Computation of the $\mathcal{L}_\infty$-norm of Finite-Dimensional Linear Systems

Yacine Bouzidi[1], Alban Quadrat[2], Fabrice Rouillier[2], and Grace Younes[2(✉)]

[1] Inria Lille Europe, IMJ – PRG, Sorbonne University, Paris, France
Yacine.Bouzidi@inria.fr

[2] Inria Paris, Ouragan Project, IMJ – PRG, Sorbonne University, Paris, France
{alban.quadrat,fabrice.rouillier,grace.younes}@inria.fr

Abstract. In this paper, we study the problem of computing the $\mathcal{L}_\infty$-norm of finite-dimensional linear time-invariant systems. This problem is first reduced to the computation of the maximal x-projection of the real solutions (x, y) of a bivariate polynomial system $\Sigma = \{P, \frac{\partial P}{\partial y}\}$, with $P \in \mathbb{Z}[x, y]$. Then, we use standard computer algebra methods to solve the problem. In this paper, we alternatively study a method based on rational univariate representations, a method based on root separation, and finally a method first based on the sign variation of the leading coefficients of the signed subresultant sequence and then based on the identification of an isolating interval for the maximal x-projection of the real solutions of Σ.

Keywords: $\mathcal{L}_\infty$-norm computation · Real roots · Symbolic computation · Complexity computation · Implementation · Control theory

1 Introduction

An important issue in robust control theory is the computation of the $\mathcal{L}_\infty$-norm of linear systems [11,18]. Contrary to the $\mathcal{L}_2$-norm, no tractable formula is known for the characterization of the $\mathcal{L}_\infty$-norm of finite-dimensional systems (i.e., systems defined either by linear ordinary differential equations or by linear recurrence relations) [11,18]. Hence, the standard methods for the $\mathcal{L}_\infty$-norm computation are numerical (e.g., bisection algorithms, eigenvalues computation of Hamiltonian matrices) [5,7]. In their paper [13], Kano and Smith develop a validated numerical algorithm for the $\mathcal{L}_\infty$-norm computation. They reduce the problem to the localization of the real solutions of a bivariate polynomial and then use Sturm chain tests to guarantee the accuracy of their algorithm. In [8], Chen, Mazza and Xie provide an equivalent study using the theory of border polynomials, which makes the presentation of their solution simpler.

When numerical methods are used, it is worth mentioning that the result is usually obtained within a short time but with a slight error up to a precise accuracy. In contrast, when using symbolic methods, the result usually takes

R. M. Corless et al. (Eds.): MC 2020, CCIS 1414, pp. 119–136, 2021.
https://doi.org/10.1007/978-3-030-81698-8_9

more time to be computed but is exact. In this paper, to compute the $\mathcal{L}_\infty$-norm, we try to develop the right balance between these two approaches.

In this paper, following the approach developed in [8,13], we shall study three different certified symbolic-numeric algorithms for the computation of the $\mathcal{L}_\infty$-norm with the goal of minimizing the drawback of the symbolic part of the computation. This symbolic part consists in computing an isolating interval of the maximal projection of the real solutions of a system of bivariate polynomials. We develop the complexity analysis of each algorithm. Finally, we compare the theoretical complexities of the algorithms and then their performances using an implementation in the computer algebra system `Maple`.

Given two coprime polynomials P and Q in $\mathbb{Z}[x, y]$ of degree bounded by d and of coefficient bitsize bounded by τ, the solving of the system $\Sigma = \{P, Q\}$ can be studied using numerous methods. Typically, isolating boxes of the solutions can be computed either directly from the input system using numerical methods (such as *subdivision* or *homotopy methods*) or indirectly by first computing intermediate symbolic representations such as *triangular sets*, *Gröbner bases*, or *rational parameterizations* [1,6].

Two methods used in the paper require putting the system in a *generic position*, i.e., require to finding a separating linear form $x + a\,y$ that defines a shear of the coordinate system $(x,\, y)$, i.e., $(x,\, y) \longmapsto (t - a\,y,\, y)$, so that no two distinct solutions of the sheared system $\Sigma_a = \{P(t - a\,y, y), Q(t - a\,y, y)\}$ are vertically aligned. This approach has long been used in the literature. For instance, a separating linear form $x + a\,y$ with $a \in \{0, \ldots, 2\,d^4\}$ can be computed as shown in [3,4]. We can then use a *Rational Univariate Representation* (RUR) for the polynomial system Σ_a followed by the computation of isolating boxes for its real solutions. For more details, see [3]. We simply apply this approach (i.e., the so-called *RUR method*) to the polynomial system associated with the $\mathcal{L}_\infty$-norm computation problem and then choose the maximal x-projection of the real solutions of the system. The complexity analysis shows that this algorithm performs $\tilde{\mathcal{O}}_B(d_x\, d_y^3\, (d_x^2 + d_x\, d_y\, + d_y\, \tau))$ bit operations in the worst case, where

$$d_x = \max(\deg_x(P), \deg_x(Q)),\;\; d_y = \max(\deg_y(P), \deg_y(Q)), \tag{1}$$

and τ is the maximal coefficient bitsize of the polynomials P and Q.

Alternatively, we can also localize the maximal x-projection of the real solutions of the polynomial system Σ by simply applying a linear separating form on the system Σ. The linear separating form $t = x + s\,y$ proposed in [9] preserve the order of the solutions of the sheared system $\Sigma_s = \{P(t - s\,y, y), Q(t - s\,y, y)\}$ with respect to the x-projection of the real solutions of the original system Σ. Thus, the projection of the solutions of Σ_s onto the new separating axis t can be done so that we can simply choose the x-projection corresponding to the maximal t-projection of the real solutions of Σ_s. The drawback of this method lies on the growth of the size of the coefficients of the sheared system for the linear separating form $t = x + s\,y$ due to the large size of s. The complexity analysis shows that this algorithm performs $\tilde{\mathcal{O}}_B(d_x^4\, d_y^5\, \tau)$ bit operations in the worst case.

The third method developed in this paper localizes the maximal x-projection of the system real solutions − denoted by $\bar{x}$ − by first isolating the real roots of the univariate *resultant polynomial* $\mathrm{Res}(P, \frac{\partial P}{\partial y}, y)$ and then verifying the existence of a real root of the univariate polynomial $P(\bar{x}, y) \in \mathbb{R}[y]$ as done in [8]. A key point is that we can compute a *Sturm-Habicht sequence* [12] of $P(\bar{x}, y)$ without any consequent overhead. As $P(x, y) = 0$ is bounded in the x-direction, it is then possible to compute the number of real solutions of $P(\bar{x}, y)$ with a good complexity in the worst case, which gives an efficient algorithm as soon as the curve $P(x, y) = 0$ has no isolated real singular points. The complexity analysis shows that this algorithm performs $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\, (d_x + \tau))$ bit operations in the worst case and $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\, \tau)$ when the plane curve $P(x, y) = 0$ has no isolated real singular points.

Finally, we conclude the paper by comparing the bit complexity of those three algorithms and then the experimental time obtained by the implementation of each of these algorithms in `Maple`.

2 Problem Description

Before stating the problem studied in this paper, we first introduce a few standard notations and definitions. If $\Bbbk$ is a field and $P \in \Bbbk[x, y]$, then $Lc_{var}(P)$ is the *leading coefficient* of P with respect to the variable $var \in \{x, y\}$ and $\deg_{var}(P)$ the *degree* of P in the variable $var \in \{x, y\}$. We also denote by $\deg(P)$ the *total degree* of P. Moreover, let $\pi_x : \mathbb{R}^2 \longrightarrow \mathbb{R}$ be the projection map from the real plane $\mathbb{R}^2$ onto the x-axis, i.e., $\pi_x(x, y) = x$ for all $(x,\, y) \in \mathbb{R}^2$. For $P, Q \in \Bbbk[x, y]$, let $\gcd(P, Q)$ be the greatest common divisor of P and Q, $I := \langle P, Q \rangle$ the ideal of $\Bbbk[x, y]$ generated by P and Q, and $V_{\mathbb{K}}(I) := \{(x,\, y) \in \mathbb{K}^2 \mid \forall\, R \in I :\; R(x, y) = 0\}$, where $\mathbb{K}$ is a field containing $\Bbbk$. Finally, let $\mathbb{C}_+ := \{s \in \mathbb{C} \mid \mathrm{Re}(s) > 0\}$ be the *open right-half plane* of $\mathbb{C}$.

Definition 1 ([11,18]). *Let $\mathcal{RH}_\infty$ be the $\mathbb{R}$-algebra of all the proper and stable rational functions with real coefficients, namely:*

$$\mathcal{RH}_\infty := \left\{ \frac{n}{d} \mid n, d \in \mathbb{R}[s],\ \gcd(n, d) = 1,\ \deg_s(n) \le \deg_s(d), V_{\mathbb{C}}(\langle d \rangle) \cap \mathbb{C}_+ = \emptyset \right\}.$$

An element g of $\mathcal{RH}_\infty$ is holomorphic and bounded on $\mathbb{C}_+$, i.e.,

$$\| \, g \, \|_\infty := \sup_{s \in \mathbb{C}_+} |g(s)| < +\infty,$$

$\mathcal{RH}_\infty$ is a sub-algebra of the *Hardy algebra* $\mathcal{H}_\infty(\mathbb{C}_+)$ of bounded holomorphic functions on $\mathbb{C}_+$. The *maximum modulus principle* of complex analysis yields:

$$\| \, g \, \|_\infty = \sup_{\omega \in \mathbb{R}} |g(i\, \omega)|.$$

Note that this equality shows that the function $g_{|i\,\mathbb{R}} : i\,\omega \in i\,\mathbb{R} \longmapsto g(i\,\omega)$ belongs to the *Lebesgue space* $\mathcal{L}_\infty(i\,\mathbb{R})$ or, more precisely, to the following $\mathbb{R}$-algebra

$$\mathcal{RL}_\infty := \left\{ \frac{n(i\,\omega)}{d(i\,\omega)} \mid n, d \in \mathbb{R}[i\,\omega],\ \gcd(n, d) = 1,\ \deg_\omega(n) \le \deg_\omega(d),\ V_{\mathbb{R}}(\langle d \rangle) = \emptyset \right\},$$

i.e., the algebra of real rational functions on the imaginary axis $i\,\mathbb{R}$ which are proper and have no poles on $i\,\mathbb{R}$, or simply, the algebra of real rational functions with no poles on $i\,\mathbb{P}^1(\mathbb{R})$, where $\mathbb{P}^1(\mathbb{R}) := \mathbb{R} \cup \infty$.

We can extend the above $\mathcal{L}_\infty$-norms defined on functions of $\mathcal{RH}_\infty$ (resp., $\mathcal{RL}_\infty$) to matrices as follows. Let $G \in \mathcal{RH}_\infty^{u\times v}$ (resp., $G \in \mathcal{RL}_\infty^{u\times v}$, $\mathbb{R}(s)^{u\times v}$), i.e., G is a $u \times v$ matrix with entries in $\mathcal{RH}_\infty$ (resp., $\mathcal{RL}_\infty$, $\mathbb{R}(s)$) and let $\bar{\sigma}\,(\cdot)$ denote the largest *singular value of a complex matrix.* Then, we can define:

$$\parallel G \parallel_\infty := \sup_{s\in\mathbb{C}_+} \bar{\sigma}\,(G(s)) \,\left(\text{resp.,} \parallel G \parallel_\infty := \sup_{\omega\in\mathbb{R}} \bar{\sigma}\,(G(i\,\omega))\right).$$

If $G \in \mathcal{RH}_\infty^{u\times v}$, then, as above, we have $\parallel G \parallel_\infty = \sup_{\omega\in\mathbb{R}} \bar{\sigma}\,(G(i\,\omega))$.

The paper aims at developing certified symbolic-numeric algorithms for the computation of $\parallel G \parallel_\infty$ for $G \in \mathbb{R}(s)^{u\times v}$ satisfying $G_{|i\,\mathbb{R}} \in \mathcal{RL}_\infty^{u\times v}$. This problem plays a fundamental role in $\mathcal{H}_\infty$-control theory [11,18].

The *conjugate* $\tilde{G}$ of $G \in \mathbb{R}(s)^{u\times v}$ is defined by $\tilde{G}(s) := G^T(-s)$.

The next proposition gives a first characterization of $\parallel G \parallel_\infty$.

Proposition 1 **([13]).** *Let $\gamma > 0$, $G \in \mathbb{R}(s)^{u\times v}$ be such that $G_{|i\,\mathbb{R}} \in \mathcal{RL}_\infty^{u\times v}$ and let us consider $\Phi_\gamma(s) = \gamma^2\, I_v - \tilde{G}(s)\, G(s)$. Then, $\gamma > \parallel G \parallel_\infty$ if and only if $\gamma > \bar{\sigma}\,(G(i\,\infty))$ and $\det(\Phi_\gamma(i\,\omega)) \neq 0$ for all $\omega \in \mathbb{R}$.*

Let $n(\gamma, \omega)$ and $d(\omega)$ be two coprime polynomials over $\mathbb{R}[\gamma, \omega]$ satisfying:

$$\det(\Phi_\gamma(i\,\omega)) = \frac{n(\gamma,\omega)}{d(\omega)}. \tag{2}$$

Note that $\det(\Phi_\gamma(s))$ is a real function in s^2 and γ^2, and thus, $\det(\Phi_\gamma(i\,\omega))$ is a real function in ω^2 and γ^2. A consequence of Proposition 1 is the next result.

Proposition 2. *Let $G \in \mathcal{RL}_\infty^{u\times v}$ and $n \in \mathbb{R}[\gamma, \omega]$ be defined by (2). We denote by $\bar{n} \in \mathbb{R}[\gamma, \omega]$ the square free part of n. Then, we have:*

$$\parallel G \parallel_\infty = \max\left\{\pi_\gamma\left(V_\mathbb{R}\left(\left\langle \bar{n}, \frac{\partial \bar{n}}{\partial \omega}\right\rangle\right)\right) \cup V_\mathbb{R}\left(\langle Lc_\omega(\bar{n})\rangle\right)\right\}.$$

Example 1. If $G \in \mathcal{RL}_\infty$ then, by definition, $\parallel G \parallel_\infty$ is the supremum of the continuous function $\omega \in \mathbb{P}^1(\mathbb{R}) := \mathbb{R} \cup \{\infty\} \longmapsto |G(i\,\omega)|$, and thus, we have $\parallel G \parallel_\infty = \max_{\omega\in\mathbb{P}^1(\mathbb{R})} |G(i\,\omega)|$, i.e., $\parallel G \parallel_\infty = \max\{|G(i\,\infty)|, \gamma_{\max}\}$, where:

$$\gamma_{\max} := \max_{\omega\in\mathbb{R}} |G(i\,\omega)| = \max\left\{\gamma \in \mathbb{R} \mid \exists\, \omega \in \mathbb{R} : \gamma^2 = |G(i\,\omega)|^2\right\}.$$

We find again Proposition 2, i.e., $\gamma > \parallel G \parallel_\infty$ iff $\Phi_\gamma(i\,\omega) = \gamma^2 - |G(i\,\omega)|^2 \neq 0$ for all $\omega \in \mathbb{R}$ and $\gamma > |G(i\,\infty)|$. Using $|G(i\,\omega)|^2 = G(-i\,\omega)\, G(i\,\omega) \in \mathbb{R}(\omega^2)$, a computation of $\parallel G \parallel_\infty$ amounts to first computing the zeros of the numerator of $\frac{\mathrm{d}|G(i\,\omega)|^2}{\mathrm{d}\omega}$, then evaluating $|G(i\,\omega)|$ on these zeros and finally choosing the maximal occurring value, that to say $\bar{\gamma}$, and (iii) $\parallel G \parallel_\infty = \max\{|G(i\,\infty)|, \bar{\gamma}\}$.

More explicitly, if we write G as $G(s) = a(s)/b(s)$, where $a, b \in \mathbb{R}[s]$, $\gcd(a, b) = 1$, $q = \deg_s(a) \leq r = \deg_s(b)$, and b does not vanish on $i\,\mathbb{R}$, then $G(i\,\infty) = 0$ if $q < r$ (i.e., G is strictly proper) or $G(i\,\infty) = a_r/b_r$ if $q = r$ (i.e., G is proper), where $a_r = Lc_s(a)$ and $b_r = Lc_s(b)$. Moreover, we have $|G(i\,\omega)|^2 = N(\omega)/D(\omega)$, where $N(\omega) = |a(i\,\omega)|^2$ and $D(\omega) = |b(i\,\omega)|^2 \in \mathbb{R}[\omega^2]$. Since $b(i\,\omega)$ has not real roots, $D(\omega) \neq 0$ for all $\omega \in \mathbb{R}$. Hence, if we note $\mathcal{Z} := \{\omega \in \mathbb{R} \mid N'(\omega)\,D(\omega) - N(\omega)\,D'(\omega) = 0\}$, then we obtain:

$$\| G \|_\infty = \max\{|G(i\,\infty)|, \bar{\gamma}\}, \quad \bar{\gamma} := \max_{\omega \in \mathcal{Z}} \left\{ (N(\omega)/D(\omega))^{1/2} \right\}.$$

Note that if $\mathcal{Z} \cap V_{\mathbb{R}}(\langle D'(\omega)\rangle) = V_{\mathbb{R}}(\langle N'(\omega)\,D(\omega), D'(\omega)\rangle) = \emptyset$, then we also have $\bar{\gamma} = \max_{\omega \in \mathcal{Z}} \left\{ (N'(\omega)/D'(\omega))^{1/2} \right\}$. For instance, if $G(s) = (2\,s + 1)/(s + 1)$, then $N(\omega) = 4\,\omega^2 + 1$, $D(\omega) = \omega^2 + 1$, $\mathcal{Z} = \{0\}, \mathcal{Z} \cap V_{\mathbb{R}}(\langle D'(\omega)\rangle) = \{0\}$, $\bar{\gamma} = (N(0)/D(0))^{1/2} = 1$, $|G(i\,\infty)| = 2$, and $\| G \|_\infty = \max\{2, \bar{\gamma}\} = 2$.

Finally, according to Proposition 2, we have $n(\gamma, \omega) = D(\omega)\,\gamma^2 - N(\omega)$ and $d(\omega) = D(\omega)$. Now, using $\gcd(N, D) = 1$, $\bar{n} = n$, $Lc_\omega(\bar{n}) = b_r^2\,\gamma^2$ if $q < r$ or $Lc_\omega(\bar{n}) = (b_r^2\,\gamma^2 - a_r^2)$ if $q = r$, which yields:

$$V_{\mathbb{R}}\left(\langle Lc_\omega(\bar{n})\rangle\right) = \begin{cases} 0, & \text{if } q < r, \\ \pm\dfrac{a_r}{b_r}, & \text{if } q = r, \end{cases}$$

and using the fact that $D(\omega) \neq 0$ for all $\omega \in \mathbb{R}$, we have

$$\begin{aligned} \pi_\gamma\left(V_{\mathbb{R}}\left(\left\langle \bar{n}, \frac{\partial \bar{n}}{\partial \omega}\right\rangle\right)\right) &= \pi_\gamma\left(V_{\mathbb{R}}(\langle D(\omega)\,\gamma^2 - N(\omega), D'(\omega)\,\gamma^2 - N'(\omega)\rangle)\right) \\ &= \left\{\gamma \in \mathbb{R} \mid \gamma^2 = \frac{N(\omega)}{D(\omega)},\ \omega \in \mathcal{Z}\right\}, \end{aligned}$$

and thus, $\| G \|_\infty = \max\{\bar{\gamma}, 0\} = \bar{\gamma}$ if $q < r$ and $\| G \|_\infty = \max\{\bar{\gamma}, a_r/b_r\}$ if $q = r$.

Corollary 1. *Let $G \in \mathcal{RL}_\infty^{u \times v}$ and $n \in \mathbb{R}[\gamma, \omega]$ be the numerator of* $\det(\gamma^2\,I_v - \tilde{G}(i\,\omega)\,G(i\,\omega))$ *defined by (2). Then, the real γ-projection $\pi_\gamma(V_{\mathbb{R}}(\langle n\rangle))$ of $V_{\mathbb{R}}(\langle n\rangle)$ is bounded by $\| G \|_\infty$.*

According to Proposition 2, given $G \in \mathcal{RL}_\infty^{u \times v}$, the problem of computing $\| G \|_\infty$ can be reduced to the computation of the maximal γ-projection of the real solutions of the following bivariate polynomial system:

$$\Sigma := \left\{\bar{n}(\gamma, \omega), \frac{\partial \bar{n}(\gamma, \omega)}{\partial \omega}\right\}. \tag{3}$$

For studying this problem, we propose three different symbolic-numeric methods − *Rational Univariate Representation method*, *Roots Separation Method* and *Sturm-Habicht method* − and compare them. Without loss of generality, we shall suppose that n is squarefree in $\mathbb{Z}[\gamma, \omega]$.

3 Rational Univariate Representation Method

In this section, we briefly state a straightforward algorithm which computes the maximal γ-projection of the real solutions of (3) based on a *Rational Univariate Representation Method* [3,4,15,16]. This algorithm consists in first computing a rational parametrization (RUR) of the solutions of (3), then isolating the roots of a univariate polynomial p defining the associated field extension, using the intervals obtained to compute isolating boxes for the real solutions of (3) and finally selecting the real solution of (3) with the maximal γ-projection.

If $P,\, Q \in \mathbb{Q}[x,y]$ are two coprime polynomials, then the computation of the RUR of $V_{\mathbb{K}}(\langle P,\, Q\rangle)$, $\mathbb{K} = \mathbb{R}, \mathbb{C}$, consists in finding $s \in \mathbb{N}$ such that $x + s\,y$ separates the $\mathbb{K}$-zeros of $\{P,\, Q\}$ and four polynomials $p,\, q,\, p_0,\, q_0 \in \mathbb{Q}[T]$ which define a 1-1 correspondence between $V_{\mathbb{K}}(\langle \Sigma\rangle)$ and $V_{\mathbb{K}}(\langle p\rangle)$, i.e., the following bijection:

$$\begin{aligned} V_{\mathbb{K}}(\langle P,\, Q\rangle) &\longrightarrow V_{\mathbb{K}}(\langle p\rangle) \\ (x,\, y) &\longmapsto \xi = x + s\,y, \\ \left(\frac{p_0(\xi)}{q(\xi)}, \frac{p_1(\xi)}{q(\xi)}\right) &\longleftarrow \xi \end{aligned}$$

Roughly speaking, using the RUR of $V_{\mathbb{K}}(\langle P,\, Q\rangle)$, we can transform the study of problems on $V_{\mathbb{K}}(\langle P,\, Q\rangle)$ into corresponding problems on $V_{\mathbb{K}}(\langle p\rangle)$ [3,15].

For the $\mathcal{L}_\infty$-norm computation, the polynomials of $\Sigma \subset \mathbb{Z}[\gamma,\omega]$, defined by (3), are coprime. Hence, to compute $\parallel G \parallel_\infty$, we first use the RUR method to obtain isolating boxes for the real solutions $(\gamma,\, \omega)$ of Σ, choose the maximal γ-projection γ_1, then compute an isolation box γ_2 for the maximal real root of the univariate polynomial $Lc_\omega(n)$, and finally compute $\parallel G \parallel_\infty = \max\{\gamma_1, \gamma_2\}$.

Algorithm 1. RUR method

Input: A zero dimensional polynomial system $\{n, \frac{\partial n}{\partial \omega}\} \subset \mathbb{Z}[\gamma,\omega]$.

Output: An isolating interval of $\max\left\{\pi_\gamma\left(V_{\mathbb{R}}\left(n, \frac{\partial n}{\partial \omega}\right)\right) \cup V_{\mathbb{R}}\left(Lc_\omega(n)\right)\right\}$.

1. Apply the RUR function (`Isolate`) for solving the polynomial system $\{n, \frac{\partial n}{\partial \omega}\}$.
2. Let γ_1 be the maximal γ-projection of the system's real solutions.
3. Let γ_2 be the maximal real root of $Lc_\omega(n)$.
4. **Return** the isolating interval of $\max\{\gamma_1, \gamma_2\}$.

Remark 1. *In step* 1 *of Algorithm 1, we obtain isolating boxes* $[a_i, b_i] \times [c_i, d_i]$ *of the system's real solutions* $(\gamma_i,\, \omega_i)$. *To compare the real values in step* 2 *and then in step* 4*, we can apply a straightforward strategy consisting in computing the resultant polynomial* $R_\gamma = \mathrm{Res}(n, \frac{\partial n}{\partial \omega}, \omega)$ *and then refining the boxes until each interval* $[a_i, b_i]$ *is included in an isolating interval of* R_γ. *Even with this naive approach, the asymptotic complexity of these operations does not exceed the algorithm's overall worst case bit complexity.*

In what follows, $\mathcal{O}_B$ denotes the *bit complexity* and $\tilde{\mathcal{O}}_B$ means that logarithmic factors have been omitted. Given two coprime polynomials P, $Q \in \mathbb{Z}[x, y]$ of degree bounded by d and coefficient bitsize bounded by τ, an algorithm for computing linear separating forms, RUR representations and isolating boxes of the solutions can be obtained in the worst case bit complexity $\tilde{\mathcal{O}}_B(d^6 + d^5\,\tau)$ [3].

Let us compute the complexity of Algorithm 1. We first need the next result.

Lemma 1. *Let $G \in \mathcal{RL}_\infty^{u\times v}$, $\Phi_\gamma(i\,\omega) = \gamma^2\, I_v - \tilde{G}(i\,\omega)\, G(i\,\omega)$, $n \in \mathbb{R}[\gamma, \omega]$ be defined by (2), $d_\gamma = \deg_\gamma(n)$, $d_\omega = \deg_\omega(n)$, and τ_n the coefficient bitsize of n. Moreover, let $\alpha = \max\{u, v\}$, $N = \max_{1\le i\le u, 1\le j\le v}\{\deg_\omega(Q_{i,j})\}$, where $G_{jk} = \frac{P_{jk}}{Q_{jk}}$ denotes the $(j, k)^{th}$ entry of G and P_{jk}, $Q_{jk} \in \mathbb{R}[i\,\omega]$ are coprime, and τ_G the maximal coefficient bitsize of $\{P_{jk}, Q_{jk}\}_{1\le j\le u, 1\le k\le v}$. Then, we have:*

$$d_\gamma = \mathcal{O}(\alpha), \quad d_\omega = \mathcal{O}(N\,\alpha^2), \quad \tau_n = \tilde{\mathcal{O}}(\tau_G\,\alpha^2).$$

Proof. Let $G_{jk} = P_{jk}/Q_{jk}$ be the $(j, k)^{\text{th}}$ entry of G, where P_{jk}, $Q_{jk} \in \mathbb{R}[i\,\omega]$ are coprime. Since $G \in \mathcal{RL}_\infty^{u\times v}$, G_{jk} is a proper rational function, and thus, $\deg_\omega(P_{jk}) \le \deg_\omega(Q_{jk}) \le N$, which shows that the degrees in ω of the numerators and the denominator of the entries of $\Phi_\gamma(i\,\omega)$ are bounded by $2\,N\,\alpha$, and thus, d_ω is bounded by $2\,N\,\alpha^2$. Similarly, the maximal coefficient bitsize of the entries of $\Phi_\gamma(i\,\omega)$ is $2\,\tau_n\,\alpha$, which yields τ_n is bounded by $2\,\tau_G\,\alpha^2$. Finally, d_γ is clearly bounded by $2\,\alpha$.

Theorem 1. *With the above notations, the complexity of Algorithm 1 for the computation of $\|\, G\,\|_\infty$, where $G \in \mathcal{RL}_\infty^{u\times v}$, is given by:*

$$\tilde{\mathcal{O}}_B(d_\gamma\, d_\omega^3\,(d_\gamma^2 + d_\gamma\, d_\omega + d_\omega\,\tau_n)) = \tilde{\mathcal{O}}_B(\alpha^9\, N^4\,(\alpha + \tau_n)).$$

Proof. According to [3], using the RUR method, the complexity of the resolution of a zero dimensional bivariate polynomial system comes first from the computation of the triangular decomposition of the system after shearing − using a separating linear form $\gamma + s\,\omega$ − then from the root isolation of the univariate polynomial defining the associated field extension, and finally from the computation of the isolating boxes for the solutions.

In the present case, the degrees in ω and γ are not of the same order. Hence, the results of [3] must be adapted.

First, we determine the size and the degree of the sheared system up to the method used in [3]: the degree with respect to the variable ω is $\tilde{\mathcal{O}}(d_\gamma + d_\omega)$ and d_γ with respect to the variable $t = \gamma + s\,\omega$. The size of the sheared system is $\tilde{\mathcal{O}}(\tau_n + d_\gamma)$. From [14], the complexity of the computation of a triangular decomposition of a system over $\mathbb{Z}[x, y]$ costs $\tilde{\mathcal{O}}_B(d_x^3\, d_y^3 + (d_x^2\, d_y^3 + d_x\, d_y^4)\,\tilde{\tau})$, where d_x and d_y are defined by (1) and $\tilde{\tau}$ is its maximal coefficient bitsize. Thus, using Lemma 1, we obtain that the complexity of the computation of the triangular decomposition of the sheared system in $\mathbb{Z}[t, \omega]$ is given by:

$$\begin{aligned}&\tilde{\mathcal{O}}_B(d_\gamma^3\,(d_\gamma + d_\omega)^3 + (d_\gamma^2\,(d_\gamma + d_\omega)^3 + d_\gamma\,(d_\gamma + d_\omega)^4)\,(\tau_n + d_\gamma))\\ &= \tilde{\mathcal{O}}_B(d_\gamma\, d_\omega^3\,(d_\gamma^2 + d_\gamma\, d_\omega + d_\omega\,\tau_n)).\end{aligned}$$

Moreover, by Lemma 1, we obtain $\tilde{\mathcal{O}}_B(\alpha^9\, N^4\, (\alpha\, +\, \tau_n))$. This triangular decomposition yields (RUR) polynomials of degree that sum up to $\tilde{\mathcal{O}}((d_\gamma + d_\omega)\, d_\gamma) = \tilde{\mathcal{O}}(\alpha^3 N)$ with coefficients of bitsize $\tilde{\mathcal{O}}((d_\gamma + d_\omega)(d_\gamma + \tau_n)) = \tilde{\mathcal{O}}(\alpha^2 N(\alpha + \tau_n))$.

Finally, according to [3], it is known that the computation of isolating boxes of all the roots of the system can be done in $\tilde{\mathcal{O}}_B((\alpha^3\, N)^3 + (\alpha^3\, N)^2(\alpha^2\, N\, (\alpha + \tau_n))) = \tilde{\mathcal{O}}_B(\alpha^8\, N^3\, (\alpha\, +\, \tau_n))$ bit operations.

4 Roots Separation Method

In this section, we localize the maximal γ-projection of the real solutions of the polynomial system Σ by only shearing the system Σ using a special linear separating form [9]. With this linear separating form $t = \gamma + s\,\omega$, we obtain:

$$t_1 = \gamma_1 + s\,\omega_1 \;<\; t_2 = \gamma_2 + s\,\omega_2 \implies \gamma_1 \le \gamma_2.$$

Let P, $Q \in \mathbb{Z}[x, y]$ be coprime and $R_x = \mathrm{Res}(P, Q, y) \in \mathbb{Z}[x]$ be their *resultant*. Moreover, let $x_1 \le \ldots \le x_m$ be the real roots of R_x with isolating intervals $[c_1, d_1], \ldots, [c_m, d_m]$. Moreover, let the real numbers δ, M and s be defined by:

$$\delta < \frac{1}{2} \min_{i=1,\ldots,m-1}(x_{i+1} - x_i),\; M > \max\{y \mid (x, y) \in V_{\mathbb{R}}(\langle P, Q\rangle)\},\; 0 < s < \frac{\delta}{M}. \tag{4}$$

We can use the general root bounds for zero dimensional systems to estimate δ and M. In fact, M is the measure of the univariate polynomial $\mathrm{Res}(P, Q, x)$. Note that the resultant computation can be avoided by using the concept of *sleeve functions* studied in [10] and [9, Lemma 3.3].

Let us consider an invertible linear map (a shear) of $\mathbb{R}^2$ to $\mathbb{R}^2$ defined by $\Psi_s : (x,\, y) \longmapsto (t,\, y) = (x + s\,y,\, y)$. Let us also note $\Psi_s(P) = P(t - s\,y,\, y)$, $\Psi_s(Q) = Q(t - s\,y,\, y)$, $R_t = \mathrm{Res}(\Psi_s(P), \Psi_s(Q), y)$ and let $t_1 \le \ldots \le t_{m'} = t_{\max}$ be the real roots of R_t.

To get a 1-1 correspondence between the zeros of $\{P, Q\}$ and the roots of R_t, $Lc_y(\Psi_s(P))$ and $Lc_y(\Psi_s(Q))$ must not both vanish. It is always possible to choose s such that this condition is satisfied. In what follows, we shall consider this case. For more details for the computation of s up to this condition, see [9].

Remark 2. In Fig. 1, we only draw a part of the plot. In fact, since $n \in \mathbb{Z}[\gamma^2, \omega^2]$, $\Sigma = 0$ is symmetric with respect to the γ and ω axes.

Proposition 3. *With the above notations, let x_m be a real root of R_x with an isolating interval $[c_m, d_m]$ and $t_{\max}$ the maximal real root of R_t. If $t_{\max} \in [c_m - \delta, d_m + \delta]$, then the maximal x-projection of $V_{\mathbb{R}}(\langle P, Q\rangle)$ is equal to x_m.*

Proof. For each real root x_i of R_x with an isolating interval $[c_i, d_i]$, let us denote by $P_{i,j} = (x_i,\, y_{i,j})$ the real solutions of $\{P, Q\}$ which project onto x_i. Then, $\Psi_s(P_{i,j}) = (x_i + s\,y_{i,j},\, y_{i,j})$, where $x_i + s\,y_{i,j}$ is the first coordinate of a real solution of $\{\Psi_s(P), \Psi_s(Q)\}$. Using (4), we obtain that different $\Psi_s(P_{i,j})$ have

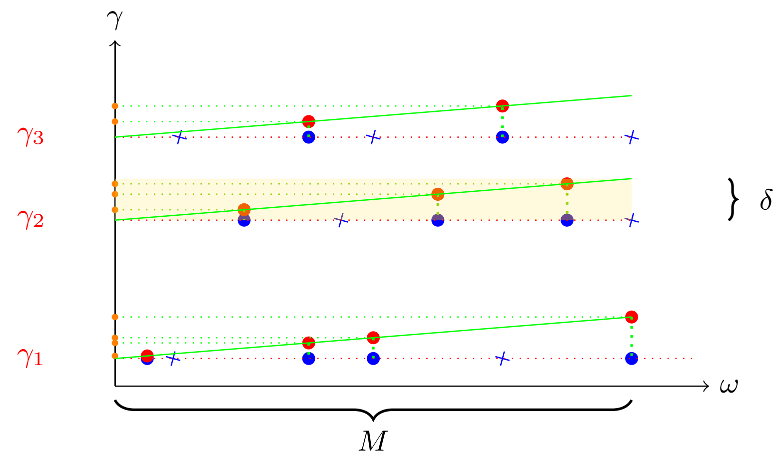

Fig. 1. The blue dots represent the real solutions of $\Sigma = 0$ and the blue crosses are the complex ones; the red dots are the solutions of $\Psi_s(\Sigma) = \{\Psi_s(n), \Psi_s(\frac{\partial n}{\partial \omega})\}$; the orange dots on the γ-axis are the roots of univariate polynomial R_t. (Color figure online)

different first coordinates, and thus, $\{\Psi_s(P, \Psi_s(Q)\}$ is in a *generic position*. Furthermore, we have $|x_i + s\,y_{i,j} - x_i| = |s\,y_{i,j}| < (\frac{\delta}{M})\,M = \delta$. Consequently $\Psi_s(P_{i,j}) \in I_i = [x_i - \delta, x_i + \delta] \times [-M, M]$. In addition, since $\delta < \frac{1}{2}(x_{i+1} - x_i)$, I_i are disjoint for different i. Hence, a real solution $(x,\, y)$ of $\{P,\, Q\}$ is mapped to $(\eta,\, y)$, where $\eta \in [x - \delta, x + \delta]$. Now, since $x_i \in [c_i, d_i]$, the real roots of R_t associated with x_i are in the interval $[c_i - \delta, d_i + \delta]$.

Algorithm 2. Roots Separation method

Input: A zero dimensional system $\{P, Q\} \subset \mathbb{Z}[x, y]$, where $Q = \frac{\partial P}{\partial y}$.
Output: An isolating interval of max $\{\pi_x\,(V_{\mathbb{R}}\,(\langle P, Q\rangle)) \cup V_{\mathbb{R}}\,(\langle Lc_y(P)\rangle)\}$

1. Isolate $R_x = \mathrm{Res}(P, Q, y)$ up to an accuracy ϵ and let $RI := \{[c_1, d_1], \ldots, [c_m, d_m]\}$ be the isolating intervals of the real roots $\{x_1, \ldots, x_m\}$ of R_x.
2. Compute M and $D = \frac{1}{2}\min_{i=1,\ldots,m-1} |c_{i+1} - d_i|$:
   - if $D > 2\,\epsilon$, let $\epsilon_1 = \epsilon$, $\delta = D - \epsilon_1$ and compute s up to the required conditions.
   - elif $D \leq 2\,\epsilon$, let $\epsilon_1 = D/2$, $\delta = D - \epsilon_1$ and compute s up to the required conditions ;
3. Expand $\{\Psi_s(P), \Psi_s(Q)\}$ and compute $R_t = \mathrm{Res}(\Psi_s(P), \Psi_s(Q), y)$.
4. Isolate R_t up to an accuracy less than ϵ_1 and set $[p_t, q_t]$ to be the isolating interval of its maximal real root t_{max}.
5. for j from 1 to m do:
   - if $[p_t, q_t] \subset [c_j - D, d_j + D]$, then $X_1 = x_j$.
6. Let X_2 be the maximal real root of $Lc_y(P)$.
7. **Return** the isolating interval of $\max\{X_1, X_2\}$.

Lemma 2. *Let $P \in \mathbb{Z}[x,y]$, $d_x = \deg_x(P)$, $d_y = \deg_y(P)$ and τ be the maximal coefficient bitsize of P. The sheared polynomial $P(t - s\,y, y)$ satisfies $\deg_y(P(t-s\,y,y)) = d_x + d_y$, $\deg_t(P(t-s\,y,y)) = d_x$, and it can be expanded in $\tilde{\mathcal{O}}_B(d_x\,d_y^2\,(\tau + d_x\,(1+\tau_s))$. The maximal bitsize of the coefficients of $P(t-s\,y,y)$ is equal to $\tilde{\mathcal{O}}(\tau + d_x\,(1+\tau_s))$, where τ_s denotes the bitsize of s.*

Proof. The proof is a direct consequence of the proof of [4, Lemma 7] by taking into account the bitsize τ_s of s.

Theorem 2. *We consider a zero dimensional system $\{P,Q\} \subset \mathbb{Z}[x,y]$, where $d_x = \max(\deg_x(P), \deg_x(Q))$, $d_y = \max(\deg_y(P), \deg_y(Q))$ and τ the maximal coefficients bitsize of the polynomials. We can compute an isolating interval for the maximal x-projection of the real solutions of $\{P,Q\}$ in $\tilde{\mathcal{O}}_B(d_x^4\,d_y^5\,\tau)$ bit operations, using Algorithm 2.*

Proof. For each root t_j of R_t defined in Algorithm 2 with an isolating interval $[p_j,q_j]$, there exists a unique $i \in \{1,\ldots,m\}$ such that $[p_j,q_j] \subset [c_i - D, d_i + D]$: we know that there exists a unique $[c_i,d_i]$ such that $t_j \in [c_i - \delta, d_i + \delta]$. From step 4, $q_j - p_j < \epsilon_1$ and $D = \delta + \epsilon_1$. Hence, $q_j < t_j + \epsilon_1 < d_i + \delta + \epsilon_1 < d_i + D$. And similarly $p_j > c_i - D$. Consequently, based on Proposition 3, Algorithm 2 outputs an isolating interval for the maximal x-projection of the real solutions of $\{P,Q\}$. As for the complexity, step 1 has worst-case bit complexity $\tilde{\mathcal{O}}_B(d_x\,d_y^3\,\tau)$ based on [2, Proposition 8.46]. Step 2 is of worst case bit complexity $\tilde{\mathcal{O}}_B(d^3 + d^2\,\tilde{\tau})$, where $d = \deg(R_x)$ and the coefficient size of R_x is equal to $\tilde{\tau}$ [3, Lemma 54]. From [2, Proposition 8.46], $d = \mathcal{O}(d_x\,d_y)$ and $\tilde{\tau} = \tilde{\mathcal{O}}(d_y\,\tau)$. Consequently, step 2 is of worst case bit complexity $\tilde{\mathcal{O}}_B(d_x^2\,d_y^3\,(d_x + \tau)$. In steps 3 and 4, we get $\delta = 2^{-\tilde{\mathcal{O}}(d_x\,d_y^2\,\tau)}$ and $M = 2^{\mathcal{O}(d_x\,\tau)}$ [3]. The bitsize of s is then equal to $\tilde{\mathcal{O}}(d_x\,d_y^2\,\tau)$. Consequently, the coefficient bitsize of the sheared system is $\tilde{\mathcal{O}}(d_x^2\,d_y^2\,\tau)$, and the worst case bit complexity of step 5 is $\tilde{\mathcal{O}}(d_x^3\,d_y^2\,(d_x+d_y)^3\,\tau)$, as computed in Lemma 2. In step 6, we isolate the resultant of the sheared system. Considering the size and degree of the sheared polynomials computed using Lemma 2, the size and degree of the resultant of the sheared system are $\tilde{\mathcal{O}}(d_x^2\,d_y^3\,\tau)$ and $\tilde{\mathcal{O}}(d_x\,(d_x+d_y))$ respectively. Then, knowing the complexity of the isolation mentioned in [3], we can say that the worst case bit complexity in this line is equal to $\tilde{\mathcal{O}}_B((d_x\,d_y)^3 + (d_x\,d_y^2)^2\,(d_x^2\,d_y^3\,\tau)) = \tilde{\mathcal{O}}_B(d_x^4\,d_y^5\,\tau)$. Finally, in the step 7, we simply compare two rational numbers. The maximal coefficients bitsize of these rationals is in $\tilde{\mathcal{O}}(d_x^3\,d_y^3\,(d_x+d_y)\,\tau)$ and the computation in this step is done in $\tilde{\mathcal{O}}(d_x^3\,d_y^3\,(d_x+d_y)\,\tau)$ bit operations. Hence, the overall bit complexity is given by $\tilde{\mathcal{O}}_B(d_x^4\,d_y^5\,\tau)$.

Corollary 2. *With the notations of Lemma 1, the worst case bit complexity for the computation of $\| G \|_\infty$ with the separation method (Algorithm 2) is given by $\tilde{\mathcal{O}}_B(\alpha^{14}\,N^5\,\tau_n)$.*

From this section, we can conclude that trying to concentrate only on the solution with the maximal γ-projection, after putting the system in a generic position, costs much more than computing isolating boxes for all the real solutions, due to the large size of the separating bound that we must use. Hence, in the next section, using another strategy than shearing the system, we shall try to find the maximal γ-projection of the polynomial solutions without computing isolating boxes for all the real solutions.

5 Sturm-Habicht Method

In this section, as in Sect. 4, we shall concentrate only on the maximal γ-projection $\bar{\gamma}$ of the real solutions of the polynomial system. But instead of shearing the system, we shall verify the existence of a real root of the polynomial system over $\bar{\gamma}$ by studying the sign variation of the leading coefficients of *subresulant polynomials* over $\bar{\gamma}$. Hence, before explaining the third proposed method, we first state again a few standard preliminaries on *subresultants* and *Sturm-Habicht* sequences.

We denote by $\mathscr{K}$ the *unique factorization domain* $\mathbb{Q}[x]$ and we consider P, $Q \in \mathscr{K}[y]$, where $p = \deg_y(P)$ and $q = \deg_y(Q)$. We assume that $p \geq q$. For $0 \leq i \leq \min(q, p-1)$, the i^{th} *subresultant polynomial* of P and Q is denoted by $\mathrm{Sres}_{y,i}(P, p, Q, q)$. When there is no ambiguity on the degrees of the polynomials P and Q, we simply denote it by $\mathrm{Sres}_{y,i}(P, Q)$. It has degree at most i in y and the coefficient of y^i is denoted by $\mathrm{sres}_{y,i}(P, Q)$. It is called the i^{th} *principal subresultant coefficient*. We recall that $\mathrm{sres}_{y,i}(P, Q) = 0$ implies that $\mathrm{Sres}_{y,i}(P, Q)$ vanishes identically. Note that $\mathrm{Sres}_{y,0}(P, Q) = \mathrm{sres}_{y,0}(P, Q)$ is the *resultant* of P and Q with respect to y, also denoted by $\mathrm{Res}(P, Q, y)$. The greatest common divisor $\gcd(P, Q)$ of the polynomials P and Q (uniquely defined up to units of $\mathscr{K}$) is the first non-zero subresultant polynomial $\mathrm{Sres}_{y,i}(P, Q)$ for increasing i.

Letting $v = p + q - 1$ and $\delta_k = (-1)^{\frac{k(k+1)}{2}}$ for $k \in \mathbb{Z}_{\geq 0}$, the j^{th} *polynomial in the Sturm-Habicht sequence* associated to (P, Q), denoted by $\mathrm{StHa}_j(P, Q)$, is then defined by $\delta_{v-j}\,\mathrm{Sres}_{y,j}(P, v+1, P^{'}Q, v)$, where $P^{'}$ denotes the derivative of P with respect to y. The *principal* j^{th} *Sturm-Habicht coefficient* is denoted by $\mathrm{stha}_j(P, Q)$ for $j = 0, \ldots, v+1$. We also denote by **SignVar** the function which maps $\{\mathrm{sign}(\mathrm{stha}_j(P, 1))\}_{j=0,\ldots,v+1}$ to the number of real roots of P. For more details on the function **SignVar**, see [12, Definition 4.1, Theorem 4.1].

As stated above, we aim at computing:

$$\bar{\gamma} = \max\left\{\pi_\gamma\left(V_{\mathbb{R}}\left(\left\langle \bar{n}, \frac{\partial \bar{n}}{\partial \omega}\right\rangle\right)\right) \cup V_{\mathbb{R}}\left(\langle Lc_\omega(\bar{n})\rangle\right)\right\}.$$

Hence, $\bar{\gamma}$ is either the maximal real root of $Lc_\omega(n)$ or an algebraic value over which $\gcd(n(\bar{\gamma}, \omega), \frac{\partial n}{\partial \omega}(\bar{\gamma}, \omega)) \in \mathbb{R}[\omega]$ has at least one real root. We recall that $\gcd(n(\bar{\gamma}, \omega), \frac{\partial n}{\partial \omega}(\bar{\gamma}, \omega))$ is proportional to the first subresultant polynomial $\mathrm{Sres}_{\omega,i}(n, \frac{\partial n}{\partial \omega})$ (for i increasing) that does not identically vanish for $\gamma = \bar{\gamma}$. If $\bar{\gamma}$ is not a real root of $Lc_\omega(n)$, then we can compute the Sturm-Habicht sequence of the univariate polynomial $n(\bar{\gamma}, \omega) \in \mathbb{R}[\omega]$ to check the existence of a real root for $\gcd(n(\bar{\gamma}, \omega), \frac{\partial n}{\partial \omega}(\bar{\gamma}, \omega))$. In what follows, we shall need the next result.

Lemma 3. *Let $P \in \mathbb{Z}[x,y]$ and $\bar{x}$ be a root of* $\mathrm{Res}(P, \frac{\partial P}{\partial y}, y)$. *Moreover, let* $\mathcal{G} = \gcd\left(P(\bar{x}, y), \frac{\partial P}{\partial y}(\bar{x}, y)\right) \in \mathbb{R}[y]$. *If the x-projection of the points of P is bounded by $\bar{x}$, then we have* $V_{\mathbb{R}}(\langle P(\bar{x}, y)\rangle) = V_{\mathbb{R}}(\langle \mathcal{G}(\bar{x}, y)\rangle)$.

Proof. If $V_{\mathbb{R}}(\langle \mathcal{G}(\bar{x}, y)\rangle) \subsetneq V_{\mathbb{R}}(\langle P(\bar{x}, y)\rangle)$, then there exists $y_0 \in \mathbb{R}$ such that $P(\bar{x}, y_0) = 0$ and $\mathcal{G}(\bar{x}, y_0) \neq 0$. This is equivalent to saying that $P(\bar{x}, y_0) = 0$ and $\frac{\partial P}{\partial y}(\bar{x}, y_0) \neq 0$. Hence, based on the theorem of implicit functions, there exists a real function φ of class C^p $(p > 0)$, defined on an open interval $V \subset \mathbb{R}$, containing $\bar{x}$, and an open neighborhood Ω of $(\bar{x}, y_0)$ in $\mathbb{R}^2$ such that for all (x, y) in $\mathbb{R}^2$, $\{(x, y) \in \Omega \mid P(x, y) = 0\}$ is equivalent to $\{x \in V \mid y = \varphi(x)\}$. This cannot be true since the x-projection of the points of the curve $P = 0$ is bounded by $\bar{x}$, and thus, an open interval containing $\bar{x}$, such as V, does not exist. Consequently, we obtain $V_{\mathbb{R}}(\langle P(\bar{x}, y)\rangle) = V_{\mathbb{R}}(\langle \mathcal{G}(\bar{x}, y)\rangle)$.

Algorithm 3. Sturm-Habicht method

Input: A bivariate polynomial $P \in \mathbb{Z}[x,y]$ such that $P = 0$ is bounded in the x-direction.
Output: Isolating interval of $\max\left\{\pi_x\left(V_{\mathbb{R}}\left(\left\langle P, \frac{\partial P}{\partial y}\right\rangle\right)\right) \cup V_{\mathbb{R}}\left(\langle Lc_y(P)\rangle\right)\right\}$.

1. Compute $\{\mathrm{Sres}_j(P, \frac{\partial P}{\partial y})\}_{j=0,\ldots,\deg_y(P)}$.
2. Compute $x_1 < \ldots < x_m$ the real roots of sres_0.
3. for i from 1 to m do:
   - if $x_{1-i+m} \in V_{\mathbb{R}}(\langle Lc_y(P)\rangle)$ then **return** the isolating interval of x_{1-i+m};
   - **elif SignVar**$(\{\mathrm{sign}(\mathrm{stha}_{d_y}(x_{1-i+m})), \ldots, \mathrm{sign}(\mathrm{stha}_1(x_{1-i+m}))\}) > 0$, then **return** the isolating interval of x_{1-i+m}.
4. end if end do.

Lemma 4. *Let $P \in \mathbb{Z}[x,y]$, $d_x = \deg_x(P)$, $d_y = \deg_y(P)$ and τ be the maximal coefficients bitsize of P. Let $\{\mathrm{StHa}_j(P(x,y), 1)\}_{j=0,\ldots,d_y}$ be the Sturm-Habicht sequence and x_j a real root of* $\mathrm{sres}_{y,0}\left(P, \frac{\partial P}{\partial y}\right)$. *Then,* $\{\mathrm{sign}(\mathrm{stha}_k(x_j))\}_{k=d_y,\ldots,1}$ *can be computed in $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\, (d_x + \tau))$ bit operations.*

Proof. We denote by sres_0 (resp., sres_i) $\mathrm{sres}_{y,0}\left(P, \frac{\partial P}{\partial y}\right)$ (resp., $\mathrm{sres}_{y,i}\left(P, \frac{\partial P}{\partial y}\right)$), where $\mathrm{sres}_i \in \mathbb{Z}[x]$. We first recall that $\mathrm{stha}_i(x_j) = \delta_{d_y-1-i}\,\mathrm{sres}_i(x_j)$. Based on [2, Proposition 8.46], sres_i is of degree $d_x\, d_y$ and of coefficients bitsize $d_y\, \tau$. Thus, the square free part of sres_0 is of coefficients bitsize $\mathcal{O}(d_y\,(d_x + \tau))$ and, based on [4, Lemma 5], can be computed in $\tilde{\mathcal{O}}_B(d_x^2\, d_y^3\, \tau)$. By following the proof of [17, Proposition 6], the overall cost for obtaining the list $\{\mathrm{sign}(\mathrm{stha}_{d_y}(x_j)), \ldots, \mathrm{sign}(\mathrm{stha}_1(x_j))\}$ is $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\, (d_x + \tau))$.

Theorem 3. *Let $P \in \mathbb{Z}[x, y]$ be such that $d_x = \deg_x(P)$, $d_y = \deg_y(P)$ and of maximal coefficients bitsize τ. Then, we can compute an isolating interval of the maximal x-projection of the real solutions of $\{P, \frac{\partial P}{\partial y}\}$ (Algorithm 3) in $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\,(d_x + \tau))$ bit operations in the worst case.*

Proof. The maximal x-projection of the real solutions of $\{P, \frac{\partial P}{\partial y}\}$ is the maximal real root of $\mathrm{sres}_0\left(P, \frac{\partial P}{\partial y}\right)$, say x_m, such that $\gcd\left(P(x_m, y), \frac{\partial P}{\partial y}(x_m, y)\right)$ has at least one real root. If the x-projection of the points of P is bounded by x_m, then, by Lemma 3, the real roots of $\gcd\left(P(x_m, y), \frac{\partial P}{\partial y}(x_m, y)\right)$ are the real roots of $P(x_m, y)$. Consequently, we can compute an isolating interval of x_m using Algorithm 3. According to [2, Proposition 8.46], we can compute the set of principal subresultants in $\tilde{\mathcal{O}}(d_x\, d_y^3\, \tau)$ bit operations and each subresultant polynomial is of degree $\mathcal{O}(d_x\, d_y)$ and of coefficient bit size $\tilde{\mathcal{O}}(d_y\, \tau)$. Thus, step 2, which performs real root isolation of sres_0, is of complexity $\tilde{\mathcal{O}}((d_x, d_y)^3 + (d_x\, d_y)^2\, d_y\, \tau)$ [3, Lemma 54]. Using Lemma 4, step 3 can be done in $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\,(d_x + \tau))$ operations since its first step is of complexity $\tilde{\mathcal{O}}_B(d_x^3 + d_x^2\, \tau)$. Hence, the overall complexity of this algorithm is $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\,(d_x + \tau))$.

Considering the notations of Lemma 1, the following result is an immediate consequence of Corollary 1 and Theorem 3.

Corollary 3. *Based on Theorem 3, $\| G \|_\infty$ can be computed by Sturm-Habicht method in the worst case bit complexity $\tilde{\mathcal{O}}_B(\alpha^{10}\, N^4\,(\alpha + \tau_n))$.*

In Algorithm 4, we suppose that there are no real isolated points, and thus, we replace the computation of signs of polynomials at real algebraic numbers by signs of polynomials at rational numbers. Syntactically, these are small modifications but the effect on the computations is consequent in practice, as well as in theory, since the evaluation of signs of polynomials at real algebraic numbers carries the theoretical worst case complexity of Algorithm 3.

Theorem 4. *Let $P \in \mathbb{Z}[x, y]$ be a bivariate polynomial of maximal coefficient bitsize τ and let $d_x = \deg_x(P)$ and $d_y = \deg_y(P)$. Moreover, let us suppose that $V_{\mathbb{R}}(\langle P \rangle)$ has no isolated singular points. Using Algorithm 4, an isolating interval of the maximal x-projection of the real solutions of $\{P, \frac{\partial P}{\partial y}\}$ can be computed in the worst case bit complexity $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\, \tau)$.*

Proof. As mentioned in the proof of Theorem 3, we can compute the set of principal subresultants in $\tilde{\mathcal{O}}(d_x\, d_y^3\, \tau)$ bit operations and each subresultant polynomial is of degree $\mathcal{O}(d_x\, d_y)$ and of coefficient bitsize $\tilde{\mathcal{O}}(d_y\, \tau)$ according to [2, Proposition 8.46]. Thus, step 2 of Algorithm 4, which performs the real root isolation of sres_0, is of complexity $\tilde{\mathcal{O}}((d_x\, d_y)^3 + (d_x\, d_y)^2\, d_y\, \tau)$ [3, Lemma 54]. Steps 3 and 4 are of same bit complexity: in these steps, we perform $\mathcal{O}(d_y)$ evaluations of the principal subresultant polynomials over a rational number which is between two real roots of sres_0. This rational number is of

Algorithm 4. Sturm-Habicht method - equidimensional

Input: A bivariate polynomial $P \in \mathbb{Z}[x, y]$ such that the curve $P = 0$ is bounded in the x-direction and has not real isolated singular points.
Output: An isolating interval of $\max\left\{\pi_x\left(V_{\mathbb{R}}\left(P, \frac{\partial P}{\partial y}\right)\right) \cup V_{\mathbb{R}}\left(Lc_y(P)\right)\right\}$

1. Compute $\{\mathrm{StHa}_j(P, 1)\}_{j=0,\ldots,\deg_y(P)}$.
2. Let $x_1 < \ldots < x_m$ be the roots of sres_0.
3. for i from 1 to m do:
   - if $x_{1-i+m} \in V_{\mathbb{R}}(Lc_y)$, then **return** the isolating interval of x_{1-i+m};
   - **else** let $X' \in \mathbb{Q}$ such that $x_{m-i} < X' < x_{1-i+m}$;
     - if **SignVar**$(\{\mathrm{sign}(\mathrm{stha}_{d_y}(X')), \ldots, \mathrm{sign}(\mathrm{stha}_1(X'))\}) > 0$, then **return** the isolating interval of x_{1-i+m};
     - end if.
   - end if.
4. end do.

worst possible coefficient bitsize $\tilde{\mathcal{O}}_B(d_x\, d_y^2\, \tau)$, which is equal to the separating bound of sres_0. According to [4, Lemma 6], the d_y evaluations are done in $\tilde{\mathcal{O}}_B(d_y\,(d_x\, d_y\,(d_y\, \tau + d_x\, d_y^2\, \tau))) = \tilde{\mathcal{O}}_B(d_x^2\, d_y^4\, \tau)$. Hence, the overall cost is given by $\tilde{\mathcal{O}}_B(d_x^2\, d_y^4\, \tau)$.

Corollary 4. *Based on Theorem 4, $\| G \|_\infty$ can be computed by the Sturm-Habicht method (Algorithm 4) in the worst case bit complexity $\tilde{\mathcal{O}}_B(\alpha^{10}\, N^4\, \tau_n)$.*

From the above complexity analysis, we can conclude that RUR method and the Sturm-Habicht method have comparable theoretical complexities since, in our case, we have $\alpha \ll N$.

6 Experiments

6.1 Practical Example

We consider the following transfer matrix:

$$G = \begin{pmatrix} \dfrac{1}{s+1} & \dfrac{1}{s+1} \\ 0 & \dfrac{1}{s+1} \end{pmatrix} \in \mathcal{RH}_\infty^{2\times 2}.$$

Let $\Phi_\gamma(s) = \gamma^2\, I_2 - \tilde{G}(s)\, G(s)$ and $\det(\Phi_\gamma(i\,\omega)) = \dfrac{n(\gamma, \omega)}{d(\omega)}$. We study the real solutions of the polynomial system $\Sigma = \{\bar{n}, \frac{\partial \bar{n}}{\partial \omega}\}$, where:

$$\bar{n}(\gamma, \omega) = \gamma^4\, \omega^4 + \gamma^2\,(2\,\gamma^2 - 3)\,\omega^2 + (\gamma^2 + \gamma - 1)(\gamma^2 - \gamma - 1).$$

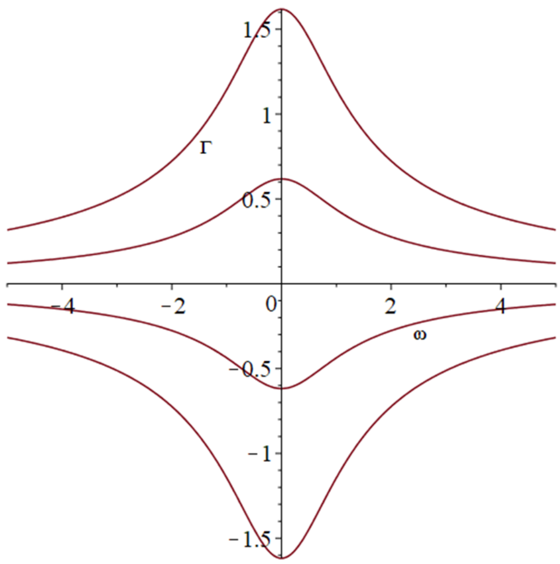

Fig. 2. Plot of $n(\gamma, \omega) = 0$, where ω/γ is in the horizontal/vertical axis.

We first compute $V_{\mathbb{R}}(\mathrm{Lc}_\omega(\bar{n})) = \{0\}$. Then, applying the RUR method, we obtain the following rational univariate representation:

$$\begin{cases} p = (t^2 + t - 1)(t^2 - t - 1), \\ \gamma = \dfrac{3\,t^2 - 2}{t\,(2\,t^2 - 3)}, \\ \omega = 0. \end{cases}$$

Thus, the system's real solutions (γ, ω) are:

$$\left(-\frac{\sqrt{5}}{2} - \frac{1}{2}, 0\right), \left(-\frac{\sqrt{5}}{2} + \frac{1}{2}, 0\right), \left(\frac{\sqrt{5}}{2} - \frac{1}{2}, 0\right), \left(\frac{\sqrt{5}}{2} + \frac{1}{2}, 0\right).$$

Thus, we simply pick their maximal γ-projection to obtain $\frac{\sqrt{5}}{2} + \frac{1}{2}$, which yields:

$$\| G \|_\infty = \max\left\{0, \frac{\sqrt{5}}{2} + \frac{1}{2}\right\} = \frac{\sqrt{5}}{2} + \frac{1}{2}.$$

Following the second approach, which consists in directly focusing on the maximal γ-projection of the system's real solutions, we first compute $\mathrm{Res}(\bar{n}, \frac{\partial \bar{n}}{\partial \omega}, \omega)$ and denote by $R = \gamma\,(\gamma^2 + \gamma - 1)\,(\gamma^2 - \gamma - 1) \in \mathbb{Z}[\gamma]$ its square free part. Then, the maximal real root of R has the following isolating interval:

$$[a, b] = \left[\frac{56929509912547}{35184372088832}, \frac{113859019825121}{70368744177664}\right].$$

Following the Root Separation method, we obtain:

$$\begin{cases} s = \dfrac{12060328540887}{281474976710656}, \\ \delta = \dfrac{43490275647441}{140737488355328}. \end{cases}$$

We have $R_t = \mathrm{Res}(\Psi_s(\bar{n}), \Psi_s(\frac{\partial \bar{n}}{\partial \omega}), \omega) = \alpha\,(t^2+t-1)(t^2-t-1)$, where α is a rational of size about 3000 bits. We denote by $t_{\max}$ the maximal real root of R_t. An isolating interval of $t_{\max}$ is then given by:

$$[c,d] = \left[\frac{113859019825095}{70368744177664}, \frac{56929509912561}{35184372088832}\right].$$

In this case, $[c,d] \subset [a-\delta, b+\delta]$, which shows that $\| G \|_\infty$ is equal to the maximal real root of R of isolating interval $[a,b]$.

Following the Sturm-Habicht method, we have to check the existence of a real root for the univariate polynomial $\bar{n}([a,b],\omega)$. To do that, we first compute $L = [\mathrm{sres}_{\omega,i}(\bar{n}, \frac{\partial \bar{n}}{\partial \omega})]_{i=1,\ldots,\deg_\omega(n)=4} = [5\,\gamma^{14}\,(2\,\gamma^2-3), 2\,\gamma^{10}\,(2\,\gamma^2-3), 4\,\gamma^4, \gamma^4]$. We then compute the list of signs of the elements of L over $[a,b]$. We obtain the list $L_s = [-,-,+,+]$. Then, $\mathbf{SignVar}(L_s) = 1$ and we conclude that $\bar{n}([a,b],\omega)$ admits one real root. Hence, $\| G \|_\infty$ is equal to the maximal real root of R of isolating interval $[a,b]$.

6.2 Experiments

The three proposed methods can be implemented in a few lines of `Maple` but we then have to use implementations at different levels that do not give valuable information about the intrinsic efficiency. For instance, the RUR is implemented in `C` but for general zero dimensional polynomial systems: a variant for bivariate polynomials, the one used for the complexity analysis, is not part of `Maple` and is much more efficient for bivariate systems.

In order to have fair comparisons, we extract the dominating operations and compare them using exactly the same implementations. Namely, resultant computations of sheared/non sheared systems and Root Isolation carry the largest percentage of the computation time. For instance, Algorithm 4 saves time on the resultant computation since it does not perform any shear while it loses time on the root isolation.

For the three methods, the principle subresultant sequence is computed using the routine `SubResultantChain` of the `Maple` package `RegularChain`.

Isolating the real roots of univariate polynomials is another common basic block shared between the three algorithms for which we use `Isolate` provided by the `Maple routine` package `RootFinding`.

In left table of Table 1, we list the main steps of the three algorithms. The check marks mean that the step makes part of the method and the double check marks indicate that this step is the bottleneck of the method. Note that *Res1* stands for the resultant of the original system and *Res2* for the resultant of the sheared system. Keep in mind that the shear done in `Hinf_RUR` is different than the one done in `Hinf_Sep`. Finally, *Iso* means `Isolate`.

In the right table of Table 1, we report the average running time in CPU seconds of the marked steps listed in the table on the left of Table 1 for the three proposed algorithms run on square matrices of size α, with entries given by

Table 1. Left: main steps considered in the implementation of the proposed method. Right: timings for $\mathcal{L}_\infty$-norm for random matrices with $\tau_G = 2$.

	Res1 + Iso	Res2 + Iso	List of signs
Hinf_RUR	✓	✓✓	
Hinf_Srep	✓	✓✓	
Hinf_Sres	✓		✓✓

α	N	Hinf_RUR	Hinf_Sep	Hinf_Sres
2	2	0.2	3	0.2
	3	0.5	7	0.5
	4	2.5	25	2
	5	10	83	6
	6	37	96	10
	7	50	186	47.5
	8	133.5	353	59
	9	236	394	130

random proper rational functions of degree N (degree of the denominators)[1]. It corresponds to a fixed input coefficient bitsize $\tau = 2$, i.e., the rational functions involved in the entries of the matrices have coefficients of magnitude $\mathcal{O}(2^\tau)$.

We finally mention that with these experiments, our goal is not to illustrate the theoretical complexity, but, on the contrary, to show that on practical examples, the results in practice are different than in theory. In theory, the RUR algorithm might asymptotically be the fastest while in practice the Sturm method performs better.

7 Conclusion

In this paper, we have presented three different algorithms for the computation of the $\mathcal{L}_\infty$-norm of the transfer matrix of a finite-dimensional linear control system. By reformulating this problem as the search for the maximal projection of the real solutions of a zero dimensional polynomial system, we have used existing methods such as the rational univariate representation (RUR method). As for the second algorithm, we have only used a special separating linear transformation to shear the polynomial system and put it in a generic position. The last method (Sturm-Habicht method) was based on verifying the existence of a real root for a univariate polynomial having real coefficients.

The complexity analysis has showed that the RUR method has the best theoretical efficiency in comparison with other algorithms. Practically, as we can notice in the tables given in Sect. 6, the practical efficiency is nearly matching with the theoretical efficiency but with a slight advantage for the Sturm-Habicht method probably due to the fact that it is the most adaptive one.

[1] The experiments were conducted on Intel(R) Core(TM) i7-7500U CPU @ 2.70 GHz 2.90 GHz, Installed RAM 8.00 GB under a Windows platform.

References

1. Aubry, P., Maza, M.M.: Triangular sets for solving polynomial systems: a comparative implementation of four methods. J. Symb. Comput. **28**(1–2), 125–154 (1999)
2. Basu, S., Pollack, R., Roy, M.F.: Existential theory of the reals, volume 10 of algorithms and computation in mathematics (2006)
3. Bouzidi, Y., Lazard, S., Moroz, G., Pouget, M., Rouillier, F., Sagraloff, M.: Solving bivariate systems using rational univariate representations. J. Complex. **37**, 34–75 (2016)
4. Bouzidi, Y., Lazard, S., Pouget, M., Rouillier, F.: Separating linear forms and rational univariate representations of bivariate systems. J. Symb. Comput. **68**, 84–119 (2015)
5. Boyd, S., Balakrishnan, V., Kabamba, P.: A bisection method for computing the h_∞ norm of a transfer matrix and related problems. Math. Control Signals Syst. **2**(3), 207–219 (1989)
6. Boztaş, S., Shparlinski, I.E. (eds.): AAECC 2001. LNCS, vol. 2227. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45624-4
7. Chen, C., Maza, M.M., Xie, Y.: A fast algorithm to compute compute the h_∞-norm of a transfer function matrix. Syst. Control Lett. **14**, 287–293 (1990)
8. Chen, C., Mazza, M.M., Xie, Y.: Computing the supremum of the real roots of a parametric univariate polynomial (2013)
9. Cheng, J.S., Gao, X.S., Li, J.: Root isolation for bivariate polynomial systems with local generic position method. In: Proceedings of the 2009 International Symposium on Symbolic and Algebraic Computation, pp. 103–110. ACM (2009)
10. Cheng, J.S., Gao, X.S., Yap, C.K.: Complete numerical isolation of real roots in zero-dimensional triangular systems. J. Symb. Comput. **44**(7), 768–785 (2009)
11. Doyle, J.C., Francis, B.A., Tannenbaum, A.R.: Feedback Control Theory. Dover Publications, New York (1992)
12. González-Vega, L., Recio, T., Lombardi, H., Roy, M.F.: Sturm–Habicht sequences, determinants and real roots of univariate polynomials. In: Caviness, B.F., Johnson, J.R. (eds.) Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation (A Series of the Research Institute for Symbolic Computation, Johannes-Kepler-University, Linz, Austria). Springer, Vienna (1998). https://doi.org/10.1007/978-3-7091-9459-1_14
13. Kanno, M., Smith, M.C.: Validated numerical computation of the l_∞-norm for linear dynamical systems. J. Symb. Comput. **41**(6), 697–707 (2006)
14. Lazard, S., Pouget, M., Rouillier, F.: Bivariate triangular decompositions in the presence of asymptotes. J. Symb. Comput. **82**, 123–133 (2017)
15. Rouillier, F.: Solving zero-dimensional systems through the rational univariate representation. Appl. Algebra Eng. Commun. Comput. **9**(5), 433–461 (1999)
16. Rouillier, F., Zimmermann, P.: Efficient isolation of polynomial's real roots. J. Comput. Appl. Math. **162**(1), 33–50 (2004)
17. Strzebonski, A., Tsigaridas, E.: Univariate real root isolation in an extension field and applications. J. Symb. Comput. **92**, 31–51 (2019)
18. Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice Hall, Upper Saddle River (1996)