by  $(t_1 = x, \dots, t_d = x^d)$  is enclosed in  $(P_d)$ .

Since  $(P_d)$  is convex, this means that any point  $C(x) = (x, \dots, x^d)$  of the curve  $(C)$  can be expressed as a linear combination of the  $(T_j)_{j=0,d}$ :  $C(x) = \sum_{j=0}^d \lambda_j(x) T_j$ , where  $\sum_{j=0}^d \lambda_j(x) = 1$  and all the  $\lambda_j(x)$  are positive or null.

The  $\lambda_j(x)$  are very easy to find in this case and are defined as

$$\begin{aligned} \forall j, 0 \leq j < d, \lambda_j(x) &= x^j - x^{j+1} \\ \lambda_d(x) &= x^d \end{aligned}$$

□

### Theorem 1.

*Proof.* The idea of the proof is the same as in the previous proposition (1). We consider the simplex  $(P_m)$  given by the vertices  $T_j$ :

$$\begin{aligned} \forall j, 0 \leq j \leq m, T_j(i) &:= \frac{d_{j+1} - d_i}{d_{j+1}} \mathbb{1}_{0 \leq i \leq j} \\ \forall i, 0 \leq i \leq m, T_m(i) &:= 1. \end{aligned}$$

We are going to prove that the curve  $(x^{d_1}, \dots, x^{d_m})$  is enclosed in the convex set  $(P_m)$ .

Taking  $d_0 = 0$ , this time, the  $\lambda_j(x)$  are defined as:

$$\begin{aligned} \forall j, 0 \leq j \leq m-2, \lambda_j(x) &= \frac{d_{j+1}}{d_{j+1} - d_j} (x^{d_j} - \frac{d_{j+2} - d_j}{d_{j+2} - d_{j+1}} x^{d_{j+1}} + \frac{d_{j+1} - d_j}{d_{j+2} - d_{j+1}} x^{d_{j+2}}) \\ \lambda_{m-1}(x) &= \frac{d_m}{d_m - d_{m-1}} (x^{d_{m-1}} - x^{d_m}) \\ \lambda_m(x) &= x^m. \end{aligned}$$

They are found as the solution of the linear system:

$$T \begin{pmatrix} \lambda_0(x) \\ \dots \\ \lambda_m(x) \end{pmatrix} = \begin{pmatrix} x^{d_0} \\ \dots \\ x^{d_m} \end{pmatrix},$$

where  $T$  is the matrix formed by the coordinates of all the vertices  $T_j$ .

The equality  $\sum_{j=0}^d \lambda_j(x) = 1$  comes from the first line of the previous system, since the first coordinate of all the vertices  $T_j$  is always 1.

We check immediately that  $\lambda_m(x)$  and  $\lambda_{m-1}(x)$  are positive for  $x \in [0, 1]$ . For the general  $\lambda_j(x)$ , we first factor  $\lambda_j(x)$  as  $\lambda_j(x) = \frac{d_{j+1}}{d_{j+1} - d_j} x^{d_j} \mu_j(x)$  and we have to check that

$$\mu_j(x) = 1 - \frac{d_{j+2} - d_j}{d_{j+2} - d_{j+1}} x^{d_{j+1} - d_j} + \frac{d_{j+1} - d_j}{d_{j+2} - d_{j+1}} x^{d_{j+2} - d_j}$$

is positive. The derivative of  $\mu(x)$  is readily seen to be negative when  $x \in [0, 1]$ . Since  $\mu_j(0) = 1$  and  $\mu_j(1) = 0$ ,  $\mu_j(x)$  decreases from 1 to 0, and is positive on the definition interval.

□

**Theorem 2.**

We proceed in two steps, and in the first one we state two introductory lemmas. The first lemma links the optimal  $(P_{m-1})$  to  $(P_m)$ . Let  $T_{m-1,j}$  (resp.  $T_{m,j}$ ) when  $j \leq m-1$  (resp.  $m$ ) be the vertices of  $(P_{m-1})$  (resp.  $(P_m)$ ). The subscript  $m$  or  $m-1$  is added to distinguish the polytope from where they are taken. However, for  $j \leq m-2$ ,  $T_{m,j}$  and  $T_{m-1,j}$  are confounded. The affine hyperplane defined by the equation  $t_m = 0$  is denoted  $H_{t_m=0}$ .

**Lemma 1.** *The three following items are true:*

1.  $T_{m-1,m}$  is on the line joining  $T_{m-2,m-1}$  to  $T_{m-1,m-1}$ .
2. The face of  $(P_m)$  not containing  $T_{0,m}$  is perpendicular to the hyperplane  $H_{t_m=0}$ .
3. The projection of  $(P_m)$  on the hyperplane  $H_{t_m=0}$  is exactly  $(P_{m-1})$ .

*Proof.* This proof can be followed on figure 8 ( $m = 3$ ). Everything stems from the fact that the face of  $(P_m)$  not containing  $T_{m,m}$  (delimited by  $T_{0,m}$ ,  $T_{m-2,m}$  and  $T_{m-1,m}$  on the figure) is perpendicular to the face not containing  $T_{0,m}$  ( $T_{m,m}$ ,  $T_{m-2,m}$  and  $T_{m-1,m}$  on the figure).

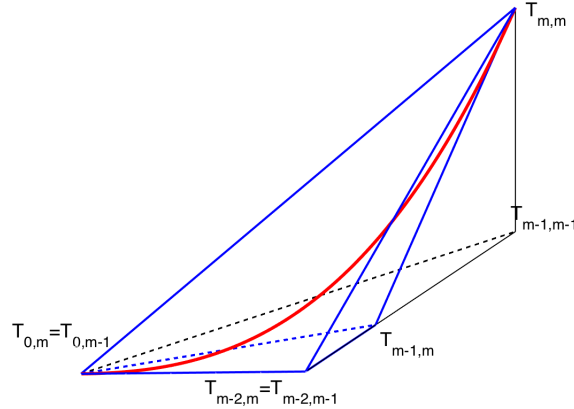


Figure 8:  $(C_m)$  in red,  $(P_{m-1})=(T_{0,m-1}, T_{m-2,m-1}, T_{m-1,m-1})$  in black and  $(P_m)=(T_{0,m}, T_{m-2,m}, T_{m-1,m}, T_{m,m})$  in blue

1. In the space  $(t_1, \dots, t_m)$  we have
 
$$T_{m-2,m-1} = (1, \frac{d_{m-1}-d_1}{d_{m-1}}, \dots, \frac{d_{m-1}-d_{m-2}}{d_{m-1}}, 0, 0)$$

$$T_{m-1,m} = (1, \frac{d_m-d_1}{d_m}, \dots, \frac{d_m-d_{m-1}}{d_m}, 0)$$

$$T_{m-1,m-1} = (1, \dots, 1, 0).$$

The vectors  $T_{m-2,m-1}T_{m-1,m-1}$  and  $T_{m-1,m}T_{m-1,m-1}$  are then collinear.

2. The vector  $T_{m-1,m-1}T_{m,m}$  is orthogonal to  $H_{t_m=0}$  and orthogonal to the vector  $T_{m-2,m-1}T_{m-1,m-1}$ . Due to the first item,  $T_{m-1,m-1}T_{m,m}$  is then orthogonal to  $T_{m-2,m}T_{m-1,m}$  which is one edge the face of  $(P_m)$  not containing  $T_{0,m}$ .

3.  $(P_m)$  and  $(P_{m-1})$  share  $m - 1$  vertices. The two vertices that are in  $(P_m)$  and not in  $(P_{m-1})$  are  $T_{m,m}$  and  $T_{m-1,m}$ . The point  $T_{m-1,m-1}$  of  $(P_{m-1})$  is the projection of  $T_{m,m}$ . And as proved in the first item,  $T_{m-1,m}$  is on the edge joining  $T_{m-2,m-1}$  to  $T_{m-1,m-1}$ .

□

In the second introductory lemma, we consider a simplex  $(P'_m)$  enclosing the curve  $(C_m)$  in  $[0, 1]^m$ .

**Lemma 2.** *The face of  $(P'_m)$  opposite to the vertex  $T_m$  belongs to the hyperplane  $H_{t_m=0}$ .*

*Proof.* We consider a vector  $v = (v_1, \dots, v_{m-1}, 0)$  belonging to  $H_{t_m=0}$ , and the projection of  $(C_m)$  onto the plane generated by  $v$  and  $t_m$ . In this plane, the curve is governed by the equation  $\begin{pmatrix} \sum_{i=1, m-1} v_i x^{d_i} \\ x^{d_m} \end{pmatrix}$ . Let  $k$  be the index of the first non null coordinate of  $v$ . The direction of the tangent at  $x = 0$  is thus given by  $\begin{pmatrix} v_k d_k \\ 0 \end{pmatrix}$ .

Suppose ad absurdum that the face not containing  $T_m$  is not included in  $H_{t_m=0}$ . Then, we can find a vector  $v$  for which the tangent at  $x = 0$  of the projection of  $(C_m)$  in the plane  $(v, t_m)$  has a non null second coordinate. Otherwise, the curve would not be included in the projection of  $(P'_m)$  on  $(v, t_m)$ . And this is impossible, because the second coordinate of the tangent is always zero. □

We can now proceed with the proof of theorem (2). As we consider mainly  $(P_m)$ , the subscript  $m$  is unnecessary and omitted in the notation of the vertices. It is only maintained for  $T_{m-1, m-1}$ .

*Proof.* Our geometrical considerations can be followed on figure 9. We proceed by recursion. Theorem (2) is easily seen to be true in dimension  $m = 2$  (see the left panel of figure 2).

Assume now that this property is true until dimension  $m - 1$ . We consider a simplex  $(P'_m)$  containing the curve  $(C_m)$ , and we are going to prove that  $(P'_m)$  contains also  $(P_m)$ .

1.  $T_0$  and  $T_m$  correspond to  $x = 0$  and  $x = 1$  respectively in the  $\mathbb{T}$  space and are necessarily two vertices of  $(P'_m)$  since they belong to the curve  $(C_m)$ .
2. The face of  $(P'_m)$  not containing  $T_m$  is included in the hyperplane  $H_{t_m=0}$  (see lemma 2).
3. The face of  $(P_m)$  not containing  $T_0$  is perpendicular to  $H_{t_m=0}$ . This results from lemma 1. Let us call  $H_{t_m=0}^\perp$  the affine hyperplane containing this face.
4. We cut now the simplex  $P'_m$  with  $H_{t_m=0}^\perp$ . The intersection always exists:  $T_m$  is contained in both  $P'_m$  and  $H_{t_m=0}^\perp$ . Since  $H_{t_m=0}^\perp$  passes through  $T_m$ , we create two simplices. Only one of them contains the curve  $(C_m)$ , since  $(C_m)$  can not cross  $H_{t_m=0}^\perp$ . We call it  $(P''_m)$ .

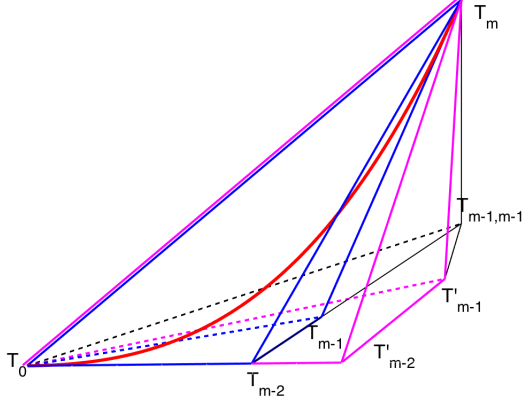


Figure 9:  $(P_m)=(T_0, T_{m-2}, T_{m-1}, T_m)$  in blue, and  $(P_{m-1})=(T_0, T_{m-2}, T_{m-1,m-1}, T_m)$  in black. The difference with figure 8 is that the simplex  $(P'_m)$  has been added in magenta and is represented by the four vertices  $(T_0, T'_{m-2}, T'_{m-1}, T_m)$ . The point  $T_m$  is the only point not belonging to the hyperplane  $H_{t_m=0}$ . The points  $T_{m-2}, T_{m-1}, T_{m-1,m-1}, T_m$  are in a hyperplane perpendicular to  $H_{t_m=0}$  called  $H_{t_m=0}^\perp$ .

5. Due to the fact that the face of  $(P''_m)$  not containing  $T_0$  is perpendicular to  $H_{t_m=0}$ , the orthogonal projection of  $(P''_m)$  on  $H_{t_m=0}$  is also a simplex. Thus, it contains  $(P_{m-1})$  by means of our recursion hypothesis. We can now conclude that all the vertices shared by  $(P_{m-1})$  and  $(P_m)$  are inside  $(P''_m)$  and thus inside  $(P'_m)$ .
6. By construction, only two points belong to  $(P_m)$  and not to  $(P_{m-1})$ :  $T_m$  and  $T_{m-1}$ .  $T_m$  is one of the vertices of  $(P'_m)$  (see the first item). We consider now  $T_{m-1} = (1, \frac{d_m-d_1}{d_m}, \dots, \frac{d_m-1}{d_m}, 0)$ .  $T_{m-1}$  is constructed as the intersection of the tangent to the curve  $(C_m)$  at the point  $T_m$  with the hyperplane  $H_{t_m=0}$ . This point is necessarily included in any simplex enclosing  $(C_m)$ , and thus inside  $(P'_m)$ .

□

### Theorem 3.

*Proof.* The proof is similar to the previous one in theorem (1). The  $x^{d_j}$  are simply replaced by  $f_j(x)$  everywhere. Due to the fourth item, the  $\lambda_j(x)$  are guaranteed to be positive. □

### Corollary (2).

*Proof.* For theorem (3) and corollary (2), the optimality comes from the fact that the face opposite to the point  $T_m$  (with all coordinates equal to 1) is always in the hyperplane  $H_{t_m=0}$ , as soon as the  $d_j$  are all different in the case of power law functions, or more generally the function  $f_j$  are independent in the vicinity of  $x = 0$ . For proposition (2), the result follows from symmetry reasons. □



**Theorem 4.**

As a preliminary lemma, we first prove this theorem when the terms in the curve  $C_m$  result from the tensor product  $\bigotimes_{k=1,v} (1, f_{k_1}(x_k), \dots, f_{k_{m_k}}(x_k))$ .

**Lemma 3.**

*Proof.* This is an immediate consequence of the preceding propositions. Indeed, we note that the issue here is to show that the curve resulting from the tensor product  $(1, f_{1_1}(x_1)), \dots, f_{1_{m_1}}(x_1)) \otimes \dots \otimes (1, f_{v_1}(x_v)), \dots, f_{v_{m_v}}(x_v))$  is in the convex hull of the points defined by

$$T_{j_1, \dots, j_v}(i_1, \dots, i_p) = \prod_{k=1}^v \frac{d_{j_k+1} - d_{i_k}}{d_{j_k+1}} \mathbb{1}_{i_1 \leq j_1, \dots, i_v \leq j_v}$$

$$T_{d_1, \dots, d_v} = 1$$

This is true since  $T_{j_1, \dots, j_v} = T_{j_1} \otimes \dots \otimes T_{j_v}$ .

In theorem (3), we showed that each individual curve  $(1, f_{k_1}(x_k), \dots, f_{k_{m_k}}(x_k))$  is enclosed in the convex polytope defined by the vertices  $T_{j_k}$  so that for some well chosen  $(\lambda_{j_k})_{j_k=1, d_k}$  we can write the decomposition:  $(1, f_{k_1}(x_k), \dots, f_{k_{m_k}}(x_k)) = \sum_{j_k=0, m_k} \lambda_{j_k}(x_k) T_{j_k}$ . Consequently,

$$\sum_{j_1, \dots, j_v} \lambda_{j_1}(x_1) \dots \lambda_{j_v}(x_v) T_{j_1, \dots, j_v} = \bigotimes_{j_k=1}^{d_k} \lambda_{j_k}(x_k) T_{j_k}$$

$$= \bigotimes (1, f_{k_1}(x_k), \dots, f_{k_{m_k}}(x_k)).$$

□

We can now proceed with the proof of theorem (4)

*Proof.* Any curve  $(C_m)$  can be completed as a curve  $(C'_m)$  whose terms come from a tensor product. Thus, any point of  $(C'_m)$  is decomposed in  $(C'_m) = \sum_{j_k=0}^{m'_k} \lambda_{j_k}(x_k) T'_{j_k}$ , where the  $T'_{j_k}$  are the vertices of  $(C'_m)$ .

Alternatively,  $(C_m)$  is defined as a projection  $Pr$  of  $(C'_m)$  on the adequate space. Consequently, the polytope defined by the projection of the vertices  $T'$  enclose  $(C_m)$ . As a matter of fact, each of these projected vertices is one of the  $T$ , some of the  $T'$  being projected on the same  $T$ . Then the previous relation becomes  $(C_m) = \sum_{j_k=0}^{m'_k} \lambda_{j_k}(x_k) Pr(T'_{j_k})$ . □

**Corollary (3).**

*Proof.* The first part of corollary 3 is a direct consequence of theorem 4 : the number of constraints is the product of the number of possible values for each  $(j_k)_{k=1,v}$ . This is also the number of monomials in  $P(x_1, \dots, x_v)$  plus one, since we must take into account the constant factor in the function  $P(x)$ .

The second part of this corollary when the terms in  $P(x)$  result from a tensor product can be explained this way: in this case, the convex hull of the  $m + 1$

points  $T_{j_k}$  is by definition a simplex, the convex polytope with the minimal number of vertices in a space of dimension  $m + 1$ . We are then in exactly the same situation as before in theorem (2).

In general, with any kind of  $P(x)$ , the projection property will apply in the same manner as in (4). □

## References

- [1] A. Antoniadis, J. Bigot, and I. Gijbels. Penalized wavelet monotone regression. *Statistics and Probability Letters*, 77(16):1608 – 1621, 2007.
- [2] R. Barlow, D. Bartholomew, J. Bremner, and H. Brunk. *Statistical Inference under Order Restrictions: Theory and Application of Isotonic Regression*. John Wiley and Sons Inc, 1978.
- [3] O. Burdakow, G. Grimvall, and M. Hussian. A generalised pav algorithm for monotonic regression in several variables. In J. Antoch, editor, *COMPSTAT, Proceedings in Computational Statistics, 16th Symposium*. Springer-Verlag, 2004.
- [4] 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.
- [5] C. De Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 2nd edition, 1978.
- [6] H. Dette and R. Scheder. Strictly monotone and smooth nonparametric regression for two or more variables. *The Canadian Journal of Statistics*, 34(4):535–561, 2006.
- [7] 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.
- [8] É. Falize and the PAMAL group. Recent advances in the experimental simulation of x-ray binary stars accretion shocks. In *9th Conference on High Energy Density Laboratory Astrophysics*, Tallahassee, Florida, May 2012.
- [9] P. Hall and L. Huang. Nonparametric kernel regression subject to monotonicity constraints. *Annals of Statistics*, 29:624–647, 2001.
- [10] F. Lauer and G. Bloch. Incorporating prior knowledge in support vector regression. *Machine Learning*, 2008.
- [11] E. Mammen, J.S. Marron, B.A. Turlach, and M. P. Wand. A general projection framework for constrained smoothing. *Statistical Science*, 16(3):232–248, 2001.

- [12] J.S. Marron, B.A. Turlach, and M.P. Wand. Local polynomial smoothing under qualitative constraints. *In: L. Billard and N.I. Fisher (Eds), Graph-Image-Vision*, 28:647–652, 1997.
- [13] M.C. Meyer. Constrained penalized splines. *Canadian Journal of Statistics*, 40(1):190–206, 2012.
- [14] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.
- [15] N. Pya and S. N. Wood. Shape constrained additive models. *Statistics and Computing*, 25(3):543–559, 2015.
- [16] J.S. Racine, C.F. Parmeter, and P. Du. Constrained nonparametric kernel regression: Estimation and inference. working paper, 2009.
- [17] J.O. Ramsay. Monotone regression splines in action. *Statistical Science*, 3(4):425–441, 1988.
- [18] H. Trevor, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2nd edition, 2009.
- [19] L. Wang and L. Xue. Generalised additive and index models with shape constraints. *Journal of Statistical Planning and Inference*, 167:27–40, 2015.
- [20] S. Wood. *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2006.
- [21] S. N. Wood. Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467):673–686, 2004.
- [22] S.N. Wood, Y. Goude, and S. Shaw. Generalized additive models for large data sets. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64(1):139–155, 2015.