A Stabilization of Algebraic Phase Unwrapping by Subresultant

Daichi KITAHARA† Isao YAMADA†
†Department of Communications and Integrated Systems, Tokyo Institute of Technology, Japan
E-mail: {kitahara, isao}@sp.ss.titech.ac.jp

Abstract The phase unwrapping problem for complex polynomial, i.e., the problem of computing the continuous phase function of a given complex polynomial was solved rigorously with the algebraic phase unwrapping. However, with increasing degree of polynomial, we sometimes fail in exact phase unwrapping mainly because the algorithm encounters certain numerical instabilities due to the coefficient growth and the truncation error. In this paper, with use of the subresultant, we present an effective way to stabilize the algebraic phase unwrapping along the real axis.

1 Introduction

The phase unwrapping problem for a univariate complex polynomial is formulated as follows.

Problem 1 (Phase unwrapping for a univariate complex polynomial along the real axis) *For a given univariate complex polynomial* $A(t) \in \mathbb{C}[t]$ *such that* $A(t) \neq 0$ *for all* $t \in [0,1] \subset \mathbb{R}$*, compute*

$$\theta_A(t^*) := \theta_A(0) + \int_0^{t^*} \left(\arctan\left\{ \frac{\Im\{A(t)\}}{\Re\{A(t)\}} \right\} \right)' dt, \qquad (1)$$

which is called the unwrapped phase of $A(t)$ *at* $t^* \in (0,1]$*, where* $\theta_A(0) \in (-\pi, \pi]$ *satisfies* $A(0) = |A(0)|e^{j\theta_A(0)}$.

A rigorous symbolic algebraic solution to Problem 1 was established [1] (See Proposition 1 in Sec. 2.2). This method does not require any numerical root finding or numerical integration technique.

However, in a direct computer implementation of the algebraic phase unwrapping algorithm (SGA-RA) (See Algorithm 1 in Sec. 2.2), for polynomials of large degree, we encounter certain numerical instabilities due to the unavoidable gap between numerical value computed by digital computer and theoretical value.[1] Therefore, thoughtless direct computation of SGA-RA for polynomials of large degree, sometimes results in the failure of a key property of the desired *general Sturm sequence*, which is generated by applying SGA-RA, leading thus the failure of the exact phase unwrapping in the end.

In this paper, based on the similarity between SGA-RA and Euclidean algorithm for finding the *greatest common divisor (GCD)* of a pair of real polynomials, we propose a replacement of a certain inductive step in SGA-RA by the subresultant [5], [6]. By this replacement, numerical values of the ideal *general Sturm sequence* can be computed without suffering propagation of errors in the process of SGA-RA, and then the algebraic phase unwrapping is stabilized greatly even for polynomials of relatively large degree. This result is useful for wider application of the algebraic phase unwrapping, e.g., in a combination with the spline smoothing [2], to practical signal and image processing problems.

[1] Recently, we presented in [2] an idea based on a combination of the algebraic phase unwrapping and the spline smoothing of small order. This phase unwrapping technique is numerically robust and expected to be applicable to many signal and image processing problems, for example, estimations of surface topography in synthetic aperture radar (SAR) interferometry [3] and the degree of magnetic field inhomogeneity in the water/fat separation problem of magnetic resonance imaging (MRI) [4].

2 Preliminaries

2.1 Notation

Let $\mathbb{N}^*$, $\mathbb{R}$ and $\mathbb{C}$ denote respectively the set of all positive integers, real numbers and complex numbers. We use $j \in \mathbb{C}$ to denote the imaginary unit satisfying $j^2 = -1$. For any $c \in \mathbb{C}$, $\Re(c)$ and $\Im(c)$ stand respectively for the real and imaginary parts of c. For any $C(t) = \sum_{k=0}^{m} c_k t^k \in \mathbb{C}[t]$ (s.t. $c_m \neq 0$ and $m \geq 0$), we define $\deg(C) := m$, $\mathrm{lc}(C) := c_m$ and $\mathrm{mmc}(C) := \max\{|c_0|, |c_1|, \ldots, |c_m|\}$.

For any $C(t) = \sum_{k=0}^{m} c_k t^k \in \mathbb{C}[t]$ ($t \in [0,1] \subset \mathbb{R}$), we have $C(t) = C_{(0)}(t) + jC_{(1)}(t)$, where $C_{(0)}(t) := \sum_{k=0}^{m} \Re\{c_k\} t^k, C_{(1)}(t) := \sum_{k=0}^{m} \Im\{c_k\} t^k \in \mathbb{R}[t]$ and

$$\left.\begin{aligned} \Re\{C(t)\} &= \Re\{C_{(0)}(t) + jC_{(1)}(t)\} = C_{(0)}(t) \\ \Im\{C(t)\} &= \Im\{C_{(0)}(t) + jC_{(1)}(t)\} = C_{(1)}(t) \end{aligned}\right\}. \qquad (2)$$

2.2 Algebraic phase unwrapping

The next proposition is a solution to Problem 1. This proposition is a relaxation of Theorem 1 in [1]. Indeed, the condition $\Re\{A(0)\} = A_{(0)}(0) \neq 0$ assumed in [1] is removed, and the definition of $\Psi_k(t) \in \mathbb{R}[t]$ is modified in Proposition 1.

Proposition 1 (Algebraic phase unwrapping along the real axis) *Let* $A(t) := A_{(0)}(t) + jA_{(1)}(t) \in \mathbb{C}[t]$ *satisfy* $A(t) \neq 0$ $(t \in [0,1])$, *where* $A_{(0)}(t), A_{(1)}(t) \in \mathbb{R}[t]$. *Define*

$$\mathcal{Z}_{A_{(0)}} := \{t \in (0,1) \mid A_{(0)}(t) = 0\} = \begin{cases} \varnothing & \text{if } A_{(0)}(t) \neq 0 \text{ for all } t \in (0,1), \\ \{\mu_1, \mu_2, \ldots, \mu_\tau\} & \text{otherwise,} \end{cases}$$

where $0 < \mu_1 < \cdots < \mu_\tau < 1$, *and*

$$\mathcal{X}(\mu_i) := \begin{cases} +1 & \text{if } \begin{cases} A_{(0)}(t)A_{(1)}(t) > 0 & \text{for } t \in (\mu_i - \varepsilon, \mu_i) \text{ and} \\ A_{(0)}(t)A_{(1)}(t) < 0 & \text{for } t \in (\mu_i, \mu_i + \varepsilon), \end{cases} \\ -1 & \text{if } \begin{cases} A_{(0)}(t)A_{(1)}(t) < 0 & \text{for } t \in (\mu_i - \varepsilon, \mu_i) \text{ and} \\ A_{(0)}(t)A_{(1)}(t) > 0 & \text{for } t \in (\mu_i, \mu_i + \varepsilon), \end{cases} \\ 0 & \text{otherwise,} \end{cases}$$

for μ_i $(i = 1, 2, \ldots, \tau)$ *and for sufficiently small* $\varepsilon > 0$. *Then we have the following relations.*

(a) *For any* $t^* \in (0,1]$,

$$\theta_A(t^*) = \theta_A(0) - \lim_{t \to 0+0} \arctan\{Q_A(t)\} + \lim_{t \to t^*-0} \arctan\{Q_A(t)\} + \Lambda(t^*)\pi,$$

where $Q_A(t) := \frac{\Im\{A(t)\}}{\Re\{A(t)\}} = \frac{A_{(1)}(t)}{A_{(0)}(t)}$ *and* $\Lambda(t^*) := \sum_{\mu_i \in (0,t^*)} \mathcal{X}(\mu_i)$.

(b) *Let* $\{\Psi_k(t)\}_{k=0}^{q}$ *be a sequence of real polynomials obtained by applying Algorithm 1 (SGA-RA) to* $A_{(0)}(t)$ *and* $A_{(1)}(t)$. *Define for each* $t \in [0,1]$ *the number of variations in the sign of* $\{\Psi_k(t)\}_{k=0}^{q}$ *by*

$$V\{\Psi(t)\} := V\{\Psi_0(t), \Psi_1(t), \ldots, \Psi_q(t)\} := \left|\{i \mid 0 \leq i < q \text{ and } \Psi_i(t)\Psi_{i+\varrho(i)}(t) < 0\}\right|,$$

where $\varrho(i) := \min\{k \in \mathbb{N}^* \mid \Psi_{i+k}(t) \neq 0\}$. *Then, for every* $t^* \in (0,1]$, *we have*

$$\theta_A(t^*) = \theta_A(0) - \begin{Bmatrix} \arctan\{Q_A(0)\} & \text{if } A_{(0)}(0) \neq 0, \\ \operatorname{sgn}(\Psi_0(0)\Psi_1(0))\pi/2 & \text{if } A_{(0)}(0) = 0, \end{Bmatrix} + \begin{cases} \arctan\{Q_A(t^*)\} + [V\{\Psi(t^*)\} - V\{\Psi(0)\}]\pi & \text{if } A_{(0)}(t^*) \neq 0, \\ \pi/2 + [V\{\Psi(t^*)\} - V\{\Psi(0)\}]\pi & \text{if } A_{(0)}(t^*) = 0, \end{cases} \quad (3)$$

where

$$\operatorname{sgn}(t) := \begin{cases} +1 & \text{if } t \geq 0, \\ -1 & \text{if } t < 0. \end{cases} \quad (4)$$

In this paper, we call the sequence of real polynomials $\{\Psi_k(t)\}_{k=0}^{q}$ *general Sturm sequence*. For the relation between Algorithm 1 and the Euclidean algorithm for $\mathrm{GCD}(\Psi_0, \Psi_1)$, see Sec. 3.1.

Example 1 (Algebraic phase unwrapping and coefficient growth) *Let us construct the unwrapped phase of the univariate complex polynomial*

$$A(t) := t(t-0.1)(t-0.5)(t-0.51) + j(t-0.49)(t-0.515)(t-0.52)(t-1).$$

Algorithm 1 Sturm generating algorithm (SGA-RA)

Input: $A_{(0)}(t), A_{(1)}(t)$

$\Psi_0(t) \leftarrow \dfrac{A_{(0)}(t)}{t^{e_0}}$, $\Psi_1(t) \leftarrow \dfrac{A_{(1)}(t)}{t^{e_1}}$

(where e_k denotes the order of $t = 0$ as a zero of polynomial $A_{(k)}(t)$)

$k \leftarrow 1$

while $\deg(\Psi_k) \neq 0$ **do**

  $\Psi_{k+1}(t) \leftarrow -\Psi_{k-1}(t) + H_k(t)\Psi_k(t)$

  (where $H_k(t) \in \mathbb{R}[t]$ and $\deg(\Psi_{k+1}) < \deg(\Psi_k)$)

  $k \leftarrow k + 1$

end while

$q \leftarrow \begin{cases} k & \text{if } \Psi_k(t) \not\equiv 0 \\ k-1 & \text{if } \Psi_k(t) \equiv 0 \end{cases}$

Output: $\{\Psi_k(t)\}_{k=0}^{q}$

$A_{(0)}(t)$ *and* $A_{(1)}(t)$ *are given respectively as*

$$A_{(0)}(t) = t^4 - 1.11t^3 + 0.356t^2 - 0.0255t,$$
$$A_{(1)}(t) = t^4 - 2.525t^3 + 2.29995t^2 - 0.906172t + 0.131222.$$

Applying Algorithm 1 to $A_{(0)}(t)$ *and* $A_{(1)}(t)$, *we obtain general Sturm sequence* $\{\Psi_k(t)\}_{k=0}^{5}$ *as*

$$\Psi_0(t) = t^3 - \frac{111}{100}t^2 + \frac{89}{250}t - \frac{51}{2000},$$
$$\Psi_1(t) = t^4 - \frac{101}{40}t^3 + \frac{45999}{20000}t^2 - \frac{226543}{250000}t + \frac{65611}{500000},$$
$$\Psi_2(t) = -t^3 + \frac{111}{100}t^2 - \frac{89}{250}t + \frac{51}{2000},$$
$$\Psi_3(t) = -\frac{3733}{10000}t^2 + \frac{94233}{250000}t - \frac{190279}{2000000},$$
$$\Psi_4(t) = -\frac{27788829033}{260102169185000}t + \frac{15335859}{278705780000},$$
$$\Psi_5(t) = \frac{3391452647840106395584666460779211811}{11996717727057501597535406952577469520000 0}.$$

From $A_{(0)}(0) = 0$ *and* $A_{(1)}(0) = \frac{65611}{500000}$, $\theta_A(0) = \pi/2$. *Moreover from* $\operatorname{sgn}(\Psi_0(0)\Psi_1(0)) = \operatorname{sgn}\left(-\frac{3346161}{1000000000}\right) = -1$ *and* $V\{\Psi(0)\} = V\left\{-\frac{51}{2000}, \frac{65611}{500000}, \frac{51}{2000}, -\frac{190279}{2000000}, \frac{15335859}{278705780000}, \frac{3391452647840106395584666460779211811}{119967177270575015975354069525774695200000}\right\} = 3$, *the unwrapped phase* $\theta_A(t)$ *in* (3) *is expressed as*

$$\theta_A(t) = \pi + \begin{cases} \arctan\{Q_A(t)\} + [V\{\Psi(t)\} - 3]\pi & \text{if } A_{(0)}(t) \neq 0, \\ \pi/2 + [V\{\Psi(t)\} - 3]\pi & \text{if } A_{(0)}(t) = 0, \end{cases}$$

which is depicted in Fig. 1.

2.3 Numerical instabilities of SGA-RA

To implement Algorithm 1 (SGA-RA) precisely, we need large number of digits to express the rational coefficients of the polynomials $\Psi_k(t)$ (e.g., See Example

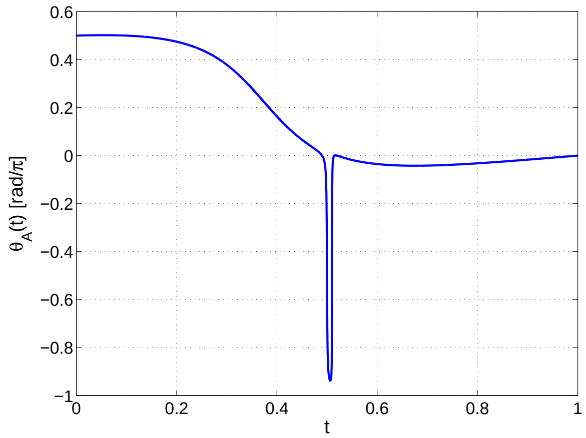

Figure 1: Exact unwrapped phase by Proposition 1

1). We call this phenomenon the *coefficient growth* in analogy with the typical cases in the computation of the *polynomial reminder sequence* through the Euclidian algorithm [6]. In computer implementation of $\theta_A(t)$ in (3) through SGA-RA, the coefficient growth causes the truncation error in the floating-point expression of the rational coefficients (or memory shortages by increasing number of digits for exact expression of the rational coefficients). In particular, once a serious *information loss* (by the addition or subtraction for a pair of numbers of ill-balanced absolute values) or *catastrophic cancellation* (by the subtraction for a pair of very close numbers) occurs, the gap between numerical value of $\{\Psi_k(t)\}_{k=0}^{q}$ by digital computer and theoretical value becomes unacceptably large (See Example 2).

Example 2 (Catastrophic cancellation) *The general Sturm sequence* $\{\Psi_k(t)\}_{k=0}^{5}$ *obtained in Example 1 is expressed in decimal number expression as*

$$\Psi_0(t) = t^3 - 1.11t^2 + 0.356t - 0.0255,$$
$$\Psi_1(t) = t^4 - 2.525t^3 + 2.29995t^2 - 0.906172t + 0.131222,$$
$$\Psi_2(t) = -t^3 + 1.11t^2 - 0.356t + 0.0255,$$
$$\Psi_3(t) = -0.3733t^2 + 0.376932t - 0.0951395,$$
$$\Psi_4(t) = -1.0683812872 \times 10^{-4}t + 5.5025263559 \times 10^{-5},$$
$$\Psi_5(t) = 2.8269837842 \times 10^{-5}.$$

From the above values, we verify that the absolute value of coefficients of $\Psi_k(t)$ *decreases drastically from* $k = 3$ *to* $k = 4$*. This phenomenon has been studied in terms of approximate GCD of degree* 3 [7], [8] *(See Appendix).*

In SGA-RA with 64-bit floating point expression, the absolute value of the leading coefficient of $\Psi_4(t)$*, i.e.,* $|\mathrm{lc}(\Psi_4)| = \dfrac{27788829033}{260102169185000}$*, is derived and truncated as*

$$-\frac{89}{250} - \frac{(-1)\times(-\frac{190279}{2000000})}{-\frac{3733}{10000}} - \frac{\left(\frac{111}{100} - \frac{(-1)\times\frac{94233}{250000}}{-\frac{3733}{10000}}\right)\times\frac{94233}{250000}}{\frac{3733}{10000}}$$
$$\approx 1.068381287250980 \times 10^{-4}.$$

However, direct computation of $\dfrac{27788829033}{260102169185000}$ *with 64-bit floating point expression is*

$$1.068381287248510 \times 10^{-4},$$

which is more precise. From this fact, we verify that the significant figures of coefficients of $\Psi_k(t)$ *are lost drastically from* $k = 3$ *to* $k = 4$*. This phenomenon is so-called the catastrophic cancellation in the floating-point expression.*

Once the *information loss* or the *catastrophic cancellation* occurs, this influences inductively in the process of SGA-RA, which results in the failure of *general Sturm sequence's* key property:

$$\Psi_k(t)=0 \text{ at } t\in[0,1] \Rightarrow \Psi_{k-1}(t)\Psi_{k+1}(t)<0 \quad \text{for } k = 1, 2, \ldots, q-1, \tag{5}$$

leading thus the failure of (3). This situation restricts the practical applicability of Proposition 1 especially for polynomials $A(t) \in \mathbb{C}[t]$ of large degree.

3 Stabilization of algebraic phase unwrapping

3.1 Relationship between Algorithm 1 and Euclidean algorithm

The Euclidian algorithm for computing $\mathrm{GCD}(\Psi_0, \Psi_1)$ generates the *polynomial remainder sequence* $\{P_k(t)\}_{k=0}^{q}$, where $P_0(t) := \Psi_0(t) := \dfrac{A_{(0)}(t)}{t^{e_0}}$, $P_1(t) := \Psi_1(t) := \dfrac{A_{(1)}(t)}{t^{e_1}}$ and

$$P_{k+1}(t) := P_{k-1}(t) + Q_k(t)P_k(t) \quad (k=1,2,\ldots,q-1)$$
$$(\text{s.t. } Q_k(t) \in \mathbb{R}[t] \text{ and } \deg(P_{k+1}) < \deg(P_k)).$$

On the other hand, in SGA-RA $\{\Psi_k(t)\}_{k=0}^{q}$ is defined as

$$\Psi_{k+1}(t) := -\Psi_{k-1}(t) + H_k(t)\Psi_k(t) \quad (k=1,2,\ldots,q-1)$$
$$(\text{s.t. } H_k(t) \in \mathbb{R}[t] \text{ and } \deg(\Psi_{k+1}) < \deg(\Psi_k)).$$

As a result, we have

$$\Psi_k(t) = (-1)^{\frac{(k-1)k}{2}} P_k(t) \qquad (k = 0, 1, \ldots, q). \tag{6}$$

3.2 Subresultant

In what follows, we assume

$$\begin{aligned}&\deg(P_{k-1}) \geq \deg(P_k)\\ &\quad \Rightarrow \deg(P_{k+1}) = \deg(P_k) - 1 \quad (k=1,2,\ldots,q-1).\end{aligned} \tag{7}$$

For the polynomials $P_0(t)$ and $P_1(t)$ which cause the coefficient growth, the assumption (7) holds almost always.

For a pair of real polynomials

$$\begin{aligned} P_0(t) &:= a_m t^m + a_{m-1} t^{m-1} + \cdots + a_1 t + a_0,\\ P_1(t) &:= b_n t^n + b_{n-1} t^{n-1} + \cdots + b_1 t + b_0,\end{aligned}$$

s.t. $a_m \neq 0$ and $b_n \neq 0$, we define $R_i(P_0, P_1, t) \in \mathbb{R}[t]^{(m+n-2i)\times(m+n+2i)}$ $(i = 0, 1, \ldots, \min\{m-1, n-1\})$ by

$$R_i(P_0,P_1,t) := \begin{pmatrix} a_m & a_{m-1} & \cdots & a_i & a_{i-1} & \cdots & a_0 & & & P_0(t)t^{n-i-1} \\ & a_m & a_{m-1} & \cdots & a_i & a_{i-1} & \cdots & a_0 & & P_0(t)t^{n-i-2} \\ & & \ddots & \ddots & & \ddots & \ddots & & \ddots & \vdots \\ & & & a_m & a_{m-1} & \cdots & a_i & a_{i-1} & \cdots & P_0(t)t^{i} \\ & & & & \ddots & \ddots & & \ddots & \ddots & \vdots \\ & & & & & a_m & a_{m-1} & \cdots & a_i & P_0(t)t \\ & & & & & & a_m & \cdots & a_{i+1} & P_0(t) \\ b_n & b_{n-1} & \cdots & b_i & b_{i-1} & \cdots & b_0 & & & P_1(t)t^{m-i-1} \\ & b_n & b_{n-1} & \cdots & b_i & b_{i-1} & \cdots & b_0 & & P_1(t)t^{m-i-2} \\ & & \ddots & \ddots & & \ddots & \ddots & & \ddots & \vdots \\ & & & b_n & b_{n-1} & \cdots & b_i & b_{i-1} & \cdots & P_1(t)t^{i} \\ & & & & \ddots & \ddots & & \ddots & \ddots & \vdots \\ & & & & & b_n & b_{n-1} & \cdots & b_i & P_1(t)t \\ & & & & & & b_n & \cdots & b_{i+1} & P_1(t) \end{pmatrix}. \tag{8}$$

Then the i-th *subresultant* of P_0 and P_1 is defined as

$$\mathrm{Sres}_i(P_0, P_1, t) := \det(R_i(P_0, P_1, t)). \tag{9}$$

Under the assumption (7), it is well-known that for $\deg(P_0) = m \geq n = \deg(P_1)$ the subresultsnt sequence $\{\mathrm{Sres}_i(P_0, P_1, t)\}$ satisfies

$$\mathrm{Sres}_{n-k+1}(P_0,P_1,t) = \lambda_k P_k(t) \quad (k=2,3,\ldots,n+1), \tag{10}$$

where $\lambda_k := \left((-1)^{k-1}\mathrm{lc}(P_1)\right)^{m-n+1} \prod_{i=2}^{k-1}(\mathrm{lc}(P_i))^2$, and for $\deg(P_0) = m < n = \deg(P_1)$ (i.e., $P_2(t) = P_0(t)$)

$$\mathrm{Sres}_{m-k+2}(P_1,P_0,t) = \lambda'_k P_k(t) \quad (k=3,4,\ldots,m+2), \tag{11}$$

where $\lambda'_k := \left((-1)^{k-2}\mathrm{lc}(P_0)\right)^{n-m+1} \prod_{i=3}^{k-1}(\mathrm{lc}(P_i))^2$ [5], [6].

For $\deg(P_0) \geq \deg(P_1)$ from (6) and (10), we can express

$$\begin{aligned}\mathrm{sgn}(\Psi_k(t^*)) &= (-1)^{(k-1)\left(m-n+1+\frac{k}{2}\right)} (\mathrm{sgn}(\mathrm{lc}(\Psi_1)))^{m-n+1}\\ &\quad \times \mathrm{sgn}(\mathrm{Sres}_{n-k+1}(\Psi_0,\Psi_1,t^*)) \quad (k=2,3,\ldots,n+1).\end{aligned} \tag{12}$$

Similarly for $\deg(P_0) < \deg(P_1)$ from (6) and (11), we can express $\mathrm{sgn}(\Psi_2(t^*)) = -\mathrm{sgn}(\Psi_0(t^*))$ and

$$\begin{aligned}\mathrm{sgn}(\Psi_k(t^*)) &= (-1)^{(k-2)(n-m+1)+\frac{(k-1)k}{2}} (\mathrm{sgn}(\mathrm{lc}(\Psi_0)))^{n-m+1}\\ &\quad \times \mathrm{sgn}(\mathrm{Sres}_{m-k+2}(\Psi_1,\Psi_0,t^*)) \quad (k=3,4,\ldots,m+1).\end{aligned} \tag{13}$$

The relations (12) and (13) imply that we can compute each $\Psi_k(t^*)$ by $\{\mathrm{Sres}_i(P_0, P_1, t^*)\}$ without passing through the inductive process in Algorithm 1.

The following algorithm uses (12) or (13) in place of the inductive step in Algorithm 1. Note that real polynomials $\Psi_k(t) \in \mathbb{R}[t]$ $(k = 2, 3, \ldots, q)$ are not necessary for evaluating the values of $\{\Psi_k(t^*)\}_{k=0}^{q}$. The computational complexity for each determinant $\mathrm{Sres}_{i-1}(P_0, P_1, t^*)$ is at most $O(n^{\log 7}) \approx O(n^{2.81})$ [9].

Algorithm 2 Proposed algorithm for computing (3)

Input: $A_{(0)}(t), A_{(1)}(t), t^*$

$\Psi_0(t) \leftarrow \dfrac{A_{(0)}(t)}{t^{e_0}}$, $\Psi_1(t) \leftarrow \dfrac{A_{(1)}(t)}{t^{e_1}}$

(where e_k denotes the order of $t = 0$ as a zero of polynomial $A_{(k)}(t)$)

$\Psi_0(t) \leftarrow |\mathrm{Const.}\,1|\,\Psi_0(t)$, $\Psi_1(t) \leftarrow |\mathrm{Const.}\,2|\,\Psi_1(t)$

$m \leftarrow \deg(\Psi_0)$, $n \leftarrow \deg(\Psi_1)$

$a \leftarrow \mathrm{lc}(\Psi_0)$, $b \leftarrow \mathrm{lc}(\Psi_1)$

if $m \geq n$ **then**

  for $k = 2$ **to** $(n+1)$ **do**

    $\Psi_k(t^*) \leftarrow (-1)^{(k-1)\left(m-n+1+\frac{k}{2}\right)}(\mathrm{sgn}(b))^{m-n+1} \times \mathrm{Sres}_{n-k+1}(\Psi_0,\Psi_1,t^*)$

  end for

else

  $\Psi_2(t^*) \leftarrow -\Psi_0(t^*)$

  for $k = 3$ **to** $(m+1)$ **do**

    $\Psi_k(t^*) \leftarrow (-1)^{(k-2)(n-m+1)+\frac{(k-1)k}{2}}(\mathrm{sgn}(a))^{n-m+1} \times \mathrm{Sres}_{m-k+2}(\Psi_1,\Psi_0,t^*)$

  end for

end if

Output: $\{\Psi_k(t^*)\}_{k=0}^{\min\{m+2,n+1\}}$

Remark: e.g., $\mathrm{Const.}\,1 = \frac{1}{\mathrm{mmc}(\Psi_0)}$, $\mathrm{Const.}\,2 = \frac{1}{\mathrm{mmc}(\Psi_1)}$

---

4 Numerical example

In this section, we examine the numerical performance of Algorithm 2 in the algebraic phase unwrapping of some univariate complex polynomials.

Define two real polynomials $A_{(0)}(t)$ and $A_{(1)}(t)$ by

$$\begin{aligned} A_{(0)}(t) &:= (t-0.205)(t-0.5)(t-0.75)(t-0.805)(t-1.2)\overline{A}_{(0)}(t),\\ A_{(1)}(t) &:= (t-0.2)(t-0.34)(t-0.35)(t-0.81)(t-1.21)\overline{A}_{(1)}(t),\end{aligned}$$

where $\overline{A}_{(0)}(t)$ is a polynomial of degree 44 whose all coefficients are generated by the uniform distribution over $[-0.5, 0.5]$ and $\overline{A}_{(1)}(t)$ is a polynomial of degree 24 whose all coefficients are generated by the uniform distribution over $[-0.5, 0.5]$. Fig. 2 depicts one example of the estimations of the unwrapped phase $\theta_A(t)$ for $A(t) := A_{(0)}(t) + jA_{(1)}(t)$ with Algorithm 1 and Algorithm 2. From Fig. 2, Algorithm 1 fails in phase unwrapping at $t = 0.2$ and $t = 0.81$. On the other hand, Algorithm 2 succeeds in phase unwrapping at all $t \in [0, 1]$. Table 1 summarizes the result for 300 polynomials generated as above, where we can see (i) the total number of polynomials in failure by Algorithm 1 is reduced to less than 1/6 by Algorithm 2, and (ii) the total number of positions in failure by Algorithm 1 is reduced to less than 1/14 by Algorithm 2.

5 Conclusion

By replacing the inductive step in SGA-RA by numerical evaluations of a subresultant sequence, we succeeded in stabilizing the algebraic phase unwrapping along the real axis. Numerical examples demonstrate the remarkable improvement achieved by the proposed technique.

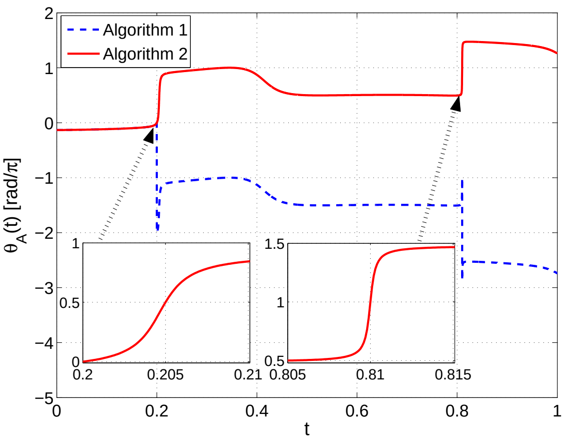

Figure 2: Estimations of the unwrapped phase

Table 1: Performance comparison for random polynomials

Algorithm	Total number of polynomials in failure	Total number of positions in failure
Algorithm 1	37 (among 300)	88
Algorithm 2	6 (among 300)	6

Appendix: Catastrophic cancellation in SGA-RA

Suppose that we have two real polynomials $P_0(t)$ and $P_1(t)$ which respectively have roots close to α_k ($k = 1, 2, \ldots, s$), i.e., $P_0(t)$ and $P_1(t)$ are expressed as

$$\left.\begin{aligned} P_0(t) &= (t-\alpha_1-\delta_{1_{(0)}})(t-\alpha_2-\delta_{2_{(0)}})\cdots(t-\alpha_s-\delta_{s_{(0)}})\overline{P}_0(t) \\ P_1(t) &= (t-\alpha_1-\delta_{1_{(1)}})(t-\alpha_2-\delta_{2_{(1)}})\cdots(t-\alpha_s-\delta_{s_{(1)}})\overline{P}_1(t) \end{aligned}\right\},$$

where $|\delta_{k_{(0)}}|, |\delta_{k_{(1)}}| \ll 1$ ($k = 1, 2, \ldots, s$) and $\overline{P}_0(t), \overline{P}_1(t) \in \mathbb{R}[t]$ do not have roots locating closely.

Define $\epsilon_0(t)$ and $\epsilon_1(t)$ as the reminders of the divisions of $P_0(t)$ and $P_1(t)$ by

$$D(t) := (t-\alpha_1)(t-\alpha_2)\cdots(t-\alpha_s), \tag{14}$$

i.e.,

$$\left.\begin{aligned} \epsilon_0(t) &:= P_0(t) + Q_0(t)D(t) \\ \epsilon_1(t) &:= P_1(t) + Q_1(t)D(t) \end{aligned}\right\},$$

where $Q_k(t) \in \mathbb{R}[t]$ and $\deg(\epsilon_k) < \deg(D)$ ($k = 0, 1$). According to [7], $\epsilon_k(t)$ satisfies $\mathrm{mmc}(\epsilon_k) \ll \mathrm{mmc}(P_k)$ ($k = 0, 1$) and $D(t)$ is called the *approximate greatest common divisor* with accuracy $\max\{\mathrm{mmc}(\epsilon_0), \mathrm{mmc}(\epsilon_1)\}$.

If the polynomial reminder sequence $\{P_k(t)\}_{k=0}^{q}$ satisfies $\deg(P_0) = m \geq n = \deg(P_1)$ and (7), we have $\deg(P_{n-s+1}) = s$ and $P_{n-s+1}(t) \approx \lambda D(t)$, where $\lambda \in \mathbb{R}$. Moreover if the absolute values of the all coefficients of $P_0(t)$ and $P_1(t)$ are not much different from 1, the values of $\mathrm{mmc}(P_k)$ ($2 \leq k \leq n-s+1$) tend to be not much different from 1 while the values of $\mathrm{mmc}(P_k)$ ($n-s+2 \leq k \leq n+1$) tend to be very small [8].

Acknowledgement

This work was supported in part by JSPS Grants-in-Aid (B-21300091).

References

[1] I. Yamada and K. Oguchi, “High-resolution estimation of the directions-of-arrival distribution by algebraic phase unwrapping algorithms,” *Multidimensional Systems and Signal Processing*, vol. 22, pp. 191–211, 2011.

[2] D. Kitahara and I. Yamada, “A robust algebraic phase unwrapping based on spline approximation,” *IEICE Technical Report*, 2012.

[3] R. M. Goldstein, H. A. Zebker, and C. L. Werner, “Satellite radar interferometry: Two-dimensional phase unwrapping,” *Radio Science*, vol. 23, no. 4, pp. 713–720, 1988.

[4] S. M. Song, S. Napel, N. J. Pelc, and G. H. Glover, “Phase unwrapping of MR phase images using Poisson equation,” *IEEE Trans. Image Process.*, vol. 4, no. 5, pp 667–676, 1995.

[5] G. E. Collins, “Subresultants and reduced polynomial reminder sequences,” *J. ACM*, vol. 14, pp.128–142, 1967.

[6] W. S. Brown and J. F. Traub, “On Euclid’s algorithm and the theory of subresultants,” *J. ACM*, vol. 18, pp. 505–514, 1971.

[7] T. Sasaki and M. Sasaki, “Polynomial remainder sequence and approximate GCD,” *ACM SIGSAM Bulletin*, vol.31, pp. 4–10, 1997.

[8] T. Sasaki and M. Sasaki, “Analysis of accuracy decreasing in polynomial remainder sequence with floating-point number coefficient,” *J. Inform. Process.*, vol. 12, pp. 394–403, 1989.

[9] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.