

Certification of Bounds of Non-linear Functions: the Templates Method

Xavier Allamigeon¹, Stéphane Gaubert², Victor Magron³, and Benjamin Werner⁴

¹ INRIA and CMAP, École Polytechnique, Palaiseau, France,
`Xavier.Allamigeon@inria.fr`

² INRIA and CMAP, École Polytechnique, Palaiseau, France,
`Stephane.Gaubert@inria.fr`

³ INRIA and LIX, École Polytechnique, Palaiseau, France,
`magron@lix.polytechnique.fr`

⁴ INRIA and LIX, École Polytechnique, Palaiseau, France,
`benjamin.werner@polytechnique.edu`

Abstract. The aim of this work is to certify lower bounds for real-valued multivariate functions, defined by semialgebraic or transcendental expressions. The certificate must be, eventually, formally provable in a proof system such as Coq. The application range for such a tool is widespread; for instance Hales' proof of Kepler's conjecture yields thousands of inequalities. We introduce an approximation algorithm, which combines ideas of the max-plus basis method (in optimal control) and of the linear templates method developed by Manna et al. (in static analysis). This algorithm consists in bounding some of the constituents of the function by suprema of quadratic forms with a well chosen curvature. This leads to semialgebraic optimization problems, solved by sum-of-squares relaxations. Templates limit the blow up of these relaxations at the price of coarsening the approximation. We illustrate the efficiency of our framework with various examples from the literature and discuss the interfacing with Coq.

Keywords: Polynomial Optimization Problems, Hybrid Symbolic-numeric Certification, Semidefinite Programming, Transcendental Functions, Semialgebraic Relaxations, Flyspeck Project, Quadratic Cuts, Max-plus Approximation, Templates Method, Proof Assistant.

1 INTRODUCTION

Numerous problems coming from various fields boil down to the computation of a certified lower bound for a real-valued multivariate function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ over a compact semialgebraic set $K \subset \mathbb{R}^n$.

Our aim is to automatically provide lower bounds for the following global optimization problem:

$$f^* := \inf_{\mathbf{x} \in K} f(\mathbf{x}) , \quad (1.1)$$

We want these bounds to be certifiable, meaning that their correctness must be, eventually, formally provable in a proof system such as Coq. One among many applications is the set of several thousands of non-linear inequalities which occur in Thomas Hales’ proof of Kepler’s conjecture, which is formalized in the Flyspeck project [1, 2]. Several inequalities issued from Flyspeck actually deal with special cases of Problem (1.1). For instance, f may be a multivariate polynomial (polynomial optimization problems (POP)), or belong to the algebra \mathcal{A} of semialgebraic functions which extends multivariate polynomials with arbitrary compositions of $(\cdot)^p$, $(\cdot)^{\frac{1}{p}}$ ($p \in \mathbb{N}_0$), $|\cdot|$, $+$, $-$, \times , $/$, $\sup(\cdot, \cdot)$, $\inf(\cdot, \cdot)$ (semialgebraic optimization problems), or involve transcendental functions (\sin , \arctan , *etc.*).

Formal methods that produce precise bounds are mandatory because of the tightness of these inequalities. However, we also need to tackle scalability issues, which arise when one wants to provide coarser lower bounds for optimization problems with a larger number of variables or polynomial inequalities of a higher degree, *etc.* A common idea to handle Problem (1.1) is to first approximate f by multivariate polynomials through a semialgebraic relaxation and then obtain a lower bound of the resulting POP with a specialized software. This implies being able to also certify the approximation error in order to conclude. Such techniques rely on hybrid symbolic-numeric certification methods, see Peyrl and Parrilo [3] and Kaltofen et al. [4]. They allow one to produce positivity *certificates* for such POP which can be checked in proof assistants such as Coq [5, 6], HOL-light [7] or MetiTarski [8]. Recent efforts have been made to perform a formal verification of several Flyspeck inequalities with Taylor interval approximations [9]. We also mention procedures that solve SMT problems over the real numbers, using interval constraint propagation [10].

Solving POP is already a hard problem, which has been extensively studied. Semidefinite programming (SDP) relaxations based methods have been developed by Lasserre [11] and Parrilo [12]. A sparse refinement of the hierarchy of SDP relaxations by Kojima [13] has been implemented in the SparsePOP solver. Other approaches are based on Bernstein polynomials [14], global optimization by interval methods (see e.g. [15]), branch and bound methods with Taylor models [16].

Inequalities involving transcendental functions are typically difficult to solve with interval arithmetic, in particular due to the correlation between arguments of unary functions (e.g. \sin) or binary operations (e.g. $+$, $-$, \times , $/$). For illustration purpose, we consider the following running example coming from the global optimization literature:

Example 1 (Modified Schwefel Problem 43 from Appendix B in [17]).

$$\min_{\mathbf{x} \in [1, 500]^n} f(\mathbf{x}) = - \sum_{i=1}^n (x_i + \epsilon x_{i+1}) \sin(\sqrt{x_i}),$$

where $x_{n+1} = x_1$, and ϵ is a fixed parameter in $\{0, 1\}$. In the original problem, $\epsilon = 0$, *i.e.* the objective function f is the sum of independent functions involving a single variable. This property may be exploited by a global optimization solver

by reducing it to the problem $\min_{x \in [1, 500]} x \sin(\sqrt{x})$. Hence, we also consider a modified version of this problem with $\epsilon = 1$.

Contributions. In this paper, we present an exact certification method, aiming at handling the approximation of transcendental functions and increasing the size of certifiable instances. It consists in combining SDP relaxations à la Lasserre / Parrilo, with an abstraction or approximation method. The latter is inspired by the linear template method of Sankaranarayanan, Sipma and Manna in static analysis [18], its nonlinear extension by Adjé et al. [19], and the maxplus basis method in optimal control introduced by Fleming and McEneaney [20], and developed by several authors [21–24].

The non-linear template method is a refinement of polyhedral based methods in static analysis. It allows one to determine invariants of programs by considering a parametric family of sets, $S(\alpha) = \{x \mid w_i(x) \leq \alpha_i, 1 \leq i \leq p\}$, where the vector $\alpha \in \mathbb{R}^p$ is the parameter, and w_1, \dots, w_p (the template) are fixed possibly non-linear functions, tailored to the program characteristics. The max-plus basis method is equivalent to the approximation of the epigraph of a function by a set $S(\alpha)$. In most basic examples, the functions w_i of the template are linear or quadratic functions.

In the present application, templates are used both to approximate transcendental functions, and to produce coarser but still tractable relaxations when the standard SDP relaxation of the semialgebraic problem is too complex to be handled. Indeed, SDP relaxations are a powerful tool to get tight certified lower bound for semialgebraic optimization problems, but their applicability is so far limited to small or medium size problems: their execution time grows exponentially with the relaxation order, which itself grows with the degree of the polynomials to be handled. Templates allow one to reduce these degrees, by approximating certain projections of the feasible set by a moderate number of nonconvex quadratic inequalities.

Note that by taking a trivial template (bound constraints, *i.e.*, functions of the form $w_i(x) = \pm x_i$), the template method specializes to a version of interval calculus, in which bounds are derived by SDP techniques. By comparison, templates allow one to get tighter bounds, taking into account the correlations between the different variables. They are also useful as a replacement of standard Taylor approximations of transcendental functions: instead of increasing the degree of the approximation, one increases the number of functions in the template. A geometrical way to interpret the method is to think of it in terms of “quadratic cuts”: quadratic inequalities are successively added to approximate the graph of a transcendental function.

The present paper is a followup of [25], in which the idea of max-plus approximation of transcendental function was applied to formal proof. By comparison, the new ingredient is the introduction of the template technique (approximating projections of the feasible sets), leading to an increase in scalability.

The paper is organized as follows. In Section 2, we recall the definition and properties of Lasserre relaxations of polynomial problems (Section 2.1), together

with reformulations by Lasserre and Putinar of semialgebraic problems classes. In Section 2.2, we outline the conversion of the numerical SOS produced by the SDP solvers into an exact rational certificate. Then we explain how to verify this certificate in Coq. The max-plus approximation, and the main algorithm based on the non-linear templates method are presented in Section 3. Numerical results are presented in Section 4. We demonstrate the scalability of our approach by certifying bounds of non-linear problems involving up to 10^3 variables, as well as non trivial inequalities issued from the Flayspeck project.

2 NOTATION AND PRELIMINARY RESULTS

Let $\mathbb{R}_d[\mathbf{x}]$ be the vector space of multivariate polynomials in n variables of degree d and $\mathbb{R}[\mathbf{x}]$ the set of multivariate polynomials in n variables. We also define the cone of sums of squares of degree at most $2d$:

$$\Sigma_d[\mathbf{x}] = \left\{ \sum_i q_i^2, \text{ with } q_i \in \mathbb{R}_d[\mathbf{x}] \right\}. \quad (2.1)$$

The set $\Sigma_d[\mathbf{x}]$ is a closed, fully dimensional convex cone in $\mathbb{R}_{2d}[\mathbf{x}]$. We denote by $\Sigma[\mathbf{x}]$ the cone of sums of squares of polynomials in n variables.

2.1 Constrained Polynomial Optimization Problems and SDP

We consider the general constrained polynomial optimization problem (POP):

$$f_{\text{pop}}^* := \inf_{\mathbf{x} \in K_{\text{pop}}} f_{\text{pop}}(\mathbf{x}), \quad (2.2)$$

where $f_{\text{pop}} : \mathbb{R}^n \rightarrow \mathbb{R}$ is a d -degree multivariate polynomial, K_{pop} is a compact set defined by inequalities $g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0$, where $g_j(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a real-valued polynomial of degree ω_j , for $j = 1, \dots, m$. Recall that the *set of feasible points* of an optimization problem is simply the domain over which the optimum is taken, i.e., here, K_{pop} .

Lasserre's hierarchy of semidefinite relaxations. We set $g_0 := 1$ and take $k \geq k_0 := \max(\lceil d/2 \rceil, \max_{1 \leq j \leq m} \lceil \omega_j/2 \rceil)$. We consider the following hierarchy of semidefinite relaxations for Problem (2.2), consisting of the optimization problems Q_k , $k \geq k_0$,

$$Q_k : \begin{cases} \sup_{\mu, \sigma_j} \mu \\ \text{s.t.} & f_{\text{pop}}(\mathbf{x}) - \mu = \sum_{j=0}^m \sigma_j(\mathbf{x})g_j(\mathbf{x}), \\ & \mu \in \mathbb{R}, \quad \sigma_j \in \Sigma_{k-\lceil \omega_j/2 \rceil}[\mathbf{x}], j = 0, \dots, m. \end{cases}$$

We denote by $\sup(Q_k)$ the optimal value of Q_k . A feasible point $(\mu, \sigma_0, \dots, \sigma_m)$ of Problem Q_k is said to be a *SOS certificate*, showing the implication $g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0 \implies f_{\text{pop}}(\mathbf{x}) \geq \mu$.

The sequence of optimal values $(\sup(Q_k))_{k \geq k_0}$ is non-decreasing. Lasserre showed [11] that it does converge to f_{pop}^* under certain assumptions on the polynomials g_j . Here, we will consider sets K_{pop} included in a box of \mathbb{R}^n , so that Lasserre's assumptions are automatically satisfied.

Application to semialgebraic optimization. Given a semialgebraic function f_{sa} , we consider the problem $f_{\text{sa}}^* = \inf_{\mathbf{x} \in K_{\text{sa}}} f_{\text{sa}}(\mathbf{x})$, where K_{sa} is a basic semialgebraic set. Moreover, we assume that f_{sa} has a basic semialgebraic lifting (for more details, see e.g. [26]). This implies that we can add auxiliary variables z_1, \dots, z_p (lifting variables), and construct polynomials $h_1, \dots, h_s \in \mathbb{R}[\mathbf{x}, z_1, \dots, z_p]$ defining the semialgebraic set $K_{\text{pop}} := \{(\mathbf{x}, z_1, \dots, z_p) \in \mathbb{R}^{n+p} : \mathbf{x} \in K_{\text{sa}}, h_1(\mathbf{x}, \mathbf{z}) \geq 0, \dots, h_s(\mathbf{x}, \mathbf{z}) \geq 0\}$, such that $f_{\text{pop}}^* := \inf_{(\mathbf{x}, \mathbf{z}) \in K_{\text{pop}}} z_p$ is a lower bound of f_{sa}^* . $f_{\text{sa}}^* := \inf_{(\mathbf{x}, \mathbf{z}) \in K_{\text{pop}}} z_p$

2.2 Hybrid Symbolic-Numeric Certification and Formalization

The previous relaxation Q_k can be solved with several semidefinite programming solvers (e.g. SDPA [27]). These solvers are implemented using floating-point arithmetics. In order to build formal proofs, we currently rely on exact rational certificates which are needed to make formal proofs: Coq, being built on a computational formalism, is well equipped for checking the correctness of such certificates.

Such rational certificates can be obtained by a rounding and projection algorithm of Peyrl and Parillo [3], with an improvement of Kaltfen et al. [4]. Note that if the SDP formulation of Q_k is not strictly feasible, then the rounding and projection algorithm fails. However, Monniaux and Corbineau proposed a partial workaround for this issue [5]. In this way, except in degenerate situations, we arrive at a candidate SOS certificate with rational coefficients, $(\mu, \sigma_0, \dots, \sigma_m)$. This certificate can straightforwardly be translated to Coq; the verification then boils down to formally checking that this SOS certificate does satisfy the equality constraint in Q_k with Coq's `field` tactic, which implies that $f_{\text{pop}}^* \geq \mu$. This checking is typically handled by generating Coq scripts from the OCaml framework, when the lower bound μ obtained at the relaxation Q_k is accurate enough.

Future improvements could build, for instance, on future Coq libraries handling algebraic numbers or future tools to better handle floating point approximations inside Coq.

3 MAX-PLUS APPROXIMATIONS AND NON-LINEAR TEMPLATES

3.1 Max-plus Approximations and Non-linear Templates

The max-plus basis method in optimal control [20, 21, 23] involves the approximation from below of a function f in n variables by a supremum

$$f \gtrsim g := \sup_{1 \leq i \leq p} \lambda_i + w_i . \quad (3.1)$$

The functions w_i are fixed in advance, or dynamically adapted by exploiting the problem structure. The parameters λ_i are degrees of freedom.

This method is closely related to the non-linear extension [19] of the template method [18]. This extension deals with parametric families of subsets of \mathbb{R}^n of the form $S(\alpha) = \{x \mid w_i(x) \leq \alpha_i, 1 \leq i \leq p\}$. The template method consists in propagating approximations of the set of reachable values of the variables of a program by sets of the form $S(\alpha)$. The non-linear template and max-plus approximation methods are somehow equivalent. Indeed, the 0-level set of g , $\{x \mid g(x) \leq 0\}$, is nothing but $S(-\lambda)$, so templates can be recovered from max-plus approximations, and vice versa.

The functions w_i are usually required to be quadratic forms,

$$w_i(x) = p_i^\top x + \frac{1}{2}x^\top A_i x \quad ,$$

where $p_i \in \mathbb{R}^n$ and A_i is a symmetric matrix. A basic choice is $A_i = -cI$, where c is a fixed constant, and I the identity matrix. Then, the parameters p remain the only degrees of freedom.

The consistency of the approximation follows from results of Legendre-Fenchel duality. Recall that a function f is said to be *c-semiconvex* if $x \mapsto f(x) + c\|x\|^2$ is convex. Then, if f is *c-semiconvex* and lowersemicontinuous, as the number of basis functions r grows, the best approximation $g \lesssim f$ by a supremum of functions of type (3.1), with $A_i = -cI$, is known to converge to f [20]. The same is true without semiconvexity assumptions if one allows A_i to vary [28].

A basic question is to estimate the number of basis functions needed to attain a prescribed accuracy. A typical result is proved in [24, Theorem 3.2], as a corollary of techniques of Grüber concerning the approximation of convex bodies by circumscribed polytopes. This theorem shows that if f is $c - \epsilon$ semiconvex, for $\epsilon > 0$, twice continuously differentiable, and if X is a full dimensional compact convex subset of \mathbb{R}^n , then, the best approximation g of f as a supremum of r functions as in (3.1), with $w_i(x) = p_i^\top x - c\|x\|^2/2$, satisfies

$$\|f - g\|_{L_\infty(X)} \simeq \frac{C(f)}{r^{2/n}} \quad (3.2)$$

where the constant $C(f)$ is explicit (it depends of $\det(f'' + cI)$ and is bounded away from 0 when ϵ is fixed). This estimate indicates that some curse of dimensionality is unavoidable: to get a uniform error of order ϵ , one needs a number of basis functions of order $1/\epsilon^{n/2}$. However, in what follows, we shall always apply the approximation to small dimensional constituents of the optimization problems ($n = 1$ when one needs to approximate transcendental functions in a single variable). We shall also apply the approximation by templates to certain relevant small dimensional projections of the set of lifted variables, leading to a smaller effective n . Note also that for optimization purposes, a uniform approximation is not needed (one only needs an approximation tight enough near the optimum, for which fewer basis functions are enough).

3.2 A Templates Method based on Max-plus Approximations

We now consider an instance of Problem (1.1). We assume that K is a box and we identify the objective function f with its abstract syntax tree t_f . We suppose

that the leaves of t_f are semialgebraic functions, and that the other nodes are either basic binary operations ($+$, \times , $-$, $/$), or unary transcendental functions (\sin , *etc*).

Our main algorithm `template_optim` (Figure 1) is based on a previous method of the authors [25], in which the objective function is bounded by means of semialgebraic functions. For the sake of completeness, we first recall the basic principles of this method.

Bounding the objective function by semialgebraic estimators. Given a function represented by an abstract tree t , semialgebraic lower and upper estimators t^- and t^+ are computed by induction. If the tree is reduced to a leaf, *i.e.* $t \in \mathcal{A}$, it suffices to set $t^- = t^+ := t$. If the root of the tree corresponds to a binary operation `bop` with children c_1 and c_2 , then the semialgebraic estimators c_1^- , c_1^+ and c_2^- , c_2^+ are composed using a function `compose_bop` to provide bounding estimators of t . Finally, if t corresponds to the composition of a transcendental (unary) function ϕ with a child c , we first bound c with semialgebraic functions c^+ and c^- . We compute a lower bound c_m of c^- as well as an upper bound c_M of c^+ to obtain an interval $I := [c_m, c_M]$ enclosing c . Then, we bound ϕ from above and below by computing parabola at given control points (function `build_par`), thanks to the semiconvexity properties of ϕ on the interval I . These parabola are composed with c^+ and c^- , thanks to a function denoted by `compose`.

These steps correspond to the part of the algorithm `template_optim` from Lines 1 to 10.

Reducing the complexity of semialgebraic estimators using templates. The semialgebraic estimators previously computed are used to determine lower and upper bounds of the function associated with the tree t , at each step of the induction. The bounds are obtained by calling the functions `min_sa` and `max_sa` respectively, which reduce the semialgebraic optimization problems to polynomial optimization problems by introducing extra lifting variables (see Section 2).

However, the complexity of solving the POPs can grow significantly because of the number n_{lifting} of lifting variables. If k denotes the relaxation order, the corresponding SDP problem Q_k indeed involve linear matrix inequalities of size $O((n + n_{\text{lifting}})^k)$ over $O((n + n_{\text{lifting}})^{2k})$ variables.

Consequently, this is crucial to control the number of lifting variables, or equivalently, the complexity of the semialgebraic estimators. For this purpose, we introduce the function `build_template`. It allows to compute approximations of the tree t by means of suprema/infima of quadratic functions, when the number of lifting variables exceeds a user-defined threshold value $n_{\text{lifting}}^{\text{max}}$. The algorithm is depicted in Figure 2. Using a heuristics, it first builds candidate quadratic forms q_j^- and q_j^+ approximating t at each control point \mathbf{x}_j (function `build_quadratic_form`, described below). Since each q_j^- does not necessarily underestimate the function t , we then determine the lower bound m_j^- of the semialgebraic function $t^- - q_j^-$, which ensures that $q_j^- + m_j^-$ is a quadratic lower-approximation of t . Similarly, the function $q_j^+ + M_j^+$ is an upper-approximation of t . The returned semialgebraic expressions $\max_{1 \leq j \leq r} \{q_j^- + m_j^-\}$

Input: tree t , box K , SDP relaxation order k , control points sequence $s = \{\mathbf{x}_1, \dots, \mathbf{x}_r\} \subset K$

Output: lower bound m , upper bound M , lower semialgebraic estimator t_2^- , upper semialgebraic estimator t_2^+

- 1: **if** $t \in \mathcal{A}$ **then**
- 2: $t^- := t, t^+ := t$
- 3: **else if** $\text{bop} := \text{root}(t)$ is a binary operation with children c_1 and c_2 **then**
- 4: $m_{c_i}, M_{c_i}, c_i^-, c_i^+ := \text{template_optim}(c_i, K, k, s)$ for $i \in \{1, 2\}$
- 5: $t^-, t^+ := \text{compose_bop}(c_1^-, c_1^+, c_2^-, c_2^+)$
- 6: **else if** $r := \text{root}(t) \in \mathcal{T}$ with child c **then**
- 7: $m_c, M_c, c^-, c^+ := \text{template_optim}(c, K, k, s)$
- 8: $\text{par}^-, \text{par}^+ := \text{build_par}(r, m_c, M_c, s)$
- 9: $t^-, t^+ := \text{compose}(\text{par}^-, \text{par}^+, c^-, c^+)$
- 10: **end**
- 11: $t_2^-, t_2^+ := \text{build_template}(t, K, k, s, t^-, t^+)$
- 12: **return** $\text{min_sa}(t_2^-, k), \text{max_sa}(t_2^+, k), t_2^-, t_2^+$

Fig. 1: `template_optim`

and $\min_{1 \leq j \leq r} \{q_j^+ + M_j^+\}$ now generate only one lifting variable (representing max or min).

Quadratic functions returned by `build_quadratic_form`(t, \mathbf{x}_j) are of the form:

$$q_{\mathbf{x}_j, \lambda} : \mathbf{x} \mapsto t(\mathbf{x}_j) + \mathcal{D}(t)(\mathbf{x}_j) (\mathbf{x} - \mathbf{x}_j) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_j)^T \mathcal{D}^2(t)(\mathbf{x}_j) (\mathbf{x} - \mathbf{x}_j) + \frac{1}{2} \lambda (\mathbf{x} - \mathbf{x}_j)^2$$

(we assume that t is twice differentiable) where λ is computed as follows. We sample the Hessian matrix difference $\mathcal{D}^2(t)(\mathbf{x}) - \mathcal{D}^2(t)(\mathbf{x}_j)$ over a finite set of random points $R \subset K$, and construct a matrix interval D enclosing all the entries of $(\mathcal{D}^2(t)(\mathbf{x}) - \mathcal{D}^2(t)(\mathbf{x}_j))$ for $\mathbf{x} \in R$. A lower bound λ^- of the minimal eigenvalue of D is obtained by applying a robust SDP method on interval matrix described by Calafiore and Dabbene in [29]. Similarly, we get an upper bound λ^+ of the maximal eigenvalue of D . The function `build_quadratic_form`(t, \mathbf{x}_j) then returns the two quadratic forms $q^- := q_{\mathbf{x}_j, \lambda^-}$ and $q^+ := q_{\mathbf{x}_j, \lambda^+}$.

Example 2 (Modified Schwefel Problem). We illustrate our method with the function f from Example 1 and the finite set of three control points $\{135, 251, 500\}$. For each $i = 1, \dots, n$, consider the sub-tree $\sin(\sqrt{x_i})$. First, we represent each sub-tree $\sqrt{x_i}$ by a lifting variable y_i and compute $a_1 := \sqrt{135}, a_2 := \sqrt{251}, a_3 := \sqrt{500}$. Then, we get the equations of $\text{par}_{a_1}^-, \text{par}_{a_2}^-$ and $\text{par}_{a_3}^-$ with `build_par`, which are three underestimators of the function \sin on the real interval $I := [1, \sqrt{500}]$. Similarly we obtain three overestimators $\text{par}_{a_1}^+, \text{par}_{a_2}^+$ and $\text{par}_{a_3}^+$. Finally, we obtain the underestimator $t_{1,i}^- := \max_{j \in \{1,2,3\}} \{\text{par}_{a_j}^-(y_i)\}$ and the overestimator $t_{1,i}^+ := \min_{j \in \{1,2,3\}} \{\text{par}_{a_j}^+(y_i)\}$. To solve the modified Schwefel problem, we

Input: tree t , box K , SDP relaxation order k , control points sequence $s = \{\mathbf{x}_1, \dots, \mathbf{x}_r\} \subset K$, lower/upper semialgebraic estimator t^-, t^+

- 1: **if** the number of lifting variables exceeds $n_{\text{lifting}}^{\max}$ **then**
- 2: **for** $\mathbf{x}_j \in s$ **do**
- 3: $q_j^-, q_j^+ := \text{build_quadratic_form}(t, \mathbf{x}_j)$
- 4: $m_j^- := \text{min_sa}(t_1^- - q_j^-, k)$ $\triangleright q_j^- + m_j^- \leq t^- \leq t$
- 5: $M_j^+ := \text{max_sa}(q_j^+ - t_1^+, k)$ $\triangleright q_j^+ + M_j^+ \geq t^+ \geq t$
- 6: **done**
- 7: **return** $\max_{1 \leq j \leq r} \{q_j^- + m_j^-\}, \min_{1 \leq j \leq r} \{q_j^+ + M_j^+\}$
- 8: **else**
- 9: **return** t^-, t^+
- 10: **end**

Fig. 2: build_template

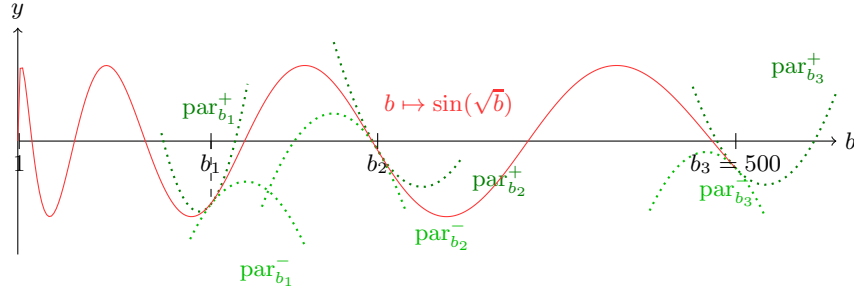


Fig. 3: Templates based on Max-plus Semialgebraic Estimators for $b \mapsto \sin(\sqrt{b})$: $t_{2,i}^- := \max_{j \in \{1,2,3\}} \{\text{par}_{b_j}^-(x_i)\} \leq \sin \sqrt{x_i} \leq t_{2,i}^+ := \min_{j \in \{1,2,3\}} \{\text{par}_{b_j}^+(x_i)\}$

consider the following POP:

$$\begin{cases} \min_{\mathbf{x} \in [1,500]^n, \mathbf{y} \in [1, \sqrt{500}]^n, \mathbf{z} \in [-1,1]^n} & - \sum_{i=1}^n (x_i + \epsilon x_{i+1}) z_i \\ \text{s.t.} & z_i \leq \text{par}_{a_j}^+(y_i), j \in \{1, 2, 3\}, i = 1, \dots, n \\ & y_i^2 = x_i, i = 1, \dots, n \end{cases}$$

Notice that the number of lifting variables is $2n$ and the number of equality constraints is n , thus we can obtain coarser semialgebraic approximations of f by considering the function $b \mapsto \sin(\sqrt{b})$ (see Figure 3). We get new estimators $t_{2,i}^-$ and $t_{2,i}^+$ of each sub-tree $\sin(\sqrt{x_i})$ with the functions `build_quadratic_form`, `min_sa` and `max_sa`. The resulting POP involves only n lifting variables. Besides, it does not contain equality constraints anymore, which improves in practice the numerical stability of the POP solver.

Dynamic choice of the control points. As in [25], the sequence s of control points is computed iteratively. We initialize the set s to a single point of K , chosen so as to be a minimizer candidate for t (e.g. with a local optimization solver).

Calling the algorithm `template_optim` on the main objective function t_f yields an underestimator t_f^- . Then, we compute a minimizer candidate \mathbf{x}_{opt} of the underestimator tree t_f^- . It is obtained by projecting a solution \mathbf{x}_{sdp} of the SDP relaxation of Section 2.1 on the coordinates representing the first order moments, following [11, Theorem 4.2]. We add \mathbf{x}_{opt} to the set of control points s . Consequently, we can refine dynamically our templates based max-plus approximations by iterating the previous procedure to get tighter lower bounds. This procedure can be stopped as soon as the requested lower bound is attained.

Remark 1 (Exploiting the system properties). Several properties of the POP can be exploited to decrease the size of the SDP relaxations such as symmetries [30] or sparsity [31]. Consider Problem (1.1) with f having some sparsity pattern or being invariant under the action of a finite subgroup symmetries. Then the same properties hold for the resulting semialgebraic relaxations that we build with our non-linear templates method.

4 RESULTS

Comparing three certification methods. We next present numerical results obtained by applying the present template method to examples from the global optimization literature, as well as inequalities from the Flyspeck project. Our tool is implemented in OCaml and interfaced with the SparsePOP solver [31].

In each example, our aim is to certify a lower bound m of a function f on a box K . We use the algorithm `template_optim`, keeping the SOS relaxation order k sufficiently small to ensure the fast computation of the lower bounds. The algorithm `template_optim` returns more precise bounds by successive updates of the control points sequence s . However, in some examples, the relaxation gap is too high to certify the requested bound. Then, we perform a domain subdivision in order to reduce this gap: we divide the maximal width interval of K in two halves to get two sub-boxes K_1 and K_2 such that $K = K_1 \cup K_2$. We repeat this subdivision procedure, by applying `template_optim` on a finite set of sub-boxes, until we succeed to certify that m is a lower bound of f . We denote by `#boxes` the total number of sub-boxes generated by the algorithm.

For the sake of comparison, we have implemented a template-free SOS method `ia_sos`, which coincides with the particular case of `template_optim` in which `#s` = 0 and `n_lifting` = 0. It computes the bounds of semialgebraic functions with standard SOS relaxations and bounds the univariate transcendental functions by interval arithmetic. We also tested the MATLAB toolbox algorithm `intsolver` [32], which is based on the Newton interval method [33]. Experiments are performed on an Intel Core i5 CPU (2.40 GHz).

Global optimization problems. The following test examples are taken from Appendix B in [17]. Some of these examples depend on numerical constants, the values of which can be found there.

- *Hartman 3 (H3)*: $\min_{\mathbf{x} \in [0,1]^3} f(\mathbf{x}) = -\sum_{i=1}^4 c_i \exp \left[-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right]$
- *Mc Cormick (MC)*, with $K = [-1.5, 4] \times [-3, 3]$:
 $\min_{\mathbf{x} \in K} f(\mathbf{x}) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 0.5x_1 + 2.5x_2 + 1$
- *Modified Langerman (ML)*:
 $\min_{\mathbf{x} \in [0,10]^n} f(\mathbf{x}) = \sum_{j=1}^5 c_j \cos(d_j/\pi) \exp(-\pi d_j)$, with $d_j = \sum_{i=1}^n (x_i - a_{ji})^2$
- *Paviani Problem (PP)*, with $K = [2.01, 9.99]^{10}$:
 $\min_{\mathbf{x} \in K} f(\mathbf{x}) = \sum_{i=1}^{10} [(\log(x_i - 2))^2 - \log(10 - x_i)]^2 - \left(\prod_{i=1}^{10} x_i \right)^{0.2}$
- *Shubert (SBT)*: $\min_{\mathbf{x} \in [-10,10]^n} f(\mathbf{x}) = \prod_{i=1}^n \left(\sum_{j=1}^5 j \cos((j+1)x_i + j) \right)$
- *Modified Schwefel (SWF)*: see Example 1

Informal certification of lower bounds of non-linear problems. In Table 1, the *time* column indicates the total informal verification time, *i.e.* without the exact certification of the lower bound m with Coq. Each occurrence of the symbol “–” means that m could not be determined within one day of computation by the corresponding solver. We see that `ia_sos` already outperforms the interval arithmetic solver `intsolver` on these examples. However, it can only be used for problems with a moderate number of variables. The algorithm `template_optim` allows us to overcome this restriction, while keeping a similar performance (or occasionally improving this performance) on moderate size examples.

Notice that reducing the number of lifting variables allows us to provide more quickly coarse bounds for large-scale instances of *SWF*. We discuss the results appearing in the two last lines of Table 1. Without any box subdivision, we can certify a better lower bound $m = -967n$ with $n_{\text{lifting}} = 2n$ since our semialgebraic estimator is more precise. However the last lower bound $m = -968n$ can be computed twice faster by considering only n lifting variables, thus reducing the size of the POP described in Example 2. This indicates that the method is able to avoid the blow up for certain hard sub-classes of problems where a standard (template free) POP formulation would involve a large number of lifting variables.

Formal certification of lower bounds of POP. For some small size instances of POP, our tool can prove the correctness of lower bounds. Our solver is interfaced with the framework mentioned in [5] to provide exact rational certificates, which can be formally checked with Coq. This formal verification is much slower. As an example, for the *MC* problem, it is 36 times slower to generate exact SOS certificates and 13 times slower to prove its correctness in Coq. Note that the interface with Coq still needs some streamlining.

High-degree polynomial approximations. An alternative approach consists in approximating the transcendental functions by polynomial functions of sufficiently

Table 1: Comparison results for global optimization examples

Problem	n	m	template_optim				ia_sos		intsolver	
			k	$\#s$	n_{lifting}	$\#boxes$	time	$\#boxes$	time	time
<i>H3</i>	3	-3.863	2	3	4	99	101 s	1096	247 s	3.73 h
<i>H6</i>	6	-3.33	2	1	6	113	102 s	113	45 s	> 4 h
<i>MC</i>	2	-1.92	1	2	1	17	1.8 s	92	7.6 s	4.4 s
<i>ML</i>	10	-0.966	1	1	6	8	8.2 s	8	6.6 s	> 4 h
<i>PP</i>	10	-46	1	3	2	135	89 s	3133	115 s	56 min
<i>SBT</i>	2	-190	2	3	2	150	36 s	258	0.6 s	57 s
<i>SWF</i> ($\epsilon = 0$)	10	-430 n	2	6	2 n	16	40 s	3830	129 s	18.5 min
	100	-440 n	2	6	2 n	274	1.9 h	> 20000	> 10 h	–
	1000	-486 n	2	4	2 n	1	450 s	–	–	–
	1000	-488 n	2	4	n	1	250 s	–	–	–
<i>SWF</i> ($\epsilon = 1$)	1000	-967 n	3	2	2 n	1	543 s	–	–	–
	1000	-968 n	3	2	n	1	272 s	–	–	–

Table 2: Results for Flyspeck inequalities using `template_optim` with $n = 6$, $k = 2$ and $m = 0$

Inequality id	$n\tau$	$\#s$	n_{lifting}	$\#boxes$	time
9922699028	1	4	9	47	241 s
9922699028	1	4	3	39	190 s
3318775219	1	2	9	338	26 min
7726998381	3	4	15	70	43 min
7394240696	3	2	15	351	1.8 h
4652969746_1	6	4	15	81	1.3 h
OXLZLEZ 6346351218_2_0	6	4	24	200	5.7 h

high degree, and then applying sums of squares approach to the polynomial problems. Given $d \in \mathbb{N}$ and a floating-point interval I , we can approximate an univariate transcendental function on I by the best uniform degree- d polynomial approximation and obtain an upper bound of the approximation error. This technique, based on Remez algorithm, is implemented in the Sollya tool (for further details, see e.g. [34]).

We interfaced our tool with Sollya and performed some numerical tests. The minimax approximation based method is eventually faster than the templates method for moderate instances. For the examples *H3* and *H6*, the speed-up factor is 8 when the function \exp is approximated by a quartic minimax polynomial.

However, this approach is much slower to compute lower bounds of problems involving a large number of variables. It requires 57 times more CPU time to solve *SWF* ($\epsilon = 1$) with $n = 10$ by considering a cubic minimax polynomial approximation of the function $b \mapsto \sin(\sqrt{b})$ on a floating-point interval $I \supseteq [1, \sqrt{500}]$. These experiments indicate that a high-degree polynomial approximation is not suitable for large-scale problems.

Certification of various Flyspeck inequalities. In Table 2, we present some test results for several non-linear Flyspeck inequalities. The information in the columns *time*, *#boxes*, and n_{lifting} is the same as above. The integer $n_{\mathcal{T}}$ represents the number of transcendental univariate nodes in the corresponding abstract syntax trees. These inequalities are known to be tight and involve sum of arctan of correlated functions in many variables, whence we keep high the number of lifting variables to get precise max-plus estimators. However, some inequalities (e.g. 9922699028) are easier to solve by using coarser semialgebraic estimators. For instance, the first line ($n_{\text{lifting}} = 9$) corresponds to the algorithm described in [25] and the second one ($n_{\text{lifting}} = 3$) illustrates our improved templates method. For the latter, we do not use any lifting variables to represent square roots of univariate functions.

5 CONCLUSION

The present quadratic templates method computes certified lower bounds for global optimization problems. It can provide tight max-plus semialgebraic estimators to certify non-linear inequalities involving transcendental multivariate functions (e.g. for Flyspeck inequalities). It also allows one to limit the growth of the number of lifting variables as well as of polynomial constraints to be handled in the POP relaxations, at the price of a coarser approximation. Thus, our method is helpful when the size of optimization problems increases. Indeed, the coarse lower bounds obtained (even with a low SDP relaxation order) are better than those obtained with interval arithmetic or high-degree polynomial approximation. For future work, we plan to study how to obtain more accurate non-linear templates by constructing a sequence of semialgebraic estimators, which converges to the “best” max-plus estimators (following the idea of [35]).

Furthermore, the formal part of our implementation, currently can only handle small size POP certificates. We plan to address this issue by a more careful implementation on the Coq side, but also by exploiting system properties of the problem (sparsity, symmetries) in order to reduce the size of the rational SOS certificates. Finally, it remains to complete the formal verification procedure by additionally proving in Coq the correctness of our semialgebraic estimators.

Acknowledgements

The authors thank the anonymous referees for helpful comments and suggestions to improve this paper.

References

1. Thomas C. Hales, John Harrison, Sean McLaughlin, Tobias Nipkow, Steven Obua, and Roland Zumkeller. A revision of the proof of the kepler conjecture. *Discrete & Computational Geometry*, 44(1):1–34, 2010.

2. Thomas C. Hales. Introduction to the flyspeck project. In Thierry Coquand, Henri Lombardi, and Marie-Françoise Roy, editors, *Mathematics, Algorithms, Proofs*, number 05021 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2006. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
3. Helfried Peyrl and Pablo A. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theor. Comput. Sci.*, 409(2):269–281, 2008.
4. Erich L. Kaltofen, Bin Li, Zhengfeng Yang, and Lihong Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *JSC*, 47(1):1–15, jan 2012. In memory of Wenda Wu (1929–2009).
5. David Monniaux and Pierre Corbineau. On the generation of Positivstellensatz witnesses in degenerate cases. In Marko Van Eekelen, Herman Geuvers, Julien Schmaltz, and Freek Wiedijk, editors, *Interactive Theorem Proving (ITP)*, volume 6898 of *Lecture Notes in Computer Science*, pages 249–264. Springer Verlag, August 2011.
6. Frédéric Besson. Fast reflexive arithmetic tactics the linear case and beyond. In *Proceedings of the 2006 international conference on Types for proofs and programs, TYPES’06*, pages 48–62, Berlin, Heidelberg, 2007. Springer-Verlag.
7. John Harrison. Verifying nonlinear real formulas via sums of squares. In Klaus Schneider and Jens Brandt, editors, *Proceedings of the 20th International Conference on Theorem Proving in Higher Order Logics, TPHOLs 2007*, volume 4732 of *Lecture Notes in Computer Science*, pages 102–118, Kaiserslautern, Germany, 2007. Springer-Verlag.
8. Behzad Akbarpour and Lawrence Charles Paulson. Metitarski: An automatic theorem prover for real-valued special functions. *J. Autom. Reason.*, 44(3):175–205, March 2010.
9. Alexey Solovyev and Thomas C. Hales. Formal verification of nonlinear inequalities with taylor interval approximations. *CoRR*, abs/1301.1702, 2013.
10. Sicun Gao, Jeremy Avigad, and Edmund M. Clarke. Delta-complete decision procedures for satisfiability over the reals. *CoRR*, abs/1204.3513, 2012.
11. Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.
12. Pablo A. Parrilo and Bernd Sturmfels. *Minimizing polynomial functions*, volume 60 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 83–99. Amer. Math. Soc., Providence, RI, 2003.
13. Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17:218–242, 2006.
14. Roland Zumkeller. *Rigorous Global Optimization*. PhD thesis, École Polytechnique, 2008.
15. Eldon R. Hansen. Sharpening interval computations. *Reliable Computing*, 12(1):21–34, 2006.
16. Coralia Cartis, Nicholas I. M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. part i: motivation, convergence and numerical results. *Math. Program.*, 127(2):245–295, 2011.
17. M. Montaz Ali, Charoenchai Khompatraporn, and Zelda B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *J. of Global Optimization*, 31(4):635–672, April 2005.

18. Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. Scalable analysis of linear systems using mathematical programming. In Radhia Cousot, editor, *Proc. of Verification, Model Checking and Abstract Interpretation (VMCAI)*, volume 3385, pages 21–47, Paris, France, January 2005. Springer Verlag.
19. A. Adje, S. Gaubert, and E. Goubault. Coupling policy iteration with semi-definite relaxation to compute accurate numerical invariants in static analysis. *Logical methods in computer science*, 8(1):1–32, 2012.
20. W. H. Fleming and W. M. McEneaney. A max-plus-based algorithm for a Hamilton-Jacobi-Bellman equation of nonlinear filtering. *SIAM J. Control Optim.*, 38(3):683–710, 2000.
21. M. Akian, S. Gaubert, and A. Lakhoua. The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis. *SIAM J. Control Optim.*, 47(2):817–848, 2008.
22. W. M. McEneaney, A. Deshpande, and S. Gaubert. Curse-of-complexity attenuation in the curse-of-dimensionality-free method for HJB PDEs. In *Proc. of the 2008 American Control Conference*, pages 4684–4690, Seattle, Washington, USA, June 2008.
23. W. M. McEneaney. A curse-of-dimensionality-free numerical method for solution of certain HJB PDEs. *SIAM J. Control Optim.*, 46(4):1239–1276, 2007.
24. Stéphane Gaubert, William M. McEneaney, and Zheng Qu. Curse of dimensionality reduction in max-plus based approximation methods: Theoretical estimates and improved pruning algorithms. In *CDC-ECC [36]*, pages 1054–1061.
25. Xavier Allamigeon, Stéphane Gaubert, Victor Magron, and Benjamin Werner. Certification of inequalities involving transcendental functions: combining sdp and max-plus approximation, 2013. To appear in the Proceedings of the European Control Conference, ECC’13, Zurich.
26. Jean B. Lasserre and Mihai Putinar. Positivity and optimization for semi-algebraic functions. *SIAM Journal on Optimization*, 20(6):3364–3383, 2010.
27. M. Yamashita, K. Fujisawa, K. Nakata, M. Nakata, M. Fukuda, K. Kobayashi, and K. Goto. A high-performance software package for semidefinite programs: Sdpa7. Technical report, Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan, 2010.
28. M. Akian, S. Gaubert, and V. N. Kolokoltsov. Set coverings and invertibility of functional galois connections. In G. L. Litvinov and V. P. Maslov, editors, *Idempotent Mathematics and Mathematical Physics*, Contemporary Mathematics, pages 19–51. American Mathematical Society, 2005.
29. G. Calafiore and F. Dabbene. Reduced vertex set result for interval semidefinite optimization problems. *Journal of Optimization Theory and Applications*, 139:17–33, 2008. 10.1007/s10957-008-9423-1.
30. Cordian Riemer, Thorsten Theobald, Lina Jansson Andrén, and Jean B. Lasserre. Exploiting symmetries in sdp-relaxations for polynomial optimization. *CoRR*, abs/1103.0486, 2011.
31. Hayato Waki, Sunyoung Kim, Masakazu Kojima, Masakazu Muramatsu, and Hiroshi Sugimoto. Algorithm 883: Sparsepop—a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Trans. Math. Softw.*, 35(2), 2008.
32. Tiago M. Montanher. Intsolver: An interval based toolbox for global optimization, 2009. Version 1.0, available from www.mathworks.com.
33. E.R. Hansen and R.I. Greenberg. An interval newton method. *Applied Mathematics and Computation*, 12(2-3):89 – 98, 1983.

34. Nicolas Brisebarre and Mioara Joldes. Chebyshev interpolation polynomial-based tools for rigorous computing. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, ISSAC '10, pages 147–154, New York, NY, USA, 2010. ACM.
35. Jean B. Lasserre and Tung Phan Thanh. Convex underestimators of polynomials. In *CDC-ECE* [36], pages 7194–7199.
36. *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC 2011, Orlando, FL, USA, December 12-15, 2011*. IEEE, 2010.