



HAL
open science

Computation of sum of squares polynomials from data points

Bruno Després, Maxime Herda

► **To cite this version:**

Bruno Després, Maxime Herda. Computation of sum of squares polynomials from data points. 2019.
hal-01946539v3

HAL Id: hal-01946539

<https://hal.science/hal-01946539v3>

Preprint submitted on 19 Jul 2019 (v3), last revised 15 Mar 2020 (v5)

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.

COMPUTATION OF SUM OF SQUARES POLYNOMIALS FROM DATA POINTS

BRUNO DESPRÉS* AND MAXIME HERDA†

Abstract. We propose an iterative algorithm for the numerical computation of sums of squares of polynomials approximating given data at prescribed interpolation points. The method is based on the definition of a convex functional G arising from the dualization of a quadratic regression over the Cholesky factors of the sum of squares decomposition. In order to justify the construction, the domain of G , the boundary of the domain and the behavior at infinity are analyzed in details. When the data interpolate a positive univariate polynomial, we show that in the context of the Lukacs sum of squares representation, G is coercive and strictly convex which yields a unique critical point and a corresponding decomposition in sum of squares. For multivariate polynomials which admit a decomposition in sum of squares and up to a small perturbation of size ε , G^ε is always coercive and so its minimum yields an approximate decomposition in sum of squares. Various unconstrained descent algorithms are proposed to minimize G . Numerical examples are provided, for univariate and bivariate polynomials.

Key words. Positive polynomials, sum of squares, convex analysis, positive interpolation, iterative methods.

AMS subject classifications. 90C30, 65K05, 90C25

1. Introduction. The numerical and algorithmic motivation of the present paper comes from a recent work [4] where an iterative algorithm for the computation of positive polynomials was proposed. In the previous algorithm, the polynomials were univariate and the interpolation points, where the data of the polynomials are given, were sliding points (it allowed for strong convergence properties). However the structure of the algorithms was such that one needs to preprocess the data to be able to use the sliding points, which brings restrictions for applications in Scientific Computing (SC). In the present work, we relax this condition by constructing a new iterative algorithm which computes a sum of squares (SOS) decomposition from the sole knowledge of prescribed interpolation data at prescribed interpolation points. Also the method is much more general so it is formulated for multivariate polynomials as well and does not need tensorization, something that was impossible with the previous method.

A modern reference in SC for control of the sign of polynomials at a *finite number of prescribed interpolation points* is in the works of C.-W. Shu [22], with application to the discretization of hyperbolic equations with high order methods. The point of view developed in this article is to control the sign of polynomials on *all points in a given compact (semi-algebraic) set* $\mathbb{K} \subset \mathbb{R}^d$ which is much more demanding. Preliminary tests for the construction of such algorithms are in [5], but the methods were inefficient in terms of the time of restitution. In a fully different direction, one must mention the theory of numerical approximation with splines, see [14, 1]: splines are widely used in scientific computing and computer aided design (CAD) but often needs tensorization in multi-dimension; this limitation is not encountered by our new methods because they can be implemented on any semi-algebraic set \mathbb{K} in any dimension.

In the community of numerical optimization [12] from which we borrow most of our notations, SOS algorithms based on SemiDefinite Programming (SDP) are extensively used. It had been noticed by Powers and Wörmann [20] that finding an SOS decomposition is equivalent to SDP, that is optimization in the cone of non-negative quadratic forms. Then algorithms based on interior-point methods were developed to solve these problems [19, 18, 23]. However, these methods seem to be hardly directly applicable in SC because they are based more on algebraic properties and not on interpolation data which are of major importance in numerical analysis and SC. This leads us to the development of the algorithm of the present paper, which is not based on SDP but rather on the resolution of a non-convex

*Sorbonne-Université, CNRS, Université de Paris, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France, and Institut Universitaire de France

†Inria, Univ. Lille, CNRS, UMR 8524 Laboratoire Paul Painlevé, F-59000 Lille, France

quadratic problem over Cholesky factors of the SOS decomposition. We solve the quadratic program using a dualization of the problem, which leads us to a nonlinear convex program. Let us mention that a similar reformulation of general SDP was proposed by Burer and Monteiro [3]. In our case however, we use the particular structure of the interpolation data of the SOS to obtain some useful coercivity properties on the dual function. Also, similar dualization ideas can be found in [16, 8], but unlike here they are formulated on the Gram matrix rather than on the Cholesky factors. Our construction will generate a functional with strong convexity properties for which standard descent algorithms are efficient, as shown in the numerical section.

Let $P[\mathbf{X}] := P[X_1, \dots, X_d]$ be the set of real polynomials with d variables. The subset of polynomials of total degree less than or equal to $n \geq 1$ is denoted by $P^n[\mathbf{X}]$. Let $\mathbb{K} \subset \mathbb{R}$ be a closed semi-algebraic set defined through a finite number j_* of polynomial inequalities

$$(1.1) \quad \mathbb{K} = \{ \mathbf{x} \in \mathbb{R}^d \text{ such that } g_j(\mathbf{x}) \geq 0 \text{ for } g_j \in P[\mathbf{X}], 1 \leq j \leq j_* \}.$$

Most standard cells (intervals in 1D, squares and triangles in 2D, ...) in SC can be implemented as semi-algebraic sets, so it is not a restriction for further applications. The convex set of non-negative polynomials of maximal degree n on \mathbb{K} is

$$(1.2) \quad P_{\mathbb{K},+}^n[\mathbf{X}] = \{ p \in P^n[\mathbf{X}] \text{ such that } p(\mathbf{x}) \geq 0 \text{ for any } \mathbf{x} \in K \}, \quad r_* = \dim P^n[\mathbf{X}].$$

Famous examples of characterizations as SOS are the Lukacs theorem [24] or Putinar's Positivstellensatz [21]: a recent state of the art can be found in the books of Lasserre [12, 13]; some recent algorithmic issues in the context of optimal control can be found in [10] and therein. In order to be constructive, we focus in this work on the following version

$$(1.3) \quad p = \sum_{j=1}^{j_*} g_j \left(\sum_{i=1}^{i_*} q_{ij}^2 \right) = \sum_{i=1}^{i_*} \left(\sum_{j=1}^{j_*} g_j q_{ij}^2 \right) = \sum_{j=1}^{j_*} \sum_{i=1}^{i_*} g_j q_{ij}^2.$$

where the maximal number of squares is equal to a predefined value $i_* \geq 1$ independent of j . Next, the notion of unisolvence which comes from the Finite Element Method (FEM) is convenient to formalize properties of interpolation points. A unisolvent set of points $(\mathbf{x}_r)_{1 \leq r \leq r_*}$ is such that any polynomial $p \in P^n[\mathbf{X}]$ is uniquely determined by its values $p(\mathbf{x}_r) = y_r$ for $1 \leq r \leq r_*$. The number i_* of polynomials in the SOS (1.3) is *a priori* independent from the number of interpolation points. However in our context the function G below is more naturally constructed assuming that

$$(1.4) \quad i_* = r_*.$$

That is why we will assume (1.4) throughout this work, except at early stages of the construction. With these notations, one formulates the notion of positive interpolation: it is a recent adaptation [4] to SC of the notion of a *certificate of positivity* for which the reader can find information in [12, 13]. A practical way to understand the model problem below is the following: from the knowledge of the values of p at only a *finite number of given interpolation points*, get a control of the sign of p at *infinite number of points* (the whole set \mathbb{K}).

PROBLEM 1.1 (Iterative positive interpolation on \mathbb{K}). *Let $p \in P_{\mathbb{K},+}^n[\mathbf{X}]$. Take a unisolvent set $(\mathbf{x}_r)_{1 \leq r \leq r_*}$, and consider the interpolated values $y_r = p(\mathbf{x}_r)$. From $(\mathbf{x}_r, y_r)_{1 \leq r \leq r_*}$, compute iteratively polynomials $(q_{ij})_{ij}$ such that the SOS representation (1.3) holds at the limit.*

The methods and results studied in this work can be summarized as follows. Consider the parametrization $q_{ij} \in P^{n_j}[\mathbf{X}]$ with $n_j = \lfloor (n - \deg(g_j))/2 \rfloor$ where $\lfloor \cdot \rfloor$ denotes the integer part of a real number. Consider the canonical basis made of monomials (but other basis can be taken as well, see

Remark 2.1), with the standard multi-index notation $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d$, $|\alpha| = \alpha_1 + \dots + \alpha_d$ and $\mathbf{X}^\alpha = X_1^{\alpha_1} \dots X_d^{\alpha_d}$. The polynomials q_{ij} write $q_{ij}(\mathbf{X}) = \sum_{|\alpha| \leq n_j} c_\alpha^{ij} \mathbf{X}^\alpha$ and we store the coefficients in a vector of coefficients $c^{ij} = (c_\alpha^{ij})_\alpha \in \mathbb{R}^{r_j}$ where $r_j = \dim(\mathbb{P}^{n_j}[\mathbf{X}]) = \binom{d+n_j}{d}$. Gather the coefficients $c^{i1}, c^{i2}, \dots, c^{ij*}$ in a single column vector (called a Cholesky factor) $\mathbf{U}_i = (c^{i1}, c^{i2}, \dots, c^{ij*})^t \in \mathbb{R}^{r_*}$ where $r_* = \sum_{j=1}^{j_*} r_j$ (this equality can be relaxed, see Remark 2.2). Define the polynomial valued block matrix $B(\mathbf{X}) = B(\mathbf{X})^t \in \mathbb{R}^{r_* \times r_*}$

$$(1.5) \quad B(\mathbf{X}) = \text{diag}(g_1(\mathbf{X})D^{n_1}(\mathbf{X}), \dots, g_{j_*}(\mathbf{X})D^{n_{j_*}}(\mathbf{X})), \quad D_{\alpha, \beta}^{n_j}(\mathbf{X}) = \mathbf{X}^\alpha \mathbf{X}^\beta, \quad |\alpha|, |\beta| \leq n_j.$$

The first diagonal block is square $r_1 \times r_1$, ... until the last block which is square $r_{j_*} \times r_{j_*}$: all other terms are zero. By construction, one has the identity

$$(1.6) \quad \sum_{j=1}^{j_*} g_j(\mathbf{X}) \sum_{i=1}^{i_*} q_{ij}^2(\mathbf{X}) = \sum_{i=1}^{i_*} \left(\sum_{j=1}^{j_*} g_j(\mathbf{X}) q_{ij}^2(\mathbf{X}) \right) = \sum_{i=1}^{i_*} \langle B(\mathbf{X}) \mathbf{U}_i, \mathbf{U}_i \rangle.$$

Denote the evaluation of $B(\mathbf{X})$ at interpolation points as $B_r = B(\mathbf{x}_r) \in \mathbb{R}^{r_* \times r_*}$. Define the function $G: \mathbb{R}^{r_*} \rightarrow \overline{\mathbb{R}}$ with domain $\mathcal{D} = \{\lambda \in \mathbb{R}^{r_*} \text{ such that } \sum_{r=1}^{r_*} \lambda_r B_r > 0\}$ as follows. For $\lambda \in \mathcal{D}$ then

$$(1.7) \quad G(\lambda) = \text{tr} \left[\left(I + \sum_{r=1}^{r_*} \lambda_r B_r \right)^{-1} \right] + \sum_{r=1}^{r_*} y_r \lambda_r,$$

otherwise $G(\lambda) = +\infty$. In the previous formula, $\text{tr}(\cdot)$ denotes the trace. Then our main results are the following.

THEOREM 1.2. *The function G has the following properties:*

1. *It is a proper closed convex function. It is C^∞ on its non-empty open convex domain and by definition equal to $+\infty$ on the complement of \mathcal{D} .*
2. *Each $\lambda \in \mathcal{D}$ defines computable polynomials $(q_{ij}[\lambda])_{1 \leq i \leq r_*, 1 \leq j \leq j_*}$ such that*

$$(1.8) \quad \frac{\partial G}{\partial \lambda_r}(\lambda) = y_r - \sum_{j=1}^{j_*} g_j(\mathbf{x}_r) \sum_{i=1}^{i_*} q_{ij}^2[\lambda](\mathbf{x}_r), \quad 1 \leq r \leq r_*.$$

If $\lambda_ \in \mathcal{D}$ is a critical point of G , that is $\nabla G(\lambda_*) = 0$, then the family $(q_{ij}[\lambda_*])_{ij}$ is solution to (1.3), that is a SOS.*

THEOREM 1.3 (Existence of critical points in \mathcal{D}). *It is proved in two cases.*

1. *Take $d > 1$, \mathbb{K} a semi-algebraic set, $p \geq 0$ on \mathbb{K} . Assume that a technical condition on the linear independence of the matrices B_r is satisfied. Then, up to an infinitesimally small perturbation (the perturbed polynomial p^ε has the interpolation data $(y_r^\varepsilon)_{1 \leq r \leq r_*}$), the function G^ε is strictly convex, coercive and admits a unique critical point in \mathcal{D} .*
2. *Take $d = 1$, \mathbb{K} a segment and $p > 0$ on \mathbb{K} . Then the technical condition the linear independence of the matrices B_r is satisfied. Moreover G is strictly convex, coercive and admits a unique critical point in \mathcal{D} .*

COROLLARY 1.4 (Solution to Problem 1.1). *Under the hypothesis of Theorem 1.3, the minimum of G (or G^ε) in \mathcal{D} yields a SOS decomposition of p (or p^ε). It can be computed by standard unconstrained descent algorithms.*

A practical scenario, which in our mind has interest for SC, is the following. Take a polynomial without knowing its sign on \mathbb{K} . If the descent method converges and recover p at the limit, then p is

non negative on \mathbb{K} . If the descent does not recover p at the limit, then for monovariate polynomials, p is non positive on \mathbb{K} . It shows that the descent method provides an iterative *certificate of positivity*. In case of non convergence, the iterations provide nevertheless a non negative surrogate to p . We refer to [4] for an illustration of the interest of non negative polynomial surrogates.

The outline of this paper is as follows. In Section 2, we propose a dual interpretation of Problem 1.1. This leads us to the introduction of the function G . Then, in Section 3, we discuss necessary and sufficient conditions characterizing asymptotic properties and strict convexity of G . In Section 4, we show that for univariate positive polynomials on a segment, the former conditions are satisfied yielding strict convexity and coercivity of the associated function G . Besides, we provide a more precise description the structure of the domain. In Section 5, we present the specific descent and Newton type methods we use to compute the critical points of G . In Section 6 we provide numerical illustrations of the efficiency of our new approach for computing SOS decomposition of polynomials in one variable on segment and two variables on triangle. Finally, we provide in Appendix A some additional theoretical results concerning the links between the asymptotic cone of the set \mathcal{D} and the Lagrange polynomials in the case of univariate polynomials.

Acknowledgements. Both authors are greatly indebted to Jean-Bernard Lasserre and Didier Henrion for their kind explanations on the theory and state of the art of semidefinite programming and sum of squares and would like to thank them for their invitation at LAAS and for their hospitality.

2. Construction of G (Proof of Theorem 1.2). The construction of G , leading to (1.7), is done by recasting the model problem 1.1 as the convex dual of a Quadratically Constrained Quadratic Program (QCQP) (see[2]). In order to have a more general discussion, we relax the condition (1.4) in this part and in the next Section 2.1, that is

$$i_* \neq r_*$$

is possible. The condition (1.4) is reintroduced end of Section 2.1. Before going too far into the details of the heuristic, we begin with some remarks on the objects introduced in the first section.

Remark 2.1. In the numerical experiments of Section 6, we use other bases than the monomials in order to optimize the robustness and accuracy of the algorithms. It only changes the definition of the matrix $B(\mathbf{X})$ in (1.5) and thus of $B_r = B(\mathbf{x}_r)$ but every result of this paper still hold. More generally, one could even generalize Problem 1.1 and replace the constraint of equality at interpolated values by constraints of the type $y_r = L_r(p)$ where the family $\{L_r : P^n[\mathbf{X}] \rightarrow \mathbb{R}, r = 1, \dots, r_*\}$ is any base of the dual space of $P^n[\mathbf{X}]$. In the context of SC, and more precisely in the numerical resolution of partial differential equations, one deals with data points in finite difference discretizations. However, if one is considering a finite volume discretization, one would rather work with mean values on some mesh cells. This variant is easily manageable with our method by choosing the adequate linear forms L_r and modifying the matrices B_r accordingly.

Remark 2.2. Consider the formula (1.3). The dimension of the polynomial space for p is r_* . The number of degrees of freedom involved in the right hand side is $i_* r_{**}$ with $r_{**} = \sum_{j=1}^{j_*} r_j$. Since the problem is to construct SOS which correspond to a given polynomial p , a reasonable assumption is to find the solution in a space with more degrees of freedom than the number of constraints. It means $i_* r_{**} \geq r_*$. So the minimal hypothesis is the equality $r_{**} = r_*$, which will be considered as true throughout this work. Let us however precise that in abstract results of existence of SOS decomposition, such as Putinar Positivstellensatz [21], the (non-optimal) degrees of the q_{ij} may be higher than in the present setting leading to $r_{**} \gg r_*$ with our notation. In practice it is harmless to increase the degree of the r_j and thus r_{**} in our algorithm but it leads to heavier notations with little gain in terms of generality in the analysis.

Remark 2.3. An interesting consequence of the Caratheodory Theorem ([9, Theorem III.1.3.6 page 98]) is that if a formula like (1.3) holds for $i_* > r_*$, then a similar one holds also for $i_* = r_*$ (but for different polynomials q_{ij}). Indeed the set $\mathcal{W} = \sum_{j=1}^{i_*} g_j(\mathbf{X})\mathbb{P}^{n_j}[\mathbf{X}]^2$ is a closed convex cone embedded in $p \in \mathbb{P}^n[\mathbf{X}]$. Therefore any convex combination of $i_* > r_*$ elements of \mathcal{W} can be expressed as a convex combination of only $r_* = \dim \mathbb{P}^n[\mathbf{X}]$ elements of \mathcal{W} (the coefficients of the convex combination can be set to 1 after proper rescaling of the new q_{ij}).

2.1. Lagrangian duality (Theorem 1.2 item 1). In any dimension, the notation $\langle \cdot, \cdot \rangle$ will denote the Euclidean dot product and $\| \cdot \|$ will denote the associated norm. We define the algebraic manifold

$$(2.1) \quad \mathcal{U} = \{ \mathbf{U} = (\mathbf{U}_1, \dots, \mathbf{U}_{i_*}) \in (\mathbb{R}^{r_*})^{i_*} \text{ such that } \sum_{i=1}^{i_*} \langle B_r \mathbf{U}_i, \mathbf{U}_i \rangle = y_r \text{ for all } 1 \leq r \leq r_* \}.$$

The vectors \mathbf{U}_i are called the Cholesky factors [18, 23] of the decomposition. With the unisolvence assumption, finding a SOS (1.3) amounts to finding one element $\mathbf{U} \in \mathcal{U}$. In order to find a $\mathbf{U} \in \mathcal{U}$ in a constructive manner, our strategy is to start at a given \mathbf{V} (probably outside \mathcal{U}) and to project on \mathcal{U} in the quadratic norm. It writes as follows.

PROBLEM 2.4. Take $\mathbf{V} = (\mathbf{V}_1, \dots, \mathbf{V}_{i_*}) \in (\mathbb{R}^{r_*})^{i_*}$. Calculate $\mathbf{U} = \underset{\mathbf{U} \in \mathcal{U}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{i_*} \|\mathbf{U}_i - \mathbf{V}_i\|^2$.

The vectors $\mathbf{V} = (\mathbf{V}_i)_i$ may be thought of as a good initial guesses for the $\mathbf{U} = (\mathbf{U}_i)_i$. The optimal value of the cost does not matter. However Problem 2.4 seems even harder to solve than the original problem we were concerned with. The finding is that the Lagrangian dual problem is endowed with good properties provided \mathbf{V} is conveniently chosen. In this case, the new Problem 2.4 provides a way to determine an admissible $\mathbf{U} \in \mathcal{U}$.

Still for any \mathbf{V} , introduce the Lagrangian which is the sum of the functional and of the dualization of the constraint (2.1) with a Lagrange multiplier $\lambda \in \mathbb{R}^{r_*}$

$$\mathcal{L}(\mathbf{U}, \lambda) = \frac{1}{2} \sum_{i=1}^{i_*} \left(\|\mathbf{U}_i - \mathbf{V}_i\|^2 + \sum_{r=1}^{r_*} \lambda_r \langle B_r \mathbf{U}_i, \mathbf{U}_i \rangle \right) - \frac{1}{2} \langle \lambda, \mathbf{y} \rangle \quad \text{where } \mathbf{y} = (y_r)_{1 \leq r \leq r_*}.$$

Both the objective and the constraints are quadratic, so the optimality constraints are linear. Define the symmetric matrix $M(\lambda) = M(\lambda)^t \in \mathbb{R}^{r_* \times r_*}$

$$(2.2) \quad M(\lambda) = I + \sum_{r=1}^{r_*} \lambda_r B_r$$

where I is the identity matrix in $\mathbb{R}^{r_* \times r_*}$. The first order optimality condition writes

$$(2.3) \quad M(\lambda) \mathbf{U}_i = \mathbf{V}_i \text{ for } 1 \leq i \leq i_* \iff M(\lambda) \mathbf{U} = \mathbf{V}.$$

If the multiplier $\lambda \in \mathbb{R}^{r_*}$ is such that the matrix $M(\lambda)$ is invertible, then the candidate solution \mathbf{U} can be computed explicitly in terms of λ and \mathbf{V} as the solution of the linear system (2.3).

It is therefore natural to concentrate on a condition on λ such that $M(\lambda)$ is invertible. In order to obtain convexity properties in the following we even restrict λ to the set of positive definiteness of $M(\lambda)$. To our knowledge, this at this stage that our analysis differs from the standard exposition of dual QCQP [2, 9] and from other dualizations in the context of SOS [3, 16, 8].

DEFINITION 2.5. The domain of positive definiteness of M is $\mathcal{D} = \{ \lambda \in \mathbb{R}^{r_*} \mid M(\lambda) > 0 \} \subset \mathbb{R}^{r_*}$. It is an open set and it is non empty since $0 \in \mathcal{D}$.

For a Lagrange multiplier $\lambda \in \mathcal{D}$, the inverse transformation of (2.3) is $\mathbf{U}(\lambda) = M(\lambda)^{-1}\mathbf{V}$. Then, one can evaluate the Lagrangian at $\mathbf{U}(\lambda)$. An elementary computation yields $\mathcal{L}(\mathbf{U}(\lambda), \lambda) = \frac{1}{2} \sum_{i=1}^{i_*} (\|\mathbf{V}_i\|^2 - \langle \mathbf{V}_i, M(\lambda)^{-1}\mathbf{V}_i \rangle) - \frac{1}{2} \langle \lambda, \mathbf{y} \rangle$. This motivates the introduction of the dual objective function $G_{\mathbf{V}} : \mathcal{D} \rightarrow \mathbb{R}$ defined by

$$(2.4) \quad G_{\mathbf{V}}(\lambda) = \sum_{i=1}^{i_*} \langle \mathbf{V}_i, M(\lambda)^{-1}\mathbf{V}_i \rangle + \langle \lambda, \mathbf{y} \rangle,$$

and which one should think of as a function to be minimized.

LEMMA 2.6. *The function $G_{\mathbf{V}}$ is smooth on \mathcal{D} . The first and second derivatives are*

$$(2.5) \quad \frac{\partial G_{\mathbf{V}}}{\partial \lambda_r}(\lambda) = y_r - \sum_{i=1}^{i_*} \langle \mathbf{U}_i(\lambda), B_r \mathbf{U}_i(\lambda) \rangle \quad \text{and} \quad \frac{\partial^2 G_{\mathbf{V}}}{\partial \lambda_r \partial \lambda_s}(\lambda) = 2 \sum_{i=1}^{i_*} \langle B_r \mathbf{U}_i(\lambda), M(\lambda)^{-1} B_s \mathbf{U}_i(\lambda) \rangle.$$

In particular $G_{\mathbf{V}}$ is convex on its domain \mathcal{D} .

Proof. The proof stems from the identity $\partial_{\lambda_r} M(\lambda)^{-1} = -M(\lambda)^{-1} B_r M(\lambda)^{-1}$ and the symmetry of the various matrices involved. Convexity follows from the positivity of $M(\lambda)$ and the expression of second derivatives yielding $\langle \nabla^2 G_{\mathbf{V}}(\lambda) \mu, \mu \rangle = 2 \sum_{i=1}^{i_*} \langle A_i(\mu, \lambda), M(\lambda)^{-1} A_i(\mu, \lambda) \rangle \geq 0$ where $A_i(\mu, \lambda) = \sum_{r=1}^{r_*} \mu_r B_r \mathbf{U}_i(\lambda)$ for $1 \leq i \leq i_*$. \square

In order to address the behavior of $G_{\mathbf{V}}$ near the boundary, we will make use of the following notion.

DEFINITION 2.7. *A convex function $f : \mathbb{R}^{r_*} \mapsto \mathbb{R} \cup \{+\infty\}$ is said to be closed over its domain $\mathcal{D}_f = \{\mathbf{x} \mid f(\mathbf{x}) < \infty\}$ if and only if the level sets $\{\mathbf{x} \mid f(\mathbf{x}) \leq t\}$ are closed for $t < +\infty$: see [9] or [2, Appendix A.3.3].*

This property is extremely important in our approach because it yields a strong control of the objective function at finite distance.

LEMMA 2.8. *Assume the equality of dimensions (1.4) and that $\mathbf{V} \in \mathbb{R}^{r_* \times r_*}$ is an orthogonal matrix. Then one has the simpler expression where $\text{tr}(\cdot)$ denotes the trace of a square matrix*

$$(2.6) \quad G(\lambda) := G_{\mathbf{V}}(\lambda) = \text{tr}(M^{-1}(\lambda)) + \langle \lambda, \mathbf{y} \rangle.$$

Moreover the extension of $G := G_{\mathbf{V}}$ with value $+\infty$ outside of \mathcal{D} is a closed convex function

Proof. The formula is a direct consequence of (2.4), because the number i_* of orthogonal vectors \mathbf{V}_i is equal to the dimension r_* of the space. Thanks to the continuity on \mathcal{D} , the closedness of $G_{\mathbf{V}}$ on \mathbb{R}^{r_*} amounts to showing that for any sequence $(\mu_k)_k$ in \mathcal{D} converging to a point of the boundary of the domain $\partial \mathcal{D} = \{\lambda \in \overline{\mathcal{D}} \mid \det(I + \sum_{r=1}^{r_*} \lambda_r B_r) = 0\}$, then one has $G_{\mathbf{V}}(\mu_k) \rightarrow +\infty$ as $k \rightarrow +\infty$. In the light of the representation formula (2.6) involving the trace of $M(\lambda)^{-1}$ it is the case since the minimal eigenvalue of $M(\mu_k)$ goes to 0 as $k \rightarrow +\infty$. Clearly the function is independent of \mathbf{V} . \square

The equality of dimensions $i_* = r_*$ provides the closed and compact form of G . The notation i_* will not be used anymore in the rest of the presentation, only r_* .

2.2. Critical points of G (Theorem 1.2 item 2). In this section, we formalize natural consequences of the formulas (2.5) for the derivatives of G . These first properties are essentially a reformulation of the previous material. For each Lagrange multiplier $\lambda \in \mathcal{D}$ one defines the vectors $(c_{\alpha}^{ij}[\lambda])_{\alpha, j} \in \mathbb{R}^{r_j}$ which are the components of $\mathbf{U}_i(\lambda)$, the latter being the i th column of $M(\lambda)^{-1}$. It defines the polynomials $q_{ij}[\lambda] \in \mathbb{P}^{n_j}[\mathbf{X}]$ by $q_{ij}[\lambda](\mathbf{X}) = \sum_{|\alpha| \leq n_j} c_{\alpha}^{ij}[\lambda] \mathbf{X}^{\alpha}$. With (1.6), these polynomials

define a sum of square $p[\lambda] \in \mathbb{P}_{\mathbb{K},+}^n[\mathbf{X}]$

$$(2.7) \quad p[\lambda](\mathbf{X}) = \sum_{j=1}^{j_*} g_j(\mathbf{X}) \left(\sum_{i=1}^{r_*} q_{ij}^2[\lambda](\mathbf{X}) \right).$$

Using (1.6), $p[\lambda](\mathbf{x}_r) = \sum_{i=1}^{r_*} \langle B_r \mathbf{U}_i, \mathbf{U}_i \rangle$. So (2.5) is rewritten as

$$(2.8) \quad \frac{\partial G}{\partial \lambda_r}(\lambda) = y_r - p[\lambda](\mathbf{x}_r).$$

The Proposition below characterizes that in order to solve Problem 1.1 it is sufficient to find critical points of G .

PROPOSITION 2.9. *Take $p \in \mathbb{P}_{\mathbb{K},+}^n[\mathbf{X}]$ and an unisolvent set of interpolation points $(\mathbf{x}_r)_{1 \leq r \leq r_*}$ in \mathbb{K} . Consider $y_r = p(x_r)$ for $1 \leq r \leq r_*$. The following properties are equivalents*

- $\lambda^* \in \mathcal{D}$ is a critical point of G , namely $\nabla G(\lambda^*) = 0$.
- $\lambda^* \in \mathcal{D}$ minimizes G .
- $p(\mathbf{X}) = p[\lambda^*](\mathbf{X})$.

Proof. Since G is closed convex, local minima coincide exactly with critical points, so the first two points are equivalent. The equivalence between the first and third assertions follows from (2.8) and the unisolvence assumption. \square

2.3. Number of squares. Let us precise the number of squares in the SOS formula (2.7). This information is additional with respect to Theorem 1.2. It brings the possibility to have a cheaper implementation.

LEMMA 2.10. *The number of non zero polynomials in $\sum_{i=1}^{r_*} q_{ij}^2[\lambda](\mathbf{X})$ is less or equal to r_j .*

Proof. By construction $(\mathbf{U}_1(\lambda), \dots, \mathbf{U}_{r_*}(\lambda)) = \mathbf{U}(\lambda) = M(\lambda)^{-1}$ is a block diagonal matrix. The blocks have size $r_1 \times r_1$ until $r_{j_*} \times r_{j_*}$. So, for a given j , the polynomials $q_{ij}[\mathbf{X}]$ vanish for $1 \leq i \leq r_1 + \dots + r_{j_*-1}$ and for $r_1 + \dots + r_{j_*-1} + r_{j_*} + 1 \leq i \leq r_*$. \square

Remark 2.11. The result of Lemma 2.10 is nevertheless non optimal in dimension $d = 1$. Indeed consider the Lukács Theorem (see Proposition 4.1) in the odd case $n = 2k + 1$ and take $g_1(\mathbf{X}) = \mathbf{X}$ and $g_2(\mathbf{X}) = (1 - \mathbf{X})$ as in (4.3). So $r_* = n + 1$ and $r_1 = r_2 = k + 1$. Assume that there exists a critical point λ_* to G . Then (2.7) yields a representation $p(\mathbf{X}) = \mathbf{X} \sum_{i=1}^k p_{i1}^2[\lambda_*](\mathbf{X}) + (1 - \mathbf{X}) \sum_{i=k+1}^{2k} p_{i2}^2[\lambda_*](\mathbf{X})$. In terms of the number of squares, here $2k$, it is clearly non optimal with respect to the result of the Lukács Theorem which involves only two polynomials whatever n .

3. Coercivity of G (Proof of Theorem 1.3 Item 1). A sufficient condition for the existence of a critical point is that G is infinite at infinity, this is called coercivity,

$$(3.1) \quad \lim_{\|\lambda\| \rightarrow +\infty} G(\lambda) = +\infty.$$

A sufficient condition for the uniqueness of the critical points is strict convexity.

In the following, we start in Section 3.1 by investigating the asymptotic behavior of G along rays starting at 0. From this knowledge we derive conditions characterizing coercivity in Section 3.2. We characterize strict convexity in Section 3.3.

3.1. The asymptotic cone. There are two types of directions in \mathcal{D} . For $\mathbf{d} \in \mathbb{R}^{r_*}$ with $\|\mathbf{d}\| = 1$, one defines the rays $R_{\mathbf{d}} := \{\lambda = t\mathbf{d} \mid t \geq 0\}$ issued from the starting point $0 \in R_{\mathbf{d}}$. Two possibilities occur: either $R_{\mathbf{d}}$ intersects the boundary ∂D either it does not. In the first case if one notes $t_{\mathbf{d}} > 0$ the

unique real number such that $t_{\mathbf{d}}\mathbf{d} \in \partial\mathcal{D}$, then $\lim_{t \rightarrow t_{\mathbf{d}}^-} G(t\mathbf{d}) = +\infty$. So the function G is bounded from below and coercive in the direction \mathbf{d} .

In this section one is interested in the rest of the directions. They generate the so-called asymptotic cone or recession cone of \mathcal{D} . The asymptotic cone is closed, independent of the starting point and is classically defined [9] by $C_\infty = \{\lambda \in \mathbb{R}^{r_*} \text{ such that } \forall \mu \in \mathcal{D}, t \geq 0, \mu + t\lambda \in \mathcal{D}\}$.

LEMMA 3.1. *The asymptotic cone of \mathcal{D} is $C_\infty = \{\lambda \in \mathbb{R}^{r_*} \mid \sum_{r=1}^{r_*} \lambda_r B_r \geq 0\}$.*

Proof. Let λ, μ such that $\sum_{r=1}^{r_*} \lambda_r B_r \geq 0$ and $I + \sum_{r=1}^{r_*} \mu_r B_r > 0$. Then, $I + \sum_{r=1}^{r_*} (\mu_r + t\lambda_r) B_r > 0$ for all $t \geq 0$, so λ belongs to the asymptotic cone. Conversely let λ such that for all μ and $t \geq 0$, $\mu + t\lambda \in \mathcal{D}$. If $\sum_{r=1}^{r_*} \lambda_r B_r$ had a negative eigenvalue then for t large enough $I + t \sum_{r=1}^{r_*} \lambda_r B_r$ would also have a negative eigenvalue which would contradict the fact that $t\lambda \in \mathcal{D}$. \square

The main question is the asymptotic behavior of G in directions in C_∞ .

Let us derive a condition which rules out unboundedness from below in the asymptotic cone. One introduces the polynomial valued vector $L(\mathbf{X})$ with components being the Lagrange polynomials associated with the set of points $(\mathbf{x}_r)_{1 \leq r \leq r_*}$ evaluated at \mathbf{x} , namely

$$(3.2) \quad L(\mathbf{X}) = (l_r(\mathbf{X}))_{1 \leq r \leq r_*} \in \mathbb{R}^{r_*},$$

where the Lagrange interpolation polynomials $l_r \in \mathbb{P}^n[\mathbf{X}]$ are defined by $l_r(\mathbf{x}_s) = \delta_{rs}$ for $1 \leq r, s \leq r_*$, where δ_{rs} denotes the Kronecker symbol. The vector $L(\mathbf{X})$ will be called a Lagrange vector. The polynomial p which takes the value y_r at \mathbf{x}_r satisfies the Lagrange interpolation formula

$$(3.3) \quad p(\mathbf{X}) = \sum_{r=1}^{r_*} y_r l_r(\mathbf{X}) = \langle \mathbf{y}, L(\mathbf{X}) \rangle.$$

One can show another interpolation property characteristics of our problem.

LEMMA 3.2. *One has $B(\mathbf{X}) = \sum_{r=1}^{r_*} l_r(\mathbf{X}) B_r$. For $\mathbf{x} \in \mathbb{K}$, $B(\mathbf{x})$ is positive semidefinite and $L(\mathbf{x}) \in C_\infty$.*

Proof. Let $\mathbf{W}, \mathbf{Z} \in \mathbb{R}^{r_*}$ be the coefficients of some polynomials $(p_j)_{1 \leq j \leq j_*}$ and $(q_j)_{1 \leq j \leq j_*}$. By definition (1.5-1.6) of $B(\mathbf{x})$ which is symmetric one knows that

$$\left\langle \mathbf{W}, \left(B(\mathbf{x}) - \sum_{r=1}^{r_*} l_r(\mathbf{x}) B_r \right) \mathbf{Z} \right\rangle = \sum_{j=1}^{j_*} \left(g_j(\mathbf{x}) p_j(\mathbf{x}) q_j(\mathbf{x}) - \sum_{r=1}^{r_*} l_r(\mathbf{x}) g_j(\mathbf{x}_r) p_j(\mathbf{x}_r) q_j(\mathbf{x}_r) \right) = 0.$$

Since \mathbf{W}, \mathbf{Z} are arbitrary, it yields the first part of the claim. Also for $\mathbf{x} \in \mathbb{K}$, one has that $g_j(\mathbf{x}) \geq 0$. Therefore $\langle \mathbf{W}_1, B(\mathbf{x}) \mathbf{W}_1 \rangle = \sum_{j=1}^{j_*} g_j(\mathbf{x}) p_j(\mathbf{x})^2 \geq 0$ which yields that $B(\mathbf{x}) \geq 0$. One gets that $L(\mathbf{x}) \in C_\infty$. \square

In the following there are three different results concerning the behavior of G in the asymptotic cone: either, Lemma 3.3, $\inf_{t>0, \lambda \in C_\infty} G(t\lambda) = -\infty$; or, Proposition 3.4, $\inf_{t>0, \lambda \in C_\infty} G(t\lambda) > -\infty$; or even better, Proposition 3.8, the function G is coercive.

LEMMA 3.3. *Assume that there exists $\mathbf{z} \in \mathbb{K}$ such that $p(\mathbf{z}) < 0$. Then $\lim_{t \rightarrow +\infty} G(tL(\mathbf{z})) = -\infty$ and thus the corresponding function G is not bounded from below in C_∞ .*

Proof. The half line generated by $L(\mathbf{z})$ is included in \mathcal{D} by Lemma 3.2 and so all for $t \geq 0$, one has $G(tL(\mathbf{z})) = \text{tr}(M(tL(\mathbf{z}))^{-1}) + tp(\mathbf{z})$. Since $\lambda = tL(\mathbf{z}) \in C_\infty$, one has $M(\lambda) \geq I$ so $G(t\lambda) \leq r_* + tp(\mathbf{z}) \xrightarrow[t \rightarrow \infty]{} -\infty$. \square

PROPOSITION 3.4. *Consider $p \in \mathbb{P}_{\mathbb{K},+}^n[\mathbf{X}]$, a unisolvent set of interpolation points $(\mathbf{x}_r)_{1 \leq r \leq r_*}$ in \mathbb{K} and define $y_r = p(\mathbf{x}_r)$ for $1 \leq r \leq r_*$. The following properties are equivalent.*

- For any $\lambda \in C_\infty$, one has $\langle \lambda, \mathbf{y} \rangle \geq 0$.
- There exists polynomials q_{ij} for $1 \leq j \leq j_*$ and $1 \leq i \leq r_* = r_*$ such that

$$p(\mathbf{X}) = \sum_{j=1}^{j_*} g_j(\mathbf{X}) \sum_{i=1}^{r_*} q_{ij}^2(\mathbf{X}).$$

Proof. For $\mathbf{W} \in \mathbb{R}^{r_*}$, define the vector $s_{\mathbf{W}} = (\langle B_r \mathbf{W}, \mathbf{W} \rangle)_{1 \leq r \leq r_*} \in \mathbb{R}^{r_*}$. A equivalent definition of C_∞ is $C_\infty = \{\lambda \in \mathbb{R}^{r_*} \text{ such that } \langle s_{\mathbf{W}}, \lambda \rangle \geq 0 \text{ for all } \mathbf{W} \in \mathbb{R}^{r_*}\}$. In order to prove the result, one can invoke the Generalized Farkas Theorem ([9, Theorem III.4.3.4 page 131] with the correspondence $\mathbf{y} = \mathbf{b}$). It already states that our first assertion is equivalent to \mathbf{y} being in the closed convex conical hull of the linear forms $s_{\mathbf{W}}$, that is $\mathbf{y} = \sum_{i=1}^{r_*} \alpha_i s_{\mathbf{W}_i}$ where $\alpha_i \geq 0$ for all i , and r_* is sufficiently large. It is rewritten as $\mathbf{y} = \sum_{i=1}^{r_*} s_{\mathbf{Z}_i}$ for $\mathbf{Z}_i = (\alpha_i)^{\frac{1}{2}} \mathbf{W}_i$. Using (1.6), the latter rewrites as our second assertion. \square

3.2. Coercivity. Now we investigate the conditions such that G is infinite at infinity (coercivity). A first negative result about coercivity is the following. The proof easily adapted from the one of Lemma 3.3.

LEMMA 3.5. *Assume there exists $\mathbf{z} \in \mathbb{K}$ such that $p(\mathbf{z}) = 0$. Then $G(tL(\mathbf{z}))$ remains bounded as $t \rightarrow +\infty$ and G is not coercive.*

Thus we can only hope for coercivity starting from strictly positive polynomials. Let us now define a specific useful polynomial denoted as p_B .

DEFINITION 3.6. *Define the polynomial $p_B(\mathbf{X}) = \text{tr}(B(\mathbf{X})) \in \mathbb{P}_{\mathbb{K},+}^n[\mathbf{X}]$, where $B(\mathbf{X})$ is the matrix defined in (1.5).*

A key property of this polynomial is the following.

LEMMA 3.7. *Assume that the matrices $\{B_r\}_{1 \leq r \leq r_*}$ are linearly independent. Then there exists a constant $c_* > 0$ such that*

$$(3.4) \quad c_* \|\lambda\| \leq \sum_{r=1}^{r_*} \lambda_r p_B(\mathbf{x}_r), \quad \forall \lambda \in C_\infty.$$

Proof. Let $\lambda \in C_\infty$. The matrix $\sum_r \lambda_r B_r$ is symmetric and positive semidefinite. So its matrix norm can be controlled by its largest eigenvalue and thus by its trace, namely $\|\sum_{r=1}^{r_*} \lambda_r B_r\| \leq \text{tr}(\sum_{r=1}^{r_*} \lambda_r B_r) = \sum_{r=1}^{r_*} \lambda_r p_B(\mathbf{x}_r)$. Second we also know that $\lambda \rightarrow \sum_{r=1}^{r_*} \lambda_r B_r$ is injective thanks to the linear independence assumption. Thus there a constant $c_* > 0$ such that $c_* \|\lambda\| \leq \|\sum_{r=1}^{r_*} \lambda_r B_r\|$. Combining both inequalities ends the proof. \square

The result of the next Proposition holds in any dimension. It shows that any polynomial which has a SOS can be approximated with our approach, provided one shifts positively the SOS by εp_B where $\varepsilon > 0$ is as small as desired. It proves Item 1 in Theorem 1.3.

PROPOSITION 3.8. *Let $p \in \mathbb{P}_{\mathbb{K},+}^n[\mathbf{X}]$ which admits a SOS (1.3). Take a unisolvent set of interpolation points $(\mathbf{x}_r)_{1 \leq r \leq r_*}$ in \mathbb{K} and assume that the corresponding matrices $\{B_r\}_{1 \leq r \leq r_*}$ are linearly independent. Take $\varepsilon > 0$ and set $p^\varepsilon = p + \varepsilon p_B$. Then the function G^ε built from \mathbf{x}_r and $y_r^\varepsilon = p^\varepsilon(\mathbf{x}_r) = y_r + \varepsilon p_B(\mathbf{x}_r)$ for $1 \leq r \leq r_*$ is coercive.*

Proof. The asymptotic cone C_∞ does not depend on \mathbf{y} or \mathbf{y}^ε and we desire to show firstly that G^ε grows linearly to infinity for directions in C_∞ . One has the identity $\sum_{r=1}^{r_*} \lambda_r y_r^\varepsilon = \sum_{r=1}^{r_*} \lambda_r y_r + \varepsilon \sum_{r=1}^{r_*} \lambda_r p_B(\mathbf{x}_r)$. Take $\lambda \in C_\infty$: proposition 3.4 yields $\sum_{r=1}^{r_*} \lambda_r y_r \geq 0$ because p is a SOS by assumption; then Lemma 3.7 shows that for any $\lambda \in C_\infty$ $\sum_{r=1}^{r_*} \lambda_r y_r \geq 0 + \varepsilon c_* \|\lambda\|$ which yields uniform coercivity in the directions in the asymptotic cone.

In order to show coercivity (3.1) which is a stronger statement, the proof is by contradiction. Assume it does not hold. Then there exists a constant $K \in \mathbb{R}$ as well as a sequence $(t_m, \mathbf{d}_m)_{m \in \mathbb{N}}$ such that $t_m \rightarrow +\infty$, $\|\mathbf{d}_m\| = 1$ and $G(t_m \mathbf{d}_m) \leq K$. By convexity, and since $G(0) = r_*$, one has $G(t \mathbf{d}_m) \leq \max(r_*, K)$ for $t \in [0, t_m]$. Up to the extraction of a sub-sequence there exists \mathbf{d}_* with $\|\mathbf{d}_*\| = 1$, such that $G(t \mathbf{d}_*) \leq \max(r_*, K)$ for $t \in \mathbb{R}^+$. In particular the ray with direction \mathbf{d}_* cannot intersect the boundary $\partial \mathcal{D}$ so it belongs to the asymptotic cone C_∞ . By the first estimate $G(t \mathbf{d}_*) \geq \varepsilon c_* t$, so it cannot be bounded which yields the contradiction. \square

Remark 3.9. With the same strategy of proof, it is possible to show that any polynomial in a neighborhood of p_B , that is $\{q \in \mathbb{P}^n[\mathbf{X}] \mid \|q - p_B\| < \varepsilon\}$, generates a function G which is coercive. It yields that all polynomials in this neighborhood admits a representation as a SOS.

3.3. Strict convexity. Strict convexity, if it holds, yields uniqueness of a critical point. This information is additional to Item 1 of Theorem 1.3.

PROPOSITION 3.10. *Let $p \in \mathbb{P}_{\mathbb{K},+}^n[\mathbf{X}]$ be strictly positive on \mathbb{K} . Take a unisolvent set of interpolation points $(\mathbf{x}_r)_{1 \leq r \leq r_*}$ in \mathbb{K} and assume that the corresponding matrices $\{B_r\}_{1 \leq r \leq r_*}$ are linearly independent. Then G is strictly convex.*

Proof. From (2.5) one has that $\langle \nabla^2 G(\lambda) \mu, \mu \rangle = 2 \sum_{i=1}^{r_*} \langle A_i(\mu, \lambda), M(\lambda)^{-1} A_i(\mu, \lambda) \rangle \geq 0$ for all $\mu \in \mathbb{R}^{r_*}$, where $A_i(\mu, \lambda) = (\sum_{r=1}^{r_*} \mu_r B_r) \mathbf{U}_i(\lambda)$ for $1 \leq i \leq r_*$. Since $M(\lambda)^{-1}$ is positive definite, its columns $\mathbf{U}_i(\lambda)$ form a basis.

By contradiction, assume now G is not strictly convex. There exists $\mu \neq 0$ such that $\langle \nabla^2 G(\lambda) \mu, \mu \rangle = 0$. So the vectors $A_i(\mu, \lambda)$ vanish for all i . So $\sum_{r=1}^{r_*} \mu_r B_r = 0$, and $\mu = 0$ by linear independence of the matrices $(B_r)_{r=1, \dots, r_*}$. This is a contradiction so $\nabla^2 G(\lambda) > 0$ and G is strictly convex. \square

The strict convexity of G can be measured with the minimal eigenvalue of its Hessian $\alpha(\lambda) = \inf_{\mu \neq 0} \frac{\langle \nabla^2 G(\lambda) \mu, \mu \rangle}{\|\mu\|^2} > 0$, for any $\lambda \in \mathcal{D}$. An important property which motivates the design of one of our numerical methods is the following.

LEMMA 3.11. *Under the assumptions of Proposition 3.10, then α has a cubic degeneracy at infinity in the interior of the asymptotic cone of \mathcal{D} . For all $\mathbf{d} \in \mathbb{R}^{r_*}$ such that $\|\mathbf{d}\| = 1$ and $\sum_{r=1}^{r_*} d_r B_r > 0$, there is $C_{\mathbf{d}} > 0$ such that $\alpha(t \mathbf{d}) \leq C_{\mathbf{d}}(1+t)^{-3}$ for all $t \geq 0$.*

Proof. Let $\lambda = t \mathbf{d}$. Then for a constant C depending only on the data one has $\langle \nabla^2 G(\lambda) \mu, \mu \rangle \leq C \|M(\lambda)^{-1}\|^3 \|\mu\|^2$. Note that under the assumptions the minimal eigenvalue of $M(\lambda)$ is given by $1 + e_{\mathbf{d}} t$ with $e_{\mathbf{d}}$ the minimal eigenvalue of $\sum_{r=1}^{r_*} d_r B_r$. Hence $\|M(\lambda)^{-1}\| = O((1+t)^{-1})$.

4. Univariate polynomials on a segment (Theorem 1.3 Item 2). In this section, we focus on univariate polynomials, namely when $d = 1$, over the segment $\mathbb{K} = [0, 1]$. This case is interesting because it is central for numerical computation of functions of one variable and also one can easily prove the coercivity and the strict convexity. The notation is simplified by using the real variable $x \in \mathbb{R}$, more adapted to analytical methods.

We check that the various assumptions granting coercivity and strict convexity are satisfied. In view of Proposition 3.4, Proposition 3.8 and Proposition 3.10 of the previous section, it suffices to exhibit an appropriate choice of functions $(g_j)_j$ and of interpolation points such that: any non-negative polynomial admits a (possibly non-explicit) SOS decomposition; and the matrices $\{B_r\}_r$ are linearly independent. The first point follows from the *Markov-Lukács* Theorem, see [24, 5, 6, 11] for a proof.

PROPOSITION 4.1 (Markov-Lukács). *Let us consider $p \in \mathbb{P}^n[x]$ and $\mathbb{K} = [0, 1]$.*

- **Even case:** *If $n = 2k$, then p is non-negative on \mathbb{K} if and only if there are polynomials a and b with degree less or equal to k and $k - 1$ respectively such that*

$$(4.1) \quad p(x) = a^2(x) + x(1-x)b^2(x).$$

- **Odd case:** If $n = 2k + 1$, then p is non-negative on \mathbb{K} if and only if there are polynomials a and b with degree less or equal to k such that

$$(4.2) \quad p(x) = xa^2(x) + (1-x)b^2(x).$$

Now let us precise the setting. One takes $j_* = 2$ and

$$(4.3) \quad \begin{cases} \text{for } n \text{ is even : } & g_1(x) = 1 \quad \text{and} \quad g_2(x) = x(1-x), \\ \text{for } n \text{ is odd : } & g_1(x) = x \quad \text{and} \quad g_2(x) = 1-x. \end{cases}$$

Concerning the interpolation points, we choose any $r_* = n + 1$ distinct points $(x_r)_{r=1, \dots, n+1}$ on the segment $[0, 1]$. The polynomials are represented along monomials so that the matrices B_r have the block structure

$$(4.4) \quad B_r = \begin{pmatrix} g_1(x_r) \mathbf{w}_1^r \otimes \mathbf{w}_1^r & 0 \\ 0 & g_2(x_r) \mathbf{w}_2^r \otimes \mathbf{w}_2^r \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}$$

where

$$\begin{cases} \text{for } n = 2k : & \mathbf{w}_1^r = (1, x_r, \dots, x_r^k)^t \quad \text{and} \quad \mathbf{w}_2^r = (1, x_r, \dots, x_r^{k-1})^t, \\ \text{for } n = 2k + 1 : & \mathbf{w}_1^r = \mathbf{w}_2^r = (1, x_r, \dots, x_r^k)^t. \end{cases}$$

With these notations, the equalities (4.1) and (4.2) are equivalent to $y_r = \langle B_r \mathbf{U}, \mathbf{U} \rangle$ for $1 \leq r \leq n + 1$. In the odd case $n = 2k + 1$ one has $\mathbf{U} = (a_0, \dots, a_k, b_0, \dots, b_k)^t \in \mathbb{R}^{n+1}$ with $a(x) = \sum_{l=0}^k a_l x^l$ and $b(x) = \sum_{l=0}^k b_l x^l$. In the even case $n = 2k$, $\mathbf{U} = (a_0, \dots, a_k, b_0, \dots, b_{k-1})^t \in \mathbb{R}^{n+1}$.

COROLLARY 4.2 (of Proposition 3.4). *Take $p \in P_{[0,1],+}^n$ and set $y_r = p(x_r)$. Then, for all $\lambda \in C_\infty$, one has that $\langle \lambda, \mathbf{y} \rangle \geq 0$.*

Proof. Indeed the second statement of Proposition 3.4 holds with $i_* = 1$ by taking $p_{11} = a$ and $p_{12} = b$ with a, b provided by Proposition 4.1. \square

Let $\lambda \in \mathbb{R}^{n+1}$. Using the structure (4.4) of the matrices B_r , one has the Hankel matrices

$$(4.5) \quad \sum_{r=1}^{n+1} \lambda_r B_r = \begin{pmatrix} H_1 & 0 \\ 0 & H_2 \end{pmatrix}$$

where

$$\begin{cases} \text{for } n = 2k : & \langle H_1 \mathbf{v}, \mathbf{w} \rangle = \sum_{i,j=0}^k s_{i+j+1} v_i w_j, \quad \langle H_2 \mathbf{v}, \mathbf{w} \rangle = \sum_{i,j=0}^k (s_{i+j} - s_{i+j+1}) v_i w_j, \\ \text{for } n = 2k + 1 : & \langle H_1 \mathbf{v}, \mathbf{w} \rangle = \sum_{i,j=0}^k s_{i+j+1} v_i w_j, \quad \langle H_2 \mathbf{v}, \mathbf{w} \rangle = \sum_{i,j=0}^{k-1} (s_{i+j+1} - s_{i+j+2}) v_i w_j. \end{cases}$$

The s_i 's are given by $s_i = \sum_{r=1}^{n+1} \lambda_r x_r^i$. The linear map $\lambda \mapsto (s_0, \dots, s_n)$ is one to one, since (s_0, \dots, s_n) is obtained by multiplying λ by a Vandermonde matrix, which is invertible. A direct consequence is the following.

LEMMA 4.3. *The matrices $\{B_r\}_{1 \leq r \leq r_*}$ are linearly independent.*

Proof. Assume $\sum_{r=0}^n \lambda_r B_r = 0$. Then (4.5) and the definition of H_1 and H_2 yields that $s_0 = \dots = s_n = 0$. It yields $\lambda = 0$. So the $\{B_r\}_{1 \leq r \leq r_*}$ are linearly independent. \square

PROPOSITION 4.4. *For any univariate polynomial p that is strictly positive on $\mathbb{K} = [0, 1]$, the associated function G is strictly convex and coercive. As a consequence, it has a unique critical point λ^* which defines a sum of squares decomposition $p[\lambda^*] = p$.*

Proof. Thanks to Corollary 4.2 and Lemma 4.3, the assumptions of Proposition 3.8 and Proposition 3.10 are satisfied which yields the result. \square

5. Numerical algorithms. Now we introduce several descent and Newton methods for the computation of sum of squares decompositions, via the minimization of G .

5.1. Descent methods. Descent methods (5.1) are based on the minimization of the dual function G either by a descent type algorithm, either by the direct search of a critical point with a Newton type methods. All the methods enter the generic iterative framework

$$(5.1) \quad \lambda_{m+1} = \lambda_m - \tau_m H_m^{-1} \nabla G(\lambda_m), \quad \lambda_0 = 0,$$

with H_m and τ_m to be defined. The latter is an adaptive time step ensuring the decay of $\|\nabla G(\lambda_m)\|$ at each step. We recall that this quantity actually measures the Euclidean norm between the current sum of squares $(p[\lambda_m](x_r))_r$ and \mathbf{y} . The adaptive time step τ_m is defined as follows. Let us define $\lambda_m^{(k)} = \lambda_m - 2^{-k} \tau_m^{(k)} H_m^{-1} \nabla G(\lambda_m)$. Then we denote by k_m the smallest integer such that $\|\nabla G(\lambda_m^{(k)})\| < \|\nabla G(\lambda_{m+1})\|$. From there we define $\tau_{m+1} = 2^{-k_m} \tau_m$ for $k_m > 0$ and $\tau_{m+1} = 2\tau_m$ for $k_m = 0$. Another possibility for the control of the steps τ_m is proposed in conclusion.

5.1.1. Forward descent method. The first method we use is the classical descent method which consists in taking $H_m = I$ the identity matrix.

5.1.2. Backward descent method. Given a sequence of positive time steps τ_m , the following iterative scheme $\tilde{\lambda}_{m+1} = \arg \min_{\lambda \in \mathcal{D}} G(\lambda) + \frac{1}{2\tau_m} \|\lambda - \tilde{\lambda}_m\|^2$ with initial guess $\tilde{\lambda}_0 = 0$ is well defined since G is convex. It corresponds exactly to the implicit Euler discretization of the gradient flow with variable time steps. At step m we look for the critical point of the strictly convex objective function by making one step of a Newton method starting at λ_m , yielding the scheme (5.1) with $H_m = I + \tau_m \nabla^2 G(\lambda_m)$. The adaptive time step is chosen as in Section 5.1.

5.2. Newton-Raphson methods. Newton-Raphson methods can be understood as acceleration techniques for descent methods. We also threshold the maximal time $\tau_m \leq 1$, since it is the theoretical value of the Newton-Raphson method.

5.2.1. Newton-Raphson method. A straightforward method for a direct search of the critical point of G is the classical Newton method $H_m = \nabla^2 G(\lambda_m)$, with $\nabla^2 G$ the Hessian of G . The time step τ_m is computed as in Section 5.1.

5.2.2. Modified Newton-Raphson method. The Hessian of G degenerates at infinity as showed in Lemma 3.11. In practice, a classical Newton-Raphson method for solving $\nabla G(\lambda) = 0$ can be inaccurate at the first iterations in some cases. Instead one may notice that λ_* is a critical point of $G(\lambda)$ if and only if it is a critical point of $(G(\lambda) - C)^2$ where C is a constant which is smaller than the infimum of G . One expect the latter function to grow quadratically at infinity thus improving the conditioning of the Hessian. This suggests the modified Newton method (5.1) with $H_m = \alpha_m \nabla G(\lambda_m) \otimes \nabla G(\lambda_m) + \nabla^2 G(\lambda_m)$. The time step τ_m is chosen as in the previous sections. Several choices are possible for α_m . Following the heuristic one could impose $\alpha_m = (G(\lambda_m) - K)^{-1}$. In practice, we found out that the (empirical) choice $\alpha_m = \|\nabla G(\lambda_m)\| / (\|\nabla G(\lambda_m)\| + \|\nabla G(0)\|)$ yields good results.

6. Numerical experiments. In this section, we perform various numerical experiments in order to illustrate the theoretical results and to explore the behavior of the numerical algorithms. The implementation has been performed with Matlab and Python, with no noticeable difficulties.

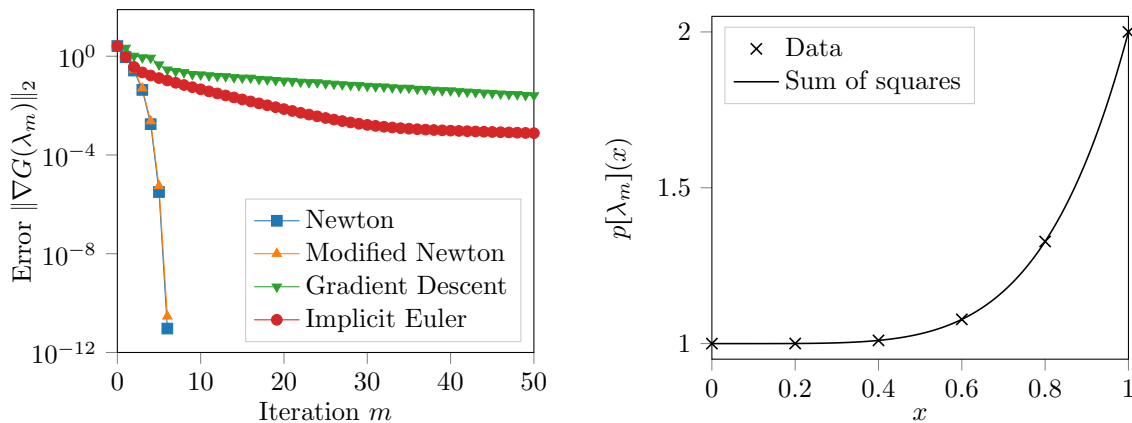


FIG. 1. Test case 1. Sum of square interpolation of $p(x) = x^5 + 1$. (Top left) Error $\|\nabla G(\lambda_m)\|_2$ vs. iteration m ; (Top right) Step size τ_m vs. iteration m ; (Bottom) Data $(y_r = p(x_r))_r$ and sum of squares $p[\lambda](x)$ satisfying $\|\nabla G(\lambda)\| < 10^{-8}$.

6.1. Univariate polynomials on a segment. Here we consider univariate SOS polynomials. We proceed as explained in Section 4, except that the monomial basis is replaced here by the orthogonal basis of shifted Chebychev polynomials $(T_i(x))_{i=1,\dots,k}$ satisfying $T_i(\cos(\theta) + 1)/2 = \cos(i\theta)$, for all $\theta \in \mathbb{R}$. The only modification of the method presented earlier concerns the definition of the D_r matrices which become $D_r = \mathbf{w}_r^t \mathbf{w}_r \in \mathbb{R}^{r_k \times r_k}$ with $\mathbf{w}_r = (T_0(x_r), T_1(x_r), \dots, T_k(x_r))^t \in \mathbb{R}^{r_k}$. The reason is that shifted Chebychev polynomials have much better behavior in terms of numerical approximation, since they produce "uniformly distributed" polynomials in $[0, 1]$, see [7] for comprehensive mathematical treatment. On the opposite, monomials x^i which concentrate at $x = 1$ for $i \rightarrow +\infty$ are non optimal for numerical approximation in the segment $[0, 1]$. One can refer to [5] for a comparison between the use of Chebychev polynomials and monomials. In the following we propose different test cases to illustrate the various properties of the various descent and Newton-Raphson type methods proposed in Section 5. For univariate polynomials, the tests 1-2-3 are performed with the odd order option (4.3) of the weights: similar results are observed with $g_1(x) = 1$ and $g_2(x) = x(1 - x)$, and so are not reported. Test 5 is performed with both the odd and even options.

6.1.1. Test case 1. We compare the convergence of the methods for an easy objective polynomial, that is a polynomial with low degree and far above 0: we take $n = 5$, $r_* = i_* = n + 1 = 6$, $p(x) = x^5 + 1$ and the weights $g_1(x) = x$ with $g_2(x) = 1 - x$ (so $j_* = 2$).

We observe on Figure 1 that the Newton type methods both reach the threshold precision of 10^{-8} after only 6 iterations. The implicit Euler and gradient descent methods need respectively 573 and 2727 iterations to reach the same error: this low convergence has been observed for many other test problems. This is why we continue the tests with the Newton methods only.

6.1.2. Test case 2. In this second test case, we illustrate the better performance of the modified Newton-Raphson method compared to the standard Newton-Raphson method. We choose a highly oscillating objective polynomial with lower bound equal to 0. It is given by $n = 21$, $r_* = i_* = n + 1 = 22$, $p(x) = T_{21}(x) + 1$ and the weights $g_1(x) = x$ with $g_2(x) = 1 - x$ (so $j_* = 2$).

We observe on Figure 2 that the modified Newton-Raphson method reaches a precision of around 10^{-8} in 40 iterations. In the case of the standard Newton-Raphson method, the adaptive time step quickly reduces to a very small value in order to keep decreasing the error at each iteration. A similar phenomena happens near convergence for the modified Newton-Raphson method. These behaviors can be interpreted thanks to the evolution of the condition number of the matrix H_m also showed on

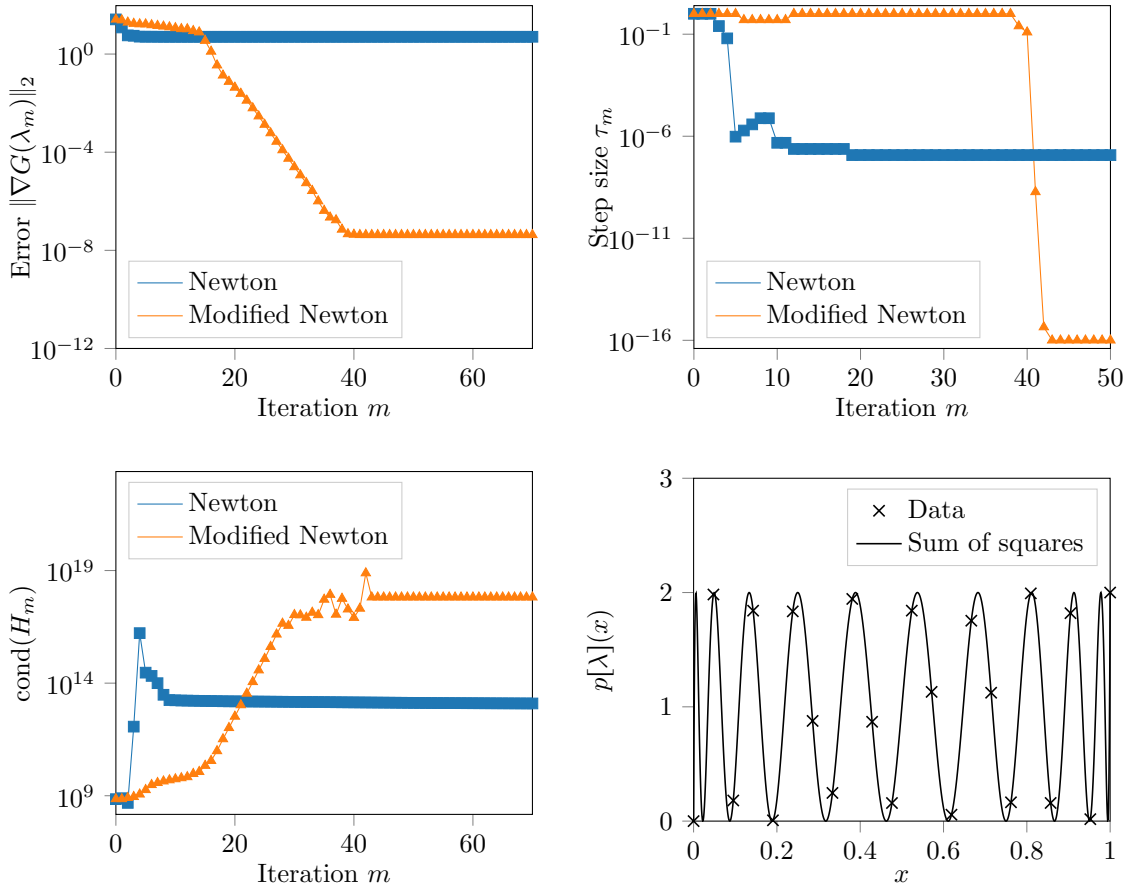


FIG. 2. Test case 2. Sum of square interpolation of $p(x) = T_{21}(x) + 1$. (Top left) error $\|\nabla G(\lambda_m)\|_2$ vs. iteration m ; (Top right) Step size τ_m vs. iteration m ; (Bottom left) Condition number of H_m vs. iteration m ; (Bottom right) Data $(y_r = p(x_r))_r$ and sum of squares $p[\lambda](x)$ satisfying $\|\nabla G(\lambda)\| < 10^{-6}$.

Figure 2. Let us recall that this matrix needs to be inverted at each iteration. On the first hand, for the Newton-Raphson method, H_m is the Hessian of G which degenerates when λ is far from the minimizer of G , as explained in Lemma 3.11. The modified Newton-Raphson method seems to prevent a bad condition number of the tweaked Hessian in the first few iterations. On the second hand, since the objective polynomial has a 0 lower bound, strict convexity and coercivity of G are not granted and it may explain the bad conditioning of H_m near convergence in the case of the modified Newton-Raphson method. Indeed recall that when $\nabla G(\lambda_m)$ is small H_m almost coincides with the Hessian in the modified Newton-Raphson method.

Nonetheless we found in many numerical experiments that the latter numerical method is the most robust and efficient of the four. This the reason why we only use the modified Newton-Raphson method in the following series of tests.

6.1.3. Test case 3. Now, we illustrate the influence of the lower bound of p on the convergence of the method. To proceed, we compute a sum of squares approximation of the polynomial $p(x) = T_{11}(x) + 1 + \alpha$ for various lower bounds α ($n = 5$, $r_* = i_* = n + 1 = 6$, $j_* = 2$). The results are displayed on Figure 3. We observe that the number of iterations required to reach a

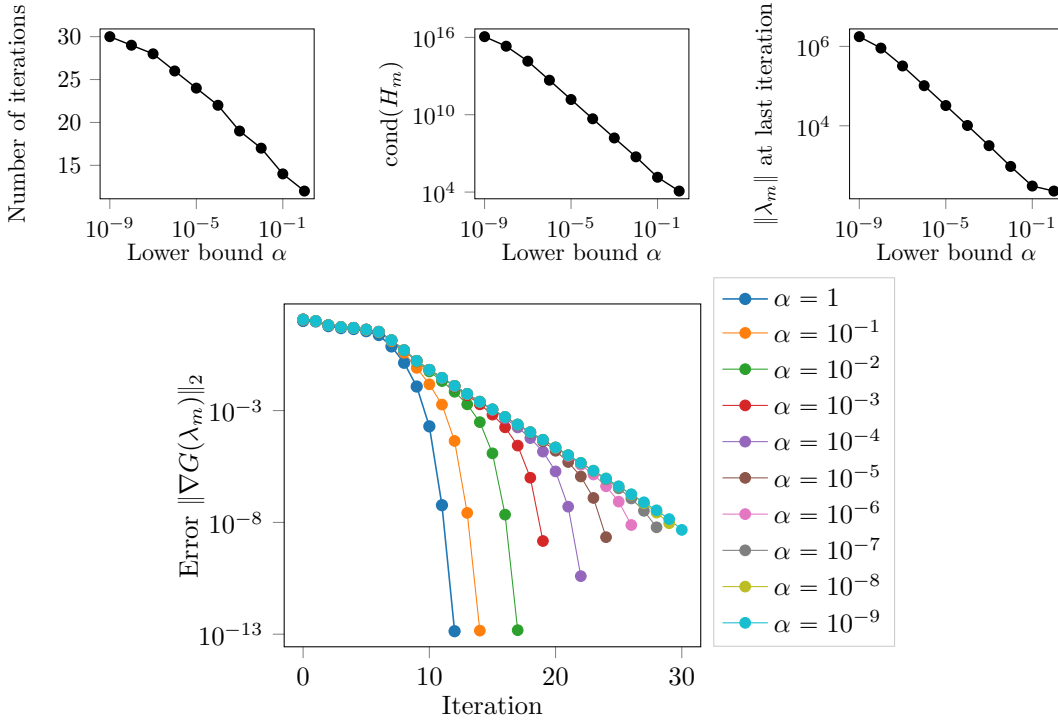


FIG. 3. Test case 3. Influence of the lower bound α in the sum of square interpolation of $p(x) = (T_{11}(x) + 1) + \alpha$. (Top left) Number of iterations to converge vs. α ; (Top right) Condition number of H_m at the last iteration vs. α ; (Bottom) Error $\|\nabla G(\lambda_m)\|_2$ vs. iteration m for different lower bounds α .

precision of 10^{-8} seems to increase proportionally with $|\log(\alpha)|$. The condition number of H_m and the norm of λ_m at convergence decays like some negative power of α . Interestingly enough, one also sees that the quadratic convergence of the (modified) Newton method seems to degenerate to linear convergence when α goes to 0. All these behaviors can be interpreted thanks to the results of Lemma 3.5 and Lemma 3.11. We know from Lemma 3.5 that for $\alpha = 0$, p has a root x_0 in $[0, 1]$, and thus the coercivity of G is lost in some direction of the asymptotic cone of \mathcal{D} (that of the Lagrange vector $L(x_0)$). Thus as $\alpha \rightarrow 0$, the minimizer λ_α^* may go to $+\infty$ in the asymptotic cone which would explain here the explosion of the norm of λ and of the condition number of H_m as predicted by Lemma 3.11 and shown on Figure 3.

6.1.4. Test case 4. In this fourth test case we illustrate the influence of the degree n of the objective polynomial $p(x) = x^n + 1$ on the convergence of our method, with $g_1(x) = x$ and $g_2(x) = 1 - x$ for n odd and $g_1(x) = 1$ and $g_2(x) = x(1 - x)$ for n even.

The result are displayed on Figure 4. We observe that the number of iterations required to reach an error of 10^{-8} increases with the degree, but weakly. We also observe that the condition number $\text{cond}(H_m) = \|H_m\| \|H_m^{-1}\|$ near convergence deteriorates with n , approximatively quadratically.

6.2. Bivariate polynomials on a triangle. We use the minimization algorithm for the computation of a sum of squares representation of some positive polynomial $p \in P_n[X, Y]$ on the triangle.

6.2.1. Numerical setting. The barycentric coordinates corresponding to the vertices S_1 , S_2 and S_3 of the triangle are denoted as μ_j for $j = 1, 2, 3$: $\mu_1(x, y) = 1 - x - y$, $\mu_2(x, y) = x$ and $\mu_3(x, y) = y$.

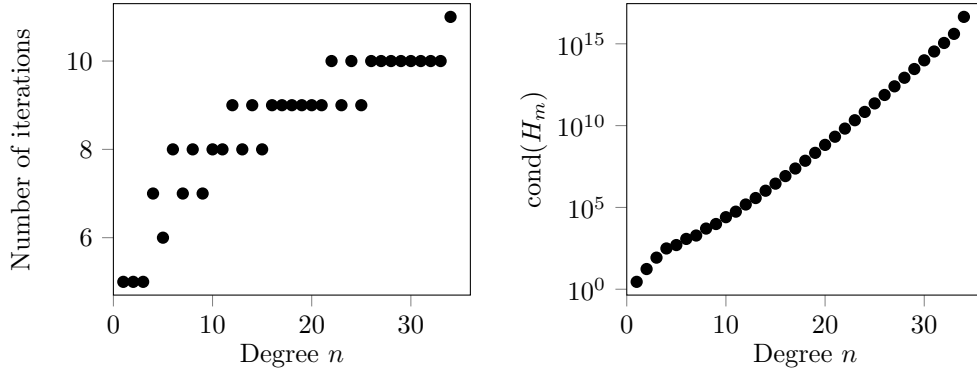


FIG. 4. *Test case 4.* Influence of the degree n in the sum of square interpolation of $p(x) = x^n + 1$. (Left) Number of iterations to converge vs. n ; (Right) Condition number of H_m at the last iteration vs. n .

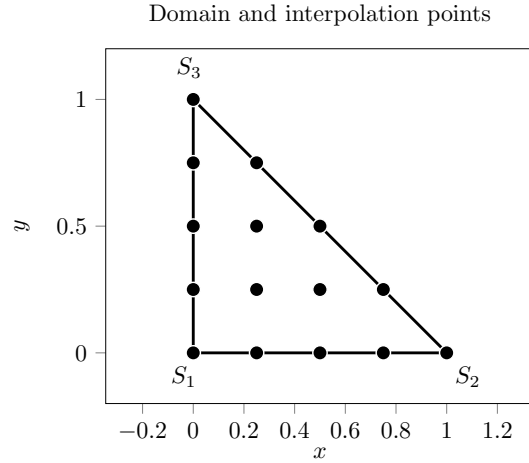


FIG. 5. The simplex \mathbb{K} and interpolation points for $n = 4$.

The triangle is $\mathbb{K} = \{\mathbf{x} = (x, y) \in \mathbb{R}^2 \mid \mu_1(\mathbf{x}) \geq 0, \mu_2(\mathbf{x}) \geq 0, \mu_3(\mathbf{x}) \geq 0\}$. The interpolation points are $\mathbf{x}_r = (x_r, y_r)$ for $1 \leq r \leq r_* = (n+1)(n+2)/2$ are the distinct points of a cartesian grid intersected with the triangle. For a given polynomial $p \in \mathbb{P}^n[\mathbf{X}]$ of a given degree, the data is $\mathbf{z} \in \mathbb{R}^{r_*}$ which is the vector with components $z_r = p(x_r, y_r)$. An illustration of the geometry is provided in Figure 5 where the degree is $n = 4$.

We consider the ansatz ($\mathbf{x} = (x, y)$)

$$(6.1) \quad p[\lambda](\mathbf{x}) = \sum_{i=1}^{r_j} g_i(\mathbf{x}) p_{i1}[\lambda](\mathbf{x})^2 + g_2(\mathbf{x}) p_{i2}[\lambda](\mathbf{x})^2 + g_3(\mathbf{x}) p_{i3}[\lambda](\mathbf{x})^2 + g_4(\mathbf{x}) p_{i4}[\lambda](\mathbf{x})^2,$$

where, arbitrarily with respect to the literature [12], the weights are

$$(6.2) \quad \begin{cases} \text{for } n = 2k + 1, & g_i = \mu_i \text{ for } i = 1, 2, 3 \text{ and } g_4 = \mu_1 \mu_2 \mu_3, \\ \text{for } n = 2k, & g_1 = \mu_2 \mu_3, g_2 = \mu_3 \mu_1, g_3 = \mu_1 \mu_2 \text{ and } g_4 = 1. \end{cases}$$

With this choice we recover in every cases $r_* = r_1 + r_2 + r_3 + r_4$. All polynomials are parametrized on

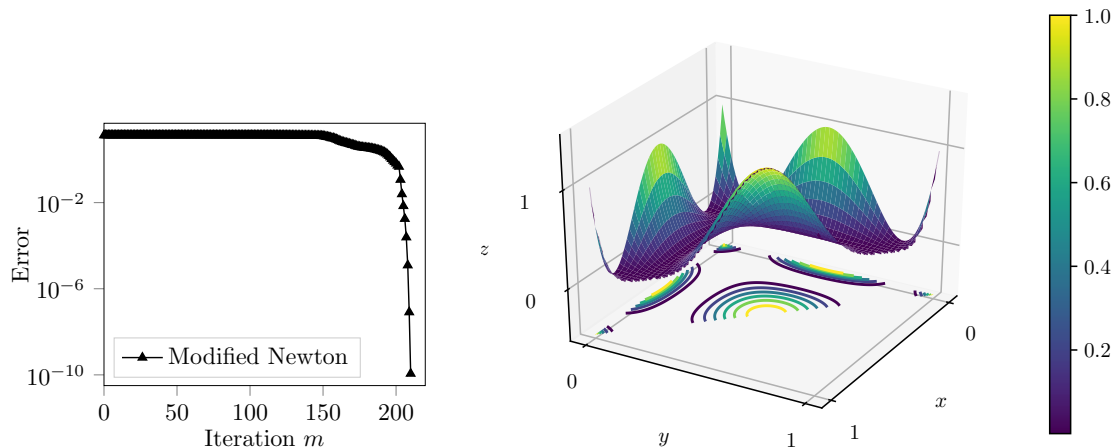


FIG. 6. Test case 5. Bivariate sum of square interpolation of the degree 8 polynomial $p(x, y) = (T_4(x) + 1)(T_4(y) + 1)/4 + 10^{-3}$ on the 2D simplex. (Left) error $\|\nabla G(\lambda_m)\|_2$ vs. iteration m ; (Right) surface plot of the converged sum of square.

the basis of bivariate monomials since Chebychev polynomials are not available on the triangle.

6.2.2. Test case 5. We approach the polynomial $p(x, y) = (T_4(x) + 1)(T_4(y) + 1)/4 + 10^{-3}$ on the 2D simplex with the modified Newton method. The parameters are $n = 8$, $r_* = i_* = 45$ and $j_* = 4$.

We observe on Figure 6 that our method converges in this multivariate setting and reaches a precision of less than 10^{-8} in 210 iterations. The error decays slowly during the first 200 iterations before reaching usual quadratic speed of convergence of the Newton method near the minimizer of G . This result illustrates the ability of our algorithms to provided a computational strategy for the computation of positive polynomials on bi-dimensional sets.

6.2.3. Test case 6. In this last test case we are interested in the SOS approximation of the Motzkin polynomial [17] $p(x, y) = x^2y^4 + y^2x^4 - 3x^2y^2 + 1$.

This polynomial is non-negative over \mathbb{R}^2 and famous for not being a sum of square in the sense that it admits no decomposition (1.3) with weights $\tilde{g}_1 = \dots = \tilde{g}_{j_*} = 1$ (whatever the choice of i_* or, equivalently in this particular case, j_*). The parameters are $n = 6$, $r_* = i_* = 28$ and $j_* = 4$. We use our method to approach this polynomial with the sum of square ansatz (6.1) but with two different weights: on the one hand we use the weights g_i (6.2) for which we expect some convergence of the algorithm; on the other other hand we use the weights $\tilde{g}_i = 1$ for $i = 1, 2, 3, 4$.

In the latter case our experiment on Figure 7 show the method does not converge (in coherence with the non-existence of a sum of square decomposition for the Motzkin polynomial). The algorithm with weights g_i converges while the algorithm with weights \tilde{g}_i does not converge (bottom right illustration in the Figure).

7. Concluding remarks. All our numerical results show that the modified Newton-Raphson algorithm is able to compute polynomials which respect a sign condition on a given simple semi-algebraic set. However more needs to be investigated to evaluate the full potential of such methods. Here we detail to possible domains of research which are consequences of the multiple connections of our methods with the ones of Scientific Computing.

- It is possible to look in more details in the case $i_* \neq r_*$. It allows greater generality of the construction, which can be convenient for optimization purposes. In such cases, the function G should be replaced

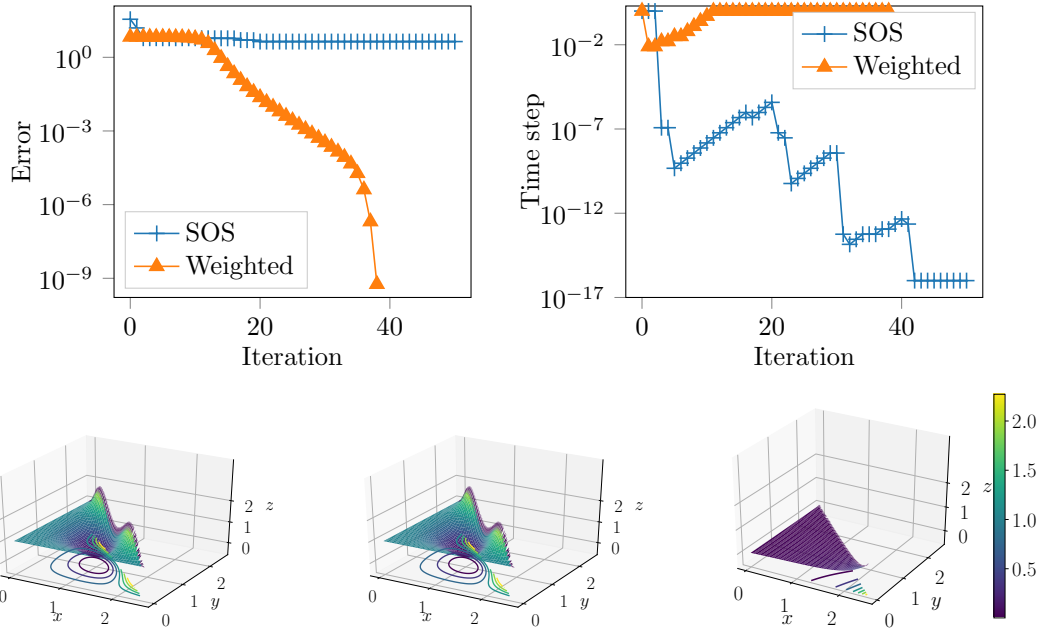


FIG. 7. **Test case 6.** Bivariate sum of square approximations (of degree $n = 6$) of the Motzkin polynomial. (Top left) error $\|\nabla G(\lambda_m)\|_2$ vs. iteration m ; (Top right) time step τ_m vs. iteration m ; (Bottom left) The Motzkin polynomial; (Bottom center) Sum of square approximation with weights $g_1 = \mu_2 \mu_3$, $g_2 = \mu_3 \mu_1$, $g_3 = \mu_1 \mu_2$ and $g_4 = 1$, the algorithm has converged; (Bottom right) Sum of square approximation without weights ($g_1 = g_2 = g_3 = g_4 = 1$), the algorithm has not converged;

by $G_{\mathbf{v}}$.

- The technical conditions on the linear independance of the matrices B_r in the multivariate case needs further examinations. In this direction there may be links with algebraic properties such as the Archimedeanity of the quadratic module associated with the weights g_j (see [12, proof of Theorem 2.14]).
- A C++ implementation needs to be tested. On this basis it will be possible to couple with codes in scientific computing (such as the ones evoked in [22] and the references therein) to evaluate the gain in robustness with the new algorithms. Comparisons with other established softwares like the primal-dual interior-point SDP Mosek-Yalmip package [15] in Matlab will be a plus. Such benchmarks are left for future work.
- Finally we mention that an implementation of the gradient algorithms with absolute guarantee of the condition $\lambda^{m+1} \in \mathcal{D}$ is possible. Indeed start from $\lambda^m \in \mathcal{D}$. Since $\lambda^{m+1} = \lambda^m - \Delta t \mathbf{d}^m$ for a given direction $\mathbf{d}^m = (d_r^m)$, the condition $\lambda^{m+1} \in \mathcal{D}$ is satisfied provided $M(\lambda^m) - \Delta t \sum_r d_r^m B_r \geq 0$. It is satisfied under the sufficient condition $\rho(\sum_r d_r^m B_r) \Delta t < \rho(M(\lambda^m))^{-1}$: here $\rho(A)$ denotes the spectral radius of a given square matrix A . This condition is very much a CFL stability condition.

Appendix A. The asymptotic cone for univariate polynomials.

One can obtain a much better understanding of the cone at infinity, which exemplifies the role of the Lagrange interpolating polynomials. Given a subset $S \subset \mathbb{R}^{n+1}$ one denotes by $\text{coni}(S)$ the *conical hull* of S that is the set of linear combinations with non-negative coefficients of elements of S . The asymptotic cone can be constructed from the matrices (4.4) or (4.5) in the univariate case. The main result is the following, where the the Lagrange vectors are defined in (3.2).

THEOREM A.1. *The asymptotic cone of \mathcal{D} is generated by the Lagrange vectors $L(x)$ for $0 \leq x \leq 1$, that is $C_\infty = \text{coni}(\{L(x) \in \mathbb{R}^{n+1} \mid x \in [0, 1]\})$.*

We need some intermediate results in order to prove Theorem A.1. First, let us define $C_\infty^1 = \{\lambda \in C_\infty \mid \sum_{r=1}^{n+1} \lambda_r = 1\} \subset C_\infty$. Since $\sum_{r=1}^{r_*} l_r(x_r) = 1$ for all $1 \leq r \leq r_*$, one has $\sum_{r=1}^{r_*} l_r(X) = 1$. Therefore, with Lemma 3.2, we know that $\{L(x) \in \mathbb{R}^{n+1} \mid x \in [0, 1]\} \subset C_\infty^1$. The main point of the proof is to show that $C_\infty^1 \subset \{L(x) \in \mathbb{R}^{n+1}, x \in [0, 1]\}$. To do so we identify C_∞^1 with a subset of Borel probability measures on $[0, 1]$ using the theory of the moment problem for which an comprehensive reference is [11]. The proof of the Theorem invoked below in the proof is strongly related to the Lukacs decomposition of Theorem 4.1.

PROPOSITION A.2. *Let $\lambda \in \mathbb{R}^{n+1}$. The following are equivalents: a) The vector λ belongs to C_∞^1 ; b) There is a Borel probability measure σ on $[0, 1]$ such that*

$$(A.1) \quad \sum_{r=1}^{n+1} \lambda_r B_r = \int_{[0,1]} B(x) d\sigma(x).$$

Proof. Using (4.5), one can say that $\lambda \in C_\infty^1 \iff (s_0, \dots, s_n)$ are such that H_1 and H_2 are positive semidefinite matrices and $s_0 = 1$. By [11, Theorem 2.3, Theorem 2.4], this is equivalent to the existence of a Borel probability measure σ such that (A.1) holds. \square

COROLLARY A.3. *The set C_∞^1 is compact.*

Proof. Since, by Proposition A.2, the s_i 's are moments of a Borel probability measure on $[0, 1]$, one has $(s_0, \dots, s_n) \in [0, 1]^{n+1}$. Therefore, since $\lambda \mapsto (s_0, \dots, s_n)$ is linear and invertible (see Lemma 4.3), C_∞^1 is bounded. \square

We recall that a point λ of a convex set C is said to be an extreme point (see [9, III, Definition 2.3.1]) of C if for any $\lambda_1, \lambda_2 \in C$ such that $\lambda = (\lambda_1 + \lambda_2)/2$, one has $\lambda = \lambda_1 = \lambda_2$. We denote by $\text{ext}(C)$ the set of extreme points of C .

PROPOSITION A.4. *The set of extreme points of C_∞^1 is $\text{ext}(C_\infty^1) = \{L(x) \mid x \in [0, 1]\}$.*

Proof. Let $\lambda \in \text{ext}(C_\infty^1)$. Since extreme points of a convex set are located on its boundary there is a vector $\mathbf{V} \neq 0$ such that $\langle \sum_{r=1}^{n+1} \lambda_r B_r \mathbf{V}, \mathbf{V} \rangle = 0$. Let σ be a Borel measure satisfying (A.1) and define $q(X) = \langle B(X) \mathbf{V}, \mathbf{V} \rangle \geq 0$. One has $\int_{[0,1]} q(x) d\sigma(x) = 0$. Since q is not identically zero, the measure σ must be supported on a subset of the finite set of roots of q intersected with $[0, 1]$. Since q has degree n , σ has the form $\sigma = \sum_{k=1}^n \alpha_k \delta_{x_k}$ where $\sum_{k=1}^n \alpha_k = 1$, $0 \leq \alpha_k \leq 1$ and $x_k \in [0, 1]$, for some distinct x_1, \dots, x_n and where δ_{x_k} is the Dirac measure at x_k . Now assume that for some index k , $\alpha_k \in (0, 1)$. Then there is $k' \neq k$ such that $\alpha_{k'} \in (0, 1)$. Then let $0 \leq \varepsilon < \min(\alpha_k, \alpha_{k'}, 1 - \alpha_k, 1 - \alpha_{k'})$ and define $\sigma_1 = \sigma - \varepsilon \delta_{x_k} + \varepsilon \delta_{x_{k'}}$ and $\sigma_2 = \sigma + \varepsilon \delta_{x_k} - \varepsilon \delta_{x_{k'}}$. The measures σ_1 and σ_2 are two Borel probability measures generating different sets of moments for at least some ε in the range. Since $\lambda \mapsto (s_0, \dots, s_n)$ is linear and invertible there are distinct $\lambda_1, \lambda_2 \in C_\infty^1$ satisfying (A.1) for the respective measures σ_1 and σ_2 and one has $\lambda = (\lambda_1 + \lambda_2)/2$. There is a contradiction. Therefore either $\alpha_k = 0$ or $\alpha_k = 1$ so σ must be a dirac measure at some point $x_* \in [0, 1]$. Hence $\sum_{r=1}^{n+1} \lambda_r B(x_r) = B(x_*)$ so in particular $\sum_{r=1}^{n+1} \lambda_r x_r^k = x_*^k$ for any $0 \leq k \leq n$ which yields $\lambda = L(x_*)$. Conversely if $\lambda = L(x_*)$ and $\lambda = (\lambda_1 + \lambda_2)/2$, then there are probability measures σ_1 and σ_2 such that $\delta_{x_*} = (\sigma_1 + \sigma_2)/2$. Therefore σ_1 and σ_2 are supported at x_* and since they have the same mass one has $\delta_{x_*} = \sigma_1 = \sigma_2$, so $\lambda \in \text{ext}(C_\infty^1)$. \square

Proof of Theorem A.1. Denote by $\text{co}(S)$ the convex hull of S , the set of linear combinations of elements of S with non-negative coefficients whose sum equals 1. By the Minkowski (or Krein-Milman) theorem [9, III, Theorem 2.3.4], any compact convex set is the convex hull of its extreme points, therefore $C_\infty^1 = \text{co}(\text{ext}(C_\infty^1))$. Remark that $C_\infty = \bigcup_{t \geq 0} t C_\infty^1$ and the result follows. \square

REFERENCES

- [1] L. BEIRÃO DA VEIGA, A. BUFFA, G. SANGALLI, AND R. VÁZQUEZ, *Mathematical analysis of variational isogeometric methods*, Acta Numer., 23 (2014), pp. 157–287.
- [2] S. BOYD AND L. VANDENBERGHE, *Convex optimization*, Cambridge University Press, Cambridge, 2004.
- [3] S. BURER AND R. D. C. MONTEIRO, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program., 95 (2003), pp. 329–357. Computational semidefinite and second order cone programming: the state of the art.
- [4] F. CHARLES, M. CAMPOS-PINTO, AND B. DESPRÉS, *Algorithms for positive polynomial approximation*, to appear in Siam J. Numer. Analysis, (2017). Online at <https://hal.sorbonne-universite.fr/hal-01527763>.
- [5] B. DESPRÉS, *Polynomials with bounds and numerical approximation*, Numerical Algor., 76 (2017), pp. 829–859.
- [6] B. DESPRES AND M. HERDA, *Correction to: Polynomials with bounds and numerical approximation*, Num. Algor., (2017).
- [7] R. A. DEVORE AND G. G. LORENTZ, *Constructive approximation*, vol. 303 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, 1993.
- [8] D. HENRION AND J. MALICK, *Projection methods for conic feasibility problems: applications to polynomial sum-of-squares decompositions*, Optimization Methods & Software, 26 (2011), pp. 23–46.
- [9] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Convex analysis and minimization algorithms. I*, vol. 305 of Fundamental Principles of Mathematical Sciences, Springer-Verlag, 1993.
- [10] M. KORDA, D. HENRION, AND C. N. JONES, *Convergence rates of moment-sum-of-squares hierarchies for optimal control problems*, Systems Control Lett., 100 (2017), pp. 1–5.
- [11] M. G. KREĪN AND A. A. NUDEL'MAN, *The Markov moment problem and extremal problems*, American Mathematical Society, Providence, R.I., 1977.
- [12] J. B. LASSERRE, *Moments, positive polynomials and their applications*, vol. 1 of Imperial College Press Optimization Series, Imperial College Press, London, 2010.
- [13] J. B. LASSERRE, *An introduction to polynomial and semi-algebraic optimization*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 2015.
- [14] B.-G. LEE, T. LYCHE, AND K. MØRKEN, *Some examples of quasi-interpolants constructed from local spline projectors*, in Mathematical methods for curves and surfaces (Oslo, 2000), Innov. Appl. Math., Vanderbilt Univ. Press, Nashville, TN, 2001, pp. 243–252.
- [15] J. LÖFBERG, *Yalmip : A toolbox for modeling and optimization in matlab*, in In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.
- [16] J. MALICK, *A dual approach to semidefinite least-squares problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 272–284.
- [17] T. S. MOTZKIN, *The arithmetic-geometric inequality*, Inequalities (Proc. Sympos. Wright-Patterson Air Force Base, Ohio, 1965), (1967), pp. 205–224.
- [18] Y. NESTEROV, *Squared functional systems and optimization problems*, in High performance optimization, vol. 33 of Appl. Optim., Kluwer Acad. Publ., Dordrecht, 2000, pp. 405–440.
- [19] Y. NESTEROV AND A. NEMIROVSKII, *Interior-point polynomial algorithms in convex programming*, vol. 13 of SIAM Studies in Applied Mathematics, SIAM, Philadelphia, PA, 1994.
- [20] V. POWERS AND T. WÖRMANN, *An algorithm for sums of squares of real polynomials*, J. Pure Appl. Algebra, 127 (1998), pp. 99–104.
- [21] M. PUTINAR, *Positive polynomials on compact semi-algebraic sets*, Ind. Univ. Math. J., 42 (1993), pp. 969–984.
- [22] C.-W. SHU, *Bound-preserving high order finite volume schemes for conservation laws and convection-diffusion equations*, in Finite volumes for complex applications VIII, vol. 199 of Springer, 2017, pp. 3–14.
- [23] J. F. STURM, *Theory and algorithms of semidefinite programming*, in High performance optimization, vol. 33 of Appl. Optim., Kluwer Acad. Publ., Dordrecht, 2000, pp. 3–19.
- [24] G. SZEGŐ, *Orthogonal polynomials*, American Mathematical Society, Providence, R.I., fourth ed., 1975. American Mathematical Society, Colloquium Publications, Vol. XXIII.