DISTANCE MINIMIZATION

FOLLOWING PATHS BACK TO THE ROOTS

CANDAN GUDUCU

Supervisor: SANDOR BOZOKI

A thesis submitted for the degree of

Master in Science

Department of Mathematics,

CEU

Budapest

2017

Contents

Acknowledgements

I would first like to thank my thesis advisor Assoc. Prof. Sandor Bozoki. He has always offered his continuous patience, help and guidance. I would also like to acknowledge Asst. Prof. Pal Hegedus and Prof. Karoly Boroczky for helping me to come to CEU, introducing me to my supervisor and supporting my research interests. My sincere gratitude goes to my dear family and friends whose love and encouragement brought me here today.

CHAPTER 1

Introduction

Solving systems of multi-variate polynomial equations is a classical problem in mathematics, and it has many applications in areas such as robotics (kinematics, motion planning, collision detection, etc.), computer vision (object modeling, surface fitting, recognition, etc.), graphics, geometric modeling (curve and surface intersections), computer-aided design, mechanical design, and optimization. Usually, we need not all but a particular solution of the system of equations for example the solution that optimizes certain conditions.

The problem of computing the solutions of a system of polynomial equations can be approached in various ways, such as algebraic techniques based on Grobner basis computations and numerical methods based on homotopy continuation [**8**]. Moreover, Newton's method is a numerical method to find roots when the number of equations and the number of variables are the same. It finds one solution very fast only if starting from a point which is close enough to the solution.Whereas in homotopy methods, a homotopy is formed between two polynomial systems (one of the systems with known solutions) and the isolated solutions of one are continued to the other.

Additionally, there are several software packages which can solve polynomial systems automatically. The software PHCpack, a general-purpose polynomial system solver that uses homotopy continuation [**14**]. Bertini is a general-purpose polynomial system solver that was created for research about polynomial continuation [**1**]. And also, HOM4PS-2 is a software package for solving polynomial systems by the polyhedral homotopy continuation method [**12**]. Last but not least, Hom4PS-3 is based on Hom4PS-2 and developed by the same team as Hom4PS-2 [**10**].

In my thesis, I study optimization problems as root finding problems. The first order optimality conditions give us the system of multivariate polynomial equations to solve for finding optimum points. I am investigating a point that minimizes the total distance to a given set of circles and some other algebraic curves, writing the corresponding polynomial systems, finding the real roots with the homotopy method (Hom4PS-3) and certify them with alphaCertified.

Why Homotopy Continuation Methods are of interest;

- Suitable to optimization problems,
- Numerical stability,
- Global convergence,
- Locating multiple solutions,
- Containing more information about the solutions,
- Adaptable to find only real solutions. [**5**]

Background is given in Chapter 2. It will contain a brief introduction into the most important concepts and definitions of algebraic geometry, for example Grobner Basis. It also provides an overview of various methods which are used to solve systems of polynomial equations such as Resultants method, and root certification.

Chapter 3 will have information about Homotopy Methods and softwares. The main emphasize will be on Hom4PS-3 which is a parallel software package specialized for solving system of polynomial equations using efficient and reliable numerical methods.

Worked examples and numerical experiments of solving multivariate polynomial systems will be on the Chapter 4. The examples will be of the following type; finding the closest point to given set of circles and some algebraic curves by using previously mentioned algorithms. The examples are chosen due to their enlightening geometric illustrations.

The last chapter provides concluding remarks and directions for further research regarding the results described in the previous chapters of this thesis.

After literature review, I have decided to focus on solving distance minimization problems by Hom4PS-3. Because it was new and claimed to be fast, it was exciting to discover more on this software. However, only its Linux version always works well and the provided interfaces may be deleted or not working sometimes. Many programs such as Maple, MATLAB, Python were needed to find a way around. By gathering background information, giving more explanation, constructing illustrative examples and plotting them, I aim to help other people understand and use the software better.

CHAPTER 2

Background

1. Algebraic Background

Definition 2.1. A commutative ring $\langle R, +, \cdot \rangle$ is a set R with the two binary operations addition $(+)$ and multiplication $(\cdot)$ defined on R such that $(R, +)$ is a commutative group, $\cdot$ is commutative and associative, and the distributive law $a \cdot (b + c) = a \cdot b + a \cdot c$ holds $\forall a, b, c \in R$.

Definition 2.2. Let $\langle R, +, \cdot \rangle$ be a commutative ring with a multiplicative identity. $\langle R, +, \cdot \rangle$ is called a field if every nonzero element of R has a multiplicative inverse in R.

Example 2.3. $\langle \mathbb{Q}, +, \cdot \rangle, \langle \mathbb{R}, +, \cdot \rangle, \langle \mathbb{C}, +, \cdot \rangle$ are fields. However, $\langle \mathbb{Z}, +, \cdot \rangle$ is a ring but not a field.

Definition 2.4. Let $\mathbb{N}$ denote the non-negative integers. Let $\theta = (\theta_1, \ldots, \theta_n)$ be a power vector in $\mathbb{N}^n$, and let $x_1, x_2, \ldots, x_n$ be any n variables. Then, a monomial x^θ in $x_1, x_2, \ldots, x_n$ is defined as the product $x^\theta = x_1^\theta \cdot x_2^\theta \cdot \ldots \cdot x_n^\theta$. Moreover, the total degree of the monomial x^θ is defined as $|\theta| = \theta_1 + \ldots + \theta_n$.

Definition 2.5. A multivariate polynomial f in $x_1, x_2, \ldots, x_n$ with coefficients in a field $\mathbb{K}$ is a finite linear combination, $f(x_1, x_2, \ldots, x_n) = \sum_\theta a_\theta x^\theta$ of monomials x^θ and coefficients $a_\theta \in \mathbb{K}$. The total degree of the polynomial f is defined as the maximum $|\theta|$ such that $a_\theta \neq 0$.

The polynomial ring in the variables $X = \{x_1, \ldots, x_n\}$ over a ring $\mathbb{K}$ is denoted by $\mathbb{K}[X]$ or $\mathbb{K}[x_1, \ldots, x_n]$.

THEOREM 2.6. **(Fundamental Theorem of Algebra)** Every non-zero, single-variable, degree n polynomial with complex coefficients has, counted with multiplicity, exactly n roots.

DEFINITION 2.7. Let $\mathbb{K}$ be a field. A polynomial $f \in \mathbb{K}[x_1, \ldots, x_n]$ is irreducible over $\mathbb{K}$ if it is non-constant and is not the product of two non-constant polynomials in $\mathbb{K}[x_1, \ldots, x_n]$.

PROPOSITION 2.8. Every non-constant polynomial $f \in \mathbb{K}[x_1, \ldots, x_n]$ can be written as a product of polynomials which are irreducible over $\mathbb{K}$.

PROPOSITION 2.9. Let $f \in \mathbb{K}[x_1, \ldots, x_n]$ be irreducible over $\mathbb{K}$ and suppose that f divides the product gh, for $g, h \in \mathbb{K}[x_1, \ldots, x_n]$. Then f divides either g or h.

PROPOSITION 2.10. Let $f, g \in \mathbb{K}[x]$ be polynomials of degree $l \geq 0$ and $m \geq 0$, respectively. Then f and g have a common factor if and only if there are polynomials $A, B \in \mathbb{K}[x]$ such that

- A and B are not both zero,
- A has degree at most $m - 1$ and B has degree at most $l - 1$,
- $Af + Bg = 0$.

In that case, let's write them as

$$f = c_l x^l + \ldots + c_0$$

$$g = d_m x^m + \ldots + d_0$$

$$A = a_{m-1} x^{m-1} + \ldots + a_0$$

$$B = b_{l-1} x^{l-1} + \ldots + b_0$$

Now, $l+m$ coefficients, $a_0, \ldots, a_{m-1}, b_0, \ldots, b_{l-1}$, can be considered as unknown variables. We are looking for $a_i, b_j \in \mathbb{K}$ such that $Af + Bg = 0$.

After plugging in the above expressions for A, B, f, and g, we have the following system of $l+m$ equations and $l+m$ unknowns;

$$\begin{array}{lll} c_l a_{m-1} & +d_m b_{l-1} & = 0 \text{ coefficient of } x^{l+m-1} \\ c_{l-1} a_{m-1} + c_l a_{m-2} & +d_{m-1} b_{l-1} + d_m b_{l-2} & = 0 \text{ coefficient of } x^{l+m-2} \\ \vdots & \vdots & \vdots \\ c_0 a_0 & +d_0 b_0 & = 0 \text{ coefficient of } x^0 \end{array}$$

There is a nonzero solution if and only if the coefficient matrix has zero determinant.

Definition 2.11. The coefficient matrix of the system of equations given above is the Sylvester matrix of f and g with respect to x, denoted by $Syl(f, g, x)$. The resultant of f and g with respect to x, denoted by $Res(f, g, x)$, is the determinant of the Sylvester matrix, $Res(f, g, x) = det(Syl(f, g, x))$.

PROPOSITION 2.12. Given $f, g \in \mathbb{K}[x]$, the resultant $Res(f, g, x) \in \mathbb{K}$ is an integer polynomial in the coefficients of f and g. Moreover, f and g have a common factor in $\mathbb{K}[x]$ if and only if $Res(f, g, x) = 0$

When we have multi-variate polynomials instead of single-variable, the followings give us a bound for number of roots.

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix} = 0 \text{ and } x = (x_1, x_2, \dots, x_n).$$

THEOREM 2.13. **(Bezout's Theorem)** Let f be as square polynomial system and, let $d_1, \dots, d_n$ be the degrees of $f_1, \dots, f_n$ respectively. If $f(x)$ has a finite number of zeros, then the number of its isolated zeros in $\mathbb{C}^n$, counting multiplicities, does not exceed the number $d = d_1 d_2 \dots d_n$.

DEFINITION 2.14. The support S_k of a polynomial f_k in the system (as above) is the set of the exponents of its monomials.

EXAMPLE 2.15. Let $f_1(x, y) = 5x^4 - 2x^3y^2 + y^2 + x$ and $f_2(x, y) = 2x^3 + 5xy^2 - 5xy$. Then, $S_1 = \{(4, 0), (3, 2), (0, 2), (1, 0)\}$, and $S_2 = \{(3, 0), (1, 2), (1, 1)\}$.

DEFINITION 2.16. The Newton polytopes P_k of polynomial f_k is the convex hull of the support S_k.

THEOREM 2.17. **(Bernstein Theorem)** Suppose the polynomial system has finitely many roots in $\mathbb{C}^n$. Then the number of these roots is bounded from above by the mixed volume of its Newton polytopes P_k, for $1 \leq k \leq n$.

The above theorem was discovered by Bernstein (1975) , Khovanski (1978) and Kushnirenko (1976) and is also called as the BKK bound [**2**].

DEFINITION 2.18. Mixed volume of its Newton polytopes $P_k, 1 \leq k \leq n$

$$\mathcal{M}(P_1, \ldots, P_n) = \sum_{i=1}^{n} (-1)^{n-i} \cdot \sum_{\substack{(j_1,\ldots,j_i) \text{ a combination} \\ \text{of } k \text{ indices from } 1,\ldots,n}} vol_i(P_{j_1} \oplus \ldots \oplus P_{j_i})$$

DEFINITION 2.19. Let $\mathbb{K}[X]$ be a polynomial ring in the variables $X = \{x_1, \ldots, x_n\}$. A subring I of $\mathbb{K}[X]$ is called an ideal of $\mathbb{K}[X]$ if

- $0 \in I$,
- if $f, g \in I$, then $f + g \in I$,
- if $f \in I$, then for all h in $\mathbb{K}[X] : hf \in I$.

DEFINITION 2.20. Let I be an ideal of a polynomial ring $\mathbb{K}[X]$. A subset $F = \{f_i\}$ of I is called a generating system of I if $\forall i \in I$ can be written as a sum of the form:

$$i = \sum_{f_i \in F} h_i f_i,$$

where $h_i \in \mathbb{K}[X]$.

DEFINITION 2.21. If an ideal I has finitely many generators, $f_1, \ldots, f_m$ it is said to be finitely generated and $F = \{f_1, \ldots, f_m\}$ is called a basis of I. The ideal generated by $f_1, \ldots, f_m$ will be denoted by $\langle f_1, \ldots, f_m \rangle$.

THEOREM 2.22. **(Hilbert Basis Theorem)** Every ideal in $\mathbb{K}[x_1, \ldots, x_n]$ is finitely generated.

Definition 2.23. Let $S = \{f_1, \ldots, f_m\}$ be polynomials in $\mathbb{K}[X]$. The affine variety of S in $\mathbb{K}^n$ is denoted by $V(S)$ and is defined as follows:

$$V(S) = \{(a_1, \ldots, a_n) \in \mathbb{K}^n : f_i(a_1, \ldots, a_n) = 0 \text{ for all } 1 \leq i \leq m$$

.

Example 2.24. Let $\mathbb{K} = \mathbb{R}$, $X = \{x_1, x_2\}$, $S = \{f_1, f_2\}$ with $f_1 = 2x_1 + x_2 - 7$, and $f_2 = 3x_1 - 4x_2 - 5$. We find the affine variety of S.

$$\begin{aligned}
V(I) &= V(\langle f_1, f_2 \rangle) \\
&= V(\langle 2x_1 + x_2 - 7, 3x_1 - 4x_2 - 5 \rangle) \\
&= V(\langle 2x_1 + x_2 - 7, x_1 - 5x_2 + 2 \rangle) \\
&= V(\langle x_1 + 6x_2 - 9, x_1 - 5x_2 + 2 \rangle) \\
&= V(\langle 11x_2 - 11, x_1 - 5x_2 + 2 \rangle) \\
&= V(\langle x_2 - 1, x_1 - 5x_2 + 2 \rangle) \\
&= V(\langle x_2 - 1, x_1 - 3 \rangle) \\
&= \{(3, 1)\}
\end{aligned}$$

Definition 2.25. Let V be an affine variety in $\mathbb{K}^n$. Then,

$$I(V) = \{f \in \mathbb{K}[X] : f(a_1, \ldots, a_n) = 0 \text{ for all } (a_1, \ldots, a_n) \in V\}$$

is the vanishing ideal of V.

DEFINITION 2.26. The most significant term of a non-zero polynomial p with respect to a monomial ordering $>$ is called the **leading term** $LT_>(p)$ of that polynomial with respect to the chosen ordering. A leading term $LT_>(p)$ of some polynomial p consists of a **leading coefficient** $LC_>(p)$ and the corresponding **leading monomial** $LM_>(p)$ such that $LT_>(p) = LC_>(p)LM_>(p)$.

DEFINITION 2.27. Let X be a set of n variables $x_1, \ldots, x_n$, $\theta = (\theta_1, \ldots, \theta_n)$, and $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{N}^n$.

Lexicographic Order: $\theta >_{lex} \beta$ if and only if the left-most nonzero entry in $\theta - \beta$ is positive. We write $x^\theta >_{lex} x^\beta$ if $\theta >_{lex} \beta$.

Graded Lex Order: $\theta >_{grlex} \beta$ if and only if $|\theta| > |\beta|$ or ($|\theta| = |\beta|$ and $\theta >_{lex} \beta$).

Graded Reverse Lex Order: $\theta >_{grevlex} \beta$ if and only if $|\theta| > |\beta|$ or ($|\theta| = |\beta|$ and the right-most nonzero entry in $\theta - \beta$ is negative).

EXAMPLE 2.28. $(5, 1, 7) >_{grlex} (2, 5, 1)$ since $|(5, 1, 7)| = 13 > 8 = |(2, 5, 1)|$.
$(5, 1, 7) >_{lex} (2, 5, 8)$ since the left-most entry in $(5, 1, 7) - (2, 5, 8) = (3, -4, -1)$ is positive.
$(3, 1, 7) >_{grevlex} (2, 5, 4)$ since $|(3, 1, 7)| = 11 = 11 = |(2, 5, 4)|$ and the right-most entry in $(3, 1, 7) - (2, 5, 4) = (1, -4, 3)$ is positive.

DEFINITION 2.29. Fix a monomial ordering $>$ and let F be a finite generating system of polynomials $\{f_1, \ldots, f_m\} \in \mathbb{K}[X]$. A polynomial $g \in \mathbb{K}[X]$ is reduced with respect to (or modulo) F if no $LM(f_i)$ divides $LM(g)$ for all $i = 1, \ldots, m$. A polynomial $g \in \mathbb{K}[X]$ is completely reduced with respect to (or modulo) F if no monomial of g is divisible by $LM(f_i)$ for all $i = 1, \ldots, m$.

Definition 2.30. Fix a monomial ordering $>$ and let F be a finite generating system of polynomials $\{f_1, \ldots, f_m\} \in \mathbb{K}[X]$. Then every $f \in \mathbb{K}[X]$ can be written as follows $f = a_1 f_1 + a_2 f_2 + \ldots + a_m f_m + r$ where $a_i, r \in \mathbb{K}[X]$ and either the remainder of division $r = 0$ or r is completely reduced with respect to F.

The remainder r is also called the normal form of f and denoted by $f \rightarrow_F^* r$.

The algorithm is taken from [**7**] on page 9.

Algorithm 1: Generalized Division

---

input: A polynomial set $F = \{f_1, \ldots, f_m\}$, and any nonzero polynomial

$f \in \mathbb{K}[x_1, \ldots, x_n]$.

output: The remainder, r, of dividing f by F.

The quotients $q_1, q_2, \ldots, q_s$ such that $f = q_1 f_1 + \ldots + q_s f_s + r$ with either $r = 0$ or

r is a completely reduced polynomial with respect to F.

$q_i = 0$ for $i = 1, \ldots, s$;

$r = 0$;

$p = f$;

while $p \neq 0$ **do**

  $i = 1$; dividing = true; **while** $i \leq s$ *and (dividing)* **do**

    if $LT(f_i)$ *divides* $LT(p)$ **then**

      $u = LT(p)/LT(f_i)$;

      $q_i = q_i + u$;

      $p = p - u \cdot f_i$;

    else

      $i = i + 1$;

      $r = r + LT(p)$;

      $p = p - LT(p)$;

    end

  end

end

EXAMPLE 2.31. Let $g = x^2y - 3xy + y^2 + y + 6$ and $F = \{f_1, f_2\} = \{x - 3, y + 1\}$ with monomial ordering $>_{lex}$. Then, $g - xyf_1 = y^2 + y + 6$ which is not completely reduced yet. $y^2 + y + 6 - yf_2 = 6$ which is not divisible by either $LM(f_1)$ or $LM(f_2)$. So, $g \rightarrow_F^* 6$.

DEFINITION 2.32. Let $I = \langle f_1, \ldots, f_m \rangle$ be an ideal in $\mathbb{K}[X]$, then we define for a given ordering $LT(I) = \{LT(f) : f \in I\}$. The ideal generated by the elements of $LT(I)$ is denoted by $\langle LT(I) \rangle$.

Note that if $I = \langle f_1, \ldots, f_m \rangle$, then $\langle LT(f_1), \ldots, LT(f_m) \rangle$ and $\langle LT(I) \rangle$ may not be equal however $\langle LT(f_1), \ldots, LT(f_m) \rangle \subseteq \langle LT(I) \rangle$.

DEFINITION 2.33. Fix a monomial ordering $>$. A finite subset $G = \{g_1, \ldots, g_m\}$ of an ideal I is said to be a Grobner basis (or standard basis) if $\langle LT(I) \rangle = \langle LT(g_1), \ldots, LT(g_m) \rangle$.

PROPOSITION 2.34. Every nonzero ideal $I \in \mathbb{K}[X]$ has a Grobner basis.

PROPOSITION 2.35. Let G be a Grobner basis for an ideal I in $\mathbb{K}[X]$ and f be a polynomial in $\mathbb{K}[X]$. Then there is a unique $r \in \mathbb{K}[X]$ with the following two properties:

- r is completely reduced with respect to G ,
- there is $h \in I$ such that $f = h + r$.

DEFINITION 2.36. Let f and g be two non-zero polynomials in $\mathbb{K}[X]$, then the $S-$*polynomial* of f and g is defined as follows

$$S(f, g) = \frac{J}{LT(f)} f - \frac{J}{LT(g)} g$$

where $J = LCM(LM(f), LM(g))$ and LCM denotes the Least Common Multiple of two monomials. The LCM is defined as follows: if $x^\theta = LM(f)$ and $x^\beta = LM(g)$, then $J = LCM(x^\theta, x^\beta) = x^\gamma$, with $\gamma_i = max(\theta_i, \beta_i)$ for each $i = 1, \ldots, n$.

EXAMPLE 2.37. Let $F = \{f_1, f_2\}$, where $f_1 = 2x^2yz+xy$ and $f_2 = xyz^2-xz$. Let these polynomials be ordered with lexicographic order. Then, $LM(f_1) = x^2yz$, $LM(f_2) = xyz^2$, and $J = LCM(x^2yz, xyz^2) = x^2yz^2$.

$$
\begin{aligned}
S(f_1, f_2) &= \frac{J}{LT(f_1)} f_1 - \frac{J}{LT(f_2)} f_2 \\
&= \frac{x^2yz^2}{LT(2x^2yz)}(2x^2yz + xy) - \frac{x^2yz^2}{LT(xyz^2)}(xyz^2 - xz) \\
&= \frac{z}{2}(2x^2yz + xy) - x(xyz^2 - xz) \\
&= \frac{2x^2yz^2 + xyz - 2x^2yz^2 + 2x^2z}{2} \\
&= \frac{xyz + 2x^2z}{2}
\end{aligned}
$$

Theorem 2.38. A set $G = \{g_1, \ldots, g_m\}$ is a Grobner basis for the ideal I it generates if and only if $S(g_i, g_j) \to_G^* 0$ for all $g_i, g_j \in G, g_i \neq g_j$

Definition 2.39. A reduced Grobner basis of a polynomial ideal I is a Grobner basis G for the ideal I such that

- $LC(p) = 1$ for all $p \in G$,
- Each $p \in G$ is completely reduced modulo $G - \{p\}$.

Theorem 2.40. Every nonzero polynomial ideal I has a unique reduced Grobner basis with respect to a monomial ordering.

The algorithm is taken from [**7**] on page 11.

Algorithm 2: The Buchberger Algorithm

input: A polynomial set $F = \{f_1, \ldots, f_m\}$ that generates ideal I

output: A Grobner basis $G = \{g_1, \ldots, g_t\}$ that generates ideal I

$G = F$;

$M = \{\{f_i, f_j\} : f_i, f_j \in G, f_i \neq f_j\}$;

while $p \neq 0$ **do**

  $i = 1$;

  dividing = true ;

  while $M \neq \emptyset$ **do**

    $\{p, q\}$ is a pair in M;

    $M = M - \{p, q\}$;

    $S = SPolynomial(p, q)$;

    $h = NormalForm(S, G)$;

    if $h \neq 0)$ **then**

      $M = M \cup \{\{g, h\} : \forall g \in G\}$;

      $G = G \cup \{h\}$;

    end

  end

end

EXAMPLE 2.41. Let $I = \langle f_1 = xy - x, f_2 = x^2 - y\rangle$. $LM(f_1) = xy$, $LM(f_2) = x^2$, and $J_{1,2} = LCM(xy, x^2) = x^2y$.

$$
\begin{aligned}
S(f_1, f_2) &= \frac{J_{1,2}}{LT(f_1)} f_1 - \frac{J_{1,2}}{LT(f_2)} f_2 \\
&= \frac{x^2y}{xy}(xy - x) - \frac{x^2y}{x^2}(x^2 - y) \\
&= -x^2 + y^2
\end{aligned}
$$

Thus, $S(f_1, f_2) = -x^2 + y^2 \rightarrow_F^* = y^2 - y$.

Let $f_3 = y^2 - y$ and $f = \{f_1, f_2, f_3\}$.

$$
\begin{aligned}
S(f_1, f_3) &= \frac{J_{1,3}}{LT(f_1)} f_1 - \frac{J_{1,3}}{LT(f_3)} f_3 \\
&= \frac{xy^2}{xy}(xy - x) - \frac{xy^2}{y^2}(y^2 - y) \\
&= (xy^2 - xy) - (xy^2 - xy) \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
S(f_2, f_3) &= \frac{J_{2,3}}{LT(f_2)} f_2 - \frac{J_{2,3}}{LT(f_3)} f_3 \\
&= \frac{x^2y^2}{x^2}(x^2 - y) - \frac{x^2y^2}{y^2}(y^2 - y) \\
&= x^2y - y^3
\end{aligned}
$$

$S(f_2, f_3) = -x^2y - y^3 = yf_2 - yf_3 \rightarrow_F^* = 0$.

Thus, $\{f_1, f_2, f_3\}$ is a Grobner basis for $I = \langle f_1, f_2 \rangle$

THEOREM 2.42. If $I =< f_1, \ldots, f_m >$ is an ideal in $\mathbb{C}[x_1, \ldots, x_n]$, then the system of polynomial equations $f_1(x_1, \ldots, x_n) = \ldots = f_m(x_1, \ldots, x_n) = 0$ is unsolvable if and only if the reduced Grobner basis of I is $\{1\}$. Thus the variety $V(I) = \emptyset$.

3. Solving polynomial systems of equations

3.1. Algebraic methods.

Grobner basis can be used to solve systems of polynomial equations by eliminating variables from the system one by one during the construction of Grobner basis.

Let $F = \{f_1, \ldots, f_n\}$ be a set of multivariate polynomials and $I =< F >$ be the ideal generated by the polynomial system. Assume that the variables $x_1, \ldots, x_n$ have this lex ordering $x_1 > x_2 > \ldots > x_n$.

To solve $F = 0$, first of all, we need to compute the Grobner basis, G, of I with respect to the lex order. Since the system in Grobner basis form has an upper triangular structure, we start by finding the roots of the generator in x_n which is univariate. By using back-substitution, we can find all roots.

The following examples show that how Grobner Basis can be used for solving a system of equations and determining whether the system is solvable or not [**3**].

Example 2.43. Consider the system of polynomial equations:

$$F = \begin{cases} f_1(x,y,z) = x^2 4xy + 4xy + 3z & = 0 \\ f_2(x,y,z) = y^2 xz & = 0 \\ f_3(x,y,z) = z^2 x + y & = 0 \end{cases}$$

The ideal $I =< f_1, f_2, f_3 >$ is considered. A Grobner basis with a lexicographic monomial ordering for this ideal I is given by $< g_1(x,y,z), g_2(y,z), g_3(y,z), g_4(z) >$:

$$G = \begin{cases} g_1(x,y,z) = x - y - z^2 & = 0 \\ g_2(x,y,z) = y^2 - yz^2 - z & = 0 \\ g_3(x,y,z) = 2yz^2 - z^4 - z^2 & = 0 \\ g_4(x,y,z) = z^6 - 4z^4 - 4z^3 - z^2 & = 0 \end{cases}$$

The solutions for g_4 are $z = -1$, $z = 0$, and $z = 1 \pm \sqrt{2}$. After plugging those solutions in g_3 and solving for y, we get the solutions:

$\{y = 1, z = -1\}, \{y = 2 + \sqrt{2}, z = 1 + \sqrt{2}\}$ and $\{y = 2 - \sqrt{2}, z = 1 - \sqrt{2}\}$.

We can do the same thing with g_2 and obtain five solutions:

$\{y = 1, z = -1\}, \{y = 0, z = 0\}, \{y = 1, z = 0\}, \{y = 2 + \sqrt{2}, z = 1 + \sqrt{2}\}$ and$\{y = 2 - \sqrt{2}, z = 1 - \sqrt{2}\}$

After substituting the solutions above in g_1 and solving for x, we get the solutions:

$$\begin{aligned}
x =&2, & y &= 1, & z &= -1,\\
x =&0, & y &= 0, & z &= 0,\\
x =&1, & y &= 1, & z &= 0,\\
x =&5 + 3\sqrt{2}, & y &= 2 + \sqrt{2}, & z &= 1 + \sqrt{2},\\
x =&5 - 3\sqrt{2}, & y &= 2 - \sqrt{2}, & z &= 1 - \sqrt{2}.
\end{aligned}$$

EXAMPLE 2.44. Consider the following system of polynomial equations:

$$F = \begin{cases} f_1(x, y, z) = x^2y - z & = 0 \\ f_2(x, y, z) = 2xy - 4z - 1 & = 0 \\ f_3(x, y, z) = y^2 - z & = 0 \\ f_4(x, y, z) = x^3 - 4y & = 0 \end{cases}$$

The reduced Grobner basis for $I =< f_1, f_2, f_3, f_4 >$ with respect to the lex ordering is $\{1\}$, hence the system is unsolvable.

3.2. Resultant methods.

Resultant methods work by computing resultants of polynomial pairs with respect to a particular variable and then eliminating an unknown till only one variable remains in a system with finitely many solutions. The Sylvester resultant method is intended to be used for determining the zeros of two polynomials in two variables.

EXAMPLE 2.45. Let's consider a system of two polynomial equations

$$\begin{cases} f(x,y) & = x^2 + x + y - 9 \\ g(x,y) & = 2x^2 + y^2 + 2xy + 3x + 3y - 9 \end{cases}$$

The Sylvester resultant of f and g with respect to y has dimension 3×3.

$$Syl(f,g,y) = \begin{bmatrix} 1 & x^2+x-9 & 0 \\ 0 & 1 & x^2+x-9 \\ 1 & 2x+3 & 2x^2+3x-9 \end{bmatrix}$$

$$\begin{aligned} \det(Syl(f,g,y)) &= x^4 - 20x^2 + 99 \\ &= (x-\sqrt{11})(x+\sqrt{11})(x-3)(x+3) \end{aligned}$$

By plugging in $\{-\sqrt{11}, \sqrt{11}, -3, 3\}$ for x, we can get the y values. Then, the zeros of f and g are $\{(-\sqrt{11}, 2+\sqrt{11}), (\sqrt{11}, 2-\sqrt{11}), (-3,3), (3,-3)\}$.

3.3. Newton's method.

Let $g : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be a function that is differentiable on D,

$$g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_n(x) \end{bmatrix}.$$

The idea behind Newton's Method is to approximate $g(x)$ near the current iterate $x^{(k)}$, and then use the solution as the next iterate $x^{(k+1)}$, after which this process is repeated. $g_k(x)$ is the linear approximation of $g(x)$ at $x(k)$ is a suitable choice is to approximate $g(x)$.

$g_k(x) = g(x^{(k)}) + J_g(x^{(k)})(x - x^{(k)})$

where $J_g(x)$ is the Jacobian matrix of $g(x)$, defined by $[J_g(x)]_{ij} = \dfrac{\partial g_i(x)}{\partial x_j}$. $J_g(x)$ is the matrix of first partial derivatives of the component functions of $g(x)$.

EXAMPLE 2.46. Let $g : D \subseteq \mathbb{R}^2 \to \mathbb{R}^2$,

$$g(x,y) = \begin{bmatrix} g_1(x,y) \\ g_2(x,y) \end{bmatrix} = \begin{bmatrix} x^2 + y^2 - 1 \\ x^4 - y^4 + xy \end{bmatrix}$$

$$J_g(x) = \begin{bmatrix} 2x & 2y \\ 4x^3 + y & -4y^3 + x \end{bmatrix}$$

Let $(x_0, y_0) = (1, 1)$. Then, we got the following by using Matlab for computations,

$$(x_1, y_1) = (0.71429, 0.85714) \text{ and } |g_1(x_1, y_1)| + |g_2(x_1, y_1)| = 2.0000$$

$$(x_2, y_2) = (0.58590, 0.87686) \text{ and } |g_1(x_1, y_1)| + |g_2(x_1, y_1)| = 0.47272$$

$$(x_3, y_3) = (0.56542, 0.87969) \text{ and } |g_1(x_1, y_1)| + |g_2(x_1, y_1)| = 0.057900$$

$$(x_4, y_4) = (0.56498, 0.87971) \text{ and } |g_1(x_1, y_1)| + |g_2(x_1, y_1)| = 1.1904 \cdot 10^{-3}$$

$$(x_5, y_5) = (0.56498, 0.87971) \text{ and } |g_1(x_1, y_1)| + |g_2(x_1, y_1)| = 5.4567 \cdot 10^{-7}$$

$$(x_6, y_6) = (0.56498, 0.87971) \text{ and } |g_1(x_1, y_1)| + |g_2(x_1, y_1)| = 1.1396 \cdot 10^{-13}$$

Then, $(x, y) = (0.56498, 0.87971)$ is a solution with error less than 10^{-12} .

Algorithm 3: Newton's Method

Input

x : Initial point for Newton's method.

t : Tolerance for termination. When the distance between two consecutive iterates is at most t, stop.

Output

xr : Computed root

fr : Function value at xroot

$n = \text{length}(x)$

$y = x - (\text{ones})(n, 1)$

$i = 0$

while $norm(x - y) > t$ **do**

  $y = x$

  $x = x - J(x)^{-1} f(x)$

  $i = i + 1$

end

$xr = x$

$fr = f(x)$

---

4. Optimality Conditions

This section is based on the lecture notes of Tamas Rapcsak [**13**], translated and presented by Sandor Bozoki at CEU.

The classical nonlinear optimization problem is as follows:

$$\min f(x)$$

$$h_j(x) = 0, j = 1, \dots, p,$$

$$g_i(x) \geq 0, i = 1, \dots, m,$$

$$x \in \mathbb{R}^n$$

From here on, it is assumed that f, h_j, g_i for $(j = 1, \dots, p, i = 1, \dots, m)$ are continuously differentiable.

Definition 2.47. The set of feasible solutions of the nonlinear optimization problem: $M[h, g] = \{x \in \mathbb{R}^n | h_j(x) = 0, \text{ for } j = 1, \dots, p, g_i(x) \geq 0, \text{ for } i = 1, \dots, m\}$.

Definition 2.48. The neighborhood of x_0 with radius δ

$$U(x_0, \delta) = \{x \in \mathbb{R}^n | \|x - x_0\| \leq \delta\}$$

Definition 2.49. A point x_0 is called a local optimum of the nonlinear optimization problem if $f(x_0) \leq f(x), \forall x \in U(x_0, \delta) \cap M[h, g]$.

Definition 2.50. A point x_0 is called a global optimum of the nonlinear optimization problem if $f(x_0) \leq f(x), \forall x \in M[h, g]$.

Definition 2.51. The Lagrangian function:

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{j=1}^{p} \mu_j h_j(x) - \sum_{i=1}^{m} \lambda_i g_i(x)$$

where $x \in \mathbb{R}^n, \mu \in \mathbb{R}^p, \lambda \in R^m, \lambda \geq 0$.

Definition 2.52. The linear independence constraint qualification (LICQ) is said to be held at $x_0 \in M[h, g]$ if the vectors

$$\nabla h_j(x_0), j = 1, \ldots p,$$

$$\nabla g_i(x_0), i \in I(x_0) = \{i \quad | g_i(x_0) = 0, i = 1, \ldots, m\}$$

are linearly independent. The index set $I(x_0)$ denotes the set of active inequality constraints.

Theorem 2.53. If x_0 is a local optimum of the classical nonlinear optimization problem and LICQ holds at this point, then there exist vectors $\mu \in \mathbb{R}^p$ and $\lambda \in \mathbb{R}^m$ (Lagrange multipliers) such that the conditions

$$\nabla_x L(x_0, \mu, \lambda) = 0,$$

$$\lambda \geq 0,$$

$$\lambda_i g_i(x_0) = 0, i = 1, \ldots, m,$$

fulfill. The first order conditions are called Karush-Kuhn-Tucker(KKT) conditions.

Suppose that we find such points x_0, μ, λ that satisfy the conditions above. Then, we can determine the nature of x_0 by Second Derivative Test.

- If the Hessian $HL(x_0, \mu, \lambda)$ is positive definite, then f a local minimum at x_0.

- If the Hessian $HL(x_0, \mu, \lambda)$ is negative definite, then f a local maximum at x_0.
- If $\det HL(x_0, \mu, \lambda) \neq 0$ but neither of the statements above hold, then f has a saddle point at x_0.

THEOREM 2.54. Suppose that the objective function and constraint functions of the classical nonlinear optimization problem are twice continuously differentiable. If x_0 is a local optimum and LICQ holds at this point, then there exist vectors $\mu \in \mathbb{R}^p$ and $\lambda \in \mathbb{R}^m$ (Lagrange multipliers) such that the conditions

$$\nabla_x L\mathcal{L}_x(x_0, \mu, \lambda) = 0,$$

$$\lambda \geq 0,$$

$$\lambda_i g_i(x_0) = 0, \text{ for } i = 1, \ldots, m,$$

$$v^T HL(x_0, \mu, \lambda)v \geq 0, \quad v \in T\tilde{M}[h, g]_{x_0}$$

fulfill, where

$$T\tilde{M}[h, g]_{x_0} = \{v \in \mathbb{R}^n \quad |\nabla h_j(x_0)v = 0 \quad j = 1, \ldots, p \quad \nabla g_i(x_0)v = 0 \quad i \in I(x_0)\}$$

and $HL(x_0, \mu, \lambda)$ denotes the Hessian of the Lagrangian function with respect to variables x in point x_0.

5. Root Certification

Symbolic computation for solving systems of both univariate and multivariate polynomial equations has advanced exponentially. With emergence of new approaches, the limits of solving systems of polynomial equations were pushed ahead. The recent implementations of numerical homotopy algorithms such as PHCpack, Bertini and Hom4PS-X reliably approximate solutions numerically and hence solve even large scale systems of polynomial equations. However, the outputs of those implementations are not certified.

Lajos Szilassi said "Mathematics can easily be mystified if mathematicians pretend to be magicians by presenting a statement (whether true or false) like the magicians take a rabbit from a stovepipe- hat to the great amazement of the audience." in his paper where he demonstrates why mathematicians must be careful while investigating an unknown geometric construction and Thoughts concerning the analysis of Erdelys Spidron System [**9**]. Yet, we can not pretend to be magicians. We need to check the validity of the statements. In this particular case, we need to certify the roots which were found by aforementioned software packages.

This upcoming sections contain basic information about Smale's alpha-theory and alphaCertified. The more detailed information can be found at [**4**].

5.1. Smale's alpha-theory.

Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a system of polynomials with zeros $Z(f) := \{\theta \in \mathbb{C}^n | f(\theta) = 0\}$, and let $J(x)$ be the Jacobian matrix of the system f at point x.

Consider the map $N_f : \mathbb{C}^n \to \mathbb{C}^n$ defined by

$$N_f(x) := \begin{cases} x - J(x)^{-1}f(x) & \text{if } J(x) \text{ is invertible,} \\ x & \text{otherwise.} \end{cases}$$

$N_f(x)$ is called the Newton iteration of f starting at x. For $k \in \mathbb{N}$, let

$$N_f^k(x) := \underbrace{N_f \circ \ldots \circ N_f}_{\text{k times}}(x)$$

be the k^{th} Newton iteration of f starting at x.

Definition 2.55. Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system. A point $x \in \mathbb{C}^n$ is an approximate solution to f with associated solution $\theta \in Z(f)$ if, for every $k \in \mathbb{N}$,

$$\|N_f^k(x) - \theta\| \leq \left(\frac{1}{2}\right)^{2^k - 1} \|x - \theta\|$$

Let's define the constants for Smale's alpha theory;

$$\alpha(f,x) = \beta(f,x)\gamma(f,x)$$

$$\beta(f,x) = \|x - N_f^k(x)\| = \|J(x)^{-1}f(x)\|$$

$$\gamma(f,x) = \sup_{k \geq 2} \|\frac{J(x)^{-1}D^k f(x)}{k!}\|$$

where $D^k f$ is k^{th} derivative of f.

Theorem 2.56. Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system. If a point $x \in \mathbb{C}^n$ with

$$\alpha(f,x) < \frac{13 - 3\sqrt{17}}{4} \approx 0.157671$$

then x is an approximate solution to f. Additionally, $\|x - \theta\| \leq 2\beta(f, x)$ where $\theta \in Z(f)$ is the associated solution to x [**6**].

5.2. alphaCertified.

alphaCertified is a program that implements elements of alpha-theory to certify numerical solutions to systems of polynomial equations. For a given finite set of points $X \subset \mathbb{C}^n$, it tells us that Newton's method converge quadratically from which points of X to some solution of f, to distinct solutions of f and Newton's method converge quadratically from which points of X to real solutions of f if f is real.

CHAPTER 3

Homotopy Continuation Method and Softwares

1. Homotopy Continuation Method

Suppose the original system of equations is denoted by $F(x_1, \ldots, x_n)$ and a parameter t is introduced where $0 \leq t \leq 1$. Define a homotopy map as follows

$$H(x_1, \ldots, x_n, t) = tF(x_1, \ldots, x_n) + (1-t)G(x_1, \ldots, x_n)$$

where $G(x_1, \ldots, x_n)$ is chosen to be an already solved or easily solvable system of polynomial equations.

Note that $H(x_1, \ldots, x_n, 0) = G(x_1, \ldots, x_n)$ and $H(x_1, \ldots, x_n, 1) = F(x_1, \ldots, x_n)$.

By following the paths of the solutions $(s_1, \ldots, s_n)$ of $H(s_1, \ldots, s_n, t)$ when the parameter t varies from 0 to 1, the solutions of the original system of equations can be obtained.

$h(x,t) = 0$ implicitly defines a curve $x = x(t)$ over $[0,1]$. This is one solution curve to the family of equations $h(x,t) = 0$ parametrized with t. Starting at $t = 0$ and $x(0) = x_0$ where x_0 is root of $g(x)$, we would like to stay on the solution curve $x(t)$ such that $h(x,t) = 0$ holds for any t, until $t = 1$ is reached.

This is can be achieved by numerically solving the following ODE:
$\dfrac{\partial h(x,t)}{dt} = 0$
with the initial value $x(0) = x_0$.

Expanding the equation, $\frac{\partial h}{\partial x}\frac{\partial x)}{\partial t} + \frac{\partial h}{\partial t} = 0$

where $\frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix}$ and $\frac{\partial h}{\partial t} = f - g$

Homotopy methods in use are numerically stable methods and the deformed systems H are commonly solved by iterative numerical methods like Newton methods.

EXAMPLE 3.1. [**3**] Suppose the original system of equations $F(x, y)$ is defined as:

$$F(x) = \begin{cases} f_1(x, y) & = 3x^3 - 4x^2y - x + y^2 + 2y^3 = 0 \\ f_2(x, y) & = -6x^3 + 2xy - 5y^3 - y = 0 \end{cases}$$

$$\text{let } G(x) = \begin{cases} g_1(x, y) & = x^3 - 1 - y = 0 \\ g_2(x, y) & = y^3 - 8 - x = 0 \end{cases}$$

Let us define the homotopy map as in the previous definition.

$$H(x, y, t) = tF(x, y) + (1 - t)G(x, y) = 0$$

$$H(x, y, t) = \begin{cases} h_1(x, y) = tf_1(x, y) + (1 - t)g_1(x, y) = 0 \\ h_2(x, y) = tf_2(x, y) + (1 - t)g_2(x, y) = 0 \end{cases}$$

Note that $H(x, y, 0) = G(x, y)$ and $H(x, y, 1) = F(x, y)$.

The 9 solutions of system $G(x, y)$ are known and as follows;

$$(-0.2267 + 1.0743i, 0.2267 - 1.0743i),$$

$$(-0.2267 - 1.0743i, -0.2267 + 1.0743i),$$

$$(-0.7052 + 1.0387i, 0.9319 + 0.4290i),$$

$$(-0.7052 - 1.0387i, 0.9319 - 0.4290i),$$

$$(0.9319 - 0.4290i, -0.7052 - 1.0387i),$$

$$(0.9319 + 0.4290i, -0.7052 + 1.0387i),$$

$$(-0.6624 + 0.5623i, -0.6624 + 0.5623i),$$

$$(-0.6624 - 0.5623i, -0.6624 - 0.5623i),$$

$$\text{and } (1.3247, 1.3247).$$

The solutions of system $F(x, y) = 0$ are

$$(0.4757 + 0.6118i, -0.2171 + 0.7286i),$$

$$(0.4757 - 0.6118i, -0.2171 - 0.7286i),$$

$$(-0.07632 + 0.1989i, -0.05853 - 0.4920i),$$

$$(-0.07632 - 0.1989i, -0.05853 + 0.4920i),$$

$$(0.04140 + 0.4376i, 0.3593 + 0.3717i),$$

$$(0.04140 - 0.4376i, 0.3593 - 0.3717i),$$

$$(0.3650, -0.3418), (-0.4862, 0.2886), \text{ and } (0, 0).$$

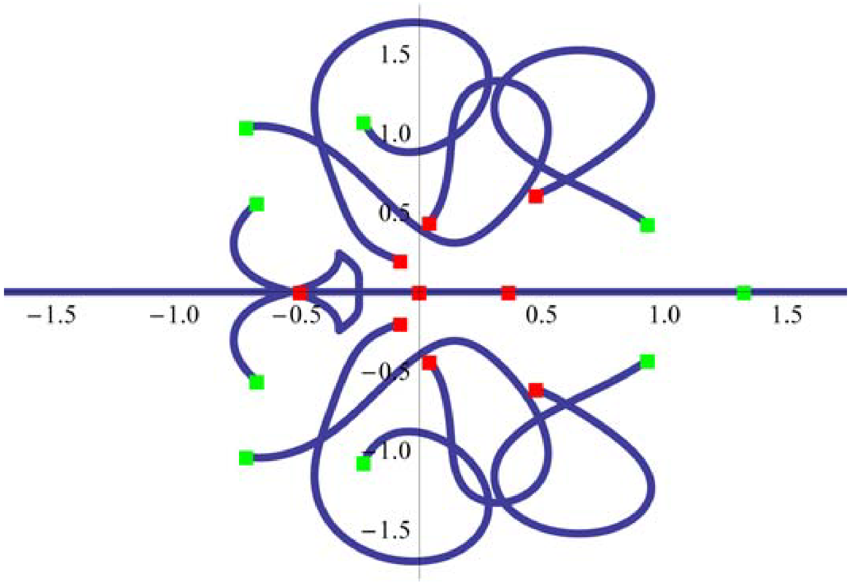

FIGURE 1. x values of the solution paths of $H(x, y, t) = 0$ [**3**]

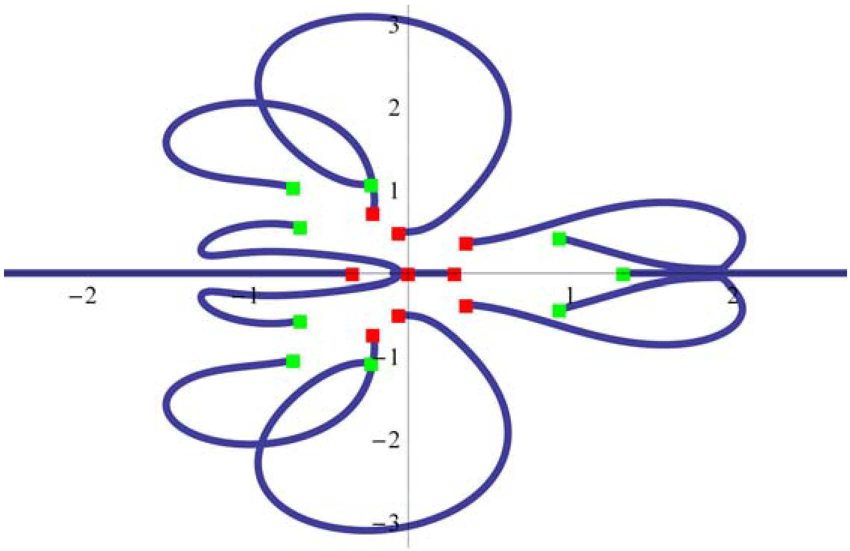

FIGURE 2. y values of the solution paths of $H(x, y, t) = 0$ [**3**]

2. Hom4PS-3

Hom4PS-3 is a parallel software package specialized for solving system of polynomial equations using efficient and reliable numerical methods. Hom4PS-3 is based on the polyhedral homotopy method by Huber and Sturmfels [**10**].

$$f(x) = \begin{cases} f_1(x) & = \sum_{a \in S_1} c_{1,a} x^a \\ \vdots & \\ f_n(x) & = \sum_{a \in S_n} c_{n,a} x^a \end{cases}$$

where $x = (x_1, \ldots, x_n)$, $a = (a_1, \ldots, a_n) \in (\mathbb{N} \cup \{0\})^n$, and $x^a = x^{a_1}, \ldots, x^{a_n}$. Here S_j is the support of $f_j(x)$.

If the system of polynomial equations with generic coefficients is solved, then it can be used to solve the system with particular coefficients with the same supports by the Cheater's homotopy. More information about Cheater's homotopy can be found at Li (1989) [**11**]. To solve $f(x)$ as above, it uses the following homotopy;

$$H(x,t) = \begin{cases} h_1(x,t) & = \sum_{a \in S_1} c_{1,a} x^a t^{\omega_1(a)} \\ \vdots & \\ h_n(x,t) & = \sum_{a \in S_n} c_{n,a} x^a t^{\omega_n(a)} \end{cases}$$

with lifting functions $\omega_1, \ldots, \omega_n$, where each $\omega_k : S_k \to \mathbb{Q}$ has randomly chosen images. Notice that $H(x,1) = P(x)$. For $a \in S_k$, $= (a, \omega_k(a))$.

3. Other Softwares

The following part contains information about other softwares. Moreover, CPU comparisons for Other Software vs Hom4PS-2, and Hom4PS-2 vs Hom4PS-3.

3.1. PHCpack.

PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. It follows the steps in the figure below.

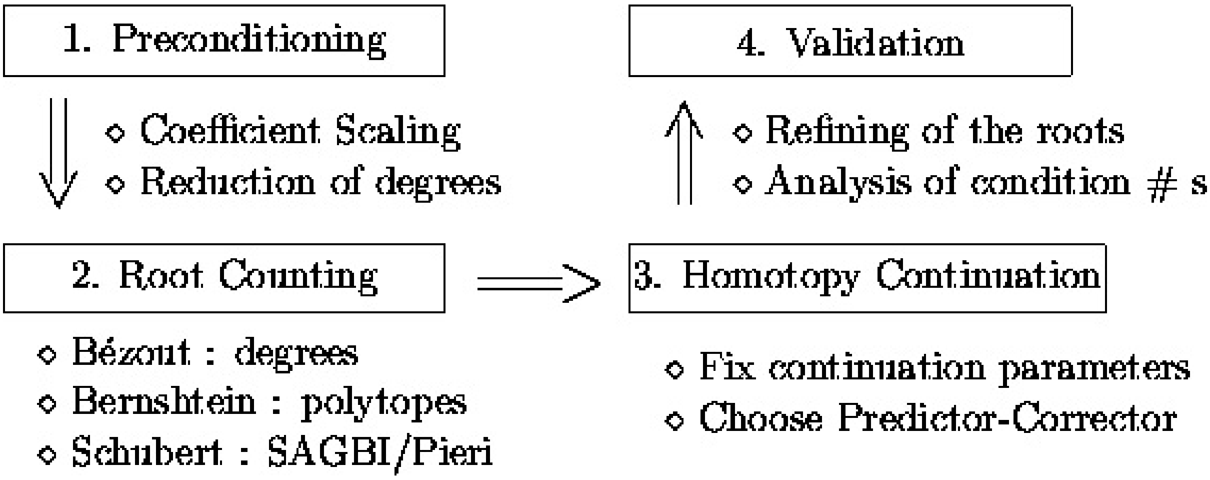

FIGURE 3. PHCpack steps [**14**]

Because of being available on different platforms, we were not able to run PHCpack and Hom4PS-3 on the same machine. The next figure gives CPU time comparisons of PHCpack and Hom4PS-2 for the text systems [**12**]. Later on, the table of CPU time comparisons of Hom4PS-2 and Hom4PS-3. This way, we can compare them indirectly.

System	Total degree	CPU time		Speed-up ratio	# of solutions obtained
		PHCpack	HOM4PS-2.0		
eco-14	1,062,882	1 h 26 m 04 s	52.9 s	97.6	4,096
eco-15	3,188,646	3 h 55 m 23 s	2 m 25 s	97.4	8,192
eco-17	28,697,814	–	22 m 23 s	–	32,768
eco-18	86,093,442	–	1 h 51 m 30 s	–	65,536
noon-9	19,683	33 m 28 s	22.2 s	90.5	19,665
noon-10	59,049	2 h 33 m 27 s	1 m 27 s	105.8	59,029
noon-11	177,147	–	5 m 32 s	–	177,125
noon-13	1,594,323	–	3 h 7 m 10 s	–	1,594,297
katsura-14	16,384	2 h 49 m 00 s	2 m 52 s	59.0	16,384
katsura-15	32,768	8 h 22 m 45 s	7 m 03 s	71.3	32,768
katsura-16	65,536	–	16 m 25 s	–	65,536
katsura-20	1,048,576	–	8 h 58 m 00 s	–	1,048,576
cyclic-9	362,880	3 h 50 m 48 s	44 s	314.7	6,642
cyclic-10	3,628,800	11 h 00 m 23 s	2 m 47 s	237.2	34,940
cyclic-11	39,916,800	–	19 m 40 s	–	184,756
cyclic-12	479,001,600	–	1 h 36 m 40 s	–	374,330
reimer-6	5,040	15 m 08 s	9.6 s	94.5	576
reimer-7	40,320	3 h 45 m 43 s	1 m 58 s	114.7	2,880
reimer-8	362,880	–	30 m 43 s	–	14,400
reimer-9	3,628,800	–	7 h 52 m 40 s	–	86,400

FIGURE 4. PHC vs HOM4PS2according to [**12**]

3.2. Bertini.

Bertini is a general-purpose polynomial system solver that was created for research about polynomial continuation. It goes through the following steps in essence [**1**].

Step 1: symbolize relevant variables and set the homotopy H

Step 2: Construct the system of ODEs

Step 3: Do the path-tracking via ODE-solving

Similarly to PHCpack, we were not able to run Bertini and Hom4PS-3 on the same machine. We use the same idea as before. The next figure gives CPU time comparisons of Bertini and Hom4PS-2 for the text systems [**12**].

System	Total degree	CPU time		Speed-up ratio	# of solutions obtain
		Bertini	HOM4PS-2.0		
eco-11	39,366	37 m 02 s	3 m 26 s	10.8	512
eco-12	118,098	2 h 16 m 25 s	12 m 56 s	10.5	1,024
eco-13	354,294	8 h 07 m 16 s	45 m 22 s	10.7	2,048
eco-14	1,062,882	–	2 h 48 m 42 s	–	4,096
noon-10	59,049	15 m 50 s	1 m 27 s	10.9	59,029
noon-11	177,147	1 h 17 m 59 s	5 m 32 s	14.1	177,125
noon-12	531,441	6 h 35 m 23 s	27 m 29 s	14.4	531,417
noon-13	1,594,323	–	3 h 07 m 10 s	–	1,594,297
katsura-16	65,536	4 h 04 m 14 s	16 m 25 s	14.9	65,536
katsura-17	131,072	10 h 42 m 57 s	40 m 48 s	15.8	131,072
katsura-18	262,144	–	1 h 35 m 47 s	–	262,144
katsura-20	1,048,576	–	8 h 58 m 00 s	–	1,048,576
cyclic-7	5,040	4 m 42 s	13 s	21.7	924
cyclic-8	40,320	1 h 11 m 55 s	2 m 43 s	26.5	2,048
cyclic-9	362,880	–	33 m 30 s	–	6,642
reimer-6	5,040	8 m 28 s	9.5 s	53.5	576
reimer-7	40,320	1 h 45 m 47 s	1 m 58 s	53.8	2,880
reimer-8	362,880	–	30 m 43 s	–	14,400
reimer-9	3,628,800	–	7 h 52 m 40 s	–	86,400

FIGURE 5. Bertini vs HOM4PS2 according to [**12**]

3.3. Hom4PS-2.

HOM4PS-2.0 is a software package which implements the polyhedral homotopy continuation method for solving polynomial systems. It has three major improvement over HOM4PS; new method for finding mixed cells, combining the polyhedral homotopy and linear homotopy in one step, and new way of dealing with the curve jumping.[**12**]

Table 1. Hom4PS-2 vs Hom4PS-3 based on own experience

System	Total Degree	CPU time(seconds)	
		Hom4PS-2	Hom4PS-3
cyclic-10	3628800	92.80800	130.13
eco-17	28697814	531.91600	170.44
katsura-15	32768	2898.43200	469.44
noon-10	59049	178.72800	60.60
reimer-7	40320	97.58000	34.27

CHAPTER 4

Numerical Experiments

EXAMPLE 4.1. Let's find the closest point to the given curves where the distance is defined as $d((x_1, y_1), (x_2, y_2)) = (x_1 - x_2)^2 + (y_1 - y_2)^2$. The curves:

$$5x_1^2 + 30x_1 - 4y_1^2 + 32y_1 - 119 = 0,$$
$$x_2^2 + 4x_2 + y_2^2 - 14y_2 + 48 = 0,$$
$$2x_3^2 + 20x_3 + y_3^2 + 4y_3 + 44 = 0,$$
$$x_4^2 - 2x_4 - 5y_4 + 1 = 0.$$

Then, our objective function that we want to minimize is

$$F(v) = (x-x_1)^2+(y-y_1)^2+(x-x_2)^2+(y-y_2)^2+(x-x_3)^2+(y-y_3)^2+(x-x_4)^2+(y-y_4)^2$$

where $v = (x, y, x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$.

We can consider the curves as our equality constraints in the classical optimization problem. By KKT conditions, we need to solve the square polynomial system with fourteen unknowns below.

$$\nabla_v \mathcal{L}(v, \mu) = \begin{bmatrix} 8x - 2x_1 - 2x_2 - 2x_3 - 2x_4 \\ 8y - 2y_1 - 2y_2 - 2y_3 - 2y_4 \\ -2x + 2x_1 + 10x_1\mu_1 + 30\mu_1 \\ -2x + 2x_2 + 2x_2\mu_2 + 4\mu_2 \\ -2x + 2x_3 + 4x_3\mu_3 + 20\mu_3 \\ -2x + 2x_4 + 2x_4\mu_4 - 2\mu_4 \\ -2y + 2y_1 - 8y_1\mu_1 + 32\mu_1 \\ -2y + 2y_2 + 2y_2\mu_2 - 14\mu_2 \\ -2y + 2y_3 + 2y_3\mu_3 + 4\mu_3 \\ -2y + 2y_4 - 5\mu_4 \\ 5x_1^2 + 30x_1 - 4y_1^2 + 32y_1 - 119 \\ x_2^2 + 4x_2 + y_2^2 - 14y_2 + 48 \\ 2x_3^2 + 20x_3 + y_3^2 + 4y_3 + 44 \\ x_4^2 - 2x_4 - 5y_4 + 1 \end{bmatrix} = 0$$

After running Hom4PS-3 and certifying the root with alphaCertified, we get $(-4.8792, 3.6252)$ as the solution.

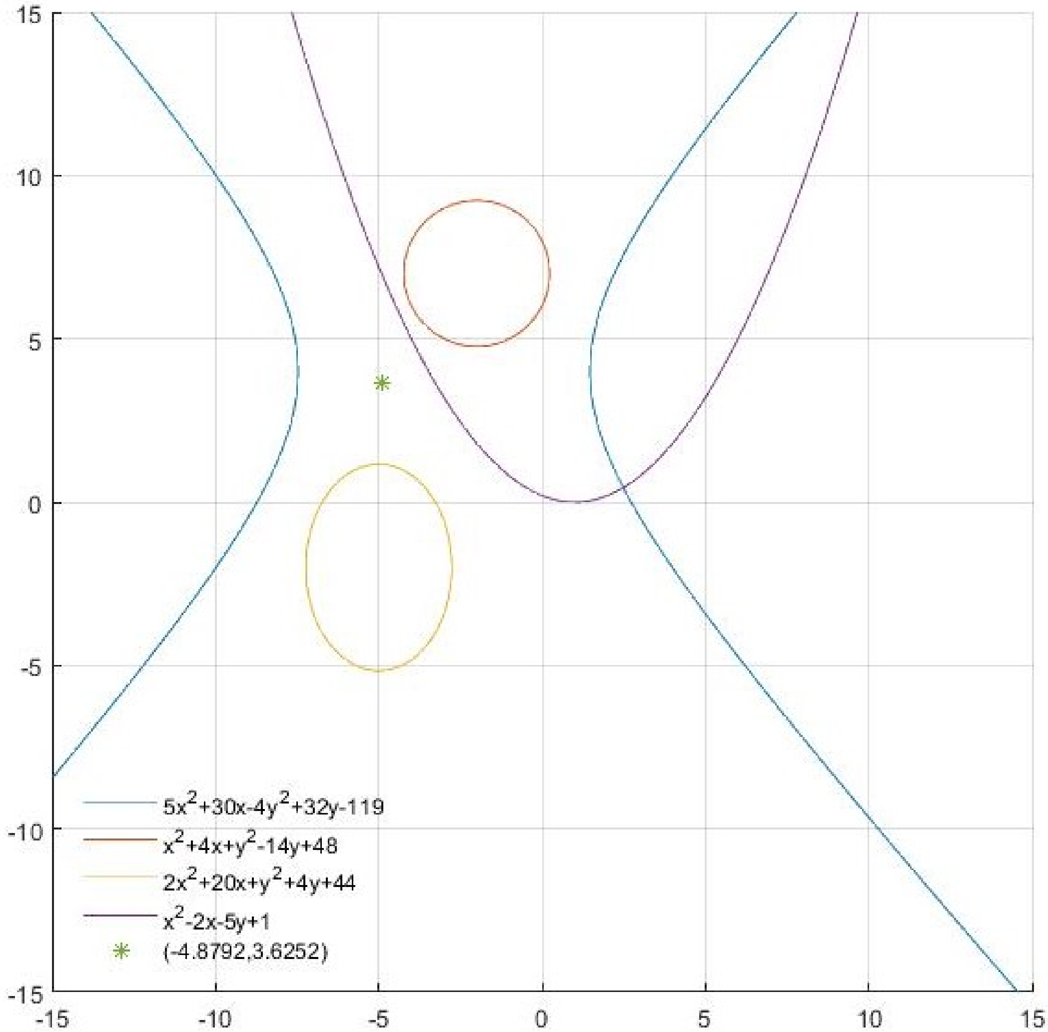

FIGURE 1. Finding the minimal total (squared) distance from four algebraic curves in Example 4.1

EXAMPLE 4.2. Let's find the closest point to the given circles where the distance is defined as before. The curves:

$$x_1^2 + y_1^2 - 1 = 0$$

$$x_2^2 - 4x_2 + 7 + y_2^2 - 4y_2 = 0$$

$$x_3^2 + 4x_3 + 7 + y_3^2 - 4y_3 = 0$$

Then, our objective function that we want to minimize is

$$F(v) = (x - x_1)^2 + (y - y_1)^2 + (x - x_2)^2 + (y - y_2)^2 + (x - x_3)^2 + (y - y_3)^2.$$

where $v = (x, y, x_1, x_2, x_3, y_1, y_2, y_3)$.

We can consider the curves as our equality constraints in the classical optimization problem. By KKT conditions, we need to solve the square polynomial system with eleven unknowns below.

$$\nabla_v \mathcal{L}(v, \mu) = \begin{bmatrix} 6x - 2x_1 - 2x_2 - 2x_3 \\ 6y - 2y_1 - 2y_2 - 2y_3 \\ -2x + 2x_1 + 2\mu_1 x_1 \\ -2x + 2x_2 + 2\mu_2 x_2 - 4\mu_2 \\ -2x + 2x_3 + 2\mu_3 x_3 + 4\mu_3 \\ -2y + 2y_1 + 2\mu_1 y_1 \\ -2y + 2y_2 + 2\mu_2 y_2 - 4\mu_2 \\ -2y + 2y_3 + 2\mu_3 y_3 - 4\mu_3 \\ x_1^2 + y_1^2 - 1 \\ x_2^2 - 4x_2 + 7 + y_2^2 - 4y_2 \\ x_3^2 + 4x_3 + 7 + y_3^2 - 4y_3 \end{bmatrix} = 0$$

After running Hom4PS-3 and certifying the root with alphaCertified again, we confirm that $(0, 1.5072)$ is a solution.

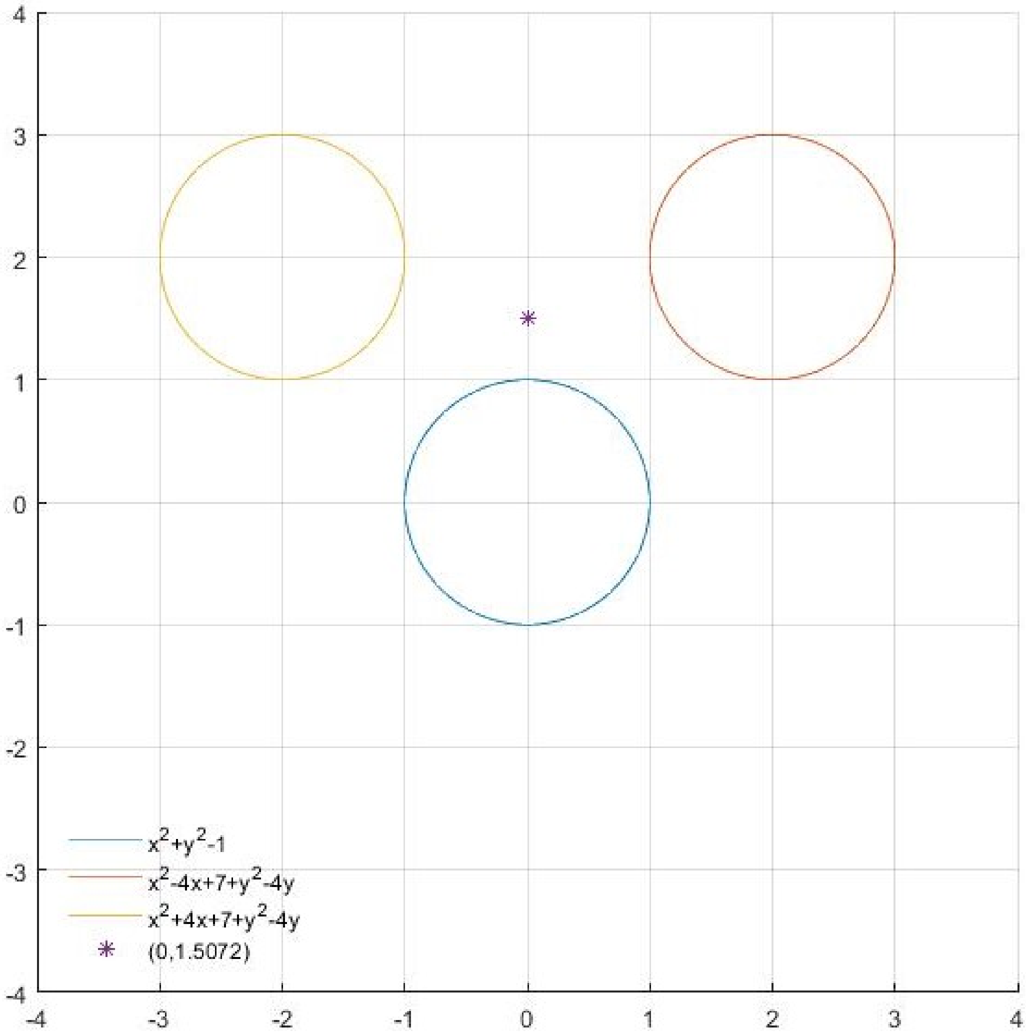

Figure 2. Finding the minimal total (squared) distance from three circles in Example 4.2

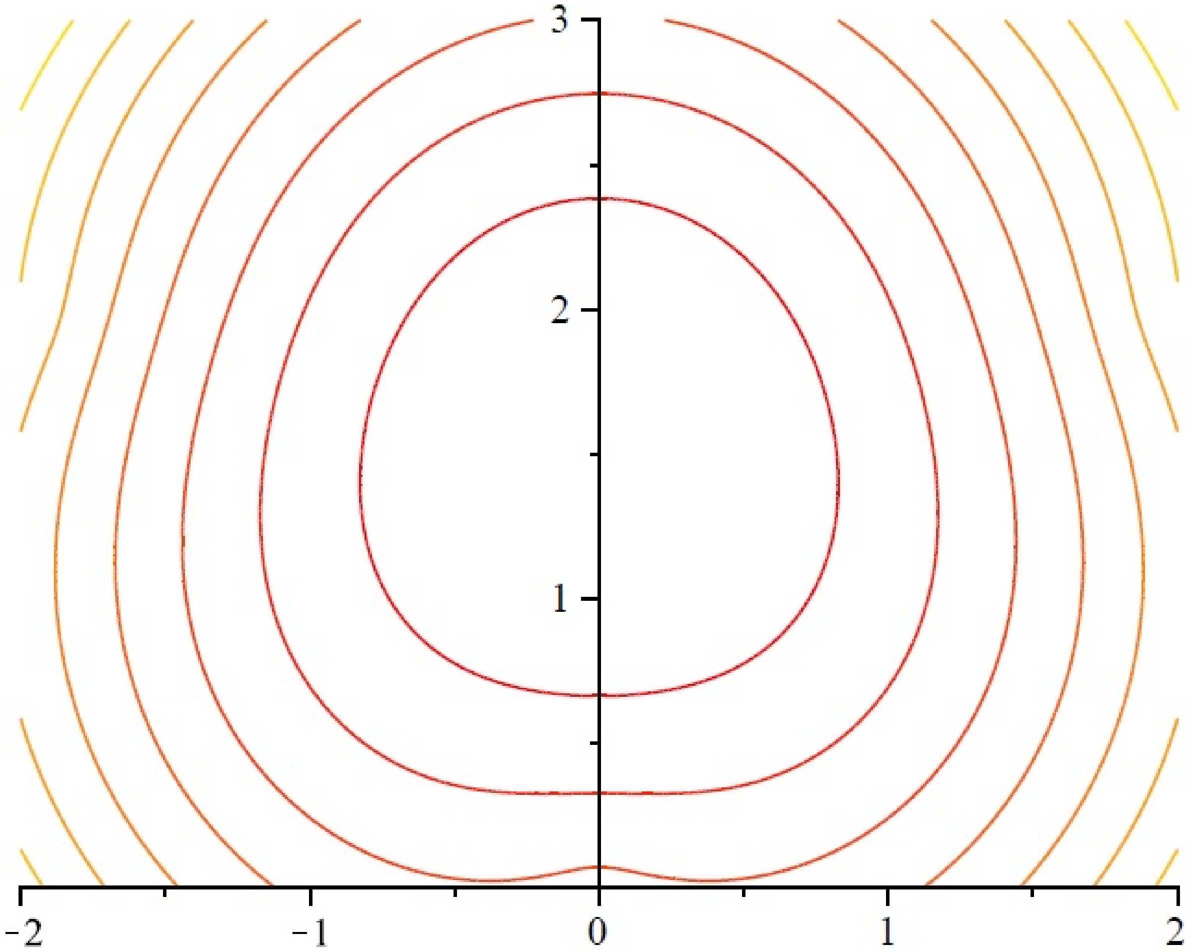

FIGURE 3. The contour plot of objective Example 4.2

EXAMPLE 4.3. Let's find the closest point to the given curves where the distance is defined as before. The curves:

$$x_1^2 - 9y_1^2 - 49 = 0$$

$$x_2^2 - y_2 - 2 = 0$$

Then, our objective function that we want to minimize is

$$F = (x - x_1)^2 + (y - y_1)^2 + (x - x_2)^2 + (y - y_2)^2$$

We again consider the curves as our equality constraints in the classical optimization problem. By KKT conditions, we need to solve the square polynomial system with fourteen unknowns below.

$$\nabla_v \mathcal{L}(v,\mu) = \begin{bmatrix} 6x - 2x_1 - 2x_2 \\ 6y - 2y_1 - 2y_2 \\ -2x + 2x_1 + 2x_1\mu_1 \\ -2x + 2x_2 + 2x_2\mu_2 \\ -2y + 2y_1 - 18y_1\mu_1 \\ -2y + 2y_2 - \mu_2 \\ x_1^2 - 9y1^2 - 49 \\ x_2^2 - y_2 - 2 \end{bmatrix} = 0$$

After running Hom4PS-3 and certifying the root with alphaCertified, we get $(-4.4580, 0.8869)$ and $(4.4580, 0.8869)$ as solutions. A vital advantage of homotopy continuation methods is being able to find all roots.

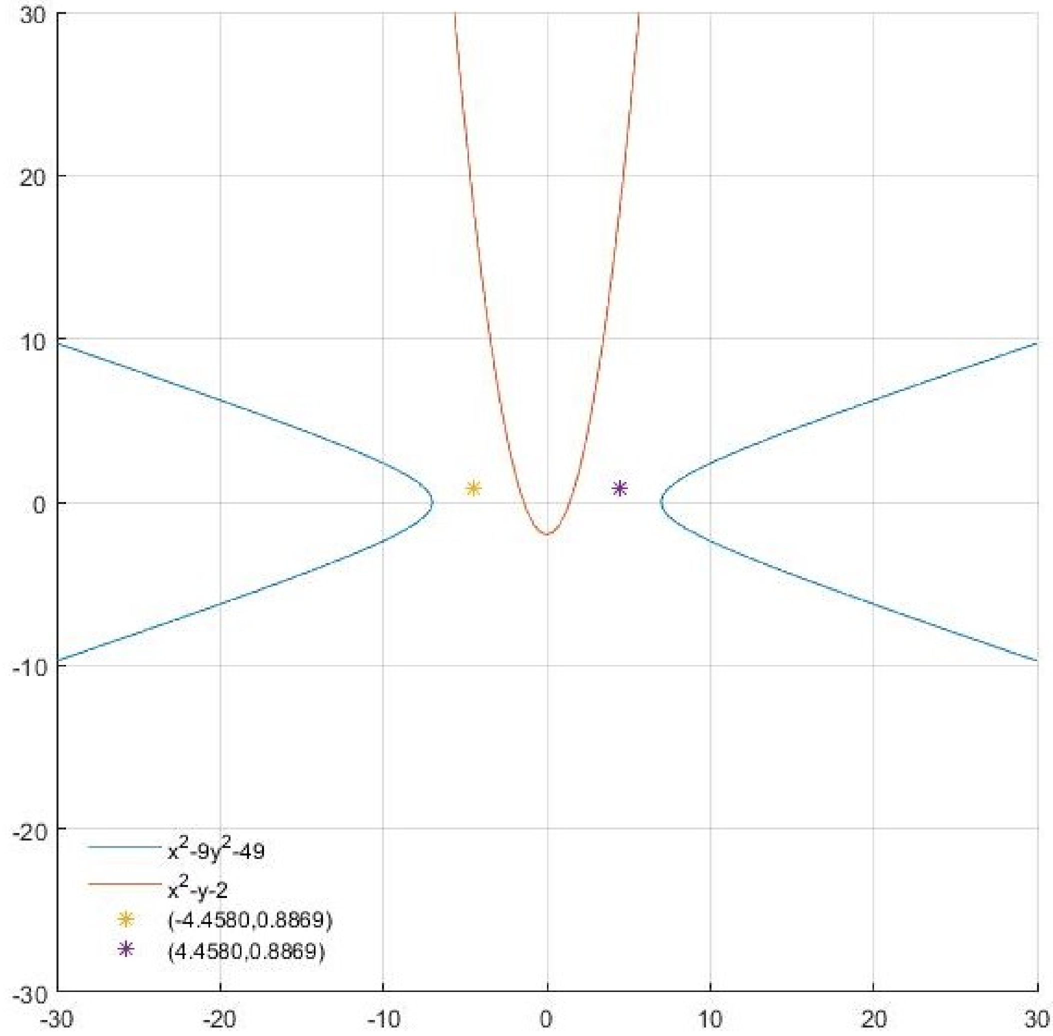

FIGURE 4. Finding the minimal total (squared) distance from two algebraic curves in Example 4.3

CHAPTER 5

Conclusion

We have covered how to solve multi-variate polynomial system of equations. Related softwares and their basic working principle are given. In continuation, Hom4PS-3 discussed more detailed. Moreover, worked examples and numerical experiments of solving multivariate polynomial systems are stated. We have seen how to solve an optimization problem by transforming to system of equations, solve them with Hom4PS-3, verify the solutions with alphaCertified, and plot them with MATLAB.

CHAPTER 6

Appendices

```
> with(alphaCertifiedMaple) ;
  alphaPath := "D:/alphaCertified" :
  Polys := [8·x - 2·x1 - 2·x2 - 2·x3 - 2·x4 , 8·y - 2·y1 - 2·y2 - 2·y3 - 2·y4 ,
      -2·x + 2·x1 + 10·x1·m1 + 30·m1 , -2·x + 2·x2 + 2·x2·m2 + 4·m2 ,
      -2·x + 2·x3 + 4·x3·m3 + 20·m3 , -2·x + 2·x4 + 2·x4·m4 -2·m4 , -2·y
      + 2·y1 - 8·y1·m1 + 32·m1 , -2·y + 2·y2 + 2·y2·m2 - 14·m2 , -2·y
      + 2·y3 + 2·y3·m3 + 4·m3 , -2·y + 2·y4 - 5·m4 , 5·x1^2 + 30·x1 -4
      ·y1^2 + 32·y1 -119 , x2^2 + 4·x2 + y2^2 -14·y2 + 48 , 2·x3^2 + 20
      ·x3 + y3^2 + 4·y3 + 44 , x4^2 - 2·x4 -5·y4 + 1] :
  Vars := [ x , x1 , x2 , x3 , x4 , y, y1 , y2 , y3 , y4 , m1 , m2 , m3 , m4 ] :
  Points := [ [     -4.879251688377796 , -7.477990553920259 ,
      -3.451319345804918 , -4.952805307410782 , -3.634891546375222 ,
       3.625252311603879 , 3.744071784969180 , 5.298920296843097 ,
       1.161573235271139 , 4.296443929332098 , -0.116067188362753 ,
       0.983885694558098 , 0.779257316847020 , 0.268476647091288 ]] :
  isReal := true ;
  numVars := 14 ;
  Settings := defaultSettings( );
  PointsData := alphaCertified(alphaPath, Polys, Vars, Points) ;
[alphaCertified, alphaCertifiedExp, defaultExpSettings, defaultSettings, loadOutput,
  printPoints, printPolyExpSystem, printPolynomialSystem, printSettings]
                              isReal := true
                              numVars := 14
Settings := Record(algorithm = 2, arithmeticType = 0, precision = 96, refineDigits
   = 0, numRandomSystems = 2, randomDigits = 10, randomSeed = 395718860534,
   newtonOnly = 0, numIterations = 2, realityCheck = 1, realityTest = 0, deleteFiles
   = true)
```

$$PointsData := Record\Big(\alpha = [9.862097245\ 10^{-13}],\ \beta = [3.807043996\ 10^{-15}],\ \gamma = [259.0486807],\ refinedPts = \Big[\Big[-\frac{1219812922094449}{250000000000000}, -\frac{7477990553920259}{1000000000000000}, -\frac{1725659672902459}{500000000000000}, -\frac{2476402653705391}{500000000000000}, -\frac{1817445773187611}{500000000000000}, \frac{3625252311603879}{1000000000000000}, \frac{187203589248459}{50000000000000}, \frac{5298920296843097}{1000000000000000}, \frac{1161573235271139}{1000000000000000}, \frac{2148221964666049}{500000000000000}, -\frac{116067188362753}{1000000000000000}, \frac{491942847279049}{500000000000000}, \frac{38962865842351}{50000000000000}, \frac{33559580886411}{125000000000000}\Big]\Big],\ isApproxSoln = [\text{"Yes"}],\ isDistinctSoln = [\text{"Yes"}],\ isRealSoln = [\text{"Yes"}]\Big) \quad \textbf{(1)}$$

FIGURE 1. Running alphaCertified for Example 4.1

```
> with(alphaCertifiedMaple) ;
  alphaPath := "D:/alphaCertified" :
  Polys := [6·x − 2·x1 − 2·x2 − 2·x3 , 6·y − 2·y1 − 2·y2 − 2·y3 ,−2·x + 2·x1
       + 2·mu1·x1 , −2·x + 2·x2 + 2·mu2·x2 − 4·mu2 , −2·x + 2·x3 + 2·mu3·x3
       + 4·mu3 , −2·y + 2·y1 + 2·mu1·y1 , −2·y + 2·y2 + 2·mu2·y2 − 4·mu2 ,
      −2·y + 2·y3 + 2·mu3·y3 − 4·mu3 , x1^2 + y1^2 − 1 , x2^2 − 4·x2 + 7 + y2
      ^2 − 4·y2 , x3^2 + 4·x3 + 7 + y3^2 − 4·y3 ] :
  Vars := [ x, x1, x2, x3, y, y1, y2, y3, mu1, mu2, mu3 ] :
  Points := [ [      0.000000000000000 , 0.000000000000000 ,
        1.029045421455892 , −1.029045421455892 , 1.507157410363700 ,
        1.000000000000000 , 1.760736115545550 , 1.760736115545550 ,
        0.507157410363699 , 1.059828589509191 , 1.059828589509190 ]] :
  isReal := true ;
  numVars := 11 ;
  Settings := defaultSettings( );
  PointsData := alphaCertified(alphaPath, Polys, Vars, Points) ;
[alphaCertified, alphaCertifiedExp, defaultExpSettings, defaultSettings, loadOutput,
   printPoints, printPolyExpSystem, printPolynomialSystem, printSettings]
                              isReal := true
                              numVars := 11
Settings := Record(algorithm = 2, arithmeticType = 0, precision = 96, refineDigits
   = 0, numRandomSystems = 2, randomDigits = 10, randomSeed = 395718860534,
   newtonOnly = 0, numIterations = 2, realityCheck = 1, realityTest = 0, deleteFiles
   = true)
PointsData := Record(α = [1.811952386 10^-13], β = [2.649278615 10^-15], γ          (1)
   = [68.39418005], refinedPts = [[0, 0, 257261355363973/250000000000000,
   −257261355363973/250000000000000, 15071574103637/10000000000000, 1, 35214722310911/20000000000000,
   35214722310911/20000000000000, 507157410363699/1000000000000000, 1059828589509191/1000000000000000,
   105982858950919/100000000000000]], isApproxSoln = ["Yes"], isDistinctSoln = ["Yes"],
   isRealSoln = ["Yes"])
```

FIGURE 2. Running alphaCertified for Example 4.2

```
> with(alphaCertifiedMaple) ;
  alphaPath := "D:/alphaCertified" :
  Polys := [  4·x − 2·x1 − 2·x2 , -2·x + 2·x1 + 2·mu1·x1 , -2·x + 2·x2 + 2
      ·mu2·x2 , 4·y − 2·y1 − 2·y2 , -2·y + 2·y1 − 18·mu1·y1 , -2·y + 2·y2
      − mu2 , x1^2 − 9·y1^2 − 49 , x2^2 − y2 − 2 ] :
  Vars := [x, x1, x2, mu1, mu2, y, y1, y2] :
  Points := [[4.4580480234334745 , 7.0274153637676040 , 1.8886806830993446 ,
       −0.36562053149461421 , 1.3604032504413466 , 0.88691309749193414 ,
       0.20671147227126088 , 1.5671147227126077 ], [ −4.4580480234334745 ,
      −7.0274153637676040 , −1.8886806830993446 , −0.36562053149461421 ,
       1.3604032504413466 , 0.88691309749193414 , 0.20671147227126088 ,
       1.5671147227126077 ]] :
  isReal := true ;
  numVars := 8 ;
  Settings := defaultSettings( );
  PointsData := alphaCertified(alphaPath, Polys, Vars, Points) ;
[alphaCertified, alphaCertifiedExp, defaultExpSettings, defaultSettings, loadOutput,
  printPoints, printPolyExpSystem, printPolynomialSystem, printSettings]
                              isReal := true
                              numVars := 8
Settings := Record(algorithm = 2, arithmeticType = 0, precision = 96, refineDigits
   = 0, numRandomSystems = 2, randomDigits = 10, randomSeed = 224085044619,
  newtonOnly = 0, numIterations = 2, realityCheck = 1, realityTest = 0, deleteFiles
   = true)
```

$$PointsData := Record\Big(\alpha = [9.269698007\ 10^{-14}, 9.269698007\ 10^{-14}], \beta = [1.448988046\ 10^{-15}, 1.448988046\ 10^{-15}], \gamma = [63.97359890, 63.97359890],$$

$$refinedPts = \Big[\Big[\frac{8916096046866949}{2000000000000000}, \frac{1756853840941901}{250000000000000}, \frac{9443403415496723}{5000000000000000}, -\frac{36562053149461421}{100000000000000000}, \frac{6802016252206733}{5000000000000000}, \frac{44345654874596707}{50000000000000000}, \frac{2583893403390761}{12500000000000000}, \frac{15671147227126077}{10000000000000000}\Big], \Big[-\frac{8916096046866949}{2000000000000000}, -\frac{1756853840941901}{250000000000000}, -\frac{9443403415496723}{5000000000000000}, -\frac{36562053149461421}{100000000000000000}, \frac{6802016252206733}{5000000000000000}, \frac{44345654874596707}{50000000000000000}, \frac{2583893403390761}{12500000000000000}, \frac{15671147227126077}{10000000000000000}\Big]\Big], isApproxSoln = ["Yes", "Yes"], isDistinctSoln = ["Yes", "Yes"], isRealSoln = ["Yes", "Yes"]\Big)$$

FIGURE 3. Running alphaCertified for Example 4.3

Bibliography

[1] Jonathan Hauenstein Andrew Sommese, Daniel Bates and Charles Wampler. Bertini: Software for numerical algebraic geometry. `www.nd.edu/~sommese/bertini`, 2012.

[2] D. N. Bernshtein. The number of roots of a system of equations. *Funct. Anal. Appl.*, 9(3):183–185, 1975.

[3] Ivo Bleylevens. *ALGEBRAIC POLYNOMIAL SYSTEM SOLVING AND APPLICATIONS*. PhD thesis, Universiteit Maastricht, 2010.

[4] Cucker F. Shub M. Blum, L. and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

[5] Max Buot and Donald Richards. Homotopy continuation. 2005. Computational Algorithms for Astrostatistics Summer School in Statistics for Astronomers and Physicists, lecture slides.

[6] J. D. Hauenstein and F. Sottile. Algorithm 921: alphacertified: Certifying solutions to polynomial systems. *ACM Transactions on Mathematical Software (TOMS)*, 38(4), 2012.

[7] P.S. Wang I.A. Ajwa, Z. Liu. Grobner bases algorithm. Technical report, ICM Technical Reports, Kent State University, 2003.

[8] Moran K. Grobner bases and their applications. `http://www.math.utah.edu/~boocher/writings/kaitlyn.pdf`, University of Utah, 2008.

[9] Lajos Szilassi. The right for doubting - and the necessity of doubt thoughts concerning the analysis of erdlys spidron system. `http://www.model.u-szeged.hu/cd/content/szilassi/Euler3d-kurzus/04%20%20%20A%20kocka%20feldarabol%25a0sa/The%20right%20for%20doubting.pdf`.

[10] T.Y. Li T. Chen, T.L. Lee. Hom4ps-3: A parallel numerical solver for systems of polynomial equations based on polyhedral homotopy continuation methods. In *Mathematical Software ICMS 2014*, pages 183–190, 2014.

[11] J. A. Yorke T. Y. Li, T. Sauer. The cheaters homotopy: An efficient procedure for solving systems of polynomial equations. *SIAM J. NUMER. ANAL.*, 26(5):1241–1251, 1989.

[12] C.H. Tsai T.L. Lee, T.Y. Li. Hom4ps-2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83:109–133, 2008.

[13] Tamas Rapcsak. Translated and presented by Sandor Bozoki at CEU. Nonlinear optimization lectures. `http://www.oplab.sztaki.hu/tanszek/download/nemlinopt.pdf`, 2007.

[14] Jan Verschelde. Phcpack: a general-purpose solver for polynomial systems by homotopy continuation. 2017. `http://homepages.math.uic.edu/~jan/PHCpack/phcpack.html`.