



HAL
open science

POLYNOMIAL ARGMIN FOR RECOVERY AND APPROXIMATION OF MULTIVARIATE DISCONTINUOUS FUNCTIONS

Didier Henrion, Milan Korda, Jean-Bernard Lasserre

► **To cite this version:**

Didier Henrion, Milan Korda, Jean-Bernard Lasserre. POLYNOMIAL ARGMIN FOR RECOVERY AND APPROXIMATION OF MULTIVARIATE DISCONTINUOUS FUNCTIONS. *Constructive Approximation*, 2026, <10.1007/s00365-026-09731-0>. <hal-03986252v3>

HAL Id: hal-03986252

<https://laas.hal.science/hal-03986252v3>

Submitted on 7 Jul 2025

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.



HAL Authorization

POLYNOMIAL ARGMIN FOR RECOVERY AND APPROXIMATION OF MULTIVARIATE DISCONTINUOUS FUNCTIONS

DIDIER HENRION^{1,2}, MILAN KORDA^{1,2}, JEAN BERNARD LASSERRE^{1,3}

ABSTRACT. We propose to approximate a (possibly discontinuous) multivariate function $f(\mathbf{x})$ on a bounded set by the partial minimizer $\arg \min_y p(\mathbf{x}, y)$ of an appropriate polynomial p whose construction can be cast in a *univariate* sum of squares (SOS) framework, resulting in a highly structured convex semidefinite program. In a number of non-trivial cases (e.g. when f is a piecewise polynomial) we prove that the approximation is exact with a low-degree polynomial p . Our approach has three distinguishing features: (i) It is mesh-free and does not require the knowledge of the discontinuity locations. (ii) It is model-free in the sense that we only assume that the function to be approximated is available through samples (point evaluations). (iii) The size of the semidefinite program is independent of the ambient dimension and depends linearly on the number of samples. We also analyze the sample complexity of the approach, proving a generalization error bound in a probabilistic setting. This allows for a comparison with machine learning approaches.

1. INTRODUCTION

Approximation of discontinuous functions in multiple dimensions is a notoriously difficult problem and a scientific challenge. A common strategy (e.g. described in [32]) which works well in the univariate setting (and is implemented for example in the `chebfun` package [11]) consists of the following steps : 1) detect the discontinuity locations and split the domain into a disjoint union of regions where the function is continuous, and 2) construct approximations of the continuous pieces on each region. However, in the multivariate case this strategy is very challenging to implement since the discontinuity set may have a positive dimension (see, e.g, [17] where an algorithm for detecting discontinuities in two dimensions is proposed).

For instance, in the *Variable Scaled Discontinuous Kernel* (VSDK) method described e.g. in [12] the authors provide interesting numerical experiments but the method indeed assumes prior knowledge of the set of discontinuities¹. In addition, the complexity of the approximant increases with the the sample size. On the other hand, *Weighed Essentially Non-Oscillatory* (WENO) methods as described in e.g. [28] (and initially designed for hyperbolic PDEs) do not assume prior knowledge of discontinuities. However, such methods construct *local* models based on an increasing number of samples (at every candidate input point of the algorithm for approximation) and so do not provide a global model of the discontinuous function to approximate. Numerical difficulties faced with approximating multivariate functions are illustrated in the example sections of the paper.

¹CNRS; LAAS; Université de Toulouse, 7 avenue du colonel Roche, F-31400 Toulouse, France.

²Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, CZ-16626 Prague, Czechia.

³Institute of Mathematics; Université de Toulouse, 118 route de Narbonne, F-31062 Toulouse, France.

The research of M. Korda and J. B. Lasserre research is partly supported by AI Interdisciplinary Institute ANITI funding, through the French “Investing for the Future PIA3” program under the Grant agreement n°ANR-19-PI3A-0004. This research is also part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme and by the European Union under the project ROBOPROX (reg. no. CZ.02.01.01/00/22 008/0004590). J.B. Lasserre also acknowledges support from ANR-NuSCAP-20-CE48-0014.

¹Indeed, in [12, p. 442] the authors state “*The only drawback of the procedure lies in the fact that the algorithm needs to know where the discontinuities occur*”.

A typical and important application is concerned with classification in data analysis and supervised learning, where powerful deep learning methods have obtained impressive results and success stories. However, such powerful methods still have some limitations (even for learning continuous functions, let alone discontinuous functions). Indeed for instance and quoting [4], “*Despite many results that establish the existence of Neural Nets (NNs) with excellent approximation properties, algorithms that can compute these NNs only exist in specific cases.*” That is, no training algorithm can obtain them in the general case. For an interesting discussion about such limits (instability, accuracy, etc.) the interested reader is referred to [3, 4] and references therein. For learning discontinuous functions by neural networks, [16] proposes a tailored architecture; however this approach requires knowledge of discontinuity locations and is limited to univariate problems.

This paper is a follow-up (but non trivial extension) of [21]. We provide an alternative approximation technique aimed at dealing with such discontinuities and the Gibbs phenomenon, which are large oscillations of the approximation near the discontinuity points, see e.g. [30, Chapter 9]. In particular, we show that our class of approximants can model exactly multivariate piecewise polynomial functions and can approximate with arbitrary accuracy other discontinuous functions.

Contribution. We introduce a new class of approximants for a possibly discontinuous function f from $\mathbf{X} \subset \mathbb{R}^n$ to $\mathbf{Y} \subset \mathbb{R}$. We propose to approximate f by the *polynomial argmin*

$$(1.1) \quad \mathbf{x} \mapsto \hat{f}(\mathbf{x}) := \min \{ \arg \min_{y \in \mathbf{Y}} p(\mathbf{x}, y) \}, \quad \mathbf{x} \in \mathbf{X},$$

where $\mathbf{Y} \supset f(\mathbf{X})$ and $p \in \mathbb{R}[\mathbf{x}, y]$ is a polynomial in (\mathbf{x}, y) .

The main features of our approach can be summarized as follows.

- The approach is mesh-free and does not require the knowledge of the discontinuities locations.
- It is not limited to univariate functions or tensor products thereof.
- It is model-free, working only with the samples of the unknown function.
- A polynomial p for our approximant (1.1) is constructed using a very specific class of convex optimization (semidefinite programming) problems whose size depends (linearly) on the number of samples and is *independent* of the ambient dimension.
- The approximant is simple to evaluate as it is the argmin of a *univariate* polynomial.
- We provide a generalization error analysis in a probabilistic setting.

In addition we also provide a result which is interesting in its own and justifies the use of the argmin approximation. Namely, we prove that any piecewise (possibly discontinuous) polynomial function $f : \mathbf{X} \rightarrow \mathbb{R}$ on a bounded set $\mathbf{X} \subset \mathbb{R}^d$ has an *exact* “arg min” representation. Specifically, provided that the partition that defines the regions of continuity of f is of a certain kind, there exists a polynomial p (not unique in general) such that

$$f(\mathbf{x}) = \arg \min_{y \in \mathbb{R}} p(\mathbf{x}, y)$$

for all $\mathbf{x} \in \mathbf{X}$, except for points of discontinuity of f . In other words, \hat{f} in (1.1) coincides with f (except for discontinuity points). Moreover and importantly, when f is known only from a sample of its values, such a polynomial p can be retrieved by our algorithm, as p belongs to its set of possible optimal outputs; see Remark 3.4.

We believe that these features make the approach a unique and promising tool with a wide variety of applications in data analysis. This is corroborated by a numerical evidence where we observe a remarkable performance on a range of examples. We also provide a solid theoretical underpinning of the method but leave some questions open, including the optimal rate of convergence of the argmin approximant.

Prior work and novelty. The idea behind our new approximant is a non-trivial extension of our previous work [21] that has provided an approximant which is the argument of the partial minimum (argmin), of a sum of squares (SOS) of polynomials, in fact the reciprocal of the Christoffel function of a measure, ideally supported on the graph of the function to recover. This recovery procedure can be seen as a non-standard application of the Christoffel-Darboux kernel. Importantly, being in a class of functions much larger than polynomials, such an approximant is able to approximate some discontinuous functions much better than polynomials can. Remarkably, in non-trivial examples, recovery is possible without oscillations and Gibbs phenomenon usually encountered in several more standard approaches. In this respect the reader is referred to the detailed discussion in [31] on kernel variants (e.g Féjer or Jackson kernels) to attenuate the Gibbs phenomenon encountered with polynomial approximations.

While in [20] our original motivation for introducing the polynomial argmin approximant was to recover the solution of a nonlinear partial differential equation from the knowledge of its approximate moments, this strategy was made rigorous and generalized to graph recovery from moments in [21]. Then it was later extended to cope with partial moment information [15]. Following our initial work, this argmin strategy was called “implicit model” and used in robotics applications [14] where the polynomial p in (1.1) is replaced by a continuous function computed by training, e.g. a neural network.

A novelty and distinguishing feature of the present paper with respect to [21] is that the polynomial p in (1.1) is *not* restricted to be the reciprocal of the Christoffel function associated with the measure supported on the graph of f . Indeed, our set of potential candidate polynomials p is now a suitably chosen subspace of $\mathbb{R}[\mathbf{x}, y]$, which is much larger than the set considered in [21]. In addition, from a computational perspective, our method has a much more favorable behavior with respect to the ambient dimension. While in [21] the size of the moment matrix whose Cholesky factorization is used to construct the Christoffel-Darboux kernel grows rapidly with respect to the ambient dimension, the size of the semidefinite programs (SDPs) solved in this work is independent of the dimension.

Outline. In the motivational Section 2 we show that our polynomial argmin strategy is already efficient in some non-trivial cases. For instance, it allows *exact recovery* when f is a polynomial, or an algebraic function, or a (possibly discontinuous) piecewise polynomial.

However, exact recovery by a polynomial argmin cannot be guaranteed in general, and in Section 3 we provide a numerical scheme to obtain the polynomial p which appears in (1.1), when knowledge on f is only through its finitely many values on a sample of points $(\mathbf{x}(i))_{i \in I} \subset \mathbf{X}$ (and without any a priori knowledge on its distribution). We reformulate our polynomial argmin strategy in the framework of univariate sum of squares (SOS) positivity certificates so that finding p amounts to solving a semidefinite optimization problem whose size is controlled by the degree of p in y and the sample size.

Finally, in Section 4 we also provide a set of numerical experiments to evaluate the efficiency of our proposed polynomial argmin strategy on a sample of problems. We observe that it performs remarkably well to approximate challenging discontinuous functions already tested in [21], as well as non trivial two-dimensional examples of functions whose set of discontinuities has positive dimension. We have also compared with machine learning methods based on neural networks. In all our experiments, the proposed method achieves far better accuracy with simpler representation of the approximant.

2. EXACT REPRESENTATIONS

The purpose of this section is to demonstrate the expressive power of the polynomial argmin. We do so by showing that for several classes of known functions, an exact (and often simple) “argmin” representation is possible. Subsequently, in Section 3, we show how a polynomial arg min approximation can be found using convex optimization, provided that a collection of finite samples of values of f is available.

Notation and definitions. Let $\mathbb{R}[\mathbf{x}, y]$ denote the ring of polynomials in the variables $\mathbf{x} \in \mathbb{R}^n$ and $y \in \mathbb{R}$, and $\mathbb{R}[\mathbf{x}, y]_d$ its subset of polynomials of total degree at most d . A polynomial p is a sum of squares (SOS) if it can be written as $\sum_k p_k^2$ for finitely many polynomials p_k . The convex cone of all SOS polynomials of degree at most d in the variable \mathbf{x} is denoted by $\Sigma_d[\mathbf{x}]$.

2.1. Representation of Polynomials. When $f \in \mathbb{R}[\mathbf{x}]$ is a given polynomial, in (1.1) choose

$$(\mathbf{x}, y) \mapsto p(\mathbf{x}, y) := \frac{1}{2}y^2 - f(\mathbf{x})y, \quad \forall \mathbf{x}, y.$$

Indeed, we observe that $\frac{dp}{dy} = y - f(\mathbf{x})$ and hence $y = f(\mathbf{x})$ is a stationary point. Since p is strictly convex in y , it follows that $y = f(\mathbf{x})$ is the global minimizer. The degree of p in x is equal to the degree of f whereas the degree in y is equal to two irrespective of f .

2.2. Representation of Algebraic functions. An *algebraic function* f is such that $q_k(\mathbf{x}, f(\mathbf{x})) = 0$ for some given polynomials $q_k \in \mathbb{R}[\mathbf{x}, y]$, $k = 1, \dots, m$.

If for each \mathbf{x} , $(\mathbf{x}, f(\mathbf{x}))$ is the unique common zero of the q_k then choose

$$(\mathbf{x}, y) \mapsto p(\mathbf{x}, y) := \sum_{k=1}^m q_k(\mathbf{x}, y)^2, \quad \forall \mathbf{x}, y,$$

and observe that the degree of p in \mathbf{x} and y is twice the maximal degree of the q_k in these variables.

Note that *semi-algebraic functions*² can also be modeled like that, provided that the inequalities are incorporated in the definitions of the domain \mathbf{X} and image sets Y .

Example 1. *The absolute value function can be expressed as*

$$|x| = \arg \min_{y \in Y} (x^2 - y^2)^2$$

with $\mathbf{Y} := [0, 1]$, for all $x \in \mathbf{X} := [-1, 1]$. Note that a more complicated degree 8 polynomial argmin model for the absolute value function was already described in [21, Example 2].

2.3. Representation of piecewise constant functions. Let $\mathbf{X} \subset \mathbb{R}^n$ be bounded and $\mathbf{Y} = \mathbb{R}$. Given N polynomials $g_1, \dots, g_N \in \mathbb{R}[\mathbf{x}]$, define the cells

$$(2.1) \quad \mathbf{X}_i = \{\mathbf{x} \in \mathbf{X} \mid g_i(\mathbf{x}) < g_j(\mathbf{x}), \forall j \neq i\}, \quad i = 1, \dots, N,$$

that partition \mathbf{X} . This model is motivated by the proof technique used and at the same time is sufficiently general to cover many situations encountered in practice, including box partitions and partitions where each cell is defined by a sublevel set of a single polynomial. We now briefly discuss these partitions and then detail the argmin representation of piecewise constant and piecewise polynomial functions defined thereon.

Consider first $\mathbf{X} = [-1, 1]$ split into N intervals $\mathbf{X}_i = (a_i, a_{i+1})$, $i = 1, \dots, N$, with $a_i < a_{i+1}$, $a_1 = -1$ and $a_{N+1} = 1$. Then $g_i(x) = (x - a_i)(x - a_{i+1})$ satisfies $g_i(x) < g_j(x)$ for all $j \neq i$ if and only if $x \in \mathbf{X}_i$.

²A semi-algebraic function is such that its graph is described by a finite union of a finite intersection of sets defined by polynomial equations and inequalities.

This example is readily generalized to $\mathbf{X} = [-1, 1]^n$ split into axis-aligned boxes $\mathbf{X}_{\mathbf{i}} = \times_{k=1}^n (a_{i_k}^k, a_{i_k+1}^k)$, where $a_{k,1}, \dots, a_{k,m_k}$ (with $m_k \in \mathbb{N}$, $a_{k,1} = -1$, $a_{k,m_k} = 1$) define the breakpoints of each coordinate axis $k \in \{1, \dots, n\}$ and $\mathbf{i} = (i_1, \dots, i_n)$ is a multi-index with $i_k \in \{1, \dots, m_k\}$. In that case, we can define $g_{\mathbf{i}}(\mathbf{x}) = \sum_{k=1}^n (x_k - a_{i_k}^k)(x_k - a_{i_k+1}^k)$. Then $g_{\mathbf{i}}(\mathbf{x}) < g_{\mathbf{j}}(\mathbf{x})$ for all multi-indices $\mathbf{j} \neq \mathbf{i}$ if and only if $\mathbf{x} \in \mathbf{X}_{\mathbf{i}}$.

Finally, partitions of \mathbf{X} into disjoint cells \mathbf{X}_i of the form

$$\mathbf{X}_i = \{ \mathbf{x} \mid g_i(\mathbf{x}) < 0 \}, \quad i = 1, \dots, N,$$

that is, defined by the sublevel set of a single polynomial g_i , can also be readily modeled using (2.1). Indeed, since \mathbf{X}_i 's are disjoint, we have $\mathbf{x} \in \mathbf{X}_i$ if and only if $g_i(\mathbf{x}) < 0$ and $g_j(\mathbf{x}) > 0$ for all $j \neq i$; hence also $g_i(\mathbf{x}) < g_j(\mathbf{x})$ for all $j \neq i$. Cartesian products of cells of this kind can be modeled using the same trick as described for boxes (i.e., cartesian products of intervals).

Now we discuss the argmin representation of piecewise constant functions defined on the partition (2.1); generalization to piecewise polynomials is in §2.4. Given N distinct³ real numbers c_1, \dots, c_N , the piecewise constant function f is defined by $f(\mathbf{x}) = c_i$ whenever $\mathbf{x} \in \mathbf{X}_i$. The value of f at the boundary points of the partition, i.e., when $g_i(\mathbf{x}) = g_j(\mathbf{x})$ for some $i \neq j$ is left arbitrary as in these regions one cannot hope for exact representation using the argmin.

Theorem 2.1. *Let $f : \mathbf{X} \rightarrow \mathbb{R}$, \mathbf{X} bounded, be a piecewise constant function satisfying $f(\mathbf{x}) = c_i$ for $\mathbf{x} \in \mathbf{X}_i$ with c_1, \dots, c_N distinct. Then there exists $p \in \mathbb{R}[\mathbf{x}, y]$ such that*

$$f(\mathbf{x}) = \arg \min_{y \in \mathbb{R}} p(\mathbf{x}, y) \quad \text{for all } \mathbf{x} \in \bigcup_{i=1}^N \mathbf{X}_i.$$

Proof. Let

$$L_i(y) = \prod_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{y - c_j}{c_i - c_j}$$

be the degree- $(n-1)$ Lagrange interpolation polynomial at c_i . Define the degree- $(2n-1)$ Hermite-cardinal polynomial

$$H_i(y) = (1 - 2L_i'(c_i)(y - c_i)) [L_i(y)]^2.$$

This polynomial satisfies $H_i(c_j) = 1$ if $i = j$ and $H_i(c_j) = 0$ if $i \neq j$. Furthermore $H_i'(c_j) = 0$ for all i, j . Define

$$p(\mathbf{x}, y) = \sum_{i=1}^N g_i(\mathbf{x}) H_i(y) + M \prod_{i=1}^N (y - c_i)^2,$$

where M is a sufficiently large constant ensuring that $p(\mathbf{x}, y)$ is coercive in y for every $\mathbf{x} \in \mathbf{X}$. Such constant exists since \mathbf{X} is bounded. The coercive penalty ensures that for every $\mathbf{x} \in \mathbf{X}$, the global minimum of $p(\mathbf{x}, \cdot)$ is attained at a critical point of $p(\mathbf{x}, \cdot)$.

Now, since $H_i'(c_j) = 0$ for all i, j , we have $(\partial p / \partial y)(\mathbf{x}, y) = 0$ if $y \in \{c_1, \dots, c_N\}$. Therefore, for each \mathbf{x} , the constants c_1, \dots, c_N are the only critical points of $p(\mathbf{x}, \cdot)$. The coercive penalty ensures that c_1, \dots, c_N are strict local minima for sufficiently large M . Furthermore, since $H_i(c_j) = 1$ if $i = j$ and zero otherwise, we have $p(\mathbf{x}, c_i) = g_i(\mathbf{x})$. Therefore, among the local minima c_1, \dots, c_N , the one with the lowest value of $g_i(\mathbf{x})$ attains the smallest value of p as desired. There are $N-1$ additional critical points ξ_i of $p(\mathbf{x}, y)$, each lying in the interval (c_i, c_{i+1}) (assuming without loss that c_i 's are ordered). These additional critical points are simple roots of $(\partial p / \partial y)(\mathbf{x}, \cdot)$. Since $\xi_i \in (c_i, c_{i+1})$ and each c_i is a strict local minimum, each ξ_i must be a strict local maximum and hence not a candidate for a minimizer of $p(\mathbf{x}, \cdot)$. This concludes the proof. \square

³If the constants c_i are not all distinct, then the cells associated with each subset of indices sharing the same value must be merged, thereby producing a new (coarser) partition with distinct values on each cell.

2.4. Piecewise polynomial functions. Theorem 2.1 can be generalized to the case of piecewise polynomial functions defined over the same partition simply by replacing the constants c_i by polynomials $h_i(\mathbf{x})$. The only caveat is that the constants c_i appear in the denominator and hence the resulting function is rational. However, given that the argmin of a function is not changed when multiplied by a positive number, we can consider the polynomial

$$\tilde{p}(\mathbf{x}, y) = p(\mathbf{x}, y) \prod_{\substack{1 \leq i, j \leq n \\ j \neq i}} (h_i(\mathbf{x}) - h_j(\mathbf{x}))^4,$$

where all polynomials appearing in the denominator of p are canceled. Therefore \tilde{p} is a polynomial in (\mathbf{x}, y) . Hence we arrive at the following corollary:

Corollary 2.2. *Let $\mathbf{X} \subset \mathbb{R}^n$ be bounded, and let \mathbf{X}_i be as in (2.1), $i = 1, \dots, N$. Let $f : \mathbf{X} \rightarrow \mathbb{R}$ be a piecewise polynomial function satisfying $f(\mathbf{x}) = h_i(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}_i$ with $h_1, \dots, h_N \in \mathbb{R}[\mathbf{x}]$. Then there exists $\tilde{p} \in \mathbb{R}[\mathbf{x}, y]$ such that*

$$f(\mathbf{x}) = \arg \min_{y \in \mathbb{R}} \tilde{p}(\mathbf{x}, y) \quad \text{for all } \mathbf{x} \in \bigcup_{i=1}^N \mathbf{X}_i \setminus \mathbf{A},$$

where $\mathbf{A} = \{\mathbf{x} \mid h_i(\mathbf{x}) = h_j(\mathbf{x}) \text{ for some } i \neq j\}$.

We note that the removal of the points \mathbf{x} where $h_i(\mathbf{x}) = h_j(\mathbf{x})$ cannot be easily avoided since for those \mathbf{x} , the polynomial \tilde{p} is identically zero. However, the Lebesgue measure of \mathbf{A} is zero, unless $h_i(\mathbf{x}) = h_j(\mathbf{x})$ for all \mathbf{x} , in which case cell merging can be carried out (see Footnote 3).

The construction via the proof of Theorem 2.1 (and hence of Corollary 2.2) is not unique. For instance in the case of two pieces, let $g, p_1, p_2 \in \mathbb{R}[\mathbf{x}]$ and consider the piecewise polynomial function

$$(2.2) \quad \mathbf{x} \mapsto f(\mathbf{x}) := \begin{cases} p_1(\mathbf{x}) & \text{if } g(\mathbf{x}) \in (0, 1) \\ p_2(\mathbf{x}) & \text{if } g(\mathbf{x}) \in (-1, 0) \end{cases}, \quad \mathbf{x} \in \mathbf{X},$$

where $\mathbf{X} := \{\mathbf{x} : g(\mathbf{x}) \in (-1, 0)\} \cup \{\mathbf{x} : g(\mathbf{x}) \in (0, 1)\}$.

Lemma 2.3. *Let f and \mathbf{X} be as in (2.2). Then there exists $p \in \mathbb{R}[\mathbf{x}, y]$ such that $f(\mathbf{x}) = \arg \min_{y \in \mathbf{Y}} p(\mathbf{x}, y)$ for all $\mathbf{x} \in \mathbf{X}$, with $\mathbf{Y} = \mathbb{R}$.*

Proof. With $r \in \mathbb{R}[\mathbf{x}]$ and $q \in \mathbb{R}[\mathbf{x}, y]$, let $p(\mathbf{x}, y) := (y - p_1(\mathbf{x}))^2(y - p_2(\mathbf{x}))^2 + r(\mathbf{x})q(\mathbf{x}, y)$ be such that $\partial q(\mathbf{x}, y)/\partial y = 6(y - p_1(\mathbf{x}))(y - p_2(\mathbf{x}))$. For instance $q(\mathbf{x}, y) := 2y^3 - 3(p_1(\mathbf{x}) + p_2(\mathbf{x}))y^2 + 6p_1(\mathbf{x})p_2(\mathbf{x})y$.

Now observe that for each given $\mathbf{x} \in \mathbf{X}$, $y \mapsto p(\mathbf{x}, y)$ is a coercive quartic univariate polynomial, and hence it has at most two local minima. Letting $r(\mathbf{x}) := g(\mathbf{x})(p_1(\mathbf{x}) - p_2(\mathbf{x}))$, the gradient $\partial p(\mathbf{x}, y)/\partial y$ vanishes at the critical points $p_1(\mathbf{x})$ resp. $p_2(\mathbf{x})$ resp. $p_3(\mathbf{x}) := (p_1(\mathbf{x})(1 - 3g(\mathbf{x})) + p_2(\mathbf{x})(1 + 3g(\mathbf{x}))/2$. At the critical points, the Hessian $\partial^2 p(\mathbf{x}, y)/\partial y^2$ is equal to $2(p_1(\mathbf{x}) - p_2(\mathbf{x}))^2(1 + 3g(\mathbf{x}))$ resp. $2(p_1(\mathbf{x}) - p_2(\mathbf{x}))^2(1 - 3g(\mathbf{x}))$ resp. $(p_1(\mathbf{x}) - p_2(\mathbf{x}))^2(-1 + 3g(\mathbf{x}))(1 + 3g(\mathbf{x}))$.

To compare the values at the critical points, we evaluate the differences $p(\mathbf{x}, p_2(\mathbf{x})) - p(\mathbf{x}, p_1(\mathbf{x})) = g(\mathbf{x})(p_1(\mathbf{x}) - p_2(\mathbf{x}))^4$, $p(\mathbf{x}, p_3(\mathbf{x})) - p(\mathbf{x}, p_2(\mathbf{x})) = (1 + g(\mathbf{x}))(1 - 3g(\mathbf{x}))^3(p_1(\mathbf{x}) - p_2(\mathbf{x}))^4/16$, $p(\mathbf{x}, p_3(\mathbf{x})) - p(\mathbf{x}, p_1(\mathbf{x})) = (-1 + g(\mathbf{x}))(1 + 3g(\mathbf{x}))^3(p_1(\mathbf{x}) - p_2(\mathbf{x}))^4/16$. The proof follows by evaluating the signs of the Hessian and the differences at the critical points for $g(\mathbf{x})$ within the intervals $(-\infty, -1)$, $(-1, -1/3)$, $(-1/3, 0)$, $(0, 1/3)$, $(1/3, 1)$, $(1, \infty)$, see Table 1. \square

Example 2. *In [21, Example 1] a degree 8 polynomial argmin model was described for the sign function $f(x)$ which is equal to -1 if $x \in [-1, 0)$ and $+1$ if $x \in (0, 1]$. A simpler representation is obtained via Lemma 2.3 with $p(x, y) = (y + 1)^2(y - 1)^2 + 4xy(y^2 - 3)$. Here we consider $\mathbf{Y} = \mathbb{R}$ and $\mathbf{X} = [-1, 1]$. Figure 1 depicts the polynomial $p(x, y)$ for different values of x .*

	$(-\infty, -1)$	$(-1, -1/3)$	$(-1/3, 0)$	$(0, 1/3)$	$(1/3, 1)$	$(1, \infty)$
p_1	max	max	min	min	min	min
p_2	min	min	max	max	max	max
p_3	min	min	max	max	min	min
argmin	p_3	p_2	p_2	p_1	p_1	p_3

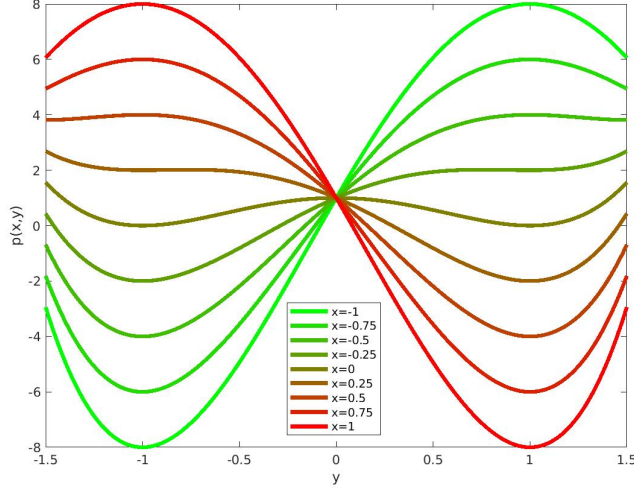
TABLE 1. Nature of critical points for different intervals of variation of $g(\mathbf{x})$.

FIGURE 1. Degree 4 polynomial $p(x, y)$ whose argmin w.r.t. y models the sign function. Represented are univariate polynomials $y \mapsto p(x, y)$ for various given values of x . We observe that for positive values of x , the argmin is precisely $+1$ whereas for negative values of x , the argmin is precisely -1 as required in order to represent the function $\text{sign}(x)$.

An even simpler argmin model of the sign function on $\mathbf{X} = [-1, 1]$ is $p(x, y) = -xy$ with $\mathbf{Y} = [-1, 1]$. Similarly to Example 1, the choice of domain and image sets \mathbf{X}, \mathbf{Y} plays here a key role.

Example 3. The indicator function of the bivariate unit disk $\{\mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 \leq 1\}$ can be modeled exactly on $\mathbf{X} := \{\mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 \leq 2\}$ via Lemma 2.3, as the degree 5 polynomial argmin of $p(\mathbf{x}, y) = y^2((y-1)^2 + (1-x_1^2-x_2^2)(3-2y))$

Example 4. The univariate function

$$x \mapsto f(x) = \max\{x, 0\}, \quad x \in (-1, 1),$$

often called the rectified linear unit or *ReLU*, is widely used in optimization and machine learning. One may ask whether there is a polynomial $p(x, y)$ for which $f(x)$ arises as $\arg \min_y p(x, y)$.

By Lemma 2.3, and with $p_1(x) = x$, $p_2(x) = 0$, $g(x) = x$, one can check that $p(x, y) = y^2((y-x)^2 + (2y-3x)x^2)$ is such that $\arg \min_{y \in \mathbb{R}} p(x, y) = \max(0, x)$ for all $x \in (-1, 1)$.

3. SAMPLE-BASED FORMULATION

Before describing our numerical scheme we first state a classical result on certificate of non-negativity in real algebraic geometry, which plays a key role in our approach.

3.1. Nonnegativity of univariate polynomials on an interval. The following result whose second part is due to F. Lukács is classical (see, e.g., [27, p. 4681] for a discussion).

Theorem 3.1. *A univariate polynomial $q \in \mathbb{R}[y]$ of degree d is nonnegative on \mathbb{R} if and only if $q \in \Sigma_d[y]$. A univariate polynomial $q \in \mathbb{R}[y]$ of degree d is nonnegative on the interval $[a, b] \subset \mathbb{R}$ if and only if*

$$\begin{cases} q = \sigma_0 + \sigma_1(b-y)(y-a), & \sigma_0 \in \Sigma_d[y], \quad \sigma_1 \in \Sigma_{d-2}[y] & d \text{ even,} \\ q = \sigma_0(y-a) + \sigma_1(b-y), & \sigma_0 \in \Sigma_{d-1}[y], \quad \sigma_1 \in \Sigma_{d-1}[y] & d \text{ odd.} \end{cases}$$

It is a simple observation that a polynomial σ belongs to Σ_d (for d even) if and only if there exists a positive semi-definite matrix $W \succeq 0$ such that $\sigma(y) = v_{d/2}(y)Wv_{d/2}(y)$, where $v_{d/2}$ is a basis of $\mathbb{R}[y]_{d/2}$, e.g., $v_{d/2} = [1, y, y^2, \dots, y^{d/2}]$. Therefore the nonnegativity of $q \in \mathbb{R}[y]$ on \mathbb{R} or $[a, b]$ is *equivalent* to the feasibility of a semidefinite programming (SDP) problem⁴. Observing that the conditions in Theorem 3.1 are affine in the coefficients of q , we conclude that one can also *optimize* over the set of polynomials (of fixed degree) nonnegative over $[a, b]$ using Semidefinite Programming; see e.g. [18, 19].

3.2. A numerical scheme. Given a function $f : \mathbb{R}^n \rightarrow \mathbf{Y}$ with $\mathbf{Y} = [a, b] \subset \mathbb{R}$ sampled at points $(\mathbf{x}_i, y_i)_{i=1}^N$ with $y_i = f(\mathbf{x}_i)$, we wish to construct an approximation \hat{f}_d of the form

$$(3.1) \quad \mathbf{x} \mapsto \hat{f}(\mathbf{x}) := \arg \min_{y \in \mathbf{Y}} p(\mathbf{x}, y),$$

where $p \in \mathbb{R}[\mathbf{x}, y]$ is a polynomial to be determined. The set $\mathbf{Y} \subset \mathbb{R}$ serves as a priori information on the range of f . If no such information is available, we set $\mathbf{Y} = \mathbb{R}$ (see Remark 3.3 for more details).

Throughout this section we assume that the argmin is unique. If this is not the case, a tiebreaker rule has to be applied (e.g., one can consider the min of the argmin). Since monomials not containing y do not influence the argmin, we use the parametrization of p as

$$(3.2) \quad (\mathbf{x}, y) \mapsto p(\mathbf{x}, y) := \sum_{k=1}^{d_y} h_k(\mathbf{x})y^k, \quad \forall \mathbf{x}, y,$$

where $h_k \in \mathbb{R}[\mathbf{x}]_{d_x}$ are polynomials of total degree at most d_x to be determined. In order to find h_k , we propose to solve the following convex optimization problem parametrized by $d_x, d_y, d_\gamma \in \mathbb{N}$ and $\alpha > 0$:

$$(3.3) \quad \begin{aligned} \min_{\gamma \in \mathbb{R}[\mathbf{x}, y]_{d_\gamma}, (h_k \in \mathbb{R}[\mathbf{x}]_{d_x})_{k=1}^{d_y}} & \int_{\mathbf{X} \times [a, b]} \gamma(x, y) \, d\mathbf{x}dy \\ \text{s.t.} & \quad p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2 \geq 0, \quad \forall y \in [a, b], \quad i = 1, \dots, N \\ & \quad \gamma(\mathbf{x}_i, y_i) \geq 0, \quad i = 1, \dots, N, \end{aligned}$$

where p is parametrized using h_k as in (3.2). The rationale behind (3.3) is as follows: If $\gamma(\mathbf{x}_i, y_i) = 0$, then $p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) \geq \alpha(y - y_i)^2$ for all $y \in \mathbf{Y} = [a, b]$ and hence $y_i = \arg \min_{y \in \mathbf{Y}} p(\mathbf{x}_i, y)$, which means $\hat{f}(\mathbf{x}_i) = y_i$. Therefore, if $\gamma(\mathbf{x}_i, y_i) = 0$ for all $i = 1, \dots, N$ we get an exact interpolation of all data points. This, however, cannot be achieved in general in which case $\gamma(\mathbf{x}_i, y_i) > 0$ for some i ; the polynomial γ therefore acts as a slack variable and is minimized in the objective function. An alternative to using a polynomial slack variable is to assign one slack variable $\gamma_i \in \mathbb{R}_+$ to each of the constraints; here we chose to use the polynomial slack variable in order to make the number of

⁴Semidefinite Programming (SDP) is a (convex) conic optimization problem on the space of positive semidefinite matrices, a generalization of Linear Programming; several efficient software packages exist (e.g. MOSEK [25] or SeDuMi [29]), and for more details, the interested reader is referred to e.g. [2] and references therein.

decision variables of (3.3) independent of the number of samples N , which facilitates the analysis of the generalization error in Section 3.3.

We also note that if the objective function cannot be evaluated in closed form (e.g., if the moments of the Lebesgue measure over $\mathbf{X} \times [a, b]$ are not known or too costly to compute), the integral can be replaced by an approximation computed from the available data

$$(3.4) \quad \frac{1}{N} \sum_{i=1}^N \gamma(\mathbf{x}_i, y_i).$$

The optimization problem (3.3) translates to a semidefinite program (SDP) by equivalently reformulating the first constraint using Theorem 3.1, as is done in the Moment-SOS hierarchy of convex relaxations for polynomial optimization. For more details the interested reader is referred to [18, 19].

For brevity, we state it explicitly only for d_y even (the difference for d_y odd is the same as in Theorem 3.1):

$$(3.5) \quad \begin{aligned} \min_{\gamma \in \mathbb{R}[\mathbf{x}, y]_{d_\gamma}, (h_k \in \mathbb{R}[\mathbf{x}]_{d_x})_{k=1}^{d_y}, (\sigma_{0,i}, \sigma_{1,i})_{i=1}^N} & \int_{\mathbf{X} \times [a, b]} \gamma(\mathbf{x}, y) \, d\mathbf{x}dy \\ p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2 &= \sigma_{0,i} + \sigma_{1,i}(b - y)(y - a), \quad i = 1, \dots, N \\ \sigma_{0,i} \in \Sigma_{d_y}[y], \sigma_{1,i} \in \Sigma_{d_y-2}[y] & \\ \gamma(\mathbf{x}_i, y_i) \geq 0, \quad i = 1, \dots, N. & \end{aligned}$$

Formulation (3.5) readily translates to an SDP and can be solved using off-the-shelf solvers such as MOSEK [25] or SeDuMi [29].

Remark 3.2 (Simplified complexity analysis). *In SOS problem (3.5), if we neglect the number of free variables parametrizing h_k and γ (polynomials of degrees typically much smaller than N), we have $2N$ positive semidefinite matrices of size at most $d_y/2$ and satisfying Nd_y equality constraints. In this simplified setup, according to [7, Section 6.6.3], the number of Newton steps of an interior-point algorithm for finding an ε -solution of SOS problem (3.5) is of the order of $\log(1/\varepsilon)\sqrt{Nd_y}$, and each Newton step has a complexity of the order of $N^3d_y^4 + N^3d_y^3 + N^2d_y^4$.*

Remark 3.3 (Range of f). *If no information on the range of f is available, the first constraint of (3.5) can be replaced by*

$$p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2 = \sigma_{0,i} \quad i = 1, \dots, N,$$

with $\sigma_{0,i} \in \Sigma_{d_y}$. This is equivalent to $p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2$ being nonnegative on \mathbb{R} for each $i = 1, \dots, N$.

We note that, up to rescaling of p and γ , the problems (3.3) and (3.5) are invariant with respect to the choice of the parameter $\alpha > 0$. We decided to include this parameter as a simple way to control the scaling of the coefficients of p and γ in the numerical implementation.

Remark 3.4. *For piecewise constant and piecewise polynomial functions described in §2.3 and §2.4, the polynomials p providing their argmin representation constructed in Theorem 2.1 and Corollary 2.2 are optimal in (3.3) after rescaling (which does not change the argmin). This holds as long as the samples \mathbf{x}_i lie outside of the points of discontinuity of f (which is of zero Lebesgue measure). In other words, our blind data-driven approach is able to recover exactly optimal solutions for a large class of functions for which a Gibbs phenomenon would occur with more classical approaches. To prove the optimality of p , it suffices to observe that by construction $(\partial^2 p / \partial y^2)(\mathbf{x}_i, y_i) > 0$ if all \mathbf{x}_i 's lie outside the points of discontinuity of f . Since there are finitely many data points, we can*

rescale p such that $(\partial^2 p / \partial y^2)(\mathbf{x}_i, y_i) > 2\alpha$ for all $i \in 1, \dots, N$. Since $(\partial p / \partial y)(\mathbf{x}_i, y_i) = 0$, we get by Taylor's theorem $p(\mathbf{x}_i, y) - p(\mathbf{x}_i, y_i) \geq \alpha(y - y_i)^2$. This estimate holds globally since y_i is the unique global minimizer of $p(\mathbf{x}_i, \cdot)$ and $p(\mathbf{x}_i, \cdot)$ is coercive. Therefore p satisfies the constraints of (3.3) with $\gamma = 0$ and is therefore optimal since γ is nonnegative.

Remark 3.5. We remark that with $\alpha > 0$ the argmin in (3.1) is unique on a given data point x_i provided that $\gamma(x_i, y_i) = 0$ (i.e., there is a zero error on the data point). Outside of the data set, we cannot guarantee the uniqueness of the argmin in which case a tiebreaker rule must be applied (e.g., taking the min of the argmin). Importantly, the results below are independent of whether the argmin is unique or not since they apply to the entire set of minimizers.

The following result bounds the error on data points of any feasible solution to (3.3) and (3.5).

Lemma 3.6 (Error on data points). *Let p , γ and α be feasible in (3.3) or (3.5) and let*

$$z_i \in \arg \min_{y \in [a, b]} p(\mathbf{x}_i, y).$$

Then we have

$$|z_i - y_i| \leq \sqrt{\frac{\gamma(\mathbf{x}_i, y_i)}{\alpha}}$$

and therefore

$$|\hat{f}(\mathbf{x}_i) - f(\mathbf{x}_i)| \leq \sqrt{\frac{\gamma(\mathbf{x}_i, y_i)}{\alpha}}$$

for $i = 1, \dots, N$.

Proof. Let (\mathbf{x}_i, y_i) be fixed. Observe that for any $z_i \in \arg \min_{y \in [a, b]} p(\mathbf{x}_i, y)$, we have $p(\mathbf{x}_i, z_i) - p(\mathbf{x}_i, y_i) \leq 0$. Therefore we have by the first constraint of (3.3) or (3.5)

$$\gamma(\mathbf{x}_i, y_i) - \alpha(z_i - y_i)^2 \geq p(\mathbf{x}_i, z_i) - p(\mathbf{x}_i, y_i) + \gamma(\mathbf{x}_i, y_i) - \alpha(z_i - y_i)^2 \geq 0.$$

Therefore

$$|z_i - y_i| \leq \sqrt{\frac{\gamma(\mathbf{x}_i, y_i)}{\alpha}}.$$

The last statement of the lemma follows by the facts that $\hat{f}(\mathbf{x}_i) \in \arg \min_{y \in [a, b]} p(\mathbf{x}_i, y)$ with p optimal in (3.3) and (3.5) and $y_i = f(\mathbf{x}_i)$. \square

3.3. Generalization error. In this section we study the generalization error of the argmin estimator in a probabilistic setting. We assume that the samples \mathbf{x}_i , $i = 1, \dots, N$, are independent identically distributed, drawn from a probability distribution \mathbb{P} on \mathbf{X} that is unknown to us. We study the generalization error using the tools of scenario optimization, which allows for analysis with minimal underlying assumptions on f . The generalization bounds obtained have no explicit dependence on the dimension of the ambient space n and on regularity of f . They depend only the number of decision variables in (3.3), which, however, may depend implicitly on n .

We first observe that Problem 3.3 can be equivalently rewritten in the form

$$(3.6) \quad \begin{aligned} & \min_{\theta \in \mathbb{R}^{n_\theta}} c^\top \theta \\ & \text{s.t.} \quad \inf_{y \in [a, b]} \{p_\theta(\mathbf{x}_i, y) - p_\theta(\mathbf{x}_i, y_i) + \gamma_\theta(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2\} \geq 0, \quad i = 1, \dots, N \\ & \quad \gamma(\mathbf{x}_i, y_i) \geq 0, \quad i = 1, \dots, N, \end{aligned}$$

where $\theta \in \mathbb{R}^{n_\theta}$ gathers all the decision variables of (3.3), i.e., the coefficients of $(h_k)_{k=1}^{d_y}$ and γ , and $c \in \mathbb{R}^{n_\theta}$ is a constant vector such that $c^\top \theta = \int \gamma d\mathbf{x}dy$.

Problem (3.6) can be rewritten as

$$(3.7) \quad \begin{aligned} \min_{\theta \in \mathbb{R}^{n_\theta}} \quad & \theta^\top c \\ \text{s.t.} \quad & \theta \in \bigcap_{i=1}^N \Theta_i, \end{aligned}$$

where the set Θ_i is defined by

$$\Theta_i := \left\{ \theta \mid \inf_{y \in [a,b]} \{p_\theta(\mathbf{x}_i, y) - p_\theta(\mathbf{x}_i, y_i) + \gamma_\theta(\mathbf{x}_i, y_i) - \alpha(y - y_i)^2 \geq 0\}, \gamma_\theta(\mathbf{x}_i, y_i) \geq 0 \right\}.$$

We also define

$$\Theta_{\mathbf{x}} := \left\{ \theta \mid \inf_{y \in [a,b]} \{p_\theta(\mathbf{x}, y) - p_\theta(\mathbf{x}, f(\mathbf{x})) + \gamma_\theta(\mathbf{x}, f(\mathbf{x})) - \alpha(y - f(\mathbf{x}))^2 \geq 0\}, \gamma_\theta(\mathbf{x}, f(\mathbf{x})) \geq 0 \right\}.$$

We observe that (3.7) is the so-called scenario counterpart of the robust optimization problem

$$(3.8) \quad \begin{aligned} \min_{\theta \in \mathbb{R}^{n_\theta}} \quad & \theta^\top c \\ \text{s.t.} \quad & \theta \in \bigcap_{\mathbf{x} \in \mathbf{X}} \Theta_{\mathbf{x}}. \end{aligned}$$

In other words, the feasible set in (3.7) is a sub-sampled version of the feasible set of (3.8), where only N constraints, drawn independently, are enforced. Crucially, we remark that both Θ_i and $\Theta_{\mathbf{x}}$ are convex and so are the feasible sets of (3.7) and (3.8). With these notations, we can state the following theorem:

Theorem 3.7. *Let n_θ denote the number of decision variables in (3.7). Suppose that $\epsilon \in (0, 1)$, $\delta \in (0, 1)$ and $N \in \mathbb{N}$ are chosen such that*

$$(3.9) \quad \sum_{k=0}^{n_\theta-1} \binom{N}{k} \epsilon^k (1-\epsilon)^{N-k} \leq \delta$$

or

$$(3.10) \quad N \geq \frac{1}{\epsilon} \left(n_\theta - 1 + \ln \frac{1}{\delta} + \sqrt{2(n_\theta - 1) \ln \frac{1}{\delta}} \right)$$

hold ((3.10) is a sufficient condition for (3.9)). Let (p, γ) be an optimal solution to (3.5) with N iid samples from a probability distribution \mathbb{P} on \mathbf{X} and denote

$$\gamma_{\max} := \sup_{(\mathbf{x}, y) \in \mathbf{X} \times [a,b]} \gamma(\mathbf{x}, y).$$

Then with probability at least $1 - \delta$ (taken over the sample $\mathbf{x}_1, \dots, \mathbf{x}_N$ with joint distribution $\underbrace{\mathbb{P} \otimes \dots \otimes \mathbb{P}}_{N \text{ times}}$, we have

$$(3.11) \quad \mathbb{P} \left(\{ \mathbf{x} \in \mathbf{X} : |\hat{f}(\mathbf{x}) - f(\mathbf{x})| \leq \sqrt{\frac{\gamma_{\max}}{\alpha}} \} \right) > 1 - \epsilon,$$

where $\hat{f}(\mathbf{x})$ is any measurable selection satisfying $\hat{f}(\mathbf{x}) \in \text{Argmin}_{y \in \mathbf{Y}} p(\mathbf{x}, y)$.

Remark 3.8. *We remark that if the points \mathbf{x}_i are sampled uniformly in \mathbf{X} , the probability in (3.11) is nothing but the normalized Lebesgue measure on \mathbf{X} .*

Remark 3.9 (Parameter choice). *Theorem 3.7 can be used as a guiding tool for selecting the parameters d_x, d_y . Specifically, these can be increased incrementally until a desired accuracy measured by γ_{\max} is obtained. This is especially appealing when the degree of γ is zero (i.e., γ is a real number) in which case the value of γ_{\max} is readily available. Otherwise, this value can be upper-bounded*

using the moment-sum-of-squares hierarchy [18] or other methods providing rigorous bounds on the range of a polynomial.

Proof of Theorem 3.7. Any optimal solution (p, γ) of (3.5) induces an optimal solution θ to (3.7) and vice-versa. Given such an optimal solution, [8, Theorem 1] asserts that if (3.9) holds, then with probability at least $1 - \delta$

$$\mathbb{P}(\{\mathbf{x} \mid \theta \notin \Theta_{\mathbf{x}}\}) \leq \epsilon$$

or equivalently

$$\mathbb{P}(\{\mathbf{x} \mid \theta \in \Theta_{\mathbf{x}}\}) > 1 - \epsilon.$$

Using the definition of $\Theta_{\mathbf{x}}$, this implies that

$$\mathbb{P}(A) > 1 - \epsilon,$$

where

$$A := \{\mathbf{x} \in \mathbf{X} \mid \inf_{y \in [a,b]} \{p(\mathbf{x}, y) - p(\mathbf{x}, f(\mathbf{x})) + \gamma(\mathbf{x}, f(\mathbf{x})) - \alpha(y - f(\mathbf{x}))^2\} \geq 0\}.$$

Given any $\mathbf{x} \in A$ and taking $\hat{f}(\mathbf{x}) \in \arg \min_{y \in [a,b]} p(\mathbf{x}, y)$, we have

$$\gamma(\mathbf{x}, f(\mathbf{x})) - \alpha(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \geq p(\mathbf{x}, \hat{f}(\mathbf{x})) - p(\mathbf{x}, f(\mathbf{x})) + \gamma(\mathbf{x}, f(\mathbf{x})) - \alpha(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \geq 0,$$

where we used the fact that $p(\mathbf{x}, \hat{f}(\mathbf{x})) - p(\mathbf{x}, f(\mathbf{x})) \leq 0$, $\hat{f}(\mathbf{x}) \in [a, b]$ and that $\mathbf{x} \in A$. It follows that

$$(\hat{f}(\mathbf{x}) - f(\mathbf{x}))^2 \leq \frac{\gamma(\mathbf{x}, f(\mathbf{x}))}{\alpha} \leq \frac{\gamma_{\max}}{\alpha}$$

for all $\mathbf{x} \in A$. Taking square roots and recalling that $\mathbb{P}(A) > 1 - \epsilon$, we obtain the result. The condition (3.10) is a sufficient condition for (3.9) derived in [1, Corollary 1]. \square

Remark 3.10. *When the integral in the objective function 3.3 is replacerced by its empirical average (3.4) computed from the same data set that is used to enforce the constraints, it is currently unknown whether the generalization bound of Theorem 3.7 remains valid [9]. A simple remedy is to use an independent sample for the objective and for the constraints, for example by splitting the data set in two.*

3.4. Beyond polynomials. In this section we briefly discuss how the proposed method extends to approximants of the form

$$\hat{f}(\mathbf{x}) = \arg \min_{y \in \mathbf{Y}} p(\mathbf{x}, y),$$

where p is not necessarily a polynomial. The key observation to make is that when parametrizing p as

$$(3.12) \quad p(\mathbf{x}, y) = \sum_{k=1}^{d_y} h_k(\mathbf{x}) y^k,$$

the functions h_k appear in (3.3) and (3.5) only via their evaluations at the data points \mathbf{x}_i . Therefore, parametrizing each h_k as $h_k(\mathbf{x}) = \sum_{i=1}^{n_k} c_{k,i} \beta_{k,i}(\mathbf{x})$, where $\beta_{k,i}$ are possibly non-polynomial basis functions, the optimization problem (3.5) remains a semidefinite programming problem with the decision variables $c_{i,k} \in \mathbb{R}$ and the coefficients of γ . The function γ can be parametrized by non-polynomial basis functions in (\mathbf{x}, y) since only evaluations of γ at the samples (\mathbf{x}_i, y_i) appear in (3.5).

If a non-polynomial parametrization of p in y was sought, one would have to resort to certificates of nonnegativity for the given function classes akin to Theorem 3.1. Currently, this is well understood for trigonometric polynomials [13] but we envision broader function classes may be considered, given the univariate nature of the nonnegativity certificate required in (3.5) which is significantly less challenging than its multivariate counterpart.

4. NUMERICAL EXAMPLES

In this section we present several examples demonstrating the effectiveness of the polynomial argmin data structure for regression of functions possessing discontinuities. All examples were solved on a MacBook Air 1.2 GHz Quad-Core Intel Core i7 with 16GB RAM, MOSEK SDP solver [25]. The problems were modeled using Yalmip [24]. The range of all functions was normalized to $\mathbf{Y} = [-1, 1]$ and the parameter α was taken to be 0.01 in all examples. The parameters that vary in the examples are d_x and d_y in the parametrization of p in (3.2) (i.e., d_x and d_y are the degrees of p in \mathbf{x} and y respectively). Matlab prototype codes reproducing the numerical experiments can be downloaded from <https://homepages.laas.fr/henrion/software/polyargmin>

4.1. Univariate: Approximation of discontinuous functions. We start by showing the effectiveness of the method on four discontinuous functions depicted in Figure 2 alongside their polynomial argmin approximations. The functions are

$$f_1 = \begin{cases} -1, & x \in [-0.75, 0.75] \\ 1, & x \in [-1, -0.75) \cup (0.75, 1], \end{cases} \quad f_2 = \begin{cases} -1, & x \in [-0.75, -0.25] \cup [0.25, 0.75] \\ 1, & x \in [-1, -0.75) \cup (-0.25, 0.25) \cup (0.75, 1]. \end{cases}$$

$$f_3 = x^2 f_1, \quad f_4 = \sin(2x) f_1.$$

In each case we used 200 data points sampled uniformly at random in $[-1, 1]$ and solved (3.5). We observe a remarkably precise recovery of the discontinuous functions. In (3.2), the minimal degrees d_x and d_y required to obtain an approximation of this accuracy are reported in the figure. For comparison, in Figure 3 we depict the performance on the same task with a neural network with five hidden layers with 20 neurons per layer with the hyperbolic tangent activation functions, trained using Matlab's neural network toolbox with gradient descent; these parameters were selected by manual hyperparameter tuning.

4.2. Univariate: Parameter dependence. Here we investigate the dependence of the approximation quality on d_x which is the degree of the polynomials h_k parametrizing the polynomial p in (3.2). We do so on the function from Eq. (66) in [22] that possesses 7 discontinuities. For data, we use two hundred equidistantly spaced samples in the interval $[-1, 1]$. The results are depicted in Figure 4. As expected the approximation quality improves as the degree of h_k increases, obtaining a very precise accuracy for $\deg h_k = 7$. For comparison, Figure 5 depicts also the results with a neural network approximation.

4.3. Univariate: Challenging continuous functions. For completeness we briefly report results for approximation of continuous functions. We do so on two functions. The first one is $f(x) = \sqrt{|\sin x|}$ which is a transcendental function with Hölder exponent 1/2 whose derivative grows unbounded near the origin. The second one is the Runge function $f(x) = (1 + 25x^2)^{-1}$ which is a smooth function that exhibits the Runge phenomenon (oscillations near the boundary) when approximated by polynomials through interpolation. The results are depicted in Figure 6. As in the previous examples, we observe an accurate fit and no oscillations with low degrees of p in x and y .

4.4. Bivariate: Approximation of discontinuous functions.

Consider the discontinuous function

$$f(\mathbf{x}) = \begin{cases} \frac{1+x_1+x_2}{2} & \text{if } x_1^2 + x_2^2 \leq \frac{1}{4} \\ 0 & \text{otherwise} \end{cases}$$

constructed by multiplying an affine function with the indicator function⁵ of a bivariate disk. On Figure 7 we represent the `chebfun2` approximation obtained with the `chebfun` package [11]:

⁵The indicator function of a set is equal to one on the set and zero outside.

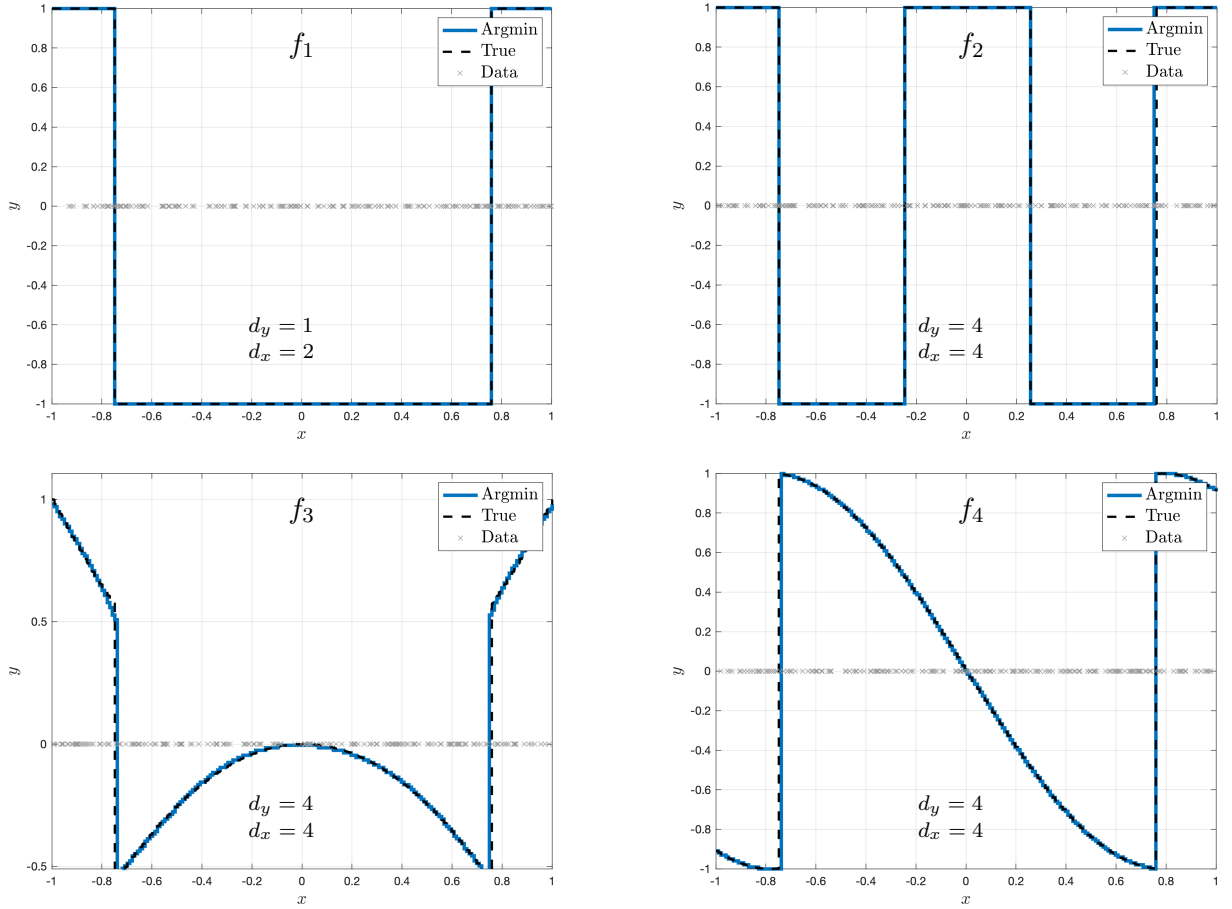


FIGURE 2. Polynomial argmin approximations of different functions with discontinuities.

```
[x1,x2]=meshgrid(linspace(-1,1,100));
plot(chebfun2(double(x1.^2+x2.^2<=1/4).*(x1+x2+1)/2));
```

We observe that the approximation is corrupted by the typical Gibbs phenomenon encountered when approximating a discontinuous function with polynomials [30, Chapter 9], namely large oscillations near the discontinuity set.

5. DISCUSSION AND CONCLUSION

We have presented a simple method based on the argmin of a polynomial for approximation of discontinuous functions. The approach is model-free and mesh-free in the sense that it does not require prior knowledge about the function being approximated as it works only with samples of its values. It is grounded in powerful tools from *univariate* sum of squares optimization, hence based only on a very specific class of convex semidefinite programming, and so it is simple to use. It shows a great promise in numerical examples and we believe that it can become a valuable tool in data analysis. We have also proved that *exact* recovery is possible on certain examples of discontinuous functions and have provided theoretical analysis of in-sample and out-of-sample error in a probabilistic setting. In the argmin approach [21] based on the Christoffel-Darboux polynomial, such an exact recovery is not possible in general as an ε -regularization term is introduced to guarantee that the associated moment matrix is non-singular. In addition, the size of the moment matrix to invert strongly depends on the dimension of data while in our optimization-based approach, the size and

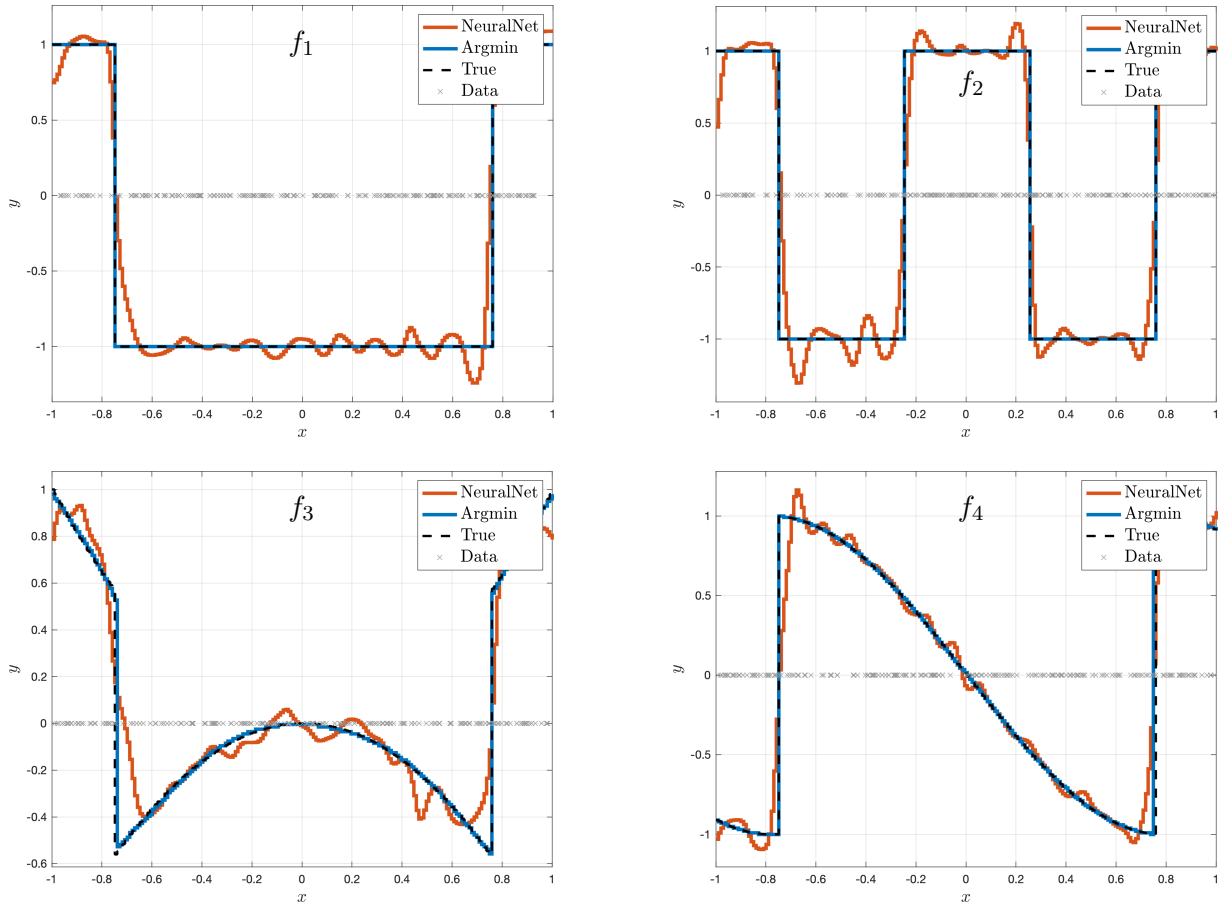


FIGURE 3. Polynomial argmin approximation versus neural network.

the number of resulting matrices to be positive semidefinite does *not* depend on the dimension of the data (their number is linear in the sample size).

While we have used general purpose semidefinite solvers to construct our argmin approximants, more efficient approaches can be envisioned. Indeed our formulation boils down to optimization over the cone of univariate non-negative polynomials, a very specific class of semidefinite optimization problems. For example, non-symmetric solvers may perform faster on these problems [26]. Another option could be to bypass numerical optimization and use tailored numerical linear algebra as in [23].

An additional interesting feature of the approximant is that its evaluation at a given point $\mathbf{x} \in \mathbf{X}$ reduces to finding the global minimum of a univariate polynomial on an interval, which can be done efficiently e.g. by matrix eigenvalue computation. A numerically stable algorithm is described in [6, Section 7] and implemented in the `roots` function of the `chebfun` package [11]. It is based on the application of the QR algorithm for finding the eigenvalues of a balanced companion matrix constructed by evaluating the polynomial at Chebyshev points.

This paper is a first step that introduces the argmin approximant and illustrates its promising potential on non trivial numerical examples. We hope that it could inspire some further developments. In particular, we have left open the question of optimal rates of convergence of the argmin approximant or more generally its worst-case performance when considering pre-defined classes of functions to approximate, e.g. in terms of the *manifold width* discussed in [10], which is a generalization of the classical Kolmogorov width. Based on Section 2.1, it is clear that the rates are at

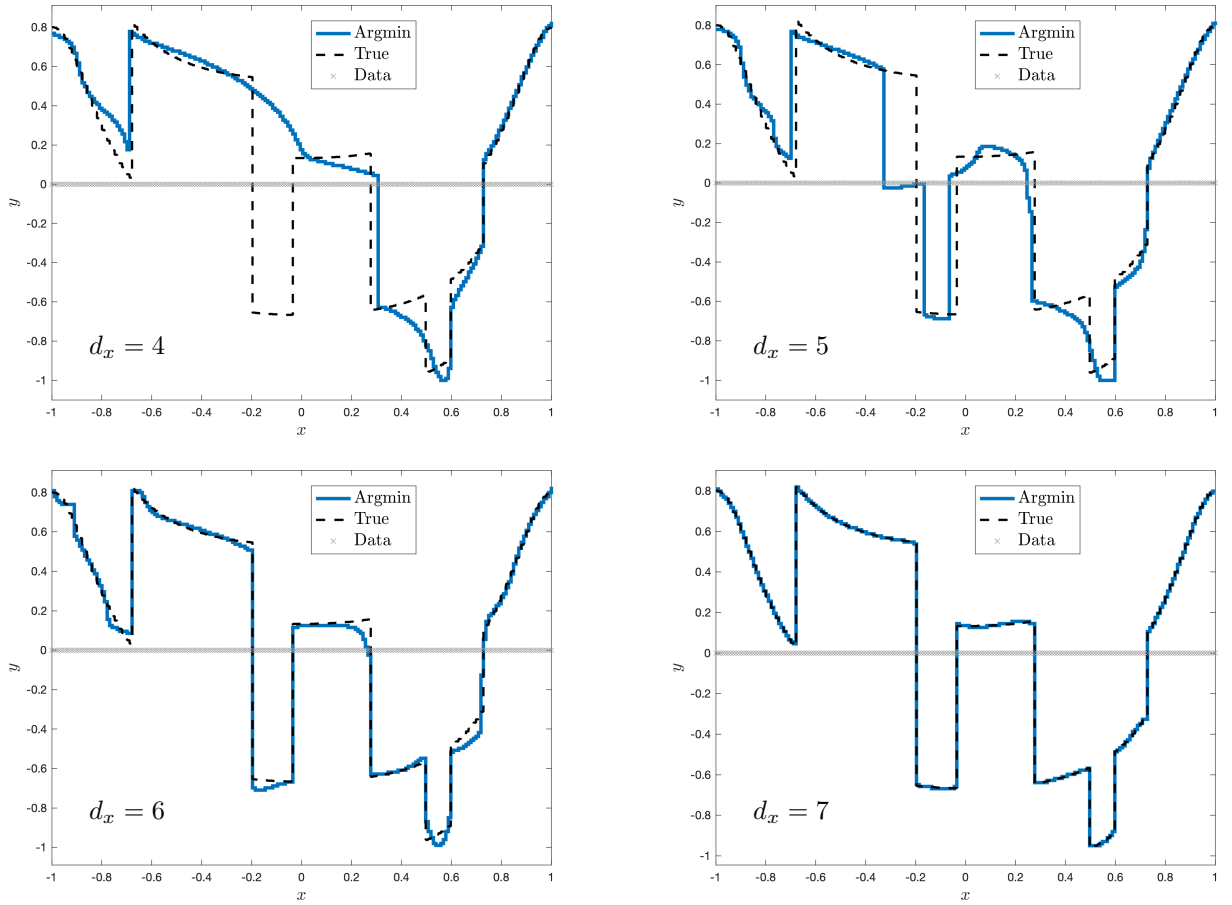


FIGURE 4. Polynomial argmin approximation on a function with 7 discontinuities with $d_y = 6$ for different values of d_x .

least as good as those of polynomial approximation whenever the degree of p in y is at least two. However, we conjecture that the rates are better for discontinuous functions.

As a final remark, the main goal of the paper is to introduce a new tool for function approximation with remarkable properties in the traditional noiseless setting when exact data is available. Of course, to validate its potential and efficiency in the more general setting of statistical learning where data can be corrupted by noise (in the \mathbf{x} and/or the value $f(\mathbf{x})$), a further detailed analysis is needed but beyond the scope of the present paper. We believe that relations to the max-margin support vector machine [33, 5] could facilitate this analysis.

6. ACKNOWLEDGEMENT

The authors would like to thank Francis Bach for pointing out the links to max-margin support vector machines. The authors would also like to acknowledge the help of Open AI's model o4-mini-high in proving Theorem 2.1.

REFERENCES

- [1] T. Alamo, R. Tempo, A. Luque, D. R. Ramirez. Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. *Automatica* 52:160-172, 2015.
- [2] M. Anjos, J.B. Lasserre. (Editors). *Handbook on Semidefinite, Conic, and Polynomial Optimization*. Springer, New York, 2012.

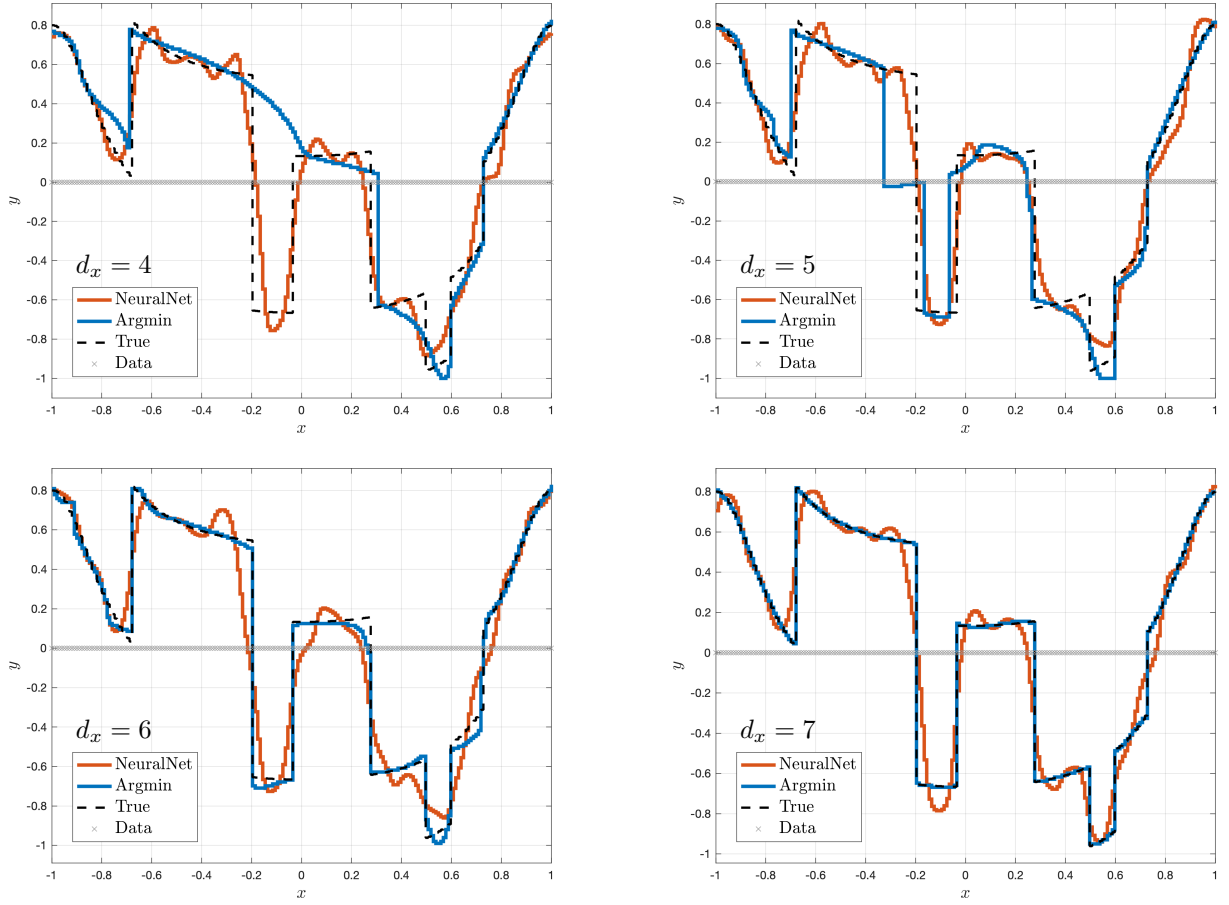


FIGURE 5. Polynomial argmin approximation of a function with 7 discontinuities for different values of the degree of h_k , $k = 1, \dots, 6$ in comparison with a neural network.

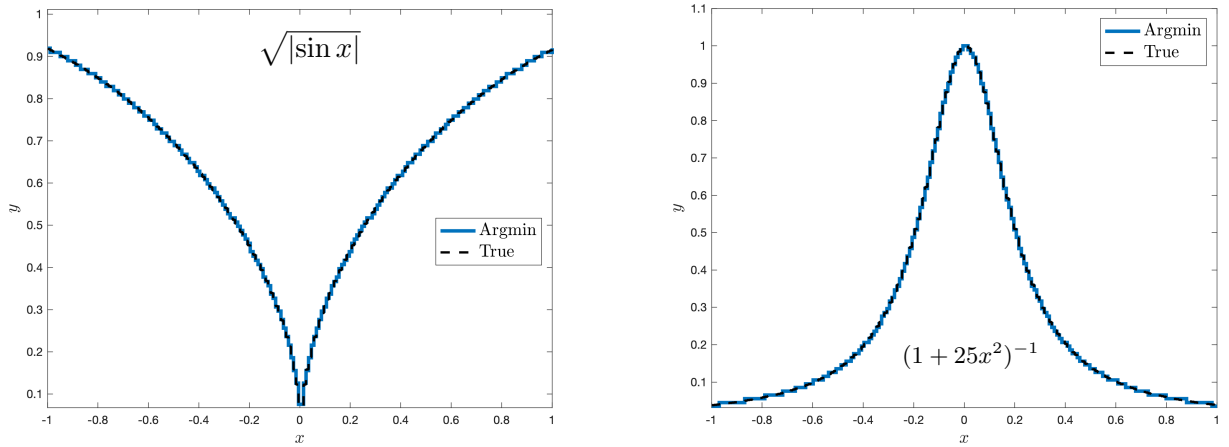


FIGURE 6. Polynomial argmin approximations of two continuous functions with $d_x = 4$, $d_y = 4$. Left: transcendental function with unbounded derivatives near the origin. Right: Runge function.

[3] V. Antun, N. M. Gottschling, A. C. Hansen, B. Adcock. Deep learning in scientific computing: understanding the instability mystery. SIAM News 54:2, 2021.

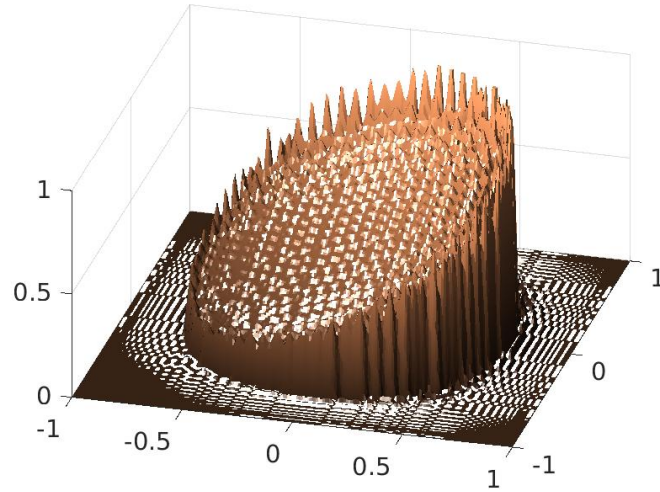


FIGURE 7. Chebyshev polynomial approximation of a discontinuous bivariate function obtained by `chebfun2`.

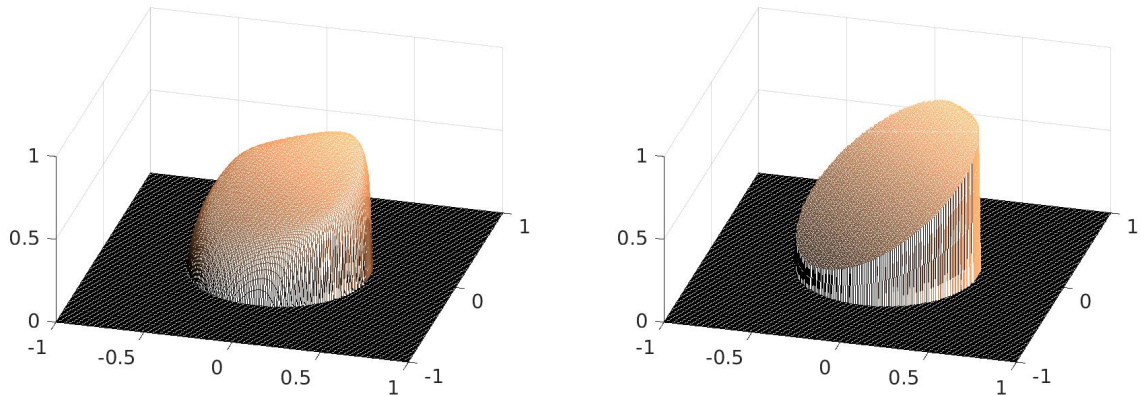


FIGURE 8. Polynomial argmin approximation of a discontinuous function for $d_x = 3, d_y = 2$ (left) and $d_x = 3, d_y = 4$ (right).

- [4] V. Antun, M. J. Colbrook, A. C. Hansen. Proving existence is not enough: mathematical paradoxes unravel the limits of neural networks in artificial intelligence. *SIAM News* 55:4, 2022.
- [5] F. Bach. *Learning Theory from First Principles*. The MIT Press, 2023. https://www.di.ens.fr/~fbach/ltfp_book.pdf
- [6] Z. Battles, L. N. Trefethen. An extension of Matlab to continuous functions and operators. *SIAM J. Sci. Comp.* 25(5):1743–1770, 2004.
- [7] A. Ben-Tal, A. Nemirovski. *Lectures on modern convex optimization*. MPS-SIAM Series on Optimization, SIAM, 2001.
- [8] M. C. Campi, S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization* 19(3):1211–1230, 2008.
- [9] M. Campi. Personal communication.
- [10] A. Cohen, R. DeVore, G. Petrova, P. Wojtaszczyk. Optimal Stable Nonlinear Approximation. *Foundations of Computational Mathematics*, 22:607–648, 2022.
- [11] T. A. Driscoll, N. Hale, L. N. Trefethen (editors). *Chebfun Guide*. Pafnuty Publications, 2014.
- [12] S. De Marchi, F. Marchetti, E. Perracchione. Jumping with variably scaled discontinuous kernels (VSDKs). *BIT Numerical Mathematics* 60:441–463, 2020
- [13] B. Dumitrescu. *Positive trigonometric polynomials and signal processing applications*. Springer, 2007.

- [14] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, J. Tompson. Implicit behavioral cloning. Proc. 5th Conf. on Robot Learning, PMLR 164:158-168, 2022.
- [15] D. Henrion, J. B. Lasserre. Graph recovery from incomplete moment information. *Constructive Approximation* 56:165-187, 2022.
- [16] B. Llanas, S. Lantarón, F. J. Sáinz. Constructive approximation of discontinuous functions by neural networks. *Neural Processing Letters* 27:209-226, 2008.
- [17] M. Bozzini, M. Rossini. The detection and recovery of discontinuity curves from scattered data. *J. Computational and Applied Mathematics* 240:148-162, 2013.
- [18] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optimization* 11(3):796-817, 2001.
- [19] J.B. Lasserre. The Moment SoS Hierarchy: Applications and Related Topics, *Acta Numerica* 33, pp. 841–908, 2024.
- [20] S. Marx, T. Weisser, D. Henrion, J. B. Lasserre. A moment approach for entropy solutions to nonlinear hyperbolic PDEs. *Mathematical Control and Related Fields*, 10(1):13-140, 2020.
- [21] S. Marx, E. Pauwels, T. Weisser, D. Henrion, J. B. Lasserre. Semi-algebraic approximation using Christoffel-Darboux kernel. *Constructive Approximation* 54:391–429, 2021.
- [22] K. S. Eckhoff. Accurate and efficient reconstruction of discontinuous functions from truncated series expansions. *Math. Comput.* 61(204):745-763, 1993.
- [23] S.-I. Filip. A robust and scalable implementation of the Parks-McClellan algorithm for designing FIR filters. *ACM Trans. Math. Software* 43(1), 7:1-24, 2016.
- [24] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. *IEEE Symp. Computer-Aided Control Design (CACSD)*, Taiwan, 2004.
- [25] MOSEK ApS. MOSEK Optimization Toolbox User’s Manual. Version 10.0, <https://www.mosek.com/>, 2024.
- [26] D. Papp, S. Yildiz. Sum-of-squares optimization without semidefinite programming. *SIAM J. Optim.* 29(1):822-851, 2019.
- [27] V. Powers, B. Reznick. Polynomials that are positive on an interval. *Trans. Amer. Math. Soc.* 352(10):4677-4692, 2000.
- [28] C.-W. Shu. High order weighted essentially non-oscillatory schemes for convection dominated problems. *SIAM Review*, 51(1), 82–126, 2009
- [29] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods and Softwares.* 11-12: 625–653, 1999.
- [30] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2013.
- [31] A. Weisse, G. Wellein, A. Alvermann, H. Fehske. The kernel polynomial method. *Reviews of Modern Physics* 78(1):275-306, 2006.
- [32] E. Tadmor. Filters, mollifiers and the computation of the Gibbs phenomenon. *Acta Numerica* 16:305-378, 2007.
- [33] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Y. Singer. Large margin methods for structured and interdependent output variables. *J. Machine Learning Research* 6(9):1453-1484, 2005.