Certifying zeros of polynomial systems using interval arithmetic

Paul Breiding[*], Kemal Rose[†], and Sascha Timme[‡§]

Abstract

We introduce a new function in `HomotopyContinuation.jl` that certifies if an isolated solution of a square system of polynomial equations is near a given approximation.

Our implementation is based on Krawczyk's method and uses interval arithmetic. We demonstrate our method on several examples.

1 Introduction

In many physical and engineering systems the underlying structure is algebraic. For example, in physics and chemistry, the geometry of molecules is often modelled with constraint on the distance or the angles between atoms. In robotics connecting joints have constraints on distances or angles. Another example is algebraic statistics, where models are defined by polynomials in the parameters. A central task in those applications is computing the isolated zeros of a system of polynomials.

The first and the third author develop `HomotopyContinuation.jl` [BT18], a software for numerically computing the isolated zeros of a system of polynomial equations. The implementation is based on *numerical homotopy continuation*. There are several implementations of numerical homotopy continuation available: `Bertini` [BHSW], `Hom4PS-3` [CLL14], `HomotopyContinuation.jl` [BT18], `NAG4M2` [Ley11] and `PHCpack` [Ver99]. Hauenstein and Sottile [HS12] remark that while all of these softwares "routinely and reliably solve systems of polynomial equations with dozens of variables having thousands of solutions" they have the shortcoming that "the output is not certified" and that "this restricts their use in some applications, including those in pure mathematics". To remedy this they developed the software `alphaCertified`. This software can rigorously certify that Newton's method starting at a given numerical approximation converges quadratically to a true solution by using Smale's α-theory [Sma86].

Hauenstein and Sottile's contribution to computational algebraic geometry was a milestone. They were the first to provide access to an implementation of a certification method in computational algebraic geometry. A major downside of `alphaCertified` is that it needs to use exact rational arithmetic to produce rigorous certificates. This turns the big advantage of numerical computations – fast floating point arithmetic – upside down and makes certification of complicated problems infeasible due to the large cost of performing rational arithmetic.

However, Hauenstein and Sottile were not the first to implement a certification algorithm. Already since the 1950's researchers have worked on effective methods for certifying solutions of

[*]PB: Technische Universität Berlin. breiding@math.tu-berlin.de.

[†]KR: Technische Universität Berlin. kemalrose@t-online.de.

[‡]ST: Technische Universität Berlin. timme@math.tu-berlin.de. Supported by the Deutsche Forschungsgemeinschaft (German Research Foundation) Graduiertenkolleg *Facets of Complexity* (GRK 2434)

[§]PB and KR have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 787840).

nonlinear systems of equations: Dwyer [Dwy51], Moore [Moo59, Moo66], Sunaga [Sun58], Warmus [War56] and others studied *interval arithmetic*. An algorithm that uses interval arithmetic for certification is *Krawczyk's method* [Kra69, Mad73, Moo66, Moo77, Nic71]. In many areas of applied mathematics interval arithmetic is a prominent tool already. The wide range of applications includes, amongst others, chemical engineering [GS05], ecological modeling and transition state analysis [PSK08], economics and finance [SH05, SH10], thermodynamics [GD05], robotics [KSS15], and the finite elements method [SxZz01].

They key advantage of using interval arithmetic is that it can be combined with floating point arithmetic: The standard model of floating point arithmetic [Hig02, Section 2.3] assumes that the result of a floating point operation is accurate up to relative unit roundoff: $\mathrm{fl}(x \circ y) = (x \circ y)(1+\delta)$, where $|\delta| \leq u$ and $\circ \in \{+, -, *, /\}$. For instance, following the IEE-754 standard the unit roundoff in double precision arithmetic is $u = 2^{-53} \approx 2.2 \cdot 10^{-16}$. The key property in the context of interval arithmetic is that each result of a floating point operation can be rounded outwards such that the resulting *interval* contains the true (exact) result; see, e.g., [May17, Section 3.2]. This enables rigorous computations using fast floating point arithmetic. We think that the upside of providing certified results without abstaining from fast floating point arithmetic will make interval arithmetic a useful and competitive tool in computational algebraic geometry.

1.1 Contribution

Our goal with this article is to introduce an implementation of Krawczyk's certification method in `HomotopyContinuation.jl`. Starting from version 2.1 the software provides the function `certify`; see the documentation at [Cer]. This function takes as input a square polynomial system F and a numerical approximation of a complex zero $x \in \mathbb{C}^n$ (or a list of zeros). If the output says "certified", then this is a rigorous proof that a solution of $F = 0$ is near x. In technical terms, our implementation returns *strong interval approximate zeros*. We introduce this notion in Definition 5.1 below. Roughly speaking, what this means is that our implementation returns a box in $\mathbb{C}^n$ in which a unique true zero of the polynomial is contained. If the output says "not certified", then this does not necessarily mean that there is no zero near x, just that the method couldn't find any strong interval approximate zeros. In addition, if the given polynomial system is real, we give a certificate whether the certified zero is a real zero. Real zeros to polynomial systems are of great interest both in applications and in pure mathematics [Sot97].

1.2 Outline

The rest of this article is organized as follows: In the next section we apply our implementation to two examples from applications and to one example in pure mathematics. We discuss the details of our implementation in Section 5. For completeness, we include a short introduction to interval arithmetic in Section 3 and a proof of Krawczyk's method in Section 4.

2 Applications

We discuss the relevance of our implementation for three applications from different fields.

2.1 3264 real conics

The first example is motivated by [BST20]. It demonstrates how our implementation of the certification method can be used to prove theorems in algebraic geometry. In particular, our implementation is useful when one wants to prove statements on the number of zeros, the number of real zeros, or the number of positive real zeros of a polynomial system. A computation with our certification always reveals lower bounds for these numbers.

In [BST20] we have used `alphaCertified` to prove that a certain arrangement of five conics in the plane had 3264 real conics, which were simultaneously tangent to each of the five given conics. Such an arrangement is called *totally real*. It was known before that such arrangements exist [RTV97, Sot, Sot97], but an explicit instance was not known. The fact that `alphaCertified` was able to provide a totally real instance highlights the relevance of certification software in algebraic geometry. However, the computation with `alphaCertified` for the 3264 real conics took more than 36 hours on a desktop computer. This is ultimately because `alphaCertified` relies on exact arithmetic, which is slow. By contrast, interval arithmetic is fast. Our implementation certifies the reality of the 3264 conics in a few seconds on a laptop.

The strategy for the computation is this. The zeros of the system (12) in [BST20] give the coordinates of the 3264 conics which are tangent to five given conics. We compute the zeros for the coordinates of the specific instance in [BST20, Figure 2] using `HomotopyContinuation.jl`. This is a numerical computation. Therefore, it is inexact and can't be used in a proof. Nevertheless, we take the inexact numerical zeros as starting points for our certification method. If our implementation outputs that it has found a real certified zero, then this is an exact result and hence it is a proof that the zero is real. This way we can prove that indeed all the 3264 conics for the instance in [BST20, Figure 2] are real. See also the proof of [BST20, Proposition 1] for a more detailed discussion.

2.2 Numerical Synthesis of Six-Bar Linkages

In this example we want to demonstrate that the certification routine can cope with large problems. We consider the kinematic synthesis of six-bar linkages that use eight prescribed accuracy points as described in [PM14]. In this article the authors derive the synthesis equations for six-bar linkages of the Watt II, Stephenson II, and Stephenson III type. Additionally, in [PM14, Eq. (35)] they construct a system of 22 polynomials in 22 unknowns and 224 parameters that can be used as a start system in a parameter homotopy to solve the synthesis equations of all three considered six-bar linkage types.

The number of non-singular solutions of this generalized start system is reported as 92,736. It was computed using `Bertini` and a multi-homogeneous start system. To certify the reported root count we solved the generalized start system using the monodromy method [DHJ$^+$18] implementation in `HomotopyContinuation.jl`. In our computation we obtained for a generic choice of the 224 parameters 92,752 non-singular solutions, sixteen *more* than reported in [PM14]. We certified this root count using our certification routine and obtained 92,752 strong interval approximate zeros. In Section 5.3 below we discuss that part of the certification process is checking if the intervals are pairwise disjoint. In this example there are 4,301,420,376 such pairs, which underlines the need for having an efficient algorithm for comparing pairs.

On an iMac with a 3.4 GHz processor the computation for the certification needed 38.34 seconds. Therefore, we have a certificate that the generalized system has in general (at least) 92,752 non-singular solution.

2.3 Stress response of Bacillus Subtilis

The example in this section is from [NTI16]. In this article reactions to environmental stress of the bacterium Bacillus subtilis are modelled. The protein σ^B is the focus of this paper. It is responsible for a general stress induced response of the bacterium. σ^B belongs to the family of σ-factors; proteins which govern the expression of genes. They enable the transcription of DNA into RNA by binding RNA polymerase to certain sequences of the DNA, called gene promoters.

In [NTI16] the regulations, which induce response to stresses such as heat or ethanol, of σ-factors are studied. There exist regulatory systems of other proteins that mediate the activity of σ-factors in different ways. Those networks comprise feedback loops that influence the σ-factor. There can be many possible substances involved in many possible reactions, and the resulting system of differential equations might be very complicated and possibly sensitive to a change of the data constituting its coefficients. This makes understanding of the regulating networks hard.

As claimed in [NTI16], in the case of Bacillus subtilis the experimantal data can be backed up by satisfactory model of the regulatory network of σ^B : the activity of σ^B is regulated by a partner-switching network consisting of an anti-σ-factor RsbW and an anti-anti-σ-factor RsbV. The factor RsbW is able to react with σ^B in its dimer form RsbW_2. Under normal circumstances RsbV is in a phosphorylated form, in which it does not effectively interact with RsbW_2. Hence, σ^B is prevented from initiating a stress response. However, stress can release σ^B from its binding with RsbW_2, so that σ^B can initiate a stress response.

In [NTI16] this biochemical reactions dynamical system is modelled by a system of differential equations in the 10 variables w, w_2, $w_{2\mathrm{v}}$, v, $w_{2\mathrm{v}2}$, v_P, σ_B, $w_{2\sigma\mathrm{B}}$, v_{Pp} and phos. These represent, amongst others, the concentrations of σ^B, the anti-σ-factor RsbW, the anti-anti-σ-factor RsbV, their respective dimers, their respective phosphorilated forms, phosphatase complexes and of several different reaction products. In addition, the variable phos measures the concentration of phosphatase, which is used to induce stress on the bacterium.

With our implementation we can determine the steady states of the described dynamical system. The vanishing of the differentials of each of the concentrations with respect to time is equivalent to the vanishing of the ten polynomials below.

$$
\begin{aligned}
(-k_{\mathrm{Deg}}w - 2k_{\mathrm{bw}}\tfrac{w^2}{2} + 2k_{\mathrm{dw}}w_2)(K+\sigma_B) + \lambda_W v_0(1+F\sigma_B) &= 0\\
-k_{\mathrm{Deg}}w_2 + k_{\mathrm{bw}}\tfrac{w^2}{2} - k_{\mathrm{dw}}w_2 - k_{\mathrm{B1}}w_2 v + k_{\mathrm{D1}}w_{2\mathrm{v}} + k_{\mathrm{K1}}w_{2\mathrm{v}} - k_{\mathrm{B3}}w_2\sigma_B + k_{\mathrm{D3}}w_{2\sigma\mathrm{B}} &= 0\\
-k_{\mathrm{Deg}}w_{2\mathrm{v}} + k_{\mathrm{B1}}w_2 v - k_{\mathrm{D1}}w_{2\mathrm{v}} - k_{\mathrm{B2}}w_{2\mathrm{v}}v + k_{\mathrm{D2}}w_{2\mathrm{v}2} - k_{\mathrm{K1}}w_{2\mathrm{v}} + k_{\mathrm{K2}}w_{2\mathrm{v}2} + k_{\mathrm{B4}}w_{2\sigma\mathrm{B}}v - k_{\mathrm{D4}}w_{2\mathrm{v}}\sigma_B &= 0\\
(-k_{\mathrm{Deg}}v - k_{\mathrm{B1}}w_2 v + k_{\mathrm{D1}}w_{2\mathrm{v}} - k_{\mathrm{B2}}w_{2\mathrm{v}}v + k_{\mathrm{D2}}w_{2\mathrm{v}2} - k_{\mathrm{B4}}w_{2\sigma\mathrm{B}}v + k_{\mathrm{D4}}w_{2\mathrm{v}}\sigma_B + k_{\mathrm{P}}v_{\mathrm{Pp}})(K+\sigma_B)&\\
+\lambda_V v_0(1+F\sigma_B) &= 0\\
-k_{\mathrm{Deg}}w_{2\mathrm{v}2} + k_{\mathrm{B2}}w_{2\mathrm{v}}v - k_{\mathrm{D2}}w_{2\mathrm{v}2} - k_{\mathrm{K2}}w_{2\mathrm{v}2} &= 0\\
-k_{\mathrm{Deg}}v_P + k_{\mathrm{K1}}w_{2\mathrm{v}} + k_{\mathrm{K2}}w_{2\mathrm{v}2} - k_{\mathrm{B5}}v_P\mathrm{phos} + k_{\mathrm{D5}}v_{\mathrm{Pp}} &= 0\\
(-k_{\mathrm{Deg}}\sigma_B - k_{\mathrm{B3}}w_2\sigma_B + k_{\mathrm{D3}}w_{2\sigma\mathrm{B}} + k_{\mathrm{B4}}w_{2\sigma\mathrm{B}}v - k_{\mathrm{D4}}w_{2\mathrm{v}}\sigma_B)(K+\sigma_B) + v_0(1+F\sigma_B) &= 0\\
-k_{\mathrm{Deg}}w_{2\sigma\mathrm{B}} + k_{\mathrm{B3}}w_2\sigma_B - k_{\mathrm{D3}}w_{2\sigma\mathrm{B}} - k_{\mathrm{B4}}w_{2\sigma\mathrm{B}}v + k_{\mathrm{D4}}w_{2\mathrm{v}}\sigma_B &= 0\\
-k_{\mathrm{Deg}}v_{\mathrm{Pp}} + k_{\mathrm{B5}}v_P\mathrm{phos} - k_{\mathrm{D5}}v_{\mathrm{Pp}} - k_{\mathrm{P}}v_{\mathrm{Pp}} &= 0\\
(\mathrm{phos} + v_{\mathrm{Pp}}) - p_{\mathrm{tot}} &= 0
\end{aligned}
$$

Here the 23 parameters k_{bw}, k_{dw}, kD, k_{B1}, k_{B2}, k_{B3}, k_{B4}, k_{B5}, k_{D1}, k_{D2}, k_{D3}, k_{D4}, k_{D5}, k_{K1}, k_{K2}, k_{P}, k_{Deg}, v_0, F, K, λ_W, λ_V, p_{tot} describe data that governs the speed of different reactions.

In our example we consider the following specific values.

$$k_{\mathrm{Bw}} = 3600; k_{\mathrm{Dw}} = 18; k_D = 18 k_{\mathrm{B1}} = 3600; k_{\mathrm{B2}} = 3600; k_{\mathrm{B3}} = 3600; k_{\mathrm{B4}} = 1800; k_{\mathrm{B5}} = 3600;$$
$$k_{\mathrm{D1}} = 18; k_{\mathrm{D2}} = 18; k_{\mathrm{D3}} = 18; k_{\mathrm{D4}} = 1800; k_{\mathrm{D5}} = 18; k_{\mathrm{K1}} = 36; k_{\mathrm{K2}} = 36; k_{\mathrm{P}} = 180; k_{\mathrm{Deg}} = 0.7;$$
$$v_0 = 0.4; F = 30; K = 0.2; \lambda_W = 4; \lambda_V = 4.5; p_{\mathrm{tot}} = 2;$$

Solutions of interest of the system of equations above are real solutions for which phos > 0. Using our implementation we can certify that 12 zeros of this system are real. By checking the returned intervals we can also check that 8 of them have phos > 0. Those 8 have the following values for phos:

$$8.99667564713 \cdot 10^{-5} \pm 2.85 \cdot 10^{-17}, \quad 8.95355355193 \cdot 10^{-5} \pm 7.06 \cdot 10^{-17},$$
$$2.035113740902 \pm 1.05 \cdot 10^{-13}, \quad 0.00406661084305 \pm 6.02 \cdot 10^{-15},$$
$$2.0160598826757 \pm 4.13 \cdot 10^{-14}, \quad 0.00413069399709 \pm 8.81 \cdot 10^{-15}.$$
$$0.0054155725325 \pm 2.57 \cdot, 10^{-14}, \quad 0.0052977778316 \pm 3.25 \cdot 10^{-14}.$$

We thank Torkel Loman for pointing out this example to us.

3 Interval arithmetic

We briefly introduce the relevant concepts from interval arithmetic.

3.1 Real interval arithmetic

Real interval arithmetic is based on computation with compact real intervals. Following [May17] we denote the set of all such intervals by

$$\mathbb{IR} := \{[a, b] \mid a, b \in \mathbb{R}, a \leq b\}.$$

For $X, Y \in \mathbb{IR}$ and the binary operation $\circ \in \{+, -, \cdot, /\}$ we define

$$X \circ Y = \{x \circ y \,|\, x \in X, y \in Y\}, \tag{1}$$

where we assume $0 \notin Y$ in the case of division. For these binary operations as well as other standard arithmetic operations, there are explicit formulas for their interval arithmetic versions. See, e.g., [May17] for more details.

3.2 Complex interval arithmetic

We define the set of *rectangular complex intervals* as

$$\mathbb{IC} := \{X + iY \mid X, Y \in \mathbb{IR}\},$$

where $X + iY = \{x + iy \mid x \in X, y \in Y\}$ and $i = \sqrt{-1}$. Following [May17] we define the algebraic operations for $I = X + iY, J = W + iZ \in \mathbb{IC}$ in terms of operations for real intervals from (1):

$$I + J := (X + W) + i(Y + Z), \qquad I \cdot J := (X \cdot W - Y \cdot Z) + i(X \cdot Z + Y \cdot W) \tag{2}$$
$$I - J := (X - W) + i(Y - Z), \qquad \frac{I}{J} := \frac{X \cdot W + Y \cdot Z}{W \cdot W + Z \cdot Z} + i\frac{Y \cdot W - X \cdot Z}{W \cdot W + Z \cdot Z}$$

The need of using (1) instead of using complex arithmetic for the definition of algebraic operations in $\mathbb{IC}$ is demonstrated by the following example from [May17]: Take $I = [1,2] + i[0,0]$ and $J = [1,1] + i[1,1]$. Then, $\{x \cdot y | x \in I, y \in J\} = \{t(1+i) \mid 1 \leq t \leq 2\}$ is not a rectangular complex interval, while $I \cdot J = [1,2] + i[1,2]$ is.

The algebraic structure of $\mathbb{IC}$ is given by following theorem; see, e.g., [May17, Theorem 9.1.4].

Theorem 3.1. *The following holds.*

1. $(\mathbb{IC}, +)$ *is a commutative semigroup with neutral elements.*
2. $(\mathbb{IC}, +, \cdot)$ *has no zero divisors.*

Furthermore, if $I, J, K, L \in \mathbb{IC}$*, then*

3. $I \cdot (J + K) \subseteq I \cdot J + I \cdot K$*, but equality does not hold in general.*
4. $I \subseteq J, K \subseteq L$*, then* $I \circ K \subseteq J \circ L$ *for* $\circ \in \{+, -, \cdot, /\}$.

In particular, the third item from the previous theorem makes working with interval arithmetic challenging. It means that distributivity does not hold in $\mathbb{IC}$. Therefore, if one wants to define the evaluation of polynomials in $\mathbb{IC}$, the order of the computation steps matter. This implies that polynomial maps $\mathbb{IC}^n \to \mathbb{IC}$ have to be defined by straight-line programs, and not just by a list of coefficients. This will become important in Section 5 below.

Arithmetic in $\mathbb{IC}^n$ is defined as follows: if $I = (I_1, \ldots, I_n), J = (J_1, \ldots, J_n) \in \mathbb{IC}^n$, then

$$I + J = (I_1 + J_1, \ldots, I_n + J_n).$$

Scalar multiplication for $I \in \mathbb{IC}$ and $J \in \mathbb{IC}^n$ is defined as $I \cdot J = (I \cdot J_1, \ldots, I \cdot J_n)$. The product of an interval matrix $A = (A_{i,j}) \in \mathbb{IC}^{n \times n}$ and an interval vector $I \in \mathbb{IC}^n$ is

$$A \cdot I := I_1 \cdot \begin{bmatrix} A_{1,1} \\ \vdots \\ A_{n,1} \end{bmatrix} + \cdots + I_n \cdot \begin{bmatrix} A_{1,n} \\ \vdots \\ A_{n,n} \end{bmatrix}. \tag{3}$$

Similar to the one-dimensional case $(\mathbb{IC}^n, +)$ is a commutative semigroup with neutral elements.

4 Krawczyk's method

In this section we recall Krawczyk's method. First, we need three definitions.

Definition 4.1 (Interval enclosement)**.** Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be a map. We say that a map $\square F : \mathbb{IC}^n \to \mathbb{IC}^n$ is an interval enclosement of F, if for every $I \in \mathbb{IC}^n$ we have $\{F(x) \mid x \in I\} \subseteq \square F(I)$.

We will use the notation $\square F$ for interval enclosements in the rest of the paper. Furthermore, in the following we will not distinguish between a point $x \in \mathbb{C}^n$ and the complex interval $[\mathrm{Re}(x), \mathrm{Re}(x)] + i[\mathrm{Im}(x), \mathrm{Im}(x)]$ defined by x. We will simply use the symbol "x" for both terms, so that $\square F(x)$ is well-defined.

Definition 4.2 (Interval matrix norm)**.** Let $A \in \mathbb{IC}^{n \times n}$. We define the operator norm of A as $\|A\| := \max_{B \in A} \max_{v \in \mathbb{C}^n} \frac{\|Bv\|_\infty}{\|v\|_\infty}$, where $\|(v_1, \ldots, v_n)\|_\infty = \max_{1 \leq i \leq n} |v_i|$ is the infinity norm in $\mathbb{C}^n$.

The next definition introduces the so-called *Krawczyk operator*.

Definition 4.3. Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be a system of polynomials, and $\mathrm{J}F$ be its Jacobian matrix seen as a function $\mathbb{C}^n \to \mathbb{C}^{n\times n}$. Let $\square F$ be an interval enclosement of F and $\square \mathrm{J}F$ be an interval enclosement of $\mathrm{J}F$. Furthermore, let $I \in \mathbb{IC}^n$ and $x \in \mathbb{C}^n$ and let $Y \in \mathbb{C}^{n\times n}$ be an invertible matrix. We define

$$K_{x,Y}(I) := x - Y \cdot \square F(x) + (\mathbf{1}_n - Y \cdot \square \mathrm{J}F(I))(I - x).$$

Here, $\mathbf{1}_n$ is the $n \times n$-identity matrix.

Remark 4.4. In the literature, $K_{x,Y}(I)$ is usually defined using $F(x)$ and not $\square F(x)$. Here, we use this definition, because in practice it is usually not feasible to evaluate $F(x)$ exactly. Instead it is replaced by an interval enclosement.

We are now ready to state the theorem, which defines Krawczyk's method. Note that all the data in the theorem can be computed using interval arithmetic.

Theorem 4.5. *Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be a system of polynomials and $I \in \mathbb{IC}^n$. Let $x \in I$ and $Y \in \mathbb{C}^{n\times n}$ be an invertible complex $n \times n$ matrix. The following holds.*

1. *If $K_{x,Y}(I) \subset I$, there is a root of F in I.*
2. *If additionally $\sqrt{2}\,\|\mathbf{1}_n - Y\square \mathrm{J}F(I)\| < 1$, where $\|\cdot\|$ is the operator norm from Definition 4.2, then F has exactly one zero in I.*

In the literature, Krawczyk's method is usually stated for maps between real vector spaces. One of the few sources, which introduces the Krawczyk operator in the complex setting, is [BLL19]. For completeness, we recall their proofs in this section.

The idea for the proof of Theorem 4.5 is to verify that under the assumptions of the theorem the map $x \mapsto x - Y \cdot F(x)$ defines a contraction within an interval I. If this is true, by Brouwer's fixed point theorem there is exactly one fixed-point of this map in I. If Y is invertible, this implies that there is exactly one zero to $F(x)$ in I. In principle, any suitable Y can be used for the theorem. But in practice, ones tries to get a "good" Y by estimating $Y \approx (\mathrm{J}F(x))^{-1}$. Thus, Krawczyk's method can be seen as an interval version of Newton's method.

Before we give the proof of Theorem 4.5, we need a lemma. It is a direct sequence of a complex version of the mean-value theorem which is shown implicitly in the proof of [BLL19, Lemma 2].

Lemma 4.6. *Fix a matrix $Y \in \mathbb{C}^{n\times n}$ and define the map $G_Y : \mathbb{C}^n \to \mathbb{C}^n, x \mapsto x - YF(x)$. Let $I \in \mathbb{IC}^n$ be an interval vector and $x, z \in I$. Then, we have*

1. $G_Y(z) - G_Y(x) \in (\mathbf{1}_n - Y \cdot \square \mathrm{J}F(I))\,\mathrm{Re}(z - x) + (\mathbf{1}_n - Y \cdot \square \mathrm{J}F(I))\,i\mathrm{Im}(z - x)$.
2. $G_Y(I) \subset K_{x,Y}(I)$.

The following proof is adapted from [BLL19, Lemma 2].

Proof of Lemma 4.6. In the proof we abbreviate $G := G_Y$. We first show the second part assuming the first part of the lemma. Then, we prove the first part. We fix an interval $I \in \mathbb{IC}^n$ and $x, z \in \mathbb{C}^n$,

For the second part, we have to show that for all $I \in \mathbb{IC}^n$ we have $G(I) \subset K_{x,Y}(I)$. To show this we define the interval matrix $M := (\mathbf{1}_n - Y\square \mathrm{J}F(I)) \in \mathbb{IC}^{n\times n}$. By definition of $K_{x,Y}$ we have $G(x) + M(I - x) \subset K_{x,Y}(I)$. Thus, we have to show that $G(z) - G(x) \in M(I - x)$, since $z \in I$

is arbitrary. The first part of the lemma implies that we can find matrices $M_1, M_2 \in M$ such that $G(z) - G(x) = M_1\mathrm{Re}(z-x) + iM_2\mathrm{Im}(z-x)$. Decomposing the matrices into real and imaginary part we find

$$G(z) - G(x) = \mathrm{Re}(M_1)\mathrm{Re}(z-x) - \mathrm{Im}(M_2)\mathrm{Im}(z-x) + i(\mathrm{Im}(M_1)\mathrm{Re}(z-x) + \mathrm{Re}(M_2)\mathrm{Im}(z-x)).$$

Since $z - x \in I$ and by definition of the complex interval multiplication from (2) and the interval matrix-vector-multiplication (3) we see that $G(z) - G(x) \in M(I - x)$. This finishes the proof for the second part.

The first part of the lemma may be shown entry-wise. We will show this by combining a complex version of the mean value theorem with the following observation: $\mathrm{J}G(x) = \mathbf{1}_n - Y \cdot \mathrm{J}F(x)$, so we have the inclusion

$$\mathrm{J}G(I) = \mathbf{1}_n - Y \cdot \mathrm{J}F(I) \subseteq \mathbf{1}_n - Y \cdot \square\mathrm{J}F(I). \tag{4}$$

We now relate $G(z) - G(x)$ to (4) using the mean value theorem. First, we define $w := \mathrm{Re}(z) + i\mathrm{Im}(x)$. Let $1 \leq j \leq n$ and G_j denote the j-th entry of G. Then, the real and imaginary part of $h(t) := G_j(tz + (1-t)w)$ are real differentiable functions of the real variable t. The mean value theorem can be applied, and we find $0 < t_1, t_2 < 1$ such that $\mathrm{Re}(h(1)) - \mathrm{Re}(h(0)) = \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Re}(h(t_1))$ and $\mathrm{Im}(h(1)) - \mathrm{Im}(h(0)) = \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Im}(h(t_2))$. Setting $c_1 = t_1 z + (1-t_1)w$ and $c_2 = t_2 z + (1-t_2)w$ this implies

$$G_j(w) - G_j(z) = (\nabla_{\mathrm{Re}}\mathrm{Re}(G_j(c_1)) + i\nabla_{\mathrm{Re}}\mathrm{Im}(G(c_2)))^T(z-w),$$

where $\nabla_{\mathrm{Re}}G$ denotes the vector of partial derivatives with respect to the real variable. Let us denote by G_j' the complex derivative of G_j; that is, $G_j' : \mathbb{C}^n \to \mathbb{C}^n$ as a function. From the Cauchy Riemann equations it follows that $\nabla_{\mathrm{Re}}\mathrm{Re}(G_j(c_1)) = \mathrm{Re}(G_j'(c_1))$ and likewise $\nabla_{\mathrm{Re}}\mathrm{Im}(G(c_2)) = \mathrm{Im}(G'(c_2))$. This yields $G_j(z) - G_j(w) = (\mathrm{Re}(G_j'(c_1) + i\mathrm{Im}(G'(c_2)))^T(z-w)$. Putting these equations ranging over j together we find $G(z) - G(w) = (\mathrm{Re}(\mathrm{J}G(c_1)) + i\mathrm{Im}(\mathrm{J}G(c_2)))(z-w)$. By construction, c_1 and c_2 are contained in I, because w and z are contained in I, and I is a product of rectangles and thus convex. Combined with (4) this yields

$$G(z) - G(w) \in (\mathbf{1}_n - Y \cdot \square\mathrm{J}F(I))(z-w).$$

Using essentially the same arguments for the path from x to w we also find

$$G(w) - G(x) \in (\mathbf{1}_n - Y \cdot \square\mathrm{J}F(I))(w-x).$$

By construction, $z - w = i\mathrm{Im}(z-x)$ and $w - x = \mathrm{Re}(z-x)$, which implies

$$G(z) - G(x) \in (\mathbf{1}_n - Y \cdot \square\mathrm{J}F(I))\,\mathrm{Re}(z-x) + (\mathbf{1}_n - Y \cdot \square\mathrm{J}F(I))\,i\mathrm{Im}(z-x).$$

This finishes the proof. □

Proof of Theorem 4.5. We fix $Y \in \mathbb{C}^{n\times n}$. The second part of Lemma 4.6 implies that, if we have $K_{x,Y}(I) \subseteq I$, then $G_Y(I) \subseteq I$. Brouwer's fixed point Theorem shows that G_Y has a fixed point in I. The fixed point is a root of F. This finishes the proof for the first part of Theorem 4.5. For the second part let $z_1, z_2 \in I$. The first part of Lemma 4.6 implies

$$G_Y(z_1) - G_Y(z_2) \in (\mathbf{1}_n - Y \cdot \square\mathrm{J}F(I))\mathrm{Re}(z_1 - z_2) + (\mathbf{1}_n - Y \cdot \square\mathrm{J}F(I))\; i\mathrm{Im}(z_1 - z_2).$$

(Note that we can't apply the distributivity law because of Theorem 3.1 3.). Applying norms and using submultiplicativity yields

$$\|G_Y(z_1) - G_Y(z_2)\| \leq \|(\mathbf{1}_n - Y \cdot \square\mathrm{J}F(I))\|\|\mathrm{Re}(z_1 - z_2)\|_\infty + \|(\mathbf{1}_n - Y \cdot \square\mathrm{J}F(I))\|\|\mathrm{Im}(z_1 - z_2)\|_\infty.$$

Since $\|\mathrm{Re}(z_1 - z_2)\|_\infty + \|\mathrm{Im}(z_1 - z_2)\|_\infty \leq \sqrt{2}\|z_1 - z_2\|_\infty$ it holds

$$\|G_Y(z_1) - G_Y(z_2)\|_\infty \leq \sqrt{2}\|\mathbf{1}_n - Y \cdot \square \mathrm{J}F(I)\|\|z_1 - z_2\|_\infty.$$

By assumption $\sqrt{2}\|\mathbf{1}_n - Y \cdot \square \mathrm{J}F(I)\|$ is smaller than 1 so G_Y is a contraction. Brouwer's fixed point theorem implies that there is a unique zero in I. □

4.1 Real zeros and positive zeros

For many applications the real zeros of a polynomial system are of most interest. Since most algorithms for numerically computing the isolated zeros of a polynomial system perform the computations over the complex numbers it is important to have a rigorous method to determine whether a zero is real.

Lemma 4.7. *Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be a real square system of polynomials and $I \in \mathbb{C}^n$ such that there exists $x \in I$ and $Y \in \mathbb{C}^{n\times n}$ satisfying $K_{x,Y}(I) \subset I$ and $\sqrt{2}\,\|\mathbf{1}_n - Y\square \mathrm{J}F(I)\| < 1$. If additionally $\{\bar{z} \,|\, z \in K_{x,Y}(I)\} \subset I$, then F has exactly one real zero in I.*

Proof. From Theorem 4.5 follows that F has a unique zero $s \in K_{x,Y}(I) \subset I$. Since F is a real polynomial system it follows that also the element wise complex conjugate $\bar{s}$ is a zero of F. If $\bar{s} \in \{\bar{z} \,|\, z \in K_{x,Y}(I)\} \subset I$ then it follows $\bar{s} = s$ since otherwise $\bar{s}$ and s would be two distinct zeros of F in I, contradicting the uniqueness result from Theorem 4.5. □

Additionally for a wide range of applications it is important to have positive real zeros.

Corollary 4.8. *Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be a real square system of polynomials and $I \in \mathbb{C}^n$ satisfying the conditions of Lemma 4.7 such that F has provenly a unique real zero in I. If $\mathrm{Re}(I) > 0$ then F has a unique positive real zero in I.*

5 Implementation details

In this section we describe the necessary considerations to implement the Krawcyzk method described in Section 4 as well as the technical realization in `HomotopyContinuation.jl`. For this we first need to introduce the following definitions.

Definition 5.1. Let $F : \mathbb{C}^n \to \mathbb{C}^n$ be a square system of polynomials and $I \in \mathbb{IC}^n$. Let $K_{x,Y}(I)$ be the associated Krawczyk operator (see Definition 4.3). If there exists a matrix $Y \in \mathbb{C}^{n\times n}$, such that $K_{x,Y}(I) \subset I$, we say that I is an *interval approximate zero* of F. We call I a *strong interval approximate zero* of F if in addition $\sqrt{2}\|\mathbf{1}_n - Y\square \mathrm{J}F(I)\| < 1$.

Definition 5.2. If I is an interval approximate zero, then, by Theorem 4.5, I contains a zero of F. We call such a zero an *associated zero* of I. Note that, if I is a strong interval approximate zero, the associated zero is unique.

The certification routine takes as input a square polynomial system $F : \mathbb{C}^n \to \mathbb{C}^n$ and a finite list $X \subset \mathbb{C}^n$ of (suspected) approximations of isolated zeros and it returns a list of strong interval approximate zeros $\mathcal{I} = \{I_1, \ldots, I_m\} \in \mathbb{IC}^n$ such that no two intervals I_k and I_ℓ, $k \neq \ell$, overlap. Additionally, if F is a real polynomial system then for each I_k it is determined whether its associated zero is real. The prototypical application of the certification routine is to take as input approximations of all isolated solutions $X \subset \mathbb{C}^n$ of F as computed by numerical homotopy continuation methods.

5.1 Interval enclosements for polynomial systems

As already discussed in Section 3 the fact that distributivity doesn't hold in $\mathbb{IC}$ requires that the polynomial system $F : \mathbb{C}^n \to \mathbb{C}^n$ and its interval enclosement $\square F$ have to be defined by a straight-line program, and not just by a list of coefficients. Additionally, since the overestimation of the interval enclosement $\square F$ increases with the size of the straight line program it is beneficial to express F and $\square F$ by the smallest straight line program possible. Therefore, `HomotopyContinuation.jl` automatically applies optimization heuristics, e.g., a multivariate version of Horner's rule, to reduce the number of operations necessary to evaluate F and $\square F$.

5.2 Determining strong interval approximate zeros

In a first step the certification routine attempts to produce for or a given $x \in X$ a strong interval approximate zero $I \in \mathbb{IC}^n$. Recall, for $I \in \mathbb{IC}^n$ to be a strong interval approximate zero we need by Theorem 4.5 to have a point $\tilde{x} \in I$ and a matrix $Y \in \mathbb{C}^{n\times n}$ such that $K_{\tilde{x},Y}(I) \subset I$ and $\sqrt{2}\,\|\mathbf{1}_n - Y\square \mathrm{J}F(I)\| < 1$ where $K_{\tilde{x},Y}(I)$ is the Krawcyzk operator defined in Definition 4.3.

Before we can discuss the details on how to construct I we need to take a closer look at some of the properties of interval arithmetic implementations. An implementation of interval arithmetic, *machine interval arithmetic*, uses finite precision floating point arithmetic where we assume that the result of a floating point operation is accurate up to relative unit roundoff u. Given $X, Y \in \mathbb{IC}$ the result of $X \circ Y$, $\circ \in \{+, -, *, /\}$, in machine interval arithmetic is the *larger* interval $\mathrm{fl}(X \circ Y) := \{(x \circ y)(1+\delta) \,|\, |\delta| \leq u, x \in X, y \in Y\}$. Additionally, for a given $x \in \mathbb{IC}$ all intervals $\{x + (|\mathrm{Re}(x_j)| + i|\mathrm{Im}(x_j)|)\delta \,|\, |\delta| \leq \mu\}$ with $0 < \mu \leq u$ are identical. A consequence of this is that even if we have an interval I, $\tilde{x} \in I$ and Y such that in exact arithmetic $K_{\tilde{x},Y}(I) \subset I$ it is still possible that in machine interval arithmetic the interval $K_{\tilde{x},Y}(I)$ is *not* contained in I due to a too large unit roundoff u. Therefore, it can be necessary to compute with a smaller unit roundoff u than provided by double precision arithmetic ($u = 2^{-53}$). Our implementation uses machine interval arithmetic based on double precision arithmetic, but if necessary also the arbitrary precision interval arithmetic implemented in `Arb` [Joh17].

Now, we can turn to the construction of I. Given a point $x \in X$ and a working precision u the point is refined using Newton's method to maximal accuracy in the given working precision u. We denote this refined point $\tilde{x}$. Here, we assume that x is already in the region of quadratic convergence of Newton's method. Next, the point $\tilde{x}$ needs to be inflated to an interval I with $\tilde{x} \in I$. This process is called in the literature ε-inflation [May17, Sec. 4.3]. However, choosing the correct I is a hard problem. I cannot be chosen too small or too large since then the Krawcyzk operator is not a contraction. Similarly even if a true zero x^* of F is contained I then the Krawcyzk operator still doesn't need to be a contraction due to the overestimation of machine interval arithmetic.

In spite of these difficulties, we found the following heuristic to determine I work very well. Since we assume $\tilde{x}$ to be in the region of quadratic convergence of Newton's method it follows from the Newton-Kantorovich theorem that $||JF(\tilde{x})^{-1}F(\tilde{x})||$ is a good estimate of the distance between $\tilde{x}$ and the convergence limit x^*. Therefore we set $Y = JF(\tilde{x})^{-1}$ (computed in floating point arithmetic) and use $I = (\tilde{x}_j \pm |(Y \cdot \square F(x)_i| u^{-\frac{1}{4}})_{j=1,\dots,n}$ where the factor $u^{-\frac{1}{4}}$ accounts for the overestimation by machine interval arithmetic. If I doesn't satisfy the conditions in Theorem 4.5 the procedure is repeated with a smaller unit roundoff u until either a minimal unit roundoff is reached or the certification is successful.

5.3 Producing distinct intervals

Assume now, that the steps in Section 5.2 have been performed for all $x \in X$. We obtain a list of strong interval approximate zeros $I_1, \ldots, I_r \in \mathbb{IC}^n$. In a final step we want to select a subset $M \subset \{1, \ldots, r\}$ such that for all $k, j \in M$, $k \neq j$, the intervals I_k and I_j do not overlap. If two strong interval approximate zeros do not overlap then it is guaranteed that they have distinct associated zeros. This is possible by comparing all intervals pairwise. However, this simple approach requires us to perform $O(r^2)$ interval vector comparisons. For larger problems this quadratic complexity quickly becomes significantly more expensive than the construction and validation of the strong interval approximate zeros.

We therefore employ the following improved scheme to determine all non-overlapping intervals. First, we pick a random point $q \in \mathbb{C}^n$ and compute for each I_k, the squared Euclidean distance between I_k and q resulting in $D = \{d_k \in \mathbb{IR} \,|\, k = 1, \ldots, r\}$. Due to the guarantees of interval arithmetic we have that d_k and d_ℓ overlap if I_k and I_ℓ overlap (but the converse it not necessarily true). It is now possible to compute all overlapping intervals in D in $O(r \log r)$ assuming that not too many intervals overlap. We then check for all overlapping intervals $d_k, d_\ell \in D$ whether I_k and I_ℓ overlap and if so group them accordingly. Finally, this allows us to construct the set M by selecting the intervals which don't overlap with any other and by picking one representative of each cluster of overlapping intervals. The worst case complexity of this procedure is still $O(r^2)$, but in the common case when no or only a small number of intervals overlap the complexity is $O(r \log r)$.

References

[BHSW] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. Bertini: Software for Numerical Algebraic Geometry. Available at bertini.nd.edu with permanent doi: dx.doi.org/10.7274/R0H41PB5.

[BLL19] Michael Burr, Kisun Lee, and Anton Leykin. Effective Certification of Approximate Solutions to Systems of Equations Involving Analytic Functions. In *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation*, ISSAC '19, page 267–274, New York, NY, USA, 2019. Association for Computing Machinery.

[BST20] Paul Breiding, Bernd Sturmfels, and Sascha Timme. 3264 Conics in a Second. *Notices of the American Mathematical Society*, 67:30–37, 2020.

[BT18] Paul Breiding and Sascha Timme. HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In *Mathematical Software – ICMS 2018*, pages 458–465, Cham, 2018. Springer International Publishing.

[Cer] `https://www.juliahomotopycontinuation.org/HomotopyContinuation.jl/stable/certification/#HomotopyContinuation.certify`.

[CLL14] Tianran Chen, Tsung-Lin Lee, and Tien-Yien Li. Hom4PS-3: A Parallel Numerical Solver for Systems of Polynomial Equations Based on Polyhedral Homotopy Continuation Methods. In Hoon Hong and Chee Yap, editors, *Mathematical Software – ICMS 2014*, pages 183–190. Springer Berlin Heidelberg, 2014.

[DHJ+18] Timothy Duff, Cvetelina Hill, Anders Jensen, Kisun Lee, Anton Leykin, and Jeff Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA Journal of Numerical Analysis*, 39(3):1421–1446, 04 2018.

[Dwy51] Paul S. Dwyer. *Linear Computations*. Wiley, 1951.

[GD05] Hatice Gecegormez and Yasar Demirel. Phase stability analysis using interval Newton method with NRTL model. *Fluid Phase Equilibria*, 237(1-2):48—58, 2005.

[GS05] Balajit Gopalan and Jay-Dean Seader. Application of interval Newton's method to chemical engineering problems. *Reliable Computing*, 1(3):215—223, 2005.

[Hig02] Nicholas J. Higham. *Accuracy and stability of numerical algorithms*, volume 80. Siam, 2002.

[HS12] Jonathan D. Hauenstein and Frank Sottile. Algorithm 921: alphaCertified: Certifying Solutions to Polynomial Systems. *ACM Trans. Math. Softw.*, 38(4), Aug 2012.

[Joh17] Frederik Johansson. Arb: efficient arbitrary-precision midpoint-radius interval arithmetic. *IEEE Transactions on Computers*, 66:1281–1292, 2017.

[Kra69] Rudolf Krawczyk. Newton-algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing*, 4(3):187–201, 1969.

[KSS15] Virendra Kumar, Soumen Sen, and Sankar Shome. Inverse Kinematics of Redundant Manipulator using Interval Newton Method. *International Journal of Engineering and Manufacturing*, 2:19—-20, 2015.

[Ley11] Anton Leykin. Numerical Algebraic Geometry for Macaulay2. *The Journal of Software for Algebra and Geometry: Macaulay2*, 3:5–10, 2011.

[Mad73] Kaj Madsen. On the solution of nonlinear equations in interval arithmetic. *BIT Numerical Mathematics*, 13(433):428—433, 1973.

[May17] Günter Mayer. *Interval Analysis*. De Gruyter, Berlin, Boston, 2017.

[Moo59] Ramon E. Moore. Automatic error analysis in digital computation. *Lockheed Missiles and Space Company*, 1959.

[Moo66] Ramon E. Moore. *Interval Analysis*, volume 4. Prentice-Hall, 1966.

[Moo77] Ramon E. Moore. A test for Existence of Solutions to Nonlinear Systems. *SIAM Journal on Numerical Analysis*, 14(4):611–615, 1977.

[Nic71] Karl Nickel. On the Newton method in interval analysis. *Mathematics Research Center Report 1136*, 1971.

[NTI16] Jatin Narula, Abhinav Tiwari, and Oleg A. Igoshin. Role of Autoregulation and Relative Synthesis of Operon Partners in Alternative Sigma Factor Networks. *PLoS Comput. Biology*, 12(12), 2016.

[PM14] Mark M. Plecnik and John M. McCarthy. Numerical Synthesis of Six-Bar Linkages for Mechanical Computation. *Journal of Mechanisms and Robotics*, 6(3), 06 2014. 031012.

[PSK08] Witold Pedrycz, Andrzej Skowron, and Vladik Kreinovich. *Handbook of Granular Computing*. Wiley, 2008.

[RTV97] Felice Ronga, Alberto Tognoli, and Thierry Vust. The number of conics tangent to five given conics: the real case. *Rev. Mat. Univ. Complut. Madrid*, 10:391–421, 1997.

[SH05] Bernito Stradi and Emmanuel Haven. Optimal Investment Strategy via Interval Arithmetic. *International Journal of Theoretical and Applied Finance*, 8(2):185—206, 2005.

[SH10] Bernito Stradi and Emmanuel Haven. The use of interval arithmetic in solving a non-linear rational expectation based multiperiod output-inflation process model: The case of the IN/GB method. *European Journal of Operational Research*, 203(1):222–229, 2010.

[Sma86] Steve Smale. Newton's Method Estimates from Data at One Point. In Richard E. Ewing, Kenneth I. Gross, and Clyde F. Martin, editors, *The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics*, pages 185–196. Springer, 1986.

[Sot] Frank Sottile. 3264 real conics. *www.math.tamu.edu/~sottile/research/stories/3264/*.

[Sot97] Frank Sottile. Enumerative geometry for real varieties. In *Proceedings of Symposia in Pure Mathematics*, volume 62, pages 435–447. American Mathematical Society, 1997.

[Sun58] Teruo Sunaga. Theory of an interval algebra and its application to numerical analysis. *Research Association of Applied Geometry*, 2:29–46, 1958.

[SxZz01] Guo Shu-xiang and Lü Zhen-zhou. Interval Arithmetic and Static Interval finite Element Method. *Applied Mathematics and Mechanics*, 22:1390–1396, 2001.

[Ver99] Jan Verschelde. Algorithm 795: PHCpack: A General-Purpose Solver for Polynomial Systems by Homotopy Continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, June 1999.

[War56] Mieczyslaw Warmus. Calculus of Approximations. *Bulletin l'academie Polonaise des Sciences*, 4(5):253–257, 1956.