近似代数の算法と応用の研究

(課題番号　１２４８００６５)

平成１２～１４年度科学研究費補助金（基盤研究(B)(1)）

研究成果報告書

平成１５年５月

研究代表者 佐々木建昭

（筑波大学 数学系 教授）

本冊子は、我が日本が創始・発展させ、数式処理において世界的トピックスの一つとなった『近似代数』の算法と応用に関して、文部省科学研究費補助金の援助を得て、平成１２年度～１４年度の３年間に行なわれた研究の成果をまとめたものである。

1. 研究課題： 近似代数の算法と応用の研究

2. 課題番号： １２４８００６５

3. 研究組織

研究代表者：	佐々木建昭	（筑波大学・数学系・教授）
研究分担者：	坂井　公	（筑波大学・数学系・助教授）
研究分担者：	赤平 昌文	（筑波大学・数学系・教授）
研究分担者：	青嶋　誠	（筑波大学・数学系・助教授）
研究分担者：	小池 健一	（筑波大学・数学系・講師）
研究分担者：	田中 秀和	（筑波大学・数学系・助手）
研究分担者：	塚田 信高	（筑波大学・数学系・助手）
研究分担者：	野田松太郎	（愛媛大学・工学部・教授）
研究分担者：	甲斐　博	（愛媛大学・工学部・講師）
研究分担者：	加古富志雄	（奈良女子大・理学部・教授）
研究分担者：	福井 哲夫	（武庫川女子大・助教授）

4. 研究経費

平成１２年度	３，２００千円
平成１３年度	２，９００千円
平成１４年度	３，０００千円
計	９，１００千円

5. 報告書目次

なお、本研究グループは本研究に先駆けて、下記の研究を行ってきた。

1. 1991～1993年：試験研究B(1)「数値数式融合計算システムの開発」、6,500千円。
2. 1994～1996年：基盤研究A(1)「近似代数計算システムの開発」、10,500千円。
3. 1997～1999年：基盤研究A(1)「近似代数の算法と応用の研究」、16,900千円。

1 近似代数と世界の現況

近似代数とは、従来の代数演算に摂動項を加え、摂動項がある一定範囲内で変動することを許容しつつ、所与の演算を行うことである。1980年代の末に本研究代表者らが世界に先駆けて提案した概念で、数式処理の分野では世界的トピックスの一つとなっている。

近似代数とは、単に従来の計算機代数の算法を浮動小数で実行するだけで、メリットは計算の高速化だけだ、と思われるかもしれないが、それは全くの誤りである。たとえば、多項式の厳密な因数分解は最終的に有限体上の1変数多項式の因数分解に帰着されるが、この算法を浮動小数で実行することは絶対に不可能であり、全く新しい算法の開発が必要である。さらに、厳密な演算では得られない有用な情報が得られるのである。

近似代数の研究は、佐々木・野田による近似GCDと近似無平方分解の算法（J. Inf. Process., 1989）と佐々木等による近似因数分解算法（Jpn. J. Indus. Appl. Math., 1991）で始まったと言ってよいだろう。当初は世界に内密にすべく、国内雑誌に発表していたが、4～5年後に外国に知られることになった。途端に近似GCDに関して、摂動項の数値的最小化に基づく算法（Karmarker & Lakshman: Proc. ISSAC'96, ACM）、2乗ノルムでの摂動項の最小値と特異値分解との関係（Corless他2名: Proc. ISSAC'95, ACM）、最近多項式理論への拡張（Hitz & Kaltofen: Proc. ISSAC'98, ACM）、など、世界で20以上もの論文が発表される状況になった（注：所定の性質を持たせるよう、与えられた多項式の係数を最小に変化させて得られる多項式を最近多項式という）。

一方、近似因数分解の研究は異なる展開を辿った。佐々木らの論文以降、外国で最初に著わされた論文（Galligo & Watt: Proc. ISSAC'97, ACM）は佐々木らの算法を効率化するものであった。次に著されたのは2001年で、同じ国際会議で同時に三つの算法が発表された（Galligo & Rupprecht:, Corless他3名:, Sasaki: Proc. ISSAC2001, ACM）。これらはいずれも異なる算法であり、アイデアに富むものであった。この頃から近似因数分解の研究が世界的にブレイクした。近似代数では数値解析の手法を利用することが多いが、最近の近似因数分解算法では数値解析の手法を"自由奔放に"利用していると言ってよい。それでいて、浮動小数近似から代数的数を復元したり、絶対的因数分解(複素数体上での因数分解)を厳密に正しく復元するのである。筆者等もこのような研究をしたいと思っていたが、先を越された気がする。これらの研究は、近似代数が前途に大きく広がっていることを如実に感じさせる。

上記以外の研究にも触れておこう。近似代数をより多くの代数演算に広げたいとは誰も思うことであるが、基本的演算で近似化されたものは意外に少い。無平方分解の近似化は近似GCDと同時に佐々木・野田が行い、野田・甲斐は有理関数近似を研究したが、他にはグレブナー基底計算の近似化（Stetter & Thallinger: Proc. ISSAC'98, ACM）、多項式の合成（Corless他3名: Proc. ISSAC'99, ACM）がある程度である（多項式 $F(x)$ が与えられたとき、$F(x) = G(h(x))$ なる多項式 $G(x), h(x)$ を見出すことを合成という）。より広い演算への近似代数の拡張はもっと追求されてしかるべきである。

近似代数では、算法の誤差解析と安定化の研究が不可欠である。この方面では、世界の研究者の動きはどういう訳か鈍く、日本人、特に本研究グループとNTTの白柳氏が気を吐いている。本研究グループの成果については次章以降を参照されたい。

2　研究の目的と成果の概要

本研究前の９年間にわたる科学研究費補助金による研究で、近似代数用計算システムはほぼ完成しており、多項式の近似 GCD の計算や近似因数分解など、基本的な演算の近似算法も荒削りながら開発を終了した。そこで、本研究では次の３点を目的とした。

目的１　多くの代数的演算を近似代数の立場から算法化する。
目的２　すでに開発した近似代数算法の誤差解析を行って安定化を計る。
目的３　近似代数計算システムをさらに改良し算法をインプリメントする。
目的４　理工学分野への近似代数の応用を計る。

実際に行われた研究は、当初の目論見どおりとは言い難いが、次のように総括できよう。

成果１　いくつかの代数演算の近似化と基本的算法のさらなる改良。
成果２　基本演算の精密な理論解析・誤差解析およびインプリメント。
成果３　有効浮動小数パッケージの有用性の実証と使用上の注意点。
成果４　近似代数の将来への展開を目指した種々の試み。

成果は当初の予定を下回ったが、将来への大きな展望が開けたので良としたい。

2.1　近似代数の算法開発・改良について

- 近似因数分解算法の飛躍的改良（佐々木・院生、論文 5, 19）：
  多変数多項式の近似因数分解については、概念の提唱とナイーヴ算法の提案 (Sasaki et al., 1991) と線形代数的算法の考案 (Sasaki et al., 1992) 以来、世界各地で研究された（第１章参照）。その中で、佐々木は上記線形代数的算法を飛躍的に改良した。べき級数根の高次項係数の間の和零関係を求めるのが算法の中心だが、従来の算法が線形従属関係を経由して求めるものだったのに対し、新算法は簡単な行列演算で直接に和零関係を求めるものであり、計算が大幅に簡単化された。さらに、複数点でべき級数根を計算することにより、算法が効率化されることも示した。

- 多変数多項式の近似 GCD 算法の開発（野田・甲斐・他、論文 15, 16, 32, 33）：
  １変数多項式の近似 GCD は世界中で深く研究されたが、意外にも多変数多項式の近似 GCD の研究は少い。野田らは、多変数ヘンゼル構成に基づく算法を提案するとともに、多項式剰余列に基づく算法 (越知・野田・佐々木、1991) および摂動項の数値的最小化に基づく算法 (Corless et al., 1995) を比較し、最小化法は不安定だが、ヘンゼル法と剰余列法は安定に作動して望ましい結果を与えることを実証した。

- ２変数関数の浮動小数による有理関数補間法（甲斐・野田、論文 17, 28, 33）：
  有限個の点での値のみが与えられた関数を有理関数で補間することは古くから研究されてきたが、正確な数係数演算を前提としている。野田と甲斐は、１変数関数の有理式補間を浮動小数で実行すると与式にはない高く細いピークが現れるが、それは近似 GCD で除き得ることを発見していた。彼らは、２変数関数の有理式補間を

浮動小数で実行すると非常に薄く高い帯状の山脈が現れるが、2変数の近似GCDで除けることを示した。

- 安定化技法を用いた多変数多項式の近似GCD計算法（甲斐・野田）：
安定化理論を利用した多変数GCD計算の実験を行ない、入力が不正確でも正しく計算できることがあるという興味深い結果を得た。

2.2 精密な理論解析と誤差解析について

- 1変数多項式が近接根を持つ場合の部分終結式（佐々木・照井、論文1, 18）：
終結式は与えられた多項式の根差の積で表されるので、多項式が近接根を持つ場合、終結式の値は非常に小さくなる。しかし、部分終結式の係数値に対する理論は存在しなかった。そこで、その理論を構築した。この理論を用いて、剰余列算法で部分終結式を計算する場合の桁落ち量を近接根の近接度で評価する理論を作った。
照井は、重根を持つ多項式とその微分に対する剰余列を、無平方分解のように再帰的に計算する場合に関して、部分終結式理論の拡張を試みている。

- 1変数多項式の微小根に対する上界定理の導出（佐々木・照井、論文3, 13, 29）：
与えられた多項式が原点近傍に近接根を持つ場合、それらの近接根全体の上界と、他の根全体の下界を、多項式の係数で表す公式を導き出した。この公式は、多項式の係数を特徴づけるパラメータがある値のとき、上界と下界が一致するという点で、精度のよい公式になっている。さらに、$x \to 1/x$ なる変換で根全体を反転するとき、公式もちょうど反転される形になっている。

- 有理関数近似の悪条件性解明と安定化（野田・甲斐・院生、論文11, 21, 22, 23）：
有理関数近似における"望ましくない"特異点の出現が、関数補間算法の中核をなす行列の悪条件性にあることを明らかにした。そして、行列が悪条件の場合、近似的線形従属な関係式を巧妙に扱って、悪条件性を安定的に除く算法を考案した。

- 連立代数方程式に対するWuの方法の安定化（野田・甲斐・院生、論文8, 9, 25）：
Wuの方法に対する浮動小数化を検討した。閾値を用いる方法、及び、白柳・Sweedlerにより提案された安定化技法を用いる方法を実装し、浮動小数係数多項式に対するWuの方法の計算実験を行なった。その結果、入力が正確な場合は、安定化技法を組み込んだ方法が近似的に最も正しい解を与えることが分かった。

- 多変数多項式のベキ級数根の桁落ち誤差（佐々木・加古、論文30）：
多変数多項式のべき級数根をニュートン法で計算する場合、原点(＝展開点)の近傍に特異点があれば、展開次数に比例する大きな桁落ちが発生しうることを1994年に示した。今回は原点から遠方で展開する場合を調べた。その結果、展開点の近傍に特異点がなければ大きな桁落ちは発生しないが、展開点近傍に特異点があるなら、場合によって大きな桁落ちが発生することを理論的および実験例で示した。

2.3 有効浮動小数の有用性の実証

- 有効浮動小数パッケージの有用性の実証（加古・福井・佐々木・院生）：
多項式剰余列計算、有理式補間、グレブナー基底計算、(多項式の全根を計算する)DKA法、および因子分離法に対して、方法1：有効浮動小数のみで計算を実行、方法2：白柳・Sweedlerの安定化を行って有効浮動小数で計算を実行し、計算は収束し結果は正しいか、誤差の見積もりは良いか、を実例でテストした。その結果、前3者に関しては、方法1では問題が悪条件でない限り計算も誤差評価もうまく行き、悪条件の場合でも安定化により計算も誤差評価もうまく行くことが分った。しかしながら、後2者に対しては、方法1では誤差評価がデタラメで、方法2では誤差評価は良いが計算結果がデタラメになることが分った。後2者はいずれも逐次近似算法であり、毎回、残余により計算値を更新するが、通常、残余は小さくて相対的に大きな誤差を含む。しかしながら、その誤差部は毎回、小さいものに更新される。ところが、有効浮動小数は相対誤差を更新せず、安定化法は必要な残余を捨ててしまうからである。有効浮動小数を逐次近似算法で用いる場合には、毎回、誤差部を更新する仕組みが必要である。

2.4 将来の展開を目指した試みについて

- 多変数多項式の拡張ヘンゼル構成の応用（佐々木・院生、論文12, 20）：
佐々木と加古が1993年に考案した拡張ヘンゼル構成は、当初はどんな発展があるかと危惧したが、時とともに重要な演算であると思うようになった。
まず、初期因子を多項式とする場合の理論を構築し(論文12)、その構成法を多変数多項式の因数分解における非零代入問題の解決に利用することを考えた。このアイデアの有用性を実証するため、現在、院生がプログラミング中である。
次に、べき級数環の中で因数分解する、いわゆる解析的因数分解へ利用できることが分り、現在、この方面の研究を遂行中である(論文20)。

- 代数関数の近似代数的解析接続法と接続保証（佐々木・院生、論文2）：
代数関数の解析接続に関しては、1996年、佐々木と院生(当時)がSmithの誤差上界定理を用いる方法を考案したが、理論が不完全であり、さらに方法が適用できない場合があることが判明した。今回、Smithの誤差上界定理を用いる方法を改善するとともに、微小根の上界定理を新たに導出し、それを用いた方法を二つ考案した。実験の結果、新たに考案した方法の方が効率的であることを示した。
解析接続を近似代数的に行う方法は、今後、種々の演算で非常に有用になると思う。

- 自動安定化システムを用いた画像処理（甲斐・野田・院生、論文27）：
2次元の数値行列の形で与えられた画像データが不完全（データの一部が欠如したり、不正確）な場合、一般逆行列を利用して画像復元を行えば、当然、復元画像は大幅に不正確となる。しかしながら、一般逆行列の計算を安定化法を用いて行えば、かなり正確な画像が復元できることを発見した。

3 発表論文リストほか

●印を付した論文は全文を本冊に収録しているものである。

3.1 Preprints submitted

1. ● T. Sasaki: Subresultant and clusters of close roots. Proc. of 2003 Intern'l Symp. on Symbolic and Algebraic Computation, ACM (New York), 2003 (to appear).
2. ● T. Sasaki and D. Inaba: Certification of analytic continuation of algebraic functions. 14 page, June 2002 (submitted).

3.2 紙上発表

3. ● T. Sasaki and A. Terui: A formula for separating small roots of a polynomial. SIGSAM Bulletin, Vol. 36, 2002, pp. 19–29.
4. T. Sasaki, Y. Takahashi and T. Sugimoto: A divide-and-conquer method for integer-to-rational conversion. Proc. of Conference on Logic, Mathematics, and Computer Science 2002, Res. Inst. Symb. Comp. (Linz, Austria), 2002, pp. 231–243.
5. ● T. Sasaki: Approximate multivariate polynomial factorization based on zero-sum relations. Proc. of 2001 Intern'l Symp. on Symbolic and Algebraic Computation, ACM (New York), 2001, pp. 284–291.
6. A. Terui and T. Sasaki: Durand-Kerner's method for the real roots. Japan J. Injus. Appl. Math., Vol. 19, 2002, pp. 19–38.
7. ● K. Li, L.H. Zhi and M.-T. Noda: On the construction of a PSE for GCD computation. Proc. of 2001 Asian Symp. on Computer Mathematics, World Scientific, 2001, pp. 76–81.
8. ● Y. Notake, H. Kai and M.-T. Noda: Symbolic-numeric computations of Wu's method – comparison of the cut-off method and the stabilization techniques. Proc. of 2001 Asian Symp. on Computer Mathematics, World Scientific, 2001, pp. 122–130.
9. ● K. Shiraishi, H. Kai and M.-T. Noda: Symbolic-numeric computation of Wu's method using stabilizing algorithms. Proc. of 6th Asian Technology Conf. in Mathematics, RMIT University (Melbourne), 2001, pp. 444–451.
10. Y. Tsukada and T. Sasaki: On checking products of modular factors in Berlekamp-Hensel type factorization. 数式処理、第8巻、2001, pp. 3-16.
11. ● H. Kai and M.-T. Noda: Hybrid rational function approximation and its accuracy analysis. Reliable Computing, Vol. 6, 2000, pp. 429–438.
12. ● T. Sasaki and D. Inaba: Hensel construction of $F(x, u_1, \ldots, u_\ell)$, $\ell \geq 2$, at a singular point. ACM SIGSAM Bulletin, Vol. 34, 2000, pp. 9-17.
13. ● A. Terui and T. Sasaki: "Approximate zero-points" of real univariate polynomial with large error terms. 情報処理学会 論文誌, Vol. 41, 2000, pp. 974–989.
14. ● Y.N. Obukhov, T. Fukui and G.F. Rubiar: Wave propagation in linear electrodynamics. Phys. Rev. D, Vol. 62, 2000, pp. 044050-1∼5.

15. L.H.Zhi and M.-T.Noda: Approximate GCD of Multivariate Polynomials. Proc. of 4th Asian Symp. on Computer Mathematics, World Scientific, 2000, pp. 9–18.
16. L. Kai, L.H. Zhi and M.-T. Noda: Solving Approximate GCD of Multivariate Polynomials by Maple/Matlab/C Combination. Proc. of 5th Asian Technology Conf. in Mathematics, Thailand, 2000, pp. 492–499.
17. H. Kai and M.-T. Noda: Hybrid Computation of Bivariate Rational Interpolation. ACM SIGSAM Bulletin, Vol. 34, 2000, pp. 20–21.

3.3 講究録、解説記事、など

18. 佐々木建昭：部分終結式と近接根。数理研講究録、2003 （印刷中）。
19. 森田泰弘、佐々木建昭：多変数多項式の近似因数分解の効率化–複数展開点での Taylor 級数根の利用–。数理研講究録、2003 （印刷中）。
20. 岩見真希、佐々木建昭：解析的因数分解について。数理研講究録、2003 （印刷中）。
21. 村上裕美、甲斐博、野田松太郎：近似代数計算と有理関数近似に関する研究。数理研講究録、2003 （印刷中）。
22. 村上裕美、甲斐博、野田松太郎：区間演算によるハイブリッド有理関数近似と安定化理論について。数理研講究録、第 1295 巻、2002、pp. 197-202。
23. ● 村上裕美、甲斐博、野田松太郎：有理関数近似と離散化における問題点。数理研講究録、第 1286 巻、2002、pp. 34-50。
24. 野田松太郎：数式処理と Web コンピューティング。ソフトウェア工学の基礎 IX（分担執筆）、日本ソフトウェア科学会、2002、pp. 7–12。
25. 野竹 禎雄、甲斐 博、支 麗紅、野田 松太郎：Wu's method の浮動小数化。数理研講究録、第 1199 巻、2001, pp. 1–9。
26. 白石 啓一、那須 英正、甲斐 博、野田 松太郎：Wu の方法の並列化における負荷分散について。数理研講究録、第 1199 巻、2001, pp. 10–19。
27. 水口 寛之、甲斐 博、野田 松太郎：自動安定化システムを用いた画像処理について。数理研講究録、第 1199 巻、2001, pp. 20–21 。
28. 甲斐 博、野田 松太郎：二変数ハイブリッド有理関数近似の誤差評価。数理研講究録、第 1199 巻、2001, pp. 36–42。
29. 佐々木建昭、照井 章：微小低次項を持つ代数方程式の根の大きさについて。数理研講究録、第 1199 巻、2001, pp. 132–136。
30. 佐々木建昭、加古富志雄：多変数多項式のべき級数根の桁落ち誤差–その 2 。数理研講究録、第 1199 巻、2001, pp. 137–148。
31. 照井章：誤差項をもつ実多項式の「近似実根」の計算とその応用。数理研講究録、第 1138 巻、2000, pp. 43–55。
32. L.H. Zhi and M.-T. Noda: Approximate GCD of Multivariate Polynomials. 数理研講究録、第 1138 巻、2000, pp. 64–76。
33. 甲斐 博、木原 信二、野田 松太郎：二変数有理関数近似のハイブリッド計算と多変数近似 GCD アルゴリズム。数理研講究録、第 1138 巻、2000, pp. 77–86。

The Subresultant and Clusters of Close Roots *

Tateaki Sasaki
Institute of Mathematics, University of Tsukuba
Tsukuba-shi, Ibaraki 305, Japan
sasaki@math.tsukuba.ac.jp

Abstract

This paper investigates the subresultant of univariate polynomials from the viewpoint of close roots. First, we derive formulas which express the subresultant and its cofactors in the root-differences. Then, we consider the case that the given polynomials contain one or more clusters of mutually close roots of closeness δ. We derive formulas showing the dependences of the coefficients of subresultant and its cofactors on δ and the number of clusters. The formulas are consistent with the famous formula: resultant $\propto$ [product of all the root-differences]. Finally, we determine the magnitudes of cancellations which occur when we apply the Euclidean algorithm to polynomials having close roots.

1 Introduction

The subresultant is a very important concept in computer algebra, it not only provides us practical methods for computing polynomial GCD's [Col67, BT71] and performing the quantifier elimination [Col75, CJ96], and so on, but also helps us in various theoretical analyses.

Let $R_k(A,B)$ be the kth subresultant of univariate polynomials $A, B \in \mathbf{C}[X]$. $R_k(A,B)$ is, except for a constant multiple, equal to an element of polynomial remainder sequence generated by A and B. $R_0(A,B)$ is the resultant of A and B, and quite many things are known on the resultant; in particular, we have a very beautiful and useful formula which expresses $R_0(A,B)$ in differences of the roots of A and those of B, or the root-differences. The subresultant has been investigated intensively and into details since its discovery in the middle of 19th century; see [GG99, Notes of Sec. 6] for the history of the subresultant. However, compared with rich knowledge on the resultant, we have still many questions on the subresultant: Q1: are there simple and useful formulas expressing $R_k(A,B)$ and its cofactors in the root-differences ?, Q2: can we express $R_k(A,A')$ and its cofactors, with $A' = \mathrm{d}A/\mathrm{d}X$, in the differences of the roots of A ?, Q3: how $R_1(A,B), R_2(A,B), \cdots$ behave when A and B have mutually close roots ?, Q4: what amounts of cancellations occur in the computation of $R_k(A,B)$ by the Euclidean algorithm ?, and so on.

As for question Q1, Sylvester [Syl1853] derived a formula, and recently Hoon [Hoo02, Hoo01A, Hoo01B] also found another formula, see Sec. 3 for these formulas. Sylvester's formula

*Work supported in part by Japanese Ministry of Education, Science and Culture under Grants 12480065.

is very beautiful but contains so many terms to handle practically. Hoon's formula is much shorter but still rather complicated. In Sec. 3, we show that the subresultant and its cofactors can be expressed in root-differences by a simple variable transformation. Our formulas are useful for investigating questions Q3 and Q4 when A and B have only one cluster of mutually close roots. However, it is not enough for investigating the case that A and B have several clusters of mutually close roots.

As for question Q3, we remind of the fact: let A and B have n' mutually close roots and $(P_1 = A, P_2 = B, \cdots, P_k, \cdots)$ be the polynomial remainder sequence, with $\deg(P_k) = n'$, then P_k is an approximate common divisor of A and B. Furthermore, some remainder P_{k+i} with a larger index is an approximate common divisor of a smaller tolerance. Why does such a phenomenon occur ? As for P_k and P_{k+1}, [SN89], [HS97] and [SS97] investigated the phenomenon by using relations on the polynomial remainder sequence. In Secs. 4 and 5, we investigate ideal cases that A and B have one or more clusters of close roots of closeness δ, and determine the dependences of the coefficients of $P_k, P_{k+1}, \cdots$ on δ and the number of clusters. By this, we can clarify the phenomenon considerably. However, bounding $\|P_k\|, \|P_{k+1}\|, \cdots$ reasonably is an open problem.

The Euclidean algorithm with floating-point numbers often causes large cancellation errors. As for question Q4, [SS89] investigated the cancellation errors in typical cases, from the viewpoint of polynomial division. In Sec. 6, we answer to the question by using the estimations of the magnitudes of the subresultants and their cofactors.

2 Generalities

By $\deg(P)$, $\mathrm{lc}(P)$, $\|P\|$, with P a univariate polynomial, we denote the degree, the leading coefficient, and the norm (may be, the infinity norm), respectively. By $\mathrm{quo}(P, Q)$ and $\mathrm{rem}(P, Q)$, with P and Q univariate polynomials, we denote the quotient and the remainder, respectively, of P divided by Q.

Let $A(X)$ and $B(X)$ be univariate polynomials over $\mathbf{C}$, being expressed as $(a_m \neq 0, b_n \neq 0)$

$$\begin{aligned} A(X) &= a_m X^m + a_{m-1} X^{m-1} + \cdots + a_0, \\ B(X) &= b_n X^n + b_{n-1} X^{n-1} + \cdots + b_0. \end{aligned} \tag{2.1}$$

Let $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_n$ be the roots of $A(X)$ and $B(X)$, respectively:

$$\begin{aligned} A(X) &= a_m (X - \alpha_1) \cdots (X - \alpha_m), \\ B(X) &= b_n (X - \beta_1) \cdots (X - \beta_n). \end{aligned} \tag{2.2}$$

Definition 1 (regular polynomial(s)) *We call the polynomial $A(X)$ regular if $a_m = \max\{|a_{m-1}|, \cdots, |a_0|\} = 1$. We call the pair of polynomials $\langle A(X), B(X)\rangle$ regular if $a_m = b_n = \max\{|a_{m-1}|, \cdots, |a_0|, |b_{n-1}|, \cdots, |b_0|\} = 1$.*

Remark 1 The well-known formula $\max\{|\alpha_1|, \cdots, |\alpha_m|\} \leq 1 + \max\{|a_{m-1}|, \cdots, |a_0|\}/|a_m|$ tells us that $|\alpha_i| \leq 2$ $(i = 1, \ldots, m)$ if $A(X)$ is regular and that $|\alpha_i| \leq 2$ $(i = 1, \ldots, m)$ and $|\beta_i| \leq 2$ $(i = 1, \ldots, n)$ if $\langle A(X), B(X)\rangle$ is regular. Furthermore, at least one root of a regular polynomial is not much smaller than 1. □

The kth subresultant of $A(X)$ and $B(X)$, $R_k(A,B)$, is defined by the following determinant.

$$R_k(A,B) = \begin{vmatrix} a_m & a_{m-1} & \cdots & \cdots & \cdots & a_{2k+2-n} & X^{n-k-1}A \\ & a_m & a_{m-1} & \cdots & \cdots & a_{2k+3-n} & X^{n-k-2}A \\ & & \ddots & \ddots & \cdots & \vdots & \vdots \\ & & & a_m & \cdots & a_{k+1} & X^0 A \\ b_n & b_{n-1} & \cdots & \cdots & \cdots & b_{2k+2-m} & X^{m-k-1}B \\ & b_n & b_{n-1} & \cdots & \cdots & b_{2k+3-m} & X^{m-k-2}B \\ & & \ddots & \ddots & \cdots & \vdots & \vdots \\ & & & b_n & \cdots & b_{k+1} & X^0 B \end{vmatrix}, \tag{2.3}$$

where $a_j = b_j = 0$ for $j < 0$ and the blank denotes 0. R_k is expressed as $R_k = R_{k,m+n-k-1}X^{m+n-k-1} + \cdots + R_{k,0}$, where each $R_{k,i}$ is a numerical determinant. For $i > k$, the last column of $R_{k,i}$ is a sum of other columns, hence $\deg(R_k) \leq k$ and $R_0(A,B) = \text{resultant}(A,B)$.

For each $R_k(A,B)$, there exist polynomials $S_k(A,B)$ and $T_k(A,B)$ satisfying

$$\begin{aligned} R_k(A,B) &= S_k(A,B)\,A(X) + T_k(A,B)\,B(X), \\ \deg(S_k) &< n-k, \quad \deg(T_k) < m-k. \end{aligned} \tag{2.4}$$

S_k and T_k are called *cofactors* and expressed by determinants which are the same as that in (2.3) except that the last column is replaced by ${}^t(X^{n-k-1}, X^{n-k-2}, \cdots, 1, 0, 0, \cdots, 0)$ and ${}^t(0, 0, \cdots, 0, X^{m-k-1}, X^{m-k-2}, \cdots, 1)$, respectively.

The next theorem is well known; we give a proof for later use. Put

$$C(X) = c_l X^l + c_{l-1}X^{l-1} + \cdots + c_0. \tag{2.5}$$

Theorem 1 *We have the following equalities.*

$$R_k(AC,BC) = c_l^{m+n+2l-2k-1} R_{k-l}(A,B)\,C, \tag{2.6}$$

$$S_k(AC,BC) = c_l^{m+n+2l-2k-1} S_{k-l}(A,B), \tag{2.7}$$

$$T_k(AC,BC) = c_l^{m+n+2l-2k-1} T_{k-l}(A,B). \tag{2.8}$$

We can use these formulas for $k < l$, by defining $R_j = S_j = T_j = 0$ for $j < 0$.

Proof $R_k(AC,BC)$ is given by

$$\begin{vmatrix} c_l a_m & c_l a_{m-1} + c_{l-1}a_m & c_l a_{m-2} + c_{l-1}a_{m-1} + c_{l-2}a_m & \cdots \\ & c_l a_m & c_l a_{m-1} + c_{l-1}a_m & \cdots \\ & & \ddots & \cdots \\ c_l b_n & c_l b_{n-1} + c_{l-1}b_n & c_l b_{n-2} + c_{l-1}b_{n-1} + c_{l-2}b_n & \cdots \\ & \ddots & \ddots & \cdots \end{vmatrix},$$

where the rightmost column is ${}^t(X^{n+l-k-1}AC, \cdots, X^0 AC, X^{m+l-k-1}BC, \cdots, X^0 BC)$. Subtracting $c_{l-1}/c_l\times$ (1st column) from the second, $c_{l-2}/c_l\times$(1st column) + $c_{l-1}/c_l\times$(resulting 2nd

column) from the third, and so on (the last column is unchanged), we can transform the above determinant into the following one.

$$\begin{vmatrix} c_l a_m & c_l a_{m-1} & c_l a_{m-2} & \cdots & X^{n+l-k-1}AC \\ & c_l a_m & c_l a_{m-1} & \cdots & X^{n+l-k-2}AC \\ & & \ddots & \cdots & \vdots \\ c_l b_n & c_l b_{n-1} & c_l b_{n-2} & \cdots & X^{m+l-k-1}BC \\ & \ddots & \ddots & \cdots & \vdots \end{vmatrix}$$

This is nothing but $c_l^{m+n+2l-2k-1}R_{k-l}(A,B)\,C$, and we obtain (2.6). Then, uniqueness of $S_k(A,B)$ and $T_k(A,B)$ leads us to (2.7) and (2.8). □

We generalize Theorem 1. Let $C(X)$ be monic for simplicity, hence $c_l = 1$, and let $D(X)$ and $E(X)$ be the following polynomials:

$$\begin{aligned} D(X) &= d_{l-1}X^{l-1} + \cdots + d_0, \\ E(X) &= e_{l-1}X^{l-1} + \cdots + e_0. \end{aligned} \tag{2.9}$$

Expanding X^l/C into the power series in X as $X^l/C = 1 - c_{l-1}/X - (c_{l-2} - c_{l-1}^2)/X^2 - (c_{l-3} - 2c_{l-2}c_{l-1} + c_{l-1}^3)/X^3 + \cdots$, we define $\breve{d}_i$ and $\breve{e}_i$ $(i = l-1, l-2, \ldots)$ as follows.

$$\begin{aligned} \breve{D}(X) &\stackrel{\text{def}}{=} D/C = \breve{d}_{l-1}X^{-1} + \breve{d}_{l-2}X^{-2} + \cdots, \\ \breve{E}(X) &\stackrel{\text{def}}{=} E/C = \breve{e}_{l-1}X^{-1} + \breve{e}_{l-2}X^{-2} + \cdots \end{aligned} \tag{2.10}$$

For example, $\breve{d}_{l-1} = d_{l-1}$, $\breve{d}_{l-2} = d_{l-2} - c_{l-1}d_{l-1}$, $\breve{d}_{l-3} = d_{l-3} - c_{l-1}d_{l-2} - (c_{l-2} - c_{l-1}^2)d_{l-1}$. Note that we define $\breve{d}_i$ and $\breve{e}_i$ for negative i. Although $\breve{D}$ and $\breve{E}$ are power series, we define $R_k(X^l(A+\breve{D}), X^l(B+\breve{E}))$ formally as in (2.3). We define $\breve{R}_{k-l}(P,Q)$, a determinant of degree $m+n+2l-2k$, by replacing the rightmost column of $R_{k-l}(X^l(A+\breve{D}), X^l(B+\breve{E}))$ by ${}^t(X^{n+l-k-1}P, \cdots, X^0P, X^{m+l-k-1}Q, \cdots, X^0Q)$:

$$\breve{R}_{k-l}(P,Q) = \begin{vmatrix} a_m & \cdots & a_0 & \breve{d}_{l-1} & \breve{d}_{l-2} & \cdots & X^{n+l-k-1}P \\ & a_m & \cdots & a_0 & \breve{d}_{l-1} & \cdots & X^{n+l-k-2}P \\ & & \ddots & \cdots & \ddots & \cdots & \vdots \\ b_n & \cdots & b_0 & \breve{e}_{l-1} & \breve{e}_{l-2} & \cdots & X^{m+l-k-1}Q \\ & b_n & \cdots & b_0 & \breve{e}_{l-1} & \cdots & X^{m+l-k-2}Q \\ & & \ddots & \cdots & \ddots & \cdots & \vdots \end{vmatrix}. \tag{2.11}$$

Theorem 2 *We have the following equalities.*

$$R_k(AC+D, BC+E) = \breve{R}_{k-l}(A,B)\,C + \breve{R}_{k-l}(D,E), \tag{2.12}$$

$$S_k(AC+D, BC+E) = \breve{R}_{k-l}(1,0), \tag{2.13}$$

$$T_k(AC+D, BC+E) = \breve{R}_{k-l}(0,1). \tag{2.14}$$

Proof Applying the transformation given in the proof of Theorem 1, we can transform the left $m+n+2l-2k-1$ columns of $R_k(AC+D, BC+E)$ into those of $\breve{R}_k(AC+D, BC+E)$. (We may

apply Theorem 1 to $R_k(AC{+}D, BC{+}E)$ by putting $AC{+}D = (A{+}\breve{D})C$ and $BC{+}E = (B{+}\breve{E})C$.) Separating the rightmost column into two columns containing A and B and containing D and E, we obtain (2.12). Similarly, we obtain (2.13) and (2.14). □

Remark 2 If $\min\{m,n\} > m{+}n{+}2l{-}2k{-}2$ then neither $\breve{d}_{l-1}$ nor $\breve{e}_{l-1}$ appears in $\breve{R}_{k-l}(P,Q)$ hence $\breve{R}_{k-l}(A,B) = R_{k-l}(A,B)$. If, however, $\min\{m,n\} \le m{+}n{+}2l{-}2k{-}2$ then $\breve{d}_{l-1}$ and/or $\breve{e}_{l-1}$ appears in $\breve{R}_{k-l}(P,Q)$ and $\deg(\breve{R}_{k-l}(A,B)) = \max\{m,n\} + l - k > k - l$. The terms of degrees $> k$ in $\breve{R}_{k-l}(A,B)\,C$ are cancelled by the higher degree terms of $\breve{R}_{k-l}(D,E)$. □

3 Subresultant in root-differences

There is a very beautiful formula for $R_0(A,B)$: $R_0(A,B) = a_m^n b_n^m \prod_{i=1}^m \prod_{j=1}^n (\alpha_i - \beta_j)$. This beautiful formula seems to make many researchers search for formulas of the subresultants in root-differences $(\alpha_i - \beta_j)$'s. The most beautiful formula is due to Sylvester [Syl1853] (see also [LP01] for a modern proof of the formula):

$$\left\{\begin{array}{l} R_k(A,B) \;=\; \displaystyle\sum_{\substack{I,J \\ |I|+|J|=k}} \frac{R_0(A_I,B_J)\,R_0(A_{\bar{I}},B_{\bar{J}})}{R_0(A_I,A_{\bar{I}})\,R_0(B_J,B_{\bar{J}})}\, A_I(X)B_J(X), \\ \qquad I \subset \{1,2,\cdots,m\} = I \cup \bar{I}, \quad I \cap \bar{I} = \phi, \\ \qquad J \subset \{1,2,\cdots,n\} = J \cup \bar{J}, \quad J \cap \bar{J} = \phi, \\ A_I(X) = \prod_{i\in I}(X-\alpha_i) \;\text{ and }\; B_J(X) = \prod_{j\in J}(X-\beta_j). \end{array}\right. \tag{3.1}$$

The formula contains so many terms to handle practically. Recently, Hoon [Hoo02] derived a much shorter but a rather complicated formula:

$$\left\{\begin{array}{l} R_k(A,B) = \displaystyle\sum_{j=1}^{k+1} |M_{k,j}|\,(X-\alpha_1)\cdots(X-\alpha_{j-1}), \\ M \;=\; \displaystyle\prod_{j=1}^{n} \begin{pmatrix} \alpha_1-\beta_j & 1 & & \\ & \ddots & \ddots & \\ & & \alpha_m-\beta_j & 1 \end{pmatrix}, \\ M_{k,j} \Longleftarrow \left\{\begin{array}{l} j\text{-th \& right } m{-}k{-}1 \text{ columns,} \\ \text{and lower } m{-}k \text{ rows of } M. \end{array}\right. \end{array}\right. \tag{3.2}$$

We derive simple formulas which express R_k, S_k and T_k in the root-differences. Let γ be a number. By shifting the origin by γ, we put $\hat{A}(X) \stackrel{\text{def}}{=} A(X{+}\gamma)$ and $\hat{B}(X) \stackrel{\text{def}}{=} B(X{+}\gamma)$ as follows ($\hat{a}_m = a_m$, $\hat{b}_n = b_n$).

$$\begin{array}{rcl} \hat{A}(X) &=& a_m(X+\gamma-\alpha_1)\cdots(X+\gamma-\alpha_m), \\ &=& \hat{a}_m X^m + \hat{a}_{m-1}X^{m-1} + \cdots + \hat{a}_0, \\ \hat{B}(X) &=& b_n(X+\gamma-\beta_1)\cdots(X+\gamma-\beta_n), \\ &=& \hat{b}_n X^n + \hat{b}_{n-1}X^{n-1} + \cdots + \hat{b}_0. \end{array} \tag{3.3}$$

Lemma 1 *The next equality holds for any number γ.*

$$R_k(A,B) = R_k(\hat{A},\hat{B})|_{X\to X-\gamma}. \tag{3.4}$$

Proof We express A and B as

$$A(X) = \hat{a}_m(X-\gamma)^m + \hat{a}_{m-1}(X-\gamma)^{m-1} + \cdots + \hat{a}_0,$$
$$B(X) = \hat{b}_n(X-\gamma)^n + \hat{b}_{n-1}(X-\gamma)^{n-1} + \cdots + \hat{b}_0.$$

The subresultant $R_k(\hat{A},\hat{B})$ is obtained by eliminating terms $X^{\max\{m,n\}}, X^{\max\{m,n\}-1}, \ldots, X^{k+1}$, from $\hat{A}$ and $\hat{B}$ successively. Similarly, we obtain $R_k(A,B)$ by expressing A and B as above and eliminating terms $(X-\gamma)^{\max\{m,n\}}, (X-\gamma)^{\max\{m,n\}-1}, \ldots, (X-\gamma)^{k+1}$, successively. Both processes of elimination are the same, which proves the lemma. □

By choosing $\gamma = \alpha_1$, we obtain the following theorem. Note that $\hat{a}_{m-i}/a_m$ $(1 \le i \le m)$ and $\hat{b}_{n-i}/b_n$ $(1 \le i \le n)$ in this choice are the elementary symmetric polynomials of degree i in $(\alpha_1-\alpha_2), \ldots, (\alpha_1-\alpha_m)$ and in $(\alpha_1-\beta_1), \ldots, (\alpha_1-\beta_n)$, respectively.

Theorem 3 *We can express R_k, S_k and T_k in root-differences as follows (below, $\hat{X} = X-\alpha_1$).*

$$R_k(A,B) = \begin{vmatrix} \hat{a}_m & \hat{a}_{m-1} & \cdots & \cdots & \hat{a}_{2k+2-n} & \hat{X}^{n-k-1}\hat{A}(\hat{X}) \\ & \ddots & \ddots & \cdots & \vdots & \vdots \\ & & \hat{a}_m & \cdots & \hat{a}_{k+1} & \hat{X}^0\hat{A}(\hat{X}) \\ \hat{b}_n & \hat{b}_{m-1} & \cdots & \cdots & \hat{b}_{2k+2-m} & \hat{X}^{m-k-1}\hat{B}(\hat{X}) \\ & \ddots & \ddots & \cdots & \vdots & \vdots \\ & & \hat{b}_n & \cdots & \hat{b}_{k+1} & \hat{X}^0\hat{B}(\hat{X}) \end{vmatrix}, \tag{3.5}$$

$$S_k(A,B) = \textit{replace the rightmost column by } {}^t((X-\alpha_1)^{n-k-1}, \cdots, (X-\alpha_1)^0, 0, \cdots, 0), \tag{3.6}$$

$$T_k(A,B) = \textit{replace the rightmost column by } {}^t(0, \cdots, 0, (X-\alpha_1)^{m-k-1}, \cdots, (X-\alpha_1)^0). \tag{3.7}$$

Corollary 1 *We can express $R_k(A,A')$, with $A'(X) = \mathrm{d}A(X)/\mathrm{d}X$, in the differences of the roots of $A(X)$.*

Proof The coefficients of $A(X-\alpha_1)$ are expressed in the root-differences $(\alpha_2-\alpha_1), \ldots, (\alpha_m-\alpha_1)$, hence so are the coefficients of $A'(X)$. Thus, the corollary is a direct consequence of Theorem 3. □

4 A cluster of close roots

We assume that $\langle A(X), B(X)\rangle$ is regular and $A(X)$ and $B(X)$ have no common root. If $A(X)$ and $B(X)$ have mutually close roots, $|R_0(A,B)|$ and also $\|R_k(A,B)\|$ with small k are very small. On the other hand, a_i and b_i are not small usually, hence there must occur large cancellations if we expand the subresultant determinant. How small is $\|R_k(A,B)\|$? If $A(X)$ and $B(X)$ have only one cluster of close roots, Lemma 1 allows us to estimate the magnitude of $\|R_k(A,B)\|$; the idea is to choose γ so that the coefficients of $R_k(A(X+\gamma), B(X+\gamma))$ become small. Below, by δ we denote a small positive number. By $|a| = O(\delta^k)$ and $|a| \le O(\delta^k)$ we denote $\lim_{\delta\to 0} |a|/\delta^k \neq 0, \infty$ and $\lim_{\delta\to 0} |a|/\delta^k \neq \infty$, respectively.

We consider a simple case that some roots of $A(X)$ and $B(X)$ form a cluster of close roots, of closeness $\delta \ll 1$, and other roots of $A(X)$ and $B(X)$ are distant each other and from the cluster. We assume that the cluster is located at $X = \gamma$ and contains m' roots of $A(X)$ and n' roots of $B(X)$. Precisely, we assume that all the root-differences $|\alpha_i - \alpha_j|$'s, $|\beta_i - \beta_j|$'s and $|\alpha_i - \beta_j|$'s are of magnitude $O(\delta^0)$ except that

$$\begin{aligned} |\alpha_i - \gamma| &= O(\delta) \quad (1 \le i \le m'), \\ |\beta_i - \gamma| &= O(\delta) \quad (1 \le i \le n'). \end{aligned} \tag{4.1}$$

Without loss of generality, we assume that $m' \ge n'$.

One may specify the cluster to be a disc of radius δ, which contains the close roots. This specification is less precise than ours because some close roots may be much closer to each other than δ, hence $\|R_k(A,B)\|$ may be much smaller than those predicted below.

We express the close-root factors of $A(X)$ and $B(X)$ as

$$\begin{aligned} \textstyle\prod_{j=1}^{m'}(X-\alpha_j) &= (X-\gamma)^{m'} + \acute{a}_{m'-1}(X-\gamma)^{m'-1} + \cdots + \acute{a}_0, \\ \textstyle\prod_{j=1}^{n'}(X-\beta_j) &= (X-\gamma)^{n'} + \acute{b}_{n'-1}(X-\gamma)^{n'-1} + \cdots + \acute{b}_0. \end{aligned} \tag{4.2}$$

We can determine the coefficient $\acute{a}_i$, for example, as follows. Rewrite $(X-\alpha_j)$ $(1 \le j \le m')$ as $(X-\gamma)+(\gamma-\alpha_j)$ and expand the l.h.s. products in (4.2) w.r.t. $(X-\gamma)$. Then, $\acute{a}_i$ is given by the $(m'-i)$th symmetric polynomial in $(\gamma-\alpha_1),\ldots,(\gamma-\alpha_{m'})$. Hence, we have

$$\begin{aligned} |\acute{a}_i| &\le O(\delta^{m'-i}) \quad (i = m'-1,\ldots,1), \\ |\acute{b}_i| &\le O(\delta^{n'-i}) \quad (i = n'-1,\ldots,1). \end{aligned} \tag{4.3}$$

If $B(X) = \mathrm{d}A(X)/\mathrm{d}X$ and $\gamma = (\alpha_1 + \cdots + \alpha_{m'})/m'$ then $\acute{a}_{m'-1} = 0$. Note that $n' = m'-1$ in this case.

Lemma 2 *We can express $A(X)$ and $B(X)$ as follows.*

$$\begin{aligned} A(X) &= \hat{a}_m^{(0)}(X-\gamma)^m + \cdots + \hat{a}_{m'}^{(0)}(X-\gamma)^{m'} + \hat{a}'_{m'-1}(X-\gamma)^{m'-1} + \cdots + \hat{a}'_0, \\ B(X) &= \hat{b}_n^{(0)}(X-\gamma)^n + \cdots + \hat{b}_{n'}^{(0)}(X-\gamma)^{n'} + \hat{b}'_{n'-1}(X-\gamma)^{n'-1} + \cdots + \hat{b}'_0, \end{aligned} \tag{4.4}$$

where $|\hat{a}_i^{(0)}| \le O(\delta^0)$ and $|\hat{b}_i^{(0)}| \le O(\delta^0)$ for any i, and

$$\begin{aligned} |\hat{a}'_i| &\le O(\delta^{\max\{1,\,m'-i\}}) \quad (i = m'-1,\ldots,1,0), \\ |\hat{b}'_i| &\le O(\delta^{\max\{1,\,n'-i\}}) \quad (i = n'-1,\ldots,1,0). \end{aligned} \tag{4.5}$$

If $B(X) = \mathrm{d}A(X)/\mathrm{d}X$ and $\gamma = (\alpha_1 + \cdots + \alpha_{m'})/m'$ then

$$\begin{aligned} |\hat{a}'_{m'-1}| &= 0, \quad |\hat{a}'_i| \le O(\delta^{\max\{2,\,m'-i\}}) \quad (i = m'-2,\ldots,0), \\ |\hat{b}'_{n'-1}| &= 0, \quad |\hat{b}'_i| \le O(\delta^{\max\{2,\,n'-i\}}) \quad (i = n'-2,\ldots,0). \end{aligned} \tag{4.6}$$

Proof In the case of $B(X) \ne \mathrm{d}A(X)/\mathrm{d}X$, put $Q_A = a_m \prod_{i=m'+1}^{m}(X-\alpha_i)$ and $Q_B = b_n \prod_{i=n'+1}^{n}(X-\beta_i)$. Multiplying the r.h.s. expressions in (4.2) to Q_A and Q_B, respectively, and expanding Q_A and Q_B in powers of $(X-\gamma)$, we obtain (4.4). Then, (4.3) leads us to (4.5).

If $B(X) = \mathrm{d}A(X)/\mathrm{d}X$, put Q_A as above and set Q_B as $Q_B = m'Q_A + (X-\gamma)\,[\mathrm{d}Q_A/\mathrm{d}X]$. Then, we obtain (4.4). Setting $\gamma = (\alpha_1 + \cdots + \alpha_{m'})/m'$, we have $\acute{a}_{m'-1} = 0$ and $\hat{b}'_{n-1} = 0$, hence (4.3) leads us to (4.6). □

Proposition 1 *Under the assumptions in* (4.1), *we have*

$$\begin{aligned} &\|R_k(A,B)\| = O(\delta^0) \quad (k \geq n'), \\ &\|R_k(A,B)\| \leq O(\delta^{(m'-k)(n'-k)}) \quad (n' > k \geq 0). \end{aligned} \tag{4.7}$$

Let $R_k(A,B)$, *with* $k \leq n'$, *be expressed as*

$$R_k(A,B) = \hat{r}_k(X-\gamma)^k + \hat{r}_{k-1}(X-\gamma)^{k-1} + \cdots + \hat{r}_0. \tag{4.8}$$

Then, we have (note that $n' = m'-1$ *if* $B = \mathrm{d}A/\mathrm{d}X$*)*

$$\begin{aligned} &|\hat{r}_{k-i}| = O(\delta^{(m'-k)(n'-k)+i}) \quad (k \leq n',\ i \leq k), \\ &|\hat{r}_{n'-1}| \leq O(\delta^2) \ \ \textit{if}\ k = n' \ \textit{and}\ B = \mathrm{d}A/\mathrm{d}X. \end{aligned} \tag{4.9}$$

Proof We separate each row of the subresultant determinant $R_k(A(X+\gamma), B(X+\gamma))$ as follows (below, we show only rows for $A(X+\gamma)$, but we define $B^{(0)}$-row and B'-row for $B(X+\gamma)$, too):

$$\begin{aligned} &(\cdots, \hat{a}_m^{(0)}, \cdots, \hat{a}_{m'}^{(0)}, \hat{a}'_{m'-1}, \cdots, \hat{a}'_0, \cdots, X^i A) \\ &= A^{(0)}\text{-row} + A'\text{-row}, \quad \text{where} \\ &\quad A^{(0)}\text{-row} = (\cdots, \hat{a}_m^{(0)}, \cdots, \hat{a}_{m'}^{(0)}, 0, \cdots, X^i A^{(0)}), \\ &\qquad \text{with } A^{(0)} = \hat{a}_m^{(0)} X^m + \cdots + \hat{a}_{m'}^{(0)} X^{m'}, \\ &\quad A'\text{-row} = (\cdots, 0, \hat{a}'_{m'-1}, \cdots, \hat{a}'_0, \cdots, X^i A'), \\ &\qquad \text{with } A' = \hat{a}'_{m'-1} X^{m'-1} + \cdots + \hat{a}'_0. \end{aligned}$$

By $R_k^{(0)}$ we denote the kth subresultant composed of the $A^{(0)}$-rows and the $B^{(0)}$-rows. The coefficients $|\hat{a}'_{m'-i}|$ and $|\hat{b}'_{n'-i}|$ decrease by magnitude $O(\delta)$ as i increases by 1. If we replace μ $A^{(0)}$-rows of $R_k^{(0)}$ by the corresponding A'-rows, the largest magnitude determinant is obtained when we replace the lower μ rows. Hence, we consider only replacing lower $A^{(0)}$-rows or lower $B^{(0)}$-rows. By $R_k^{(\mu,0)}(A')$ and $R_k^{(0,\nu)}(B')$ we denote determinants obtained by replacing the lower μ $A^{(0)}$-rows and the lower ν $B^{(0)}$-rows of $R_k^{(0)}$ by the corresponding μ A'-rows and ν B'-rows, respectively.

For $k \geq n'$, $R_k^{(0)}$ contains no 0-column. Hence, $\|R_k\| = O(\delta^0)$ because $\mathrm{resultant}(Q_A, Q_B) = O(\delta^0)$ by assumption. Note that, for $k = n'$, $R_k^{(0)}$ contribute to only $\mathrm{lc}(R_k)$.

For $k < n'$, $R_k^{(0)}$ does not contribute to R_k, because $R_k^{(0)}$ contains only $m+n-k-n'$ nonzero columns and $m+n-k-n' < m+n-2k$. In order to obtain nonzero determinant, we must replace at least $n'-k$ lower $A^{(0)}$-rows by the corresponding A'-rows or $m'-k$ lower $B^{(0)}$-rows by the corresponding B'-rows. Hence, we estimate the coefficients of R_k by either $R_k^{(n'-k,0)}(A')$ or $R_k^{(0,m'-k)}(B')$. Let L be either an $(n'-k)\times(n'-k)$ submatrix constructed by the $(n'-k)$ A' rows and the right $n'-k$ columns of $R_k^{(n'-k,0)}(A')$ or an $(m'-k)\times(m'-k)$ submatrix constructed by the $(m'-k)$ B' rows and the right $m'-k$ columns of $R_k^{(0,m'-k)}(B')$, respectively. We show the matrix L for $R_k^{(0,m'-k)}(B')$:

$$L = \begin{pmatrix} \hat{b}'_k & \hat{b}'_{k-1} & \cdots & X^{m'-k-1}B' \\ \hat{b}'_{k+1} & \hat{b}'_k & \cdots & X^{m'-k-2}B' \\ \vdots & \ddots & \ddots & \vdots \\ \hat{b}'_{m'-1} & \hat{b}'_{m'-2} & \cdots & X^0 B' \end{pmatrix}. \tag{4.10}$$

Since $\hat{a}_i^{(0)} = O(\delta^0)$ and $\hat{b}_i^{(0)} = O(\delta^0)$, we can order estimate $\|R_k\|$ by the determinant $|L|$.

We can express $|L|$ as $|L| = |L_k|X^k + |L_{k-1}|X^{k-1} + \cdots$, where each L_i is a numerical matrix. Let l be $l = n' - k$ for $R_k^{(n'-k,0)}(A')$ and $l = m' - k$ for $R_k^{(0,m'-k)}(B')$. Putting $L_k = (e_{ij})$, we have $e_{ij} = 0$ or $e_{ij} \propto O(\delta^{l-i+j})$ if $B \neq \mathrm{d}A/\mathrm{d}X$. We can order estimate $\mathrm{lc}(R_k)$ by the determinant $|L_k|$. Expanding the determinant, we obtain terms $e_{i_1 j_1} \cdots e_{i_l j_l}$'s, where $\{i_1, \cdots, i_l\} = \{j_1, \cdots, j_l\} = \{1, \cdots, l\}$. Since $e_{i_1 j_1} \cdots e_{i_l j_l} \propto O(\delta^{l^2 - (i_1 + \cdots + i_l) + (j_1 + \cdots + j_l)}) = O(\delta^{l^2})$, each term is of the same order. Hence, we estimate $\mathrm{lc}(R_k)$ by the product of diagonal elements of L_k. Thus, both $R_k^{(n'-k,0)}(A')$ and $R_k^{(0,m'-k)}(B')$ give the same estimation :

$$\begin{aligned} |\mathrm{lc}(R_k)| &\leq O((\hat{b}'_{n'-(n'-k)})^{m'-k}) = O(\delta^{(n'-k)(m'-k)}), \\ |\mathrm{lc}(R_k)| &\leq O((\hat{a}'_{m'-(m'-k)})^{n'-k}) = O(\delta^{(m'-k)(n'-k)}). \end{aligned}$$

In the case of $B = \mathrm{d}A/\mathrm{d}X$, we have $|\hat{a}'_{m'-1}| \leq O(\delta^2)$ and $|\hat{b}'_{n'-1}| \leq O(\delta^2)$. If $\hat{a}'_{m'-1}$ or $\hat{b}'_{n'-1}$ appears as the diagonal elements of L_k, we consider non-diagonal elements, obtaining (4.7).

Finally, we consider the non-leading coefficients $\hat{r}_{k-i}$ $(i < k)$. Note that, for $k > n'$, we must consider $R_k^{(1,0)}(A')$ and $R_k^{(0,\max\{1,m'-k\})}(B')$. The $\hat{r}_{k-i}$ $(i < k)$ is estimated by $|L_{k-i}|$, and we obtain the upper estimation in (4.9) in most cases. Only one case we must be careful is that of $k = n'$ and $B = \mathrm{d}A/\mathrm{d}X$, in which we have

$$L_{k-1} = \text{either } (\hat{a}'_{m'-2}) \text{ or } \begin{pmatrix} \hat{b}'_{n'-1} & \hat{b}'_{n'-3} \\ 0 & \hat{b}'_{n'-2} \end{pmatrix}.$$

Although $|\hat{b}'_{n'-1}\hat{b}'_{n'-2}| \leq O(\delta^4)$, we have $|\hat{a}'_{m'-2}| \leq O(\delta^2)$ hence $|\hat{r}_{n'-1}| \simeq |\hat{a}'_{m'-2}| \leq O(\delta^2)$, obtaining the lower estimation in (4.9). □

Proposition 2 *Under the assumptions in* (4.1), *we have*

$$\begin{aligned} &\|S_k\|, \|T_k\| = O(\delta^0) \quad (k \geq n' - 1), \\ &\|S_k\|, \|T_k\| \leq O(\delta^{(m'-k-1)(n'-k-1)}) \quad (k < n' - 1). \end{aligned} \tag{4.11}$$

Proof The determinants expressing S_k and T_k are different from R_k only at the rightmost columns which give $O(\delta^0)$ contribution to S_k and T_k. Hence, $\|S_k\|, \|T_k\| = O(\delta^0)$ for $k \geq n' - 1$. For $k < n' - 1$, the dominant terms of S_k and T_k, or $\mathrm{lc}(S_k)$ and $\mathrm{lc}(T_k)$, are estimated by the left lower $(l-1) \times (l-1)$ submatrix of L defined above, and we obtain (4.11). □

Remark 3 What happens if the cluster contains close roots of different closenesses ? Let $0 < \delta_1 \ll \delta_2 \ll 1$, and suppose that the cluster contains m'_1 and m'_2 close roots of $A(X)$ and n'_1 and n'_2 close roots of $B(X)$, of closenesses δ_1 and δ_2, respectively, such that

$$\begin{aligned} &|\alpha_i - \gamma| = O(\delta_1) \quad (i = 1, \ldots, m'_1), \qquad |\alpha_{m'_1+j} - \gamma| = O(\delta_2) \quad (j = 1, \ldots, m'_2), \\ &|\beta_i - \gamma| = O(\delta_1) \quad (i = 1, \ldots, n'_1), \qquad |\beta_{n'_1+j} - \gamma| = O(\delta_2) \quad (j = 1, \ldots, n'_2). \end{aligned}$$

We also assume that $m'_1 \geq n'_1$ and $m'_2 \geq n'_2$. Expanding the close-root factors of $A(X)$, for example, as

$$[\prod_{i=1}^{m'_1}(X - \alpha_i)] \cdot [\prod_{j=1}^{m'_2}(X - \alpha_{m'_1+j})] = (X - \gamma)^{m'_2+m'_1} + \acute{a}_{m'_2+m'_1-1}(X - \gamma)^{m'_2+m'_1-1} + \cdots + \acute{a}_0,$$

we have

$$|\acute{a}_{m_2'+m_1'-i}| \le O(\delta_2^i) \quad (i=1,\ldots,m_2'),$$
$$|\acute{a}_{m_1'-i}| \le O(\delta_2^{m_2'}\delta_1^i) \quad (i=1,\ldots,m_1').$$

Then, we find that $\|R_k\| = O(\delta_2^0)$ for $k \ge n_1'+n_2'$ and, as k decreases from $n_1'+n_2'$ to n_1', the close-root factors of closeness δ_2 are "stripped off" from the subresultant, and we have $R_{n_1'} \propto (X-\gamma)^{n_1'}$ + lower-order terms. □

5 Clusters of close roots

If there are several clusters of close roots then we cannot apply Lemma 1 but a much more complicated analysis is necessary than that in the previous section.

We assume again that $\langle A(X), B(X)\rangle$ is regular and $A(X)$ and $B(X)$ have no common root. We consider a simple case that some roots of $A(X)$ and $B(X)$ form λ clusters of close roots, of closeness $\delta \ll 1$, such that the clusters are distant each other and the other roots of $A(X)$ and $B(X)$ are distant each other and from the clusters. Let the clusters be located at $X=\gamma_1$, $\ldots$, $X=\gamma_\lambda$, and assume that each cluster contains m' roots of $A(X)$ and n' roots of $B(X)$. Precisely, we assume that all the root-differences $|\alpha_i-\alpha_j|$'s, $|\beta_i-\beta_j|$'s and $|\alpha_i-\beta_j|$'s are of magnitude $O(\delta^0)$ except that

$$\begin{aligned} |\alpha_{(l-1)m'+i}-\gamma_l| &= O(\delta) \quad (1\le l\le\lambda;\ 1\le i\le m'),\\ |\beta_{(l-1)n'+i}-\gamma_l| &= O(\delta) \quad (1\le l\le\lambda;\ 1\le i\le n'). \end{aligned} \tag{5.1}$$

Without loss of generality, we assume that $m'\ge n'$. We redefine $C(X)$ to be

$$C(X) \stackrel{\text{def}}{=} (X-\gamma_1)\cdots(X-\gamma_\lambda). \tag{5.2}$$

We express the close-root factors of $A(X)$ and $B(X)$ as

$$\begin{aligned} \textstyle\prod_{l=1}^{\lambda}[\prod_{j=(l-1)m'+1}^{lm'}(X-\alpha_j)] &= C^{m'}+\acute{A}_{m'-1}(X)\,C^{m'-1}+\cdots+\acute{A}_0(X),\\ \textstyle\prod_{l=1}^{\lambda}[\prod_{j=(l-1)n'+1}^{ln'}(X-\beta_j)] &= C^{n'}+\acute{B}_{n'-1}(X)\,C^{n'-1}+\cdots+\acute{B}_0(X), \end{aligned} \tag{5.3}$$

where $\deg(\acute{A}_i)<\lambda$ and $\deg(\acute{B}_i)<\lambda$ $(i=0,1,2,\ldots)$. We can determine $\acute{A}_i$ as follows. Rewrite the close-root factor of $A(X)$ as $\prod_{j=1}^{m'}[\prod_{l=1}^{\lambda}(X-\gamma_l+\gamma_l-\alpha_{(l-1)m'+j})]$, expand it w.r.t. $(X-\gamma_l)$ $(1\le l\le\lambda)$, collect terms $(X-\gamma_1)^{i_1}\cdots(X-\gamma_\lambda)^{i_\lambda}$, with $i_1+\cdots+i_\lambda\ge i\lambda$, for $i=m'$ $\Rightarrow i=m'-1\Rightarrow\cdots$ successively, and convert them as $\acute{A}_i(X)C(X)^i+$(terms of degrees $<i\lambda$) by rewriting $(X-\gamma_{l'})\Rightarrow(X-\gamma_l)+(\gamma_{l'}-\gamma_l)$ if necessary. For example, for $\lambda=3$, the term $(X-\gamma_1)^3(X-\gamma_2)^3(X-\gamma_3)$ is converted as $C(X)^2(X-\gamma_1-\gamma_2+\gamma_3)+C(X)(X-\gamma_1)(X-\gamma_2)(\gamma_3-\gamma_1)(\gamma_3-\gamma_2)$. By this, we find

$$\begin{aligned} \|\acute{A}_i\| &\le O(\delta^{m'-i}) \quad (i=m'-1,\ldots,0),\\ \|\acute{B}_i\| &\le O(\delta^{n'-i}) \quad (i=n'-1,\ldots,0). \end{aligned} \tag{5.4}$$

If $B(X)=\mathrm{d}A(X)/\mathrm{d}X$ and $\gamma_l=(\alpha_{(l-1)m'+1}+\cdots+\alpha_{lm'})/m'$ $(l=1,\ldots,\lambda)$ then $\deg(\acute{A}_{m'-1})\le\lambda$–2 and $\|\acute{A}_{m'-1}\|\le O(\delta^2)$. Note that $n'=m'-1$ in this case.

Lemma 3 *We can express $A(X)$ and $B(X)$ as follows.*

$$\begin{aligned} A(X) &= \bar{A}_{m'}(X)\,C^{m'} + \bar{A}_{m'-1}(X)\,C^{m'-1} + \cdots + \bar{A}_0(X), \\ &\deg(\bar{A}_{m'}) = m-m', \quad \deg(\bar{A}_{m'-i}) < \lambda \ \ (i \geq 1), \\ B(X) &= \bar{B}_{n'}(X)\,C^{n'} + \bar{B}_{n'-1}(X)\,C^{n'-1} + \cdots + \bar{B}_0(X), \\ &\deg(\bar{B}_{n'}) = n-n', \quad \deg(\bar{B}_{n'-i}) < \lambda \ \ (i \geq 1), \end{aligned} \tag{5.5}$$

where

$$\begin{aligned} \|\bar{A}_i\| &\leq O(\delta^{m'-i}), \quad (i = m'-1, \ldots, 0), \\ \|\bar{B}_i\| &\leq O(\delta^{n'-i}), \quad (i = n'-1, \ldots, 0). \end{aligned} \tag{5.6}$$

If $B(X) = \mathrm{d}A(X)/\mathrm{d}X$ and $\gamma_l = (\alpha_{(l-1)m'+1} + \cdots + \alpha_{lm'})/m'$ $(l = 1, \ldots, \lambda)$ then

$$\begin{aligned} \|\bar{A}_i\| &\leq O(\delta^{\max\{2,\, m'-i\}}), \quad (i = m'-1, \ldots, 0), \\ \|\bar{B}_i\| &\leq O(\delta^{\max\{2,\, n'-i\}}), \quad (i = n'-1, \ldots, 0). \end{aligned} \tag{5.7}$$

Proof In the case of $B(X) \neq \mathrm{d}A(X)/\mathrm{d}X$, put $\bar{Q}_A = a_m \prod_{i=\lambda m'+1}^{m}(X-\alpha_i)$ and $\bar{Q}_B = b_n \prod_{i=\lambda n'+1}^{n}(X-\beta_i)$. We compute $\bar{A}_0$ as $\bar{A}_0 = \mathrm{rem}(\bar{Q}_A \acute{A}_0, C)$, and set Q as $Q := \mathrm{quo}(\bar{Q}_A \acute{A}_0, C)$. Then, we compute $\bar{A}_j$ $(j = 1, 2, \ldots)$ successively as $\bar{A}_j = \mathrm{rem}(\bar{Q}_A \acute{A}_j + Q, C)$ and reset $Q := \mathrm{quo}(\bar{Q}_A \acute{A}_j + Q, C)$. We compute $\bar{B}_0, \bar{B}_1, \ldots$ similarly, obtaining (5.5). Then, (5.4) leads us to (5.6).

If $B(X) = \mathrm{d}A(X)/\mathrm{d}X$, put $\bar{Q}_A$ as above and set $\bar{Q}_B$ as $\bar{Q}_B = m'[\mathrm{d}C/\mathrm{d}X]\,\bar{Q}_A + C\,[\mathrm{d}\bar{Q}_A/\mathrm{d}X]$. Then, we obtain (5.5) again. Setting $\gamma_l = (\alpha_{(l-1)m'+1} + \cdots + \alpha_{lm'})/m'$ for each $l \in \{1, \ldots, \lambda\}$, we have $\|\bar{A}_{m'-1}\| \leq O(\delta^2)$. Differentiating $A(X)$ in (5.5), we obtain (5.7). □

We want to show that the coefficients of $R_k(A, B)$ can be order estimated by the determinants the elements of which are coefficients of $\bar{A}_i(X)$ and $\bar{B}_i(X)$ $(i = 0, 1, \ldots)$, as Lemma 4 claims. We explain this by an example.

Example Let $m' = n' = 2$ and let A and B be expressed as

$$\begin{aligned} A &= A'C + D', \quad A' = A''C + D'', \quad \|A''\| = O(\delta^0), \\ B &= B'C + E', \quad B' = B''C + E'', \quad \|B''\| = O(\delta^0), \end{aligned}$$

where D', E', D'', E'' are polynomials of degrees $< \deg(C)$, and of magnitudes $\|D''\|, \|E''\| = O(\delta^1)$ and $\|D'\|, \|E'\| = O(\delta^2)$. We note that A and B are expressed in a nested form; we say that A and B are of *nesting level* 0, A', B', D', E' are of *nesting level* 1, and A'', B'', D'', E'' are of *nesting level* 2. Furthermore, we consider polynomials of definite degrees such that $C = X^2 + c_1 X + c_0$, $D' = d'_1 X + d'_0$, $E' = e'_1 X + e'_0$, $A'' = a''_1 X + a''_0$, $B'' = b''_2 X^2 + b''_1 X + b''_0$, $D'' = d''_1 X + d''_0$, $E'' = e''_1 X + e''_0$. By a_i, b_i, a'_i, b'_i we denote the coefficients of terms of degree i of A, B, A', B', respectively.

The $\mathrm{lc}(R_k(A, B))$ in our case is a determinant with elements $a_5, \ldots, a_0$ and $b_6, \ldots, b_0$. We apply the transformation given in the proof of Theorem 2 to the left 6 columns of the determinant, then to the left 4 columns of the resulting determinant. By these, the determinant

becomes (we use the symbols $\breve{*}$, with $* = d_i, e_i$, etc., as defined in (2.10))

$$\mathrm{lc}(R_k(A,B)) = \begin{vmatrix} a_1'' & a_0'' & \breve{d}_1''' & \breve{d}_0''' & \breve{d}_1'' & \breve{d}_0'' & & & \\ & a_1'' & a_0'' & \breve{d}_1''' & a_0' & \breve{d}_1' & a_0 & & \\ & & a_1'' & a_0'' & a_1' & a_0' & a_1 & a_0 & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ b_2'' & b_1'' & b_0'' & \breve{e}_1''' & b_0' & \breve{e}_1' & b_0 & & \\ & b_2'' & b_1'' & b_0'' & b_1' & b_0' & b_1 & b_0 & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{vmatrix} \tag{5.8}$$

$$= a_1'' R_k^{(1)} - a_0'' R_k^{(2)} + \breve{d}_1''' R_k^{(3)} - \breve{d}_0''' R_k^{(4)} + \breve{d}_1'' R_k^{(5)} - \breve{d}_0'' R_k^{(6)},$$

where the last expression is obtained by expanding the determinant w.r.t. the first row. The determinant in (5.8) has the following three properties.

Property P1 Each element of the top row has been reduced maximally (i.e., reduced to a number of the same magnitude as the corresponding coefficient of $\bar{A}_i(X)$).

Property P2 The columns are ordered so as to preserve the order of the corresponding columns of $\mathrm{lc}(R_k(A,B))$.

Property P3 All the elements of left-hand-side columns, middle columns and other right-hand-side columns are coefficients (or linear combinations of them) of polynomials of nesting level 2, 1 and 0, respectively.

We show that $R_k^{(1)}, \ldots, R_k^{(6)}$ can be transformed into determinants having the properties P1, P2 and P3.

We transform $R_k^{(1)}$, as follows. Replacing the 6th column ${}^t(a_0, a_1, \cdots, b_0, b_1, \cdots)$ of $R_k^{(1)}$ by ${}^t(c_0 a_0' + d_0'',\ c_0 a_1' + c_1 a_0' + d_1'',\ \cdots,\ c_0 b_0' + e_0',\ c_0 b_1' + c_1 b_0' + e_1',\ \cdots)$, and using the 4th and 5th columns, we can transform the 6th column into ${}^t(\breve{d}_0'', \breve{d}_1'', \cdots, \breve{e}_0', \breve{e}_1', \cdots)$, because $d_1'' = \breve{d}_1''$, $e_1' = \breve{e}_1'$ and $d_0'' - c_1 \breve{d}_1'' = \breve{d}_0''$. Similarly, we can transform the 4th column ${}^t(a_0', a_1', \cdots, b_0', b_1', \cdots)$ into ${}^t(\breve{d}_0''', \breve{d}_1''', \cdots, \breve{e}_0'', \breve{e}_1'', \cdots)$. By these, $R_k^{(1)}$ is transformed as follows.

$$R_k^{(1)} \Rightarrow \begin{vmatrix} a_1'' & a_0'' & \breve{d}_1''' & \breve{d}_0''' & \breve{d}_1'' & \breve{d}_0'' & & \\ & a_1'' & a_0'' & \breve{d}_1''' & a_0' & \breve{d}_1'' & a_0 & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ b_1'' & b_0'' & \breve{e}_1''' & \breve{e}_0''' & \breve{e}_1'' & \breve{e}_0'' & & \\ b_2'' & b_1'' & b_0'' & \breve{e}_1''' & b_0' & \breve{e}_1'' & b_0 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{vmatrix} \tag{5.9}$$

This determinant has the properties P1, P2 and P3. We can transform $R_k^{(2)}$ similarly.

In order to transform $R_k^{(i)}$, $i \geq 3$, we need another technique. Consider, for example, $R_k^{(3)}$.

Removing the top B-row, we have

$$\frac{R_k^{(3)}}{\pm b_2''} = \begin{vmatrix} a_1'' & \breve{d}_1''' & a_0' & \breve{d}_1' & a_0 & & \\ & a_0'' & a_1' & a_0' & a_1 & a_0 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ b_2'' & b_0'' & b_1' & b_0' & b_1 & b_0 & \\ & b_1'' & b_2' & b_1' & b_2 & b_1 & b_0 \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{vmatrix}. \tag{5.10}$$

Transforming the 5th column as above, we express the 3rd column by the sum of columns as ${}^t(a_0', a_1', \cdots, b_1', b_2', \cdots) = c_0\,{}^t(a_0'', a_1'', \cdots, b_1'', b_2'', \cdots) + c_1\,{}^t(\breve{d}_1''', a_0'', \cdots, b_0'', b_1'', \cdots)$ + ${}^t(\breve{d}_0''', \breve{d}_1''', \cdots, e_1'', b_0'', \cdots)$, because $d_1'' = \breve{d}_1'''$ and $d_0'' - c_1\breve{d}_1''' = \breve{d}_0'''$. Then, the determinant becomes the sum of three determinants, among which the second one is 0 because it has two same columns. Hence, $R_k^{(3)}/(\pm b_2'')$ is transformed into the sum of two determinants:

$$-c_0 \begin{vmatrix} a_1'' & a_0'' & \breve{d}_1''' & \breve{d}_1' & \breve{d}_0''' & & \\ & a_1'' & a_0'' & a_0' & \breve{d}_1' & a_0 & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{vmatrix} + \begin{vmatrix} a_1'' & \breve{d}_1''' & \breve{d}_0''' & \breve{d}_1' & \breve{d}_0''' & & \\ & a_0'' & \breve{d}_1''' & a_0' & \breve{d}_1' & a_0 & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{vmatrix}.$$

These determinants have the properties P1, P2 and P3. □

Lemma 4 *The magnitude of* $\mathrm{lc}(R_k(A,B))$ *w.r.t.* δ *can be estimated by* $\mathrm{lc}(R_k(\bar{A},\bar{B}))$, *where* $\bar{A}(X)$ *and* $\bar{B}(X)$ *are the following polynomials.*

$$\begin{aligned} \bar{A}(X) &= \bar{A}_{m'}X^{m'\lambda} + \bar{A}_{m'-1}X^{(m'-1)\lambda} + \cdots + \bar{A}_0, \\ \bar{B}(X) &= \bar{B}_{n'}X^{n'\lambda} + \bar{B}_{n'-1}X^{(n'-1)\lambda} + \cdots + \bar{B}_0. \end{aligned} \tag{5.11}$$

Proof We will prove the lemma by generalizing property P3 accordingly, by transforming the subresultant into such a determinant as in (5.8), and expanding it successively to minors having properties P1, P2 and P3. We call a column to be reduced, such as the 5th or the 3rd one in (5.10), a *reducend*. We call λ successive columns to be used to reduce a reducend, such as the 3rd and the 4th ones in (5.10), *reducers*. We call a column which is obtained by reducing a reducend, such as the 6th or 4th one in (5.9), *reductant*. Then, the nested structure tells us that

$$\text{reducend} = c_0\,[\text{1st reducer}] + \cdots + c_{\lambda-1}\,[\lambda\text{th reducer}] + \text{reductant}. \tag{5.12}$$

We assume, as an induction assumption, that if the reducend is of nesting level ℓ then the reducers and the reductant are of nesting level $\ell+1$. We consider a determinant which has the properties P1, P2 and P3. We expand it w.r.t. the first row, obtaining minors. These minors do not have the property P1, because of the existence of reducends. We transform the minors into determinants having properties P1, P2 and P3, as in the above example.

Let R be a minor obtained by the determinant as above. If R contains all the necessary reducers, as $R_k^{(}1)$ and $R_k^{(}2)$ in the above example, we can replace each reducend in R by the corresponding reductant and the resulting minor has the properties P1, P2 and P3. However, many minors lack necessary reducers. If R lacks some reducers, we replace the reducend

by the right-hand-side expression in (5.12) and express R by the sum of $\lambda+1$ determinants $R^{(1)},\ldots,R^{(\lambda+1)}$, as we have done for $R_k^{(3)}$ in the avove example. If the jth reducer is contained in R then $R^{(j)}=0$, otherwise $R^{(j)}$ is a determinant obtained from R by replacing the reducend by the jth reducer. In the latter case, we exchange columns of $R^{(j)}$ if necessary, so as to satisfy property P2. The $R^{(\lambda+1)}$ is a determinant obtained from R by replacing the reducend by the reductant. Repeating this process for all the reducends, we obtain minors which have the properties P1 and P2. The recursive structure of R tells us that this reduction can be continued to prove the lemma. □

Proposition 3 *Let $k=q\lambda+r$ with $\lambda>r\geq 0$. Under the assumptions in* (5.1), *we have*

$$\begin{aligned}&\|R_k(A,B)\|=O(\delta^0)\quad(k\geq\lambda n'),\\&\|R_k(A,B)\|=O(\delta^{\lambda(m'-q)(n'-q)-(m'+n'-2q-1)r})\quad(k<\lambda n').\end{aligned}\tag{5.13}$$

Let $R_k(A,B)$, with $k\leq\lambda n'$, be expressed as

$$\begin{aligned}&R_k(A,B)=\bar{R}_q(X)\,C^q+\bar{R}_{q-1}(X)\,C^{q-1}+\cdots+\bar{R}_0(X),\\&\deg(\bar{R}_q)=r,\quad\deg(\bar{R}_i)<\lambda\quad(i=q-1,\ldots,0).\end{aligned}\tag{5.14}$$

Then, we have (note that $n'=m'-1$ if $B=\mathrm{d}A/\mathrm{d}X$)

$$\begin{aligned}&\|\bar{R}_{q-i}\|\leq O(\delta^{\lambda(m'-q)(n'-q)-(m'+n'-2q-1)r+i})\quad(0<i\leq q),\\&\|\bar{R}_{q-1}\|\leq O(\delta^2)\quad\textit{if } k=\lambda n' \textit{ and } B=\mathrm{d}A/\mathrm{d}X.\end{aligned}\tag{5.15}$$

Proof Formula (2.12) in Theorem 2 can be rewritten as

$$\begin{aligned}&R_k(AC+D,BC+E)=\breve{R}_k^{(1)}C+\breve{R}_k^{(0)},\\&\breve{R}_k^{(1)}=\breve{R}_k(A,B)+\mathrm{quo}(\breve{R}_k(D,E),C),\\&\breve{R}_k^{(0)}=\mathrm{rem}(\breve{R}_k(D,E),C),\quad\deg(\breve{R}_k^{(0)})<\deg(C).\end{aligned}$$

Applying Theorem 2 to $A(X)$ and $B(X)$ in (5.5) repeatedly, we see that $R_k(A,B)$ can be expressed as in (5.14). For $k\leq\lambda n'$, the coefficients of $R_k(A,B)$ decrease as those of terms of degrees $\leq\lambda n'$, of $B(X)$. Hence, we have only to estimate the magnitude of $\mathrm{lc}(R_k(A,B))$.

For convenience, we decompose $A(X)$ and $B(X)$ as

$$\begin{aligned}&A(X)=\bar{A}_{m'}C^{m'}+\Delta_A,\quad\Delta_A=\bar{A}_{m'-1}C^{m'-1}+\cdots+\bar{A}_0,\\&B(X)=\bar{B}_{n'}C^{n'}+\Delta_B,\quad\Delta_B=\bar{B}_{n'-1}C^{n'-1}+\cdots+\bar{b}_0.\end{aligned}$$

For $k\geq\lambda n'$, we have $R_k(A,B)\simeq R_k(\bar{A}_{m'}C^{m'},\bar{B}_{n'}C^{n'})=C^{n'}R_{k-\lambda n'}(\bar{A}_{m'}C^{m'-n'},\bar{B}_{n'})$. Hence, $\|R_k(A,B)\|=O(\delta^0)$ because $\bar{A}_{m'}C^{m'-n'}$ and $\bar{B}_{n'}$ have no mutually close root.

For $k<\lambda n'$, we have $R_{k-\lambda n'}(\bar{A}_{m'}C^{m'-n'},\bar{B}_{n'})=0$ because the corresponding determinant contains 0-column(s). We estimate $\mathrm{lc}(R_k)$ by replacing rows of the determinant for $\mathrm{lc}(R_k(\bar{A}_{m'}C^{m'},\bar{B}_{n'}C^{n'}))$ by the corresponding Δ_A-rows and/or Δ_B-rows. Replacing lower rows gives lower order terms w.r.t. δ (i.e., larger magnitude terms), hence we replace either lower $\bar{A}_{m'}C^{m'}$-rows or lower $\bar{B}_{n'}C^{n'}$-rows.

We first consider the case of $m'=n'$ (hence $B\neq\mathrm{d}A/\mathrm{d}X$). The determinant for $\mathrm{lc}(R_{k-\lambda n'}(\bar{A}_{m'}C^{m'-n'},\bar{B}_{n'}))$ contains $m+n-2k$ columns among which the right $\lambda n'-k$ ones are 0-columns. Therefore, we replace the lower $(\lambda n'-k)$ $\bar{B}_{n'}C^{n'}$-rows by the corresponding Δ_B-rows.

We construct the kth subresultant for $\bar{A}$ and $\bar{B}$, and let L be the right lower $(\lambda n'-k)\times(\lambda n'-k)$ submatrix of the determinant. Lemma 4 tells us that $|\mathrm{lc}(R_k(A,B))| = O(|L|)$. We show the magnitudes of elements of L for $\lambda = 2$:

$$L = \left(\begin{array}{cc|cc|cc} \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \hline O(\delta^2) & O(\delta^3) & O(\delta^3) & O(\delta^4) & O(\delta^4) & \ddots \\ O(\delta^2) & O(\delta^2) & O(\delta^3) & O(\delta^3) & O(\delta^4) & \ddots \\ \hline O(\delta^1) & O(\delta^2) & O(\delta^2) & O(\delta^3) & O(\delta^3) & \ddots \\ O(\delta^1) & O(\delta^1) & O(\delta^2) & O(\delta^2) & O(\delta^3) & \ddots \end{array}\right),$$

where an upper right triangular part may be null. By dividing L into cells of size $\lambda\times\lambda$, as illustrated above, we see that the matrix is of a similar form as those in the previous section. If $k = q\lambda$ then L contains $(n'-q)^2$ cells and we can estimate $|L|$ by any combination of the cells. Since upper right cells may be null, we estimate $|L|$ by the product of diagonal cells, obtaining $|L| = O(\delta^{\lambda(n'-q)(n'-q)})$. If $k = q\lambda+r$, $0<r<\lambda$ hence $\lambda n'-k = \lambda(n'-q)-r$, we remove upper r rows and right r columns from L and estimate $|L|$ by using the leftmost lowest cell, as illustrated in Figure 1 for the case of $\lambda = 3$ and $n'-q = 2$. Hence, we obtain $|L| = O(\delta^{\lambda(n'-q)^2-(2n'-2q-1)r})$.

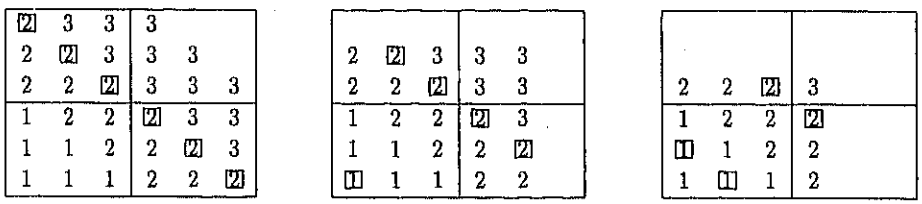


Figure 1. Order estimation of $|L|$ by the boxed elements.
Case of $n'-q = 2$, and $r=0$ (left), $r=1$ (middle), $r=2$ (right).
The element ℓ denotes a number of $O(\delta^\ell)$.

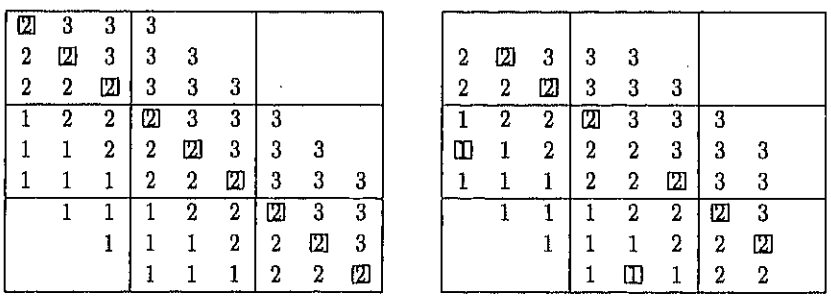


Figure 2. Order estimation of $|L|$ by the boxed elements.
Case of $m'-n' = 1$, $m'-q = 3$, and $r=0$ (left), $r=1$ (right).

We next consider the case of $m' > n'$. In this case, we replace the lower $(\lambda n'-k)$ $\bar{A}_{m'}C^{m'}$-rows by the corresponding Δ_A-rows. Then, the matrix L is now of the form (we show the case

of $\lambda = 2$; the form is valid even if $B = \mathrm{d}A/\mathrm{d}X$).

$$L = \left(\begin{array}{cc|cc} \ddots & \ddots & \ddots & \ddots \\ \hline O(\delta^{m'-n'+1}) & O(\delta^{m'-n'+2}) & O(\delta^{m'-n'+2}) & \ddots \\ O(\delta^{m'-n'+1}) & O(\delta^{m'-n'+1}) & O(\delta^{m'-n'+2}) & \ddots \end{array}\right)$$

If $k = q\lambda$ then the diagonal elements of the diagonal cells are of magnitude $\delta^{m'-q}$. Hence, estimating $|L|$ as for the case of $m' = n'$, we obtain (5.15). (If we replace $\tilde{B}_{n'}C^{n'}$-rows, we must replace $\lambda m' - k$ rows. Thus, we obtain a matrix L of size $(\lambda m' - k) \times (\lambda m' - k)$, with a left lower triangular part null (see, Figure 2). We can estimate $|L|$ similarly as above, with some complication illustrated by Figure 2, and we obtain (5.15) again.) □

Proposition 4 *Under the assumption in* (5.1), *we have*

$$\begin{aligned} &\|S_k\|, \|T_k\| = O(\delta^0) \quad (k \geq \lambda n' - 1), \\ &\|S_k\|, \|T_k\| = O(\|R_{k+1}\|) \quad (k < \lambda n' - 1). \end{aligned} \tag{5.16}$$

Remark 4 Propositions 1 and 3 tell us that $(X-\gamma)^{n'}$ or $C(X)^{n'}$ is an approximate common divisor of $A(X)$ and $B(X)$, of tolerance $O(\delta)$. In particular, if $B = \mathrm{d}A/\mathrm{d}X$ then it is an approximate common divisor of tolerance $O(\delta^2)$. Therefore, among various algorithms for univariate square-free decomposition, the naive algorithm which computes approximate common divisors successively by the Euclidean algorithm, is quite stable. □

6 On cancellation in Euclidean algorithm

If polynomials A, B, E are such that $\|A\| \simeq \|B\| \gg \|E\|$ and $A - B = E$ then we say that the cancellation of amount of $\|A\|/\|E\|$ occurred in the computation of E. With the results in the previous sections, we are able to estimate the magnitudes of cancellations which occur in the Euclidean algorithm. For examples of cancellations, see [SS89].

The remainder sequence $(P_1 = A, P_2 = B, \cdots, P_k, \cdots)$ and its cofactor sequences $(U_1 = 1, U_2 = 0, \cdots, U_k, \cdots)$, $(V_1 = 0, V_2 = 1, \cdots, V_k, \cdots)$ are computed by the formulas

$$\left.\begin{aligned} P_{k+1} &:= c_k P_{k-1} - Q_k P_k \\ U_{k+1} &:= c_k U_{k-1} - Q_k U_k \\ V_{k+1} &:= c_k V_{k-1} - Q_k V_k \end{aligned}\right\} \quad (k = 2, 3, \ldots), \tag{6.1}$$

where $c_k \in \mathbf{C}$ and $Q_k \in \mathbf{C}[X]$. We normalize the cofactors as

$$\max\{\|U_k\|, \|V_k\|\} = 1 \quad (k = 3, 4, \ldots). \tag{6.2}$$

In order to satisfy (6.2), we set c_k and Q_k as

$$\max\{|c_k|, \|Q_k\|\} = O(1). \tag{6.3}$$

Then, the cancellation in the division of P_{k-1} by P_k occurs when $\|c_k P_{k-1}\| \approx \|Q_k P_k\|$ and $\|P_{k+1}\| \ll \|P_{k-1}\|$, and the amount of the cancellation is about $\|c_k P_{k-1}\|/\|P_{k+1}\|$. Thus, the total amount of the cancellation in the computation of P_k from A and B is given by $1/\|P_k\|$, or

$$\text{total-cancellation} = \max\{\|S_k\|, \|T_k\|\}/\|R_k\|. \tag{6.4}$$

Table I shows the magnitudes of cancellations when $A(X)$ and $B(X)$ have only one cluster of close roots of closeness δ, where $\nu = m' - n'$.

k	$\|R_k\|$	$\|S_k\|$	$\|S_k\|/\|R_k\|$
n'	$O(\delta^0)$	$O(\delta^0)$	$O((1/\delta)^0)$
$n'-1$	$O(\delta^{1(\nu+1)})$	$O(\delta^0)$	$O((1/\delta)^{\nu+1})$
$n'-2$	$O(\delta^{2(\nu+2)})$	$O(\delta^{1(\nu+1)})$	$O((1/\delta)^{\nu+3})$
$n'-3$	$O(\delta^{3(\nu+3)})$	$O(\delta^{2(\nu+2)})$	$O((1/\delta)^{\nu+5})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table I. Magnitude of the total cancellation ($\|S_k\|/\|R_k\|$).
Case of single cluster of close roots ($\nu = m' - n'$).

References

[BT71] W. S. Brown and J. F. Traub: On Euclid's algorithm and the theory of subresultants. J. ACM 18 (1971), 505-514.

[CJ96] B. Caviness and J. Johnson: *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Springer Verlag, 1996. Texts and Monographs in Symbolic Computation.

[Col67] J. E. Collins: Subresultants and the reduced polynomial remainder sequences. J. ACM 14 (1967), 128-142.

[Col75] J. E. Collins: Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. in *Lecture Notes in Computer Science*, Vol. 33, 134-183. Springer-Verlag, Berlin, 1975.

[GG99] J. von zur Gathen and J. Gerhard: *Modern Computer Algebra.* Cambridge University Press, 1999.

[Hoo01A] Hoon Hong: Ore principal subresultant coefficients in solutions. J. Applicable Algebra in Eng., Commun., and Comput. **11** (2001), 227-237.

[Hoo01B] Hoon Hong: Ore subresultant coefficients in solutions. J. Applicable Algebra in Eng., Commun., and Comput. **12** (2001), 421-428.

[Hoo02] Hoon Hong: Subresultants in roots. in *Abstracts of 8th Intern'l Conference on Applications of Computer Algebra*, 27-28. June 2002, Univ. of Thessarly, Volos, Greece.

[HS97] V. Hribernig and H. J. Stetter: Detection and validation of clusters of zeros of polynomials. J. Symbolic Comput. **24** (1997), 667-681.

[LP01] A. Lascoux and P. Pragacz: Double sylvester sums for Euclidean division, multi-Shur functions, and gysin maps for grassmann bundles. 2001 (submitted).

[SN89] T. Sasaki and M-T. Noda: Approximate square-free decomposition and root-finding of ill-conditioned algebraic equations. J. Inf. Process., **12** (1989), 159-168.

[SS89] T. Sasaki and M. Sasaki: Analysis of accuracy decreasing in polynomial remainder sequence with floating-point number coefficients. J. Inform. Proces. **12** (1989), 394-403.

[SS97] T. Sasaki and M. Sasaki: Polynomial remainder sequence and approximate GCD. SIGSAM Bulletin **31** (1997), 4-10.

[Syl1853] J. J. Sylvester: On a theory of syzygetic relations of two rational integral functions, comprising an application to the theory of Sturm's function and that of the greatest algebraic common measure. Trans. Roy. Soc. London, 1853. Reprinted in *The Collected Mathematical Papers of James Joseph Sylvester*, Chelsea Publ., New York 1973, Vol. 1, 429-586.

Certification of Analytic Continuation of Algebraic Function

Tateaki Sasaki [†] and Daiju Inaba[‡]

[†] Institute of Mathematics, [‡] Doctoral Program in Mathematics
University of Tsukuba, Tsukuba-shi, Ibaraki 305, Japan
{sasaki,inaba}@math.tsukuba.ac.jp

Abstract

This paper investigates the analytic continuation of algebraic functions being defined as roots of a bivariate polynomial, focusing the attention on certification of the continuation. The analytic continuation in this case is nothing but to determine the one-to-one correspondence among the roots expanded into power series at different points. We propose three methods for the continuation of Taylor-series roots. The first one is based on Smith's theorem, and the second one is based on an upper bound for the "smallest root" of a univariate polynomial, derived in this paper. These methods determine the correspondence among the constant terms of the Taylor series. The third method utilizes leading several terms of the Taylor series. We also propose a method for the continuation of Puiseux-series roots which may appear at singular points. We analyze these methods both theoretically and experimentally.

Key words: algebraic function, analytic continuation, certification, power series root, Puiseux series root, Taylor series root.

1 Introduction

The analytic continuation is a very important operation in mathematics; it plays an important role for the determination of Riemann surface of algebraic function, the monodromy group of the solutions of ordinary differential equation, and so on. Recently, the analytic continuation of algebraic function becomes an important tool for the approximate factorization of multivariate polynomial. About ten years ago, Sasaki et al. proposed an algorithm for approximate factorization [SSKS91, SSH92]. The algorithm constructs approximate factors by combining (truncated) power-series roots, but the algorithm often falls down due to explosion of numerical errors. Recently, Sasaki presented an effective and simple method of combining power-series roots, and proposed to utilize power-series roots expanded at several points [Sas01], requiring to determine the correspondence between the roots expanded at different points. Corless et al. proposed a numerical algorithm for the approximate factorization [CG..01]. The algorithm evaluates an algebraic function being defined implicitly at many points, and it interpolates an approximate factor by the values. Similarly, Galligo et al. proposed algorithms for absolute factorization of multivariate polynomials [GW97, GR01]. Their algorithms determine polynomial

factors by computing power-series roots at several points near a singular point. All of these algorithms rely on the analytic continuation in some sense.

In this paper, we investigate the analytic continuation of algebraic functions being defined as roots of a bivariate polynomial, focusing the attention on certification of the continuation. Let $F(x,u) \in \mathbf{C}[x,u]$ be a given bivariate polynomial of degree n, and let $\bar{\varphi}_i(u)$ $(i = 1,\ldots,n)$ be the roots of $F(x,u)$ w.r.t. x :

$$\begin{aligned} F(x,u) &= f_n(u)x^n + f_{n-1}x^{n-1} + \cdots + f_0(u) \\ &= f_n(u)\,(x - \bar{\varphi}_1(u)) \cdots (x - \bar{\varphi}_n(u)). \end{aligned} \tag{1.1}$$

We assume that $F(x,u)$ is square-free (i.e., has no duplicated factor), hence $\bar{\varphi}_i(u) \neq \bar{\varphi}_j(u)$ $(\forall i \neq j)$. We say that a point $u = s$, $s \in \mathbf{C}$, is a singular point if $f_n(s) = 0$ and/or $F(x,s)$ is not square-free. (The singular points in the sense of algebraic geometry are included in the singular points defined here). Let $u = a$, $a \in \mathbf{C}$, be a non-singular point, then the root $\bar{\varphi}_i(u)$ can be expanded into Taylor series at $u = a$. At a singular point $u = s$, some roots may be expanded into Puiseux series (fractional-power series) or may become infinite. We call the roots expanded into Taylor series and Puiseux series *Taylor-series roots* and *Puiseux-series roots*, respectively, and call them *power-series roots* in both cases. Below, for simplicity, we describe "point $u = a$" and "expanded at $u = a$" as "point a" and "expanded at a", respectively.

Let a and b be non-singular points (we may have $a = b$, because the continuation is made along a path connecting a and b), and let the root $\bar{\varphi}_i(u)$ be expanded into power series at a and b and truncated at the kth power, as follows.

$$\left.\begin{aligned} \varphi_i^{(a)}(u;k) &= f_{i,0}^{(a)} + f_{i,1}^{(a)}(u-a) + \cdots + f_{i,k}^{(a)}(u-a)^k, \\ \varphi_i^{(b)}(u;k) &= f_{i,0}^{(b)} + f_{i,1}^{(b)}(u-b) + \cdots + f_{i,k}^{(b)}(u-b)^k, \end{aligned}\right\} \quad (i = 1,\ldots,n). \tag{1.2}$$

Since $\bar{\varphi}_i(u) = \varphi_i^{(a)}(u;\infty) = \varphi_i^{(b)}(u;\infty)$, there is a one-to-one correspondence between each element of $\{\varphi_1^{(a)},\cdots,\varphi_n^{(a)}\}$ and some element of $\{\varphi_1^{(b)},\cdots,\varphi_n^{(b)}\}$, and the analytic continuation in our case is nothing but to determine the correspondence by using truncated power-series roots.

The textbook method of analytic continuation, i.e., restructuring of an infinite power series, is not suited for computer. Because of the importance of analytic continuation, we should search for better methods of computational analytic continuation. For the holonomic functions, i.e., the solutions of some kind of ordinary differential equations, Chudnovsky and Chudnovsky [CC90] and van der Hoeven [vdH99] discussed the analytic continuation. These authors compute the power-series solutions to a high power, and estimate an upper bound of the Taylor remainder by using the behavior of $|f_{i,k}|/|f_{i,0}|$ at $k \to \infty$. As for the power-series roots of polynomial in (1.1), Shiihara and Sasaki [SS96] and Doconinck and van Hoeij [DH99] discussed the analytic continuation so as to determine the Riemann surface of the algebraic function. In [SS96], the authors connect the Puiseux-series roots with the Taylor-series roots and certify the continuation by using Smith's theorem [Smi70]. Their method is quite simple but not applicable if both Puiseux-series and Taylor-series roots coexist at the expansion point, as we will explain in Section 5. Furthermore, the certification of continuation is not complete. In [DH99], the authors investigate the analytic continuation around singular points, but they do not consider the certification. Furthermore, Corless et al. [CG..01] investigated tracing an algebraic function along a path by using a technique of solving a partial differential equation.

In Section 2, we present the first method of certified analytic continuation which utilizes Smith's theorem. In Section 3, we derive a theorem for the upper bound for the "smallest" root of a univariate polynomial, and present the second method of certified analytic continuation based on the theorem. These two methods treat only the constant terms of Taylor-series roots. In Section 4, we describe the third method which treats leading several terms of Taylor-series roots. In Section 5, we test the three methods proposed by polynomials of degrees 10, 20 and 50 generated randomly. In section 6, we consider to connect the Taylor-series roots and the Puiseux-series roots expanded at singular points.

2 A method based on Smith's theorem (method S)

First of all, we make an important restriction on the description of this paper. In most cases of actual computation, the continuation is made along a path connecting given points a and b. The path is usually not short, then we divide the path into short sub-paths and repeat continuations so that, at each continuation, the roots at one edge of a sub-path are connected with those at another edge directly. In this paper, for clarity, we describe only **the continuation along a sub-path.** Thus, we consider a and b to be the edge points of a sub-path, and we assume that

$$0 \neq |a-b| \ll 1. \tag{2.1}$$

Let the roots of $F(x,a)$ and $F(x,b)$ be $\alpha_1,\ldots,\alpha_n$ and $\beta_1,\ldots,\beta_n$, respectively.

$$\begin{aligned} F(x,a) &= f_n(a)\,(x-\alpha_1)\cdots(x-\alpha_n), \quad \alpha_i \neq \alpha_j \quad (\forall i \neq j), \\ F(x,b) &= f_n(b)\,(x-\beta_1)\cdots(x-\beta_n), \quad \beta_i \neq \beta_j \quad (\forall i \neq j). \end{aligned} \tag{2 2}$$

The power-series roots are uniquely determined by the leading terms (constant terms), if the expansion point is non-singular. Hence, we can determine the correspondence between the truncated power series $\varphi_i^{(a)}(u;k)$ and $\varphi_j^{(b)}(u;k)$ by determining the correspondence between the numbers α_i and β_j.

We can determine the correspondence between α_i and β_j by using Smith's theorem which is famous in numerical analysis.

Theorem 1 (Smith 1970) *Let $G(x)$ be a monic univariate polynomial in $\mathbf{C}[x]$, and let $c_1,\ldots,c_n$ be n distinct numbers in $\mathbf{C}$. Let n numbers $r_1,\ldots,r_n$ be defined as follows.*

$$r_i = \frac{n\,|G(c_i)|}{|\prod_{j=1,\neq i}^{n}(c_i-c_j)|} \quad (i=1,\ldots,n). \tag{2.3}$$

Let $D_1,\ldots,D_n$ be n discs in the complex plane, such that $\mathrm{center}(D_i) = c_i$ *and* $\mathrm{radius}(D_i) = r_i$ *$(i=1,\ldots,n)$. Then, the union $D_1 \cup \cdots \cup D_n$ contains all the roots of $G(x)$. Furthermore, if a union $D_1 \cup \cdots \cup D_m$ is simply connected and does not intersect with $D_{m+1},\ldots,D_n$, then the number of roots contained in this union is m.*

Corollary 1 *Let a polynomial $\tilde{G}(x,\lambda)$ be*

$$\tilde{G}(x,\lambda) = F(x,\, a+\lambda(b-a)), \tag{2.4}$$

where λ is a parameter such that $0 \leq |\lambda| \leq 1$. Let $\tilde{D}_i(\lambda)$ $(i = 1, \ldots, n)$ be Smith's discs for $\tilde{G}(x, \lambda)$ with $c_i = \alpha_i$. If $\tilde{D}_1(\lambda), \ldots, \tilde{D}_n(\lambda)$ do not overlap one another for any λ on a path connecting 0 and 1, then we have the correspondence $\alpha_i \longleftrightarrow \beta_i$ $(i = 1, \ldots, n)$, where β_i is the root closest to α_i.

Proof. Since the roots of the univariate polynomial are continuous functions in its coefficients, the roots of $\tilde{G}(x, 0)$ change continuously to the roots of $\tilde{G}(x, 1)$ if we change λ from 0 to 1 continuously. On the other hand, the assumption tells us that, for any value of λ, $0 \leq |\lambda| \leq 1$, each disc $\tilde{D}_i(\lambda)$ contains only one root of $\tilde{G}(x, \lambda)$. The roots of $\tilde{G}(x, \lambda)$ at $\lambda = 0$ are α_i $(i = 1, \ldots, n)$ and those at $\lambda = 1$ are β_j $(j = 1, \ldots, n)$. Hence, there must be one-to-one correspondence between each root of $F(x, a)$ and the closest one among the roots of $F(x, b)$. □

In order to use the above corollary in actual continuation, we have to devise a simple method for checking the condition that $\tilde{D}_1(\lambda), \ldots, \tilde{D}_n(\lambda)$ do not overlap for any value of λ on the path. First, we choose the path connecting 0 and 1 to be a line, hence λ is a real parameter such that $0 \leq \lambda \leq 1$. Next, we replace the above condition by the following sufficient condition, so that we can treat each disc separately.

Condition S The discs $\tilde{D}_1(\lambda_1), \ldots, \tilde{D}_n(\lambda_n)$ are disconnected one another, where λ_i is a number, $0 \leq \lambda_i \leq 1$, at which radius$(\tilde{D}_i(\lambda))$ becomes largest $(1 \leq i \leq n)$.

We further make the following device. Usually, we have $\partial F/\partial u|_{x=\alpha_i, u=a} \neq 0$, then radius$(\tilde{D}_i(\lambda))$ will increase monotonously as λ increases from 0 to 1, so long as $|b - a|$ is sufficiently small. Furthermore, the denominator factor in (2.3) is independent of λ. Hence, before checking the condition S, we check a stronger and sufficient condition S'1 $\wedge$ S'2, with S'1 and S'2 given below, and we skip checking the condition S if the condition S'1 $\wedge$ S'2 is satisfied.

Pre-condition S'1 The discs $\tilde{D}_1(1), \ldots, \tilde{D}_n(1)$ are disconnected one another.

Pre-condition S'2 For each $i \in \{1, \ldots, n\}$, $|F(\alpha_i, a + \lambda(b - a))|$ increases monotonously in the interval $[0, 1]$.

In an actual program, we check the condition S'2 simply as follows. Let $P(\lambda)$ be a polynomial in λ with real coefficients. We sum the coefficient successively from the lowest to higher terms, and if the sum does not change the sign during the summation, we see that $P(\lambda)$ increases or decreases monotonously in the interval $0 \leq \lambda \leq 1$. Experiments show that this simple check is quite effective and very quick. We compute the maximum of $|P(\lambda)|$ in the interval $[0, 1]$ by evaluating it at 11 points 0.0, 0.1, $\cdots$, 0.9, 1.0.

Example 1 Checking the condition S'2 simply. Let

$$\begin{aligned} F(x, u) &= x^{10} + (u - 2)x^9 + (2u^2 - 3u - 1)x^8 + (u^5 + 3u^2 - 3)x^5 \\ &\quad - (u^5 + 2u^3 + 3u - 3)x^3 + (u^4 - 3u^2 + 5)x + (u^5 - 3u^2 - 7). \end{aligned}$$

Let $a = 0.0$ and $b = 0.1$, then $F(x, a)$ has a root $\alpha_1 \simeq 2.47054$, and we have

$$F(\alpha_1, 0.1\lambda) \simeq 0.000779\lambda^5 + 0.000247\lambda^4 - 0.0301\lambda^3 + 30.4\lambda^2 - 78.0\lambda.$$

Successive summations ((((−78.0+30.4)−0.0301)+0.000247)+0.000779) do not change the sign, hence we see that $F(\alpha_1, 0.1\lambda)$ increases monotonously in the interval $[0, 1]$. If $F(\alpha_i,\ a+\lambda(b-a))$ is of complex coefficients, then we check the real and imaginary parts separately. □

Example 2 Smith's discs for $F(x, u)$ in Example 1.
We first set $a = 0.0$ and $b = 0.1$. In Table 1, α_i and β_i are roots of $F(x,\ 0.0)$ and $F(x,\ 0.1)$, respectively, and r_i is the radius of the ith Smith's disc for $F(x,\ 0.1)$, with $c_i = \alpha_i$ $(1 \le i \le n)$. We see that the discs $\tilde{D}_1(1), \ldots, \tilde{D}_n(1)$ are disconnected one another. We have checked that the radius of each disc $\tilde{D}_i(\lambda)$ increases monotonously in the interval $[0, 1]$, hence we have the correspondence $\alpha_i \longleftrightarrow \beta_i$ $(i = 1, \ldots, 10)$.

Table 1: Smith's discs for $F(x, u)$ in Example 1.

i	α_i	β_i	$r_i/\lvert\beta_i - \alpha_i\rvert$
1	2.470	2.481	10.37
2	$0.914 + 0.274\,i$	$0.907 + 0.287\,i$	10.46
3	$0.914 - 0.274\,i$	$0.907 - 0.287\,i$	10.46
4	$0.623 + 1.094\,i$	$0.608 + 1.082\,i$	10.20
5	$0.623 - 1.094\,i$	$0.608 - 1.082\,i$	10.20
6	$-0.227 + 0.994\,i$	$-0.232 + 0.997\,i$	10.09
7	$-0.227 - 0.994\,i$	$-0.232 - 0.997\,i$	10.09
8	$-0.953 + 0.823\,i$	$-0.975 + 0.813\,i$	9.724
9	$-0.953 - 0.823\,i$	$-0.975 - 0.813\,i$	9.724
10	−1.186	−1.197	9.430

For $a = 0.0$ and $b = 0.15$, the certification of continuation succeeds for all the roots, although a simple check described in Example 1 fails for $F(\alpha_1,\ 0.15\lambda)$ and Smith's discs $\tilde{D}_8(1)$ and $\tilde{D}_9(1)$ nearly overlap. For $a = 0.0$ and $b = 0.2$, $\tilde{D}_2(1)$ overlaps with $\tilde{D}_3(1)$, and $\tilde{D}_8(1)$ with $\tilde{D}_9(1)$. □

The ratio $r_i/|\beta_i - \alpha_i|$ in Table 1 shows clearly that $\text{radius}(\tilde{D}_i(1)) \simeq n \times |\beta_i - \alpha_i|$. The reason is as follows. Let $\beta_1, \ldots, \beta_n$ be the roots of a univariate polynomial $G(x)$, and let $\alpha_1^{(k)}, \ldots, \alpha_n^{(k)}$ be their approximations, respectively. The Durand-Kerner formula for computing the roots of $G(x)$ iteratively is

$$\alpha_i^{(k+1)} = \alpha_i^{(k)} - G(\alpha_i^{(k)}) \,/ \prod_{j=1, \neq i}^{n} (\alpha_i^{(k)} - \alpha_j^{(k)}). \tag{2.5}$$

Note that r_i in (2.3) is $n\times$(correction term in (2.5)). The sequence $(\alpha_i^{(1)}, \alpha_i^{(2)}, \cdots)$ converges quadratically to β_i, i.e., $|\alpha_i^{(k+1)} - \beta_i| = O(|\alpha_i^{(k)} - \beta_i|^2)$, hence we have $r_i \simeq n \cdot |\beta_i - \alpha_i|$ so long as $|\beta_i - \alpha_i| \ll 1$.

The fact $\text{radius}(\tilde{D}_i(1)) \simeq n \times |\beta_i - \alpha_i|$ suggests that method S is not useful for polynomials of large degrees. This fact also tells us that the disc radius does not depend on $|\alpha_i - \alpha_{i'}|$, where $\alpha_{i'}$ is the root closest to α_i. On the other hand, the value of $|\alpha_i - \alpha_{i'}|$ determines the overlapping of discs crucially. Hence, we may set $|b - a|$ small for some root but may have to set large for another root. We will test method S on polynomials of large degrees in Section 5.

Note that, we must compute and check all the discs in any case. Hence, the method S is quite wasteful for the continuation of a single or several roots.

3 A method based on a bound for the smallest root (method B)

Suppose a univariate polynomial $G(x)$ has m small roots around the origin. There is a theorem which allows us to separate these m small roots from others [TS00], see also [ST02]. In this section, we derive a stronger theorem for the case of $m = 1$, and we give a method for determining the correspondence $\alpha_i \longleftrightarrow \beta_i$, by using the theorem. Below, by the *smallest root* of $G(x)$ we denote the root of the smallest magnitude.

Let a univariate polynomial $G(x) \in \mathbf{C}[x]$ be

$$\begin{cases} G(x) = g_n x^n + \cdots + g_2 x^2 + x + \varepsilon, \quad g_n \neq 0, \\ \max\{|g_n|, \cdots, |g_2|\} = 1, \quad |\varepsilon| \ll 1. \end{cases} \tag{3.1}$$

Obviously, $G(x)$ has a small root $\hat{\gamma}$ of magnitude $|\varepsilon|$. On the other hand, any root γ of polynomial $\breve{G}(x) = g_n x^{n-1} + \cdots + g_2 x + g_1$, $g_1 \neq 0$, is bounded as (see Mignotte [Mig92])

$$|\gamma| \geq \frac{|g_1|}{|g_1| + \max\{|g_2|, \cdots, |g_n|\}} = \frac{1}{1 + 1/|g_1|}. \tag{3.2}$$

If $G(x) \simeq (x - \hat{\gamma})\breve{G}(x)$, we may consider that $|g_1| \simeq 1$. Hence, the roots of $G(x)$, other than $\hat{\gamma}$, are approximately not smaller than 0.5, and we see that $\hat{\gamma}$ is the smallest root of $G(x)$. The following theorem separates $\hat{\gamma}$ from the other roots rigorously.

Theorem 2 *Let $\hat{\gamma}$ be the smallest root of $G(x)$ and $\breve{\gamma}$ be a root of $G(x)$, other than $\hat{\gamma}$. Then, so long as*

$$0 < |\varepsilon| < 3 - 2\sqrt{2} \approx 0.1715, \tag{3.3}$$

$|\hat{\gamma}|$ and $|\breve{\gamma}|$ are bounded as follows.

$$|\hat{\gamma}| < \frac{(1 + |\varepsilon|) - \sqrt{(1 + |\varepsilon|)^2 - 8|\varepsilon|}}{4}, \qquad \frac{(1 + |\varepsilon|) + \sqrt{(1 + |\varepsilon|)^2 - 8|\varepsilon|}}{4} \leq |\breve{\gamma}|. \tag{3.4}$$

Proof. Since $G(\hat{\gamma}) = 0$, we have $(g_n \hat{\gamma}^{n-1} + \cdots + g_2 \hat{\gamma} + 1) \cdot \hat{\gamma} = -\varepsilon$. Assuming that $|\hat{\gamma}| \ll 1$, which is assured by (3.4), we obtain

$$\begin{aligned} |\hat{\gamma}| &= |\varepsilon| \,/\, |1 + g_2 \hat{\gamma} + \cdots + g_n \hat{\gamma}^{n-1}| \\ &\leq |\varepsilon| \,/\, \{1 - |\hat{\gamma}| - \cdots - |\hat{\gamma}|^{n-1}\} \\ &< |\varepsilon| \,/\, \{1 - |\hat{\gamma}|/(1 - |\hat{\gamma}|)\} \\ \Longrightarrow \quad & 2|\hat{\gamma}|^2 - (1 + |\varepsilon|)|\hat{\gamma}| + |\varepsilon| > 0. \end{aligned}$$

The discriminant of the polynomial at the l.h.s. of the above last inequality is $D = (1+|\varepsilon|)^2 - 8|\varepsilon|$. In order that we have an upper bound for $|\hat{\gamma}|$, we must have $D > 0$, or (3.3). With the condition (3.3), the above last inequality gives us the bound (3.4) for $|\hat{\gamma}|$.

Similarly, dividing $G(\check{\gamma}) = 0$ by $\check{\gamma}$, we obtain $g_n\check{\gamma}^{n-1} + \cdots + g_2\check{\gamma} + g_1 = 0$, where $g_1 = 1 + \varepsilon/\check{\gamma}$. We see $g_1 \simeq 1$ because $|\check{\gamma}| \gtrsim 0.5$. We regard this as an equation in $\check{\gamma}$, of degree $n-1$ with the constant term g_1, and apply formula (3.2) to it. (We can state this situation as follows. Consider a set of polynomials $\{g_n z^{n-1} + \cdots + g_2 z + g_1 \,|\, |g_1| \le \bar{g}\}$, where $\bar{g}$ is so chosen that the set contains a polynomial having the root $\check{\gamma}$. Since g_1 is a number, we can apply formula (3.2) to every polynomial in the set.) Then, we obtain the following inequality for $|\check{\gamma}|$.

$$\begin{aligned} |\check{\gamma}| \;&\ge\; \frac{1}{1 + 1/|1 + \varepsilon/\check{\gamma}|} \;\ge\; \frac{1}{1 + 1/(1 - |\varepsilon/\check{\gamma}|)} \\ \Longrightarrow \quad & 2|\check{\gamma}|^2 - (1 + |\varepsilon|)|\check{\gamma}| + |\varepsilon| \ge 0. \end{aligned}$$

Remembering that $\check{\gamma}$ is not the smallest root, we obtain the bound (3.4) for $|\check{\gamma}|$. □

If $F(x, a)$ has no close root around $x = \alpha_i$ then $F(x + \alpha_i, b)$, with $|b - a| \ll 1$, will have a small root $\hat{\gamma}_i$, $|\hat{\gamma}_i| = |\beta_i - \alpha_i| \ll 1$, and other roots will not be so small as $\hat{\gamma}_i$. By investigating the separation between the smallest root $\hat{\gamma}_i$ and other roots, we can determine the correspondence $\alpha_i \longleftrightarrow \hat{\gamma}_i$, as the following corollary asserts.

Corollary 2 *Let a polynomial $\hat{G}_i(x, \lambda)$ be*

$$\hat{G}_i(x, \lambda) = F(x + \alpha_i,\, a + \lambda(b - a)), \tag{3.5}$$

where λ is a parameter such that $0 \le |\lambda| \le 1$. Let $\hat{\gamma}_i(\lambda)$ be a root of $\hat{G}_i(x, \lambda)$ satisfying $\hat{\gamma}_i(0) = 0$. Let $\hat{G}_i(x, \lambda)$ and $\varepsilon_i(\lambda)$ be expressed and defined, respectively, as

$$\hat{G}_i(x, \lambda) = g_n x^n + \cdots + g_2 x^2 + g_1 x + g_0, \quad g_1 \ne 0, \tag{3.6}$$

$$\varepsilon_i(\lambda) = |g_0/g_1| \times \max\{\, \sqrt[n-1]{|g_n/g_1|},\; \cdots,\; \sqrt[3]{|g_3/g_1|},\; |g_2/g_1| \,\}. \tag{3.7}$$

Then, so long as $\varepsilon_i(\lambda) < 3 - 2\sqrt{2}$ for any λ on a path connecting 0 and 1, we have the correspondence $\alpha_i \longleftrightarrow \hat{\gamma}_i(1)$ $(= \beta_i - \alpha_i) \longleftrightarrow \beta_i$, where β_i is the root closest to α_i.

Proof. Determining a positive number η to satisfy $|g_1\eta| = \max\{\, |g_n\eta^n|,\; \cdots,\; |g_2\eta^2| \,\}$, we have $\eta = 1/\max\{\, \sqrt[n-1]{|g_n/g_1|},\; \cdots,\; \sqrt[3]{|g_3/g_1|},\; |g_2/g_1| \,\}$. Making the transformation $x \to \eta x$ in (3.5), we see that the condition $\varepsilon_i(\lambda) < 3 - 2\sqrt{2}$ is equivalent to the condition for $|\varepsilon|$ in (3.3).

Let $\hat{D}_i(\lambda)$ be a disc of radius $\left|(1 + \varepsilon_i(\lambda)) - \sqrt{(1 + \varepsilon_i(\lambda))^2 - 8\varepsilon_i(\lambda)}\right| / 4$, located at the origin. At $\lambda = 0$, $\hat{D}_i(0)$ is of radius 0 and it contains a root $\hat{\gamma}_i(0) = 0$. At $0 < |\lambda| \le 1$, the assumption tells us that $\hat{D}_i(\lambda)$ contains only one root that is the smallest root of $\hat{G}_i(x, \lambda)$. Hence, the corollary is a direct consequence of the fact that the roots of the univariate polynomial are continuous functions in its coefficients. □

In order to use the above corollary in actual continuation, we have to devise a simple method for checking the condition $|\varepsilon_i(\lambda)| < 0.1715$ for any value of λ on a path connecting 0 and 1. First, we choose the path to be a line. Next, consider the polynomial

$$F(x + \alpha_i,\, a + \lambda(b - a)) = \hat{f}_n(\lambda)x^n + \hat{f}_{n-1}(\lambda)x^{n-1} + \cdots + \hat{f}_0(\lambda). \tag{3.8}$$

Since $\hat{f}_0(\lambda) = F(\alpha_i,\, a + \lambda(b - a))$ and $\hat{f}_0(0) = 0$, if we increase λ from 0 to 1, the value of $|\hat{f}_0(\lambda)|$ will change (increase) rapidly while the other coefficients will not change much, so long as $|b - a| \ll 1$. Therefore, we replace the above condition by the following sufficient condition.

Table 2: The smallest roots for $F(x,u)$ in Example 1.

i	$\hat{\gamma}_i$	ε_i	$B_i/\lvert\hat{\gamma}_i\rvert$
1	0.0382	0.0397	1.085
2	$-0.0203+0.0367\,i$	0.0421	1.052
3	$-0.0203-0.0367\,i$	0.0421	1.052
4	$-0.0575-0.0477\,i$	0.0698	1.018
5	$-0.0575+0.0477\,i$	0.0698	1.018
6	$-0.0169+0.0092\,i$	0.0195	1.033
7	$-0.0169-0.0092\,i$	0.0195	1.033
8	$-0.0940-0.0413\,i$	0.1073	1.219
9	$-0.0940+0.0413\,i$	0.1073	1.219
10	-0.0428	0.0447	1.098

Condition B The relation $\varepsilon_i(\lambda) < 3-2\sqrt{2}$ still holds if, in formula (3.7), we replace $|g_1|$ and $|g_j|$ $(j \neq 1)$ by the minimum of $|\hat{f}_1(\lambda)|$ and the maximum of $|\hat{f}_j(\lambda)|$, respectively, in the interval $[0,1]$.

In an actual program, we check the following necessary condition B' before the condition B, and judge that $|b-a|$ is too large if the condition B' is not satisfied.

Pre-condition B' $\varepsilon_i(1) < 3-2\sqrt{2}$.

Example 3 The smallest roots for $F(x,u)$ in Example 1.
We first set $a=0.0$ and $b=0.1$. In Table 2, $\hat{\gamma}_i$ is the smallest root of $F(x+\alpha_i,\,0.1)$, ε_i is the value of $\varepsilon_i(1)$, and B_i is the upper bound for the smallest root $(i=1,\ldots,10)$.

We see that the upper bound for the smallest root is very sharp, so long as $\varepsilon_i(1) \lesssim 0.1$. On the other hand, the lower bound for the other roots is not so sharp. We have checked that the condition B is satisfied by all the roots, hence we have the correspondence $\alpha_i \longleftrightarrow \beta_i$ $(i=1,\ldots,10)$.

For $a=0.0$ and $b=0.15$, the simple check described in Example 1 fails for $F(x+\alpha_1,\,0.15\lambda)$ and $\hat{\gamma}_8=\hat{\gamma}_9 \simeq 0.1742$, hence the certification of continuation fails for β_8 and β_9 although the value of $\hat{\gamma}_i$ is much smaller than 0.1715 for for $i \neq 8,9$. □

We see that the values of ε_i's are distributed widely, which means that we may set $|b-a|$ rather large for some root but may have to set small for another root. Note that method B requires no information on the other roots, hence the method does not make any wasteful computation in the continuation of a single or several roots. This is a very advantageous point over method S, as we will see below.

4 A method using the Taylor series (method T)

Let $\varphi_1^{(k)}(u-a),\ldots,\varphi_n^{(k)}(u-a)$ be the power-series roots expanded at a and truncated at power k. (For convenience, we change the notation for truncated power-series roots slightly).

Note that $\varphi_i^{(0)} = \alpha_i$ $(i = 1, \ldots, n)$. In previous two sections, we have utilized only $\varphi_i^{(0)}$ to localize the existence domain of the root β_i of $F(x, b)$. If we utilize the Taylor series $\varphi_i^{(k)}(u-a)$, $k \geq 1$, then we will be able to localize the existence domain much more sharply.

Let us consider generalizing method S to utilize the Taylor-series roots. Smith's discs $\tilde{D}_i(\lambda)$ $(i = 1, \ldots, n)$, with λ a parameter such that $0 \leq |\lambda| \leq 1$, are now defined as

$$\text{center}(\tilde{D}_i(\lambda)) = \varphi_i^{(k)}(\lambda(b-a)), \tag{4.1}$$

$$\text{radius}(\tilde{D}_i(\lambda)) = \frac{n\,|F(\varphi_i^{(k)}(\lambda(b-a)),\, a+\lambda(b-a))|}{|\prod_{j=1,\neq i}^{n}[\varphi_i^{(k)}(\lambda(b-a)) - \varphi_j^{(k)}(\lambda(b-a))]|}. \tag{4.2}$$

The above expressions are much more complicated than those in Section 2, and the check of non-overlapping is not easy because the denominator factor in (4.2) depends on λ. Therefore, we do not consider this approach any more.

On the other hand, generalizing method B to utilize the Taylor-series roots is straightforward. We generalize Corollary 2 as follows (the proof is the same as that for Corollary 2).

Corollary 3 *Let the truncation power k be positive, and let a polynomial $\hat{G}_i(x, \lambda)$ be*

$$\hat{G}_i(x, \lambda) = F(x + \varphi_i^{(k)}(\lambda(b-a)),\, a + \lambda(b-a)), \tag{4.3}$$

where λ is a parameter such that $0 \leq |\lambda| \leq 1$. Let $\hat{\gamma}_i(\lambda)$ be a root of $\hat{G}_i(x, \lambda)$ satisfying $\hat{\gamma}_i(0) = 0$. We express $\hat{G}_i(x, \lambda)$ as in (3.6), and define $\varepsilon_i(\lambda)$ by formula (3.7). Then, so long as $\varepsilon_i(\lambda) < 3 - 2\sqrt{2}$ for any λ on a path connecting 0 and 1, we have the correspondence $\alpha_i \longleftrightarrow \hat{\gamma}_i(1)$ $(= \beta_i - \varphi_i^{(k)}(b-a)) \longleftrightarrow \beta_i$, where β_i is the root closest to α_i.

Checking the condition $\varepsilon_i(\lambda) < 3 - 2\sqrt{2}$ is the same as that in method B. Only one remarkable difference is that $\deg_\lambda(\hat{G}_i(x, \lambda))$ is much larger in method T than in method B, because $x + \varphi_i^{(k)}(\lambda(b-a))$ is substituted for x in $F(x, u)$. As we will see in Section 5, this makes method T quite time-consuming.

Example 4 Testing method T near a singular point.
We again consider $F(x, u)$ in Example 1. $F(x, u)$ has a singular point at $u \simeq 1.843774074$. We set $a = 1.844$ and $a = 1.8438$, respectively, and set $b = a + w$, with $w > 0$. We measured the maximum width $w_{\max}$, where by the maximum width we mean that the certification of continuation succeeds for w such that $w < w_{\max}$ and fails for $w > w_{\max}$.

Table 3 shows the results, where k is the truncation power, $w_{\max}^{(k)}$ is the maximum width for the Taylor-series root truncated at power k, and $T^{(k)}$ is the computation time for the continuation of all the roots. Note that the distance between the singular point and the expansion point is 2.26×10^{-4} for $a = 1.844$ and 2.6×10^{-5} for $a = 1.8438$. Hence, the method T allows us to choose w considerably large. □

5 Experimental test of methods S, B and T

We have tested methods S, B and T on various polynomials of degrees 10, 20, and 50 w.r.t. x and of degree 3 w.r.t u. For each degree, we generate 10 monic polynomials with coefficients chosen rather randomly from $\{-7, -6, \cdots, 6, 7\}$, so that the polynomial contains about 30 terms

Table 3: Testing method T near a singular point.

	$a = 1.844$		$a = 1.8438$	
k	$w_{\max}^{(k)}$	$T^{(k)}$ (sec)	$w_{\max}^{(k)}$	$T^{(k)}$ (sec)
0	1.54×10^{-4}	0.041	1.77×10^{-5}	0.044
1	3.72×10^{-4}	0.071	4.29×10^{-5}	0.073
2	4.44×10^{-4}	0.082	5.12×10^{-5}	0.086
3	7.62×10^{-4}	0.091	8.85×10^{-5}	0.096
4	8.66×10^{-4}	0.106	9.97×10^{-5}	0.109

for $\deg_x(F) = 10$ or 20 and about 40 terms for $\deg_x(F) = 50$. For each polynomial, we set $a = 0$ and b as shown in the first row of each sub-table.

The program was written in Japanese algebra system GAL and executed on a machine with CPU/celeron (733 MHz) and 64 M-byte memory. We comment that the computation was performed as efficiently as possible. For example, in method T, we must handle $F(x + \varphi_i^{(k)}(\lambda(b-a)), a+\lambda(b-a))$. However, substitution of $x+\varphi_i^{(k)}(\lambda(b-a))$ for x in $F(x,u)$ is very time-consuming if $\deg_x(F)$ is large. On the other hand, we need only three coefficients of the resulting polynomial (coefficients of x^0-, x^1- and x^m-terms, $m = 2$ usually). Hence, we compute only the necessary coefficients directly. Furthermore, we discard the terms s^l, $l > 20$ at largest.

Table 4 shows the results, where $T_{\rm av}$ is the average CPU time and the integers in the third to seventh columns in each sub-table show the number of samples for which the continuation succeeds for all the n roots; for example, the number 8 at the sixth column in the table for Method S denotes that the continuation succeeds for 8 samples among 10 if we set b as $|b-a| = 0.03$. We see that, in method S, we must choose the width $|b-a|$ smaller and smaller as n increases. We need not choose the width so small in method B, but this method is more time-consuming than method S. We can choose the width much larger in method T, but this method is even more time-consuming.

In an actual continuation, the expansion points a and b are given and we must perform the continuation by dividing the interval $[a,b]$ into sub-intervals $[a,c_1]$, $[c_1,c_2]$, $\cdots$, $[c_{l-1},c_l]$, $[c_l,b]$. Then, the cost of the continuation is $(l+1)\times$sub-cost, where the sub-cost is the cost for the continuation in the sub-interval. Therefore, usefulness of each method may be summarized as follows.

Method S :

1. Method S is useful for polynomials of low and medium degrees.
2. It should be used only for the continuation of all the roots.

Method B :

1. Method B is equally useful, especially for polynomials of large degrees.
2. It should be used for the continuation of several roots.

Method T :

1. Method T is equally efficient as method B but complicated.
2. It may be useful for the continuation near singular points.

Table 4: Testing methods S,B,T on random polynomials.

$T_{\rm av}$ is the computation time averaged over 10 samples.
First row : each number shows the value of b for $a = 0$.
Right 5 columns : each integer shows the # of samples for which the continuation succeeds (see the text).

Method S

degree	$T_{\rm av}$(sec)	0.001	0.003	0.010	0.033	0.100
10	0.021	10	10	10	8	2
20	0.056	10	10	7	3	
50	0.260	10	7	2		

Method B

degree	$T_{\rm av}$(sec)	0.001	0.003	0.010	0.033	0.100
10	0.031	10	10	10	9	3
20	0.121	10	10	9	5	3
50	0.974	10	10	7	3	1

Method T : $k = 1$

degree	$T_{\rm av}$(sec)	0.020	0.050	0.080	0.200	0.500
10	0.066	10	10	10	4	
20	0.226	10	9	8	2	
50	1.527	10	10	6	1	

Method T : $k = 3$

degree	$T_{\rm av}$(sec)	0.020	0.050	0.080	0.200	0.500
10	0.138	10	10	10	9	3
20	0.377	10	10	9	7	
50	2.182	10	10	10	3	

6 On determining the width $|b - a|$

So far, we have assumed that the width $|b - a|$ is sufficiently small. In this section, we consider how small the width should be.

On method S. As mentioned in Section 2, $r_i \simeq n \cdot |\beta_i - \alpha_i|$ if $\alpha_i \simeq \beta_i$ $(i = 1, \ldots, n)$, where β_i is the root closest to α_i among the roots of $F(x, b)$. Therefore, b should be chosen to satisfy

$$n|\beta_i - \alpha_i| < |\alpha_i - \alpha_{i'}|/2, \quad i = 1, \ldots, n, \tag{6.1}$$

where $\alpha_{i'}$ is the root closest to α_i among the roots $\{\alpha_1, \cdots, \alpha_n\} \setminus \{\alpha_i\}$. Once we have computed the roots of $F(x, a)$, we can compute the roots of $F(x, b)$ quickly, so long as $|b - a| \ll 1$. Therefore, in the actual computation, we temporarily set b close to a and check the condition (6.1) by computing the roots of $F(x, b)$. Furthermore, we had better choose the denominator in the r.h.s. of (6.1) as 3 or 4.

On methods B and T. Suppose $\varepsilon_i(\lambda) = |g_0/g_1|\sqrt[m-1]{|g_m/g_1|}$ in (3.7), then we have to determine b to satisfy $|g_0/g_1|\sqrt[m-1]{|g_m/g_1|} < 3-2\sqrt{2}$. What we handle actually is the polynomial in (3.8), and we have to determine b so as to satisfy

$$\frac{|\hat{f}_{0,\max}|}{|\hat{f}_{1,\min}|} \cdot \left(\frac{|\hat{f}_{m,\max}|}{|\hat{f}_{1,\min}|}\right)^{1/(m-1)} < 3-2\sqrt{2}, \tag{6.2}$$

where $|\hat{f}_{1,\min}|$ and $|\hat{f}_{j,\max}|$ $(j \neq 1)$ are the minimum of $|\hat{f}_1(\lambda)|$ and the maximum of $|\hat{f}_j(\lambda)|$, respectively, in the interval $[0,1]$. We comment that $m = 2$ in most practical cases.

Actually, we determine the width as follows. The most influential factor in (6.2) is $|\hat{f}_{0,\max}|$. Since $\hat{f}_0(\lambda) = F(\alpha_i,\, a+\lambda(b-a))$, we have $\hat{f}_0(\lambda) = \partial F(\alpha_i, u)/\partial u|_{u=a} \cdot \lambda(b-a) + O(\lambda^2(b-a)^2)$, so long as $|b-a| \ll 1$. Therefore, $|\hat{f}_0(\lambda)|$ usually increases almost linearly as λ increases from 0 to 1. Thus, we set b temporarily and compute the value of $\varepsilon_i(1)$, and if the value is L times larger (smaller) than 0.1715 then we narrow (widen) the width by about L times.

7 Continuation of roots at singular point

In this section, without loss of generality, we assume that the origin is a singular point of $F(x,u)$. If $f_n(0) = 0$, we can put $f_n(u) = u^d \tilde{f}_n(u)$, with $\tilde{f}_n(0) \neq 0$, and we perform the transformation $F(x,u) \mapsto \tilde{F}(x,u) = u^{(n-1)d}F(x/u^d,\, u) = \tilde{f}_n(u)x^n + f_{n-1}(u)x^{n-1} + u^d f_{n-2}(u)x^{n-2} + \cdots$. Let a root of $\tilde{F}(x,u)$ w.r.t. x be $\tilde{\varphi}(u)$, then the corresponding root $\varphi(u)$ of $F(x,u)$ is given as $\varphi(u) = \tilde{\varphi}(u)/u^d$. Therefore, we further assume that $f_n(0) \neq 0$. In this case, $F(x,0)$ has multiple roots and all the above three methods fall down for the multiple roots.

We will perform the continuation of roots expanded at different singular points via the roots expanded at non-singular points. Therefore, in this section, we assume that b is a non-singular point such that $|b| \ll 1$, and consider the continuation between the roots expanded at the origin and those expanded at b.

One may think that if we use Puiseux-series roots instead of Taylor-series roots in method T then we can perform the continuation. This is the appraoch of Shiihara and Sasaki [SS96], although they used Smith's theorem. This approach gives correct results in most cases, however, we cannot certify the results because some roots of $\tilde{G}(x,\lambda)$ in (2.4) or $\hat{G}_i(x,\lambda)$ in (4.3) coincide as $\lambda \to 0$. In order to perform the certification, we "blow up" the u-coordinate at the origin so that we can distinguish the multiple roots of $F(x,0)$ at the origin.

Under the assumptions given above, there exists an integer m, $2 \leq m \leq n$, satisfting $f_0(0) = f_1(0) = \cdots = f_{m-1}(0) = 0$, $f_m(0) \neq 0$. By $\mathrm{ord}(f_j)$, with $f_j(u)$ a univarite polynomial in u, we denote the *order* of $f_j(u)$, that is the minimum exponent among the terms of $f_j(u)$. We determine a rational number ν as follows.

$$\nu \stackrel{\mathrm{def}}{=} \min\{\, \mathrm{ord}(f_j)/(m-j) \mid j = 0, 1, \ldots, m-1 \,\}. \tag{7.1}$$

With this number, we perform the transformation

$$F(x,u) \mapsto \tilde{F}(y,u) = F(u^\nu y,\, u)/u^{\nu m}, \tag{7.2}$$

where y is a new variable. $\tilde{F}(y,u)$ is of the following form.

$$\begin{aligned}\tilde{F}(y,u) \;=\; & f_n(u)u^{\nu(n-m)}y^n + f_{n-1}(u)u^{\nu(n-m-1)}y^{n-1} + \cdots \\ & + f_m(u)y^m + (f_{m-1}(u)/u^\nu)y^{m-1} + \cdots + (f_0(u)/u^{\nu m}).\end{aligned}$$

At $u = 0$, the coefficients of terms of degrees greater than m, of $\tilde{F}(y, u)$ disappear, which means that $n - m$ roots corresponding to nonzero roots of $F(x, 0)$ move to infinity as $u \to 0$. $\tilde{F}(y, 0)$ has m roots, some of them may be single and others may be multiple. The single roots of $\tilde{F}(y, 0)$ generate Taylor-series roots of $\tilde{F}(y, u)$, and we can apply method B or method T to these roots. For the multiple roots of $\tilde{F}(y, 0)$, we must apply the blow up procedure recursively.

Note that method S is inapplicable to $\tilde{F}(y, u)$ because $\tilde{F}(y, u)$ has roots which go to infinity as $u \longrightarrow 0$ hence we cannot define Smith's disc for them.

Example 5 Continuation of Puiseux-series roots. Let

$$F(x, u) = x^4 + (3 + 2u)x^3 + (2 - 4u + 5u^2)x^2 - (3u + 7u^2)x - (2u + 5u^2 - 8u^3).$$

We see $F(x, 0) = x^4 + 3x^3 + 2x^2 = x^2(x + 1)(x + 2)$ and $F(x, u)$ has at least two Taylor-series roots. We also have $m = 2$ and $\nu = \min\{\mathrm{ord}(f_0)/2, \mathrm{ord}(f_1)/1\} = 1/2$, hence $F(x, u)$ has two Puiseux-series roots. The transformation (7.2) is now $\tilde{F}(y, u) = F(\sqrt{u}y, u)/u$:

$$\tilde{F}(y, u) = uy^4 + \sqrt{u}(3 + 2u)y^3 + (2 - 4u + 5u^2)y^2 - \sqrt{u}(3 + 7u)y - (2 + 5u - 8u^2).$$

$\tilde{F}(y, 0)$ has two single roots $\alpha_1 = 1$ and $\alpha_2 = -1$. We put $b = 0.04$, then $\tilde{F}(y, 0.04)$ has four roots $\beta_1 \simeq 1.063$, $\beta_2 \simeq -1.096$, $\beta_3 \simeq -4.194$, $\beta_4 \simeq -11.17$. Investigating $\tilde{F}(y+1, 0.04)$ and $\tilde{F}(y-1, 0.04)$, we find that $\varepsilon_1(1) \simeq 0.0524$ and $\varepsilon_2(1) \simeq 0.0403$. Furthermore, the condition B holds for both $\tilde{F}(y+1, 0.04\lambda)$ and $\tilde{F}(y-1, 0.04\lambda)$. Therefore, we have the correspondence $\alpha_i \longleftrightarrow \beta_i$ $(i = 1, 2)$.

The certification of continuation succeeds for $b = 0.09$ but fails for $b = 0.16$. □

References

[CG..01] R.M. Corless, M.W. Giesbrecht, M. van Hoeij, I.S. Kotsireas and S.M. Watt, Towards factoring bivariate approximate polynomials, in: Proc. ISSAC 2001, ACM Press, 2001, pp.85-92.

[GR01] A. Galligo and D. Rupprecht, Semi-numerical determination of irreducible branches of a reduced space curve, in: Proc. ISSAC 2001, ACM Press, 2001, pp.137-142.

[GW97] A. Galligo and S.M. Watt, A numerical absolute primality test for bivariate polynomials, in: Proc. ISSAC'97, ACM Press, 1997, pp.217-224.

[KT78] H.T. Kung and J.F. Traub, All algebraic functions can be computed fast, *J. ACM* **25** (1978), 245-260.

[Mig92] M. Mignotte M, *Mathematics for Computer Algebra*, Springer-Verlag, 1992, Ch. 4.

[SK99] T. Sasaki and F. Kako, Solving multivariate algebraic equation by Hensel construction, *Japan J. Indus. Appl. Math.* **16** (1999), 257-285. (This paper was written in January 1993 and submitted soon, and the authors received the referees' reports in September 1996. The authors sent a revised manuscript immediately, and they received a letter of acceptance in June 1998.)

[SSKS91] T. Sasaki, M. Suzuki, M. Kolář and M. Sasaki, Approximate factorization of multivariate polynomials and absolute irreducibility testing, *Japan J. Indus. Appl. Math.* **8** (1991), 357-375.

[SSH92] T. Sasaki, T. Saito and T. Hilano, Analysis of approximate factorization algorithm I, *Japan J. Indust. Appl. Math.* 9 (1992), 351-368.

[SS96] K. Shiihara and T. Sasaki, Analytic continuation and Riemann surface determination of algebraic functions by computer, *Japan J. Indus. Appl. Math.* 13 (1996), 107-116.

[Sas01] T. Sasaki T, Approximate multivariate polynomial factorization based on zero-sum relations, in: Proc. ISSAC 2001, ACM Press, 2001, pp.285-291.

[Smi70] B.T. Smith, Error bounds for zeros of a polynomial based on Gerschgorin's theorems, *J. ACM* 17 (1970), 661-674.

[ST02] T. Sasaki and A. Terui, A formula for separating small roots of a polynomial, Preprint of Univ. Tsukuba, 2002 (to appear).

[TS00] A. Terui and T. Sasaki, "Approximate zero-points" of real univariate polynomial with large error terms, *IPSJ Journal (Information Processing Society of Japan)* 41 (2000), 974-989.

A Formula for Separating Small Roots of a Polynomial *

Tateaki Sasaki and Akira Terui

Institute of Mathematics, University of Tsukuba
Tsukuba-shi, Ibaraki 305-8571, Japan
sasaki,terui@math.tsukuba.ac.jp

Abstract

Let $P(x)$ be a univariate polynomial over C, such that $P(x) = c_n x^n + \cdots + c_{m+1}x^{m+1} + x^m + e_{m-1}x^{m-1} + \cdots + e_0$, where $\max\{|c_n|, \cdots, |c_{m+1}|\} = 1$ and $e = \max\{|e_{m-1}|, |e_{m-2}|^{1/2}, \cdots, |e_0|^{1/m}\} \ll 1$. $P(x)$ has m small roots around the origin so long as $e \ll 1$. In 1999, we derived a formula that if $e < 1/9$ then $P(x)$ has m roots inside a disc $D_{\rm in}$ of radius $R_{\rm in}$ and other $n-m$ roots outside a disc $D_{\rm out}$ of radius $R_{\rm out}$, located at the origin, where $R_{\rm in(out)} = [1 - (+)\sqrt{1-(16e)/(1+3e)^2}] \cdot (1+3e)/4$. Note that $R_{\rm in} = R_{\rm out}$ if $e = 1/9$. Our formula is essentially the same as that derived independently by Yakoubsohn at almost the same time. In this short article, we introduce the formula and check its sharpness on many polynomials generated randomly.

1 Introduction

Given a univariate polynomial, there are many formulas which express upper-bounds or lower-bounds of the roots of the polynomial in terms of its coefficients; see [Mig91] for example. A formula we introduce in this article separates the roots into two groups, small and larger ones. We derived the formula in 1999; see [TS00]. Yakoubsohn [Yak00] derives essentially the same formula with a different proof, and Smale [Sma86] and Sasaki-Inaba [SI02] also derive similar formulas for separating a single root.

The formula will be useful in at least three points. First, the formula is useful to verify the convergence of Newton's method for iteratively computing roots of the univariate polynomial; see [BCSS98]. In fact, Smale and Yakoubsohn derived their formulas from this point. Second, the formula gives us existence domains of the roots of polynomials having error terms; let $P(x) = P_0(x) + E(x)$, where $P_0(x)$ is a polynomial with exact coefficients and $E(x)$ is the sum of error terms such that $\|E\| \ll \|P_0\|$. The roots of $P_0(x)$ are blurred by $E(x)$, and we want to bound the existence domains of the roots. In fact, in [TS00], we derived our formula from this point. Third, given a bivariate polynomial $P(x,u)$ and numbers a and δ, with $|\delta| \ll 1$, we want to determine the existence domains of the roots of univariate polynomial $P(x, a+\delta)$ by the coefficients of $P(x,a)$. In fact, Sasaki and Inaba derived and applied their formula to construct algorithms of analytic continuation of algebraic functions; see [SI02].

*Work supported in part by Japanese Ministry of Education, Science and Culture under Grants 12480065.

In Section 2, following [TS00], we derive the formula. In Section 3, we check the sharpness of our formula numerically on many polynomials generated randomly. We compare our formula with a bound which is derived by using Smith's theorem [Smi70]. We will see that our formula is much sharper than the bound derived by Smith's theorem.

2 Derivation of the formula

Let $P(x)$ be the following univariate polynomial over C:

$$P(x) = c_n x^n + \cdots + c_{m+1}x^{m+1} + x^m + e_{m-1}x^{m-1} + \cdots + e_0, \tag{1}$$

where the coefficients satisfy the following two conditions

$$\begin{cases} e \stackrel{\rm def}{=} \max\{|e_{m-1}|, |e_{m-2}|^{1/2}, \cdots, |e_0|^{1/m}\} \ll 1, \\ \max\{|c_n|, \cdots, |c_{m+1}|\} = 1. \end{cases} \tag{2}$$

$P(x)$ has m small roots around the origin. In [TS00], we proved the following theorem so as to separate the m small roots from others.

Theorem 1 *If $e < 1/9$ then $P(x)$ has m small roots inside a disc $D_{\rm in}$ of radius $R_{\rm in}$ and other $n-m$ roots outside a disc $D_{\rm out}$ of radius $R_{\rm out}$, located at the origin, where*

$$R_{\rm in(out)} = \frac{1+3e}{4} \cdot \left[1 - (+)\sqrt{1 - \frac{16e}{(1+3e)^2}}\right]. \tag{3}$$

The radii $R_{\rm in}$ and $R_{\rm out}$ satisfy the following inequalities.

$$R_{\rm in} < 2e \cdot \left[\frac{1}{1+3e} + \frac{16e}{(1+3e)^3}\right], \tag{4}$$

$$R_{\rm out} > \frac{1}{2} - \frac{e(1-9e)}{2(1+3e)} - \frac{32e^2}{(1+3e)^3}. \tag{5}$$

We can prove the theorem by using Rouche's theorem, as Yakoubsohn did in [Yak00], but we have proved it by using the following well-known formulas (for the proof, see [Mig91]).

Lemma 1 *Let $A(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$, with $a_n a_0 \neq 0$, be a polynomial with complex coefficients. Let the roots of $A(x)$ be $\zeta_1, \ldots, \zeta_n$, then*

$$\max\{|\zeta_1|, \cdots, |\zeta_n|\} \leq \frac{|a_n| + \max\{|a_{n-1}|, \cdots, |a_0|\}}{|a_n|}, \tag{6}$$

$$\min\{|\zeta_1|, \cdots, |\zeta_n|\} \geq \frac{|a_0|}{|a_0| + \max\{|a_1|, \cdots, |a_n|\}}. \tag{7}$$

Let $\zeta_1, \ldots, \zeta_n$ be the roots of $P(x)$, satisfying

$$|\zeta_1| \le \cdots \le |\zeta_m| < |\zeta_{m+1}| \le \cdots \le |\zeta_n|. \tag{8}$$

We first investigate the magnitudes of the roots roughly. Put $P'(x) = x^m + e_{m-1}x^{m-1} + \cdots + e_0$ and $P''(x) = c_n x^{n-m} + \cdots + c_{m+1}x + 1$. Lemma 1 tells us that the roots of $P'(x)$ are smaller than or equal to $2e$ and those of $P''(x)$ are larger than or equal to $1/2$. Since $e \ll 1$, we have $P(x) \approx P'(x)P''(x)$. Therefore, we see $|\zeta_m| \lesssim 2e$ and $|\zeta_{m+1}| \gtrsim 1/2$.

Proof of Theorem 1 (same as Appendix of [TS00]) We first consider a root ζ such that $|\zeta| \le |\zeta_m| \lesssim 2e$. Putting $\zeta = e\bar{\zeta}$ in $P(\zeta) = 0$ and dividing $P(e\bar{\zeta})$ by e^m, we obtain

$$\begin{aligned}&[c_n(e\bar{\zeta})^{n-m} + \cdots + c_{m+1}(e\bar{\zeta}) + 1] \cdot \bar{\zeta}^m \\ &\quad + (e_{m-1}/e)\bar{\zeta}^{m-1} + \cdots + (e_0/e^m)\bar{\zeta}^0 \;=\; 0.\end{aligned} \tag{9}$$

If we consider (9) as an equation w.r.t. $\bar{\zeta}$, then the root $\bar{\zeta}$ satisfying $|\bar{\zeta}| \lesssim 2$ is determined mostly by the terms of degrees $\le m$ and the terms $c_{m+j}(e\bar{\zeta})^j$ $(j = 1, \ldots, n-m)$ in $[\cdots]$ contribute only as small corrections because $e \ll 1$. Thus, we regard (9) as an equation in $\bar{\zeta}$, of degree m with the numerical leading coefficient $a_m = 1 + c_{m+1}(e\bar{\zeta}) + \cdots + c_n(e\bar{\zeta})^{n-m}$. (We can state this situation as follows. Consider a set of equations in z, of degree m:

$$\begin{cases} a_m z^m + (e_{m-1}/e)z^{m-1} + \cdots + (e_0/e^m)z^0 \;=\; 0, \\ a_m \;\in\; \{1 + c_{m+1}(e\bar{z}) + \cdots + c_n(e\bar{z})^{n-m} \mid |\bar{z}| \le \bar{\zeta}_{\max}\}, \end{cases}$$

where $\bar{\zeta}_{\max}$ is so chosen that the set contains a polynomial having the root $\bar{\zeta} = \zeta_m/e$. Since a_m is a number, we can apply Lemma 1 to each equation in the set. Note that, if we put $\zeta_{\max} = R_{\rm in}/e$, then the following calculation is valid for every polynomial in the set.) Thus, regarding (9) as above and applying Lemma 1, we obtain

$$\begin{aligned}|\bar{\zeta}| \;&\le\; 1 + \max\{|e_{m-1}/e|, \cdots, |e_0/e^m|\}/|a_m| \\ &\le\; 1 + \frac{1}{1 - |e\bar{\zeta}| - \cdots - |e\bar{\zeta}|^{n-m}} \\ &<\; 1 + \frac{1}{1 - |e\bar{\zeta}|/(1 - |e\bar{\zeta}|)} \;=\; \frac{2 - 3|e\bar{\zeta}|}{1 - 2|e\bar{\zeta}|},\end{aligned}$$

or

$$|\zeta| < \frac{2e - 3e|\zeta|}{1 - 2|\zeta|} \implies 2|\zeta|^2 - (1 + 3e)|\zeta| + 2e > 0. \tag{10}$$

Let z_- and z_+ be two solutions of $2z^2 - (1 + 3e)z + 2e = 0$, with $|z_-| \le |z_+|$. We see that $z_\pm$ are real and positive if and only if $e \le 1/9$, and $z_- \simeq 2e$ and $z_+ \simeq (1 - e)/2$ for $e \ll 1$. Therefore, we have the formula for $R_{\rm in}$ in (3). Using the identity $\sqrt{1-x} > 1 - x/2 - x^2/2$, which is valid for $0 < x < 1$, and putting $x = 16e/(1 + 3e)^2$, we obtain (4).

Next, we consider a root ζ such that $|\zeta| \gtrsim |\zeta_{m+1}| \gtrsim 1/2$. Dividing $P(\zeta) = 0$ by ζ^m, we obtain

$$\begin{aligned}&c_n\zeta^{n-m} + \cdots + c_{m+1}\zeta \\ &\quad + [1 + e_{m-1}/\zeta + \cdots + e_0/\zeta^m] \;=\; 0.\end{aligned} \tag{11}$$

Since $|\zeta| \gtrsim 1/2$, the terms e_{m-j}/ζ^j $(j = 1, \ldots, m)$ in $[\cdots]$ contribute only as small corrections because $|e_{m-j}| \ll 1$. Thus, with the same reasoning as for (9), we regard (11) as an equation in ζ, of degree $n - m$ with the constant term $a_0 = 1 + e_{m-1}/\zeta + \cdots + e_0/\zeta^m \approx 1$. Then, by Lemma 1, we obtain

$$\begin{aligned}|\zeta| \;&\ge\; \frac{1}{1 + \max\{|c_n|, \cdots, |c_{m+1}|\}/|a_0|} \\ &\ge\; \frac{1}{1 + 1/(1 - |e/\zeta| - \cdots - |e/\zeta|^m)} \\ &>\; \frac{1}{1 + \frac{1}{1 - |e/\zeta|/(1 - |e/\zeta|)}} \;=\; \frac{1 - 2|e/\zeta|}{2 - 3|e/\zeta|},\end{aligned}$$

or

$$2|\zeta|^2 - (1 + 3e)|\zeta| + 2e > 0. \tag{12}$$

This equation is the same as that in (10), and we find $|\zeta| > z_+$ so long as $e \le 1/9$, or the formula for $R_{\rm out}$ in (3). Using the identity $\sqrt{1-x} > 1 - x/2 - x^2/2$, which is valid for $0 < x < 1$, we obtain (5). □

Remark 1 *The small roots and other roots (large roots) are invertible in the following sense. Consider the transformation $P(x) \mapsto \tilde{P}(x) = (x^n/e^m)P(e/x)$, then we have*

$$\begin{aligned}\tilde{P}(x) \;&=\; (e_0/e^m)x^n + \cdots + (e_{m-1}/e)x^{n-m+1} \\ &\quad + x^{n-m} + (c_{m+1}e)x^{n-m+1} + \cdots + (c_n e^{n-m}).\end{aligned} \tag{13}$$

This polynomial satisfies the similar conditions as those in (2), *and $\tilde{P}(x)$ has $n - m$ small roots and m large roots. Thus, the small roots (large roots, resp.) of $P(x)$ correspond to large roots (small roots, resp.) of $\tilde{P}(x)$. Furthermore, the inequality for $|\zeta|$ in* (10) *or* (12) *is invariant under the transformation $|\zeta| \mapsto e/|\zeta|$, hence the $n-m$ small roots and the m large roots of $\tilde{P}(x)$ are, respectively, upper-bounded by $R_{\rm in}$ and lower-bounded by $R_{\rm out}$.*

Remark 2 *If $m = 1$ then the formula in Theorem 1 can be strengthened as follows (note that $e = |a_0|$); for the proof, see [SI02] which shows numerically that the formula bounds the smallest root very sharply. (Precisely, the bound for large roots is not "$|\zeta| > R_{\rm out}$" but "$|\zeta| \ge R_{\rm out}$".)*

$$e \;<\; 3 - 2\sqrt{2} \simeq 0.172, \tag{14}$$

$$R_{\rm in(out)} \;=\; \frac{(1 + e) - (+)\sqrt{(1 + e)^2 - 8e}}{4}. \tag{15}$$

The smallest root $\hat{\zeta}$ and the other roots $\check{\zeta}$ satisfy inequalities $2|\hat{\zeta}|^2 - (1 + e)|\hat{\zeta}| + e > 0$ and $2|\check{\zeta}|^2 - (1 + e)|\check{\zeta}| + e \ge 0$, respectively.

The smallest root and other roots are also invertible, not exactly as in Remark 1 but with a small modification. The transformation $P(x) \mapsto \tilde{P}(x) = (x^n/e) \cdot P(e/x)$ gives us

$$\tilde{P}(x) = (e_0/e)x^n + x^{n-1} + (c_2 e)x^{n-2} + \cdots + (c_n e^{n-1}).$$

So long as $0 < e \ll 1$, $\tilde{P}(x)$ has one normal root $\tilde{\zeta}$, $|\tilde{\zeta}| \simeq 1$, and $n-1$ small roots $\hat{\zeta}, |\hat{\zeta}| \lesssim 2e$. Similar calculation as that in the proof in [SI02] shows that $\hat{\zeta}$ and $\check{\zeta}$ satisfy inequalities $|\hat{\zeta}|^2 - (1 + e)|\hat{\zeta}| + 2e > 0$ and $|\check{\zeta}|^2 - (1 + e)|\check{\zeta}| + 2e \ge 0$, respectively. Therefore, we have $|\hat{\zeta}| > 2R_{\rm out}$ and $|\check{\zeta}| \le 2R_{\rm in}$, with $R_{\rm in}$ and $R_{\rm out}$ given in (15). *Note that the condition for e is unchanged.*

Remark 3 *If $P(x)$ has m close roots around $x = a$ and $P(x)$ is not normalized as in (2) but we have*

$$P(x) = c_n(x-a)^n + \cdots + c_m(x-a)^m + c_{m-1}(x-a)^{m-1} + \cdots + c_0, \quad c_m \neq 0, \tag{16}$$

then we shift $x = a$ to the origin and make the scale transformation such that $P(x) \mapsto \tilde{P}(x) = P(\gamma(x+a))/\gamma^m$, where γ is a positive number. We set γ so that

$$\max\{|c_n|\gamma^{n-m}, \cdots, |c_{m+1}|\gamma\} = |c_m|$$
$$\Longrightarrow \gamma = \max_{m+1 \leq j \leq n} |c_j/c_m|^{1/(j-m)}.$$

Then, $\tilde{P}(x) = \tilde{c}_n x^n + \cdots + \tilde{c}_m x^m + (c_{m-1}/\gamma)x^{m-1} + \cdots + (c_0/\gamma^m)$, where $\tilde{c}_m = c_m$ and $\max\{|\tilde{c}_n|, \cdots, |\tilde{c}_{m+1}|\} = |\tilde{c}_m|$. Thus, we determine e as

$$e = \max\{|c_{m-1}/c_m\gamma|, \cdots, |c_0/c_m\gamma^m|^{1/m}\} = \beta/\gamma,$$
$$\beta = \max\{|c_{m-1}/c_m|, \cdots, |c_0/c_m|^{1/m}\}.$$

β and γ are expressed as follows ($P^{(j)}(x)$ is the j-th derivative of $P(x)$).

$$\beta = \max_{0 \leq j \leq m-1} \left| \frac{P^{(j)}(a)/j!}{P^{(m)}(a)/m!} \right|^{1/(m-j)}, \tag{17}$$

$$\gamma = \max_{m+1 \leq j \leq n} \left| \frac{P^{(j)}(a)/j!}{P^{(m)}(a)/m!} \right|^{1/(j-m)}. \tag{18}$$

These are the expressions given in [Yak00].

3 Numerical test of sharpness of the formula

In numerical analysis, the following theorem is very famous to bound the existence domains of the true roots by approximate roots computed numerically.

Theorem 2 (Smith 1970) *Let $G(x)$ be a monic univariate polynomial in $\mathbf{C}[x]$, of degree n, and let $\zeta_1, \ldots, \zeta_n$ be n distinct numbers in $\mathbf{C}$. Let n numbers $r_1, \ldots, r_n$ be defined as follows.*

$$r_i = \frac{n|G(\zeta_i)|}{|\prod_{j=1, \neq i}^{n}(\zeta_i - \zeta_j)|} \quad (i = 1, \ldots, n). \tag{19}$$

Let $D_1, \ldots, D_n$ be n discs in the complex plane, such that $\mathrm{center}(D_i) = \zeta_i$ and $\mathrm{radius}(D_i) = r_i$ $(i = 1, \ldots, n)$. Then, the union $D_1 \cup \cdots \cup D_n$ contains all the roots of $G(x)$. Furthermore, if a union $D_1 \cup \cdots \cup D_m$ is simply connected and does not intersect with $D_{m+1}, \ldots, D_n$, then the number of roots contained in this union is m.

We cannot apply this theorem directly to separate the close roots. However, if we interprete that $P(x) = P_0(x) + E(x)$, $P_0(x)$ has roots $\zeta_1, \ldots, \zeta_n$, and $E(x)$ is an error polynomial such that $\|E\| = O(\epsilon^m)$, then Smith's theorem allows us to bound the roots of $P(x)$. Therefore, we suppose that Smith's discs D_i $(i = 1, \ldots, n)$ are of the following radii.

$$S_i = \frac{n\epsilon^m}{|\prod_{j=1, \neq i}^{n}(\zeta_i - \zeta_j)|} \quad (i = 1, \ldots, n). \tag{20}$$

We have tested sharpness of our formula (3) and S_i above, as follows. For each tuple (n, m, ϵ), we construct 10 polynomials $P_1(x), \ldots, P_{10}(x)$ with real coefficients generated randomly, satisfying the conditions in (2). For each polynomial $P_i(x)$, $1 \leq i \leq 10$, we compute n roots, obtaining ζ_m and ζ_{m+1}. Then, we average $|\zeta_m|$ and $|\zeta_{m+1}|$ over 10 samples.

In the next three pages, we show n-, ϵ- and m-dependences of R_{in}, R_{out}, average($|\zeta_m|$), average($|\zeta_{m+1}|$), and average(S) where $S = \max\{|S_1|, \cdots, |S_m|\}$. (In the figures, $\odot$ shows average(S), $\bullet$ and $\circ$ show average($|\zeta_m|$) and average($|\zeta_{m+1}|$), respectively, $\top$ and $\bot$ show R_{in} and R_{out}, respectively.) We see that R_{in} and R_{out} bound, respectively, m small roots and other roots fairly well: average($|\zeta_m|$)$/R_{\mathrm{in}} \gtrsim 1/2$ and average($|\zeta_{m+1}|$)$/R_{\mathrm{out}} \lesssim 2$. On the other hand, the bound S is considerably larger than average($|\zeta_m|$) and it becomes larger and larger as n increases. This is because S_i is proportional to n.

References

[BCSS98] L. Blum, F. Cucker, M. Shub and S. Smale: *Complexity and Real Computation*, Section 8. Springer-Verlag, 1998.

[Mig91] M. Mignotte: *Mathematics for Computer Algebra*, Section 4. Springer-Verlag, 1991.

[SI02] T. Sasaki and D. Inaba: On analytic continuation of algebraic functions. Preprint of Univ. Tsukuba, 14 pages (2002).

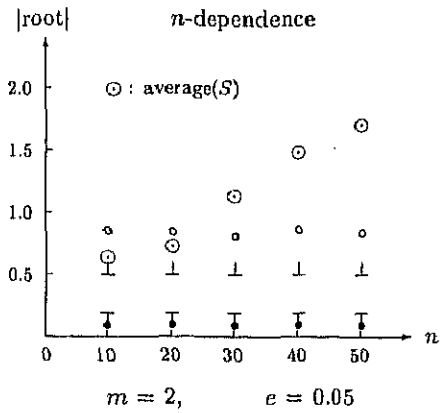
[Sma86] S. Smale: Newton's method estimates from data at one point. In R. Ewing, K. Gross, and C. Martin (Eds.), *The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics*. Springer-Verlag, 1986.

[Smi70] B.T. Smith: Error bounds for zeros of a polynomial based on Gerschgorin's theorems. J. ACM, **17**, 661–674 (1970).

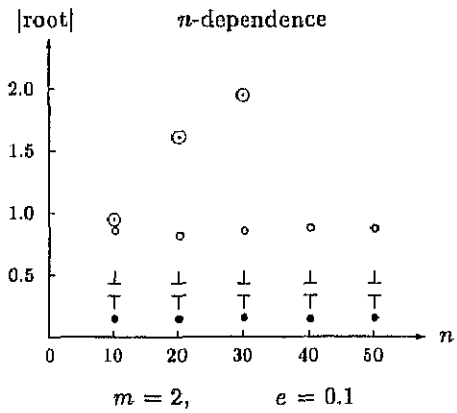
[TS00] A. Terui and T. Sasaki: "Approximate Zero-points" of Real Univariate Polynomial with Large Error Terms. IPSJ(Information Processing Society of Japan) Journal, **41**, 974–989, 2000.

[Yak00] J.C. Yakoubsohn: Finding a cluster of zeros of univariate polynomials. J. Complexity **16**, 603–636 (2000).

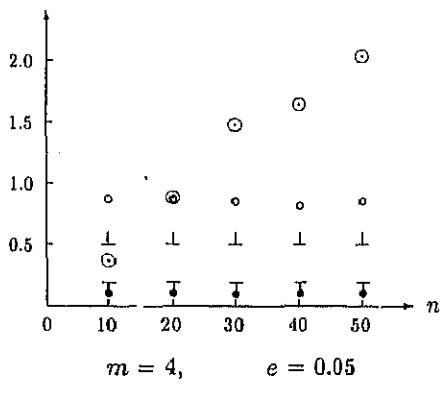
(continued to the next pages)



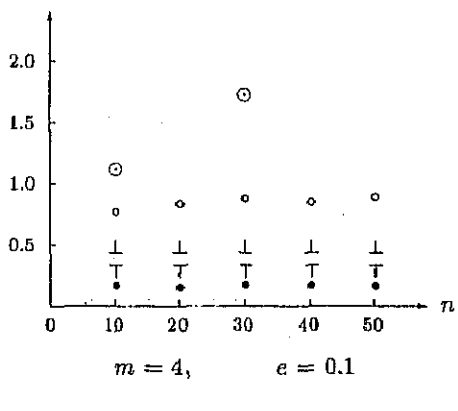
$m = 2, \quad e = 0.05$



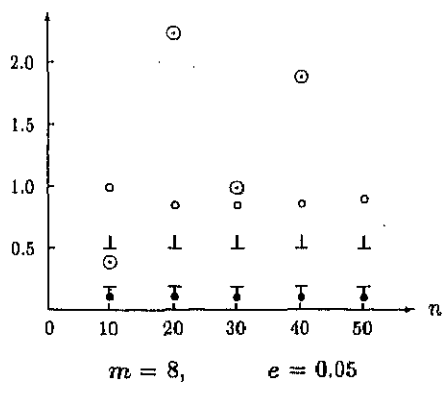
$m = 2, \quad e = 0.1$



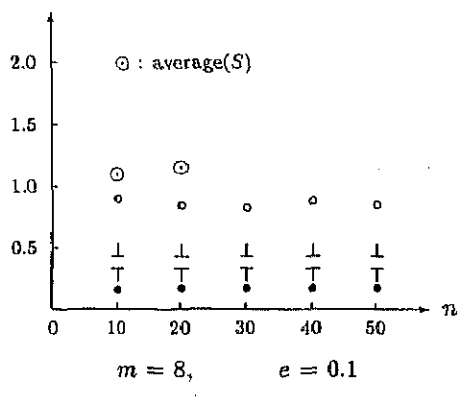
$m = 4, \quad e = 0.05$



$m = 4, \quad e = 0.1$



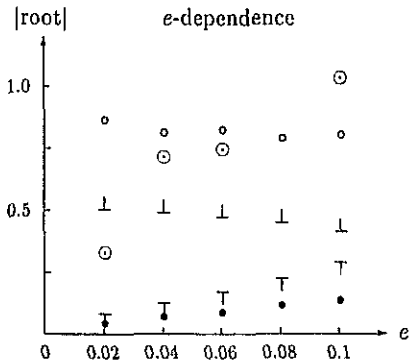
$m = 8, \quad e = 0.05$



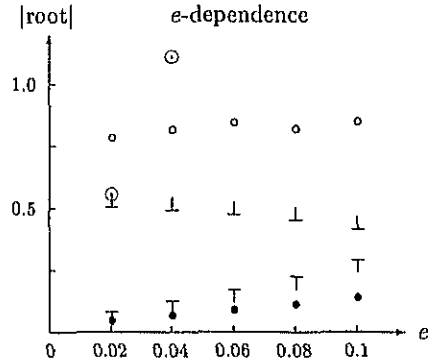
$m = 8, \quad e = 0.1$

$\perp$: R_{out} $\quad$ ∘ : average($|\zeta_{m+1}|$)
$\top$: R_{in} $\quad$ • : average($|\zeta_m|$)

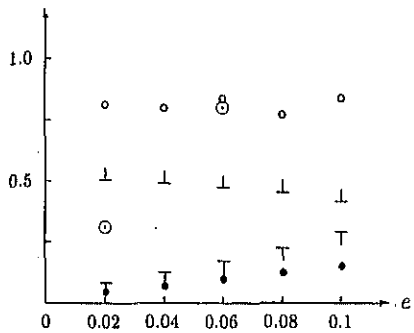
$\perp$: R_{out} $\quad$ ∘ : average($|\zeta_{m+1}|$)
$\top$: R_{in} $\quad$ • : average($|\zeta_m|$)



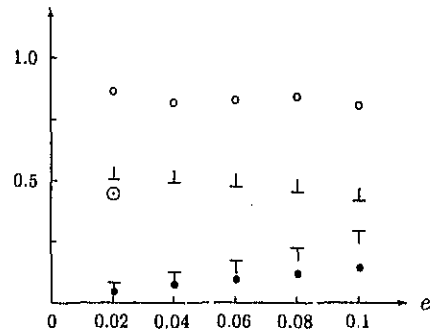
$m = 3, \quad n = 20$



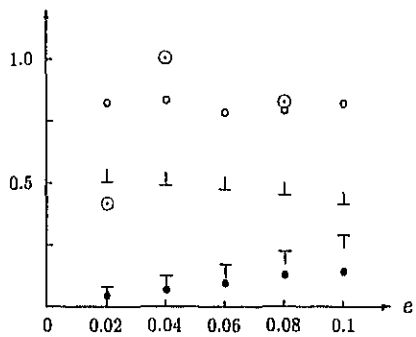
$m = 3, \quad n = 40$



$m = 6, \quad n = 20$

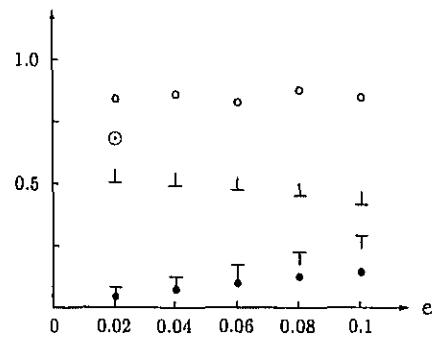


$m = 6, \quad n = 40$



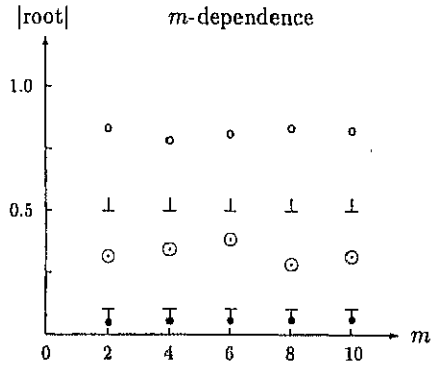
$m = 9, \quad n = 20$

$\perp$: R_{out} $\quad$ $\circ$: average($|\zeta_{m+1}|$)
$\top$: R_{in} $\quad$ $\bullet$: average($|\zeta_m|$)

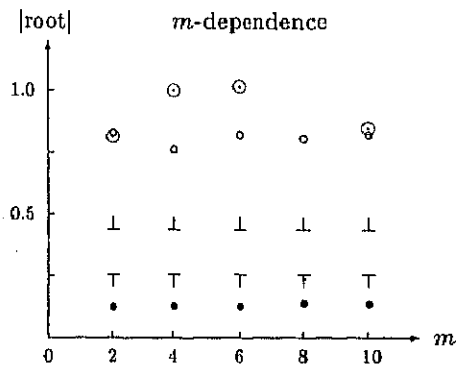


$m = 9, \quad n = 40$

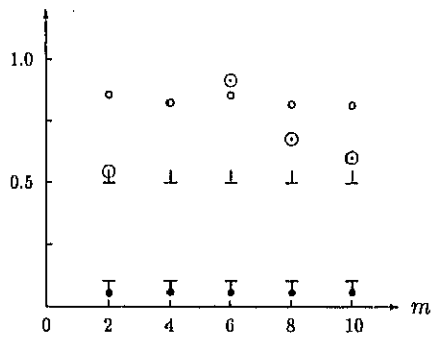
$\perp$: R_{out} $\quad$ $\circ$: average($|\zeta_{m+1}|$)
$\top$: R_{in} $\quad$ $\bullet$: average($|\zeta_m|$)



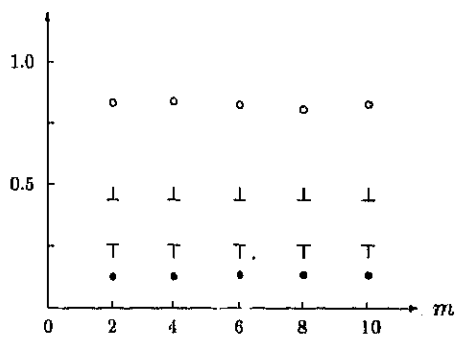
$n = 15, \quad e = 0.03$



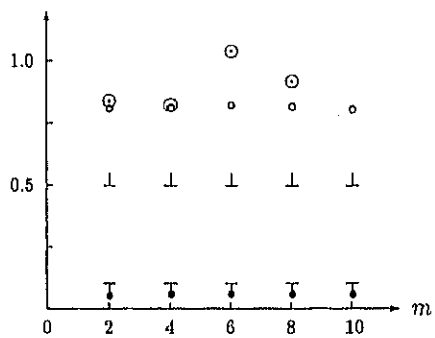
$n = 15, \quad e = 0.09$



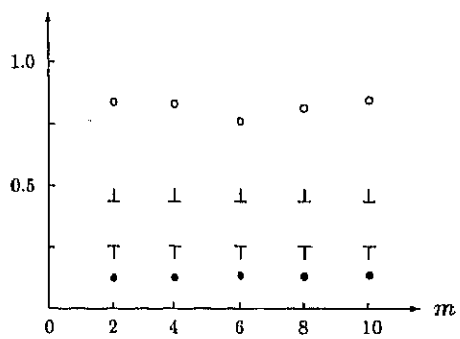
$n = 30, \quad e = 0.03$



$n = 30, \quad e = 0.09$



$n = 45, \quad e = 0.03$



$n = 45, \quad e = 0.09$

$\perp$: R_{out} $\quad$ o : average($|\zeta_{m+1}|$)
$\top$: R_{in} $\quad$ • : average($|\zeta_m|$)

$\perp$: R_{out} $\quad$ o : average($|\zeta_{m+1}|$)
$\top$: R_{in} $\quad$ • : average($|\zeta_m|$)

Approximate Multivariate Polynomial Factorization Based on Zero-Sum Relations *

Tateaki Sasaki

Institute of Mathematics, University of Tsukuba
Tsukuba-shi, Ibaraki 305, Japan
sasaki@math.tsukuba.ac.jp

Abstract

Conventional algorithms for approximate factorization of multivariate polynomial suffer from a dilemma: a polynomial-time algorithm which is based on zero-sum relations among power-series roots is practically very time-consuming and unstable, while practically stable algorithms are of combinatorial nature. In this paper, we present two ideas: one is a numeric matrix manipulation method to find zero-sum relations efficiently and the other is a method to utilize power-series roots expanded at different points. We analyze the methods theoretically and investigate their practicality by applying to several examples. We also discuss numerical stability of the matrix method.

Key words: algebraic-numeric computation, approximate algebra, approximate algebraic computation, approximate factorization, multivariate factorization, zero-sum relation. [1]

1 Introduction

Let $F(x,u)$ be a given multivariate polynomial in $\mathbf{C}[x,u] = \mathbf{C}[x,u_1,\ldots,u_\ell]$, with $\mathbf{C}$ the field of complex numbers, and let $G(x,u)$, $H(x,u)$ and $\Delta(x,u)$ be unknown polynomials. Let $\|F\|$ denote the norm of polynomial F; in this paper, we use the infinity norm, i.e., $\|F\|$ denotes the maximum of absolute numerical coefficients of F. If F is decomposed as

$$F = GH + \Delta, \quad \|\Delta\|/\|F\| = \varepsilon \ll 1, \tag{1.1}$$

then we say that F *is approximately factored to* G *and* H *at tolerance* ε. The approximate factorization is a natural extension of conventional polynomial factorization. For example, if we use floating-point numbers in algebraic computation, the polynomial factorization becomes approximate one with numerical errors $O(\varepsilon_{\rm M})$, where $\varepsilon_{\rm M}$ is the *machine epsilon* ($\varepsilon_{\rm M} \simeq 2\times 10^{-16}$ in many commercial machines). Note that the approximate factorization may be defined for much larger tolerance, such as $\varepsilon = 10^{-3}$.

Although the approximate factorization is not popular now, it will become an important operation in many application areas. This is because people in application areas often treat polynomials the coefficients of which are approximate numbers. Furthermore, the approximate factorization can be used in mathematics, such as for testing the absolute irreducibility of multivariate polynomial, see [9] and [1].

To author's knowledge, the concept of approximate factorization has appeared first in a paper on control theory [4]. The algorithm proposed in the paper is as follows: first, express G and H with unknown coefficients by fixing their terms, then determine the numerical coefficients so as to minimize $\|\Delta\|$. Huang et al. [2] pursuit this approach, but the algorithm seems to be rarely successful, unless G (or H) is a polynomial of several terms. Before [4], Kaltofen presented a modern algorithm for performing the absolute irreducible factorization [3], and he suggested to perform his algorithm by floating-point numbers, then the factor obtained is an approximate one.

In 1991, Sasaki et al. [9] proposed a modern algorithm: first, calculate the *power-series roots* $\varphi_1,\ldots,\varphi_n$ (truncated power series in the actual computation) such that $F(x,u) = (x-\varphi_1)\cdots(x-\varphi_n)$, then search for approximate factors by multiplying some of the linear factors $x-\varphi_1, \ldots, x-\varphi_n$. This algorithm is successful for polynomials of small degrees, such as $n \lesssim 5 \sim 8$, but it is quite time-consuming if $n \gtrsim 10$. Subsequently, Sasaki et al. [8] presented another algorithm which utilizes sums of powers of roots. For simplicity, consider the case of $\Delta = 0$. If $G = (x-\varphi_1)\cdots(x-\varphi_m)$, $m < n$, then we have the following relations (non-monic case will be described in 2)

$$\varphi_1^k + \cdots + \varphi_m^k = h_k(u) \in \mathbf{C}[u] \quad (k = 1,\ldots,m). \tag{1.2}$$

These relations give the following *zero-sum relations* (we have *approximate zero-sum relations* in the case of approximate factorization), where, by $\lfloor h \rfloor_d$ we denote the sum of terms of degrees $\geq d$, of polynomial $h(u) \in \mathbf{C}[u]$:

$$\left.\begin{array}{c} \lfloor \varphi_1^k + \cdots + \varphi_m^k \rfloor_{d_k+1} = 0 \\ d_k = \deg(h_k) \end{array}\right\} \quad (k = 1,\ldots,m). \tag{1.3}$$

Conversely, by searching for the above kind of zero-sum relations, we can find $\{\varphi_{i_1},\ldots,\varphi_{i_m}\}$ such that $(x-\varphi_{i_1})\cdots(x-\varphi_{i_m})$ is a polynomial factor of F. The algorithm described

*Work supported in part by Japanese Ministry of Education, Science and Culture under Grants 09308008.

ISSAC 2001, UWO, Canada
 1-58113-218-2/ 00/ 0008 $5.00

in [8] is incomplete in that the truncation degree of power series was wrongly given, and Nagasaka and Sasaki [5] improved the algorithm and showed that the time-complexity of the algorithm is $O(n^{2\ell+3})$.

The zero-sum relations in (1.3) are quite effective for determining approximate factors. In fact, Galligo and Watt [1] showed that only zero-sum relations with $k = 1$ and 2 determine all the approximate factors in most cases. (In [1], authors did not mention on the zero-sum relations in (1.3) but utilized the relation $\lfloor \varphi_1\varphi_2 + \cdots + \varphi_1\varphi_m + \cdots + \varphi_{m-1}\varphi_m \rfloor_{d_2+1} = 0$).

In spite of the advancements mentioned above, we have still many practical problems. Two big problems are

Problem A: how to decrease the computation time ?

Problem B: how to control the numerical errors ?

As for Problem A, so long as the zero-sum relations in (1.3) are used, the asymptotic complexity is polynomial-time w.r.t. $\deg_x(F)$. However, computation based on (1.3) is practically very time-consuming. If we utilize only zero-sum relations of small powers, such as $k = 1$ or 2, we can determine the approximate factors rather quickly for polynomials of degrees $\leq 20 \sim 30$, as [1] demonstrates. However, we must check a combinatorial number of possibilities, hence we meet the knapsack problem. As for Problem B, power-series roots computed numerically contain sometimes very large errors, as pointed out by [10] and [7]. Furthermore, power-series roots often show pathological features, as an example in 6 shows. If we treat such pathological roots, the algorithm often becomes very unstable.

In this paper, we present two methods to improve the approximate factorization algorithm based on zero-sum relations. The first one is to find as many zero-sum relations as possible by $O(n^3)$ matrix operations. The second one is to utilize the power-series roots expanded at different points. In 2, we explain zero-sum relations among power-series roots and survey underlying theory. In 3, we present the matrix manipulation method to find zero-sum relations. In 4, we describe how to use the power-series roots expanded at different points. In 5, we investigate approximate zero-sum relations which are caused by the "perturbation term" Δ. In 6, we investigate our method empirically by applying to bivariate polynomials of degrees 10 and 20.

2 Zero-sum relations among power-series roots

In this and the next section, we consider only exact zero-sum relations, and approximate zero-sum relations corresponding to approximate factors will be considered in 4.

Let $F(x, u_1, \ldots, u_\ell)$ be a given multivariate square-free polynomial over $\mathbf{C}$, the field of complex numbers, with $\deg_x(F) = n$. Abbreviating $(u_1, \ldots, u_\ell)$ to (u), we express $F(x, u)$ as

$$F(x,u) = f_n(u)x^n + f_{n-1}(u)x^{n-1} + \cdots + f_0(u). \qquad (2.1)$$

By $\mathrm{tdeg}(f)$, with $f(u) \in \mathbf{C}[u]$, we denote the *total-degree* w.r.t. $u_1, \ldots, u_\ell$, of f: if $T = cu_1^{a_1}\cdots u_\ell^{a_\ell}$, $c \in \mathbf{C}$, then $\mathrm{tdeg}(T) = a_1 + \cdots + a_\ell$, and $\mathrm{tdeg}(f)$ is the maximum of the total-degrees of the terms of $f(u)$. By $\lfloor g \rfloor_e$ and $\lceil g \rceil^e$, with $g(v)$ a power-series in $v_1, \ldots, v_\ell$ and e an integer, we denote the sum of all the terms of total-degrees $\geq e$ and total-degrees $\leq e$, respectively, of $g(v)$.

Let $(s) \stackrel{\mathrm{def}}{=} (s_1, \ldots, s_\ell) \in \mathbf{C}^\ell$, and assume that $F(x, s)$ is square-free (since $F(x, u)$ is square-free, $F(x, s)$ is square-free for almost all (s)). Let the roots of $F(x, u)$ w.r.t. the main variable x be $\bar{\varphi}_1(u), \ldots, \bar{\varphi}_n(u)$ which are algebraic functions in general:

$$F(x,u) = f_n(u)\cdot(x - \bar{\varphi}_1(u))\cdots(x - \bar{\varphi}_n(u)).$$

Let $\alpha_1, \ldots, \alpha_n$ be the roots of $F(x, s)$, where $\alpha_i \neq \alpha_j$ ($\forall i \neq j$) by assumption. Let $(v_1, \ldots, v_\ell) \stackrel{\mathrm{def}}{=} (u_1 - s_1, \ldots, u_\ell - s_\ell)$ be a set of new sub-variables, then the roots $\bar{\varphi}_i(s+v)$ can be expanded into (infinite) power series $\varphi_i(v)$ in the variables $v_1, \ldots, v_\ell$.

Suppose that $F(x, u)$ is factored in $\mathbf{C}[x, u]$ as $F(x, u) = G(x, u)H(x, u)$, where

$$\begin{cases} G(x,u) = g_m(u)x^m + g_{m-1}(u)x^{m-1} + \cdots + g_0(u), \\ H(x,u) = h_{n-m}(u)x^{n-m} + h_{n-m-1}(u)x^{n-m-1} \\ \qquad\qquad + \cdots + h_0(u). \end{cases} \qquad (2.2)$$

Putting

$$e_i = \mathrm{tdeg}(f_i) \quad (i = n, n-1, \ldots, 0), \qquad (2.3)$$

we first consider the degree bounds for $g_m, \ldots, g_0$.

Proposition 1 (Nagasaka-Sasaki 98) *Let $\mathcal{D}$ be the smallest two-dimensional convex hull containing $n+2$ points $(0,0)$, $(0,e_0)$, $(1,e_1)$, $\cdots$, (n,e_n) and $(n,0)$. Let $F(x,u) = G(x,u)H(x,u)$, where F and G, H are given by (2.1) and (2.2), respectively. For any non-negative integers i and j satisfying $i + j \leq n$, point $(i+j, \mathrm{tdeg}(g_i h_j))$ is not plotted outside $\mathcal{D}$.*

Corollary 1 *Let the $n+1$ points $(0,\bar{e}_0)$, $(1,\bar{e}_1)$, $\cdots$, $(n,\bar{e}_n)$ be on the upper edges of $\mathcal{D}$, hence $\bar{e}_0 = e_0$ and $\bar{e}_n = e_n$ ($\bar{e}_i$ is a rational number). Then, we have*

$$\mathrm{tdeg}((f_n/g_m)g_{m-i}) = \mathrm{tdeg}(h_{n-m}g_{m-i}) \leq \bar{e}_{n-i},$$
$$\bar{e}_{n-i} \leq e_n + i(\bar{e}_{n-1} - e_n) = i\bar{e}_{n-1} - (i-1)e_n.$$

Throughout this paper, w.l.o.g., we assume that

$$G(x,u) = g_m(u)\cdot(x - \bar{\varphi}_1(u))\cdots(x - \bar{\varphi}_m(u)), \quad m \geq 2.$$

Then, we have

$$\begin{cases} g_m\cdot(\varphi_1 + \varphi_2 + \cdots + \varphi_m) = -g_{m-1}, \\ g_m\cdot(\varphi_1\varphi_2 + \cdots + \varphi_1\varphi_m + \cdots + \varphi_{m-1}\varphi_m) = g_{m-2}, \\ \qquad\qquad \cdots \\ g_m\cdot(\varphi_1\varphi_2\cdots\varphi_m) = (-1)^m g_0. \end{cases} \qquad (2.4)$$

Therefore, since $\mathrm{tdeg}_u(f(u)) = \mathrm{tdeg}_v(f(s+v))$, we have the following zero-sum relations.

$$\begin{cases} \lfloor g_m\cdot(\varphi_1 + \varphi_2 + \cdots + \varphi_m)\rfloor_{\bar{e}_{n-1}+1} = 0, \\ \lfloor g_m\cdot(\varphi_1\varphi_2 + \cdots + \varphi_1\varphi_m + \cdots + \varphi_{m-1}\varphi_m)\rfloor_{\bar{e}_{n-2}+1} = 0, \\ \qquad\qquad \cdots \\ \lfloor g_m\cdot(\varphi_1\varphi_2\cdots\varphi_m)\rfloor_{\bar{e}_{n-m}+1} = 0. \end{cases}$$

Here, g_m denotes $g_m(s+v)$.

The above zero-sum relations, except for the first, are rather complicated to treat, so [8] proposed to convert them to zero-sum relations which are easier to treat. Let $S_k(X_1, \ldots, X_m)$ and $P_k(X_1, \ldots, X_m)$ ($k = 1, 2, \ldots, n$) be

the elementary symmetric polynomials and the kth power sum, respectively, of $X_1, \ldots, X_m$:

$$\left\{ \begin{array}{l} S_k = \sum_{(i_1,\ldots,i_k)} X_{i_1} \cdots X_{i_k} \quad (k \le m), \\ S_k = 0 \quad (k > m), \\ P_k = X_1^k + \cdots + X_m^k \quad (1 \le k \le n). \end{array} \right. \tag{2.5}$$

For each i, $1 \le i \le n$, we have (Newton's formula)

$$P_i - P_{i-1}S_1 + P_{i-2}S_2 - \cdots + (-1)^{i-1} P_1 S_{i-1} + (-1)^i i S_i = 0 \tag{2.6}$$

This formula, with $X_j = g_m \varphi_j$ $(j = 1, \ldots, m)$, allows us to convert system (2.4) to the following system.

$$\left\{ \begin{array}{rcl} g_m \cdot (\varphi_1 + \cdots + \varphi_m) & = & -g_{m-1}, \\ g_m^2 \cdot (\varphi_1^2 + \cdots + \varphi_m^2) & = & g_{m-1}^2 - 2 g_m g_{m-2}, \\ \cdots & & \cdots \\ g_m^m \cdot (\varphi_1^m + \cdots + \varphi_m^m) & = & (-1)^m \{ g_{m-1}^m - \\ & & m g_m g_{m-1}^{m-2} g_{m-2} + \cdots - (-1)^m m g_m^{m-1} g_0 \}. \end{array} \right. \tag{2.7}$$

Newton's formula and Corollary 1 allow us to bound the total-degrees of the above r.h.s. expressions as follows.

Proposition 2 (Nagasaka-Sasaki 98) *Let*

$$\bar{e} = \bar{e}_{n-1}. \tag{2.8}$$

Then, we have the following total-degree bounds.

$$\left\{ \begin{array}{rcl} \mathrm{tdeg}[(f_n/g_m)(g_{m-1})] & \le & 1\bar{e}, \\ \mathrm{tdeg}[(f_n/g_m)^2 (g_{m-1}^2 - 2 g_m g_{m-2})] & \le & 2\bar{e}, \\ \cdots & & \cdots \\ \mathrm{tdeg}[(f_n/g_m)^m (g_{m-1}^m - m g_m g_{m-1}^{m-2} g_{m-2} & & \\ \quad + \cdots - (-1)^m m g_m^{m-1} g_0)] & \le & m\bar{e}. \end{array} \right.$$

Thus, we have the following zero-sum relations.

$$\left\{ \begin{array}{rcl} \lfloor f_n \cdot (\varphi_1 + \varphi_2 + \cdots + \varphi_m) \rfloor_{\bar{e}+1} & = & 0, \\ \lfloor f_n^2 \cdot (\varphi_1^2 + \varphi_2^2 + \cdots + \varphi_m^2) \rfloor_{2\bar{e}+1} & = & 0, \\ \cdots & & \cdots \\ \lfloor f_n^m \cdot (\varphi_1^m + \varphi_2^m + \cdots + \varphi_m^m) \rfloor_{m\bar{e}+1} & = & 0. \end{array} \right. \tag{2.9}$$

Here, we have multiplied $f_n, \ldots, f_n^m$ instead of $g_m, \ldots, g_m^m$, respectively. The reason is that we do not know g_m until the factorization finishes, therefore the zero-sum relations in (2.7) cannot be used in the algorithm.

In [8] and [5], the authors proposed to utilize the following zero-sum relations, where $c_1, \ldots, c_n \in \mathbf{C}$.

$$\left\{ \begin{array}{rcl} \lfloor f_n \cdot (c_1 \varphi_1^1 + \cdots + c_n \varphi_n^1) \rfloor_{1\bar{e}+1} & = & 0, \\ \cdots & & \\ \lfloor f_n^m \cdot (c_1 \varphi_1^m + \cdots + c_n \varphi_n^m) \rfloor_{m\bar{e}+1} & = & 0. \end{array} \right. \tag{2.10}$$

The method in this paper is based on the following theorem which is simpler than the main theorem in [8].

Theorem 1 *Let $h_1(u), \ldots, h_m(u) \in \mathbf{C}[u]$, and suppose that we have the relations*

$$\left\{ \begin{array}{l} \lceil f_n \cdot (\varphi_1^1 + \cdots + \varphi_m^1) \rceil^{n\bar{e}} = h_1(u), \quad \mathrm{tdeg}(h_1) \le 1\bar{e}, \\ \cdots \\ \lceil f_n^m \cdot (\varphi_1^m + \cdots + \varphi_m^m) \rceil^{n\bar{e}} = h_m(u), \quad \mathrm{tdeg}(h_m) \le m\bar{e}. \end{array} \right. \tag{2.11}$$

Then, $\tilde{G} = f_n \cdot (x - \varphi_1) \cdots (x - \varphi_m)$ is a polynomial factor of $f_n F(x, u)$, and vice versa.

Proof. Consider the elementary symmetric polynomial S_k and the kth power sum P_k in (2.5). By Newton's formula (2.6) and Corollary 1, we obtain

$$\left. \begin{array}{c} P_k(f_n\varphi_1, \ldots, f_n\varphi_m) = h_k(u) \in \mathbf{C}[u] \\ \mathrm{tdeg}(h_k) \le k\bar{e} \end{array} \right\} \quad \text{for } k > m.$$

Put $G' = \lceil f_n^m (x - \varphi_1) \cdots (x - \varphi_m) \rceil^{n\bar{e}}$ and $H' = \lceil f_n^{n-m} (x - \varphi_{m+1}) \cdots (x - \varphi_n) \rceil^{n\bar{e}}$, then Newton's formula and the degree bounds for $P_1, \ldots, P_n$ lead us to

$$\mathrm{tdeg}_v(G') \le m\bar{e}, \quad \mathrm{tdeg}_v(H') \le (n-m)\bar{e}.$$

Furthermore, Corollary 1 gives us $\mathrm{tdeg}_u(f_n^{n-1}F) \le n\bar{e}$. Therefore, G' and H' must be polynomial factors of $f_n^{n-1}F$. The converse is easy. □

This theorem shows that we can find all the irreducible factors by finding zero-sum relations among the coefficient vectors of power-series roots. However, we must compute the power-series roots up to quite a high power.

3 Finding zero-sum relations

The principle of our factorization is to find subsets of $\{\lceil \varphi_1 \rceil^{n\bar{e}}, \ldots, \lceil \varphi_n \rceil^{n\bar{e}}\}$, that satisfy the zero-sum relations in (2.9). The zero-sum relations are for the (truncated) power series, but they can be converted easily to zero-sum relations for numerical vectors, where each vector is formed by taking out numerical coefficients of each power series.

Let $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \cdots, a_{in'})$, $i = 1, \ldots, n$, be n'-dimensional vectors over $\mathbf{C}$, where $\boldsymbol{a}_i$ is formed by coefficients of $\varphi_i(v), \varphi_i^2(v), \cdots$. Therefore, we have

$$\boldsymbol{a}_1 + \boldsymbol{a}_2 + \cdots + \boldsymbol{a}_n = 0. \tag{3.1}$$

We assume that $n \le n' \le n\bar{e}$.

Let M be an $n \times n'$ matrix, with rows $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$:

$$M = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n'} \\ a_{21} & a_{22} & \cdots & a_{2n'} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn'} \end{pmatrix} \begin{array}{l} \longleftarrow \varphi_1 \\ \longleftarrow \varphi_2 \\ \quad \vdots \\ \longleftarrow \varphi_n \end{array} \tag{3.2}$$

By applying the algorithm Elimi given below, we first convert the matrix M to $\tilde{M}$ which is of the following form, where $\star$ denotes either 0 or a nonzero element.

$$\tilde{M} = \begin{pmatrix} \tilde{a}_{11} & 0 & \cdots & 0 & 0 & \cdots \\ 0 & \tilde{a}_{22} & \ddots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & \cdots \\ 0 & \cdots & 0 & \tilde{a}_{mm} & 0 & \cdots \\ \star & \star & \cdots & \star & \vdots & \vdots \\ \star & \star & \cdots & \star & 0 & \cdots \end{pmatrix} \tag{3.3}$$

Algorithm Elimi(M): % Eliminate rows of M %

E-0: $i := 1$ and go to **E-3**.

E-1: If all the ith, $(i+1)$th, ..., n'th columns are zero columns then **stop**.

E-2: If the ith row is such that $(\tilde{a}_{i1}, \cdots, \tilde{a}_{i,i-1}, 0, \cdots, 0)$ then exchange the ith row for the jth row, $j > i$, such that $\tilde{a}_{ij} \neq 0$.

E-3: If the (i,i) element is zero then exchange the ith column for the jth column, $j > i$, such that $\tilde{a}_{ij} \neq 0$.

E-4: (Now, $\tilde{a}_{ii} \neq 0$.) Eliminate the ith row by the ith column, so that the ith row becomes $(\cdots, 0, \tilde{a}_{ii}, 0, \cdots)$ after the elimination (only the ith to nth rows are changed by this elimination);
set $i := i + 1$ and go to **E-1**. □

Note that this elimination preserves the zero-sum relations: if $a_{i_1} + \cdots + a_{i_m} = 0$ in M then we have $\tilde{a}_{i_1} + \cdots + \tilde{a}_{i_m} = 0$ in $\tilde{M}$, where each $\tilde{a}_i$ is a row corresponding to a_i.

Relation (3.1) tells us the following properties of $\tilde{M}$.

Property 1. *Let the j-th column of $\tilde{M}$, $j \leq m$, be $(\cdots, 0, \tilde{a}_{jj}, 0, \cdots, 0, \tilde{a}_{m+1,j}, \cdots, \tilde{a}_{nj})^t$, then*

$$\tilde{a}_{m+1,j} + \cdots + \tilde{a}_{nj} = -\tilde{a}_{jj}. \tag{3.4}$$

Property 2. *If $\tilde{a}_{i_1} + \cdots + \tilde{a}_{i_r} + \tilde{a}_{i_s} = 0$, with $i_1 < \cdots < i_r \leq m < i_s$, hence the i_1th, ..., i_rth rows have been eliminated by algorithm* **Elimi**, *then the i_sth row of $\tilde{M}$ is such that*

$$(\cdots, 0, -\tilde{a}_{i_1 i_1}, \star, \cdots, \star, -\tilde{a}_{i_r i_r}, 0, \cdots), \tag{3.5}$$

where $\star$ denotes either 0 or $-\tilde{a}_{jj}$, $i_1 < j < i_r$.

The above i_1th, ..., i_rth rows and i_sth row form, if collected together, the following matrix which we call *zero-sum cell* (we have discarded zero columns).

$$\begin{pmatrix} \tilde{a}_{i_1 i_1} & 0 & \cdots & 0 \\ 0 & \tilde{a}_{i_2 i_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{a}_{i_r i_r} \\ -\tilde{a}_{i_2 i_1} & -\tilde{a}_{i_2 i_2} & \cdots & -\tilde{a}_{i_r i_r} \end{pmatrix} \tag{3.6}$$

Let us investigate the matrix $\tilde{M}$ further.

Definition 1 (minimal zero-sum relation) *A zero-sum relation $a_{k_1} + \cdots + a_{k_t} = 0$ is called minimal if it cannot be divided into two zero-sum relations $a_{i_1} + \cdots + a_{i_r} = 0$ and $a_{j_1} + \cdots + a_{j_s} = 0$, where $t = r + s$ and $\{i_1, \cdots, i_r\} \cup \{j_1, \cdots, j_s\} = \{k_1, \cdots, k_t\}$.*

Property 3. *Each zero-sum cell in $\tilde{M}$ corresponds to a minimal zero-sum relation.*

Proof. Let $R_1 : \tilde{a}_{i_1} + \cdots + \tilde{a}_{i_r} = 0$, $R_2 : \tilde{a}_{j_1} + \cdots + \tilde{a}_{j_s} = 0$, and suppose that i_1th, ..., i_rth rows form a zero-sum cell, and that j_1th, ..., j_sth rows form another zero-sum cell. Here, some rows may be shared by these zero-sum cells, and the i_rth and j_sth rows correspond to some of the bottom rows in (3.6). Note that $i_r \neq j_s$, because these two zero-sum cells are different. For convenience, we put $I = \{i_1, \cdots, i_r\}$ and $J = \{j_1, \cdots, j_s\}$.

Case 1: $I \subset J$. Since $i_1 \in \{j_1, \cdots, j_{s-1}\}$, the i_1th column, for example, contains two $-\tilde{a}_{i_1 i_1}$ elements. On the other hand, let $\{k_1, \cdots, k_{n-s}\} = \{1, 2, \cdots, n\} \setminus \{j_1, \cdots, j_s\}$, then we have $\tilde{a}_{k_1} + \cdots + \tilde{a}_{k_{n-s}} = 0$. Therefore, adding the elements of the i_1th column, we have $\tilde{a}_{i_1 i_1} - \tilde{a}_{i_1 i_1} - \tilde{a}_{i_1 i_1} \neq 0$, which contradicts (3.1).

Case 2: $I \cap J \neq \phi$, $I \not\subset J$ and $I \not\supset J$. Let $\{k_1, \cdots, k_t\} = \{i_1, \cdots, i_r\} \cap \{j_1, \cdots, j_s\}$. Then, the above two zero-sum relations give another zero-sum relation $R_3 : \tilde{a}_{k_1} + \cdots + \tilde{a}_{k_t} = 0$ the vectors of which are included in both R_1 and R_2. Therefore, the Case 1 shows that this case should not occur, too. □

Remark 1 *In the case of $\{i_1, \cdots, i_r\} \subset \{j_1, \cdots, j_s\}$, let $R'_2 : \tilde{a}_{k_1} + \cdots + \tilde{a}_{k_{s-r}} = 0$, with $\{k_1, \cdots, k_{s-r}\} = \{j_1, \cdots, j_s\} \setminus \{i_1, \cdots, i_r\}$. If R_1 and R'_2 are minimal then two zero-sum cells corresponding to R_1 and R'_2 will be formed.*

Remark 2 *By virtue of* **Property 3**, *we can remove the rows contained in the zero-sum cell and continue the elimination recursively.*

The second step of zero-sum relation finding is to find zero-sum cells by exchanging rows and columns of $\tilde{M}$. The algorithm is as follows.

Algorithm findCell($\tilde{M}$): % Find zero-sum cells %

C-0: $i := m + 1$.

C-1: If the ith row of $\tilde{M}$ is not of the form $(\cdots, 0, -\tilde{a}_{rr}, \star, \cdots, \star, -\tilde{a}_{ss}, 0, \cdots)$, where $\star$ is either 0 or $-\tilde{a}_{jj}$, $r < j < s$, then go to **C-3** by setting $i := i + 1$.

C-2: (Here, we have found a zero-sum cell.) Save the list $(r, \star, \cdots, \star, s, i)$ into **ZSCells**. Remove the rth, $\star$th, ..., $\star$th, sth, and ith rows from $\tilde{M}$, and go to **C-3** by setting $n := n - 1$.

C-3: If $i > n$ then **stop**, else go to **C-1**. □

Finally, if we found zero-sum cells then remove the rows contained in the zero-sum cells and apply the algorithms **Elimi** and **findCell** recursively to the remaining rows.

Example 1 Finding the zero-sum relations.

$$\begin{aligned} F \;=\; & [(x-2+u)^3 - (1+u+u^2+u^3)] \\ & \times [(x+2-u)^3 + (1+u+u^2+u^3)]. \end{aligned}$$

Six power-series roots w.r.t. x are as follows:

$$\begin{cases} \varphi_i = 2 - u + \omega^{i-1}\varphi(u), & i = 1, 2, 3, \\ \varphi_i = u - 2 - \omega^{i-4}\varphi(u), & i = 4, 5, 6, \end{cases}$$

where ω is a primitive cube root of 1 ($\omega^2 + \omega + 1 = 0$) and

$$\varphi(u) = 1 + 1/3u + 2/9u^2 + 14/81u^3 - 46/243u^4 + \cdots.$$

We have many zero-sum relations among the roots: $\lfloor \varphi_1 + \varphi_4 \rfloor_0 = 0$, $\lfloor \varphi_2 + \varphi_5 \rfloor_0 = 0$, $\lfloor \varphi_3 + \varphi_6 \rfloor_0 = 0$, $\lfloor \varphi_1^3 + \varphi_4^3 \rfloor_0 = 0$, $\lfloor \varphi_2^3 + \varphi_5^3 \rfloor_0 = 0$, $\lfloor \varphi_3^3 + \varphi_6^3 \rfloor_0 = 0$, $\lfloor \varphi_1 + \varphi_2 + \varphi_3 \rfloor_2 = 0$, $\lfloor \varphi_4 + \varphi_5 + \varphi_6 \rfloor_2 = 0$, $\lfloor \varphi_1^2 + \varphi_2^2 + \varphi_3^2 \rfloor_3 = 0$, $\lfloor \varphi_4^2 + \varphi_5^2 + \varphi_6^2 \rfloor_3 = 0$, etc. Among these, only the last four zero-sum relations correspond to the polynomial factors of $F(x, u)$.

Let M be the following 6×6 matrix, where the 1st to 6th columns represent the coefficients of u^3-terms, u^4-terms

and u^5-terms of $\varphi_i(u)$, and u^3-terms, u^4-terms and u^5-terms of $\varphi_i(u)^2$, respectively.

$$M = \begin{pmatrix} \frac{14}{81} & -\frac{46}{243} & \cdots \\ \frac{14}{81}\omega & -\frac{46}{243}\omega & \cdots \\ -\frac{14}{81}(\omega+1) & \frac{46}{243}(\omega+1) & \cdots \\ -\frac{14}{81} & \frac{46}{243} & \cdots \\ -\frac{14}{81}\omega & \frac{46}{243}\omega & \cdots \\ \frac{14}{81}(\omega+1) & -\frac{46}{243}(\omega+1) & \cdots \end{pmatrix} \begin{matrix} \leftarrow \varphi_1, \varphi_1^2 \\ \leftarrow \varphi_2, \varphi_2^2 \\ \leftarrow \varphi_3, \varphi_3^2 \\ \leftarrow \varphi_4, \varphi_4^2 \\ \leftarrow \varphi_5, \varphi_5^2 \\ \leftarrow \varphi_6, \varphi_6^2 \end{matrix}$$

Applying the algorithm **Elimi**, we obtain the following matrix $\tilde{M}$ (we give the result without showing the exchange of rows and columns).

$$\tilde{M} = \begin{pmatrix} \frac{14}{81} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{80}{81}(\omega+1) & 0 & 0 \\ -\frac{14}{81} & 0 & 0 & \frac{80}{81}(\omega+1) & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{484}{243} & 0 \\ -\frac{14}{81}\omega & 0 & 0 & -\frac{80}{81}(\omega+1) & -\frac{484}{243}\omega & 0 \\ \frac{14}{81}\omega & 0 & 0 & \frac{80}{81}(\omega+1) & \frac{484}{243}(\omega+1) & 0 \end{pmatrix}$$

$\tilde{M}$ contains one zero-sum cell. Removing the 1st to 3rd rows, deleting the zero columns, reorder the columns, and applying the algorithm **Elimi**, we obtain the following result.

$$\begin{pmatrix} -\frac{484}{243} & 0 & 0 \\ -\frac{484}{243}\omega & -\frac{14}{81}\omega & -\frac{80}{81}(\omega+1) \\ \frac{484}{243}(\omega+1) & \frac{14}{81}\omega & \frac{80}{81}(\omega+1) \end{pmatrix} \Longrightarrow \begin{pmatrix} -\frac{484}{243} & 0 & 0 \\ 0 & -\frac{14}{81}\omega & 0 \\ \frac{484}{243} & \frac{14}{81}\omega & 0 \end{pmatrix}$$

Thus, we have found two zero-sum relations $a_1+a_2+a_3=0$ and $a_4+a_5+a_6=0$. □

If we form M by the coefficients of terms of total-degrees *up to* $n\bar{e}$, *of* $\varphi_{i_1}\cdots\varphi_{i_m}^m$ $(i=1,\ldots,n)$, *then we can find* the zero-sum relations corresponding to all the irreducible factors of F, as Theorem 1 assures. However, we usually do not compute the power-series roots up to so high power, then the above algorithm cannot always find all the necessary zero-sum relations and we still suffer from the knapsack problem. However, the above algorithm often find many zero-sum relations. The following proposition clarifies how many zero-sum relations the above algorithm finds.

Proposition 3 *Let ν be the number of irreducible factors of $F(x,u)$, over* C, *and let $n-\mu$ be* colrank(M), *i.e., column rank of M (the number of linearly independent columns of M). The algorithm* **Elimi** *finds* $\min\{\mu,\nu\}$ *zero-sum cells at most and* $\max\{\nu-\lfloor\mu/2\rfloor, 0\}$ *zero-sum cells at least, where* $[\cdot]$ *denotes Gauss' symbol.*

Proof. By assumption, $\tilde{M}$ contains $m=n-\mu$ top diagonal elements. Let us consider the bottom μ rows of $\tilde{M}$. If all of these rows correspond to the bottom rows of zero-sum cells, then $\tilde{M}$ contains μ zero-sum cells. Hence, $\tilde{M}$ can contain $\min\{\mu,\nu\}$ zero-sum cells, at most. On the other hand, if two vectors of a zero-sum relation $R: a_{i_1}+\cdots+a_{i_r}=0$ are contained in the bottom μ rows of $\tilde{M}$, *then* $\tilde{M}$ *does* not contain a zero-sum cell corresponding to R. Hence, $\tilde{M}$ contains $\min\{\nu-\lfloor\mu/2\rfloor, 0\}$ zero-sum cells at least. □

4 Roots expanded at different points

As we will see in 6, some power-series roots may have very large numerical coefficients at high powers, causing the computation very unstable. Hence, we want to avoid computing the roots to a high power. We show in this section that computing the power-series roots at several different points is equivalent to computing the roots to a high power at a single expansion point.

Suppose we expand the roots at P different points $(s^{(1)}),\ldots,(s^{(P)})$, $P\geq 2$:

$$(s^{(p)}) \stackrel{\text{def}}{=} (s_1^{(p)},\cdots,s_\ell^{(p)}) \in \mathbf{C}^\ell, \quad p=1,\ldots,P,$$

where we assume that each $F(x,s^{(p)})$ is square-free. Let the roots of $F(x,s^{(p)})$ be $\alpha_1^{(p)},\ldots,\alpha_n^{(p)}$ and the power-series roots expanded at $(s^{(p)})$ be $\varphi_1^{(p)}(v),\ldots,\varphi_n^{(p)}(v)$:

$$F(x,s^{(p)}+v) = f_n(s^{(p)}+v)\cdot(x-\varphi_1^{(p)}(v))\cdots(x-\varphi_n^{(p)}(v)).$$

We must consider the following two problems in this approach: 1) how can we find the correspondence between the power-series roots expanded at different points ?, and 2) how *effective is the usage of roots expanded* at different points ?. We solve the problem 1) by utilizing Smith's theorem [11].

Theorem 2 (Smith 70) *Let $A(x)$ be a monic univariate polynomial in* C$[x]$, *and let $\zeta_1,\ldots,\zeta_n$ be n distinct numbers in* C. *Let n numbers $\rho_1,\ldots,\rho_n$ be defined as follows.*

$$\rho_i = \left|\frac{n\,A(\zeta_i)}{\prod_{j\neq i}^{n}(\zeta_j-\zeta_i)}\right| \quad (i=1,\ldots,n).$$

Let $D_1,\ldots,D_n$ be n discs in the complex plane, such that center$(D_i)=\zeta_i$ *and* radius$(D_i)=\rho_i$ $(i=1,\ldots,n)$. *Then, the union $D_1\cup\cdots\cup D_n$ contains all the roots of $A(x)$. Furthermore, if a union $D_1\cup\cdots\cup D_m$ is connected and does not intersect with $D_{m+1},\ldots,D_n$, then this union contains exactly m roots.*

Suppose that $\alpha_1^{(p)},\ldots,\alpha_n^{(p)}$ are not close to each other *and the expansion point $(s^{(q)})$ is well close to $(s^{(p)})$*. Then, $\alpha_1^{(p)},\ldots,\alpha_n^{(p)}$ will be good approximations of $\alpha_1^{(q)},\ldots,\alpha_n^{(q)}$. Therefore, putting $A(x)=F(x,s^{(q)})$ and $\zeta_i=\alpha_i^{(p)}$ $(i=1,\ldots,n)$, we evaluate Smith's error bounds ρ_i $(i=1,\ldots,n)$. If the resulting discs $D_1,\ldots,D_n$ do not intersect one another *and $\alpha_i^{(q)}\in D_i$ $(1\leq i\leq n)$, then we have the correspondence* $\alpha_i^{(p)} \Longleftrightarrow \alpha_i^{(q)}$ $(i=1,\ldots,n)$.

Example 2 Correspondence between the roots expanded at different points.

$$\begin{aligned} F &= x^5-2x^3+(3u^3+2u+3)x^2 \\ &+ (-4u^4-8u^2-3)x+(6u^5+7u^3+6u^2+2u+3). \end{aligned}$$

We consider the case $(s^{(1)})=(0)$ and $(s^{(2)})=(0.2)$. $F(x,0)$ has the following 5 different roots $\alpha_1^{(1)},\ldots,\alpha_5^{(1)}$.

$$-2.1038\cdots, \quad \pm i, \quad 1.0519\cdots\pm 0.5652\cdots i.$$

Choosing $\alpha_i^{(1)}$ as an approximate root of $F(x,0.2)$, we evaluate Smith's error bound $\rho_i^{(1)}$:

$$\begin{cases} \alpha_1^{(1)} & \longrightarrow \quad \rho_1 = 0.1589\cdots, \\ \alpha_{2,3}^{(1)} & \longrightarrow \quad \rho_{2,3} = 0.1395\cdots, \\ \alpha_{4,5}^{(1)} & \longrightarrow \quad \rho_{4,5} = 0.1842\cdots. \end{cases}$$

$F(x, 0.2)$ has the following 5 different roots $\alpha_1^{(2)}, \ldots, \alpha_5^{(2)}$.

$$-2.1599\cdots, \quad \pm 1.0392\cdots i, \quad 1.0799\cdots \pm 0.6472\cdots i.$$

Since $|\alpha_i^{(2)} - \alpha_i^{(1)}| < \rho_i$, we have the correspondence $\alpha_i^{(1)} \Longleftrightarrow \alpha_i^{(2)}$ $(i = 1, \ldots, 5)$. □

In the above example, we are able to "connect" the expansions at $(s^{(1)})$ and $(s^{(2)})$ directly. If $\|s^{(1)} - s^{(2)}\|$ is not small, however, Smith's error bounds $\rho_1, \ldots, \rho_n$ mentioned above become large. In such a case, we choose intermediate points between $(s^{(1)})$ and $(s^{(2)})$, such as

$$(\hat{s}^{(i)}) = \frac{r-i}{r}(s^{(1)}) - \frac{i}{r}(s^{(2)}), \quad i = 1, \ldots, r-1,$$

and connect the expansions at $(s^{(1)}) \Rightarrow (\hat{s}^{(1)}) \Rightarrow \cdots \Rightarrow (\hat{s}^{(r-1)}) \Rightarrow (s^{(2)})$, successively.

Remark 3 *The roots of $F(x, \hat{s}^{(i)})$ can be computed quickly by numerical iteration methods if we have good initial approximations of the roots. In the above scheme, we can use the roots of $F(x, \hat{s}^{(i-1)})$ as the initial approximations of the roots of $F(x, \hat{s}^{(i)})$.*

Remark 4 *If $F(x, u)$ has a singular point near $(\hat{s}^{(i)})$, the convergence radius of some root become very small. In such a case, we must choose the points $(\hat{s}^{(1)}), \ldots, (\hat{s}^{(r-1)})$ so as to bypass the singular point.*

Let the n roots $\bar{\varphi}_i(u)$ $(i = 1, \ldots, n)$ be expanded at two different points (s) and (s'), and let the roots expanded be $\bar{\varphi}_i(s+v) = \varphi_i(v)$ and $\bar{\varphi}_i(s'+v) = \varphi_i'(v)$:

$$\begin{cases} \varphi_i(v) &= \alpha_i + \displaystyle\sum_{k=1}^{\infty} \sum_{k_1+\cdots+k_\ell=k} c_{i;k_1,\ldots,k_\ell} v_1^{k_1} \cdots v_\ell^{k_\ell}, \\ \varphi_i'(v) &= \alpha_i' + \displaystyle\sum_{k=1}^{\infty} \sum_{k_1+\cdots+k_\ell=k} c'_{i;k_1,\ldots,k_\ell} v_1^{k_1} \cdots v_\ell^{k_\ell}. \end{cases} \tag{4.1}$$

Let $(s') = (s) + (d)$, then we have

$$c'_{i;k_1,\ldots,k_\ell} = \sum_{j=0}^{\infty} \sum_{j_1+\cdots+j_\ell=j} c_{i;k_1+j_1,\ldots,k_\ell+j_\ell} \binom{k_1+j_1}{j_1} \cdots \binom{k_\ell+j_\ell}{j_\ell} d_1^{j_1} \cdots d_\ell^{j_\ell}. \tag{4.2}$$

We have similar relations on $\varphi_i^j(v)$ $(j = 2, 3, \ldots)$. In order to show it, we have only to consider polynomials $F_j(x^j, u) = F(1x, u)\, F(\omega x, u) \cdots F(\omega^{j-1}x, u)$, where ω is a primitive jth root of 1, hence $F_j(x, u)$ has n roots $\{\bar{\varphi}_1(u)\}^j, \cdots, \{\bar{\varphi}_\ell(u)\}^j$. Thus, we obtain the following proposition.

Proposition 4 *Let $\varphi_i(v)$ $(i = 1, \ldots, n)$ and $\varphi_i'(v)$ $(i = 1, \ldots, n)$ be the power-series roots of $F(x, u)$, expanded at points (s) and (s'), respectively, and let $d \stackrel{\text{def}}{=} \|s - s'\| \ll 1$. Then, so long as we consider the coefficients of power-series roots to $O(d^\kappa)$, computing $\varphi_i(v)$ and $\varphi_i'(v)$ to total-degree k w.r.t. $v_1, \ldots, v_\ell$ is equivalent to computing $\varphi_i(v)$ to total-degree $k + \kappa$.*

The above analysis shows that, by using the power-series roots expanded at different points, we can avoid computing the roots to a high power. However, we should not choose the expansion points (s) and (s') such that $\|s - s'\| \ll 1$.

5 Finding approximate zero-sum relations

In the actual computation, what we treat are not the exact zero-sum relations in (2.9) but approximate ones among the row vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ in (3.2), such that

$$\begin{cases} \boldsymbol{a}_{i_1} + \cdots + \boldsymbol{a}_{i_m} = \boldsymbol{d}, \\ \|\boldsymbol{d}\| / \max\{\|\boldsymbol{a}_{i_1}\|, \cdots, \|\boldsymbol{a}_{i_m}\|\} < \varepsilon_{\text{cut}}, \end{cases} \tag{5.1}$$

where ε_{cut} is a suitably chosen small number. In this section, we explain a method to calculate the approximate zero-sum relations and investigate stability of the method. As we will see below, the method becomes very unstable if some power-series roots show "wild" behavior, and we will propose an idea to stabilize the method.

We calculate the roots $\alpha_1, \ldots, \alpha_n$ of $F(x, s)$ numerically by using floating-point numbers. Then, $\alpha_1, \ldots, \alpha_n$ contain numerical errors of magnitude $O(\varepsilon_M)$ which may cause large errors in the power-series roots. However, in this paper, we do not consider the numerical errors caused by rounding and cancellation. Compared with instability caused by "wild" roots, the numerical error is not a big problem; we have only to perform the computation with a higher precision.

We assume that the input polynomial has been "regularized" to satisfy the following relations (the regularization can be done easily by a transformation $F(x, u) \mapsto aF(bx, u)$, where a and b are suitably chosen real numbers).

$$\|f_n(u)\| \simeq \|F(x, u) - f_n(u)x^n\| \simeq 1. \tag{5.2}$$

Furthermore, we assume that $F(x, u)$ has been made approximately square-free at tolerance ε, by the multivariate approximate GCD operation, see [6], [12].

The algorithm **Elimi** *given in* 3 is for the exact zero-sum relations. We replace the "zero check" in the algorithm by the following "approximately-zero check":

1. if $|b_i| < \varepsilon_{\text{cut}}\|\boldsymbol{b}\|$, with b_i an element of a column vector $\boldsymbol{b}$, then treat b_i as an approximately-zero element of *tolerance* ε_{cut}, *of* $\boldsymbol{b}$;
2. if $\|\boldsymbol{b}'\| < \varepsilon_{\text{cut}}\|\boldsymbol{b}\|$, with $\boldsymbol{b}'$ a column vector eliminated by vector $\boldsymbol{b}$, then treat $\boldsymbol{b}'$ as an approximately-zero vector of tolerance ε_{cut}, in the elimination.

In the actual computation, we set ε_{cut} as $\varepsilon_{\text{cut}} = 10^{-2}$ and decrease it as $10^{-2} \Rightarrow 10^{-3} \Rightarrow 10^{-4} \Rightarrow \cdots$ until a desired tolerance. Then, we form the candidate factors and check the tolerance ε by the relation (1.1) *a posteriori*.

We define $\tilde{G}(x, v)$ and $\tilde{H}(x, v)$ as

$$\begin{cases} \tilde{G}(x, v) &= f_n(s+v) \cdot (x - \varphi_1(v)) \cdots (x - \varphi_m(v)), \\ \tilde{H}(x, v) &= f_n(s+v) \cdot (x - \varphi_{m+1}(v)) \cdots (x - \varphi_n(v)). \end{cases} \tag{5.3}$$

Therefore, we have $f_n F = \tilde{G}\tilde{H}$. We put

$$\begin{cases} \tilde{G}(x, v) &= (f_n / g_m) G(x, v) + \Delta_G(x, v), \\ \tilde{H}(x, v) &= (f_n / h_{n-m}) H(x, v) + \Delta_H(x, v). \end{cases} \tag{5.4}$$

Note that the expressions computed by the algorithm are not G and H but $\tilde{G}$ and $\tilde{H}$. Note further that the coefficients of

$\tilde{G}$ and $\tilde{H}$ (hence Δ_G and Δ_H) are infinite power-series. We represent $\tilde{G}$ and Δ_* ($*$ is either G or H) as

$$\begin{cases} \tilde{G}(x,v) &= f_n(v)x^m + \tilde{g}_{m-1}(v)x^{m-1} + \cdots + \tilde{g}_0(v), \\ \Delta_*(x,v) &= \delta_{*;n-1}(v)x^{n-1} + \cdots + \delta_{*;0}(v). \end{cases} \tag{5.5}$$

Then, the relations in (2.9) become as follows.

$$\begin{cases} \lfloor f_n \cdot (\varphi_1 + \cdots + \varphi_m) \rfloor_{\bar{e}+1} \simeq -\lfloor \delta_{G;m-1} \rfloor_{\bar{e}+1}, \\ \lfloor f_n^2 \cdot (\varphi_1^2 + \cdots + \varphi_m^2) \rfloor_{2\bar{e}+1} \simeq 2\lfloor \tilde{g}_{m-1}\delta_{G;m-1} \\ \qquad\qquad -f_n\delta_{G;m-2} \rfloor_{2\bar{e}+1}, \\ \qquad\qquad \cdots \qquad\qquad \cdots \\ \lfloor f_n^m \cdot (\varphi_1^m + \cdots + \varphi_m^m) \rfloor_{m\bar{e}+1} \simeq (-1)^m m \times \\ \qquad \lfloor \tilde{g}_{m-1}^{m-1}\delta_{G;m-1} - (m-2)f_n\tilde{g}_{m-1}^{m-3}\tilde{g}_{m-2}\delta_{G;m-1} \\ \qquad\qquad + \cdots - (-1)^m f_n^{m-1}\delta_{G,0} \rfloor_{m\bar{e}+1}, \end{cases}$$

where we have discarded $O([\delta_{G;ij}]^2)$ terms. These are approximate zero-sum relations.

In order to see the effect of perturbation Δ on $\tilde{G}$ and $\tilde{H}$, we consider the Hensel construction of $\tilde{G}$ and $\tilde{H}$. Let univariate polynomials $A_j(x), B_j(x)$ $(0 \le j < n)$ be defined to satisfy the following (in)equalities.

$$\begin{cases} A_j(x)\tilde{G}(x,0) + B_j(x)\tilde{H}(x,0) = x^j \quad (0 \le j < n), \\ \deg(A_j) < \deg(H), \quad \deg(A_j) < \deg(H). \end{cases} \tag{5.6}$$

We define a special multiplication notation $\circ$ as

$$A \circ F \stackrel{\text{def}}{=} A_{n-1}(x)f_{n-1}(u) + + \cdots + A_0(x)f_0(u). \tag{5.7}$$

Then, $\lceil \tilde{G} \rceil^k$ and $\lceil \tilde{H} \rceil^k$ are calculated iteratively as follows.

$$\begin{cases} \lceil \tilde{G} \rceil^k &= \lceil \tilde{G} \rceil^{k-1} + B \circ \lceil f_n F - \lceil \tilde{G} \rceil^{k-1} \lceil \tilde{H} \rceil^{k-1} \rceil^k, \\ \lceil \tilde{H} \rceil^k &= \lceil \tilde{G} \rceil^{k-1} + A \circ \lceil f_n F - \lceil \tilde{G} \rceil^{k-1} \lceil \tilde{H} \rceil^{k-1} \rceil^k. \end{cases} \tag{5.8}$$

Substituting $f_nGH + f_n\Delta$ for f_nF in the above formulas, we see that

$$\frac{\|\lceil \Delta_G \rceil^k\|}{\|\lceil \tilde{G} \rceil^k\|}, \ \frac{\|\lceil \Delta_H \rceil^k\|}{\|\lceil \tilde{H} \rceil^k\|} \simeq \frac{\|\Delta\|}{\|F\|} \simeq \varepsilon. \tag{5.9}$$

Similarly, we define $\hat{\varphi}_i$ and $\delta\varphi_i$ $(i = 1, \ldots, n)$ as follows.

$$\begin{cases} \varphi_i = \hat{\varphi}_i + \delta\varphi_i \quad (i = 1, \ldots, n), \\ GH = f_n \cdot (x - \hat{\varphi}_1) \cdots (x - \hat{\varphi}_n). \end{cases} \tag{5.10}$$

The Hensel construction allows us to calculate $\varphi_1, \ldots, \varphi_n$; in order to do so, we have only to set $\tilde{G} = f_n \cdot (x - \varphi_i)$ and $\tilde{H} = f_nF/(x - \varphi_i)$ in (5.6) and (5.8). Thus, we see

$$\frac{\|\lceil \delta\varphi_i \rceil^k\|}{\|\lceil \varphi_i \rceil^k\|} = O(\varepsilon) \quad (i = 1, \ldots, n). \tag{5.11}$$

Following Cauchy-Hadamard's theorem, we define

$$\gamma_i = \lim_{k \to \infty} \frac{\|\lceil \varphi_i \rceil^{k+1}\|}{\|\lceil \varphi_i \rceil^k\|} \quad (i = 1, \ldots, n). \tag{5.12}$$

If $\gamma_i \simeq 1$ hence we have $\|\lceil \varphi_i \rceil^k\| = O(1)$ for considerably large k then we say that the root φ_i is *tame*, otherwise it is called *wild*. Note that, if some root is wild then we have at least one other root which is also wild, because $\|f_{n-1}\| = O(1)$ and $f_n \cdot (\varphi_1 + \cdots + \varphi_n) = f_{n-1}$. Furthermore, even if some power-series roots are wild, $F(x,u)$ has usually many tame roots.

Assume that some power-series roots are wild, and put $\gamma = \max\{\gamma_1, \ldots, \gamma_n\}$. Then, (5.11) and (5.12) tell us that coefficients of terms of total-degree k, of wildest roots contain perturbations of magnitude $O(\gamma^k\varepsilon)$, while corresponding coefficients of tame roots are of magnitude $O(1)$. If these coefficients are contained in the matrix M randomly, the algorithm **Elimi** with any high precision cannot determine desired zero-sum relations unless $\gamma^k\varepsilon \ll 1$.

In many cases, the wild roots can be tamed as follows.

Strategy: Before applying the procedure **Elimi**, compute $(x - \varphi_{w_1}) \cdots (x - \varphi_{w_r})$ as a single factor, where $\varphi_{w_1}, \ldots, \varphi_{w_r}$ are the wildest roots, and treat $\varphi_{w_1}^k + \cdots + \varphi_{w_r}^k$ combinedly (the strategy can be applied recursively).

This strategy is quite useful in many cases, as we will show by an example in 6. However, if G and H have an approximately common factor hence we have

$$\|\lceil \tilde{G} \rceil^k\|, \ \|\lceil \tilde{H} \rceil^k\| \gg O(1) \quad \text{for large } k,$$

then the strategy does not work.

6 Empirical study

We show two examples, one is a bivariate polynomial of degree 20 containing two approximate factors at tolerance $O(10^{-5})$ and the other is a bivariate polynomial of degree 10 containing two approximate factors at tolerance $O(10^{-12})$. By the first example, we show that our method works quite well if the power-series roots are tame. On the other hand, the second example shows that wild roots make the computation very unstable. In order to show the instability clearly, we perform the computation with double-precision floating-point numbers and compute the power-series roots up to rather a high power.

We have implemented our method on Japanese algebra system GAL. The computer used is a SPARC Station 5 (CPU: microSPARC II, 70 MHz).

Example 3 Case of tame power-series roots.

$$\begin{aligned} F = {} & GH + 10^{-5}D, \quad \text{where} \\ G = {} & x^{10} + (u+1)x^9 + (u^2-2)x^8 + (2u^3 - u^2 + 2)x^7 \\ & + (u^7 - u^5 - 2u + 1)x^3 + (2u^8 - u^7 + 3u^3 - 2)x^2 \\ & + (u^9 + 5u^5 - u^3 + 4)x + (3u^{10} + 2u^6 + 3u^3 + 2), \\ H = {} & x^{10} + (u-2)x^9 + (u^2 + 3u - 3)x^7 + (u^3 + 3u^2 + 2)x^5 \\ & + (2u^6 - u^4 + 3u^2 - 4)x^4 + (u^8 + 4u^6 - 2u^2 + 2)x^2 \\ & + (3u^9 - u^7 + 2u^2 - 4)x + (u^{10} - 3u^7 - 2u^4 + 3), \\ D = {} & 2ux^9 - (3u^3 + u)x^8 + (u^5 - 3u^2)x^3 + (3u^7 - 4u^3)x. \end{aligned}$$

We note that G and H are absolutely irreducible and $G(x,0)$ and $H(x,0)$ are square-free. Calculating the power-series roots up to u^{20}, we see that

$$\|\lceil \varphi_i \rceil^{20}\| \lesssim 4.7 \ \text{ for } G, \qquad \|\lceil \varphi_i \rceil^{20}\| \lesssim 54 \ \text{ for } H,$$

hence all the roots are tame.

We see $\bar{e} = 1$ in this example, hence we formed the matrix M by the coefficients of $u^2, u^3, \ldots, u^{11}$ terms of $\varphi_i(u)$ and $u^3, u^4, \ldots, u^{11}$ terms of $\varphi_i^2(u)$. Then, after one application of algorithm Elimi, with the cutoff parameter $\varepsilon_{\rm cut} \simeq 10^{-3}$, we found that $\{1, 2, 3, 4, 6, 7, 12, 13, 14, 15\}$ rows in $\tilde{M}$ satisfy an approximate zero-sum relation with the tolerance 7.9×10^{-5} and $\{5, 8, 9, 10, 11, 16, 17, 18, 19, 20\}$ rows do with the tolerance 1.0×10^{-3}. (The computation failed at the value of $\varepsilon_{\rm cut} = 10^{-4}$). The computed factors $\tilde{G}$ and $\tilde{H}$, with $\mathrm{tdeg}(\tilde{G}) = \mathrm{tdeg}(\tilde{H}) = 10$, are almost dense, containing many small terms of magnitude $O(10^{-4})$ or less: $\|\tilde{G} - G\| \simeq 0.000053$ and $\|\tilde{H} - H\| \simeq 0.000015$. The total computation time is 2,080 ms (1910 ms for the power-series root computation, 120 ms for the zero-sum relation finding and 50 ms for the factor formation).

Example 4 Case of wild power-series roots.

$$
\begin{aligned}
F = {} & GH + 10^{-12} D, \quad \text{where} \\
G = {} & x^5 + (u-1)x^4 - (2u^2+3)x^3 + (2u^3 - 3u^2 - 2)x^2 \\
& + (u^4 - u^3 + 3u - 3)x + (2u^5 - 3u^3 - 4u^3 + 3), \\
H = {} & x^5 + (u+2)x^4 + (2u^2 - 3u - 1)x^3 \\
& + (2u^4 + 3u^3 - 4u^2 - 3u)x + (u^5 + 2u^4 - 4u^2 - 3), \\
D = {} & 2ux^9 + (3u^3 - 4u)x^6 + (u^5 + 3u^2)x^3 - (3u^7 + 4u^3)x.
\end{aligned}
$$

G and H are absolutely irreducible and $G(x,0)$ and $H(x,0)$ are square-free. One may think that this example is much simpler than Example 3, but it is not.

Calculating the power-series roots up to u^{11}, we see that $\|\lceil\varphi_i\rfloor^{11}\| \lesssim 2.57$ $(1 \le i \le 5)$ for $G(x,u)$. However, for $H(x,u)$, we find that $\|\lceil\varphi_i\rfloor^{11}\| \lesssim 13$ for $i = 3,4,5$ while $\|\lceil\varphi_i\rfloor^{11}\| \sim 4.4 \times 10^6$ for $i = 1,2$, hence φ_1 and φ_2 are wild:

$$
\begin{aligned}
\varphi_1 &\simeq -2.3 + \cdots - 190.u^5 + \cdots + 7.7 \times 10^5 u^{10} + \cdots, \\
\varphi_2 &\simeq -1.1 + \cdots + 191.u^5 + \cdots - 7.7 \times 10^5 u^{10} + \cdots.
\end{aligned}
$$

The power-series roots truncated at u^{11} contain errors of magnitude $O(10^7 \varepsilon_{\rm M})$. We have formed the matrix M by the coefficients of $u^2, u^3, \ldots, u^{11}$ terms of $\varphi_i(u)$. Then, one application of Elimi, with the cutoff parameter $\varepsilon_{\rm cut} = 10^{-4}$, gives us two approximate factors. (The computation failed at the value of $\varepsilon_{\rm cut} = 10^{-5}$). Note that the value of $\varepsilon_{\rm cut}$ $(= 10^{-4})$ is slightly larger than $\gamma^{11}\varepsilon = O(10^{7-12}) = O(10^{-5})$. The tolerance of approximate zero-sum relations among the row vectors in $\tilde{M}$ is 2.3×10^{-9} for rows corresponding to G and 1.2×10^{-4} for H. The computed factors $\tilde{G}$ and $\tilde{H}$, with $\mathrm{tdeg}(\tilde{G}) = \mathrm{tdeg}(\tilde{H}) = 5$, are such that $\|\tilde{G} - G\| \simeq 0.000000014$ and $\|\tilde{H} - H\| \simeq 0.0000000078$. The total computation time is 430 ms the 97% of which is spent to compute the power-series roots.

Finally, we comment that, if we apply the strategy mentioned in 5 to our example (i.e., we combine the roots φ_1 and φ_2 first), then even the polynomial $F = GH + 10^{-5}D$ can be factored by the algorithm Elimi with $\varepsilon_{\rm cut} = 10^{-4}$.

References

[1] Galligo A. and Watt S. A numerical absolute primality test for bivariate polynomials. Proc. ISSAC'97, ACM Press, 1997, 217-224.

[2] Huang Y., Wu W., Stetter H. J. and Zhi L. Pseudofactors of multivariate polynomials. Proc. ISSAC'00, ACM Press, 2000, 161-168.

[3] Kaltofen E. Fast parallel absolute irreducibility testing. J. Symb. Comput., 1, 57-67 (1985).

[4] Mou-Yan Z. and Unbehausen R. Approximate factorization of multivariable polynomials. Signal Proces., 14, 141-152 (1988).

[5] Nagasaka K. and Sasaki T. Approximate multivariate polynomial factorization and its time complexity. Preprint of Univ. Tsukuba, 16 pages, Sep. 1998 (submitted).

[6] Ochi M, Noda M.-T. and Sasaki T. Approximate greatest common divisor of multivariate polynomials and its application to ill-conditioned system of algebraic equations. J. Inf. Proces., 14, 292-300 (1991).

[7] Sasaki T., Kitamoto T. and Kako F. On cancellation error in Newton's method for power series roots of multivariate polynomial. Preprint of Univ. Tsukuba, 30 pages, Sep. 2000 (submitted).

[8] Sasaki T., Saito T. and Hilano T. Analysis of approximate factorization algorithm I. Japan J. Indust. Appl Math., 9, 351-368 (1992).

[9] Sasaki T., Suzuki M., Kolář M. and Sasaki M. Approximate factorization of multivariate polynomials and absolute irreducibility testing. Japan J. Indust. Appl. Math., 8, 357-375 (1991).

[10] Sasaki T. and Yamaguchi T. An analysis of cancellation error in multivariate Hensel construction with floating-point number arithmetic. Proc. ISSAC'98, ACM Press, 1988, 1-8.

[11] Smith B. T. Error bounds for zeros of a polynomial based on Gerschgorin's theorems. J. ACM, 17, 661-674 (1970).

[12] Zhi L. H. and Noda M.-T. Approximate GCD of multivariate polynomials. Proc. of Asian Technol. Conf. in Mathematics, 2000, 492-499.

ON THE CONSTRUCTION OF A PSE FOR GCD COMPUTATION

K. LI, L. H. ZHI AND M.-T. NODA

Dept. of Computer Science, Ehime University, Matsuyama 790-8577, Japan
E-mail: {likai, lzhi, noda}@hpc.cs.ehime-u.ac.jp

The increasing complexity in scientific and engineering computation is motivating the building of powerful Problem Solving Environments (PSEs). In this paper, we discuss current PSE-related research and propose a preliminary prototype and easy-to-use PSE for multivariate polynomial GCD computation using a combination of the Maple and Matlab packages and programs in C. This integrated computing environment enables improved use of existing resources to deliver more efficient solutions. This approach is of particular importance for large scale symbolic and numerical computations. An example is given to demonstrate the efficiency gains with solving approximate multivariate polynomial GCDs using Hensel lifting. Various issues related to the implementations are presented.

1 Introduction

The potential for Problem Solving Environments(PSEs) was recognized very early, but inadequate computing power made such systems infeasible until the 1980s when serious work started again. In April 1991, a research conference was held and issued a long report exploring this field [1]. In this report, PSE was defined to be *a computer system that provides all the computational facilities necessary to solve a target class of problems.* Thus PSE is essentially an integrated framework. Projects such as PSEware, IAMC and OpenXM are working on this area.

In this paper, we propose an easy-to-use PSE for GCD computation by Maple 6/Matlab/C integration(MMC). We discuss the relevant issues in detail, try to answer such questions as: Why do we need a PSE rather than a monolithic package? Why should we choose the combination of Maple 6, Matlab and C? How can those heterogeneous systems be available as components? What concern should we take during the implementation? Some examples that computing approximate GCDs of multivariate polynomials are given to demonstrate the efficiency and necessity to make use of the PSE rather than using a single package.

2 GCD Computation and its Solving Environments

2.1 CAS and GCD Computation

Problem of computing the greatest common divisor(GCD) of polynomials is a fundamental concern of algebraic manipulation. Most of the previous discussions are based on symbolic computation, and the relevant problems can be solved by computer algebra systems(CAS), which try to carry out problems in precise, exact mathematical models.

The research of GCD computation is evolving. Extensively, due to the accumulation of floating-point error or to imprecise input, the coefficients of polynomials may be inexact, and this leads to the conception of approximate GCD [6]. Considering that more and more symbolic-based algorithms involve a large amount of numerical computations, or those so called hybrid algorithms, should be addressed appropriately, people are expecting to obtain a more powerful system which is equipped with both of high-performance symbolic and numerical solvers.

2.2 MMC: the Architecture

So, who can play the role of PSE for GCD computation? By providing the framework for specifying the mathematical problem in a manner close to the standard scientific notation, CAS satisfy several PSE functions. However, their intrinsic limitations restrict their roles to be "problem solvers" rather than sophisticated PSEs. One of the critical reasons lies on that the manner of handling problems in exact models also leads to expensive memory consuming. In case of solving such problem as approximate GCD which involves both of symbolic and numerical computations, making full use of the advantages of existing CAS and traditional numerical approaches with high speed and low memory consuming might be the only choice right now.

From the view of GCD computation, considering that CAS have provided main facilities, we propose a simplified architecture which is a relatively succinct model illustrated in Fig. 1. Our strategy is treating a computer algebra system as one of computational engine or symbolic problem solver, as well as a "glue" environment; linking with other resources such as numerical based library NAG, and high performance interactive matrix computation system Matlab. We also hope it should be equipped with high level programming language such as C or Fortran, for handling the user-defined computations.

The reason why we choose Maple 6, Matlab and C to be main components of our framework is simple: They are powerful in their own appropriate domain; they are familiarized application packages and programming language;

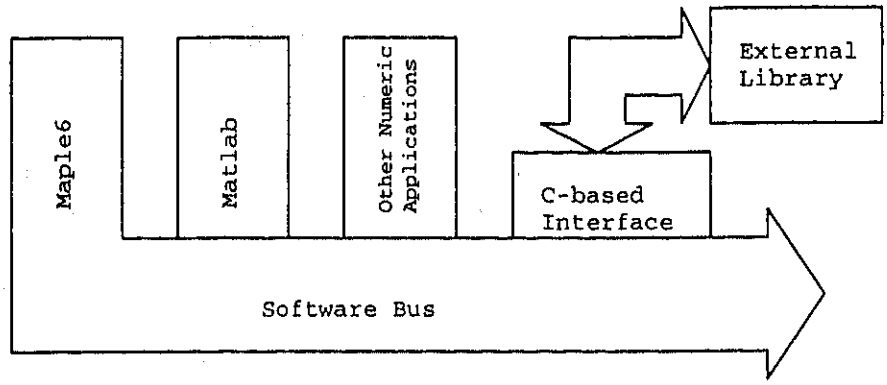


Figure 1. Architecture of PSE for GCD computation.

and the integration is easy to be done. The third one is the most attractive. Maple 6 provides a convenient interface to other packages and also the capability of invoking external C-based applications from within Maple. So itself can perform the function of "software bus" or "glue system" to wrap the necessary facilities. Thus people can benefit from this approach to construct their favorite system due to different cases.

3 Integrated Environment Construction

Maple 6 provides an easy-to-use interface linking with Matlab. By entering the command `with(Matlab)`, users can access all Matlab packages freely from within Maple. The following command lines show the process: in Maple's runtime worksheet, we link Matlab library with Maple, execute Matlab's command, get the computing answer, and convert it to Maple's data format.

```
with(Matlab);
setvar(''Matlab_data'', Maple_data);
Matlab[evalM](''Matlab_command'');
getvar(''Matlab_data'');
convert(%, Maple_type);
```

Besides the interface linking to Matlab, Maple 6 also provides the ability to call external functions implemented in C. These functions can be user-written, or supplied by a third party from a library. This external calling facility leads to a capability of constructing a more flexible PSE by sharing with other external solvers from within Maple. The external calling mecha-

nism works by generating and using wrapper functions. The `define_external` function generates a pair of wrapper functions, one written in Maple while the other in C. The generated C wrapper is compiled into a shared object file, and then linked into the running Maple kernel along with the actual external function to be called.

In addition to the above two integrations, the facility of linking with NAG also equips Maple 6 a powerful tool to handle numerical computations in an efficient way. In the next section we will find that it makes sense to apply the `LinearAlgebra` solver in place of invoking Matlab in some cases.

4 Implementation

The problem of approximate GCD of multivariate polynomials is drawing increasing attentions. In addition to two typical methods previously proposed by Noda, M.-T, Sasaki, T [3] and Corless, R. M. et al [2], another approach that modifies the EZGCD based on Hensel construction was briefly discussed[4]. Here we detail in the implementation of the algorithm using the prototype of this PSE framework.

The problem we consider here is: Given two multivariate polynomials $F(x_1, \cdots, x_n)$ and $G(x_1, \cdots, x_n)$ with coefficients of limited accuracy, are there nearby polynomials with a non-trivial GCD for a given tolerance ϵ?

When extending the Hensel algorithm from polynomials with exact coefficients to floating-point case, substantial modifications should be given to assure a satisfied performance. In accordance with the most important issues such as *Hensel construction, Hensel lifting bounding* and *candidate factors correction*, we proposed the corresponding methods: *Sylvester matrix QR, Solving overdetermined linear system* and *Linear optimization* to be as the solution[5]. The algorithm has been performed in Maple 6, Matlab and C. The framework of the implementation is illustrated in Fig. 2.

The main solver `Approx_GCD` is written in Maple, and can be executed in Maple 6. The arguments of the solver include the two given multivariate polynomials, the order of variables and the precision. It gives an expression of a polynomial acting as the optimized approximate GCD of the two given polynomials, or gives a constant "1" if there is no non-trivial GCD as an output. During the computing process, some heterogeneous solvers provided by Matlab and C routines are also be invoked from within Maple 6 by the main solver, in an effort to obtain a more efficient solution.

The *Sylvester matrix QR* solver is given by C routine `qr_sylvester.c`. Being as an external defined function, it is called as a Maple procedure after successfully defined by Maple 6's function `define_external` which generates

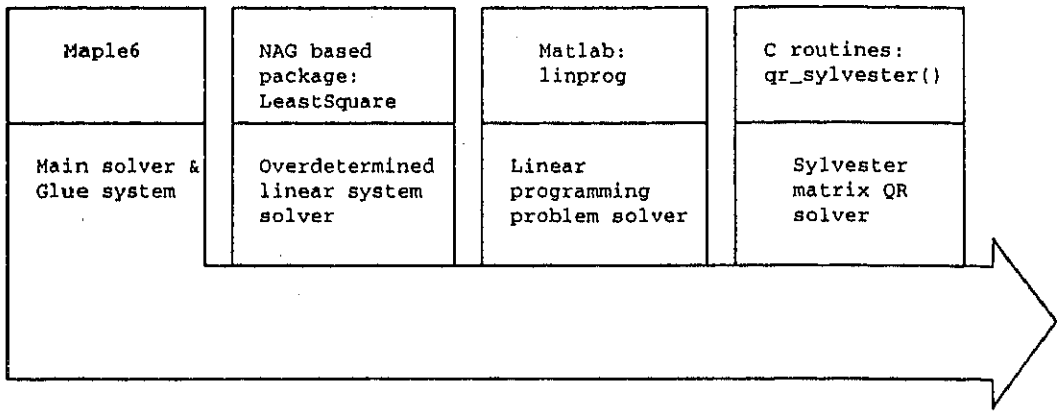


Figure 2. Framework of PSE for approx_GCD.

C wrappers to be linked into the running Maple kernel.

The *overdetermined linear system* can generally be solved by leastsquare method. In Maple 6, the solution has been offered by both of `linalg` package and `LinearAlgebra` package. Here we choose the NAG-based `LinearAlgebra` package because it achieves more than 100 times faster than the former one.

On the third point, we are facing the challenge of solving minimization problem, in an effort to obtain an optimized GCD and its cofactor with a sufficient small backward error:

$$\min \|A^T x - b\|_\infty, \tag{1}$$

$$\min\{y : A^T x - ye \leq b, A^T x + ye \geq b, y \in R\}. \tag{2}$$

Where A is a matrix and x, b are vectors. It is usually a large-scaled linear programming problem, since A is big. For example, we let $x = (x_1, x_2, x_3)$, suppose the degrees of two candidate factors are 2 and 4, then A is a 84×45 matrix. Although Maple offers an algorithm in its `simplex` package, it is designed for medium-scaled linear programming problems. On the other hand, Matlab provides an efficient large-scaled linear programming solver `linprog`, which is a primal-dual infeasible-interior-point approach. The solver offers us a satisfied solution when being invoked as a component from within Maple.

The algorithm has been implemented in both of Maple and Maple/Matlab/C(MMC) integrated PSE. The following table shows their comparison. The experiments are based on bivariate polynomials, the solver is `Approx_GCD`(F, G, ord, ϵ), where we let ord=$[x, y]$, and $\epsilon = 10^{-3}$.

Table 1. Computing Result Comparison(on DEC Alpha)

Number of terms		Maple		PSE(MMC)	
$F(x,y)$	$G(x,y)$	time(sec.)	backward error	time(sec.)	backward error
14	16	7.088	2.24e-3	1.448	2.01e-3
26	27	19.535	1.41e-3	3.798	1.82e-3
32	27	79.570	3.43e-3	3.808	2.45e-3
45	36	93.354	6.80e-4	4.303	8.66e-4
55	44	118.164	2.83e-2	5.840	2.47e-2
84	63	199.058	2.46e-3	6.954	1.60e-3
90	74	325.495	2.45e-3	7.331	2.46e-3
107	78	1783.306	1.36e-3	7.959	1.43e-3

5 Conclusion

In this paper, we briefly discussed the construction of an easy-to-use PSE by Maple 6, Matlab and C for GCD computation. It is just a preliminary attempt on the road of the relevant research. As an example, solutions are given to compute the approximate GCD of multivariate polynomials by this MMC-based PSE. Efficient and effective results are obtained. This research is still on-going, and more mature approaches are expected.

References

1. Gallopoulos, E., Houstis, E., and Rice, J. R.: Future research directions in problem solving environments for computational science, http://www.cs.purdue.edu/people/jrr, (1992).
2. Corless, R. M., Gianni, P. M., Trager, G. M. and Watt, S. M.: The singular value decomposition for polynomial systems, *Proc. ISSAC '95*, ACM Press, New York, 195-207 (1995).
3. Noda, M.-T. and Sasaki, T.: Approximate GCD and its application to illconditioned algebraic equations, *J. Comput. Appl. Math., 38*, 335-351 (1991).
4. Zhi, L. H. and Noda, M.-T.: Approximate GCD of Multivariate Polynomials, *Proc. ASCM'2000*, World Scientific Press, Singapore, 9-18 (2000).
5. Li, K., Zhi, L. H. and Noda, M.-T.: Solving approximate GCD of multivariate polynomials by Maple/Matlab/C combination, *Proc. ATCM'2000*, Thailand, 492-499 (2000).
6. Sasaki, T. and Noda, M.-T.: Approximate square-free decomposition and root-finding of ill-conditioned algebraic equations, *J. Info. Proces.*, Vol. 12, 159-168 (1989).

SYMBOLIC-NUMERIC COMPUTATIONS OF WU'S METHOD: COMPARISON OF THE CUT-OFF METHOD AND THE STABILIZATION TECHNIQUES

NOTAKE YOSHIO, HIROSHI KAI AND MATU-TAROW NODA
Department of Computer Science, Ehime University,
Bunkyo-cho 3, Matsuyama, Japan
E-mail: {notake,kai,noda}@hpc.cs.ehime-u.ac.jp

In this paper we propose two symbolic-numeric combined methods to solve polynomial equations by using Ritt-Wu's characteristic sets method. One method uses multivariate approximate-GCD to compute pseudo remainders and a cutoff parameter ϵ to neglect polynomials if the coefficients are sufficiently small. The other method uses Shirayanagi- Sweedler's stabilization techniques to guarantee reliability of obtained solutions. We discuss the two methods by using experimental results obtained from a problem of inverse kinematics of robot manipulators.

1 Introduction

Problems in robotics, computer aided design and control theory involve finding the solutions to systems of non-linear polynomial equations. Some of features of them are as follows :

- coefficients of the polynomials are floating point numbers or parameters.
- systems may be ill-conditioned depending on numerical coefficients, then it becomes difficult to solve them accurately by numerical methods.

Thus, symbolic algorithms should be considered to solve such problems.

It is well known that the Gröbner basis method can not be applied safely with floating point arithmetic. In this paper, the Ritt-Wu's characteristic sets method (abbreviated as Wu's method) is modified to solve systems of polynomial equations with floating point coefficients.

Software implementation of Wu's method to Maple V have already been done by D. Wang[1]. We first implement Wu's method in Risa/Asir by using a similar method to the one described by D. Wang[1], and then modify it to allow computations of polynomials with floating point coefficients. In the modification, a cutoff parameter is introduced to neglect polynomials whose coefficients are sufficiently small. Further, we show that the multivariate approximate-GCD proposed by Ochi,Noda and Sasaki[3] should be used in pseudo remainders.

Another method to compute Wu's method with floating point arithmetic uses Shirayanagi-Sweedler's stabilization technique[4]. We will show that solutions obtained by the latter are more reliable than those by the former.

2 Wu's method

Wu's method reduces input polynomial set PS to a family of triangular sets, which is called as characteristic set CS. Notations used in the algorithm are as follows:

- Let $x_1, x_2, \cdots, x_n$ be a set of indeterminates with order $x_1 \prec x_2 \prec \cdots \prec x_n$.
- PS, CS and RS are polynomial sets.
- $\mathrm{lvar}(p_i)$, $\mathrm{ldeg}(p_i)$ and $\mathrm{ini}(p_i)$ are the *leading variable*, the *leading degree* and the *initial* of p_i with respect to $\mathrm{lvar}(p_i)$.
- A finite set of polynomials $\{p_1, p_2, \ldots, p_n\}$ is called an *ascending set* if the following conditions are satisfied.
  - $\mathrm{lvar}(p_1) \prec \mathrm{lvar}(p_2) \prec \cdots \prec \mathrm{lvar}(p_n)$
  - For $i < j$, $\deg(p_i)$ with respect to $\mathrm{lvar}(p_i)$ is smaller than $\deg(p_j)$ with respect to $\mathrm{lvar}(p_i)$

Then, the algorithm of Wu's method is written as follows.
Algorithm 2.1 (Wu's method)
Input: a polynomial set PS
Output: a characteristic set CS

step1 $CS \leftarrow \mathtt{basset}(PS)$

step2 $RS \leftarrow \mathtt{remset}(PS, CS)$

step3 *If* $RS = \{\ \}$, *then return* CS *as the solution, else set* $PS = PS \cup RS$ *and go to* **step1**.

The algorithm is separated into two parts, `basset` and `remset`. The `basset` is a procedure to obtain an ascending set from a given polynomial set PS. Operations used in the `basset` are only comparison among degrees of each polynomial in PS.

The `remset` is a procedure to obtain a remainder set RS from PS and CS. Here, details of `remset` are as follows.

Algorithm 2.2 (remset)
Input: $PS = \{p_1, p_2, \cdots, p_n\}$ and $CS = \{c_1, c_2, \cdots, c_m\}$
Output: a remainder set RS

step1 $RS \leftarrow \{\}$ *and* $i \leftarrow 1$

step2 $r \leftarrow p_i$

step3 $r \leftarrow \mathrm{prem}(r, c_j)$ *for* $j = m, m-1, \cdots, 1$

step4 $RS \leftarrow RS \cup \{r\}$ *and* $i \leftarrow i + 1$

step5 *if* $i \leq n$, *go to* **step2**

The basic operation underlying all characteristic-set-based algorithms is the pseudo-remainder (prem) of two polynomials r and c_j with respect to some variable x. While dividing r by c_j, one can get a remainder formula of the form

$$I^s \cdot r = Q \cdot c_j + R,$$

where the polynomial I is the leading coefficient of c_j in x. The integer s is expressed as $s = 1 + \mathrm{ldeg}(r) - \mathrm{ldeg}(c_j)$. If $\mathrm{ini}(r)$ and $\mathrm{ini}(c_j)$ are relatively prime, $\mathrm{GCD}(\mathrm{ini}(r), \mathrm{ini}(c_j))$, then $I = \mathrm{ini}(c_j)/\mathrm{GCD}(\mathrm{ini}(r), \mathrm{ini}(c_j))$.

3 Wu's method using a cutoff parameter

If we use floating point arithmetic in Wu's method, pseudo-remainder computations in **Algorithm 2.2** produces rounding errors. Thus, we must consider error estimation for pseudo-remainders including such errors.

We discuss the pseudo-division

$$I^s f_1 = f_2 Q + R,$$

where $s = 1 + \mathrm{ldeg}(f_1) - \mathrm{ldeg}(f_2)$ and $I = \mathrm{ini}(f_2)/\mathrm{GCD}(\mathrm{ini}(f_1), \mathrm{ini}(f_2))$. Let $F_1(X)$ and $F_2(X)$ be polynomials as

$$F_1(X) = f_1(X) + \epsilon_1(X),$$
$$F_2(X) = f_2(X) + \epsilon_2(X).$$

Polynomials $\epsilon_1(X)$ and $\epsilon_2(X)$ are assumed to have negligible small coefficients. Then, pseudo-remainder for F_1 and F_2 are shown as follows.

$$\tilde{I}^s F_1(X) = F_2(X)\tilde{Q} + \tilde{R},$$
$$\tilde{I} = \mathrm{quo}(\mathrm{ini}(F_2)/\mathrm{GCD}(\mathrm{ini}(F_1), \mathrm{ini}(F_2); \epsilon)),$$

where $\mathrm{GCD}(\mathrm{ini}(F_1), \mathrm{ini}(F_2); \epsilon)$ shows an approximate-GCD with cutoff ϵ defined by Ochi, Noda and Sasaki[3], and the function quo means the quotient of $\mathrm{ini}(F_2)$ divided by $\mathrm{GCD}(\mathrm{ini}(F_1), \mathrm{ini}(F_2); \epsilon)$. The above pseudo-remainder, thus, may be called as an approximate pseudo-remainder. Then the error estimate of $\tilde{R} - R$ becomes important.

L_∞ norm for polynomials is used here for the error estimation. If it is assumed that s is sufficiently small and the following strong conditions,

$$||f_1|| \approx ||f_2|| \approx ||Q|| \approx 1$$

and

$$||\epsilon_1|| \approx ||\epsilon_2|| \approx ||\tilde{Q} - Q|| \approx \epsilon,$$

are satisfied, the error $\tilde{R} - R$ is estimated as follows.

$$||\tilde{R} - R|| = O(\epsilon)$$

By the error estimation above, Wu's method using a cutoff parameter is implemented as follows.

Algorithm 3.1 (Wu's method using cutoff parameter)
Input: a polynomial set PS, cutoff ϵ
Output: a characteristic set CS

step1 $CS \leftarrow$ `basset`(PS)

step2 $RS \leftarrow$ `apx-remset`(PS, CS)

step2' *For all* $r_i \in RS$
If $||r_i|| < \epsilon$, *then* $RS \leftarrow RS \backslash \{r_i\}$.

step3 *If* $RS = \{\ \}$, *then return* CS *as the solution, else set* $PS = PS \cup RS$ *and go to* **step1**.

The `remset` in **Algorithm 2.1** is replaced by `apx-remset`. The **step2'** is added to neglect small polynomials. The procedure `apx-remset` is written as follows.

Algorithm 3.2 (apx-remset)
Input: $PS = \{p_1, p_2, \cdots, p_n\}$, $CS = \{c_1, c_2, \cdots, c_m\}$ and cutoff ϵ
Output: a remainder set RS

step1 $RS \leftarrow \{\}$ *and* $i \leftarrow 1$

step2 $r \leftarrow p_i$

step3 $r \leftarrow$ `apx-prem`$(r, c_j; \epsilon)$ *for* $j = m, m-1, \cdots, 1$.

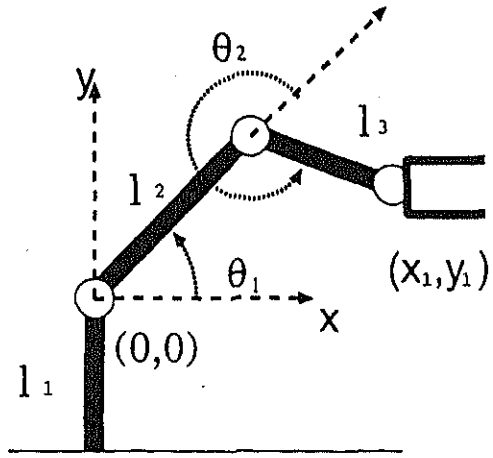


Figure 1. robot arms

step3' *If* $||r|| < \epsilon$, *then* $r \leftarrow 0$.

step4 $RS \leftarrow RS \cup \{r\}$ *and* $i \leftarrow i + 1$

step5 *if* $i \leq n$, *goto* **step2**

The procedure apx-prem computes the pseudo-remainder with tolerance ϵ, in the algorithm. The **step3'** is added to neglect small polynomials, again.

4 An example: a problem of robot manipulators

Algorithm 3.1 is applied to a problem of inverse kinematics of robot manipulators[2]. When the orthogonal frame (x, y) is introduced to a robot arm as in **Figure 1**, we consider how to obtain the rotation angle of the ith joint, θ_i.

The problem is modeled by the length of robot arms, $l_i, i = 1, \ldots, 3$, θ_1, θ_2, and joints. The joint (x_1, y_1) is expressed as

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} l_3 \cos(\theta_1 + \theta_2) + l_2 \cos(\theta_1) \\ l_3 \sin(\theta_1 + \theta_2) + l_2 \sin(\theta_1) \end{pmatrix}.$$

The equation and restrictions of trigonometric functions are written as

$$\begin{cases} x_1 = l_3(c_1c_2 - s_1s_2) + l_2c_1, \\ y_1 = l_3(c_1s_2 + c_2s_1) + l_2s_1, \\ c_1^2 + s_1^2 = 1, \\ c_2^2 + s_2^2 = 1. \end{cases}$$

where notations $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$ are used.

If the coordinate of the joint $(x_1, y_1) = (1.5, 0.3)$ and the length of arms $l_2 = 1.0, l_3 = 1.2$ are substituted into the equation, then the input polynomials for Wu's method are expressed with floating point number as follows.

$$\begin{cases} 1.2(c_1c_2 - s_1s_2) + 1.0c_1 - 1.5 = 0, \\ 1.2(c_1s_2 + c_2s_1) + 1.0s_1 - 0.3 = 0, \\ c_1^2 + s_1^2 - 1 = 0, \\ c_2^2 + s_2^2 - 1 = 0. \end{cases}$$

Here, we apply **Algorithm 3.1**, that is the modification of Wu's method proposed in previous section, to the above equations. The input polynomial set PS doesn't have any errors. However, since floating point computations are used in the algorithm, numeric errors should be considered during computations. Thus, to eliminate errors, the cutoff parameter is introduced. In the computation, it is taken as $\epsilon = 10^{-5}$. Then the solutions are obtained as follows:

$$\begin{cases} 40.2463s_1^4 - 9.80358s_1^3 - 23.176s_1^2 = 0, \\ -6.2208s_1^2c_1 - 1.24416s_1^3 + 3.93984s_1^2 = 0, \\ 11.6453s_1^3 + (8.95795s_2 - 1.41834)s_1^2 = 0, \\ 10.7495s_1^2c_2 - 4.44089 \times 10^{-16}s_1^3 + 0.447898s_1^2 = 0. \end{cases} \tag{1}$$

We obtained the characteristic set (1) by using 19 digits big-float on Risa/Asir. Rotation angles θ_1 and θ_2 can be successfully obtained by numerical computations from above triangular form.

The fourth polynomial in (1) contains the very small coefficient term, $-4.44089 \times 10^{-16}s_1^3$. In the example, the term has no effect to the results and may be neglected. However, in general, it is difficult to decide whether small term is necessary for the exact symbolic solution or not. Further, sometimes, the results may depend on cutoff values.

In the next section, we will consider another implementation of Wu's method using Shirayanagi-Sweedler's stabilization technique.

5 Wu's method and its stabilization

Shirayanagi and Sweedler proposed a method of algorithm stabilization techniques [4]. Their motivation was that computations by symbolic algorithms waste memory space by an intermediate swell of coefficients. Thus, if the algorithm is combined with a numeric computation carefully, the results may be accurate and stable, and furthermore computations may be done quickly. As a numeric computation, a concept of interval arithmetic is introduced.

Coefficients are described by rectangular interval numbers and are called as Bracket Coefficients. The stabilized algorithm is executed by an increasing precision of inputs, and then the result converges to the true output obtained by symbolic computation. If the bracket coefficient contains zero, then it is rewritten to zero. This process is called "Zero Rewriting".

Wu's method is modified by using the stabilization techniques as follows:

- Variables take values from Bracket coefficients
- Zero Rewriting is applied to the steps to obtain pseudo remainders `prem`
- Repeat the algorithm by increasing digits of inputs and for computations.

Wu's method using the stabilization techniques is summarized as follows:

Algorithm 5.1 (Wu's method using the stabilization techniques)
Input: a polynomial set PS (interval coefficients)
Output: a characteristic set CS (interval coefficients)

step1 $CS \leftarrow$ `basset`(PS)

step2 $RS \leftarrow$ `interval-remset`(PS, CS)

step3 *If* $RS = \{\ \}$, *then return* CS *as the solution, else set* $PS = PS \cup RS$ *and go to* **step1**.

In the procedure `interval-remset`, the process, Zero Rewriting, is applied to eliminate error for the coefficients of remainder polynomials. Further, it is necessary to increase the precision of big floating point arithmetic, and to repeat **Algorithm 5.1**.

The stabilized Wu's method is applied to the example discussed in the previous section. Input polynomial equations are expressed by using Bracket coefficients as

$$\begin{cases} [1.2, 1.2]\,(c_1c_2 - s_1s_2) + [1.0, 1.0]c_1 - [1.5, 1.5] = 0, \\ [1.2, 1.2]\,(c_1s_2 - c_2s_1) + [1.0, 1.0]s_1 - [0.3, 0.3] = 0, \\ [1.0, 1.0]\,c_1^2 + [1.0, 1.0]s_1^2 - [1.0, 1.0] = 0, \\ [1.0, 1.0]\,c_2^2 + [1.0, 1.0]s_2^2 - [1.0, 1.0] = 0. \end{cases} \tag{2}$$

If **Algorithm 5.1** is computed in 19 digits, the following solution is obtained.

$$
\begin{cases}
[40.24628674559999990, 40.24628674560000006]s_1^4 \\
+[-9.803582668800000024, -9.803582668799999971]s_1^3 \\
+[-23.17601341440000012, -23.17601341439999985]s_1^2, \\
[-6.220800000000000005, -6.220799999999999993]s_1^2c1 \\
+[-1.244160000000000001, -1.244159999999999998]s_1^3 \\
+[3.939839999999999993, 3.939840000000000006]s_1^2, \\
[11.64533759999999998, 11.64533760000000001]s_1^3 \\
+([8.957951999999999998, 8.957952000000000009]s_2 \\
+[-1.418342400000000003, -1.418342399999999997])s_1^2, \\
[10.74954239999999998, 10.74954240000000001]s_1^2c_2 \\
+[0.44789759999999999670, 0.44789760000000003094]s_1^2.
\end{cases}
\tag{3}
$$

Next, **Algorithm5.1** is repeatedly used but in an increasing precision. The same problem is solved by 29 digits by using the same algorithm. Obtained solutions are nearly the same as the previous solutions (3). Thus the computation terminates. Unlike the cutoff method discussed in the last section, the solutions have no negligible term. Thus, it seems that the **Algorithm 5.1** is more reliable than **Algorithm 3.1** for floating point arithmetic.

6 Conclusion

Symbolic-numeric combined methods to solve polynomial equations by using Ritt-Wu's characteristic sets method are discussed. Wu's method is modified and then applied to input polynomials with floating point coefficients. Two approximation methods for modified Wu's method are examined. They use 1) cutoff parameters, and 2) the stabilization techniques for computing polynomials with floating point coefficients. An example of the reverse problem of robot arms, shows that both methods give satisfactory results. Further, by detailed comparison of the two methods, we may conclude that the method 2) is more reliable than 1).

Several works remain to be done. They include

- establishing error estimation for pseudo-remainders
- making algorithms for symbolic-numeric combined computation even faster or parallelized.

References

1. D. Wang, Implementation and Applications of Characteristic Set Method, *Lecture Notes for 1994 Summer Graduate School in Mathematics*, Preliminary Version, (1994).
2. D. Manocha, Numerical Methods for Solving Polynomial Equations, *Proceedings of Symposia in Applied Mathematics*, AMS, pp.41–66, (1997).
3. M. Ochi, M. T. Noda and T. Sasaki, Approximate Greatest Common Divisor of Multivariate Polynomials and Its Application to ill-Conditioned Systems of Algebraic Equations, *Journal of Information Processing*, **14**(3), pp. 292–300, (1991).
4. K. Shirayanagi and M. Sweedler, A Theory Stabilizing Algebraic Algorithms, *Tech.Rep.95-28*, Cornell Univ., pp.1–92, (1995).
5. W.-t. Wu, A Mechanization method of equation-solving and theorem-proving, *Advances in Computing Research*, **6**, pp103–138, (1992).

Symbolic-numeric computation of Wu's method using stabilizing algorithm

Kei-ichi SHIRAISHI
Department of Control Engineering
Takuma National College of Technology
siraisi@dc.takuma-ct.ac.jp

Hiroshi KAI, Matu-Tarow NODA
Department of Computer Science
Ehime University
{kai,noda}@hpc.cs.ehime-u.ac.jp

Abstract

A symbolic-numeric combined method to solve polynomial equations have been proposed by using Ritt-Wu's characteristic sets method. The method is extended 1) to solve a system of polynomial equations with floating point coefficients and 2) to speed up by using parallel computations. Especially, in 1), above, the stabilization technique proposed by Shirayanagi and Sweedler is used. A parallelized method for the stabilizing algorithm of Wu's method proposed here is applied to an inverse kinematics problem of robot manipulators.

1 Introduction

Problems in robotics, computer aided design and control theory involve finding the solutions to systems of polynomial equations. Some of features of them are as follows :

- coefficients of the polynomials are floating point numbers or parameters,
- systems may be ill-conditioned depending on numerical coefficients, then it becomes difficult to solve them accurately by numerical methods.

Thus, symbolic algorithms, such as the Gröbner basis method and the Ritt-Wu's characteristic sets method (abbreviated as Wu's method)[1, 2], are be used to solve such problems.

It is well known that the Gröbner basis method can not be applied safely with floating point arithmetic. In this paper, Wu's method is modified to solve systems of polynomial equations with floating point coefficients and further is parallelized.

Wu's method is first implemented in a computer algebra system(CAS) Maple V by D. Wang[3] and also in the CAS Risa/Asir[4]. In the implementation in the Risa/Asir, Wu's

method is extended to allow computations of polynomials with floating point coefficients. In [4], an algorithm stabilization technique proposed by Shirayanagi and Sweedler[5] is used to obtain accurate solutions of given system of polynomial equations. We call it as a stabilized Wu's method. The stabilized Wu's method is executed in an increasing precision of inputs, then the output converges to the exact output obtained by the symbolic computation.

In this paper, Wu's method is implemented on a parallel computer, Fujitsu's AP3000. Parallelization of Wu's method have already been discussed by Wang[6] and I. A. Ajwa[7, 8]. However, a system of polynomial equations with floating point coefficients is not computed in their implementation. Thus here, the stabilized Wu's method is parallelized. In our parallel stabilized Wu's method, computations of increasing precision of inputs should be done. Each computation of precision is done on each processor(worker) in our parallelized method. Results of our parallel implementation are compared with that of by Wang and Ajwa. It is shown that our approach gives results faster than results by Wang and Ajwa.

2 Wu's method

Wu's method reduces input polynomial set PS to a family of triangular sets, which is called as characteristic set CS. Notations used in the algorithm are as follows:

- Let $x_1, x_2, \cdots, x_n$ be a set of indeterminates with order $x_1 \prec x_2 \prec \cdots \prec x_n$.
- PS, CS and RS are polynomial sets.
- $\mathrm{lvar}(p_i)$, $\mathrm{ldeg}(p_i)$ and $\mathrm{ini}(p_i)$ are the *leading variable*, the *leading degree* and the *initial* of p_i with respect to $\mathrm{lvar}(p_i)$.
- A finite set of polynomials $\{p_1, p_2, \ldots, p_n\}$ is called an *ascending set* if the following conditions are satisfied.
  - $\mathrm{lvar}(p_1) \prec \mathrm{lvar}(p_2) \prec \cdots \prec \mathrm{lvar}(p_n)$
  - For $i < j$, $\deg(p_i)$ with respect to $\mathrm{lvar}(p_i)$ is smaller than $\deg(p_j)$ with respect to $\mathrm{lvar}(p_i)$

Then, the algorithm of Wu's method is written as follows.

Algorithm 2.1 (Wu's method)
Input: a polynomial set PS
Output: a characteristic set CS

step1 $CS \leftarrow \mathtt{basset}(PS)$

step2 $RS \leftarrow \mathtt{remset}(PS, CS)$

step3 *If* $RS = \{\ \}$, *then return* CS *as the solution, else set* $PS = PS \cup RS$ *and go to* **step1**.

The algorithm is separated into two parts, **basset** and **remset**. The **basset** is a procedure to obtain an ascending set from a given polynomial set PS. Operations used in the **basset** are only comparison among degrees of each polynomial in PS.

The **remset** is a procedure to obtain a remainder set RS from PS and CS. Here, details of **remset** are as follows.

Algorithm 2.2 (remset)
Input: $PS = \{p_1, p_2, \cdots, p_n\}$ and $CS = \{c_1, c_2, \cdots, c_m\}$
Output: a remainder set RS

step1 $RS \leftarrow \{\}$ *and* $i \leftarrow 1$

step2 $r \leftarrow p_i$

step3 $r \leftarrow \mathrm{prem}(r, c_j)$ *for* $j = m, m-1, \cdots, 1$

step4 $RS \leftarrow RS \cup \{r\}$ *and* $i \leftarrow i+1$

step5 *if* $i \leq n$, *go to* **step2**

The basic operation underlying all characteristic-set-based algorithms is the pseudo-remainder (prem) of two polynomials r and c_j with respect to some variable x. While dividing r by c_j, one can get a remainder formula of the form

$$I^s \cdot r = Q \cdot c_j + R,$$

where the polynomial I is, the leading coefficient of c_j in x. The integer s is expressed as $s = 1 + \mathrm{ldeg}(r) - \mathrm{ldeg}(c_j)$. If $\mathrm{ini}(r)$ and $\mathrm{ini}(c_j)$ are not relatively prime, then $I = \mathrm{ini}(c_j)/\mathrm{GCD}(\mathrm{ini}(r), \mathrm{ini}(c_j))$.

3 Wu's method and its stabilization

Shirayanagi and Sweedler proposed a stabilization technique [5]. Their motivation was that computations by symbolic algorithms waste memory space by an intermediate swell of numerical coefficients. Thus, if the algorithm is combined with a numeric computation carefully, the results may be accurate and stable, and furthermore computations may be done quickly. As a numeric computation, a concept of interval arithmetic is introduced. Numerical coefficients are described by rectangular interval numbers and are called as Bracket Coefficients. The stabilized algorithm is executed in an increasing precision of inputs, and then the result converges to the true output obtained by the symbolic computation. If the bracket coefficient contains zero, then it is rewritten to zero. This process is called "Zero Rewriting".

Wu's method is modified by using the stabilization techniques as follows:

- Variables take values from Bracket Coefficients
- Zero Rewriting is applied to the steps to obtain pseudo remainders prem
- Repeat the algorithm in increasing digits of inputs and for computations.

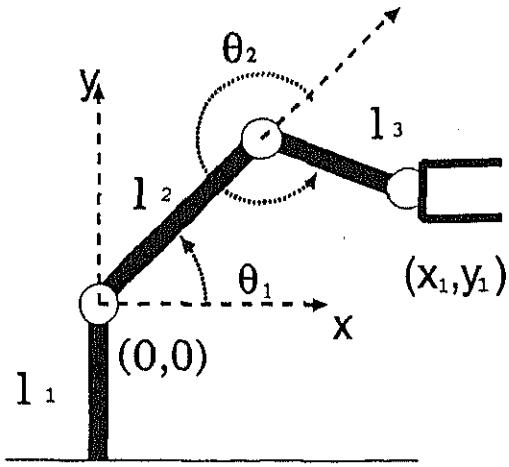


Figure 1: Robot arm

Wu's method using the stabilization technique is summarized as follows:

Algorithm 3.1 (Wu's method using the stabilization technique)
Input: a polynomial set PS (interval coefficients)
Output: a characteristic set CS (interval coefficients)

step1 $CS \leftarrow$ `basset`(PS)

step2 $RS \leftarrow$ `interval-remset`(PS, CS)

step3 *If* $RS = \{\ \}$, *then return* CS *as the solution, else set* $PS = PS \cup RS$ *and go to* **step1**.

In the procedure `interval-remset`, the process, Zero Rewriting, is applied to eliminate error for coefficients of remainder polynomials. Further, it is necessary to increase the precision of big floating point arithmetic, and to repeat **Algorithm 3.1**.

The stabilized Wu's method is applied to an inverse kinematics problem of robot manipulators [9]. When the orthogonal frame (x, y) is introduced to a robot arm as in Figure 1, we consider how to obtain the rotation angle of the ith joint, θ_i. The problem is modeled by the length of links, $l_i, i = 1, \ldots, 3$, θ_1, θ_2, and joints. The joint (x_1, y_1) is expressed as

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} l_3 \cos(\theta_1 + \theta_2) + l_2 \cos(\theta_1) \\ l_3 \sin(\theta_1 + \theta_2) + l_2 \sin(\theta_1) \end{pmatrix}.$$

The equation and restrictions of trigonometric functions are written as

$$\begin{cases} x_1 = l_3(c_1c_2 - s_1s_2) + l_2c_1, \\ y_1 = l_3(c_1s_2 + c_2s_1) + l_2s_1, \\ c_1^2 + s_1^2 = 1, \\ c_2^2 + s_2^2 = 1. \end{cases}$$

where notations $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$ are used.

If the coordinate of the joint $(x_1, y_1) = (1.5, 0.3)$ and the length of links $l_2 = 1.0$, $l_3 = 1.2$ are substituted into the equation, then the input polynomials for Wu's method are expressed with Bracket Coefficients as follows.

$$\begin{cases} [1.2, 1.2]\,(c_1c_2 - s_1s_2) + [1.0, 1.0]c_1 - [1.5, 1.5] = 0, \\ [1.2, 1.2]\,(c_1s_2 - c_2s_1) + [1.0, 1.0]s_1 - [0.3, 0.3] = 0, \\ [1.0, 1.0]\,c_1^2 + [1.0, 1.0]s_1^2 - [1.0, 1.0] = 0, \\ [1.0, 1.0]\,c_2^2 + [1.0, 1.0]s_2^2 - [1.0, 1.0] = 0. \end{cases} \tag{1}$$

If **Algorithm 3.1** is computed in 19 digits big-float on Risa/Asir, the following solution is obtained.

$$\begin{cases} [40.24628674559999990, 40.24628674560000006]s_1^4 \\ +[-9.80358266880000024, -9.80358266879999971]s_1^3 \\ +[-23.17601341440000012, -23.17601341439999985]s_1^2, \\ [-6.220800000000000005, -6.220799999999999993]s_1^2c1 \\ +[-1.244160000000000001, -1.244159999999999998]s_1^3 \\ +[3.939839999999999993, 3.939840000000000006]s_1^2, \\ [11.64533759999999998, 11.64533760000000001]s_1^3 \\ +([8.957951999999999998, 8.957952000000000009]s_2 \\ +[-1.418342400000000003, -1.418342399999999997])s_1^2, \\ [10.74954239999999998, 10.74954240000000001]s_1^2c_2 \\ +[0.44789759999999999670, 0.44789760000000003094]s_1^2. \end{cases} \tag{2}$$

Next, **Algorithm3.1** is repeatedly used but in increasing precisions. The same problem is solved by using the same algorithm in 28, 38 and 48 digits big-float. Obtained solutions are nearly the same as the previous solution (2). Thus the computation terminates.

In the next section, the stabilized Wu's method is implemented on a parallel computer for obtaining solutions quickly.

4 Wu's method and its parallelization

In a parallel implementation of the stabilized Wu's method, the following two types of computations should be considered. They are

1. allocating prem to workers,
2. allocating computation in differential precisions to each worker.

Above two types of allocating methods are discussed. They are applied to an inverse kinematics problem of robot manipulators.

In **Algorithm2.2**, the step2 and step3, the pseudo-remainder of p_i with respect to CS can be computed relatively independent. Most of time consuming steps of Wu's method are in these steps. We allocate polynomial pseudo-remainder computations to many processors in the parallelize Wu's method. A Master-Worker model is used for programming model. In **Algorithm2.1**, the master process computes the step 1, sends PS and CS to worker

processes, and receives RS from them. Then finally it computes the step 3. Worker processes receive p_i and CS, compute pseudo-remainders of p_i with respect to CS and send the pseudo-remainders back to the master process, i.e., worker processes compute the step2 in **Algorithm2.1**. This idea is proposed by Wang[6] and Ajwa[7, 8].

The stabilized Wu's method **Algorithm3.1** is executed in an increasing precision of inputs, and then the stability of obtained solutions is checked. We allocate the computation in different precision to many worker processes. The master process sends PS, CS and a precision to the worker process. It receives RSs computed in different precision and checks the stability of obtained solutions. Each worker process receives PS, CS and each precision, and computes RS in given precision. It sends RS back to the master process.

The parallel stabilized Wu's method is applied to an example of robot manipulators. The example here is an extended problem discussed in the previous section. The arm consists of 4 links. It is expressed as

$$\begin{cases} [1.4, 1.4]\,(c_1c_2c_3 - c_1s_2s_3 - s_1c_2s_3 - s_1s_2c_3) \\ +\,[1.2, 1.2]\,(c_1c_2 - s_1s_2) + [1.0, 1.0]\,c_1 - [3.0, 3.0] = 0, \\ [1.4, 1.4]\,(c_1c_2s_3 + c_1s_2c_3 + s_1c_2c_3 - s_1s_2s_3) \\ +\,[1.2, 1.2]\,(c_1s_2 + c_2s_1) + [1.0, 1.0]\,s_1 - [1.3, 1.3] = 0, \\ [1.0, 1.0]\,(c_1^2 + s_1^2 - 1.0) = 0, \\ [1.0, 1.0]\,(c_2^2 + s_2^2 - 1.0) = 0, \\ [1.0, 1.0]\,(c_3^2 + s_3^2 - 1.0) = 0. \end{cases}$$

where notations $c_i = \cos\theta_i, s_i = \sin\theta_i, i = 1, \ldots, 3$ are used.

In the problem, the number of equations is less than the number of unknowns. Thus solutions are positive-dimensional. If we substitute numerical constraints to the solution, we can obtain numerical values for each unknown. For example, $c_1 = 0.7071, s_1 = 0.7071, c_2 = 0.5453, s_2 = -0.8382, c_3 = 0.6574, s_3 = 0.7535$ is a solution obtained from the constraints. In Table 1, we show computational times for sequential Wu's method(Sequential), the parallel stabilized Wu's method with allocating prem(Prem parallel) and the parallel stabilized Wu's method with allocating computation in differential precisions(Precision parallel). These computations are done in 19, 28, 38, ..., 115 digits big-float. All experiments have been performed on a scalar parallel server Fujitsu AP3000 which have 24 nodes and APnet. Each node consists of 2 UltraSPARCII(360MHz) processors and 640MB memory. 4, 8, 12 nodes are used as workers. To obtain computation times, a built-in function of Risa/Asir, time(), is used. All computation times are shown in seconds.

Table 1: Computation times for parallel stabilized Wu's method(sec.)

No. of Workers	Sequential	prem parallel	precision parallel
1	216	—	—
4	—	89.4	64.2
8	—	64.6	43.8
12	—	59.8	29.2

The stabilized Wu's method is executed in increasing digits of inputs, and then the stability of obtained solutions is checked. Two types of parallel implementations are compared.

Comparisons of computation times show that the precision parallel case are faster than the prem parallel case. The reason of the fact depends on a number of communications among processors. When allocating prem computations to many workers, there occur too many processor communications than the method of allocating the computation in different precision to many workers.

5 Conclusion

The parallel implementation of the symbolic-numeric combined method to solve polynomial equations by using the Ritt-Wu's characteristic sets method(Wu's method) is discussed. Wu's method is modified and then applied to input polynomials with floating point coefficients. The algorithm stabilization technique is used for modifying Wu's method. The stabilized Wu's method is then parallelized. Here, computations are executed in an increasing precision and the stability of obtained solutions is checked. Two types of parallel implementations are discussed. They are

1. allocating prem to workers,

2. allocating computation in differential precisions to each worker.

Though an example of the inverse kinematics problem of robot manipulators, it is shown that the latter is faster than the former. From the number of communications among processors, the parallelized method of a allocating Wu's method computation in different precision to each worker is better than the method of allocating prem computations to workers.

The following problems remain for our future studies.

- How to check the stability of obtained solutions.

References

[1] W.-t. Wu, A Mechanization Method of Equations-solving and Theorem-proving, *Advances in Computing Research*, 6, pp.103–138, 1992.

[2] W.-t. Wu, Mechanical Theorem Proving in Geometries(translated by X. Jin and D. Wang), Springer-Verlag, Wien New York, 1994.

[3] D. Wang, Implementation and Applications of Characteristic Set Method, *Lecture Notes for 1994 Summer Graduate School in Mathematics*, Preliminary Version, 1994.

[4] Y. Notake, H. Kai and M. T. Noda, Symbolic-numeric computations of Wu's method: comparison of the cut-off method and the stabilization techniques, ASCM'2001, in printing.

[5] K. Shirayanagi and M. Sweedler, A Theory Stabilizing Algebraic Algorithms, *Tech.Rep.95-28*, Cornell Univ., pp.1–92, 1995.

[6] D. Wang, On the Parallelization of Characteristic-Set-Based Algorithms, *Proceedings of the 1st International ACPC Conference (Salzburg, Austria, September 30 - October 2, 1991)*, Springer's LNCS 591, pp.338–349, 1991.

[7] I. A. Ajwa, Parallel Algorithms and Implementations for the Gröbner Bases Algorithm and the Characteristic Sets Method, Ph.D. Dissertation, Kent State University, Kent, 1998.

[8] I. A. Ajwa, P. S. Wang, and D. Lin, Another Attempt for Parallel Computation of Characteristic Sets, *Computer Mathematics - Proceedings of the 4th Asian Symposium (ASCM 2000) (X.-S. Gao and D. Wang, eds.)*, World Scientific, Singapore New Jersey, pp. 63–66, 2000.

[9] D. Manocha, Numerical Methods for Solving Polynomial Equations, *Proceedings of Symposia in Applied Mathematics*, AMS, pp.41–66, 1997.

Reliable Computing 6: 429–438, 2000.

Hybrid Rational Function Approximation and Its Accuracy Analysis

HIROSHI KAI and MATU-TAROW NODA
Department of Computer Science, Faculty of Engineering, Ehime University, Bunkyo-cho 3, Matsuyama 790-8577, Japan, e-mail: {kai, noda}@cs.ehime-u.ac.jp

(Received: 30 June 1999; accepted: 4 February 2000)

Abstract. We propose a rational function approximation method combining numeric and symbolic computations. Given functions or data are first interpolated by a rational function, i.e. the ratio of polynomials. Undesired poles appearing in the rational interpolant are removed by an approximate-GCD method. We call the rational approximation a Hybrid Rational Function Approximation and abbreviate it as HRFA. In this paper we give a short survey of the HRFA and then discuss its accuracy analysis by using the approximate-GCD proposed by Pan.

1. Introduction

Among rational approximation methods for a given set of data, rational interpolation may be one of the simplest method. The set of data is interpolated by a rational function. For that purpose, the function is first evaluated at several data points in an interval and changed to the set of data. Rational functions discussed in the literature [17], [18] are sometimes restricted to be irreducible, i.e, numerator and denominator polynomials have no common factors other than a constant. However, if the interpolation is done in floating point computations, pathological facts have been observed by Noda et al. [14].

The following is shown through numerical experiments.

1. The denominator polynomial has a zero in the interval for the approximated range and this makes the rational function singular. We call the zero an *undesired zero.*
2. A zero of the numerator polynomial may arise which is very close to the undesired zero discussed above.

Experimental facts show that the singularity of the rational function may be removed by removing an approximate common factor from its numerator and denominator polynomials. We use an approximate-GCD algorithm to compute such an approximate common factor. We call this procedure Hybrid Rational Function Approximation and abbreviate it simply as HRFA.

2. Hybrid Rational Function Approximation

Rational interpolation is a method for interpolating a set of discrete data

$$D = \{(x_i, f_i) \mid i = 0, \ldots, m+n\}, \tag{2.1}$$

where $x_i \neq x_j$ for $i \neq j$, by a rational function

$$r_{m,n}(x) = P_m(x) / Q_n(x) \tag{2.2}$$

which satisfies the conditions $f_i = P_m(x_i) / Q_n(x_i)$, $i = 0, \ldots, m+n$. Here, m and n are the degree of the numerator and the denominator polynomials, respectively. $P_m(x) / Q_n(x)$ is called the (m, n) rational function. Rational interpolation can be used as a method of approximating functions $f(x)$, $x \in [a, b]$ as follows.

- Give sample points x_i, $i = 0, \ldots, m+n$ s.t. $a = x_0 < \cdots < x_{m+n} = b$ and compute the values, $f_i = f(x_i)$, $i = 0, \ldots, m+n$.
- Obtain a rational interpolant $r_{m,n}(x) = P_m(x)/Q_n(x)$ satisfying $f_i = P_m(x_i)/Q_n(x_i)$.

The computation of the rational interpolant is done based on a set of linearized equations, Thiele continued fractions, and so on [3], [5], [18].

There are two well-known problems in rational interpolation:

- there may be unattainable points in D,
- there may be undesired poles in $[a, b]$.

Several improvements have been proposed by several authors as follows:

- Van Barel and Bultheel [22] have suggested a modified rational interpolation that has no unattainable point. However, the second problem remains if the method is applied to function approximation.
- A condition is imposed that ensures that the rational interpolant (2.2) does not have any poles on $[a, b]$. However, the condition is valid only for the case that a continuous function $f(x)$ is given.
- The method proposed by Berrut [1], [2] overcomes the two problems if $f(x)$ is continuous in $[a, b]$. However, if $f(x)$ is discontinuous and has poles in $[a, b]$, the method is not applicable.

Thus, there is no known method which avoids the above difficulties when any $f(x)$ is approximated. In numerical computations of rational interpolants (2.2), the following interesting phenomena are observed.

EXAMPLE 2.1. We consider that a function $f(x) = 1/\sin(x - \pi/4)$ is approximated by (6, 6) rational interpolation. The set of data D is given as equidistant points $x_i = i/12$ and $f_i = f(x_i)$ for $i = 0, \ldots, 12$. The (6, 6) rational interpolant is expressed as

$$R_{6,6}(x) = \frac{P_6(x)}{Q_6(x)},$$

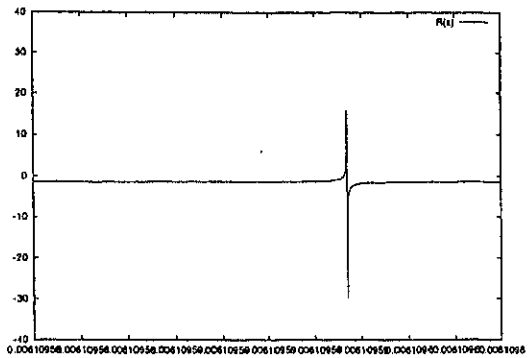


Figure 1. The undesired pole of $R_{6,6}(x)$.

$$\begin{aligned} P_6(x) &= -0.18131x^6 + 3.3199x^5 - 9.2545x^4 \\ &\quad + 70.090x^3 - 75.525x^2 + 231.93x - 1.4142, \\ Q_6(x) &= -5.5692x^6 + 17.993x^5 + 2.0987x^4 \\ &\quad - 20.796x^3 + 216.91x^2 - 165.00x + 1. \end{aligned}$$

The denominator polynomial $Q_6(x)$ has an undesired zero $z_0 = 0.0061095\ldots$ Thus, the rational interpolant $R_{6,6}(x)$ is not suitable for the approximation of $f(x), x \in [0, 1]$ because the error of $R_{6,6}(x)$ near z_0 is very large. $R_{6,6}(x)$ around $x = z_0$ is shown in Figure 1. On the other hand, the numerator polynomial $P_6(x)$ has a zero very close to z_0. This fact suggests that if we can remove both zeros of the numerator and denominator polynomials from the rational interpolation, we obtain a modified rational function without the undesired zero.

We compute an approximate-GCD of $P_6(x)$ and $Q_6(x)$ to obtain an approximate common factor of $P_6(x)$ and $Q_6(x)$. The approximate-GCD proposed by Sasaki and Noda with accuracy $\varepsilon = 10^{-3}$ is given by

$$g_1(x) = \mathrm{GCD}(P_6(x), Q_6(x);\ \varepsilon) = x - 0.006109593908.$$

$g_1(x)$ is removed from the rational interpolant $R_{6,6}(x)$ by division, and then rational approximation $r_{5,5}(x)$ is obtained as follows.

$$\begin{aligned} r_{5,5} &= \frac{p_5(x)}{q_5(x)}, \\ p_5(x) &= -0.18131x^5 + 3.3188x^4 - 9.2343x^3 \\ &\quad + 70.033x^2 - 75.097x + 231.47, \\ q_5(x) &= -5.5692x^5 + 17.959x^4 + 2.2084x^3 \\ &\quad - 20.782x^2 + 216.78x - 163.68. \end{aligned}$$

The rational approximant $r_{5,5}(x)$ has no undesired pole. Thus, we obtain a useful rational approximant for the given set of data D. The above procedure is put into the following algorithm.

Algorithm 1. Hybrid Rational Function Approximation (HRFA) [14].

Input: A set of discrete data D and accuracy ε for the approximate-GCD.

Output: *A rational approximation $r(x)$ to the set of data D, which satisfies*

$$|f_i - r(x_i)| \leq E, \qquad i = 0, \ldots, m+n, \tag{2.3}$$

where E is a small positive real number.

Method:

1. *Approximate given points and values with a rational function.*

$$R_{m,n}(x) = \frac{P_m(x)}{Q_n(x)} = \frac{\sum_{i=0}^{m} a_i x^i}{\sum_{i=0}^{n} b_i x^i}. \tag{2.4}$$

   We normalize the rational function by requiring that $b_0 = 1$.

2. Obtain the approximate-GCD of $P_m(x)$ and $Q_n(x)$ with certain accuracy parameter ε,

$$g(x) = \mathrm{GCD}\big(P_m(x), Q_n(x); \varepsilon\big),$$

   using the algorithm of approximate-GCD proposed by Sasaki and Noda [19].

3. Divide $P_m(x)$ and $Q_n(x)$ by $g(x)$ and let

$$r_{m,n}(x) = \frac{p(x)}{q(x)} = \frac{\mathrm{quo}(P_m(x), g(x))}{\mathrm{quo}(Q_n(x), g(x))}, \tag{2.5}$$

   where $\mathrm{quo}(A, B)$ denotes the quotient of A and B.

The process to remove the approximate-GCD from the rational interpolation in the last step of the algorithm may be replaced by constructing a continued fraction. This may give more accurate results of HRFA than those of [8]. The HRFA algorithm has also a close relation with Padé approximation, as shown in [7].

We applied the HRFA to the Cauchy Principal Value integral (CPV) and a kind of the Cauchy-type singular integral equation. Through examples, we showed how computations by the HRFA gives more accurate and stable results than usual numerical computations. Their detailed discussions are described in [8] and [9].

In the HRFA algorithm, two difficulties occur:

- No relation between the parameter ε for the approximate-GCD and the error E of the obtained rational function is obtained.

- There may be unattainable points of the rational interpolant.

We will consider the first problem in Section 3. A relation between the accuracy of the approximate-GCD and error bounds of the HRFA will be given. For the second problem, we can check the values $Q_n(x_i)$ for $i = 0, \ldots, m+n$ to decide whether unattainable points may arise. Unattainable points are defined for the set of points (x_i, f_i) as

$$R_{m,n}(x_i) \neq f_i. \tag{2.6}$$

The linearized equations to obtain the rational interpolant (2.2) are

$$Q_n(x_i)f_i - P_m(x_i) = 0, \qquad i = 0, \ldots, m+n.$$

Thus, if (2.6) holds, we must have $P_m(x_i) = Q_n(x_i) = 0$. Thus it is enough to check whether $Q_n(x_i)$ is equal to zero or not for the rational interpolant obtained by the linearized equations.

3. Accuracy Analysis of HRFA

Several algorithms for calculating approximate-GCD and for polynomial division have been proposed other than Schönhage [20] and Sasaki and Noda [19]. They use different norms for the definition of approximate-GCD. For example, the L_1 norm is used in [6], [16], and the L_2 norm is used in [4], [12]. Sederberg and Chang [21] proposed first degree best approximate common factor for a norm defined in a domain.

In further discussions to establish the accuracy analysis of HRFA, we consider approximate-GCDs defined by the L_1 norm, i.e. these proposed by Hribernig and Stetter [6], and by Pan [16].

3.1. HRFA by Approximate-GCD Proposed by Hribernig and Stetter

The approximate-GCD of two polynomials $f_1, f_2 \in C[x]$, where $C[x]$ is the set of complex number coefficient polynomials, is the following concept.

DEFINITION 3.1 Hribernig and Stetter. At the accuracy level α, two polynomials $\tilde{f}_i \in C[x]$, $i = 1, 2$, possess a near-GCD $\tilde{g}$ if there exist polynomials $f_i^* \in C[x]$, $i = 1, 2$, which satisfy

$$\gcd(f_1^*, f_2^*) = \tilde{g} \quad \text{and} \quad \|\tilde{f}_i - f_i^*\| \leq \alpha, \quad i = 1, 2. \tag{3.1}$$

Equivalently, a near-GCD $\tilde{g}$ of $\tilde{f}_1$ and $\tilde{f}_2$ satisfies

$$\tilde{f}_i = \tilde{g} \cdot \tilde{q}_i + \tilde{r}_i \quad \text{with} \quad \|\tilde{r}_i\| \leq \alpha, \quad i = 1, 2. \tag{3.2}$$

A near-GCD at accuracy level α will be denoted by α-GCD$(\tilde{f}_1, \tilde{f}_2)$.

Here, the norm of the polynomial means the L_1 norm, that is $\|p(x)\| = \sum_{j=0}^{n} |a_j|$ for polynomials $p(x) = \sum_{j=0}^{n} a_j x^j$.

If α-GCD is used in place of the approximate-GCD by Sasaki and Noda, $\tilde{f}_1$ and $\tilde{f}_2$ in the equation (3.2) should be read as $P_m(x)$ and $Q_n(x)$, respectively. Also $\tilde{q}_1$, $\tilde{q}_2$ and $\tilde{g}$ should be read as $p(x)$, $q(x)$ and $g(x)$, respectively. Therefore, (3.2) is expressed as

$$\begin{cases} P(x) = p(x)g(x) + \delta P(x), \\ Q(x) = q(x)g(x) + \delta Q(x), \end{cases} \tag{3.3}$$

where $\delta P(x) = \tilde{r}_1$ and $\delta Q(x) = \tilde{r}_2$ in the HRFA with the $\tilde{r}_i$ $(i = 1, 2)$ of Definition 3.1.

We consider the error between $p(x) / q(x)$ and the interpolated rational function $P(x) / Q(x)$. The following theorem holds [10].

THEOREM 3.1. *Let $P(x)$ and $Q(x)$ be polynomials in $C[x]$ and let $p(x)$ and $q(x)$ be polynomials satisfying $\|\delta P(x)\| \leq \alpha < 1$ and $\|\delta Q(x)\| \leq \alpha < 1$ in the expression (3.3). Then*

$$\left| \frac{P(z)}{Q(z)} - \frac{p(z)}{q(z)} \right| \leq \left(1 + \left| \frac{P(z)}{Q(z)} \right| \right) \cdot \frac{\alpha}{1 - \alpha}, \tag{3.4}$$

for every $z \in [-1, 1]$ s.t. $|Q(z)| \geq 1$.

Theorem 3.1 gives the relation between the accuracy α of near-GCD and the error E of the HRFA. We will normalize the error and rewrite the HRFA Algorithm 1, as follows:

Algorithm 2. HRFA by using near-GCD.

Input: A set of data D and accuracy E of the HRFA.

Output: A rational function $p(x) / q(x)$ satisfying

$$|p(x_i) / q(x_i) - y_i| \leq E, \quad i = 0, \ldots, m + n.$$

Method:

1. Obtain the rational interpolant $P_m(x) / Q_n(x)$.
2. Compute $c = \min_{i=0,\ldots,m+n} |Q_n(x_i)|$. If $c = 0$, unattainable points appear in D and one must stop.
3. Normalize $P_m(x)$ and $Q_n(x)$ as follows:

$$\bar{P}_m(x) = P_m(x) / c, \quad \bar{Q}_n(x) = Q_n(x) / c,$$

4. Obtain α-GCD $g(x)$ of $\tilde{P}(x)$ and $\tilde{Q}(x)$ where $\alpha = E/\left(1+\max_{i=0,\ldots,m+n}\{|f_i|\}+E\right)$ and the rational approximation $p(x)/q(x)$.

3.2. HRFA Using Approximate-GCD Proposed by Pan

The approximate-GCD proposed by Pan is defined as follows. For two polynomials $u(x)$ and $v(x)$, for the polynomial norm $\|\sum_i p_i x^i\| = \sum_i |p_i|$ and for a real b, an approximate gcd, $d^* = \gcd^*(u(x), v(x))$, has been nonuniquely defined as $\gcd(u^*(x), v^*(x))$, where

$$\deg u^*(x) \le \deg u(x), \qquad \deg v^*(x) \le \deg v(x), \tag{3.5}$$

$$\|u^*(x) - u(x)\| \le 2^{-b}\|u(x)\|, \qquad \|v^*(x) - v(x)\| \le 2^{-b}\|v(x)\|, \tag{3.6}$$

and $u^*(x)$ and $v^*(x)$ satisfying (3.5) and (3.6) are (nonuniquely) chosen so as to maximize $\deg(d^*(x))$.

Several methods for computing the approximate-GCD are proposed in [16]. We consider δ-gcds which are summarized as follows:

- The two polynomials

$$u(x) = u \times \prod_{i=1}^{n}(x - y_i), \qquad v(x) = v \times \prod_{j=1}^{m}(x - z_j),$$

  and δ are given.

- A δ-gcd $\tilde{d}_\delta(x)$ of $u(x)$ and $v(x)$ is defined by the equation

$$\tilde{d}_\delta(x) = \prod_{q=1}^{r}(x - x_q), \qquad x_q = (y_{i_q} + z_{j_q})/2, \quad q = 1, \ldots, r,$$

  where the set of pairs $(y_{i_1}, z_{j_1}), \ldots, (y_{i_r}, z_{j_r})$ is a maximum matching which maximizes r and satisfies $|y_{i_q} - z_{j_q}| \le 2\delta$.

- If δ is bounded by the equation $\delta \le (1 + 2^{-b})^{1/n} - 1$ and

$$p(x) = u \times \prod_{k=r+1}^{m}(x - y_{i_k}), \qquad q(x) = v \times \prod_{l=r+1}^{n}(x - z_{j_l}),$$

  where y_{i_k} and z_{j_l}, which are not used in the computation of the δ-gcd, are zeros of $u(x)$ and $v(x)$ respectively, then polynomials $u^*(x) = p(x)\tilde{d}_\delta(x)$ and $v^*(x) = q(x)\tilde{d}_\delta(x)$ satisfy the definitions (3.5) and (3.6).

If the δ-gcds are used in the HRFA, we can easily give an accuracy analysis of the HRFA from Theorem 3.1. The polynomials, $u(x)$, $v(x)$, $u^*(x) - u(x)$, $v^*(x) - v(x)$ and $d^*(x)$ appearing in the δ-gcd, should be read as $P_m(x)$, $Q_n(x)$, $\delta P(x)$, $\delta Q(x)$ and $g(x)$, respectively. Thus, for $\alpha = 2^{-b}\max\{\|P(x)\|, \|Q(x)\|\}$, the inequality

$$\left|\frac{P(z)}{Q(z)} - \frac{p(z)}{q(z)}\right| \le \left(1 + \left|\frac{P(z)}{Q(z)}\right|\right) \cdot \frac{\alpha}{1 - \alpha},$$

holds for every $z \in [-1, 1]$ such that $|Q(z)| \geq 1$. We can rewrite the HRFA using the δ-gcds as follows:

Algorithm 3. HRFA using approximate-GCD proposed by Pan.

Input: A set of data D and accuracy E of the HRFA.

Output: A rational function approximation $p(x) / q(x)$ which satisfies

$$|f_i - p(x_i) / q(x_i)| \leq E. \tag{3.7}$$

Method:

1. Obtain the rational interpolant $P_m(x) / Q_n(x)$.
2. Compute $c = \min_{i=0,\ldots,m+n} |Q_n(x_i)|$. If $c = 0$, unattainable points appear in D, one must stop.
3. Normalize $P_m(x)$ and $Q_n(x)$ as
$$\tilde{P}_m(x) = P_m(x) / c, \qquad \tilde{Q}_n(x) = Q_n(x) / c.$$
4. Obtain a δ-gcd $g(x)$ of $\tilde{P}_m(x)$ and $\tilde{Q}_n(x)$ where $\delta = (1 + 2^{-b})^{1 / \max\{m,n\}} - 1$ and $2^{-b} \leq E / \left(\max\{\|\tilde{P}_m(x)\|, \|\tilde{Q}_n(x)\|\}(1 + \max_{i=0,\ldots,m+n} \{|f_i|\} + E)\right)$.
5. Obtain $p(x), q(x)$ as follows:
$$p(x) = u \times \prod_{k=\deg(g(x))+1}^{m} (x - y_{i_k}),$$
$$q(x) = v \times \prod_{l=\deg(g(x))+1}^{n} (x - z_{j_l}),$$
where y_{i_k} and z_{j_l}, which are not used in the computation of the δ-gcd, are zeros of $P_m(x)$ and $Q_n(x)$ and $u = \mathrm{lcoef}(\tilde{P}_m(x))$, $v = \mathrm{lcoef}(\tilde{Q}_n(x))$, where $\mathrm{lcoef}(\tilde{P}_m(x))$ means the leading coefficient of the polynomial $\tilde{P}_m(x)$.

We demonstrate through the next example that the HRFA by the δ-gcds may have advantage over the HRFA by near-GCD.

EXAMPLE 3.1. The function $f(x) = \sin(x) / (x - \pi/4)$, $x \in [-1, 1]$ is approximated with the equidistant points from -1 to 1 as the sample points x_i, $i = 0, \ldots, m+n$. That is, $x_i = -1 + 2 \times i / (m+n)$, $i = 0, \ldots, m+n$. The error of the rational approximation is computed as

$$E_{Ave} = \frac{\sum_{i=0,\ldots,99} |r(x_i) - f(x_i)|}{100}, \qquad x_i = -1 + 2 \times i / 99.$$

Table 1 shows numerical comparisons between the error of the HRFA using near-GCD and δ-gcds. Here the near-GCD computed with $E = 10^{-3}$ and the δ-gcds

Table 1. The error, E_{Ave}, of rational approximations for $\sin(x) / (x - \pi/4), x \in [-1, 1]$.

interpolation	HRFA(Stetter)		HRFA(Pan)	
$R_{m,n}$	$r_{m,n}$	E_{Ave}	$r_{m,n}$	E_{Ave}
(4,4)	(3,3)	2.4858×10^{-4}	(3,3)	1.2885×10^{-4}
(6,6)	(5,5)	3.6042×10^{-9}	(5,5)	1.8385×10^{-9}
(11,11)	(7,7)	2.2271×10^{-8}	(7,7)	4.3366×10^{-10}
(12,12)	(7,7)	7.5450×10^{-7}	(7,7)	2.7414×10^{-9}
(13,13)	(7,7)	2.4439×10^{-9}	(7,7)	3.6617×10^{-11}

computed with $E = 10^{-3}$ are shown. In this example, the Durand-Kerner method is used to compute the zeros of the numerator and denominator polynomials for the δ-gcds.

Results are shown in Table 1. A rational interpolant $R_{m,n}$ contains the approximate-GCD. By the HRFA algorithms, it is reduced to a hybrid rational function $r_{m,n}$ with the error E_{Ave}. Table 1 show that the rational approximations obtained with Algorithm 3 are more accurate than those obtained with Algorithm 2.

4. Conclusion and Future Work

We give a brief discussion of the HRFA and its accuracy analysis.

The approximate-GCD algorithm plays the most important role for the accuracy of the HRFA. Through experimental error estimations, we have demonstrated that the approximate-GCD algorithm proposed by Pan seems to be adequate for the HRFA.

There remain several future applications of the HRFA as follows:

- Approximate-GCD algorithms may be applied to approximate transfer functions of a high order system by a low order one in the critical case, where the transfer function has closely located poles and zeros. It is known that the Routh approximation method has a serious drawback in that the reduced model may approximate the nondominant poles of the system.
- Bivariate rational interpolation computed by means of interpolating branched continued fractions [5] may have undesired singularities [11]. Only multivariate approximate-GCD methods (e.g., [4], [15]) are available in this case.

Acknowledgement

Authors would like to thank referees who gave valuable comments on improving the paper.

References

1. Berrut, J.-P.: Rational Functions for Guaranteed and Experimentally Well-Conditioned Global Interpolation, *Comput. Math. Applic.* **15** (1988), pp. 1–16.
2. Berrut, J.-P. and Mittelmann, H. D.: Lebesgue Constant Minimizing Linear Rational Interpolation of Continuous Function over the Interval, *Comput. Math. Applic.* **33** (1997), pp. 77–86.
3. Braess, D.: *Nonlinear Approximation Theory*, Springer-Verlag, 1986.
4. Corless, R. M., Gianni, P. M., Trager, B. M., and Watt, S. M.: The Singular Value Decomposition for Polynomial Systems, in: Levelt, A. H. M. (ed.), *ISSAC'95*, 1995, pp. 195–207.
5. Cuyt, A. and Wuytack, L.: *Nonlinear Methods in Numerical Analysis*, 1987.
6. Hribernig, V. and Stetter, H. J.: Detection and Validation of Clusters of Polynomial Zeros, *J. Symb. Comp.* **24** (6) (1997), pp. 667–681.
7. Kai, H. and Noda, M. T.: Approximate-GCD and Padé Approximation, in: Shi, H. and Kobayashi, H. (eds), *Proceedings of Asian Symposium on Computer Mathematics*, 1995, pp. 81–89.
8. Kai, H. and Noda, M. T.: Cauchy Principal Value Integral Using Hybrid Integral, *SIGSAM Bulletin* **31** (3) (1997), pp. 37–38.
9. Kai, H. and Noda, M. T.: Hybrid Computation of Cauchy-Type Singular Integral Equations, *SIGSAM Bulletin* **32** (2) (1998), pp. 59–60.
10. Kai, H. and Noda, M. T.: Accuracy Analysis of Hybrid Rational Interpolation, in: *Proceedings of IMACS ACA'98, Electronic Proceedings*, `http://www-troja.fjfi.cvut.cz/aca98/sessions/approximate/kai/index.html`, 1998, pp. 1–8.
11. Kai, H. and Noda, M. T.: Hybrid Computation of Bivariate Rational Interpolation, in: *International Symposium on Symbolic and Algebraic Computation*, poster session, 1999.
12. Karmarkar, N. and Lakshman, Y. N.: Approximate Polynomial Greatest Common Divisors and Nearest Singular Polynomials, in: Lakshman, Y. N. (ed.), *ISSAC'96*, 1996, pp. 35–39.
13. Noda, M. T. and Miyahiro, E.: A Hybrid Approach for the Integration of a Rational Function, *J. CAM* **40** (1992), pp. 259–268.
14. Noda, M. T., Miyahiro, E., and Kai, H.: Hybrid Rational Function Approximation and Its Use in the Hybrid Integration, in: Vichnevetsky, R., Knight, D., and Richter, G. (eds), *Advances in Computer Methods for Partial Differential Equations VII*, IMACS, 1992, pp. 565–571.
15. Ochi, M., Noda, M. T., and Sasaki, T.: Approximate Greatest Common Divisor of Multivariate Polynomials and Its Application to Ill-Conditioned Systems of Algebraic Equations, *J. Information Processing* **14** (3) (1991).
16. Pan, V. Y.: Computation of Approximate Polynomial GCDs and an Extensions, Accepted by Information and Computation, in: *Proc. 9th Annual ACM–SIAM Symp. on Discrete Algorithms*, ACM Press, New York and SIAM Publications, Philadelphia, 1998, pp. 68–77.
17. Rice, J. R.: *The Approximation of Functions* II, Addison-Wesley, 1969, pp. 76–122.
18. Rivlin, T. J.: *An Introduction to the Approximation of Functions*, Blaisdell, 1969, pp. 120–141.
19. Sasaki, T. and Noda, M. T.: Approximate Square-Free Decomposition and Root-Finding of Ill-Conditioned Algebraic Equations, *J. Inf. Proc.* **12** (1989), pp. 159–168.
20. Schönhage, A.: Quasi-GCD Computations, *J. Complexity* **1** (1985), pp. 118–137.
21. Sederberg, T. W. and Chang, G. Z.: Best Linear Common Divisors for Approximate Degree Reduction, *Computer-Aided Design* **25** (1993), pp. 163–168.
22. Van Barel, M. and Bultheel, A.: A New Approach to the Rational Interpolation Problem, *J. Comp. Appl. Math.* **32** (1990), pp. 281–289.

Hensel Construction of $F(x, u_1, \ldots, u_\ell)$, $\ell \geq 2$, at a Singular Point and Its Applications *

Tateaki Sasaki [†] and Daiju Inaba [‡]
sasaki@math.tsukuba.ac.jp inaba@math.tsukuba.ac.jp

[†] Institute of Mathematics, University of Tsukuba
[‡] Doctoral Program in Mathematics, University of Tsukuba
Tsukuba-shi, Ibaraki 305, Japan

Abstract

In 1993, Sasaki and Kako proposed a Hensel-like construction of $F(x, u_1, \ldots, u_\ell), \ell \geq 2$, at a singular point where the conventional generalized Hensel construction breaks down. In this paper, we first extend Sasaki-Kako's method so as to apply to polynomials with vanishing leading coefficients. Then, we investigate a special case that the initial factors are polynomials. We prove that the multivariate polynomial can be decomposed at any singular point into factors which are polynomials in a main variable with coefficients being (infinite) series of rational functions such that $\sum_{k=0}^{\infty}[N_k(u_1, \ldots, u_\ell)/D_k(u_1, \ldots, u_\ell)]$. Here, N_k and D_k are homogeneous polynomials in $u_1 - s_1, \ldots, u_\ell - s_\ell$ and $\mathrm{tdeg}(N_k) - \mathrm{tdeg}(D_k) = k$, where $(s_1, \ldots, s_\ell)$ is the expansion point and *tdeg* denotes the total-degree. The extended Hensel construction can be used to factorization of multivariate polynomials having a singular point at the origin. After performing the extended Hensel construction at the origin, we search for the smallest subsets of Hensel factors such that the product of the members of each subset contains no rational function. Then, we obtain the factorization in $\mathrm{K}\{u_1, \ldots, u_\ell\}[x]$, with K a number field. Next, we search for the smallest subsets such that the product of the members of each subset contains no infinite series. Then, we obtain the factorization in $\mathrm{K}[x, u_1, \ldots, u_\ell]$, without employing the so-called nonzero substitution.

Key words: extended Hensel construction, Hensel construction, generalized Hensel construction, analytic factorization, multivariate factorization, nonzero substitution, polynomial factorization.

1 Introduction

Let K be a number field and $F(x, u_1, \ldots, u_\ell)$ be a square-free polynomial over K. Let $\bar{\mathrm{K}}$ be either K or an algebraic closure of K. The generalized Hensel construction invented by Musser [Mus71] and improved by Wang and Rothschild [WR75] and Moses and Yun [MS73] is a very important operation in computer algebra. However, it breaks down if the expansion point $(s_1, \ldots, s_\ell) \in \bar{\mathrm{K}}^\ell$ is so chosen that $F(x, s_1, \ldots, s_\ell) = G(x)^m$, with $G(x)$ an irreducible polynomial. We note that if $F(x, s_1, \ldots, s_\ell)$ is square-free and the leading coefficient w.r.t. x, of F does not vanish at $(s_1, \ldots, s_\ell)$, hence we have $F(x, s_1, \ldots, s_\ell) = c(x - \alpha_1)\cdots(x - \alpha_n)$, with $n = \deg_x(F)$ and $\alpha_i \neq \alpha_j$ $(\forall i \neq j)$, then the parallel Hensel construction with initial factors $c(x - \alpha_1), (x - \alpha_2), \ldots, (x - \alpha_n)$, gives us the Taylor expansions of the roots w.r.t. x, of polynomial $F(x, u_1, \ldots, u_\ell)$. Therefore, we say that a point $(s_1, \ldots, s_\ell) \in \bar{\mathrm{K}}^\ell$ is a *singular point for the Hensel construction* if $F(x, s_1, \ldots, s_\ell)$ is not square-free and it is a *singular point of the leading coefficient* if the leading coefficient w.r.t. x, of $F(x, u_1, \ldots, u_\ell)$ vanishes at the point. The case $F(x, s_1, \ldots, s_\ell) = G(x)^m$ is the extremal case of singular points.

The Hensel construction at a singular point is important in two points. One is that it leads us to an extension of the Puiseux series solutions of the bivariate algebraic equation $F(x, u_1) = 0$ to the solutions of multivariate one $F(x, u_1, \ldots, u_\ell) = 0$, $\ell \geq 2$. For researches in this direction, see [SK99] and [McD95]. The other is that it leads us to the factorization in $\bar{\mathrm{K}}\{u_1, \ldots, u_\ell\}[x]$, of $F(x, u_1, \ldots, u_\ell)$, where the origin is a singular point. This kind of factorization is closely related with analytic factorization, i.e., factorization in $\bar{\mathrm{K}}\{x, u_1, \ldots, u_\ell\}$, and it is very important in mathematics. For the analytic factorization in $\bar{\mathrm{K}}\{x, u_1\}$, see [Abh90] or [McC97], for example.

For a bivariate polynomial $F(x, u_1)$, Kuo [Kuo89] describes a direct extension of Hensel's lemma in which the expansion point is chosen at a singular point. Kuo's method seems to be derived from ideas of Abhyankar [Abh89], and it gives us the Puiseux series in u_1 in a special case. Kuo's description is quite informal from the viewpoint of computational algorithm, and McCallum [McC97] gives a complete algorithm of the method. All of these works have been done in the context of analytic factorization of $F(x, u_1)$. On the other hand, Sasaki and Kako extended the Hensel construction of $F(x, u_1, \ldots, u_\ell)$, $\ell \geq 2$, to the case of singular expansion point, so as to generalize the Puiseux series solutions of a bivariate algebraic equation to the case of $\ell \geq 2$. Sasaki-Kako's method reduces to Kuo's method if the number of sub-variables is reduced to one. Sasaki and Kako called their method *extended Hensel construction* and the present authors use this naming in this paper.

The extended Hensel construction itself is the same in both $\ell = 1$ and $\ell \geq 2$ cases: we plot all the monomials of F on a two-dimensional plane (see 2 for details), draw the Newton polygon $\mathcal{N}$ for the dots plotted, determine the initial factors by factoring a homogeneous polynomial which is

*Work supported in part by Japanese Ministry of Education, Science and Culture under Grants 12480065.

obtained by summing all the monomials plotted on the rightmost bottom side of $\mathcal{N}$, and construct the Hensel factors by increasing the modulus by $1/\hat{n}$ step ($\hat{n}$ is a positive integer). The results obtained are, however, pretty different in $\ell = 1$ and $\ell \geq 2$ cases. Any homogeneous polynomial $G(x, u_1)$ in x and u_1 is equivalent to $G(x, 1)$, and the number of different monomials of the same degree in $\mathbf{K}[u_1]$ is one (except for the coefficients). These facts make the Hensel factors quite simple in the $\ell = 1$ case. In the $\ell \geq 2$ case, the Hensel factors are expressed in terms of algebraic functions in $u_1, \ldots, u_\ell$, in general, hence they are not easy to handle.

[SK99] investigates only the case of monic polynomials (hence the leading coefficients are not singular). Therefore, in this paper, we first extend Sasaki-Kako's method so as to apply to the polynomials with singular leading coefficients. This extension is straightforward. Then, we investigate a special case that the initial factors are polynomials over $\bar{\mathbf{K}}$. We find that any singular multivariate polynomial can be decomposed into factors which are polynomials in x with coefficients being series of homogeneous rational functions in $u_1, \ldots, u_\ell$, over $\bar{\mathbf{K}}$.

The extended Hensel factors can be used to factorization of a multivariate polynomial having a singular point at the origin. Choosing polynomials as the initial factors, we obtain the Hensel factors which contain rational functions in their coefficients usually. We search for the smallest subsets of factors such that the product of the members of each subset contains no rational function. Then, we obtain the factorization in $\bar{\mathbf{K}}\{u_1, \ldots, u_\ell\}[x]$, of F at a singular point. This factorization is not the analytic factorization for which we must take unit factors in $\bar{\mathbf{K}}\{x, u_1, \ldots, u_\ell\}$ into account. However, the factorization in $\bar{\mathbf{K}}\{u_1, \ldots, u_\ell\}[x]$ has a close relationship to the analytic factorization, and our factorization itself is quite interesting mathematically. Next, we search for the smallest subsets such that the product of the members of each subset contains no infinite series. Then, we obtain the factorization in $\bar{\mathbf{K}}[x, u_1, \ldots, u_\ell]$. Note that, in the conventional multivariate factorization algorithms, we shift the origin if the origin is a singular point. This is called the nonzero substitution, and it usually causes a large expression growth making the computation very time consuming. The nonzero substitution problem has been attacked by by Wang [Wan77] and [KT90] etc., but it seems to the authors that the problem has not been solved satisfactorily. There is no nonzero substitution in our factorization method !

It must be mentioned, however, that the factorization method to be described in this paper is not able to give the irreducible factorization in $\bar{\mathbf{K}}\{u_1, \ldots, u_\ell\}[x]$ except in simple cases. In this paper, we focus our attention on clarifying the extended Hensel construction, and we discuss the irreducible factorization in $\bar{\mathbf{K}}\{u_1, \ldots, u_\ell\}[x]$ elsewhere.

In 2, we survey Sasaki-Kako's method briefly. In 3, we extend Sasaki-Kako's method so that it becomes applicable to the case of singular leading coefficients. In 4, we derive a decomposition theorem which is a main theorem in this paper. In 5, we describe the application of the extended Hensel factors to factorization in $\bar{\mathbf{K}}\{u_1, \ldots, u_\ell\}[x]$ and in $\bar{\mathbf{K}}[x, u_1, \ldots, u_\ell]$, respectively. We clarify the factors which may appear in the denominators of the Hensel factors, and present a strategy for finding the required combinations of Hensel factors efficiently. In 6, we briefly discuss a problem which we must solve to attain the irreducible factorization in $\bar{\mathbf{K}}\{u_1, \ldots, u_\ell\}[x]$.

2 Brief survey of the extended Hensel construction

Let $\mathbf{K}$ be a field of numbers, and let $\bar{\mathbf{K}}$ be either $\mathbf{K}$ or an algebraic closure of $\mathbf{K}$. Let $\mathbf{K}[u_1, \ldots, u_\ell]$, $\mathbf{K}(u_1, \ldots, u_\ell)$ and $\mathbf{K}\{u_1, \ldots, u_\ell\}$ be the ring of polynomials, the field of rational functions and the ring of formal power series, respectively, over $\mathbf{K}$ in variables $u_1, \ldots, u_\ell$. Let $(s_1, \ldots, s_\ell) \in \bar{\mathbf{K}}^\ell$, and we abbreviate $(u_1, \ldots, u_\ell)$ and $(s_1, \ldots, s_\ell)$ to (u) and (s), respectively. Let a given polynomial $F(x, u) \in \mathbf{K}[x, u]$ be square-free, primitive w.r.t. every variable and expressed as

$$F(x,u) = f_n(u)x^n + f_{n-1}(u)x^{n-1} + \cdots + f_0(u)x^0, \quad f_n(u) \neq 0. \tag{2.1}$$

By $\deg(F)$, $\mathrm{lc}(F)$ and $\mathrm{tdeg}(f_i)$, we denote the degree and the leading coefficient of F w.r.t. the main variable x, and the total-degree of f_i w.r.t. $u_1, \ldots, u_\ell$, respectively: if $T = c\,u_1^{e_1} \cdots u_\ell^{e_\ell}$, $c \in \mathbf{K}$, then $\mathrm{tdeg}(T) = e_1 + \cdots + e_\ell$. By $\mathrm{ord}(f_i)$, we denote the order of f_i, i.e., the minimum of the total-degrees of terms of f_i. For the rational function $f(u)/g(u)$, we define the order as $\mathrm{ord}(f/g) = \mathrm{ord}(f) - \mathrm{ord}(g)$. By $\gcd(F, G)$, we denote the greatest common divisor of F and G. By $\mathrm{cont}(F)$ and $\mathrm{pp}(F)$, we denote the content of $F(x, u)$ and the primitive part of $F(x, u)$, respectively, w.r.t. x, i.e., $\mathrm{cont}(F) = \gcd(f_n, f_{n-1}, \cdots, f_0)$ and $\mathrm{pp}(F) = F/\mathrm{cont}(F)$. By $\mathrm{rem}(F, G)$ and $\mathrm{res}(F, G)$, we denote the remainder and the resultant, respectively, w.r.t. x, of polynomials F and G. By $\langle p_1, \ldots, p_l \rangle$ we denote the ideal generated by $p_1, \ldots, p_l$. Let $G(u)$ be a finite or infinite series of rational functions such that

$$\left\{ \begin{array}{c} G(u) = \dfrac{g_0(u)}{d_0(u)} + \dfrac{g_1(u)}{d_1(u)} + \cdots + \dfrac{g_k(u)}{d_k(u)} + \cdots, \\ g_k(u) \text{ and } d_k(u) \text{ are homogeneous polynomials in } \bar{\mathbf{K}}[u], \\ \mathrm{ord}(g_k/d_k) = k \quad (k = 0, 1, 2, \ldots). \end{array} \right. \tag{2.2}$$

By $\bar{\mathbf{K}}\{(u)\}$, we denote the ring of series of homogeneous rational functions of nonnegative orders, such as $G(u)$ in (2.2).

Definition 1 (singular point, singular leading coefficient) *We call the expansion point (s) a singular point for the Hensel construction, or a singular point in short, if $F(x, s)$ is not square-free. If $f_n(s) = 0$ then we say the leading coefficient is singular at (s).*

One may think that we can avoid the case of singular leading coefficients by the well-known transformation

$$\mathcal{T}_{\mathrm{M}} : F(x,u) \mapsto f_n^{n-1} F(x/f_n, u) \stackrel{\mathrm{def}}{=} \tilde{F}(x,u). \tag{2.3}$$

We see $\tilde{F}(x, u) = x^n + f_{n-1}x^{n-1} + f_n f_{n-2}x^{n-2} + \cdots + f_n^{n-1} f_0$. Hence, although $\mathcal{T}_{\mathrm{M}}$ makes the leading coefficient of $\tilde{F}$ non-singular, it makes $\tilde{F}$ highly singular at (s). Therefore, we consider $F(x, u)$ instead of $\tilde{F}(x, u)$.

For multivariate polynomial $\hat{F}(x, u)$ such that $\deg(\hat{F}) = m$ and $\hat{F}(x, 0) = cx^m$ (that is, $\hat{F}$ has a singular point at the origin but its leading coefficient is not singular at the origin), Sasaki and Kako [SK99] proposed to extend the Hensel construction as follows. First, introduce the *total-degree variable* t for sub-variables $u_1, \ldots, u_\ell$ by the transformation $u_i \mapsto t u_i$ $(i = 1, \ldots, \ell)$. (We may introduce the weighted total-degree variable t by the transformation

$u_i \mapsto t^{\omega_i} u_i$ $(i = 1, \ldots, \ell)$, where $\omega_1, \ldots, \omega_\ell$ are positive integers). Next, we define Newton line and Newton polynomial for $\hat{F}(x,u)$, as follows; Fig. 1 in 3 illustrates the Newton line in general cases.

Definition 2 (Newton line $\mathcal{L}_{\rm New}$ and Newton polynomial $\hat{F}_{\rm New}(x,u)$ for $\hat{F}(x,u)$: *the case of nonsingular leading coefficient*)

1. *For each monomial $cx^i t^j u_1^{j_1} \cdots u_\ell^{j_\ell}$ of $\hat{F}(x,tu)$, with $c \in$ K and $j = j_1 + \cdots + j_\ell$, plot a dot at the point (i,j) in the (e_x, e_t)-plane;*
2. *Let $\mathcal{L}_{\rm New}$ be a straight line in (e_x, e_t)-plane, such that it passes the point $(m,0)$ and another dot plotted and that any dot plotted is not below $\mathcal{L}_{\rm New}$;*
3. *Construct $\hat{F}_{\rm New}(x,tu)$ by summing all the monomials which are plotted on $\mathcal{L}_{\rm New}$.*

$\hat{F}_{\rm New}(x,tu)$ is homogeneous w.r.t. x and t^λ, where $-\lambda$ is the slope of the Newton line $(\lambda > 0)$. Then, we define the ideal $\hat{I}_k$ as follows. Let the Newton line be $e_x/m + e_t/\mu = 1$ (i.e., the Newton line $\mathcal{L}_{\rm New}$ and the e_t-axis intersect at $(0,\mu)$), and determine positive integers $\hat{m}, \hat{\mu}$ to satisfy $\hat{\mu}/\hat{m} = \mu/m = \lambda$ and $\gcd(\hat{m}, \hat{\mu}) = 1$, then

$$\hat{I}_k = \langle\, x^m t^{(k+0)/\hat{m}},\ x^{m-1} t^{(k+\hat{\mu})/\hat{m}},\ \cdots,\ x^0 t^{(k+m\hat{\mu})/\hat{m}} \,\rangle. \tag{2.4}$$

Since the Newton polynomial $\hat{F}_{\rm New}(x,u)$ contains two or more terms, we assume that it is factorized as follows (see [SK99] for the case of $\hat{F}_{\rm New}(x,u) = \hat{G}(x,u)^m$).

$$\begin{cases} \hat{F}_{\rm New}(x,tu) = \hat{G}_1^{(0)}(x,tu) \cdots \hat{G}_r^{(0)}(x,tu), \quad r \geq 2, \\ \qquad \gcd(\hat{G}_i^{(0)}, \hat{G}_j^{(0)}) = 1 \ \text{ for any } \ i \neq j. \end{cases} \tag{2.5}$$

We note that $\hat{G}_i^{(0)}(x,tu)$ is usually a polynomial in x and t^λ with coefficients of algebraic functions in $tu_1, \ldots, tu_\ell$, however, $\hat{G}_i^{(0)}(x,tu)$ may be a polynomial in $tu_1, \ldots, tu_\ell$ and we discuss this case in 4. Using $\hat{G}_1^{(0)}, \ldots, \hat{G}_r^{(0)}$ as initial factors, we can construct $\hat{G}_1^{(k)}, \ldots, \hat{G}_r^{(k)}$ $(k = 1, 2, \ldots)$ such that

$$\hat{F}(x,tu) \equiv \hat{G}_1^{(k)}(x,tu) \cdots \hat{G}_r^{(k)}(x,tu) \pmod{\hat{I}_{k+1}}. \tag{2.6}$$

The procedure of construction is the same as that of the generalized Hensel construction, see [SK99] for details.

3 The case of singular leading coefficient

Sasaki-Kako's method mentioned in 2 cannot directly be applied to polynomials with singular leading coefficients. However, it can be made applicable if we modify the method slightly. In the following, we assume that $F(x,u)$ and its leading coefficient are singular at the origin.

$$\begin{cases} F(x,u) = f_n(u)x^n + \cdots + f_1(u)x + f_0(u), \\ \qquad \mathrm{ord}(f_n) = \nu > 0, \quad f_n(0) = 0. \end{cases} \tag{3.1}$$

Definition 3 (Newton line $\mathcal{L}_{\rm New}$ and Newton polynomial $F_{\rm New}(x,u)$ for $F(x,u)$: *the case of singular leading coefficient*)

1. *For each monomial $cx^i t^j u_1^{j_1} \cdots u_\ell^{j_\ell}$ of $F(x,tu)$, with $c \in$ K and $j = j_1 + \cdots + j_\ell$, plot a dot at the point (i,j) in the (e_x, e_t)-plane;*
2. *Let $\mathcal{L}_{\rm New}$ be a straight line in (e_x, e_t)-plane, such that it passes the point (n,ν) and another dot plotted and that any dot plotted is not below $\mathcal{L}_{\rm New}$;*
3. *Construct $F_{\rm New}(x,tu)$ by summing all the monomials which are plotted on $\mathcal{L}_{\rm New}$.*

The slope of the Newton line may be positive, zero, or negative, as illustrated by Fig. 1.

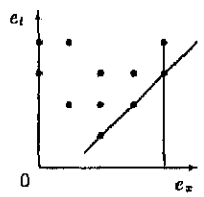


Fig. 1-1

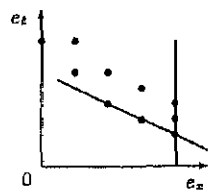


Fig. 1-2

Newton lines for polynomials with singular leading coeffs. The slope may be positive (Fig.1-1) or negative (Fig.1-2)

Let the slope of the Newton line be λ ($\lambda > 0$ in Fig. 1-1 and $\lambda < 0$ in Fig. 1-3), and determine positive integers $\hat{n}$ and $\hat{\nu}$ to satisfy $\hat{\nu}/\hat{n} = |\lambda|$ and $\gcd(\hat{n}, \hat{\nu}) = 1$. According to Lemma 1 in [SK99], if we shift the Newton line by step $1/\hat{n}$ in the e_t-direction successively, all the integer lattice points above the Newton line are ridden by the line. Therefore, we determine polynomial $\bar{F}(x,u,t)$ and ideal $\bar{I}_k$ as follows.

$$\bar{F}(x,u,t) \stackrel{\rm def}{=} \frac{F(x/t^\lambda, tu)}{t^{\nu - n\lambda}}, \tag{3.2}$$

$$\bar{I}_k = \langle\, t^{k/\hat{n}} \,\rangle, \quad k = 1, 2, 3, \cdots. \tag{3.3}$$

Note that the Newton lines become horizontal by the above transformations.

Since the Newton polynomial $\bar{F}_{\rm New}(x,u,t)$ for $\bar{F}(x,u,t)$ has two or more terms, we assume that it is factorized as follows (see [SK99] for the case of $\bar{F}_{\rm New}(x,u,t) = \bar{G}(x,u,t)^m$).

$$\begin{cases} \bar{F}_{\rm New}(x,u,t) = \bar{G}_1^{(0)}(x,u,t) \cdots \bar{G}_r^{(0)}(x,u,t), \quad r \geq 2, \\ \qquad \gcd(\bar{G}_i^{(0)}, \bar{G}_j^{(0)}) = 1 \ \text{ for any } \ i \neq j. \end{cases} \tag{3.4}$$

Then, we can construct $\bar{G}_i^{(k)}(x,u,t)$ $(i = 1, \ldots, r)$, $k = 1, 2, \ldots$, such that

$$\bar{F}(x,u,t) \equiv \bar{G}_1^{(k)}(x,u,t) \cdots \bar{G}_r^{(k)}(x,u,t) \pmod{\bar{I}_{k+1}}. \tag{3.5}$$

For the later use, we describe the construction procedure of $\bar{G}_i^{(k)}(x,u,t)$ briefly. We first calculate "Moses-Yun's polynomials" $\bar{W}_i^{(l)}$ $(i = 1, \ldots, r;\ l = 0, 1, \ldots, n-1)$ to satisfy

$$\begin{cases} \bar{W}_1^{(l)} \cdot \dfrac{\bar{F}_{\rm New}}{\bar{G}_1^{(0)}} + \cdots + \bar{W}_r^{(l)} \cdot \dfrac{\bar{F}_{\rm New}}{\bar{G}_r^{(0)}} = x^l, \\ \deg(\bar{W}_i^{(l)}) < \deg(\bar{G}_i^{(0)}) \quad (i = 1, \ldots, r). \end{cases} \tag{3.6}$$

Suppose that we have constructed $\bar{G}_i^{(k')}$ $(k' = 0, 1, \ldots, k-1)$. Then, we calculate

$$\begin{aligned} \bar{D}^{(k)} &\equiv \bar{F} - \bar{G}_1^{(k-1)} \cdots \bar{G}_r^{(k-1)} \pmod{\bar{I}_{k+1}} \\ &= \bar{d}_n^{(k)} x^n + \bar{d}_{n-1}^{(k)} x^{n-1} + \cdots + \bar{d}_0^{(k)}, \end{aligned} \tag{3.7}$$

and construct $\bar{G}_i^{(k)} \stackrel{\text{def}}{=} \bar{G}_i^{(k-1)} + \delta\bar{G}_i^{(k)}$ $(i = 1, \ldots, r)$ as

$$\delta\bar{G}_i^{(k)} = \bar{W}_i^{(n)} \bar{d}_n^{(k)} + \bar{W}_i^{(n-1)} \bar{d}_{n-1}^{(k)} + \cdots + \bar{W}_i^{(0)} \bar{d}_0^{(k)}. \tag{3.8}$$

Remark 1 *Since the negative slope case is essentially the same as that treated in* [SK99], *we can unify Sasaki-Kako's method and the above construction procedure.*

Example 1 The extended Hensel construction of a bivariate polynomial $F(x, y)$ with singular leading coefficient. For easiness of reading, in this and the following examples, we omit the total-degree variable t and do not show $\bar{F}(x, u, t)$ etc. but show $F(x, u)$ etc.

$$\begin{aligned} F &= x^4y^2 + x^3(3y^2 + y) + x^2(y^3 - 2y^2 + 3y - 2) \\ &\quad + x(3y^3 - 9y^2 - 5y) + (-2y^4 - 5y^3 + 3y^2). \end{aligned} \tag{3.9}$$

The Newton polynomial and its irreducible factorization are as follows.

$$F_{\text{New}} = x^4y^2 + x^3y + 2x^2 = x^2 \cdot (xy + 2) \cdot (xy - 1). \tag{3.10}$$

Putting $G_1^{(0)} = x^2$, $G_2^{(0)} = xy + 2$ and $G_3^{(0)} = xy - 1$, we calculate Moses-Yun's polynomials $W_i^{(l)}$ $(i = 1, 2, 3;\ l = 0, 1, 2, 3)$ as follows.

$$\begin{array}{lll} W_1^{(0)} = -(xy+2)/4, & W_2^{(0)} = -y^2/12, & W_3^{(0)} = y^2/3, \\ W_1^{(1)} = -x/2, & W_2^{(1)} = y/6, & W_3^{(1)} = y/6, \\ W_1^{(2)} = 0, & W_2^{(2)} = -1/3, & W_3^{(2)} = 1/3, \\ W_1^{(3)} = 0, & W_2^{(3)} = 2/(3y), & W_3^{(3)} = 1/(3y). \end{array}$$

We calculate $\delta G_i^{(k)}$ $(k = 1, 2)$ explicitly:

$k = 1$:

$$\begin{aligned} &D^{(1)} = 3x^3y^2 + 3x^2y \Rightarrow d_3^{(1)} = 3y^2,\ d_2^{(1)} = 3y, \\ &\delta G_1^{(1)} = W_1^{(3)} \cdot (3y^2) + W_1^{(2)} \cdot (3y) = 0, \\ &\delta G_2^{(1)} = W_2^{(3)} \cdot (3y^2) + W_2^{(2)} \cdot (3y) = y, \\ &\delta G_3^{(1)} = W_3^{(3)} \cdot (3y^2) + W_3^{(2)} \cdot (3y) = 2y, \end{aligned}$$

$k = 2$:

$$\begin{aligned} &D^{(2)} = -4x^2y^2 - 5xy \Rightarrow d_2^{(2)} = -4y^2,\ d_1^{(2)} = -5y, \\ &\delta G_1^{(2)} = W_1^{(2)} \cdot (-4y^2) + W_1^{(1)} \cdot (-5y) = 5xy/2, \\ &\delta G_2^{(2)} = W_2^{(2)} \cdot (-4y^2) + W_2^{(1)} \cdot (-5y) = y^2/2, \\ &\delta G_3^{(2)} = W_3^{(2)} \cdot (-4y^2) + W_3^{(1)} \cdot (-5y) = -3y^2. \end{aligned}$$

Thus, we obtain the following Hensel factors.

$$\begin{aligned} G_1^{(2)} = x^2 + 5xy/2, \quad G_2^{(2)} &= xy + 2 + y + y^2/2, \\ G_3^{(2)} &= xy - 1 + 2y - 3y^2. \end{aligned}$$

□

4 When initial factors are polynomials

In this section, we confine ourselves to the case that the initial factors are polynomials in x and $u_1, \ldots, u_\ell$. For simplicity, we treat $F(x, tu)$ etc. without the transformation in (3.2). We first define a Newton polygon for $G(x, u) \in \bar{\mathrm{K}}\{(u)\}[x]$.

Definition 4 (Newton polygon and Newton polynomials for $G(x, u)$) *For each term $cx^it^ju_1^{j_1} \cdots u_\ell^{j_\ell}/D(tu)$ of $G(x, tu)$, where $c \in \bar{\mathrm{K}}$, $j = j_1 + \cdots + j_\ell$ and $D(u)$ is a homogeneous polynomial in $u_1, \ldots, u_\ell$ with $\mathrm{ord}(D) = d$, plot a dot at the point $(i, j-d)$ in the (e_x, e_t)-plane. The Newton polygon $\mathcal{N}$ for $G(x, u)$ is a convex hull containing all the dots plotted. Let the bottom sides of $\mathcal{N}$, counted clockwise, be $\mathcal{S}_1, \ldots, \mathcal{S}_\rho$. For each $i \in \{1, \ldots, \rho\}$, Newton polynomial $F_{\mathcal{S}_i}(x, u)$ is defined by the sum of all the terms plotted on $\mathcal{S}_i$.*

Figure 2 shows the Newton polygon for polynomials in the examples in this paper. Note that the Newton line is nothing but the rightmost bottom side of the polygon ($\mathcal{S}_1$ in Fig. 2).

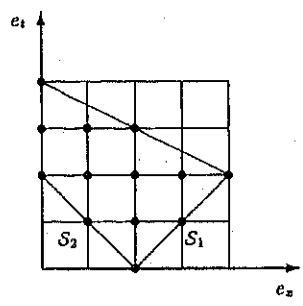


Fig. 2
The Newton polygon for polynomials in Examples 1 ∼ 4.

According to [SK99], Moses-Yun's polynomials $W^{(l)}$ $(l = 0, 1, \ldots, n-1)$ corresponding to an initial factor $G^{(0)}$ are determined uniquely by $G^{(0)}$ and the associated initial factor $H^{(0)} \stackrel{\text{def}}{=} F_{\text{New}}/G^{(0)}$. Furthermore, the corresponding Hensel factors $G^{(k)}$ $(k = 1, 2, \ldots)$ are also determined uniquely by $G^{(0)}$ and $H^{(0)}$, so long as we fix the leading coefficient of $G^{(k)}$. Hence, in the following theoretical analysis, we consider the case that the initial factors are only $G^{(0)}$ and $H^{(0)}$.

Lemma 1 *Let an initial factor $G^{(0)}(x, u)$ be a polynomial in x and $u_1, \ldots, u_\ell$, then the corresponding Moses-Yun's polynomials $W^{(l)}$ $(l = 0, 1, \ldots, n-1)$ and the Hensel factors $G^{(k)}$ $(k = 1, 2, \ldots)$ are elements of $\bar{\mathrm{K}}(u)[x]$:*

$$\begin{cases} W^{(l)}(x, u) \in \bar{\mathrm{K}}(u)[x] & (l = 0, 1, \ldots, n-1), \\ G^{(k)}(x, u) \in \bar{\mathrm{K}}(u)[x] & (k = 1, 2, 3, \ldots). \end{cases} \tag{4.1}$$

The rational function coefficients of $W^{(l)}$ and $G^{(k)} - G^{(k-1)}$ are homogeneous w.r.t. $u_1, \ldots, u_\ell$.

Proof. We can calculate $W^{(l)}$ by applying the extended Euclidean algorithm and the division to $G^{(0)}$ and

$H^{(0)}$. These are rational operations over $\bar{\mathbf{K}}(u)$, hence $W^{(l)} \in \bar{\mathbf{K}}(u)[x]$. Similarly, $G^{(k)}$ is constructed by the addition and the multiplication of terms of F, $G^{(0)}$, $H^{(0)}$, and corresponding Moses-Yun's polynomials which are in $\bar{\mathbf{K}}(u)[x]$. Hence, $G^{(k)} \in \bar{\mathbf{K}}(u)[x]$. Since the coefficients of $G^{(0)}$ and $H^{(0)}$ are homogeneous polynomials in $u_1, \ldots, u_\ell$, and since the extended Euclidean algorithm and the division preserve the homogeneousness, the rational function coefficients of $W^{(l)}$ are homogeneous w.r.t. $u_1, \ldots, u_\ell$ and so are for $G^{(k)} - G^{(k-1)}$. □

Lemma 1 tells us that we have the following Hensel construction of $F(x,u)$:

$$\begin{cases} \dfrac{F(x/t^\lambda, tu)}{t^{\nu-n\lambda}} \equiv G^{(k)}(x/t^\lambda, tu)\, H^{(k)}(x/t^\lambda, tu) \pmod{\tilde{I}_{k+1}}, \\ G^{(k)}(x,u), H^{(k)}(x,u) \in \bar{\mathbf{K}}(u)[x], \quad (k = 1, 2, \ldots). \end{cases} \tag{4.2}$$

Let the Newton polygons for $G^{(\infty)}(x,u)$ and $H^{(\infty)}(x,u)$ be $\mathcal{N}'$ and $\mathcal{N}''$, respectively. Let the sets of bottom sides of $\mathcal{N}'$ and $\mathcal{N}''$, counted clockwise, be $(\mathcal{S}'_1, \cdots, \mathcal{S}'_\rho)$ and $(\mathcal{S}''_1, \cdots, \mathcal{S}''_\rho)$, respectively, where one of the pair $(\mathcal{S}'_i, \mathcal{S}''_i)$ may be of length 0. (If $\mathrm{length}(\mathcal{S}'_i) = 0$ then the side $\mathcal{S}'_i$ is nil). For each $i \in \{1, \ldots, \rho\}$, let $F'_{\mathcal{S}'_i}(x,u)$ and $F''_{\mathcal{S}''_i}(x,u)$ denote the sums of all the terms of $G^{(\infty)}$ and $H^{(\infty)}$ that are plotted on $\mathcal{S}'_i$ and $\mathcal{S}''_i$, respectively. (If $\mathrm{length}(\mathcal{S}'_i) = 0$ then we put $F'_{\mathcal{S}'_i}(x,u) = 1$). Let the coordinates of the left edges of $\mathcal{S}_i$, $\mathcal{S}'_i$ and $\mathcal{S}''_i$ be (n_i, ν_i), (n'_i, ν'_i) and (n''_i, ν''_i), respectively. Putting $n_0 = n$, we express $F_{\mathcal{S}_i}$ as follows. ($f_j^{(0)}(u)$ is the terms plotted on $\mathcal{S}_i$, of $f_j(u)$).

$$F_{\mathcal{S}_i} = f^{(0)}_{n_{i-1}}(u)x^{n_{i-1}} + \cdots + f^{(0)}_{n_i}(u)x^{n_i} \quad (i = 1, \ldots, \rho). \tag{4.3}$$

Lemma 2 *For each $i \in \{1, \ldots, \rho\}$, we have*

$$\begin{cases} \mathrm{length}(\mathcal{S}_i) = \mathrm{length}(\mathcal{S}'_i) + \mathrm{length}(\mathcal{S}''_i), \\ F_{\mathcal{S}_i}(x,u) = F'_{\mathcal{S}'_i}(x,u) \cdot F''_{\mathcal{S}''_i}(x,u). \end{cases} \tag{4.4}$$

Proof. The Newton polygon for the product $G^{(\infty)}H^{(\infty)}$ is $\mathcal{N}$ and $G^{(\infty)}H^{(\infty)}$ contains all the terms of $F_{\mathcal{S}_1}, \ldots, F_{\mathcal{S}_\rho}$. The terms of $G^{(\infty)}$ and $H^{(\infty)}$, which give $F_{\mathcal{S}_i}$ must be plotted on the sides of the same slope, of $\mathcal{N}'$ and $\mathcal{N}''$, respectively; the reason is that, if $\mathcal{S}'_i$ and $\mathcal{S}''_j$ are of different slopes, then the terms of $F'_{\mathcal{S}'_i} \cdot F''_{\mathcal{S}''_j}$ as a whole are not plotted on a line. Therefore, due to the convex property of the Newton polygon, we can pair the sides of $\mathcal{N}'$ and $\mathcal{N}''$ as $(\mathcal{S}'_1, \mathcal{S}''_1), \ldots, (\mathcal{S}'_\rho, \mathcal{S}''_\rho)$, satisfying (4.4). □

Now, $F_{\mathcal{S}_1}$ is of the form $F_{\mathcal{S}_1}(x,u) = x^{n_1} F_1^{(0)}(x,u)$, hence, by factoring $F_1^{(0)}(x,u)$ into relatively prime polynomials, it can be factorized in $\bar{\mathbf{K}}[x,u]$ as follows.

$$\begin{cases} F_{\mathcal{S}_1}(x,u) = x^{n_1} \cdot \mathrm{cont}(F_{\mathcal{S}_1})\, G^{(0)}_{11}(x,u) \cdots G^{(0)}_{1r_1}(x,u), \\ \gcd(G^{(0)}_{1j}, G^{(0)}_{1j'}) = 1 \quad (\forall j \neq j'). \end{cases} \tag{4.5}$$

By applying the extended Hensel construction to $F(x,u)$ with initial factors x^{n_1} and $G^{(0)}_{1j}(x,u)$ $(j = 1, \ldots, r_1)$, $F(x,u)$ can be factorized as

$$F(x,u) = F_2(x,u)\,\mathrm{cont}(F_{\mathcal{S}_1})\, G^{(\infty)}_{11}(x,u) \cdots G^{(\infty)}_{1r_1}(x,u). \tag{4.6}$$

Here, $F_2(x,u)$ is a Hensel factor corresponding to x^{n_1}. Lemma 2 tells us that the bottom sides of the Newton polygon for $F_2(x,u)$ are $\mathcal{S}_2, \ldots, \mathcal{S}_\rho$. Therefore, we have the correspondence $F_{\mathcal{S}_2}(x,u) \iff$ [Newton polynomial for $F_2(x,u)$]. Since $f^{(0)}_{n_1}(u)$ is contained in both the lowest degree term of $F_{\mathcal{S}_2}$ and the highest degree term of $F_{\mathcal{S}_2}$, we have

$$F_{\mathcal{S}_2}(x,u) = [\text{Newton polynomial for } F_2(x,u) f^{(0)}_{n_1}(u)]. \tag{4.7}$$

Factorizing $F_{\mathcal{S}_2}(x,u)$ in $\bar{\mathbf{K}}[x,u]$ similarly, and continuing this procedure, we obtain the following theorem.

Theorem 1 (decomposition theorem) *$F(x,u)$ can be factorized in $\bar{\mathbf{K}}\{(u)\}[x]$ as*

$$\begin{cases} F(x,u) = f_0^{(0)}(u) \displaystyle\prod_{i=1}^{\rho} \left[\hat{g}_i(u) \cdot G^{(\infty)}_{i1}(x,u) \cdots G^{(\infty)}_{ir_i}(x,u) \right], \\ \hat{g}_i(u) = \mathrm{cont}(F_{\mathcal{S}_i}) / f^{(0)}_{n_i}(u) \quad (i = 1, \ldots, \rho), \\ G^{(0)}_{i1} \cdots G^{(0)}_{ir_i} = \mathrm{pp}(F_{\mathcal{S}_i}) \quad (i = 1, \ldots, \rho), \\ \gcd(G^{(0)}_{ij}, G^{(0)}_{i'j'}) = 1 \quad (\forall i \neq i' \text{ or } \forall j \neq j'), \\ G^{(\infty)}_{ij}(x,u) \in \bar{\mathbf{K}}\{(u)\}[x] \quad (i = 1, \ldots, \rho;\ j = 1, \ldots, r_i). \end{cases} \tag{4.8}$$

Proof. The above discussion shows that $F(x,u)$ can be factorized as in the first to the fourth equalities in (4.8). The problem is only the fifth relation in (4.8). We note that $G^{(0)}_{ij} \in \bar{\mathbf{K}}[x,u]$ for any i and j, and the coefficients of $G^{(0)}_{ij}$ are homogeneous rational functions of nonnegative orders. Furthermore, the terms added to $G^{(0)}_{ij}$ by the extended Hensel construction are plotted above the bottom side of the Newton polygon for $G^{(0)}_{ij}$, hence Lemma 1 tells us that the coefficients of the terms are homogeneous rational functions of positive orders. Therefore, we see $G^{(k)}_{ij} \in \bar{\mathbf{K}}\{(u)\}[x]$ for any k. □

The above procedure constructs Hensel factors successively on $\mathcal{S}_1 \Rightarrow \mathcal{S}_2 \Rightarrow \cdots \Rightarrow \mathcal{S}_\rho$. We can also construct Hensel factors successively on $\mathcal{S}_\rho \Rightarrow \mathcal{S}_{\rho-1} \Rightarrow \cdots \Rightarrow \mathcal{S}_1$. In order to do so, we apply the following transformation $\mathcal{T}_{\mathrm{Rev}}$ to $F(x,u)$, perform the extended Hensel construction of the transformed polynomial, and apply the inverse transformation $\mathcal{T}^{-1}_{\mathrm{Rev}}$ to the Hensel factors obtained.

$$\mathcal{T}_{\mathrm{Rev}} : F(x, u_1, \ldots, u_\ell) \mapsto x^{\deg(F)} F(1/x, u_1, \ldots, u_\ell). \tag{4.9}$$

Now, we have two sets of Hensel factors, one contains factors $G^{(\infty)}_{ij}$'s in (4.8), which are constructed from the right to the left of the bottom sides of $\mathcal{N}$, another contains factors $H^{(\infty)}_{ij}$'s below, which are constructed from the left to the right.

$$\begin{cases} F(x,u) = f_n^{(0)}(u) \displaystyle\prod_{i=1}^{\rho} \left[\hat{h}_i(u) \cdot H^{(\infty)}_{i1}(x,u) \cdots H^{(\infty)}_{ir'_i}(x,u) \right], \\ \hat{h}_i(u) = \mathrm{cont}(F_{\mathcal{S}_i}) / f^{(0)}_{n_{i-1}}(u) \quad (i = 1, \ldots, \rho). \end{cases} \tag{4.10}$$

Theorem 2 *Determine the initial factors $G_{i1}^{(0)}, \ldots, G_{ir_i}^{(0)}$ and $H_{i1}^{(0)}, \ldots, H_{ir_i'}^{(0)}$ $(i = 1, \ldots, \rho)$ so that each factor is an irreducible polynomial in $\bar{\mathrm{K}}[x, u]$ or its power. Then $r_i = r_i'$ and $G_{ij}^{(\infty)}(x, u) = U_{ij}(u) H_{ij}^{(\infty)}(x, u)$ $(j = 1, \ldots, r_i)$, where U_{ij} is a unit in $\bar{\mathrm{K}}\{\{u\}\}$.*

Proof. Since the irreducible factorization of $\mathrm{pp}(F_{S_i})$ in $\bar{\mathrm{K}}[x, u]$ is unique up to units in $\bar{\mathrm{K}}$, we have $r_i = r_i'$ and we may assume without loss of generality that $G_{ij}^{(0)} = H_{ij}^{(0)}$ $(j = 1, \ldots, r_i)$. Both $G_{ij}^{(\infty)}$ and $H_{ij}^{(\infty)}$ are factors of $F(x, u)$ in $\bar{\mathrm{K}}\{\{u\}\}[x]$. However, $G_{ij}^{(\infty)} \nmid H_{i'j'}^{(\infty)}$ for any $i \neq i'$ or any $j \neq j'$, because $\gcd(G_{ij}^{(0)}, H_{i'j'}^{(0)}) = 1$. Therefore, we have $G_{ij}^{(\infty)} | H_{ij}^{(\infty)}$, where the division is over $\bar{\mathrm{K}}\{\{u\}\}$. Considering this division along the Newton line for $G_{ij}^{(\infty)}$, we see that the quotient is of the form $1 + \frac{g(u)}{h(u)} \in \bar{\mathrm{K}}\{\{u\}\}$, $\mathrm{ord}(g/h) \geq 1$, hence the quotient is a unit in $\bar{\mathrm{K}}\{\{u\}\}$. □

Remark 2 *Theorem 2 does not imply the uniqueness of the Hensel factors in (4.8) for any weighting of sub-variables; we have different Hensel factors $G_{ij}^{(\infty)}$'s for different weighting $u_i \mapsto t^{\omega_i} u_i$ $(i = 1, \ldots, \ell)$.*

Example 2 We check Theorem 2 by the polynomial used in Example 1.

$$\begin{aligned} F &= x^4y^2 + x^3(3y^2 + y) + x^2(y^3 - 2y^2 + 3y - 2) \\ &\quad + x(3y^3 - 9y^2 - 5y) + (-2y^4 - 5y^3 + 3y^2). \end{aligned}$$

We first determine the Hensel factors from the right to the left. Performing the extended Hensel construction of F up to order 6, as in Example 1, we obtain the following Hensel factors.

$$\begin{aligned} F_2^{(6)} &= x^2 + x(5y/2 + 33y^2/4 + 9y^3/2 - 259y^4/8 \\ &\quad - 4537y^5/32) - 3y^2/2 + y^3/4 + 19y^4/4, \\ G_{11}^{(6)} &= xy + 2 + y + y^2/2 - 5y^3/4 + y^4/2 \\ &\quad + 3y^5/8 - 39y^6/32, \\ G_{12}^{(6)} &= xy - 1 + 2y - 3y^2 - 7y^3 - 5y^4 + 32y^5 + 143y^6. \end{aligned}$$

Since $F_2^{(0)} = x^2$, we can factorize $F_2^{(6)}$ further as follows (we can perform the extended construction only up to order 2, because of lack of the accuracy of $F_2^{(6)}$).

$$\begin{aligned} F_2^{(6)} &\equiv G_{21}^{(2)} \cdot G_{22}^{(2)} \pmod{\langle x^2y^2, xy^3, y^4 \rangle}, \\ G_{21}^{(2)} &= x + 3y + 7y^2 + 5y^3, \\ G_{22}^{(2)} &= x - y/2 + 5y^2/4 - y^3/2. \end{aligned}$$

Next, we perform the extended Hensel construction from the left to the right. We apply the transformation T_{Rev} in (4.9) to F, obtaining $\tilde{F}$ and its Newton polynomial $\tilde{F}_{\mathrm{New}}$ as follows.

$$\begin{aligned} \tilde{F} &= x^4(3y^2 - 5y^3 - 2y^4) + x^3(-5y - 9y^2 + 3y^3) \\ &\quad + x^2(-2 + 3y - 2y^2 + y^3) + x(y + 3y^2) + y^2, \\ \tilde{F}_{\mathrm{New}} &= 3x^4y^2 - 5x^3y - 2x^2 \\ &= (3x^2) \cdot (xy + 1/3) \cdot (xy - 2). \end{aligned}$$

Putting $\tilde{F}_1^{(0)} = 3x^2$, $\tilde{H}_{21}^{(0)} = xy + 1/3$, $\tilde{H}_{22}^{(0)} = xy - 2$, we perform the extended Hensel construction of $\tilde{F}$ up to order 4. Applying T_{Rev}^{-1} to the Hensel factors computed, let the results be $F_1^{(4)} = T_{\mathrm{Rev}}^{-1}\tilde{F}_1^{(4)}$, $H_{21}^{(4)} = T_{\mathrm{Rev}}^{-1}\tilde{H}_{21}^{(4)}$, $H_{22}^{(4)} = T_{\mathrm{Rev}}^{-1}\tilde{H}_{22}^{(4)}$, we obtain

$$\begin{aligned} F_1^{(4)} &= -x^2(3y^2/2) - x(3y/2 + 17y^2/4 - 37y^3/4) \\ &\quad + 3 - 5y - 2y^2, \\ H_{21}^{(4)} &= x(1/3 - 7y/9 + 34y^2/27 \\ &\quad + 155y^3/81 + 250y^4/243) + y, \\ H_{22}^{(4)} &= -x(2 + 5y + 21y^2/2 + 79y^3/4 + 40y^4) + y. \end{aligned}$$

According to Theorem 2, we have the correspondence $\{G_{21}^{(\infty)}, G_{22}^{(\infty)}\} \Longleftrightarrow \{H_{21}^{(\infty)}, H_{22}^{(\infty)}\}$ with ambiguity of units $U_{2j}(u)$ $(j = 1, 2)$. We remove the ambiguity by normalizing the leading coefficients: we divide H_{2j} by $\mathrm{lc}(H_{2j})$, the leading coefficient of H_{2j} $(j = 1, 2)$.

$$\begin{aligned} G_{21}^{(4)} &\Leftarrow H_{21}^{(4)}/\mathrm{lc}(H_{21}^{(4)}) \\ &= x + 3y + 7y^2 + 5y^3 - 32y^4 - 143y^5, \\ G_{22}^{(4)} &\Leftarrow H_{22}^{(4)}/\mathrm{lc}(H_{22}^{(4)}) \\ &= x - y/2 + 5y^2/4 - y^3/2 - 3y^4/8 + 39y^5/32. \end{aligned}$$

Note that we can calculate $G_{2j}^{(k)}$ more economically by performing the Hensel construction from the left to the right. □

5 Factorization in $\bar{\mathrm{K}}\{u\}[x]$ and in $\bar{\mathrm{K}}[x, u]$, $\ell \geq 2$

In this section, we assume that $\ell \geq 2$; if $\ell = 1$ then $G(u)$ in (2.2) is an integral power series in u_1 and the coefficients of $G^{(k)}(x, u_1)$ are polynomials in u_1.

The theme in this section is two kinds of factorization: one is in $\bar{\mathrm{K}}\{u\}[x]$ and the other is in $\bar{\mathrm{K}}[x, u]$. The principle of factorization is as follows. First of all, note that $\bar{\mathrm{K}}[x, u] \subset \bar{\mathrm{K}}\{u\}[x] \subset \bar{\mathrm{K}}\{\{u\}\}[x]$. Therefore, if $F(x, u)$ is factorized into irreducible factors in $\bar{\mathrm{K}}\{\{u\}\}[x]$ then we obtain the irreducible factors in $\bar{\mathrm{K}}\{u\}[x]$ by the products of some irreducible factors in $\bar{\mathrm{K}}\{\{u\}\}[x]$, and also obtain the irreducible factors in $\bar{\mathrm{K}}[x, u]$ similarly.

However, the theory we have developed in the previous sections is not enough to perform the irreducible factorization in $\bar{\mathrm{K}}\{\{u\}\}[x]$; if an initial factor $G_{ij}^{(0)}$ is such that $G_{ij}^{(0)} = \hat{G}^m$, with $m > 1$ and $\hat{G}$ an irreducible polynomial in $\bar{\mathrm{K}}[x, u]$, then the corresponding Hensel factor $G_{ij}^{(\infty)}$ may not be irreducible in $\bar{\mathrm{K}}\{\{u\}\}[x]$. In this paper, we treat only the partial factorization in $\bar{\mathrm{K}}\{\{u\}\}[x]$. In some cases, the extended Hensel factors in $\bar{\mathrm{K}}\{\{u\}\}[x]$ give the irreducible factorization in $\bar{\mathrm{K}}\{u\}[x]$, as the following theorem shows.

Theorem 3 *Assume that the Newton polynomials $F_{S_i}(x, u)$ $(i = 1, \ldots, \rho)$ defined in Definition 4 are square-free. Then, the decomposition of $F(x, u)$ given in Theorem 1 is the irreducible factorization in $\bar{\mathrm{K}}\{\{u\}\}[x]$, so long as, for each $i \in \{1, \ldots, \rho\}$, $\mathrm{pp}(F_{S_i}) = G_{i1}^{(0)} \cdots G_{ir_i}^{(0)}$ is the irreducible factorization in $\bar{\mathrm{K}}[x, u]$.*

Proof. Considering the mapping $F \mapsto F_{S_1} \cdots F_{S_\rho}$, we see that any irreducible factor in $\bar{\mathrm{K}}\{\{u\}\}[x]$, of F corresponds to some factor of $F_{S_1} \cdots F_{S_\rho}$. On the other hand,

Theorem 1 tells us that each irreducible factor in $\bar{\mathbf{K}}[x,u]$, of F_{S_i}, $1 \le i \le \rho$, corresponds to a factor in $\bar{\mathbf{K}}\{(u)\}[x]$, of F. Therefore, we have the one-to-one correspondence between the irreducible factors in $\bar{\mathbf{K}}\{(u)\}[x]$, of F and the irreducible factors in $\bar{\mathbf{K}}[x,u]$, of $F_{S_1}\cdots F_{S_\rho}$. □

Before investigating the factorization method, we remark on two points. The first is on the processing of factors $\hat{g}_1(u),\ldots,\hat{g}_\rho(u)$ in (4.8). This processing is the same as that of leading coefficients of factor polynomials in the conventional factorization algorithms: we attach each $\hat{g}_i(u)$ $(1 \le i \le \rho)$ to $G_{i1}^{(0)}(x,u)$, say, and adjust the leading coefficients of the products of factors which give the required polynomials in $\bar{\mathbf{K}}\{u\}[x]$ or $\bar{\mathbf{K}}[x,u]$. The second is on the representation of the elements of $\bar{\mathbf{K}}\{(u)\}$. Given a homogeneous rational function $N(u)/D(u)$, we reduce it to a unique representation by a suitable method, and one method is as follows. We introduce the lexicographic ordering for the terms in $\bar{\mathbf{K}}[u]$, which orders all the monic monomials in $\bar{\mathbf{K}}[u]$ uniquely. Let $\mathrm{ht}(D)$ denote the highest order monomial of $D(u)$. If $N(u)$ contains a monomial $M(u)$ which is a multiple of $\mathrm{ht}(D)$ then we reduce M/D as

$$\frac{M}{D} \Longrightarrow M/\mathrm{ht}(D) - (M/\mathrm{ht}(D))\cdot\frac{D-\mathrm{ht}(D)}{D}.$$

Note that $M/\mathrm{ht}(D) \in \bar{\mathbf{K}}[u]$. Continuing this reduction until the numerator contains no monomial which is a multiple of $\mathrm{ht}(D)$, we obtain the required representation of $N(u)/D(u)$. (This is nothing but the M-reduction in the Gröbner basis theory). It is easily proved that the result of this reduction is unique. Below, we assume that *each element of* $\bar{\mathbf{K}}\{(u)\}$ *is fully reduced.*

Combining elements of $\bar{\mathbf{K}}\{u\}[x]$ to get an element of $\bar{\mathbf{K}}[x,u]$ is a main theme in the conventional factorization algorithm, and it has been well investigated so far. Therefore, in this paper, we investigate how to combine Hensel factors in $\bar{\mathbf{K}}\{(u)\}[x]$ to get an element of $\bar{\mathbf{K}}\{u\}[x]$.

Definition 5 (integral and rational Hensel factors) *If a Hensel factor $G_{ij}^{(\infty)}(x,u)$ in (4.8) is an integral power series in $u_1,\ldots,u_\ell$ then we call it integral, otherwise rational.*

We first investigate the denominators in the extended Hensel factors. We assume as above that the initial factors are only $G^{(0)}$ and $H^{(0)}$, and we denote the corresponding Moses-Yun's polynomials by $W^{(l)}$ and $V^{(l)}$, respectively:

$$\begin{cases} F_{\mathrm{New}}(x,u) = G^{(0)}(x,u)H^{(0)}(x,u), \\ G^{(0)} = g_{n'}x^{n'} + \cdots + g_0x^0, \\ H^{(0)} = h_mx^m + \cdots + h_0x^0, \end{cases} \quad n = n'+m, \tag{5.1}$$

$$\begin{cases} V^{(l)}G^{(0)} + W^{(l)}H^{(0)} = x^l, \\ \deg(V^{(l)}) < m, \quad \deg(W^{(l)}) < n', \\ l = 0,1,\ldots,n-1. \end{cases} \tag{5.2}$$

The sub-resultant theory tells us that $V^{(0)}$ and $W^{(0)}$ are expressed as

$$V^{(0)} = \begin{vmatrix} g_{n'} & \cdots & g_1 & g_0 & & x^{m-1} \\ & \ddots & \cdots & \ddots & \ddots & \vdots \\ & & g_{n'} & \cdots & g_1 & x^0 \\ h_m & \cdots & h_1 & h_0 & & 0 \\ & \ddots & \cdots & \ddots & \ddots & \vdots \\ & & h_m & \cdots & h_1 & 0 \end{vmatrix} / D, \tag{5.3}$$

where $D = \mathrm{res}(G^{(0)}, H^{(0)})$,

$$W^{(0)} = [\text{replace the last column by } (0,\cdots,0,x^{n'-1},\cdots,x^0)^{\mathrm{T}}]. \tag{5.4}$$

For $l \ge 1$, $V^{(l)}$ and $W^{(l)}$ are calculated as

$$\begin{cases} V^{(l)} = \mathrm{rem}(x^lV^{(0)}, H^{(0)}), \\ W^{(l)} = \mathrm{rem}(x^lW^{(0)}, G^{(0)}). \end{cases} \tag{5.5}$$

In particular, in the case of $H^{(0)} = x^m$, we can express $V^{(l)}$ and $W^{(l)}$ explicitly as

$$\text{for } l \ge m \quad \begin{cases} V^{(l)} = 0, \\ W^{(l)} = x^{l-m}, \end{cases} \tag{5.6}$$

$$\text{for } l < m \quad \begin{cases} V^{(l)} = x^l \cdot G^{(0)}_{\mathrm{Inv}(x^{m-l})}, \\ W^{(l)} = [G^{(0)}_{\mathrm{Inv}(x^{m-l})}\cdot G^{(0)} - 1]/x^{m-l}, \end{cases} \tag{5.7}$$

where $G^{(0)}_{\mathrm{Inv}(x^{m-l})} = [\text{Inverse of } G^{(0)} \text{ modulo } x^{m-l}]$.

Proposition 1 *If the Hensel factors $G^{(\infty)}$ and $H^{(\infty)}$ are rational, the denominators in their rational function coefficients are only $\mathrm{res}(G^{(0)}, H^{(0)})$, $g_{n'}$, h_m, powers of them and their products. In particular, if $H^{(0)} = x^m$ then only g_0 and its powers appear as the denominators.*

Proof. We have $F, G^{(0)}, H^{(0)} \in \bar{\mathbf{K}}[x,u]$ and only $V^{(l)}$ and $W^{(l)}$ may contain rational functions in their coefficients. Eqs. (5.3) and (5.4) show that $V^{(0)}$ and $W^{(0)}$ contain $\mathrm{res}(G^{(0)}, H^{(0)})$ in their denominators, and the division by $H^{(0)}$ and $G^{(0)}$ introduces $h_m, g_{n'}$ and their powers additionally in the denominators of $V^{(l)}$ and $W^{(l)}$ $(l \ge 1)$, respectively. In particular, if $H^{(0)} = x^m$ then $G^{(0)}_{\mathrm{Inv}(x^{m-l})}$ is calculated by the power-series division of 1 by $G^{(0)}$ modulo x^{m-l}, hence $G^{(0)}_{\mathrm{Inv}(x^{m-l})}$ contains only g_0 and its powers as its denominators. For example, $G^{(0)}_{\mathrm{Inv}(x)} = 1/g_0$, $G^{(0)}_{\mathrm{Inv}(x^2)} = -g_1x/g_0^2 + 1/g_0$, and so on. □

Putting the initial factors $G^{(0)}$ and $H^{(0)}$ as $G^{(0)} = F_{S_1}/x^{n_1} = \mathrm{cont}(F_{S_1})\,G_{11}^{(0)}\cdots G_{1r_1}^{(0)}$ and $H^{(0)} = x^{n_1}$, we see from the above proposition that the most denominator factors which may appear in $G_{11}^{(\infty)},\ldots,G_{1r_1}^{(\infty)}$ cancel one another in $G^{(\infty)}$ and only $f_{n_1}^{(0)}$ appears in $G^{(\infty)}$ and $H^{(\infty)}$. Thus, we obtain the following corollary.

Corollary 1 *Except for $f_{n_i}^{(0)}$, the denominator factors appearing in $G_{i1}^{(\infty)},\ldots,G_{ir_i}^{(\infty)}$ in Theorem 1 do not propagate to $G_{j1}^{(\infty)},\ldots,G_{jr_j}^{(\infty)}$ $(j \ge i+1)$.*

Proposition 2 *Let the initial factors be primitive w.r.t. x. The product of two extended Hensel factors, one is integral and the other is rational, is not integral. The product of two extended Hensel factors the denominators of which are essentially different (i.e., different after removing the multiplicity and the common factors) is not integral.*

Proof. Let $G^{(\infty)}$ be integral while $H^{(\infty)}$ be rational, and assume that the rational term that appears first in $H^{(\infty)}$ is T/h which has been fully reduced. If $G^{(\infty)}H^{(\infty)}$ is integral then the denominator h must be canceled in $G^{(0)}T/h$. This means that $G^{(0)}$ must be divided by h, because T has been reduced. This contradicts that $G^{(0)}$ is primitive.

Next, let both $G^{(\infty)}$ and $H^{(\infty)}$ be rational, and let the denominators which appear first in $G^{(\infty)}$ and $H^{(\infty)}$ be g and h, respectively:

$$G^{(\infty)} = G^{(0)} + \cdots + S/g + \cdots,$$
$$H^{(\infty)} = H^{(0)} + \cdots + T/h + \cdots.$$

Here, $g, h \in \bar{\mathbf{K}}[u]$, $S, T \in \bar{\mathbf{K}}[x,u]$, S/g and T/h have been fully reduced, and we assume for the moment that $\gcd(g,h) = 1$. Let $G^{(0)}$ and $H^{(0)}$ be fully reduced w.r.t. h and g, respectively, and the results be $G^{(0)} = hQ_G + \hat{G}^{(0)}$ and $H^{(0)} = gQ_H + \hat{H}^{(0)}$, with $Q_G, Q_H \in \bar{\mathbf{K}}[x,u]$. If $G^{(\infty)}H^{(\infty)}$ is integral then we have

$$G^{(0)}T/h + H^{(0)}S/g \in \bar{\mathbf{K}}[x,u]$$
$$\Longrightarrow \hat{G}^{(0)}T/h + \hat{H}^{(0)}S/g \in \bar{\mathbf{K}}[x,u].$$

Since $\hat{G}^{(0)}$ and T have been reduced w.r.t. h and $\hat{H}^{(0)}$ and S have been reduced w.r.t. g, the above relation requires that $\deg(\hat{G}^{(0)}T) = \deg(\hat{H}^{(0)}S)$ and $\mathrm{lc}(\hat{G}^{(0)}T)/h + \mathrm{lc}(\hat{H}^{(0)}S)/g \in \bar{\mathbf{K}}[u]$. Let $\mathrm{lc}(\hat{G}^{(0)}T)$ and $\mathrm{lc}(\hat{H}^{(0)}S)$ be fully reduced w.r.t. h and g, respectively, and the results be $\mathrm{lc}(\hat{G}^{(0)}T) = q_t h + \hat{t}$ and $\mathrm{lc}(\hat{H}^{(0)}S) = q_s g + \hat{s}$, with $q_t, q_s \in \bar{\mathbf{K}}[u]$. Then, we have $\hat{t}/h + \hat{s}/g \in \bar{\mathbf{K}}[u] \Longrightarrow \exists c \in \bar{\mathbf{K}}[u]$ s.t. $\hat{t}g + \hat{s}h = cgh$. However, since $\hat{t}$ and $\hat{s}$ have been fully reduced w.r.t. h and g, respectively, no term of $\hat{t}g$ and $\hat{s}h$ gives a multiple of $\mathrm{ht}(gh)$, hence we have $c = 0$. Thus, $\hat{t}g + \hat{s}h = 0 \Longrightarrow g|\hat{s}$ and $h|\hat{t} \Longrightarrow \hat{t} = \hat{s} = 0 \Longrightarrow \mathrm{lc}(\hat{G}^{(0)}T) = q_t h$ and $\mathrm{lc}(\hat{H}^{(0)}S) = q_s g \Longrightarrow g \,|\, \mathrm{lc}(\hat{H}^{(0)})$ and $h \,|\, \mathrm{lc}(\hat{G}^{(0)}) \Longrightarrow \mathrm{lc}(\hat{H}^{(0)}) = \mathrm{lc}(\hat{G}^{(0)}) = 0$. Similarly, the other coefficients of $\hat{G}^{(0)}$ and $\hat{H}^{(0)}$ must be zero. This contradicts that $G^{(0)}$ and $H^{(0)}$ are primitive.

Finally, in the case that $g = c\hat{g}$ and $h = c\hat{h}$ with $\gcd(\hat{g}, \hat{h}) = 1$, the factors $\hat{g}$ and $\hat{h}$ must be canceled if $G^{(\infty)}H^{(\infty)}$ is integral, hence the above proof is valid in this case, too. □

Corollary 2 *In the above Proposition 2, an integral (rational) Hensel factor may be a set of Hensel factors the product of which is integral (resp., rational).*

Proposition 2 leads us to the following strategy for combining the rational Hensel factors to obtain an integral Hensel factor.

1. First, for each $i \in \{1, \ldots, \rho\}$, do the following: if two or more Hensel factors on $\mathcal{S}_i$ have a denominator $d_i(u)$ which is peculiar to $\mathcal{S}_i$ then combine Hensel factors containing $d_i(u)$ and eliminate it.
2. Next, if some Hensel factors on different sides $\mathcal{S}_{i_1}, \ldots, \mathcal{S}_{i_m}$ have the same denominator $d(u)$ then combine Hensel factors containing $d(u)$ and eliminate it.

Example 3 Combining rational Hensel factors.

$$\begin{aligned} & F(x,y,z) \\ = \;& x^4(y^2 - z^2) + x^3(y + 3z + 3y^2 + 3z^2) \\ + \;& x^2(-2 + 3y - 4z - 2y^2 + 4yz - 2z^2 + y^3 + 5y^2z + 3z^3) \\ + \;& x^1(-5y - 9y^2 - 5yz - 5z^2 + 3y^3 + y^2z - 5z^3) \\ + \;& (3y^2 - 5y^3 - 7y^2z - yz^2 - 2y^4 - 3y^2z^2 - 3yz^3 - 2z^4). \end{aligned} \tag{5.8}$$

The Newton polynomial $F_{\mathcal{S}_1}$ for F and its irreducible factorization are as follows.

$$\begin{aligned} F_{\mathcal{S}_1} &= x^4(y^2 - z^2) + x^3(y + 3z) - 2x^2 \\ &= x^2 \cdot [x(y - z) + 2] \cdot [x(y + z) - 1]. \end{aligned}$$

Put $F_2^{(0)} = x^2$, $G_{11}^{(0)} = x(y-z) + 2$ and $G_{12}^{(0)} = x(y+z) - 1$. Moses-Yun's polynomials $V_2^{(l)}$ and $W_{12}^{(l)}$ $(l = 0, 1, 2, 3)$ for $F_2^{(0)}$ and $G_{12}^{(0)}$ are calculated as follows.

$$\begin{aligned} V_2^{(0)} &= -[x(y+3z) + 2]/4, & W_{12}^{(0)} &= (y+z)^3/(3y+z), \\ V_2^{(1)} &= -x/2, & W_{12}^{(1)} &= (y+z)^2/(3y+z), \\ V_2^{(2)} &= 0, & W_{12}^{(2)} &= (y+z)/(3y+z), \\ V_2^{(3)} &= 0, & W_{12}^{(3)} &= 1/(3y+z). \end{aligned}$$

As we see, $W_{1j}^{(l)}$ $(j = 1, 2)$ are rational functions in $u_1, \ldots, u_t$. Performing the extended Hensel construction, we see that rational functions appear in Hensel factors of order 2 or more.

$$\begin{aligned} F_2^{(2)} &= x^2 + 5xy/2, \\ G_{11}^{(2)} &= x(y-z) + 2 + (y + 2z) \\ &\quad + (y^2/2 - yz/6 - 4z^2/9) + 4z^3/(27y + 9z), \\ G_{12}^{(2)} &= x(y+z) - 1 + (2y - z) \\ &\quad - (3y^2 + 10yz/3 + 2z^2/9) + 2z^3/(27y + 9z). \end{aligned}$$

The above $G_{11}^{(2)}$ and $G_{12}^{(2)}$ are the Hensel factors on $\mathcal{S}_1$. Since $V_2^{(l)} \in \mathbf{Q}[x,y,z]$, we have $F_2^{(\infty)} \in \mathbf{Q}\{y,z\}[x]$ hence Prop. 2 tells us that $G_{11}^{(\infty)}G_{12}^{(\infty)} \in \mathbf{Q}\{y,z\}[x]$. That is, $F(x,y,z)$ is reducible in $\mathbf{Q}\{y,z\}[x]$ into at least two factors $F_2^{(\infty)}$ and $G_{11}^{(\infty)}G_{12}^{(\infty)}$.

Let us next calculate the Hensel factors on $\mathcal{S}_2$. The Newton polynomial $F_{\mathcal{S}_2}$ on $\mathcal{S}_2$ and its irreducible factorization are as follows.

$$F_{\mathcal{S}_2} = -2x^2 - 5xy + 3y^2 = -2(x + 3y)(x - y/2).$$

Putting $G_{21}^{(0)} = x + 3y$ and $G_{22}^{(0)} = x - y/2$, we calculate the corresponding Moses-Yun's polynomials $W_{2j}^{(l)}$ $(j = 1, 2)$ as follows.

$$\begin{aligned} W_{21}^{(0)} &= -2/(7y), & W_{22}^{(0)} &= 2/(7y), \\ W_{21}^{(1)} &= 6/7, & W_{22}^{(1)} &= 1/7. \end{aligned}$$

By this, we see that the Hensel factors on $\mathcal{S}_2$ may contain only y as the denominator factor. In fact, calculating $G_{2j}^{(4)}$

$(j = 1, 2)$, we see that $G_{2j}^{(\infty)}$ contains y as the denominator factor. Therefore, Prop. 2 tells us that the denominators in $G_{1i}^{(\infty)}$ $(i = 1, 2)$ cannot be eliminated by multiplying $G_{2j}^{(\infty)}$ $(j = 1, 2)$, and we see that $F(x, y, z)$ is irreducible in $\mathbf{C}[x, y, z]$. In addition, we see that $F_2^{(\infty)}$ and $G_{11}^{(\infty)} G_{12}^{(\infty)}$ are irreducible factors in $\mathbf{C}\{y, z\}[x]$. □

Example 4 Combining integral Hensel factors.

$$
\begin{aligned}
& F(x, y, z) \\
= \;& x^4(y^2 - z^2) + x^3(y + 3z + 3y^2 + 3z^2) \\
+ \;& x^2(-2 + 3y - 4z - 2y^2 + 5yz - 2z^2 + y^3 + 6y^2 z + 3z^3) \\
+ \;& x^1(-5y - 9y^2 - 5yz - 5z^2 + 3y^3 + y^2 z - 5z^3) \\
+ \;& (3y^2 - 5y^3 - 7y^2 z - yz^2 - 2y^4 - 3y^2 z^2 - 3yz^3 - 2z^4).
\end{aligned} \tag{5.9}
$$

This $F(x, y, z)$ is the same as in (5.8), except that the coefficients of $4x^2yz$ and $5x^2y^2z$ are increased by 1, hence the initial factors and Moses-Yun's polynomials are also the same as those in Example 3. Using the same symbols and performing the extended Hensel construction up to order 4, we obtain the following results (the denominators disappear magically).

$$
\begin{aligned}
F_2^{(4)} &= x^2 + x(5y/2 + 33y^2/4 - 5yz/2 + 5z^2/2 \\
&\quad + 9y^3/2 - \cdots - 5z^3/2) - 3y^2/2, \\
G_{11}^{(4)} &= x(y - z) + 2 + y + 2z + y^2/2 - yz/2 \\
&\quad - 5y^3/4 - \cdots + z^3/2 + y^4/2 + \cdots - z^4/2, \\
G_{12}^{(4)} &= x(y + z) - 1 + 2y - z - 3y^2 - 3yz \\
&\quad - 7y^3 - \cdots - 2z^3 - 5y^4 + \cdots + 2z^4.
\end{aligned}
$$

Performing the extended Hensel construction of $F_2^{(6)}$ with initial factors $G_{21}^{(0)} = x + 3y$ and $G_{22}^{(0)} = x - y/2$, we see that $G_{2j}^{(4)}$ $(j = 1, 2)$ are integral. Since $G_{1j}^{(4)}$ $(j = 1, 2)$ are integral, we have a chance to get polynomial factors by combining Hensel factors on sides $\mathcal{S}_1$ and $\mathcal{S}_2$. In fact, calculating the products $G_{1j}^{(4)} G_{2j}^{(4)}$ $(j = 1, 2)$, we see that these products divide $F(x, y, z)$ and they are irreducible factors in $\mathbf{Q}[x, y, z]$. □

6 Discussions

In [SK99], the Hensel factors are expressed in terms of algebraic functions which are the roots of Newton polynomials, hence the Hensel factors obtained seem to be difficult to use for practical applications. In this paper, we have shown that, by restricting the initial factors within the polynomials, the resulting Hensel factors become useful in many applications.

However, in order to apply to the irreducible factorization in $\bar{\mathbf{K}}\{u\}[x]$ and in $\bar{\mathbf{K}}[x, u]$, we must solve a problem: how to perform the Hensel construction in $\bar{\mathbf{K}}\{(u)\}[x]$, of $F(x, u)$ for which the Newton polynomial is $F_{\mathrm{New}}(x, u) = G(x, u)^m$, with $G(x, u)$ an irreducible polynomial in $K[x, u]$. In the case of $\ell = 1$, this problem has been solved by introducing the concept of *expansion base*, see [Abh90] or [McC97] for the expansion base. In the case of $\ell \geq 2$, following [SK99], we are considering a different approach.

References

[Abh89] S. S. Abhyankar: Irreducibility criterion for germs of analytic functions of two complex variables. Adv. in Math., Vol. 74, 190-267 (1980).

[Abh90] S. S. Abhyankar: *Algebraic Geometry for Scientists and Engineers.* Number 35 in Mathematical Surveys and Monographs. Providence, RI: American Mathematical Society.

[KT90] E. Kaltofen and B. M. Trager: Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators. J. Symb. Comput., Vol. 9, 301-320 (1990).

[Kuo89] T.-C. Kuo: Generalized Newton-Puiseux theory and Hensel's lemma in $\mathbf{C}[[x, y]]$. Canad. J. Math., Vol. XLI, 1101-1116 (1989).

[McC97] S. McCallum: On testing a bivariate polynomial for analytic reducibility. J. Symb. Comput., Vol. 24, 509-535 (1997).

[McD95] J. McDonald: Fiber polytopes and fractional power series. J. Pure and Applied Algebra, Vol. 104, 213-233 (1995).

[MS73] J. Moses and D. Y. Y. Yun: The EZGCD algorithm. *Proc. 1973 ACM National Conference*, ACM, 159-166 (1073).

[Mus71] D. R. Musser: Algorithms for polynomial factorizations. Ph. D. Thesis, University of Wisconsin, 1971.

[SK99] T. Sasaki and F. Kako: Solving multivariate algebraic equation by Hensel construction. Japan J. Indus. Appl. Math., 16, 257-285 (1999). (This paper was submitted in March, 1993, and the authors received referees' reports in Sep., 1996 and the letter of acceptance in June, 1998.)

[Wan77] P. S. Wang: Preserving sparseness in multivariate polynomial factorization. *Proc. 1977 MACSYMA Users Conference*, 55-61 (1977).

[WR75] P. S. Wang and L. P. Rothschild, Factoring multivariate polynomials over the integers. Math. Comp., 29, 935-950 (1975).

Vol. 41 No. 4 IPSJ Journal Apr. 2000

Regular Paper

"Approximate Zero-points" of Real Univariate Polynomial with Large Error Terms

AKIRA TERUI† and TATEAKI SASAKI†

Let $P(x)$ be a given real univariate polynomial and let $\tilde{P}(x) = P(x) + \Delta(x)$, where $\Delta(x)$ is the sum of error terms, that is, a polynomial with small real unknown but bounded coefficients. We first consider specifying the "existence domain" of the values of $\tilde{P}(x)$, or the domain in which the value of $\tilde{P}(x)$ exists for any real number x, by the coefficient bounds for $\Delta(x)$, and then introduce a concept of an "approximate real zero-point" of $\tilde{P}(x)$. We present a practical method for estimating the existence domain of zero-points of $\tilde{P}(x)$ by applying Smith's celebrated theorem. We next consider counting the number of real zero-points of $\tilde{P}(x)$. If all the zero-points are sufficiently far apart from each other, the number of real zero-points of $\tilde{P}(x)$ is the same as that of $P(x)$, and we derive a condition for which we can assert that $P(x)$ and $\tilde{P}(x)$ have the same number of real zero-points. We calculate the actual number of real zero-points by Sturm's method, which encounters the so-called small leading coefficient problem. For this problem, we show that, under some conditions, small leading terms can be discarded. Furthermore, we investigate four methods for evaluating the effect of error terms on the elements of the Sturm sequence.

1. Introduction

In traditional computer algebra on polynomials, we usually assume that the coefficients of polynomials are given rigorously by integers, rational numbers, or algebraic numbers, and that manipulation on the polynomials is also exact. However, in many practical applications or real-world problems, the coefficients contain errors; that is, polynomials have "error terms." In such cases, many of the traditional algorithms in computer algebra break down.

This paper considers the real zero-points of a real univariate polynomial with error terms, or "approximate polynomial," where the coefficients of error terms can be much larger than the machine epsilon ε_M. In fact, even if the initial errors in coefficients are as small as ε_M, the errors can become much larger than ε_M after the calculation. Furthermore, in approximate algebraic calculation, we handle polynomials with perturbed terms that are much larger than ε_M in general.

If a polynomial $P(x)$ has error terms, we cannot draw the graph of function $y = P(x)$; all we can draw is the "existence domain" of $P(x)$, or the domain in which values of $P(x)$ can exist. Similarly, in such a case, the positions of its zero-points cannot be determined exactly; all we can handle is the domains in which zero-points can exist. Therefore, in this paper, we introduce a concept of an "approximate real zero-point" by defining a minimal interval outside of which no real zero-points can exist. Although the existence domains of real zero-points can be calculated rigorously, we propose methods for calculating them approximately and efficiently by using Smith's theorem on the error bounds of zero-points of a polynomial [11].

Next, we consider calculation of the number of real zero-points of an approximate polynomial by Sturm's method. If all the zero-points are single and well separated, the number of real zero-points is definite unless some error term is quite large, although the positions of zero-points are changed by the error terms. However, in the calculation of the Sturm sequence, the leading coefficient of some element may become too small to determine whether it is equal to zero or not. Since the sign of the leading coefficient in the Sturm sequence is essential in determining the number of real zero-points, this is a serious problem. Our answer to it is that, under some conditions, we may discard the small leading term and continue further calculation of the Sturm sequence. Shirayanagi and Sekigawa [10] also attacked this problem, and proposed an interval arithmetic method with zero rewriting. We will investigate the Sturm sequence with interval coefficients in Section 5.

In Section 2, we investigate the existence domains of the values of a real approximate poly-

† Institute of Mathematics, University of Tsukuba

nomial, then define an approximate real zero-point. In Section 3, we propose a practical method for calculating the existence domains of the zero-points of an approximate polynomial. In Section 4, on the assumption that the polynomial does not have multiple or close zero-points, we derive a sufficient condition for the number of real zero-points not to be changed by error terms. In Section 5, we propose and investigate several methods for checking the effect of the error terms of a given polynomial on the Sturm sequence.

2. Approximate Polynomials and Approximate Real Zero-points

Let $P(x)$ be a given univariate polynomial with real coefficients such that

$$P(x) = c_n x^n + \cdots + c_0 x^0, \tag{1}$$

and let $\tilde{P}(x)$ be a real univariate polynomial such that

$$\tilde{P}(x) = P(x) + \Delta(x), \tag{2}$$

where $\Delta(x)$ represents the sum of real "error terms," that is, a polynomial with unknown small real coefficients. Hence, we know neither $\tilde{P}(x)$ nor $\Delta(x)$; what we know usually is an upper bound for each coefficient in $\Delta(x)$. Representing $\Delta(x)$ as

$$\Delta(x) = \delta_{n-1} x^{n-1} + \cdots + \delta_0 x^0, \tag{3}$$

we assume that we know upper bounds $\varepsilon_{n-1}, \ldots, \varepsilon_0$ such that

$$|\delta_i| \le \varepsilon_i, \quad i = n-1, \ldots, 0. \tag{4}$$

Throughout this paper, we write $\tilde{P}(x \mid \delta_i = \varepsilon_i' \ (i = n-1, \ldots, 0))$ to denote that the values of $\delta_{n-1}, \ldots, \delta_0$ in $\tilde{P}(x)$ are specified as $\delta_{n-1} = \varepsilon_{n-1}', \ldots, \delta_0 = \varepsilon_0'$, and so on.

2.1 Existence Domain of Values of $\tilde{P}(x)$

Supposing that the variable x is fixed to x_0 and that $\delta_{n-1}, \ldots, \delta_0$ are changed continuously under the restrictions in Eq. (4); the value of $\tilde{P}(x_0)$ moves continuously inside an interval. By changing x_0 in $\mathbf{R}$, we will have the minimal domain outside of which there is no possibility of the existence of the value of $\tilde{P}(x)$.

Definition 1 (existence domain) Let x_0 be a real number and δ_i move continuously in the whole interval $[-\varepsilon_i, \varepsilon_i]$ for $i = 0, \ldots, n-1$. Define $P_U(x_0)$ and $P_L(x_0)$ as

$$P_U(x_0) = \max_{\substack{\delta_i \in [-\varepsilon_i, \varepsilon_i] \\ i=0,\ldots,n-1}} \tilde{P}(x_0), \tag{5}$$

$$P_L(x_0) = \min_{\substack{\delta_i \in [-\varepsilon_i, \varepsilon_i] \\ i=0,\ldots,n-1}} \tilde{P}(x_0). \tag{6}$$

By changing the value x_0 in $\mathbf{R}$, we obtain a domain

$$\{[P_L(x), P_U(x)] \mid x \in \mathbf{R}\}. \tag{7}$$

We call this domain the "existence domain of $\tilde{P}(x)$." □

The existence domain of $\tilde{P}(x)$ can be specified rigorously by using $P(x)$.

Lemma 1 Let the value of δ_i in $\tilde{P}(x)$ be changed continuously within the range $[-\varepsilon_i, \varepsilon_i]$, while the values of δ_j's $(j \neq i)$ are fixed, and, for each real value of x, define $P_{U_i}(x)$ and $P_{L_i}(x)$ as

$$P_{U_i}(x) = \max_{\delta_i \in [-\varepsilon_i, \varepsilon_i]} \tilde{P}(x), \tag{8}$$

$$P_{L_i}(x) = \min_{\delta_i \in [-\varepsilon_i, \varepsilon_i]} \tilde{P}(x). \tag{9}$$

Then, we have

$$P_{U_i}(x) = \begin{cases} \tilde{P}(x \mid \delta_i = \varepsilon_i) \\ \quad \text{if } x \ge 0 \text{ or } i \text{ is even}, \\ \tilde{P}(x \mid \delta_i = -\varepsilon_i) \\ \quad \text{if } x \le 0 \text{ and } i \text{ is odd}, \end{cases} \tag{10}$$

$$P_{L_i}(x) = \begin{cases} \tilde{P}(x \mid \delta_i = -\varepsilon_i) \\ \quad \text{if } x \ge 0 \text{ or } i \text{ is even}, \\ \tilde{P}(x \mid \delta_i = \varepsilon_i) \\ \quad \text{if } x \le 0 \text{ and } i \text{ is odd}. \end{cases} \tag{11}$$

Furthermore, for any real value x_0, $\tilde{P}(x_0)$ moves all the points inside $[P_{L_i}(x_0), P_{U_i}(x_0)]$.

Proof. Let x_0 be any real number. We see that $-\varepsilon_i |x_0|^i \le \delta_i |x_0|^i \le \varepsilon_i |x_0|^i$, and since $\delta_i |x_0|^i$ moves all the points inside $[-\varepsilon_i |x_0|^i, \varepsilon_i |x_0|^i]$, we obtain the lemma. □

This lemma directly leads us to the following theorem:

Theorem 2 Let the polynomials $P(x)$ and $\tilde{P}(x)$ be as above; then the functions $P_U(x)$ and $P_L(x)$ in Eq. (7) are given as follows:

$$P_U(x) = \begin{cases} \tilde{P}(x \mid \delta_i = \varepsilon_i \\ \qquad (i = n-1, \ldots, 0)) \\ \qquad \text{for } x \ge 0, \\ \tilde{P}(x \mid \delta_i = (-1)^i \varepsilon_i \\ \qquad (i = n-1, \ldots, 0)) \\ \qquad \text{for } x < 0, \end{cases} \tag{12}$$

$$P_L(x) = \begin{cases} \tilde{P}(x \mid \delta_i = -\varepsilon_i \\ \qquad (i = n-1, \ldots, 0)) \\ \qquad \text{for } x \ge 0, \\ \tilde{P}(x \mid \delta_i = (-1)^{i+1} \varepsilon_i \\ \qquad (i = n-1, \ldots, 0)) \\ \qquad \text{for } x < 0. \end{cases} \tag{13}$$

Furthermore, for any real number x_0, the values of $\tilde{P}(x_0)$ move all the points inside $[P_L(x_0), P_U(x_0)]$. □

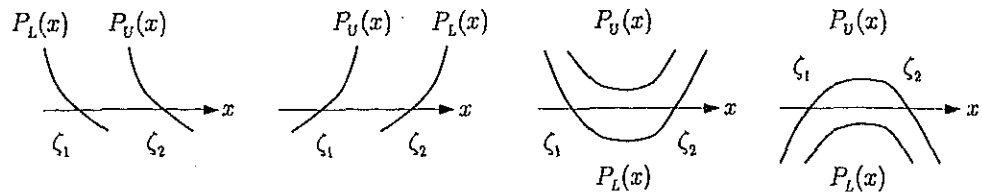


Fig. 1 Existence domain of an approximate real zero-point.

2.2 Approximate Real Zero-points and Their Existence Domains

We first define a concept of "approximate real zero-points" and their existence domains.

Definition 2 (approximate real zero-point) A real number ζ is an "approximate real zero-point of $\tilde{P}(x)$" if there exist numbers $\varepsilon_i' \in [-\varepsilon_i, \varepsilon_i]$ $(i = n-1, \ldots, 0)$ such that $\tilde{P}(\zeta \mid \delta_i = \varepsilon_i' \ (i = n-1, \ldots, 0)) = 0$. Let $[\zeta_{1,1}, \zeta_{1,2}], \ldots, [\zeta_{r,1}, \zeta_{r,2}]$, with $\zeta_{1,1} \le \zeta_{1,2} < \cdots < \zeta_{r,1} \le \zeta_{r,2}$, be the set of all the approximate real zero-points of $\tilde{P}(x)$. Then, we call each interval $[\zeta_{i,1}, \zeta_{i,2}]$, $1 \le i \le r$, an "existence domain" of the approximate real zero-point of $\tilde{P}(x)$. □

Theorem 2 tells us that the existence domains of all the approximate real zero-points can be specified rigorously by drawing graphs of $P_L(x)$ and $P_U(x)$. Suppose $[\zeta_1, \zeta_2]$ is an existence domain of an approximate real zero-point. Since ζ_1 and ζ_2 are real zero-points of $P_U(x)$ and/or $P_L(x)$, and since $P_L(x_0) < P_U(x_0)$ for any real number x_0, the graphs of $P_L(x)$ and $P_U(x)$ around this interval can be classified into one of the following four cases:

(a) $P_L(\zeta_1) = P_U(\zeta_2) = 0$, $P_L(x) < 0$ for $\zeta_1 < x \le \zeta_2$, $P_U(x) > 0$ for $\zeta_1 \le x < \zeta_2$, and there exists $\delta > 0$ such that $P_L(\zeta_1 - x) > 0$ and $P_U(\zeta_2 + x) < 0$ for any $x \in [0, \delta]$,

(b) $P_U(\zeta_1) = P_L(\zeta_2) = 0$, $P_U(x) > 0$ for $\zeta_1 < x \le \zeta_2$, $P_L(x) < 0$ for $\zeta_1 \le x < \zeta_2$, and there exists $\delta > 0$ such that $P_U(\zeta_1 - x) < 0$ and $P_L(\zeta_2 + x) > 0$ for any $x \in [0, \delta]$,

(c) $P_L(\zeta_1) = P_L(\zeta_2) = 0$, $P_U(x) > 0$ for $\zeta_1 \le x \le \zeta_2$, $P_L(x) < 0$ for $\zeta_1 < x < \zeta_2$, and there exists $\delta > 0$ such that $P_L(\zeta_1 - x) > 0$ and $P_L(\zeta_2 + x) > 0$ for any $x \in [0, \delta]$,

(d) $P_U(\zeta_1) = P_U(\zeta_2) = 0$, $P_L(x) < 0$ for $\zeta_1 \le x \le \zeta_2$, $P_U(x) > 0$ for $\zeta_1 < x < \zeta_2$, and there exists $\delta > 0$ such that $P_U(\zeta_1 - x) < 0$ and $P_U(\zeta_2 + x) < 0$ for any $x \in [0, \delta]$.

Figure 1 illustrates these four cases conceptually. Cases (a) and (b) usually correspond to a single zero-point, while Cases (c) and (d) correspond to multiple zero-points.

We now give a simple example of approximate real zero-points and their existence domains. We will see that one of the existence domains is fairly wide, which indicates that the concept of approximate zero-point is indispensable in handling polynomials with error terms.

Example 1 Let $F(x, y)$ be

$$F(x, y) = x^3 - x^2 + y^2. \tag{14}$$

We calculate a singular point of $F(x, y)$ with approximate arithmetic of precision $\varepsilon_M = 1.0 \times 10^{-6}$. First, let us calculate the discriminant $R(y)$ of $F(x, y)$ with respect to x:

$$\begin{aligned} R(y) &= \mathrm{res}(F, dF/dx) \\ &= 27y^4 - 4y^2. \end{aligned} \tag{15}$$

$R(y)$ has zero-points at $y = 0$ and $\pm 2\sqrt{3}/9$. Assume that we have calculated the value of $y = 2\sqrt{3}/9$ approximately as 0.384900. (Note that if $\deg(R) \ge 5$ then use of approximate arithmetic is necessary in general to solve $R(y) = 0$.) Let $P(x)$ and $\tilde{P}(x)$ be

$$\begin{aligned} P(x) &= x^3 - x^2 + (0.384900)^2, \\ \tilde{P}(x) &= P(x) + \delta_0, \end{aligned} \tag{16}$$

where $|\delta_0| \le 1.0 \times 10^{-6}$, and let us calculate the approximate real zero-points of $\tilde{P}(x)$. From Theorem 2, we have

$$\begin{aligned} P_U(x) &= x^3 - x^2 + 0.148149, \\ P_L(x) &= x^3 - x^2 + 0.148147. \end{aligned} \tag{17}$$

$P_U(x)$ has a real zero-point at $x \simeq -0.333334$, and $P_L(x)$ has real zero-points at $x \simeq -0.333332$, 0.665595, and 0.667738. From Definition 2, the existence domains of approximate real zero-points of $\tilde{P}(x)$ are intervals $[-0.333334, -0.333332]$, and $[0.665595, 0.667738]$. Therefore, with an approximate arithmetic of precision $\varepsilon_M = 1.0 \times 10^{-6}$, the singular point (x_0, y_0) of $F(x, y)$ can be specified only vaguely as $y_0 \in [0.384899, 0.384901]$ and $x_0 \in [0.665595, 0.667738]$. □

3. Bounding Existence Domains by Using Smith's Theorem

Although we have defined rigorously the existence domain of only real zero-points, we present in this section a method for bounding the existence domains of both real and complex zero-points by means of discs in the complex plane, because the method is common to both of them.

A key to bounding existence domains is Smith's celebrated theorem. (For the proof, see Smith [11].)

Theorem 3 (Smith) Let $P(x)$ be as above. Let $x_1, \ldots, x_n$ be n distinct numbers in C and $r_1, \ldots, r_n$ be defined as

$$r_j = \left| \frac{nP(x_j)}{a_n \prod_{k=1,\neq j}^{n}(x_j - x_k)} \right|, \quad j = 1, \ldots, n. \tag{18}$$

Let D_j $(1 \leq j \leq n)$ be a disc of radius r_j with its center at x_j. Then, the union $D_1 \cup \cdots \cup D_n$ contains all the zero-points of $P(x)$. Furthermore, if a union $D_1 \cup \cdots \cup D_m$ $(m \leq n)$ is connected and does not intersect with $D_{m+1}, \ldots, D_n$, then this union contains exactly m zero-points. □

3.1 Single Zero-points

Without loss of generality, we assume that P and $\tilde{P}$ are monic. Let $\zeta_1, \ldots, \zeta_n$ and $\tilde{\zeta}_1, \ldots, \tilde{\zeta}_n$ be the zero-points of $P(x)$ and $\tilde{P}(x)$, respectively:

$$P(x) = (x - \zeta_1)(x - \zeta_2) \cdots (x - \zeta_n), \tag{19}$$

$$\tilde{P}(x) = (x - \tilde{\zeta}_1)(x - \tilde{\zeta}_2) \cdots (x - \tilde{\zeta}_n). \tag{20}$$

First, we consider the case in which ζ_1 is a single zero-point such that $|\zeta_1 - \zeta_j| \gg \varepsilon_M$ for $j = 2, \ldots, n$. Let $z_1, \ldots, z_n$ be approximate values for $\zeta_1, \ldots, \zeta_n$, respectively. (Actually, we may determine $z_1, \ldots, z_n$ by solving equation $P(x) = 0$ numerically, and hence approximately, with accuracy ε_M.) Using Theorem 3, we can formally calculate the domain that contains $\tilde{\zeta}_1$ in C, as follows. Let R_1 be

$$R_1 = n \cdot \frac{|\tilde{P}(z_1)|}{\left|\prod_{j=2}^{n}(z_1 - z_j)\right|}, \tag{21}$$

then $\tilde{\zeta}_1$ is contained in the disc of radius R_1 with its center at z_1. Although we cannot calculate $\tilde{P}(z_1)$ explicitly, we have

$$\begin{aligned} |\tilde{P}(z_1)| &\leq |P(z_1)| + |\Delta(z_1)| \\ &\leq |P(z_1)| + \sum_{j=0}^{n-1} \varepsilon_j |z_1|^j. \end{aligned} \tag{22}$$

Therefore, R_1 is bounded as

$$R_1 \leq n \cdot \frac{|P(z_1)| + \sum_{j=0}^{n-1} \varepsilon_j |z_1|^j}{\left|\prod_{j=2}^{n}(z_1 - z_j)\right|}. \tag{23}$$

In ordinary numerical computation, we calculate an error bound by the above formula with $\varepsilon_j = 0$, which gives a good estimate such that the magnitude of the error bound is only several times larger than the true error. Therefore, we expect that the above formula gives a good bound.

3.2 Multiple or Close Zero-points

Next, we consider the case of multiple or close zero-points. Without loss of generality, let $\zeta_1 \simeq \cdots \simeq \zeta_m$ $(m \leq n)$ and assume that $\zeta_{m+1}, \ldots, \zeta_n$ satisfy $|\zeta_j - \zeta_1| \gg \sqrt[n]{\varepsilon_M}$ for $j = m+1, \ldots, n$. In this case, we cannot apply Eq. (23) directly, for the following reason. Let $z_1, \ldots, z_n$ be the same as above and assume that we have calculated them by a numerical method. Then $z_1, \ldots, z_m$ usually satisfy $|z_j - z_1| \simeq \sqrt[n]{\varepsilon_M}$ for $j = 2, \ldots, m$; hence, in Eq. (23), we have

$$\left|\prod_{j=2}^{n}(z_1 - z_j)\right| \simeq \varepsilon_M \cdot \left|\prod_{j=m+1}^{n}(z_1 - z_j)\right|. \tag{24}$$

Therefore, if $|\Delta(z_i)| \gg \varepsilon_M$, an upper bound calculated by Eq. (23) will be an overestimate.

We determine $z_1, \ldots, z_m$ so that the radius R_1 in Eq. (21) becomes as small as possible. (The determination method is the same as that described in the literature; for example, see Iri [5]; the only difference is that our setting of error terms is different from the conventional ones.) We express $P(x)$ as

$$P(x) = (x - \zeta_1) \cdots (x - \zeta_m) \cdot Q(x). \tag{25}$$

From our assumption, we have

$$\begin{aligned} Q(z_1) &= \prod_{j=m+1}^{n} (z_1 - \zeta_j) \\ &\simeq \prod_{j=m+1}^{n} (z_1 - z_j); \end{aligned} \tag{26}$$

hence R_1 defined by Eq. (21) can be approximated as follows:

$$R_1 = n \cdot \frac{\left|\prod_{j=1}^{n}(z_1 - \zeta_j) + \Delta(z_1)\right|}{\left|\prod_{j=2}^{n}(z_1 - z_j)\right|} \tag{27}$$

$$\simeq n \cdot \frac{\left|\prod_{j=1}^{m}(z_1 - \zeta_j) + \Delta(z_1)/Q(z_1)\right|}{\left|\prod_{j=2}^{m}(z_1 - z_j)\right|}. \tag{28}$$

If $z_1, \ldots, z_m$ are distributed equally on a disc of radius r with its center at $(\zeta_1 + \cdots + \zeta_m)/m$, we have

$$\left|\prod_{j=1}^{m}(z_1 - \zeta_j)\right| \approx r^m, \quad \left|\prod_{j=2}^{m}(z_1 - z_j)\right| = mr^{m-1}, \tag{29}$$

and Eq. (28) can be evaluated as

$$R_1 \simeq n \cdot \frac{r^m + C}{mr^{m-1}}, \tag{30}$$

where $C = |\Delta(z_1)/Q(z_1)|$. We can almost minimize the magnitude of R_1 by setting r as

$$r = \sqrt[m]{(m-1)C}. \tag{31}$$

With the above consideration, we calculate an upper bound for R_1 as follows:

(1) Calculate r from Eq. (31).
(2) Let $\beta = (\zeta_1 + \cdots + \zeta_m)/m$ and
$$z_j = \beta + r\exp(2\pi ji/m) \tag{32}$$
for $j = 1, \ldots, m$. The approximate values $z_1, \ldots, z_m$ are distributed equally on a disc of radius r with its center at β.
(3) Substitute $z_1, \ldots, z_m$ into Eq. (23) to obtain a rigorous bound of R_1.

4. Calculating the Number of Real Zero-points of a Real Approximate Polynomial

If a real approximate polynomial has multiple or close zero-points, they may change significantly, or some real zero-points may become complex, when the coefficients are changed slightly. Therefore, it is not adequate to count the number of real zero-points of a real approximate polynomial that may have multiple or close zero-points. On the other hand, if a polynomial has only single zero-points, the number of its real zero-points rarely changes, although their positions may change considerably, when the coefficients are changed slightly. In this section, we focus on calculating the number of real zero-points of a real approximate polynomial containing only single zero-points.

4.1 Sufficient Condition for Fixing the Number of Real Zero-points

We first derive a sufficient condition for asserting that $P(x)$ and $\tilde{P}(x)$ have the same number of real zero-points.

Theorem 4 Let $P(x)$ and $\tilde{P}(x)$ be as in Eqs. (1) and (2), respectively. The number of real zero-points of $\tilde{P}(x)$ is the same as that of $P(x)$ if the discriminant of $\tilde{P}$, or $\mathrm{res}(\tilde{P}, d\tilde{P}/dx)$ does not become zero for any values $\delta_{n-1}, \ldots, \delta_0$ satisfying Eq. (4).

Proof. As the coefficients of $\tilde{P}(x)$ change continuously, the number of real zero-points of $\tilde{P}(x)$ changes only if there exist $\delta_i \in [-\varepsilon_i, \varepsilon_i]$ for $i = 0, \ldots, n-1$ such that $\tilde{P}(x)$ has real multiple zero-points. Its contraposition shows the validity of the theorem. □

Theorem 4 tells us that we can calculate the number of real zero-points of an unknown polynomial $\tilde{P}(x)$ by calculating the number of the real zero-points of $P(x)$, so long as the discriminant $\mathrm{res}(\tilde{P}, d\tilde{P}/dx)$ does not become zero for any values $\delta_{n-1}, \ldots, \delta_0$ satisfying Eq. (4). Therefore, we can check the definiteness of the number of real zero-points by checking whether or not $\mathrm{res}(\tilde{P}, d\tilde{P}/dx)$ becomes zero because of the error terms.

4.2 Problem of Small Leading Coefficient in the Sturm Sequence

Below, the leading coefficient and the degree of $P(x)$ are denoted as $\mathrm{lc}(P)$ and $\deg(P)$, respectively. Let $\zeta_{\max}$ be the maximum of the absolute values of real zero-points of $P(x)$.

The p-norm of $P(x)$, with $P(x)$ given in Eq. (1), is defined as

$$\|P\|_p = \left(\sum_{i=1}^{n}|c_i|^p\right)^{1/p}, \quad p = 1, 2, \ldots, \infty. \tag{33}$$

In this paper, we use the 2-norm for polynomials.

Assuming that $P(x)$ and $\tilde{P}(x)$ satisfy the condition in Theorem 4, $\|P\|_2 \simeq 1$, and $\|\tilde{P}\|_2 \simeq 1$, let us consider calculation of the number of real zero-points of $\tilde{P}(x)$ by application of Sturm's famous method to $P(x)$. Sturm's theorem is as follows (for the proof, see Cohen [3], for example):

Theorem 5 (Sturm) Let $P(x)$ be a real square-free polynomial of degree n, and define a polynomial sequence (the Sturm sequence)

$$(P_0(x), P_1(x), \ldots, P_n(x)) \tag{34}$$

as

$$\begin{cases} P_0 = P(x), \\ P_1 = \frac{d}{dx}P(x), \\ P_i = -\mathrm{rem}(P_{i-2}, P_{i-1}) \\ \quad \text{for } i = 2, \ldots, n, \end{cases} \tag{35}$$

where $\mathrm{rem}(P_{i-2}, P_{i-1})$ denotes the remainder of P_{i-2} divided by P_{i-1}. For a real number x, let $N(x)$ be the number of sign changes, counting

from the left to the right without counting zeros, in the sequence (34), and let s and t be real numbers satisfying $s < t$. Then, the number of the real zero-points of P in the interval $[s, t]$ is equal to $N(s) - N(t)$. □

Note that we can calculate the number of all the real zero-points of P by putting $s = -\infty$ and $t = \infty$ in Theorem 5. In the following, the zeros of the Sturm sequence and its modifications are not counted as sign changes.

Consider calculation of the Sturm sequence of $P(x)$ by means of floating-point arithmetic. During the calculation, we may encounter the leading coefficient problem: (1) it is hard for us to decide whether or not a very small leading coefficient is equal to zero, and (2) the division by a polynomial by a small leading coefficient will cause large cancellation errors in the coefficients of the remainder polynomial.

Let P, s, and t be the same as in Theorem 5. A Sturm sequence of P with $P_n \equiv$ (constant) $\neq 0$ has the following properties (for example, see Cohen [3]):

1° For any real number x, consecutive elements $P_{i-1}(x)$ and $P_i(x)$ do not simultaneously become zero.

2° If $P_j(x) = 0$ for some j $(1 \leq j < n)$ and $x \in \mathbf{R}$, then we have $P_{j-1}(x)P_{j+1}(x) < 0$.

3° P_n has no real zero-point.

With Property 1°, we can calculate the number of sign changes by investigating each P_i separately. Let $P_j(x_j) = 0$ for some $x_j \in \mathbf{R}$; then Property 2° means that P_{j-1} and P_{j+1} have no zero-point in the neighborhood of $x = x_j$. Property 3° is trivial in our case, because $P_n =$ (constant), but it is not trivial for the general Sturm sequence. The above three properties are sufficient for determining the number of real zero-points, and a sequence that has those properties is called a general Sturm sequence.

We note that the sign change of $P_j(x)$ at $x = x_j$, $j \geq 1$, does not affect the number of sign changes in the sequence (34); the value of $N(x)$ changes only when the evaluation point x passes a real zero-point of $P_0(x)$ $(= P(x))$. Furthermore, we can prove the following property of the Sturm sequence:

Lemma 6 Let $P(x)$ and $P_0, \ldots, P_n$ be the same as in Theorem 5, and assume that $P_k(x) = 0$ $(1 < k < n)$ at $x = x_{k,1}, \ldots, x_{k,l_k}$, where $l_k < \deg(P_k)$ and $|x_{k,j}| > \zeta_{\max}$ for $j = 1, \ldots, l_k$. Define $P_k''(x)$ as

$$P_k''(x) = \frac{P_k(x)}{(x - x_{k,1}) \cdots (x - x_{k,l_k})}, \quad (36)$$

and let s and t be real numbers satisfying $s < t$. For real number x, let $N(x)$ be the same as in Theorem 5, and let $N_k''(x)$ be the numbers of sign changes in the sequence

$$(P_0(x), \ldots, P_{k-1}(x), P_k''(x), P_{k+1}(x), \ldots, P_n(x)). \quad (37)$$

Then we have

$$N_k''(s) - N_k''(t) = N(s) - N(t). \quad (38)$$

That is, $N_k''(s) - N_k''(t)$ is equal to the number of real zero-points of $P(x)$ in the interval $[s, t]$.

Proof. Property 1° assures us that there exists a small positive number δ such that $[x_{k,j_1} - \delta, x_{k,j_1} + \delta] \cap [x_{k,j_2} - \delta, x_{k,j_2} + \delta] = \emptyset$ for $1 \leq j_1 < j_2 \leq l_k$ and $P_{k\pm1}(x) \neq 0$ for any $x \in [x_{k,j} - \delta, x_{k,j} + \delta]$. We show $N_k''(x) = N(x)$ for any $x \in [x_{k,j} - \delta, x_{k,j} + \delta]$. Consider a case in which $dP_k/dx < 0$ at $x = x_{k,1}$, $P_{k-1}(x_{k,1}) > 0$, and $P_{k+1}(x_{k,1}) < 0$. Property 2° says that the sequence of signs of polynomials $(P_{k-1}(x), P_k(x), P_{k+1}(x))$ at $x = x_{k,1} - \delta$, $x = x_{k,1}$ and $x = x_{k,1} + \delta$ are $(+, +, -)$, $(+, 0, -)$ and $(+, -, -)$, respectively; hence the number of sign changes of the sequence $(P_{k-1}(x), P_k(x), P_{k+1}(x))$ is equal to 1 for any $x \in [x_{k,1} - \delta, x_{k,1} + \delta]$. Now, assume that $P_k''(x) > 0$ for $x \in [x_{k,1} - \delta, x_{k,1} + \delta]$; then the sequence of signs of polynomials $(P_{k-1}(x), P_k''(x), P_{k+1}(x))$ is $(+, +, -)$ for any $x \in [x_{k,1} - \delta, x_{k,1} + \delta]$. Therefore, we have $N_k''(x) = N(x)$ for any $x \in [x_{k,1} - \delta, x_{k,1} + \delta]$. The other cases can be proved similarly. □

Theorem 7 Assume the same hypotheses as in Lemma 6, and define a polynomial sequence

$$(P_0(x), \ldots, P_{k-1}(x), P_k''(x), \ldots, P_{n''}''(x)) \quad (39)$$

as

$$\begin{cases} P_k'' &= \dfrac{P_k(x)}{(x - x_{k,1}) \cdots (x - x_{k,l_k})}, \\ P_{k+1}'' &= -\mathrm{rem}(P_{k-1}, P_k''), \\ P_i'' &= -\mathrm{rem}(P_{i-2}'', P_{i-1}'') \\ & \quad \text{for } i = k+2, \ldots, n'', \end{cases} \quad (40)$$

where $\deg(P_{n''}'') = 0$. For a real number x, let $N''(x)$ be the number of sign changes in the sequence (39), and let s and t be real numbers satisfying $s < t$. Then, the number of real zero-points of $P(x)$ in the interval $[s, t]$ is equal to $N''(s) - N''(t)$.

Proof. From Lemma 6, we need not consider $x_{k,1}, \ldots, x_{k,l_k}$ for calculating the number of real zero-points of $P(x)$. Let x_k be any zero-point of P_k''; hence $P_{k-1}(x_k) \neq 0$. Then, $P_{k-1}(x_k) \cdot P_{k+1}''(x_k) < 0$ because $-P_{k+1}''(x) =$

$P_{k-1}(x) - Q_k''(x)P_k''(x)$. Repeating this argument for P_{k+1}'', P_{k+2}'', and so on, we see that the new polynomial sequence (39) satisfies Properties 1°, 2°, and 3° described above, and that the sequence (39) is a general Sturm sequence of $P(x)$. Thus, we can count all the real zero-points of $P(x)$ by using the sequence (39). □

Remark 1 Properties 1°, 2°, and 3° are enough to prove Theorem 7, and Lemma 6 is unnecessary. We introduced Lemma 6 to help the reader to understand what happens when large real zero-points of P_k are removed. □

In Theorem 7, calculating the general Sturm sequence by using P_k'' in Eq. (40) is theoretically simple but not practical, because we have to calculate the real zero-points of P_k rigorously. We next show that, if a polynomial has small leading terms, these terms correspond to zero-points of large magnitudes.

Lemma 8 Let $\varepsilon_n, \ldots, \varepsilon_{n-s+1}$ be real numbers such that $0 < |\varepsilon_j| \ll 1$, and, without loss of generality, let $Q(x)$ be

$$Q(x) = \varepsilon_n x^n + \cdots + \varepsilon_{n-s+1}x^{n-s+1} + b_{n-s}x^{n-s} + \cdots + b_0x^0, \quad (41)$$

where $|b_i| \geq 1$ $(i = n-s, \ldots, 0)$ for $b_i \neq 0$. Let $x_1, \ldots, x_n$ be the zero-points of $Q(x)$ such that $|x_1| < \cdots < |x_n|$. Then we have

$$\lim_{\substack{(\varepsilon_n,\ldots,\varepsilon_{n-s+1})\to(0,\ldots,0) \\ j = n-s+1,\ldots,n.}} |x_j| = \infty, \quad (42)$$

Proof. Define $Q_I(x)$ as

$$Q_I(x) = x^n \cdot Q(1/x) = \bar{b}_n x^n + \cdots + \bar{b}_0 x^0, \quad (43)$$

and let $\bar{x}_1, \ldots, \bar{x}_n$ be the zero-points of $Q_I(x)$ with $|\bar{x}_1| < \cdots < |\bar{x}_n|$. Then we have $\bar{b}_{n-j} = \varepsilon_j$ for $j = n, \ldots, n-s+1$ and $\bar{x}_{n-i+1} = 1/x_i$ for $i = 1, \ldots, n$. We have $|\bar{x}_i| \to 0$ $(i = n, \ldots, n-s+1)$ for $|\bar{b}_{n-j}| \to 0$ $(j = n, \ldots, n-s+1)$; hence $|x_i| \to \infty$ for $\varepsilon_j \to 0$. □

Remark 2 Although Lemma 8 is a limiting case of $(\varepsilon_n, \ldots, \varepsilon_{n-s+1}) \to (0, \ldots, 0)$ and is sufficient to prove Theorem 9, we investigate the location of zero-points of $Q_I(x)$ in the appendix. □

Theorem 7 and Lemma 8 lead us to an idea of discarding the small leading terms to calculate a general Sturm sequence in practice. Since the zero-points of $P_k(x)$ are moved slightly by discarding the small leading terms, we must be more careful than in Theorem 7.

Theorem 9 Define $P(x)$ and $\tilde{P}(x)$ as in Eqs. (1) and (2), respectively. Let $(P_0 = P(x), P_1 = dP/dx, P_2, \ldots, P_i, \ldots)$ be the Sturm sequence of $P(x)$ and assume that $P_k(x)$ has small leading terms as

$$P_k(x) = \varepsilon_{k,n_k}x^{n_k} + \cdots + \varepsilon_{k,n_k-s+1}x^{n_k-s+1} + b_{k,n_k-s}x^{n_k-s} + \cdots + b_{k,0}x^0, \quad (44)$$

where

$$\max\{|\varepsilon_{k,n_k}|, \ldots, |\varepsilon_{k,n_k-s+1}|\} \ll \min_{b_{k,j}\neq 0}\{|b_{k,n_k-s}|, \ldots, |b_{k,0}|\}.$$

Define a polynomial sequence

$$(P_0(x), \ldots, P_{k-1}(x), P_k'(x), \ldots, P_{n'}'(x)) \quad (45)$$

as

$$\begin{cases} P_k' = b_{k,n_k-s}x^{n_k-s} + \cdots + b_{k,0}x^0, \\ P_{k+1}' = -\mathrm{rem}(P_{k-1}, P_k'), \\ P_i' = -\mathrm{rem}(P_{i-2}', P_{i-1}') \\ \quad \text{for } i = k+2, \ldots, n', \end{cases} \quad (46)$$

where $\deg(P_{n'}') = 0$. For a real number x, let $N'(x)$ be the number of sign changes in the sequence (45), and let s and t be real numbers such that $s < -\zeta_{\max}$ and $\zeta_{\max} < t$. Then, if $\tilde{P}(x)$, $P_{k-1}(x)$, and $P_k(x)$ satisfy the following two conditions, the number of real zero-points of $\tilde{P}(x)$ is equal to $N'(s) - N'(t)$:

(1) The resultant $\mathrm{res}(\tilde{P}, P_k)$ does not become zero for any values $\delta_{n-1}, \ldots, \delta_0$ satisfying Eq. (4) or when the values of $\varepsilon_{k,n_k}, \ldots, \varepsilon_{k,n_k-s+1}$ are changed to zero.

(2) The resultant $\mathrm{res}(P_{k-1}, P_k)$ does not become zero when the values of $\varepsilon_{k,n_k}, \ldots, \varepsilon_{k,n_k-s+1}$ are changed to zero.

Proof. Even if $P_k(x)$ has real zero-points whose magnitudes are larger than that of any zero-point of $\tilde{P}(x)$, Lemma 8 and Condition (1) assure us that these real zero-points will be "safely removed" from $P_k(x)$ by changing the values of $\varepsilon_{k,n}, \ldots, \varepsilon_{k,n-s+1}$ to 0. We also see that the removed zero-points do not affect the calculation of the number of real zero-points, as Theorem 7 shows. Next, changing the values of $\varepsilon_{k,j}$'s to 0 will change the values of the other zero-points of $P_k(x)$ slightly. However, Condition (2) assures us that none of the real zero-points of $P_k(x)$ passes through the real zero-points of $P_{k-1}(x)$; hence the sequence (45) is a general Sturm sequence. Therefore, as in Theorem 7, we can calculate the number of real zero-points of $\tilde{P}(x)$ by using the sequence (45). □

Theorem 9 tells us that the problem of small leading coefficients reduces to checking whether or not any resultants become zero. We will propose several methods for this in Section 5.

We explain Theorem 9 by means of an example with exact arithmetic.

Example 2 Let $P(x)$ and $\tilde{P}(x)$ be

$$\begin{aligned} P(x) &= x^5 + 4x^4 + \tfrac{6401}{1000}x^3 \\ &\quad - 20x^2 + 5x + 1, \\ \tilde{P}(x) &= P(x) + \delta_{0,4}x^4 \\ &\quad + \delta_{0,3}x^3 + \cdots + \delta_{0,0}x^0, \end{aligned} \tag{47}$$

where numbers $\delta_{0,4}, \ldots, \delta_{0,0}$ are unknown but bounded as

$$|\delta_{0,j}| \leq \varepsilon = 1/10000. \tag{48}$$

We obtain $(P_0, \ldots, P_5)$, the Sturm sequence of $P(x)$, as follows:

$$\begin{aligned} P_0 &= P(x), \\ P_1 &= \tfrac{d}{dx}P(x) \\ &= 5x^4 + 16x^3 + \tfrac{19203}{1000}x^2 \\ &\quad -40x + 5, \\ P_2 &= -\tfrac{1}{2500}x^3 + \tfrac{94203}{6250}x^2 \\ &\quad - \tfrac{52}{5}x - \tfrac{1}{5}, \\ P_3 &= -\tfrac{7099837085603}{1000}x^2 \\ &\quad + 4898974540x + 94210995, \\ P_4 &= -\tfrac{1838986143841703970}{5040768664210370074987360 9}x \\ &\quad + \tfrac{581470528239934409}{5040768664210370074987360 9}, \\ P_5 &= -(31566508567667286525829957694414724087925197085 57) \\ &\quad / (33818700372417803243846409931137609000000). \end{aligned} \tag{49}$$

Therefore, we have $N(-\infty) - N(\infty) = 3$.

In Eq. (49), P_2 has a small leading coefficient. (Correspondingly, $P_2(x)$ has a real zero-point at $x \simeq 37680.5$.) The conditions in Theorem 9 are satisfied as follows. First, the existence domains of approximate zero-points of $\tilde{P}(x)$ in the neighborhood of $x = 0$ are the intervals $[-0.12992, -0.12989]$, $[0.44536, 0.44541]$, and $[0.19803, 0.19810]$, while the existence domains of approximate zero-points of $P_2(x)$ when we change the value of the leading coefficient continuously from $-1/2500$ to 0 are the intervals $[-0.01877227\cdots, -0.01877227\cdots]$, $[0.708722, 0.708735]$, and $[37680.5, \infty)$. Therefore, the existence domains of the real zero-points of $\tilde{P}(x)$ and $P_2(x)$ do not overlap; hence we have $\mathrm{res}(\tilde{P}, P_2) \neq 0$. Second, the existence domains of approximate zero-points of $P_1(x)$ are the intervals $[0.134731, 0.134738]$, and $[0.910227, 0.910260]$. Therefore, the existence domains of the real zero-points of $P_1(x)$ and $P_2(x)$ do not overlap; hence we have $\mathrm{res}(P_1, P_2) \neq 0$. Since $P(x)$, $\tilde{P}(x)$, P_1, and P_2 satisfy the conditions in Theorem 9, we can calculate $P_2', \ldots, P_4'$ as follows:

$$\begin{aligned} P_2' &= \tfrac{94203}{6250}x^2 - \tfrac{52}{5}x - \tfrac{1}{5}, \\ P_3' &= \tfrac{14367059719609325}{835976753303427}x \\ &\quad - \tfrac{18170016322960675}{3343907013213708}, \\ P_4' &= (65440159831618155883480530106785213) \\ &\quad / (33025984797891324206068900312900000). \end{aligned} \tag{50}$$

We have $N'(-\infty) - N'(\infty) = 3 = N(-\infty) - N(\infty)$. □

5. Evaluating the Effects of Error Terms

Theorems 4 and 9 show that some important problems in counting the number of approximate real zero-points can be reduced to checking whether or not some resultants become zero owing to the error terms. In this section, we consider how to evaluate errors in the resultant of an approximate univariate polynomial. We investigate four methods: (1) evaluating the "subresultant determinant" by using Hadamard's inequality, (2) calculating the Sturm sequence with the coefficients of interval numbers, (3) solving a linear system on polynomial coefficients and evaluating errors in the solution by a standard method in numerical analysis, and (4) calculating the Sturm sequence with parametric error terms. The experiments were performed with GAL (General Algebraic Language/Laboratory, a LISP-based computer algebra system) on NS-LISP (Nara Standard LISP) running on a SPARC Station 5 (CPU: microSPARC II, 70 MHz) and SunOS 4.1.4.

5.1 Evaluation of the Subresultant Determinant

Except for the overall signs of polynomials, the Sturm sequence is the same as the polynomial remainder sequence (PRS) for which the subresultant theory has been developed. (For subresultant theory, see Mishra[7], for example.) With this theory, we can express the elements in the Sturm sequence by the determinants of the coefficients of two consecutive elements. Let $(P_0 = P, P_1 = dP/dx, P_2, \ldots, P_{k-1}, P_k, \ldots)$ be a Sturm sequence, and assume that

$$\begin{aligned} P_{k-1}(x) &= a_l x^l + \cdots + a_0 x^0, \\ P_k(x) &= \varepsilon_m x^m + \cdots \\ &\quad \cdots + \varepsilon_{m-s+1}x^{m-s+1} \\ &\quad + b_{m-s}x^{m-s} + \cdots + b_0 x^0, \end{aligned} \tag{51}$$

where

$$\begin{aligned} &\max\{|\varepsilon_{k,n_k}|, \ldots, |\varepsilon_{k,n_k-s+1}|\} \\ &\ll \min_{b_{k,j}\neq 0}\{|b_{k,n_k-s}|, \ldots, |b_{k,0}|\} \end{aligned}$$

as before.

Let $\mathrm{S}_i(P_{k-1}, P_k)$ be the following determinant:

$$\mathrm{S}_i(P_{k-1}, P_k) = \begin{vmatrix} a_l \cdots & \cdots & \cdots & \cdots \\ & \ddots & & \\ & & a_l & \cdots & \cdots \\ \varepsilon_m \cdots & \varepsilon_{m-s+1} & b_{m-s} & \cdots \\ & \ddots & & \ddots & \ddots \\ & & \varepsilon_m & \cdots & \varepsilon_{m-s+1} \\ \cdots & \cdots & a_{l-2i+1} & x^{i-1}P_{k-1} \\ & & \vdots & \vdots \\ \cdots & \cdots & a_{l-i} & x^0 P_{k-1} \\ \cdots & \cdots & b_{m-2i+1} & x^i P_k \\ & & \vdots & \vdots \\ b_{m-s} \cdots & & b_{m-i+1} & x^0 P_k \end{vmatrix}. \quad (52)$$

$\mathrm{S}_i(P_{k-1}, P_k)$ is called the i-th subresultant of $P_{k-1}(x)$ and $P_k(x)$, and we have $P_{k+i}(x) = \gamma_i \mathrm{S}_i(P_{k-1}, P_k)$, with γ_i a constant. For example, if $\deg(P_{k-1}) = \deg(P_k) + 1$, we have

$$P_{k+1}(x) = \mathrm{S}_1(P_{k-1}, P_k) = \begin{vmatrix} a_l & a_{l-1} & P_{k-1}(x) \\ \varepsilon_m & \varepsilon_{m-1} & xP_k(x) \\ & \varepsilon_m & P_k(x) \end{vmatrix}. \quad (53)$$

Below, we consider only the leading coefficients of P_{k+1}, P_{k+2}, and so on. Applying Hadamard's inequality to the subresultant, we can bound the effect of $\varepsilon_m, \ldots, \varepsilon_{m-s+1}$ on $\mathrm{lc}(P_{k+i})$, as follows:

Proposition 10 Define P'_k and L as follows.

$$P'_k = P_k - (\varepsilon_m x^m + \cdots + \varepsilon_{m-s+1} x^{m-s+1}) = b_{m-s}x^{m-s} + \cdots + b_0, \quad (54)$$

$$L = \|P_k\|_2^{(i-1)} \times \left\{ (i-s)\,|a_l|^s \,\|P_{k-1}\|_2^{(i-s)} + \sum_{j=1}^{s} |a_l|^{(j-1)} \|P_{k-1}\|_2^{(i-j+1)} \right\}. \quad (55)$$

If $\mathrm{lc}(\mathrm{S}_i(P_{k-1}, P_k)) \neq 0$ and

$$\{|\varepsilon_m| + \cdots + |\varepsilon_{m-s+1}|\} \cdot L < |a_l{}^s \cdot \mathrm{lc}(\mathrm{S}_i(P_{k-1}, P'_k))|, \quad i = s, \ldots, m, \quad (56)$$

then

$$\mathrm{lc}(\mathrm{S}_i(P_{k-1}, P_k)) \times a_l{}^s \cdot \mathrm{lc}(\mathrm{S}_i(P_{k-1}, P'_k)) > 0. \quad (57)$$

Proof. Note that

$$\mathrm{lc}(\mathrm{S}_i(P_{k-1}, P_k)) = \begin{vmatrix} a_l \cdots & \cdots & \cdots & \cdots \\ & \ddots & & \\ & & a_l & \cdots & \cdots \\ \varepsilon_n \cdots & \varepsilon_{m-s+1} & b_{m-s} & \cdots \\ & \ddots & & \ddots & \ddots \\ & & \varepsilon_m & \cdots & \varepsilon_{m-s+1} \\ & & \cdots & \cdots & a_{l-2i} \\ & & & & \vdots \\ & & \cdots & \cdots & a_{l-i-1} \\ & & \cdots & \cdots & b_{m-2i} \\ & & & & \vdots \\ & & b_{m-s} & \cdots & b_{m-i} \end{vmatrix}, \quad (58)$$

and

$$\mathrm{lc}(\mathrm{S}_i(P_{k-1}, P'_k)) = \begin{vmatrix} a_l & \cdots & \cdots & \cdots & a_{l-2i+s} \\ & \ddots & & & \vdots \\ & & a_l & \cdots & a_{l-i-1} \\ b_{m-s} & \cdots & \cdots & \cdots & b_{m-2i} \\ & \ddots & & & \vdots \\ & & b_{m-s} & \cdots & b_{m-i} \end{vmatrix}, \quad (59)$$

where $a_j = b_j = 0$ for $j < 0$. By expanding the determinant in Eq. (58) with respect to the $(i+1)$-th row as

$$\begin{vmatrix} \cdots & & \cdots \\ \varepsilon_m \cdots \varepsilon_{m-s+1} & b_{m-s} \cdots & b_{m-2i} \\ \cdots & & \cdots \end{vmatrix} = \begin{vmatrix} \cdots & & \cdots \\ \varepsilon_m \cdots \varepsilon_{m-s+1} & 0 \cdots & 0 \\ \cdots & & \cdots \end{vmatrix} + \begin{vmatrix} \cdots & & \cdots \\ 0 \cdots 0 & b_{m-s} \cdots & b_{m-2i} \\ \cdots & & \cdots \end{vmatrix}, \quad (60)$$

and expanding the last determinant similarly, we finally obtain

$$\mathrm{lc}(\mathrm{S}_i(P_{k-1}, P_k)) = a_l{}^s \cdot \mathrm{lc}(\mathrm{S}_i(P_{k-1}, P'_k)) + \sum_{j=1}^{i+1} \det(R_{i,j}), \quad (61)$$

where

$$R_{i,j} =$$

$$\begin{pmatrix}
a_l & \cdots & \cdots & \cdots & \cdots \\
 & \ddots & \cdots & \cdots & \cdots \\
 & & a_l & \cdots & \cdots \\
 & & & b_{m-s} & \cdots \\
 & & & & \ddots \\
 & & & & \\
 & & & \varepsilon_m & \cdots \\
 & & & & \varepsilon_m \\
 & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
b_{m-s} & \cdots & \cdots & \cdots & \cdots \\
\varepsilon_{m-s+1} & 0 & \cdots & \cdots & \cdots \\
\cdots & \varepsilon_{m-s+1} & b_{m-s} & \cdots & \cdots \\
\ddots & \cdots & \ddots & \ddots & \\
 & \varepsilon_m & \cdots & \varepsilon_{m-s+1} & \\
\cdots & \cdots & a_{l-2i} & & \\
 & & \vdots & & \\
\cdots & \cdots & a_{l-i-1} & & \\
\cdots & \cdots & b_{m-2i} & & \\
 & & \vdots & & \\
\cdots & \cdots & b_{m-2i+j-2} & & \\
\cdots & \cdots & 0 & & \\
\cdots & \cdots & b_{m-2i+j} & & \\
 & & \vdots & & \\
b_{m-s} & \cdots & b_{m-i} & &
\end{pmatrix} \tag{62}$$

Expanding $\det(R_{i,j})$ with respect to the $(i+j)$-th row, or the row

$$(0 \cdots 0\, \varepsilon_m \cdots \varepsilon_{m-s+1}\, 0 \cdots 0), \tag{63}$$

we have

$$\begin{aligned}
&\det(R_{i,j}) \\
&= (-1)^{i+2j}\varepsilon_m \det(\tilde{R}_{i,j,j}) + \cdots \\
&\quad \cdots + (-1)^{i+2j+s}\varepsilon_{m-s+1} \det(\tilde{R}_{i,j,j+s}),
\end{aligned} \tag{64}$$

where $\tilde{R}_{i,j,q}$ is a $2i \times 2i$ submatrix obtained by removing the $(i+j)$-th row and the q-th column from $R_{i,j}$. After removing several top-left diagonal elements a_l's of $\tilde{R}_{i,j,q}$, and applying Hadamard's inequality to $\det(\tilde{R}_{i,j,q})$, with inequalities $|a_l|^2 + |a_{l-1}|^2 + \cdots + |a_{l-2i}|^2 \le \|P_{k-1}\|_2^2$ and $|\varepsilon_m|^2 + \cdots + |\varepsilon_{m-s+1}|^2 + |b_{m-s}|^2 + \cdots + |b_{m-2i}|^2 \le \|P_k\|_2^2$, we finally obtain the following inequality:

$$\begin{aligned}
&|\det(R_{i,j})| \\
&\le M_i\{|a_l|^{(j-1)}\, \|P_{k-1}\|_2^{(i-j+1)}\}, \\
&\qquad j = 1, \ldots, s,
\end{aligned} \tag{65}$$

$$\begin{aligned}
&|\det(R_{i,j})| \\
&\le M_i\{|a_l|^{s}\, \|P_{k-1}\|_2^{(i-s)}\}, \\
&\qquad j = s+1, \ldots, i+1,
\end{aligned} \tag{66}$$

where

$$M_i = \{|\varepsilon_m| + \cdots + |\varepsilon_{m-s+1}|\} \cdot \|P_k\|_2^i. \tag{67}$$

From the assumption (56), we have

$$\begin{aligned}
&\left|\sum_{j=1}^{i+1} \det(R_{i,j})\right| \le \sum_{j=1}^{i+1} |\det(R_{i,j})| \\
&\le \{|\varepsilon_m| + \cdots |\varepsilon_{m-s+1}|\} \cdot L \\
&< |a_l{}^s \cdot \mathrm{lc}(\mathrm{S}_i(P_{k-1}, P_k'))|.
\end{aligned} \tag{68}$$

Therefore, from Eqs. (61) and (68), we obtain Eq. (57). □

From the fundamental theorem of subresultants [2], we have

$$\begin{aligned}
&\mathrm{S}_i(P_{k-1}, P_k') = P_{k+h}'\, \mathrm{lc}(P_{k+h}')^{d_{k+h-1}-1} \\
&\times \prod_{l=1}^{h} \Big\{ \mathrm{lc}(P_{k+l-1}')^{(d_{k+l-2}+d_{k+l-1})} \\
&\times (-1)^{(n_{k+l-2}-n_{k+h})(n_{k+l-1}-n_{k+h})} \Big\},
\end{aligned} \tag{69}$$

where $h = i - s$, $n_{k+j} = \deg(P_{k+j}')$ and $d_j = n_j - n_{j+1}$. Therefore, we can calculate $\mathrm{lc}(\mathrm{S}_i(P_{k-1}, P_k'))$ easily from $\mathrm{lc}(P_{k+h}')$.

Proposition 10 shows that, so long as $\varepsilon_m, \ldots, \varepsilon_{m-s+1}$ satisfy the condition (56), discarding terms $\varepsilon_m x^m, \ldots, \varepsilon_{m-s+1}x^{m-s+1}$ in P_k does not change the signs of leading coefficients of the subresultants $\mathrm{S}_i(P_{k-1}, P_k')$ for $i = 0, \ldots, m-s-1$. However, in actual calculation of the Sturm sequence, the number L in Eq. (55) seems to become too large, hence the condition (56) is not useful in practice.

5.2 Utilization of Interval Arithmetic

In this method, we transform the coefficients of the given polynomial into interval numbers each of which includes the corresponding error, and calculate the Sturm sequence by using interval arithmetic.

By observing how the widths of intervals increased during the calculation, we found that the increase of the width of each interval was about one decimal-digit for each remainder computation. In fact, the division of polynomials of degree difference 1 requires two "polynomial × number" multiplications and two polynomial subtractions. The width of an interval is increased to about twice that of the original interval by one arithmetic operation if the

operands are of almost the same widths; hence the width increases by about $2^4 = 16$ times after the polynomial division. Thus, for a polynomial of degree 10, for example, the width of an interval in the last element of the Sturm sequence may become about 10^{10} times larger than the initial widths, which shows that this method is not useful in practice.

5.3 Standard Method in Numerical Analysis

In numerical analysis, we have a good method of error estimation for the solution of a system of linear equations. Calculation of the resultant can be reduced to solving a linear system.

Usually, the norm of vectors and matrices are defined as follows. Let $\boldsymbol{x} = (x_1, \ldots, x_m)^T$ be a vector in $\mathbf{R}^m$. Then, the p-norm of $\boldsymbol{x}$ is defined as

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{m} |x_i|^p\right)^{1/p}, \quad p = 1, 2, \infty. \tag{70}$$

Let $A = (a_{ij})$ be a real (m, m)-matrix. Then, by using the norm of a vector, we define the p-norm of A as

$$\|A\|_p = \max_{\boldsymbol{x} \neq 0} \frac{\|A\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p}. \tag{71}$$

In this paper we use only $\|A\|_1$ and $\|A\|_\infty$.

Let $F(x)$ and $G(x)$ be

$$F(x) = f_m x^m + \cdots + f_0 x^0, \quad f_m \neq 0, \tag{72}$$

$$G(x) = g_n x^n + \cdots + g_0 x^0, \quad g_n \neq 0, \tag{73}$$

where $m \geq n$. Calculation of the PRS is equivalent to eliminating the terms of higher degrees of F and G to derive R_s, a polynomial of degree s, for $0 \leq s \leq n-1$. For each R_s, there exist polynomials U_s and V_s such that

$$\begin{aligned} &U_s F + V_s G = R_s, \\ &\deg(U_s) \leq n - s - 1, \\ &\deg(V_s) \leq m - s - 1. \end{aligned} \tag{74}$$

We consider calculating $R_0 = \mathrm{res}(F, G)$. Let U_0 and V_0 be expressed as

$$U_0 = u_{n-1} x^{n-1} + \cdots + u_0 x^0, \tag{75}$$

$$V_0 = v_{m-1} x^{m-1} + \cdots + v_0 x^0. \tag{76}$$

From the relation $U_0 F + V_0 G = R_0$, we obtain a system of linear equations on the coefficients in U_0 and V_0, as follows:

$$\begin{pmatrix} f_m & & & g_n & & \\ \vdots & f_m & & \vdots & g_n & \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & f_m & \vdots & \vdots & \vdots & g_n \\ f_0 & \vdots & \vdots & \vdots & g_0 & \vdots & \vdots & \vdots \\ & f_0 & \vdots & \vdots & & g_0 & \vdots & \vdots \\ & & \ddots & \vdots & & & \ddots & \vdots \\ & & & f_0 & & & & g_0 \end{pmatrix} \times \begin{pmatrix} u_{n-1} \\ u_{n-2} \\ \vdots \\ u_0 \\ v_{m-1} \\ v_{m-2} \\ \vdots \\ v_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ R_0 \end{pmatrix}. \tag{77}$$

U_0 and V_0 can be normalized in any way so long as U_0 and V_0 satisfy the above relation. Therefore, we normalize U_0 and V_0 as $u_{n-1} = g_n$ and $v_{m-1} = -f_m$. With this normalization, we can rewrite the relation (77) as

$$\begin{pmatrix} f_m & & & g_n & & \\ \vdots & \ddots & & \vdots & \ddots & \\ \vdots & \vdots & f_m & \vdots & \vdots & g_n \\ f_1 & \vdots & \vdots & f_m\, g_1 & \vdots & \vdots & g_n \\ f_0 & \ddots & \vdots & \vdots & g_0 & \ddots & \vdots & \vdots \\ & \ddots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ & & f_1 & \vdots & & & g_1 & \vdots \\ & & f_0 & f_1 & & & g_0\, g_1 \end{pmatrix} \begin{pmatrix} u_{n-2} \\ \vdots \\ \vdots \\ u_0 \\ v_{m-2} \\ \vdots \\ \vdots \\ v_0 \end{pmatrix} = \begin{pmatrix} g_{n-1} f_m - f_{m-1} g_n \\ \vdots \\ g_{n-m} f_m - f_0 g_n \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{78}$$

where $g_j = 0$ for $j < 0$, and

$$R_0 = f_0 u_0 + g_0 v_0. \tag{79}$$

The linear system (78) is of the form

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{80}$$

where A is a "coefficient matrix," and $\boldsymbol{x}$ and $\boldsymbol{b}$ are vectors of unknowns and given numbers, respectively. We briefly describe a perturbation

Table 1 Condition number of the matrix in Eq. (78) computed for 10 polynomials with random-number coefficients.

Degree of $P(x)$	Condition number					
	1-norm			∞-norm		
	Maximum	Minimum	Average	Maximum	Minimum	Average
10	8.73×10^3	1.69×10^2	2.55×10^3	7.96×10^3	2.92×10^2	2.99×10^3
20	2.57×10^6	4.44×10^3	2.95×10^5	8.51×10^5	1.83×10^3	1.08×10^5
30	1.16×10^7	4.97×10^4	2.46×10^6	5.97×10^7	2.45×10^4	1.18×10^6
40	5.37×10^7	1.48×10^5	7.44×10^6	4.76×10^7	6.01×10^4	6.00×10^6
50	1.47×10^8	1.56×10^5	2.25×10^7	6.42×10^7	7.38×10^4	8.09×10^6

theory for linear system. (The theory can be found in various works on numerical analysis; see Higham [4] for example.) Assume that b has an error Δb that causes an error Δx_1 in the solution x. Then we have

$$A(x + \Delta x_1) = b + \Delta b. \tag{81}$$

Using Eq. (80), we can easily evaluate the magnitude of Δx_1 as

$$\frac{\|\Delta x_1\|}{\|x\|} \leq \|A\| \, \|A^{-1}\| \, \frac{\|\Delta b\|}{\|b\|}. \tag{82}$$

Furthermore, assume that A has an error ΔA and that the error of x becomes $\Delta x_1 + \Delta x_2$, as follows:

$$(A + \Delta A)(x + \Delta x_1 + \Delta x_2) = b + \Delta b. \tag{83}$$

Using Eq. (81), we derive the following evaluation of Δx_2.

$$\frac{\|\Delta x_2\|}{\|x + \Delta x_1 + \Delta x_2\|} \leq \|A\| \, \|A^{-1}\| \, \frac{\|\Delta A\|}{\|A\|}. \tag{84}$$

Equations (82) and (84) lead us to the following evaluation:

$$\frac{\|\Delta x_1\| + \|\Delta x_2\|}{\|x\| + \|\Delta x_1\| + \|\Delta x_2\|} \leq \|A\| \, \|A^{-1}\| \left\{ \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right\}. \tag{85}$$

The number $\|A\| \, \|A^{-1}\|$, which is called the "condition number," specifies how the initial errors are magnified in the solution.

Although we did not consider rounding errors in floating-point arithmetic in the above evaluation, the evaluation of rounding errors can easily be included by adding ΔR, a term representing rounding errors, into A. It is known that, if we solve Eq. (80) by Gaussian elimination with pivoting, for example, the errors $\Delta x_1 + \Delta x_2$ in the solution x are well bounded by Eq. (85) (see Higham [4]).

Applying Eq. (85) to the linear system (78), we can bound the errors $|\delta_{u_0}|$ and $|\delta_{v_0}|$ of the solutions u_0 and v_0, due to the perturbations δ_{f_i} of f_i $(i = 0, \ldots, m)$ and δ_{g_j} of g_j $(j = 0, \ldots, n)$. Equation (79) tells us that if $|f_0 u_0 + g_0 v_0| \gg |f_0 \cdot \delta_{u_0}|, |g_0 \cdot \delta_{v_0}|$ then we can say definitely that $R_0 \neq 0$ for the perturbations of the coefficients of F and G. If $|f_0 u_0 + g_0 v_0| \ll |f_0 u_0|, |g_0 v_0|$ then this case corresponds to F and G having mutually close zero-points, and the above method cannot be applied to such cases. If $|f_0 u_0 + g_0 v_0|$ is not small, then we can apply the above method so long as $|\delta_{u_0}|$ and $|\delta_{v_0}|$ are not large. Equation (85) shows that the measure of largeness of $|\delta_{u_0}|$ and $|\delta_{v_0}|$ is the condition number. Therefore, in order to check whether or not the above method is useful, we check the largeness of the condition number for polynomials of degrees from 10 to 50. We generate a real univariate polynomial $P(x)$ with random coefficients, and construct the matrix in the left-hand-side of Eq. (78) by putting $F = P$ and $G = dP/dx$. We generate each coefficient c of $P(x)$ to satisfy $|c| \leq 10$. We set $\deg(P) = 10, 20, 30, 40, 50$, and generate 10 polynomials for each degree. We used the LAPACK library [1] linked to GAL to estimate the condition number (for estimating the condition number, see Natori [8], for example).

Table 1 shows the result of computations. For each degree of polynomial, we show the maximum, minimum, and average values of our estimates of 10 condition numbers. We see from this result that, for a polynomial of degree 10, for example, the error in $\mathrm{res}(P, dP/dx)$ may become 10^3 or 10^4 times larger than the error in the initial polynomial. Although these numbers are rather large, they are much smaller than the increase of the interval width explained above.

5.4 Calculating Error Terms Parametrically

The method described in this subsection gives good estimates of errors in the Sturm sequence, but the calculated value does not give the rigorous error bound.

For simplicity, we assume that $P(x)$ is monic in Eq. (1), and express $\tilde{P}(x)$ in Eq. (2) as

Table 2 $\|\tilde{P}_n(x,\delta_{n-1},\ldots,\delta_0)\|/\|\tilde{P}_n(x,0,\ldots,0)\|$ for 10 polynomials, where $\tilde{P}_n$ is the last element of the Sturm sequence.

Degree of $\tilde{P}(x)$	Polynomial norm					
	1-norm			∞-norm		
	Maximum	Minimum	Average	Maximum	Minimum	Average
10	2.57×10^4	1.02×10^2	5.97×10^3	1.52×10^4	1.68×10^2	5.00×10^3
20	2.83×10^5	1.34×10^5	1.81×10^5	3.65×10^5	3.44×10^5	2.62×10^5
30	2.68×10^9	7.01×10^8	1.13×10^9	4.75×10^9	1.78×10^9	1.76×10^9
40	3.50×10^{13}	2.99×10^{11}	5.39×10^{12}	1.60×10^{14}	1.42×10^{11}	3.65×10^{12}
50	2.80×10^{17}	7.81×10^{15}	9.19×10^{16}	2.47×10^{17}	1.32×10^{16}	1.36×10^{17}

Table 3 Computing times for calculating Sturm sequences with and without parameterized error terms.

Degree of $\tilde{P}(x)$	Computing time (msec.)					
	With error terms			Without error terms		
	Maximum	Minimum	Average	Maximum	Minimum	Average
10	70	50	55	10	< 10	< 10
20	420	400	403	10	< 10	< 10
30	1420	1330	1357	20	10	11
40	3280	3210	3244	50	10	31
50	6080	6030	6050	50	30	37

$$\begin{aligned}&\tilde{P}(x,\delta_{n-1},\ldots,\delta_0)\\&=x^n+(c_{n-1}+\delta_{n-1})x^{n-1}+\cdots\\&\qquad\cdots+(c_0+\delta_0)x^0,\end{aligned}\tag{86}$$

where $\delta_{n-1},\ldots,\delta_0$ are parameters representing errors in the coefficients. Exact calculation of the Sturm sequence of a parametric polynomial $\tilde{P}$ exactly is extremely time-consuming, because $\tilde{P}$ is $(n+1)$-variate. However, if we neglect all the quadratic and higher-order terms with respect to $\delta_{n-1},\ldots,\delta_0$, then the computation cost is only $O(n)$ times larger than that of a numerical Sturm sequence. Therefore, we calculate the i-th element $\tilde{P}_i$ of the Sturm sequence as

$$\begin{aligned}&\tilde{P}_i(x,\delta_{n-1},\ldots,\delta_0)\simeq\tilde{P}_i(x,0,\ldots,0)\\&+\tilde{P}_{i,n-1}(x,0,\ldots,0)\delta_{n-1}+\cdots\\&\qquad\cdots+\tilde{P}_{i,0}(x,0,\ldots,0)\delta_0,\end{aligned}\tag{87}$$

where $\tilde{P}_{i,j}=\partial\tilde{P}_i/\partial\delta_j$ $(j=n-1,\ldots,0)$. Then, by neglecting the terms of order $O(\delta^2)$, we can approximately bound the effect of error terms fairly well, as

$$\begin{aligned}&|\tilde{P}_i-P_i|\\&\lesssim|\tilde{P}_{i,n-1}(x,0,\ldots,0)|\,\varepsilon_{n-1}+\cdots\\&\qquad\cdots+|\tilde{P}_{i,0}(x,0,\ldots,0)|\,\varepsilon_0,\end{aligned}\tag{88}$$

where |(polynomial)| denotes a polynomial with the coefficients replaced by their absolute values.

Actually, the calculation is performed by introducing the total-degree variable t for $\delta_{n-1},\ldots,\delta_0$ as $\delta_i\to\delta_i t$ $(i=0,\ldots,n-1)$. We calculate the Sturm sequence only up to the total-degree 1, and substitute 1 for t after the calculation.

We calculated the Sturm sequences with and without parameterized error terms. For this experiment, we used the same polynomials as in Section 5.3.

Table 2 shows the value $\|\tilde{P}_n(x,\delta_{n-1},\ldots,\delta_0)\|/\|\tilde{P}_n(x,0,\ldots,0)\|$, where $\tilde{P}_n(x,\delta_{n-1},\ldots,\delta_0)$ is the last element of the Sturm sequence, and Table 3 shows the computing times of Sturm sequences with and without parametric errors. In Table 2, for each degree of polynomial, we show the maximum, the minimum, and the average of 10 ratios. Note that the values in Table 2 show how the initial errors are magnified by the computation of Sturm sequence, just as the values in Table 1 show. Comparing with Table 1, we see that the numbers are too large for polynomials of higher degrees. Table 3 shows the maximum, minimum, and average values of the computation times for ten examples. We see that, very roughly speaking, the computation time for a parameterized sequence is about $\deg(P)$ times larger than that for a numerical sequence. These results indicate that we can use this method only for polynomials of low or medium degrees.

6. Discussion

In this paper we have considered the real zero-points of a real univariate polynomial with error terms whose coefficients may be much larger than ε_M. For such an approximate polynomial, we introduced the concept of an

"approximate real zero-point" and proposed a method for calculating the existence domains of zero-points fairly accurately and simply.

Next, we considered how to calculate the number of real zero-points of an approximate polynomial by Sturm's method. We gave a sufficient condition for the number of real zero-points to be definite. We also derived a sufficient condition for the small leading coefficients in the Sturm sequence to be discarded, and showed that these problems can be reduced to a problem to that of estimating the errors in the resultants of univariate polynomials.

Finally, in order to estimate the errors in the Sturm sequence, we investigated four methods: (1) evaluating the "subresultant determinant" by using Hadamard's inequality, (2) calculating the Sturm sequence with coefficients of interval numbers, (3) solving a linear system on polynomial coefficients and evaluating errors in the solution by a standard method in numerical analysis, and (4) calculating the Sturm sequence with parametric error terms. Method 1 is theoretically correct, but the calculated upper bound is too large, and with method 2 the width of each interval number grows too rapidly during the calculation of the Sturm sequence; hence methods 1 and 2 do not seem to be useful in practice. Method 3 gives a rather practical estimation, and thus seems to be useful in practice. Method 4 gives the errors rather accurately, and we have seen that calculating the resultant by PRS gives much larger errors than method 3. This means that the errors contained in the resultant depend on which method we have used to calculate the resultant, and method 3 seems to be the best for evaluating the errors.

We still have a problem in cases where $P(x)$ has multiple or close zero-points. Let us briefly mention what happens if $P(x)$ has close zero-points. Let $\|.\|$ be an appropriate norm of a polynomial defined by Eq. (33), and assume that $\|P\| = 1$ and P_k contains m close zero-points of closeness δ, $0 < \delta \ll 1$, around the origin. Then, Sasaki and Sasaki[9] tell us that $\|P_k\| = O(\delta^0)$ and $\|P_{k+1}\| = O(\delta^2)$, $\|P_{k+2}\| = O(\delta^3), \ldots, \|P_{k+m}\| = O(\delta^{m+1})$. Therefore, if these close zero-points can be separated and counted as m single zero-points, we must have $\|P_{k+m}\| \gg \varepsilon_M$ or $\delta \gg \sqrt[m+1]{\varepsilon_M}$. On the other hand, if we change coefficients of $P(x)$ slightly, the positions of these close zero-points are changed considerably. Thus, the treatment of close zero-points is not easy and remains an open problem.

Acknowledgments The authors thank to anonymous referees for their valuable comments.

References

1) Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Croz, J.D., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S. and Sorensen, D.: *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia (1995).
2) Brown, W.S. and Traub, J.F.: On Euclid's Algorithm and the Theory of Subresultants, *J. ACM*, Vol.18, No.4, pp.505–514 (1971).
3) Cohen, H.: *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics, Vol.138, Springer-Verlag, Berlin (1993).
4) Higham, N.J.: *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia (1996).
5) Iri, M.: *Numerical Analysis* (in Japanese), Asakura Publishing, Tokyo (1981).
6) Mignotte, M.: *Mathematics for Computer Algebra*, Springer-Verlag (1992).
7) Mishra, B.: *Algorithmic Algebra*, Texts and Monographs in Computer Science, Springer-Verlag, New York (1993).
8) Natori, M.: *Numerical Analysis and Its Applications* (in Japanese), Corona Publishing, Tokyo (1990).
9) Sasaki, T. and Sasaki, M.: Analysis of Accuracy Decreasing in Polynomial Remainder Sequence with Floating-point Number Coefficients, *J. Inform. Process.*, Vol.12, No.4, pp.394–403 (1989).
10) Shirayanagi, K. and Sekigawa, H.: An Interval Method Based on Zero Rewriting and Its Application to Sturm's Algorithm (in Japanese), *Trans. Inst. Electronics, Inform. and Comm. Engineers A*, Vol.J80-A, No.5, pp.791–802 (1997).
11) Smith, B.T.: Error Bounds for Zeros of a Polynomial Based upon Gerschgorin's Theorems, *J. ACM*, Vol.17, No.4, pp.661–674 (1970).

Appendix: On the Zero-points of Eq. (43)

Let $0 < \varepsilon \ll 1$ and let $P(x)$ be

$$P(x) = c_n x^n + \cdots + c_{m+1}x^{m+1} + x^m + \varepsilon_{m-1}x^{m-1} + \cdots + \varepsilon_0, \quad (89)$$

where $n > m$ and $c_n, \ldots, c_{m+1}, \varepsilon_{m-1}, \ldots, \varepsilon_0$ are numbers such that

$$\max\{|c_n|, \ldots, |c_{m+1}|\} = 1,\ c_n \neq 0,\quad |\varepsilon_{m-i}| \leq (\sqrt[m]{\varepsilon})^i\ (i = 1, \ldots, m). \quad (90)$$

We choose ε to satisfy $\sqrt[m]{\varepsilon} = \max\{\sqrt[i]{|\varepsilon_{m-i}|} \mid i = 1, \ldots, m\}$. Putting $e = \sqrt[m]{\varepsilon}$, we prove the following theorem in this appendix:

Theorem 11 Let $\zeta_1, \ldots, \zeta_n$ be the zero-points of $P(x)$, where

$$\begin{aligned} &|\zeta_1| \leq \cdots \leq |\zeta_m| \\ &< |\zeta_{m+1}| \leq \cdots \leq |\zeta_n|. \end{aligned} \tag{91}$$

If $e = \sqrt[m]{\varepsilon} \leq 1/9$ then $|\zeta_m|$ and $|\zeta_{m+1}|$ are bounded as

$$\begin{aligned} &|\zeta_m| \\ &< \frac{1+3e}{4}\left[1 - \sqrt{1 - \frac{16e}{(1+3e)^2}}\right], \\ &|\zeta_{m+1}| \\ &> \frac{1+3e}{4}\left[1 + \sqrt{1 - \frac{16e}{(1+3e)^2}}\right]. \end{aligned} \tag{92}$$

Furthermore, we can approximate the right-hand-side expressions of Eq. (92) as

$$\begin{aligned} &|\zeta_m| \\ &< 2e \cdot \left[\frac{1}{1+3e} + \frac{16e}{(1+3e)^3}\right], \\ &|\zeta_{m+1}| \\ &> \frac{1}{2} - \frac{e(1-9e)}{2(1+3e)} - \frac{32e^2}{(1+3e)^3}. \end{aligned} \tag{93}$$

□

Before the proof, we investigate the zero-points of $P(x)$ roughly. Put

$$\begin{aligned} P'(x) &= x^m + \varepsilon_{m-1}x^{m-1} + \cdots \\ &\quad \cdots + \varepsilon_1 x + \varepsilon_0, \\ P''(x) &= c_n x^{n-m} + \cdots \\ &\quad \cdots + c_{m+1}x + 1. \end{aligned} \tag{94}$$

Note that $P(x) \approx P''(x)P'(x)$. Using the following well-known theorem (see Mignotte[6] for example), we can bound the zero-points of $P'(x)$ and $P''(x)$ easily as in Corollaries 13 and 14 below.

Theorem 12 Let $A(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$, with $a_n a_0 \neq 0$, be a polynomial with complex coefficients and with zero-points $\zeta_1, \ldots, \zeta_n$. Then, we have the following bounds for the zero-points of $A(x)$:

$$\begin{aligned} &\max\{|\zeta_1|, \ldots, |\zeta_n|\} \\ &\leq \frac{|a_n| + \max\{|a_{n-1}|, \ldots, |a_0|\}}{|a_n|}, \\ &\min\{|\zeta_1|, \ldots, |\zeta_n|\} \\ &\geq \frac{|a_0|}{|a_0| + \max\{|a_1|, \ldots, |a_n|\}}. \end{aligned} \tag{95}$$

□

Applying Theorem 12 to $\varepsilon^{-1}P'(\sqrt[m]{\varepsilon}x)$ and $P''(x)$, respectively, we obtain the following corollaries:

Corollary 13 Let the zero-points of $P'(x)$ be $\zeta'_1, \ldots, \zeta'_m$; then we have $\max\{|\zeta'_1|, \ldots, |\zeta'_m|\} \leq 2\sqrt[m]{\varepsilon}$. □

Corollary 14 Let the zero-points of $P''(x)$ be $\zeta''_{m+1}, \ldots, \zeta''_n$; then we have $\min\{|\zeta''_{m+1}|, \ldots, |\zeta''_n|\} \geq 1/2$. □

These corollaries show that $P(x)$ has m zero-points of magnitude $\lesssim 2\sqrt[m]{\varepsilon}$ and that the other $(n-m)$ zero-points have absolute values $\gtrsim 1/2$. We now prove Theorem 11.

Proof of Theorem 11. We first consider the zero-point ζ of Eq. (89), such that $|\zeta| \lesssim 2\sqrt[m]{\varepsilon}$. Applying the transformation $\zeta = e\bar{\zeta}$ $(= \sqrt[m]{\varepsilon}\bar{\zeta})$ to $P(\zeta) = 0$, we obtain

$$\begin{aligned} &c_n e^{n-m}\bar{\zeta}^n + \cdots + c_{m+1}e\bar{\zeta}^{m+1} \\ &+ \bar{\zeta}^m + (\varepsilon_{m-1}/e)\bar{\zeta}^{m-1} + \cdots \\ &\quad + (\varepsilon_0/e^m)\bar{\zeta}^0 = 0. \end{aligned} \tag{96}$$

We are considering the zero-point $\bar{\zeta}$ such that $|\bar{\zeta}| \lesssim 2$, hence the zero-point is determined mostly by the terms of degree $\leq m$ and the terms $c_{m+j}e^j\bar{\zeta}^{m+j}$ $(j = 1, \ldots, n-m)$ contribute only as small correction terms because $e \ll 1$. (We can state this situation as follows. Consider a set of equations of degree m:

$$\left\{\begin{aligned} &a_m z^m + (c_{m-1}/e)z^{m-1} + \cdots \\ &\qquad \cdots + (c_0/e^m)z^0 = 0, \\ &a_m \in \{1 + c_{m+1}e\bar{z} + \cdots \\ &\qquad \cdots + c_n e^{n-m}\bar{z}^{n-m} \mid \\ &\qquad |\bar{z}| \leq \bar{\zeta}_{\max}\}, \end{aligned}\right. \tag{97}$$

where $\bar{\zeta}_{\max}$ is an upper bound of $|\zeta_m/e|$. Obviously, $\bar{\zeta} = \zeta_m/e$ is a solution of one equation in this set. For the solution of any equation in this set, we can derive an upper bound.) Thus, rewriting the above equation as

$$\begin{aligned} &(c_n e^{n-m}\bar{\zeta}^{n-m} + \cdots + c_{m+1}e\bar{\zeta} + 1)\bar{\zeta}^m \\ &+ (\varepsilon_{m-1}/e)\bar{\zeta}^{m-1} + \cdots \\ &\quad \cdots + (\varepsilon_0/e^m)\bar{\zeta}^0 = 0, \end{aligned} \tag{98}$$

we can regard Eq. (98) as an equation of degree m with the leading coefficient $a_m = 1 + c_{m+1}e\bar{\zeta} + \cdots + c_n e^{n-m}\bar{\zeta}^{n-m} \approx 1$. Therefore, from Theorem 12, we obtain

$$\begin{aligned} |\bar{\zeta}| &\leq 1 + \max\{|\varepsilon_{m-1}/e|, \ldots, |\varepsilon_0/e^m|\}/|a_m| \\ &\leq 1 + \frac{1}{1 - |e\bar{\zeta}| - \cdots - |e\bar{\zeta}|^{n-m}} \\ &< 1 + \frac{1}{1 - |e\bar{\zeta}|/(1 - |e\bar{\zeta}|)} \\ &= \frac{2 - 3|e\bar{\zeta}|}{1 - 2|e\bar{\zeta}|}, \end{aligned} \tag{99}$$

or

$$|\zeta| < \frac{2e - 3e|\zeta|}{1 - 2|\zeta|}. \tag{100}$$

Inequality (100) gives us

$$2|\zeta|^2 - (1+3e)|\zeta| + 2e > 0. \tag{101}$$

Let z_- and z_+ be the solutions of equation $2z^2 -$

$(1+3e)z+2e=0$ with $z_- \le z_+$. We see that $z_\pm$ are real if and only if $e \le 1/9$, and $z_- \simeq 2e$ and $z_+ \simeq (1-e)/2$ for $|e| \ll 1$. Therefore, we have

$$4|\zeta| < (1+3e) \times \left[1-\sqrt{1-\frac{16e}{(1+3e)^2}}\right] \quad (102)$$

for $e \le 1/9$. Using the inequality $\sqrt{1-x} > 1-x/2-x^2/2$, which is valid for $0<x<1$, and putting $x=16e/(1+3e)^2$, we obtain

$$4|\zeta| < (1+3e) \times \left[\frac{8e}{(1+3e)^2}+\frac{128e^2}{(1+3e)^4}\right], \quad (103)$$

or

$$|\zeta| < 2e\cdot\left[\frac{1}{1+3e}+\frac{16e}{(1+3e)^3}\right]. \quad (104)$$

This inequality is valid for $16e/(1+3e)^2 < 1$, or for $e < 1/9$.

Next, we consider the zero-point ζ of Eq. (89), such that $1/2 \lesssim |\zeta|$. Dividing $P(\zeta)=0$ by ζ^m, we obtain the equality

$$c_n\zeta^{n-m}+\cdots+c_{m+1}\zeta+1 + \varepsilon_{m-1}/\zeta+\varepsilon_{m-2}/\zeta^2+\cdots \cdots+\varepsilon_0/\zeta^m=0. \quad (105)$$

Since we are considering the zero-point ζ such that $1/2 \lesssim |\zeta|$, the terms $\varepsilon_{m-j}/\zeta^j$ $(j=1,\ldots,m)$ contribute only as small correction terms because $|\varepsilon_{m-j}| \ll 1$. Thus, following the same reasoning as for Eq. (98), we can regard Eq. (105) as an equation of degree $n-m$ with the constant term $a_0 = 1+\varepsilon_{m-1}/\zeta+\cdots+\varepsilon_0/\zeta^m \approx 1$. From Theorem 12, we obtain

$$\begin{aligned}|\zeta| &\ge \frac{1}{1+\max\{|c_n|,\ldots,|c_{m+1}|\}/|a_0|} \\ &\ge \frac{1}{1+1/(1-|e/\zeta|-\cdots-|e/\zeta|^m)} \\ &> \frac{1}{1+\frac{1}{1-|e/\zeta|/(1-|e/\zeta|)}} \quad (106)\\ &= \frac{1-2|e/\zeta|}{2-3|e/\zeta|}.\end{aligned}$$

Inequality Eq. (106) gives us

$$2|\zeta|-(1+3e)|\zeta|+2e>0. \quad (107)$$

Solving Eq. (107) with condition $e \le 1/9$, we obtain

$$4|\zeta| > (1+3e) \times \left[1+\sqrt{1-\frac{16e}{(1+3e)^2}}\right]. \quad (108)$$

Using the inequality $\sqrt{1-x} > 1-x/2-x^2/2$ again, we obtain

$$4|\zeta| > (1+3e) \times \left[2-\frac{8e}{(1+3e)^2}-\frac{128e^2}{(1+3e)^4}\right], \quad (109)$$

or

$$|\zeta| > \frac{1}{2}-\frac{e(1-9e)}{2(1+3e)}-\frac{32e^2}{(1+3e)^3} \quad (110)$$

for $e < 1/9$. □

(Received December 11, 1998)
(Accepted January 6, 2000)

Akira Terui was born in 1971. He received his M.S. degree from Univ. Tsukuba in 1997. Since 1999 he has been in Univ. Tsukuba as a research associate. His research interests include theory and application of approximate algebraic computation: solving system of algebraic equations, calculation of singularities of algebraic functions, etc., by means of computer algebra with approximate computation. He is a member of IPSJ, JSIAM, JSSAC and ACM.

Tateaki Sasaki was born in 1946. He received M.S. and D.S. degrees from Univ. Tokyo in 1970 and 1973, respectively. He had been a researcher of RIKEN: The Institute of Physical and Chemical Research (Information Science Laboratory) in 1974–1991 and a visiting researcher of Univ. Utah (Dept. Computer Science) in 1978–1979. Since 1991, he has been a professor of Univ. Tsukuba (Institute of Mathematics). His research interests include algorithm development of computer algebra and development of formula manipulation system. In particular, he is an initiator of approximate algebra. He is a member of IPSJ, JSIAM, JSSAC, MSJ and ACM.

Wave propagation in linear electrodynamics

Yuri N. Obukhov,* Tetsuo Fukui,† and Guillermo F. Rubilar‡
Institute for Theoretical Physics, University of Cologne, D-50923 Köln, Germany
(Received 11 February 2000; published 27 July 2000)

The Fresnel equation governing the propagation of electromagnetic waves for the most general linear constitutive law is derived. The wave normals are found to lie, in general, on a fourth order surface. When the constitutive coefficients satisfy the so-called reciprocity or closure relation, one can define a duality operator on the space of the two-forms. We prove that the closure relation is a sufficient condition for the reduction of the fourth order surface to the familiar second order light cone structure. We finally study whether this condition is also necessary.

PACS number(s): 04.20.Cv, 04.30.Nk

I. INTRODUCTION

The electromagnetic wave represents perhaps the most important classical device with the help of which one can carry out physical measurements and transmit information. The intrinsic properties and motion of material media, as well as the geometrical structure of spacetime, can affect the propagation of electromagnetic waves. In the most general setting [1,2], electromagnetic phenomena are described by the pair of two-forms H,F (called the electromagnetic excitation and the field strength, respectively) which satisfy the Maxwell equations $dH=J, dF=0$, *together* with the constitutive law $H=H(F)$. The latter relation contains crucial information about the underlying physical continuum (i.e., material medium and/or spacetime). Mathematically, this constitutive law arises either from a suitable phenomenological theory of a medium or from the electromagnetic field Lagrangian.

In general, the constitutive law establishes a nonlinear (or even nonlocal) relation between the electromagnetic excitation and the field strength. The function (or functional) $H(F)$ may depend on the polarization and magnetization properties of matter, and/or on the spacetime geometry, i.e., metric, curvature, torsion, and nonmetricity. Previously, the propagation of electromagnetic waves was analyzed for a variety of constitutive laws: for nonlinear models in Minkowski and Riemannian spacetimes [3], for electrodynamics in a Riemann-Cartan manifold [4], and also for certain nonminimal and higher derivative gravity models [5]. Numerous authors [6] discussed electromagnetic waves in Einstein-Maxwell theory. The main aim of this paper is to investigate wave propagation in Maxwell electrodynamics with the most general *linear* constitutive law. We derive the generalized Fresnel equation which determines the wave normals directly from the constitutive coefficients. This result is of interest, e.g., for various applications in crystaloptics and related domains.

Another motivation for the present work comes from the study of a deep relationship between the duality operators defined on two-forms and the conformal classes of spacetime metrics in four dimensions. Within classical Maxwell electrodynamics, Toupin, Schönberg, and others [8] have noticed that the constitutive coefficients define a duality operator, provided a certain reciprocity or closure condition is fulfilled, and gave first demonstrations of the existence of the corresponding conformal metric structure. Later these observations were rediscovered and developed in mathematics [9] and in gravity theory [10]. Recently the complete explicit solution of the closure relation has been given [11], and it was conjectured that the reciprocity condition is a necessary and sufficient condition for the standard null-cone structure for the light propagation (see also independent arguments in Ref. [12]). Here we give a partial answer to this question.

II. ELECTRODYNAMICS WITH LINEAR CONSTITUTIVE LAW

Let us consider the Maxwell equations in vacuum

$$dH=0, \quad dF=0, \tag{2.1}$$

i.e., we assume that the electric current three-form J vanishes in the spacetime region under consideration. Given the local coordinates x^i, $i=0,1,2,3$, we can decompose the exterior forms as

$$H=\frac{1}{2}H_{ij}dx^i\wedge dx^j, \quad F=\frac{1}{2}F_{ij}dx^i\wedge dx^j. \tag{2.2}$$

Following Refs. [11,13], we write the linear constitutive law in terms of the electromagnetic excitation and field strength tensors as

$$H_{ij}=\frac{1}{4}\epsilon_{ijkl}\chi^{klmn}F_{mn}, \quad i,j,\ldots=0,1,2,3. \tag{2.3}$$

*Also at Department of Theoretical Physics, Moscow State University, 117234 Moscow, Russia. Email address: yo@thp.uni-koeln.de
†On leave from Department of Human Informatics, Mukogawa Women's University, 663-8558 Nishinomiya, Japan. Email address: fukui@mwu.mukogawa-u.ac.jp
‡Email address: gr@thp.uni-koeln.de

0556-2821/2000/62(4)/044050(5)/$15.00

Here ϵ_{ijkl} is the Levi-Civita symbol and $\chi^{ijkl}(x)$ an even tensor density of weight +1 (called the constitutive tensor density) which can be decomposed according to

$$\chi^{ijkl}=f(x)\overset{\circ}{\chi}{}^{ijkl}+\alpha(x)\epsilon^{ijkl}, \quad \text{with } \overset{\circ}{\chi}{}^{[ijkl]}\equiv 0. \quad (2.4)$$

Here $f(x)$ is a dimensionful scalar function such that $\overset{\circ}{\chi}{}^{ijkl}$ is dimensionless. The pseudo-scalar constitutive function $\alpha(x)$ can be identified (on the *kinematic* level) as an Abelian axion field, whereas $f(x)$ can be interpreted as a dilaton scalar field. Note that $\overset{\circ}{\chi}{}^{ijkl}$ has the same algebraic symmetries and therefore the same number of 20 independent components as a Riemannian curvature tensor:

$$\overset{\circ}{\chi}{}^{ijkl}=-\overset{\circ}{\chi}{}^{jikl}=-\overset{\circ}{\chi}{}^{ijlk}=\overset{\circ}{\chi}{}^{klij}, \quad \overset{\circ}{\chi}{}^{[ijkl]}=0. \quad (2.5)$$

This follows from the existence and the structure of the Lagrangian for the linear electrodynamics $V_{\rm lin}=-\frac{1}{2}H\wedge F$, see Refs. [2,13]. It is convenient to adopt a more compact (essentially *bivector*) notation by defining the three-(co)vector quantities

$$\mathcal{D}^a:=\begin{pmatrix} H_{23}\\ H_{31}\\ H_{12}\end{pmatrix}, \quad \mathcal{H}_a:=\begin{pmatrix} H_{01}\\ H_{02}\\ H_{03}\end{pmatrix}$$

and

$$B^a:=\begin{pmatrix} F_{23}\\ F_{31}\\ F_{12}\end{pmatrix}, \quad E_a:=\begin{pmatrix} F_{10}\\ F_{20}\\ F_{30}\end{pmatrix}, \quad (2.6)$$

for the electric and magnetic excitations, and for the magnetic and electric field strengths, respectively. The Latin indices label now $a,b,c,\ldots=1,2,3$. The constitutive tensor is then naturally parametrized by a triplet of 3×3 matrices, $\overset{\circ}{\chi}{}^{ijkl}=\{\mathcal{A}^{ab},\mathcal{B}_{ab},\mathcal{C}^a{}_b\}$, so that the constitutive law (2.4) is finally recast into

$$\begin{pmatrix}\mathcal{H}_a\\ \mathcal{D}^a\end{pmatrix}=f(x)\begin{pmatrix}\mathcal{C}^b{}_a & \mathcal{B}_{ab}\\ \mathcal{A}^{ab} & \mathcal{C}^a{}_b\end{pmatrix}\begin{pmatrix}-E_b\\ B^b\end{pmatrix}+\alpha(x)\begin{pmatrix}-E_a\\ B^a\end{pmatrix}. \quad (2.7)$$

Here the 3×3 matrices satisfy $\mathcal{A}^{ab}=\mathcal{A}^{ba}$, $\mathcal{B}_{ab}=\mathcal{B}_{ba}$, and $\mathcal{C}_a{}^a=0$, thereby providing the algebraic properties (2.5).

III. WAVE PROPAGATION: FRESNEL EQUATION

In the theory of partial differential equations, the propagation of waves is described by Hadamard discontinuities of solutions across a characteristic (wave front) hypersurface S [7]. One can locally define S by the equation $\Phi(x^l)=\text{const}$. The Hadamard discontinuity of any function $\mathcal{F}(x)$ across the hypersurface S is determined as the difference $[\mathcal{F}](x):=\mathcal{F}(x_+)-\mathcal{F}(x_-)$, where $x_\pm:=\lim_{\varepsilon\to 0}(x\pm\varepsilon)$ are points on the opposite sides of $S\ni x$. An ordinary electromagnetic wave is a solution of the Maxwell equations (2.1) for which the derivatives of H and F have regular discontinuities across the wave front hypersurface S.

In terms of the (co)vector components, we have on the characteristic hypersurface S:

$$[\mathcal{D}^a]=0, \quad [\partial_i\mathcal{D}^a]=d^a q_i, \quad [\mathcal{H}_a]=0, \quad [\partial_i\mathcal{H}_a]=h_a q_i, \quad (3.1)$$

$$[B^a]=0, \quad [\partial_i B^a]=b^a q_i, \quad [E_a]=0, \quad [\partial_i E_a]=e_a q_i, \quad (3.2)$$

where d^a,h_a,b^a,e_a describe discontinuities of the corresponding quantities across S, and the wave covector normal to the front is given by

$$q_i:=\partial_i\Phi. \quad (3.3)$$

Equations (3.1),(3.2) represent the Hadamard geometrical compatibility conditions. Substituting Eq. (2.2) into Eq. (2.1), and using Eqs. (2.6) and (3.1),(3.2), we find

$$q_0 d^a-\epsilon^{abc}q_b h_c=0, \quad q_0 b^a+\epsilon^{abc}q_b e_c=0, \quad (3.4)$$

$$q_a d^a=0, \quad q_a b^a=0, \quad (3.5)$$

where ϵ^{abc} is the three-dimensional Levi-Civita symbol. In this system only the six equations (3.4) are independent. Assuming that $q_0\neq 0$, one finds that Eqs. (3.5) are trivially satisfied if one substitutes Eq. (3.4) into them. (Note that the characteristics with $q_0=0$ do not have intrinsic meaning for the evolution equations, since they obviously depend on the arbitrary choice of coordinates.)

Differentiating Eq. (2.7) and using the compatibility conditions (3.1),(3.2), we find additionally six algebraic equations

$$\begin{pmatrix}h_a\\ d^a\end{pmatrix}=f(x)\begin{pmatrix}\mathcal{C}_a{}^b & \mathcal{B}_{ab}\\ \mathcal{A}^{ab} & \mathcal{C}_b{}^a\end{pmatrix}\begin{pmatrix}-e_b\\ b^b\end{pmatrix}+\alpha(x)\begin{pmatrix}-e_a\\ b^a\end{pmatrix}. \quad (3.6)$$

Note that the constitutive coefficients and their first derivatives are assumed to be continuous across S.

We can now substitute d^a and h_a from Eq. (3.6) into the first equation (3.4), which gives

$$f(x)q_0(-\mathcal{A}^{ab}e_b+\mathcal{C}_b{}^a b^b)+\alpha(x)q_0 b^a =f(x)\epsilon^{abc}q_b(-\mathcal{C}_c{}^d e_d+\mathcal{B}_{cd}b^d)-\alpha(x)\epsilon^{abc}q_b e_c. \quad (3.7)$$

The terms proportional to the axion field $\alpha(x)$ drop out completely due to Eq. (3.4), and then one can also remove the common dilaton factor $f(x)$ on both sides of the equation. [We assume $f(x)\neq 0$, since otherwise there is no hyperbolic evolution system.] Finally we substitute b^a in terms of e_b from the second equation (3.4), and after some rearrangements one finds

$$(q_0^2\mathcal{A}^{ab}+q_0 q_d[\mathcal{C}^a{}_c\epsilon^{cdb}+\mathcal{C}^b{}_c\epsilon^{cda}]+q_c q_f\epsilon^{aec}\epsilon^{bfd}\mathcal{B}_{cd})e_b =0. \quad (3.8)$$

This homogeneous algebraic equation has a nontrivial solution when

$$W:=\det\left|q_0^2\mathcal{A}^{ab}+q_0q_d[\mathcal{C}^a{}_c\epsilon^{cdb}+\mathcal{C}^b{}_c\epsilon^{cda}]+q_eq_f\epsilon^{aec}\epsilon^{bfd}\mathcal{B}_{cd}\right|=0. \tag{3.9}$$

This is a Fresnel equation which is central in the wave propagation analysis. It determines the geometry of the wave normals in terms of the constitutive coefficients $\mathcal{A},\mathcal{B},\mathcal{C}$. A direct calculation yields the general result

$$\begin{aligned}W&=q_0^2(q_0^4M+q_0^3q_aM^a+q_0^2q_aq_bM^{ab}\\&\quad+q_0q_aq_bq_cM^{abc}+q_aq_bq_cq_dM^{abcd})\\&=0,\end{aligned} \tag{3.10}$$

where we have denoted

$$M:=\det\mathcal{A},\quad M^a:=2\,\epsilon_{bcd}\mathcal{A}^{ab}\mathcal{C}^c{}_e\mathcal{A}^{ed}, \tag{3.11}$$

$$M^{ab}:=\mathcal{B}_{cd}(\mathcal{A}^{ab}\mathcal{A}^{cd}-\mathcal{A}^{ac}\mathcal{A}^{bd})-\mathcal{A}^{cd}\mathcal{C}^a{}_c\mathcal{C}^b{}_d+4\mathcal{A}^{ac}\mathcal{C}^b{}_d\mathcal{C}^d{}_c-2\mathcal{A}^{ab}\mathcal{C}^c{}_d\mathcal{C}^d{}_c, \tag{3.12}$$

$$M^{abc}:=2\,\epsilon^{cde}[\mathcal{B}_{df}(\mathcal{A}^{ab}\mathcal{C}^f{}_e-\mathcal{A}^{af}\mathcal{C}^b{}_e)+\mathcal{C}^a{}_e\mathcal{C}^b{}_f\mathcal{C}^f{}_d] \tag{3.13}$$

$$M^{abcd}:=\epsilon^{cef}\epsilon^{dgh}\mathcal{B}_{fh}[\tfrac{1}{2}\mathcal{A}^{ab}\mathcal{B}_{eg}-\mathcal{C}^a{}_e\mathcal{C}^b{}_g]. \tag{3.14}$$

Note that only the completely symmetric parts $M^{(a_1\cdots a_p)}$, $p=2,3,4$, contribute to the Fresnel equation. Since $q_0\neq0$, one can delete the first factor in Eq. (3.10), and thus we finally find that the wave covector q_i lies, in general, on a fourth order surface. This is different from the light *cone* (i.e., second order) structure which arises only in a particular case. In the next section we demonstrate that the latter corresponds to the closure condition. Earlier, the relation between the fourth- and the second-order wave geometry was studied by Tamm [16] for a special case of the linear constitutive law.

IV. THE CLOSURE RELATION AS A SUFFICIENT CONDITION

The linear constitutive law defines a duality operator when the constitutive coefficients satisfy the "reciprocity" or "closure" relation [8,11]:

$$\frac{1}{4}\epsilon_{ijmn}\epsilon_{pqrs}\overset{\circ}{\chi}{}^{mnpq}\overset{\circ}{\chi}{}^{rskl}=-\delta_{ij}^{kl}, \tag{4.1}$$

or in terms of the 3×3 matrices

$$\mathcal{A}^{ac}\mathcal{B}_{cb}+\mathcal{C}^a{}_c\mathcal{C}^c{}_b=-\delta^a_b,\quad \mathcal{C}^{(a}{}_c\mathcal{A}^{b)c}=0,\quad \mathcal{C}^c{}_{(a}\mathcal{B}_{b)c}=0. \tag{4.2}$$

The general solution of the closure condition (4.1),(4.2) reads [11]

$$\mathcal{A}^{ab}=\frac{1}{\det\mathcal{B}}(k^2\mathcal{B}^{ab}-k^ak^b)-\mathcal{B}^{ab}, \tag{4.3}$$

$$\mathcal{C}^a{}_b=\mathcal{B}^{ad}\epsilon_{dbc}k^c=\frac{1}{\det\mathcal{B}}\epsilon^{adc}\mathcal{B}_{db}k_c. \tag{4.4}$$

Here k^a is an arbitrary three-vector, $k_b:=\mathcal{B}_{ab}k^a$, $k^2:=\mathcal{B}_{ab}k^ak^b$, and $\mathcal{B}^{ab}$ denotes the inverse matrix to $\mathcal{B}_{ab}$.

Starting from Eqs. (4.3),(4.4), the direct calculation yields

$$M=-\frac{1}{\det\mathcal{B}}\left(1-\frac{k^2}{\det\mathcal{B}}\right)^2, \tag{4.5}$$

$$M^a=\frac{1}{\det\mathcal{B}}4k^a\left(1-\frac{k^2}{\det\mathcal{B}}\right), \tag{4.6}$$

$$M^{ab}=-\frac{1}{\det\mathcal{B}}4k^ak^b+2\mathcal{B}^{ab}\left(1-\frac{k^2}{\det\mathcal{B}}\right), \tag{4.7}$$

$$M^{abc}=-4\mathcal{B}^{b(a}k^{c)}, \tag{4.8}$$

$$M^{(abcd)}=-(\det\mathcal{B})\mathcal{B}^{(ab}\mathcal{B}^{cd)}. \tag{4.9}$$

Substituting all this into the general Fresnel equation (3.10), we find

$$\begin{aligned}W&=-\sigma q_0^2\left[\frac{q_0^2}{\sqrt{|\det\mathcal{B}|}}\left(1-\frac{k^2}{\det\mathcal{B}}\right)-\frac{2q_0(q_ak^a)}{\sqrt{|\det\mathcal{B}|}}\right.\\&\quad\left.-\sqrt{|\det\mathcal{B}|}(q_aq_b\mathcal{B}^{ab})\right]^2\\&=-\sigma q_0^2(q_iq_jg^{ij})^2.\end{aligned} \tag{4.10}$$

Here $\sigma=\mathrm{sgn}(\det\mathcal{B})$, and g^{ij} is the (inverse) four-dimensional metric which arises from the duality operator and the closure relation [11,13]

$$g^{00}=\frac{1}{\sqrt{|\det\mathcal{B}|}}\left(1-\frac{k^2}{\det\mathcal{B}}\right), \tag{4.11}$$

$$g^{0a}=-\frac{k^a}{\sqrt{|\det\mathcal{B}|}}, \tag{4.12}$$

$$g^{ab}=-\sqrt{|\det\mathcal{B}|}\mathcal{B}^{ab}. \tag{4.13}$$

This metric g_{ij} (defined up to a conformal factor) always has the Lorentzian signature, although it is not necessarily interpretable as a spacetime metric (this is a so called *optical metric*, in general; see, e.g., Ref. [14]). As shown in Ref. [13], the constitutive tensor density (2.4) can be rewritten in terms of this metric as

$$\chi^{ijkl}=f(x)\sqrt{-g}(g^{ik}g^{jl}-g^{jk}g^{il})+\alpha(x)\epsilon^{ijkl}. \tag{4.14}$$

Thus we indeed recover the null cone $q_i q^i = q_i q_j g^{ij} = 0$ structure for the propagation of electromagnetic waves from our general analysis: provided the constitutive matrices satisfy the closure relation (4.1),(4.2), the quartic surface (3.10) degenerates to the null cone for the induced metric g_{ij}.

It is worthwhile to note that the Fresnel equation (3.10) can be rewritten in an explicitly covariant form

$$G^{ijkl} q_i q_j q_k q_l = 0, \quad i,j,\ldots = 0,1,2,3, \tag{4.15}$$

where the fourth order totally symmetric tensor density G^{ijkl} is constructed as the cubic polynomial of the components of the constitutive tensor

$$G^{ijkl} = \frac{1}{4!} \overset{\circ}{\chi}{}^{mnp(i} \overset{\circ}{\chi}{}^{j|qr|k} \overset{\circ}{\chi}{}^{l)stu} \epsilon_{mnrs} \epsilon_{pqtu}. \tag{4.16}$$

(Here the total symmetrization is extended only over the four indices i,j,k,l with all the summation indices excluded.) Tamm [16] has introduced a similar "fourth-order metric" for the particular case of the linear constitutive law.

V. THE CLOSURE RELATION AS A NECESSARY CONDITION

It was conjectured [11,13] that the closure relation is not only sufficient, but also a necessary condition for the reduction of the quartic geometry (3.10) to the null cone. The complete proof of this conjecture requires a rather lengthy algebra and will be considered elsewhere. Here we demonstrate the validity of the necessary condition in a particular case when the matrix $C = 0$.

Putting $C^a{}_b = 0$, we find from Eqs. (3.11)–(3.14) that $M^a = 0$ and $M^{abc} = 0$, whereas

$$M^{ab} = \mathcal{B}_{cd} (\mathcal{A}^{ab} \mathcal{A}^{cd} - \mathcal{A}^{ac} \mathcal{A}^{bd}), \tag{5.1}$$

$$M^{(abcd)} = (\det \mathcal{B}) \mathcal{A}^{(ab} \mathcal{B}^{cd)}. \tag{5.2}$$

Consequently, Eq. (3.10) reduces to

$$\mathcal{W} = q_0^2 (\det \mathcal{A} q_0^4 + q_0^2 \gamma + \det \mathcal{B} \alpha \beta), \tag{5.3}$$

where $\alpha = A^{ab} q_a q_b$, $\beta = B^{ab} q_a q_b$, and $\gamma = M^{ab} q_a q_b$. Assuming that the last equation describes a null cone, one concludes that the roots for q_0^2 should coincide and thus necessarily

$$\gamma^2 = 4 \det \mathcal{A} \det \mathcal{B} \alpha \beta. \tag{5.4}$$

Let us write $(\det \mathcal{A} \det \mathcal{B}) = s |\det \mathcal{A} \det \mathcal{B}|$, with $s = \mathrm{sgn}(\det \mathcal{A} \det \mathcal{B})$. Then Eq. (5.4) yields

$$2 \sqrt{|\det \mathcal{A} \det \mathcal{B}|}\, \frac{\alpha}{\gamma} = s\lambda, \quad 2 \sqrt{|\det \mathcal{A} \det \mathcal{B}|}\, \frac{\beta}{\gamma} = \frac{1}{\lambda}, \tag{5.5}$$

where λ is an arbitrary scalar factor. Recalling the definitions of α, β, γ, we then find

$$\mathcal{A}^{ab} = s\lambda^2 \mathcal{B}^{ab}. \tag{5.6}$$

Consequently, $M = \det \mathcal{A} = s\lambda^6 / \det \mathcal{B}$ and $M^{ab} = 2\lambda^4 \mathcal{B}^{ab}$, and therefore one verifies that

$$\mathcal{W} = \frac{s\lambda^2 q_0^2}{\det \mathcal{B}} (\lambda^2 q_0^2 + s q_a q_b \mathcal{B}^{ab} \det \mathcal{B})^2. \tag{5.7}$$

We immediately see that for $s = -1$ the quadratic form in Eq. (5.7) can have either the $(+---)$ signature or $(+++-)$. Similarly, for $s = 1$ the signature is either $(++++)$ or $(++--)$. Therefore, the Fresnel equation describes a correct light *cone* (hyperbolic) structure only in the case $s = -1$. Finally, one can verify that the above solutions satisfy

$$\frac{1}{4} \epsilon_{ijmn} \epsilon_{pqrs} \overset{\circ}{\chi}{}^{mnpq} \overset{\circ}{\chi}{}^{rskl} = s\lambda^2 \delta^{kl}_{ij}, \tag{5.8}$$

which for $s = -1$ reproduces the closure relation (4.1) after a trivial rescaling of the constitutive tensor density (and subsequently absorbing the factor λ into the "dilaton" field f).

VI. CONCLUSIONS

In this paper we have derived, extending the earlier results (see, e.g., Refs. [6,14,16]), the Fresnel equation governing the propagation of electromagnetic waves for the most general linear constitutive law. The wave covector lies, in general, on a *fourth order surface*. Such generic fourth order structure is not affected by the axionlike and dilatonlike parts of the constitutive tensor. Note, however, that the linear constitutive law $H = \alpha(x) F$ does not lead to hyperbolic evolution equations, and hence necessarily $f(x) \neq 0$.

We have proved that the closure relation (4.1) is a sufficient condition for the reduction of the fourth order surface to the familiar second order light cone structure. The corresponding family of conformally related metrics g coincides with that derived in Ref. [11], see also Ref. [13]. This result may be considered as an alternative (as compared to Urbantke's scheme [9,10]) derivation of the Lorentzian metric g from a duality operator. In terms of the Lagrangian, the closure relation is equivalent to the statement that $V_{\rm lin} = -\frac{1}{2}[f(x) F \wedge {}^*F + \alpha(x) F \wedge F]$, where the Hodge operator is defined by the metric g.

For the special case $C^a{}_b = 0$ we have proved that the requirement of reduction of the fourth order Fresnel structure to a second order one implies a relation between the constitutive coefficients which is slightly weaker than the closure relation (4.1), in that it allows for an arbitrary scalar factor. The latter though can be removed by the redefinition of the dilaton field $f(x)$. Also the signature of the resulting quadratic form is not fixed, so that one has to impose hyperbolicity as a separate condition.

It is worthwhile to note that the results obtained can be directly applied to the refinement and generalization of the previous analyses of the observational tests of the equivalence principle. See, for instance, Ref. [15] where some particular cases of the Fresnel equation have been studied in this context.

ACKNOWLEDGMENTS

We are grateful to Friedrich W. Hehl for useful comments and discussion of the results obtained. T.F. thanks the Institute for Theoretical Physics, University of Cologne, for the warm hospitality. G.F.R. would like to thank the German Academic Exchange Service (DAAD) for financial support (Kennziffer A/98/00829).

[1] E. J. Post, *Formal Structure of Electromagnetics—General Covariance and Electromagnetics* (North Holland, Amsterdam, 1962); C. Wang, *Mathematical Principles of Mechanics and Electromagnetism, Part B: Electromagnetism and Gravitation* (Plenum Press, New York, 1979); J. Stachel, Acta Phys. Pol. **35**, 689 (1969).

[2] F. W. Hehl and Yu. N. Obukhov, "*How does the electromagnetic field couple to gravity, in particular to metric, nonmetricity, torsion, and curvature?*," Report No. IASSNS-HEP-99/116, *Institute of Advanced Study, Princeton University, 1999*, in *Testing Relativistic Gravity in Space: Gyroscopes, Clocks, Interferometers . . . , Proceedings of the 220th Heraeus-Seminar*, 1999, Bad Honnef, edited by C. Lämmerzahl *et al.* (Springer, Berlin) (in press) gr-qc/0001010.

[3] S. A. Gutiérrez, A. L. Dudley, and J. F. Plebanski, J. Math. Phys. **22**, 2835 (1981); H. S. Ibarguen, A. Garcia, and J. Plebanski, *ibid.* **30**, 2689 (1989); M. Novello, V. A. De Lorenci, J. M. Salim, and R. Klippert, Phys. Rev. D **61**, 045001 (2000).

[4] R. de Ritis, M. Lavorgna, and C. Stornaiolo, Phys. Lett. **98A**, 411 (1983); L. L. Smalley, *ibid.* **117A**, 267 (1986).

[5] I. T. Drummond and S. J. Hathrell, Phys. Rev. D **22**, 343 (1980); S. Mohanty and A. R. Prasanna, Nucl. Phys. **B526**, 501 (1998).

[6] G. V. Skrotskii, Sov. Phys. Dokl. **2**, 226 (1957); A. M. Volkov, A. A. Izmest'ev, and G. V. Skrotskii, Zh. Éksp. Teor. Fiz. **59**, 1254 (1970) [Sov. Phys. JETP **32**, 686 (1971)]; J. Plebański, Phys. Rev. **118**, 1396 (1960); J. Ehlers, Z. Naturforsch. A **22**, 1328 (1967); B. Mashhoon, Phys. Rev. D **11**, 2679 (1975); J. Manzano and R. Montemayor, *ibid.* **56**, 6378 (1997).

[7] J. Hadamard, *Leçons sur la Propagation des Ondes et les Équations de l'Hydrodynamique* (Hermann, Paris, 1903); A. Lichnerowicz, in *Astrofisica e Cosmologia Gravitazione Quanti e Relatività, Centenario di Einstein* (Giunti Barbera, Firenze, 1979).

[8] R. A. Toupin, in *Non-Linear Continuum Theories, C.I.M.E. Conference*, Bressanone, Italy, 1965 (unpublished), pp. 206-342; M. Schönberg, Riv. Bras. Fis. **1**, 91 (1971); A. Peres, Ann. Phys. (N.Y.) **19**, 279 (1962); A. Z. Jadczyk, Bull. Acad. Pol. Sci., Ser. Sci., Phys. Astron. **27**, 91 (1979); C. Piron and D. J. Moore, Turk. J. Phys. **19**, 202 (1995).

[9] H. Urbantke, Acta Phys. Austriaca, Suppl. **XIX**, 875 (1978); G. Harnett, J. Math. Phys. **32**, 84 (1991).

[10] C. H. Brans, J. Math. Phys. **12**, 1616 (1971); R. Capovilla, T. Jacobson, and J. Dell, Phys. Rev. Lett. **63**, 2325 (1989); G. 't Hooft, Nucl. Phys. **B357**, 211 (1991).

[11] Yu. N. Obukhov and F. W. Hehl, Phys. Lett. B **458**, 466 (1999).

[12] C. Lämmerzahl *et al.*, report, University of Konstanz, August, 1999.

[13] F. W. Hehl, Yu. N. Obukhov, and G. F. Rubilar, in *Proceedings of the International European Conference on Gravitation "Journées Relativistes 99,"* Weimar, Germany, 1999 [Ann. Phys. (Leipzig) (in press)], gr-qc/9911096.

[14] H. F. Kremer, J. Math. Phys. **8**, 1197 (1967).

[15] M. P. Haugan and T. F. Kauffmann, Phys. Rev. D **52**, 3168 (1995); V. I. Denisov and M. I. Denisov, *ibid.* **60**, 047301 (1999).

[16] I. E. Tamm, J. Russ. Phys. Chem. Soc. **57**, 209 (1925); reprinted in I. E. Tamm, *Collected Papers* (Nauka, Moscow, 1975), Vol. I, pp. 33–61 (in Russian).

有理関数近似の離散化における問題点

愛媛大学 理工学研究科　村上 裕美 (Yumi MURAKAMI) *
愛媛大学 工学部　甲斐 博 (Hiroshi KAI)†
愛媛大学 工学部　野田 松太郎 (Matu-Tarow NODA) ‡

1 はじめに

有理関数補間を用いて関数近似を行った場合、補間区間内に不必要な極が現れる場合があるという問題がある [3]。この不必要な極は、補間を行う有理関数の分子分母の多項式が、補間区間内に非常に近い値の零点を持つために現れる。これまでの研究では、このことを利用して分子分母の多項式の近似 GCD を求めて、この零点を近似的な共通因子として取り除く方法 [1, 2] が提案されている。本研究では、有理関数補間を行うときに現れる不必要な極を生じる原因の解明を目的として、素朴な有理関数近似の計算に関する再検討を行うとともに、安定化理論 [4] を用いた有理関数補間についての検討を行った。

2 素朴な有理関数近似

関数 $f(x) \in C[a,b]$ に対する有理関数補間は次のように計算される。有限個の離散点 $a = x_0 < x_1 < \cdots < x_{m+n} = b$ を与え、対応する関数値 $f(x_k) = f_k$, $k = 0, 1, \cdots, m+n$ を求める。ここで与えられた m, n に対して、

$$r(x_k) = \frac{p(x_k)}{q(x_k)} = f_k \quad k = 0, 1, \cdots, m+n$$

を満たすような

$$r_{m,n}(x) = \frac{p_m(x)}{q_n(x)} = \frac{\sum_{i=0}^{m} a_i x^i}{\sum_{j=0}^{n} b_j x^j}$$

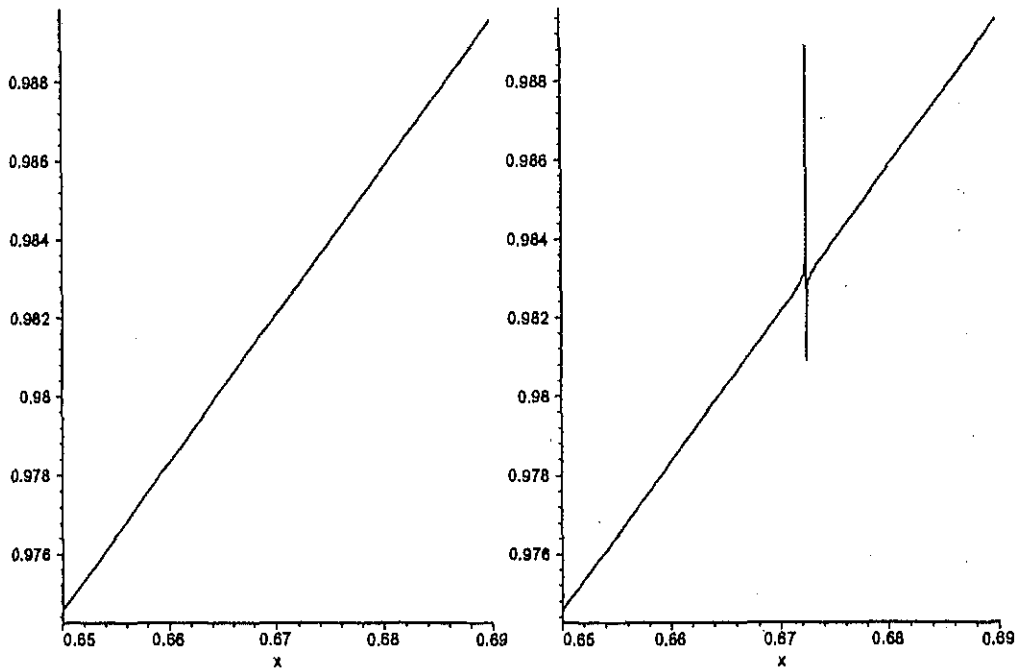
を求める。この有理関数を (m,n) 有理関数と呼び、便宜上 $b_0 = 1$ と規格化する。多項式の係数 a_i, b_j は一般に浮動小数であり、以下のような連立一次方程式を解くことによって求められる。

*cumi@hpc.cs.ehime-u.ac.jp
†kai@cs.ehime-u.ac.jp
‡noda@cs.ehime-u.ac.jp

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^m & -f_0x_0 & \cdots & -f_0x_0^n \\ 1 & x_1 & \cdots & x_1^m & -f_1x_1 & \cdots & -f_1x_1^n \\ 1 & x_2 & \cdots & x_2^m & -f_2x_2 & \cdots & -f_2x_2^n \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^m & -f_mx_m & \cdots & -f_mx_m^n \\ 1 & x_{m+1} & \cdots & x_{m+1}^m & -f_{m+1}x_{m+1} & \cdots & -f_{m+1}x_{m+1}^n \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_{m+n} & \cdots & x_{m+n}^m & -f_{m+n}x_{m+n} & \cdots & -f_{m+n}x_{m+n}^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \\ f_{m+1} \\ \vdots \\ f_{m+n} \end{pmatrix}$$

2.1 素朴な有理関数近似の問題点

前節で述べたような有理関数を用いて関数近似を行うと、元の関数 $f(x)$ が連続であるのに対して得られた有理関数が不必要な極を持ち、不連続になってしまう場合がある。これは、不必要な極に対応する有理関数の分子 $p_m(x)$ の零点が分母 $q_n(x)$ の零点に非常に近い値をとっていることが原因となっている [2]。例えば、関数 $\log(x+2)$ を補間区間を $[-1, 1]$ の間で分子分母の次数が 4 次 (m=n=4) の有理関数で近似することを考える。有効桁数 7 桁で連立一次方程式を解くと、次のような有理関数が得られる。



(a) 元の関数 $\log(x+2)$　　(b) 有理関数 $r_{4,4}(x)$

図 1: 有効桁数 4 桁で得られた係数による有理関数近似

$$r_{4,4}(x) \simeq \frac{0.6931471 + 0.01180017x - 1.168640x^2 - 0.5353216x^3 - 0.04802865x^4}{1 - 0.7043236x - 0.9975914x^2 - 0.2398265x^3 - 0.01131871x^4}$$

$$= \frac{-0.04802865(x + 8.201649)(x + 2.616697)(x + 0.9999999)(x - 0.6724661)}{-0.01131871(x + 15.91957)(x + 3.727162)(x + 2.214231)(x - 0.6724660)}$$

この有理関数では、補間区間内に存在する分母の零点 0.6724660 と非常に近いところに分子の零点 0.6724661 が存在している。このような有理関数を用いて関数近似を行うと、この近似的に近い値の零点の付近に不必要な極が現れる (図 1)。

3 ハイブリッド有理関数近似

有理関数近似を行った際に現れる不必要な極を除去し、高精度な近似を得る方法の一つにハイブリッド有理関数近似がある [1, 2, 3]。ハイブリッド有理関数近似は、有理関数の分子分母に存在する近似的に近い値の零点を、近似 GCD を用いて共通因子として取り除くことで、分子分母が近似的な共通因子を持たないような有理関数を構成する方法である。ハイブリッド有理関数近似のアルゴリズムを図 2 に示す。図 2 のアルゴリズムの中で用いられる近似 GCD を求めるには図 3 や図 4 のアルゴリズムを利用する。

[入力] : 有理関数 $r_{m,n}(x) = \frac{p_m(x)}{q_n(x)}$

[出力] : 共通因子を除去した有理関数 $\tilde{r}(x) = \frac{\tilde{p}(x)}{\tilde{q}(x)}$

[アルゴリズム] :

1. $\mathrm{A}ppGCD(p_m(x), q_n(x)) = g(x)$
2. $\tilde{r}(x) = \frac{p_m(x)/g(x)}{q_n(x)/g(x)}$

図 2: ハイブリッド有理関数近似のアルゴリズム

図 3 のアルゴリズムは、Euclid の互除法を浮動小数点係数に対応できるように拡張したアルゴリズムとなっている。図 4 のアルゴリズムは、入力の多項式の根が既に分かっているものとしてその根の値を比較し、ある一定の精度 δ によって近いと判断されたものについてはその根の値 (y_{ik}, z_{jk}) の中点を取り、この値を近似 GCD $\tilde{d}_\delta(x)$ の根となるようにして近似 GCD を構成するようなアルゴリズムになっている。ハイブリッド有理関数近似では、分子分母の多項式の近似的な共通因子だけを取り除くことが必要であるため、入力の多項式の根の値を直接比較して近似 GCD を求める図 4 のアルゴリズムの方が適している。

[入力] : 多項式 $P_1(x), P_2(x)$
[出力] : 近似 GCD $\mathrm{A}ppGCD(P_1(x), P_2(x)) = g(x)$

[アルゴリズム] :
1. $F \leftarrow P_1(x)$, $G \leftarrow P_2(x)$
2. $F = QG + \max(1, \|Q\|)R$ を満たす Q, R を求める。
   ($\|Q\|$: 多項式 Q の係数の絶対値の最大値)
3. if all coefficients of $R \leq \varepsilon$
   then $\mathrm{A}ppGCD(P_1(x), P_2(x)) = R$
   else $F \leftarrow G$, $G \leftarrow R$ go to step2.

図 3: 近似 GCD のアルゴリズム

[入力] : 多項式 $P_1(x) = u \times \prod_{i=1}^{m}(x - y_i), P_2(x) = v \times \prod_{j=1}^{n}(x - z_j)$, 精度 δ
[出力] : 近似 GCD $\delta - gcd$: $\tilde{d}_\delta(x)$

[アルゴリズム] :
$\tilde{d}_\delta(x) = \prod_{k=1}^{r}(x - x_k)$, $x_k = \frac{y_{ik}+z_{jk}}{2}$ $(k = 1, 2, \cdots, r)$
y_{ik}, z_{jk} はそれぞれ多項式 $P_1(x), P_2(x)$ の根で、$|y_{ik} - z_{jk}| \leq 2\delta$ を満たすもの。

図 4: Pan の近似 GCD のアルゴリズム

3.1 ハイブリッド有理関数近似の例

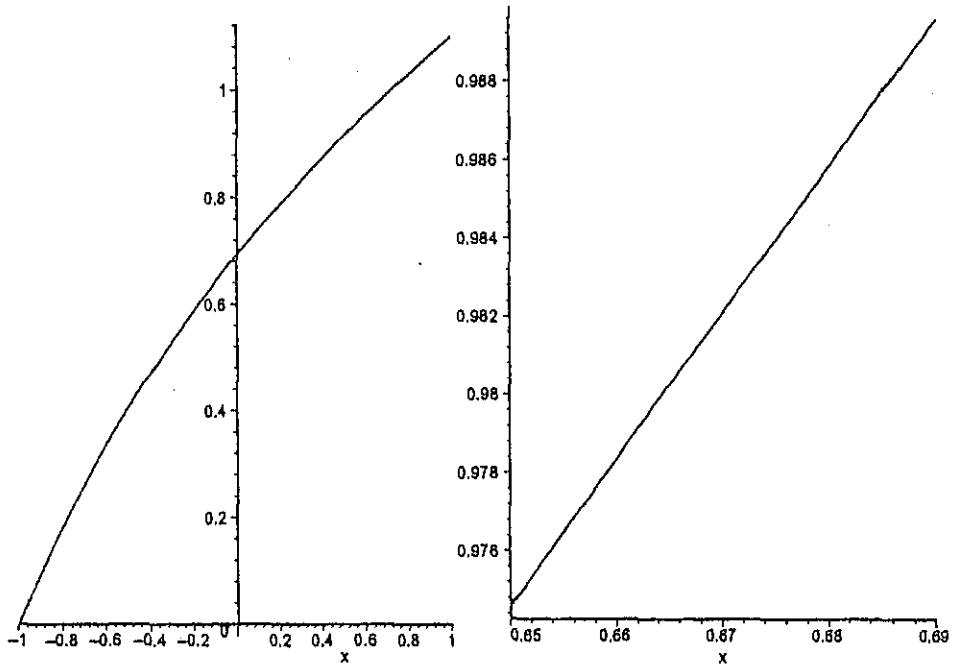
2.1 節の例で使用した有理関数 $r_{4,4}(x)$ を用いてハイブリッド有理関数近似を行った例を示す。図 3 のアルゴリズムで近似 GCD の値を求めると、

$$\mathrm{A}ppGCD(p_4(x), q_4(x)) = g(x) \simeq -0.008251166x + 0.005548411$$

となる。この近似 GCD の値で元の関数を割ると、以下のような $\tilde{r}(x)$ が得られる。

$$\begin{aligned}
\tilde{r}(x) &= \frac{p_4(x)/g(x)}{q_4(x)/g(x)} \\
&\simeq \frac{5.820832x^3 + 68.79246x^2 + 187.8921x + 124.9160}{1.371771x^3 + 29.98820x^2 + 141.0683x + 180.2204} \\
&= \frac{5.820832(x + 8.201647)(x + 2.616740)(x + 0.9999340)}{1.371771(x + 15.91957)(x + 3.727228)(x + 2.214140)}
\end{aligned}$$

この $\tilde{r}$ を用いて関数近似を行うと、不必要な極のない高精度な近似を得ることができる (図 5)。



(a) (b)

図 5: ハイブリッド有理関数近似

図 5 の (a) は、補間区間 $[-1,1]$ 全体を近似したもので、(b) は図 1 と同じ範囲を拡大したものである。この図から、元の関数に対して不必要な極のない高精度な近似が得られることが確認できる。

4 有理関数近似の例

4.1 不必要な極のふるまい

ここでは、有効桁数は補間多項式の次数を変化させたときの不必要な極のふるまいについて再検討を行う。$\log(x+2)$ を補間区間 $[-1,1]$ で有理関数近似を行った場合の不必要な極の位置を表 1 に示す。表で $r(5,5)$ と書かれているものは、分子分母の次数が 5 次の有理関数、$r(10,10)$ ならば分子分母が 10 次の有理関数で近似することを意味する。この結果から、極の位置に関しては規則性がないが、有効桁数が少ない場合や補間を行う多項式の次数が大きい場合に不必要な極が出やすくなっていることがわかる。表 1 では補間点を等間隔にとった場合の不必要な極の位置を示しているが、補間点の取り方を変えた場合でも、不必要な極は位置が異なるものの、同じような特徴が見られる。

表 1: 不必要な極の位置 ($\log(x+2)$, 補間区間 $[-1,1]$)

有効桁	$r(5,5)$	$r(10,10)$	$r(15,15)$	$r(20,20)$	$r(25,25)$
10 桁	-0.080667	-0.66028 -0.87680	-0.54814	-0.17765 0.59266	0.74821
20 桁	–	0.35248	-0.37456 0.54999 0.72561	-0.098106 0.92383	0.24963
30 桁	–	–	-0.56625	–	0.29373
40 桁	–	–	–	0.81169	–
50 桁	–	–	–	-0.78729	0.96187
60 桁	–	–	–	–	-0.75259

4.2 行列の条件数

前節の結果から、同じ次数の有理関数で関数近似を行った場合でも、有効桁数によって不必要な極の位置が変化していることがわかる。また、有効桁数を上げていくと不必要な極は現れなくなる。そこで、Gauss 消去法を適用する係数行列の条件数を求め、行列の性質と不必要な極の関係についての検証を行った。表 1 で使用したものと同じ関数 $\log(x+2)$ で、同じ補間区間 $[-1,1]$ とした場合の行列の条件数を、図 6 に示す。条件数は $\mathrm{cond}\, A = \|A\|\,\|A^{-1}\|$ によって求めている。

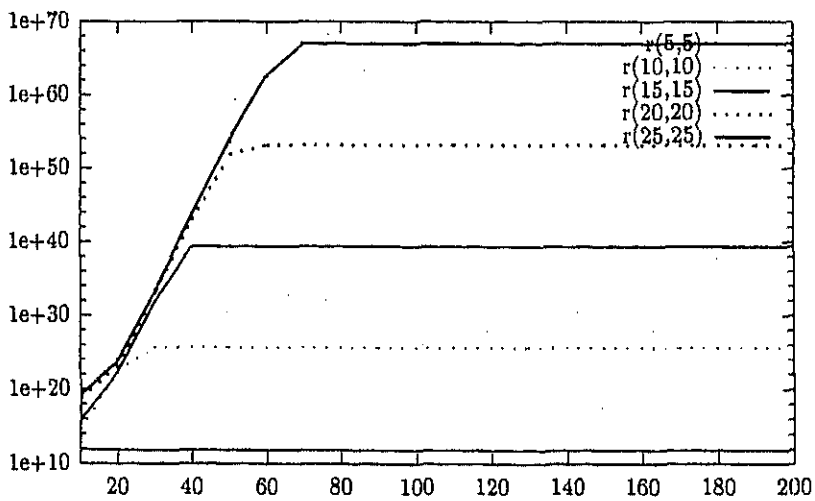


図 6: $\log(x+2)$ を有理関数近似する場合の係数行列の条件数

図 6 からわかるように、有理関数補間に使用される行列は非常に悪条件となってい

る。また、行列の条件数は有効桁数の少ない場合には有効桁ごとに条件数の値が全く異なっているが、有効桁数を多くするにしたがって一定の値に収束してくることがわかる。条件数の変化からも明らかだが、分子分母に5次のような低次の多項式を持つような有理関数補間においても、有効桁数が少ない場合には各係数の値は不安定となり、当然得られる有理関数も不安定になる。一方、このような場合にも有効桁数を20桁以上にすると、各係数の値も得られる有理関数の値も安定してくることがわかる。このような安定な有理関数補間が得られるのは、図6から$r(10,10)$有理関数では有効桁数40桁、$r(15,15)$有理関数では60桁等となることがわかる。ここで、条件数は、残差が何倍拡大されて相対誤差に反映するかの倍率となっていることに注目する。分子分母が5次の有理関数では、条件数が10^{11}程度で安定していることから、10進数にして10桁で計算を行うと、得られた解の精度はほとんど1桁もないといえる。図1で、有効桁10桁では不必要な極が生じ、20桁以上では不必要な極が現れないのは、計算に使用する有効桁数が条件数10^{11}に対して十分な精度を保つことができる大きさになっているためであると考えられる。

ここで、少ない有効桁数では得られる解の値が不安定であっても、結果として得られた有理関数で近似を行うと図1に示されるように不必要な極を生じるが、それ以外の部分は正しく関数近似を行うことができる。したがって、ハイブリッド有理関数近似を利用して不必要な極を除去すれば、高精度な近似が得られる。ハイブリッド有理関数近似により不必要な極を除去した場合の元の関数との誤差を表2に示す。誤差は、補間区間[-1,1]を100分割した点x_k, $(k=1,2,\cdots,100)$を用いて、次のような方法で求めた。

$$E = \max(|\log(x_k+2) - \tilde{r}(x_k)|), (k=,1,\cdots,100)$$

表2: 元の関数との誤差 ($\log(x+2)$, 補間区間$[-1,1]$)

有効桁	$r(5,5)$	$r(10,10)$	$r(15,15)$	$r(20,20)$	$r(25,25)$
10桁	9.5e-9	1.0e-9	3.0e-8	2.6e-9	4.0e-8
20桁	1.9e-10	2.7e-16	1.6e-17	3.0e-18	3.6e-18
30桁	1.9e-10	1.5e-19	6.3e-24	1.6e-26	6.5e-26
40桁	1.9e-10	1.5e-19	1.3e-28	3.1e-31	4.7e-33
50桁	1.9e-10	1.5e-19	1.3e-28	1.4e-35	2.2e-43
60桁	1.9e-10	1.5e-19	1.3e-28	9.8e-38	6.5e-47

4.3 残差

前節では、有効桁数が少ないときには得られる解が不安定であるが、不必要な極を除けば正しい有理関数近似が得られることを前節で示した。そこで、有効桁数が少ない場合に得られる解の正当性を検証するために、残差の値を求めた。

表3の結果から、解の値が不安定となっている有効桁数の少ない部分でも、不必要な極を除けば残差の値を見ると十分な精度で正しく解けていると判断することができる。なお、解が不安定となる原因としては、入力時の浮動小数近似による丸め誤差の影響によって、Gauss消去法の計算過程で行われるピボット選択の選択場所が異なっていることなどが考えられる。

表 3: 残差 $\|Ax-b\|$ $(\log(x+2)$, 補間区間 $[1,2])$

有効桁	$r(5,5)$	$r(10,10)$	$r(15,15)$	$r(20,20)$	$r(25,25)$
10桁	1.4e-8	2.3e-9	6.2e-8	1.3e-8	1.5e-8
20桁	4.2e-19	2.3e-18	6.5e-19	2.3e-18	1.7e-18
30桁	1.8e-29	4.2e-28	2.0e-28	1.0e-28	3.4e-28
40桁	2.5e-39	6.0e-39	5.9e-38	7.3e-39	1.4e-38
50桁	2.4e-49	7.8e-49	2.5e-48	1.0e-47	2.0e-48
60桁	1.1e-59	4.8e-59	6.6e-58	9.7e-58	1.6e-57

4.4 対称関数の有理関数近似

補間区間内で対称となるような関数の近似を行う場合、補間を行う有理関数の分子分母の次数によっては、係数行列の行列式が厳密には0になるような場合がある。このような行列に対して厳密計算を行うと、Gauss消去法の計算過程でランク落ちが生じるため、解を一意に求めることはできない。ところが、浮動小数近似した値を利用すると、浮動小数近似による誤差からランク落ちを生じず、解が一意に求まってしまう場合がある。このような関数の例としては、Rungeの例として知られる $1/(25x^2+1)$ のような関数や $\cos(x)$ などの関数が挙げられる。厳密には行列式の値が0となりランク落ちを生じるこのような関数では、浮動小数近似することで行列式の値がわずかに0から離れた値となるので、条件数の値は有効桁数を上げるほど急激に悪化することになる。

このような浮動小数近似された値を利用してGauss消去法を行うと、本来はランク落ちを引き起こすはずの成分に誤差が含まれることで、非常に小さな値をとなり、連立一次方程式の解が一意に求まる。こうして求まった有理関数は、どれも不必要な極を生じる部分以外は正しく元の関数の近似できるものになっている。このような不必要な極を持つ関数は、ハイブリッド有理関数近似を用いて分子分母の共通因子を除去すれば、不必要な極のない高精度な有理関数を構成することができるので、本来は厳密計算では求めることができないような有理関数を、浮動小数近似することによって求めることができるようになる。この過程を、Rungeの例として知られる関数 $1/(25x^2+1)$ を例として以下に解析する。

4.4.1 Rungeの例の場合

Rungeの例は、元の関数が有理関数の形をしているため、得られる関数が元の有理関数と一致することが理想的である。ここで、分子m次、分母n次の有理関数を$r_{m,n}(x)$と表すとすると、$r_{1,2}(x), r_{2,2}(x)$による近似は、実際に元の関数と完全に一致する。次に、元の有理関数よりも明らかに次数の大きな有理関数$r_{3,3}(x)$や$r_{4,4}(x)$などを用いて近似を行うことを考える。

1. 厳密計算を行った場合
   厳密計算で$r_{3,3}(x)$による近似を求めようとすると、Gauss消去法の計算過程で次のようなランク落ちが生じる。

$$\left[\begin{array}{cccc|ccc|c} 1 & -1 & 1 & -1 & \frac{1}{26} & -\frac{1}{26} & \frac{1}{26} & \frac{1}{26} \\ 0 & 2 & 0 & 2 & -\frac{1}{13} & 0 & -\frac{1}{13} & 0 \\ 0 & 0 & -1 & 0 & 0 & \frac{1}{26} & 0 & \frac{25}{26} \\ 0 & 0 & 0 & -\frac{10}{27} & -\frac{125}{4251} & -\frac{250}{12753} & \frac{5}{4251} & -\frac{6250}{12753} \\ 0 & 0 & 0 & 0 & -\frac{1250}{24089} & -\frac{2050}{216801} & \frac{50}{24089} & -\frac{51250}{216801} \\ 0 & 0 & 0 & 0 & 0 & -\frac{100}{1989} & 0 & -\frac{2500}{1989} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

   ここで、未定係数aを利用すれば、厳密計算で得られる$r_{3,3}(x)$は次のようになる。

$$r_{3,3}(x) = \frac{25+ax}{(1+25x^2)(25+ax)}$$

   未定係数の含まれている項$(25+ax)$は分子分母の共通因子となっており、簡約化すればRungeの例の元の関数に一致することは明らかである。

   次に、ランク落ちを生じている最後の行の対角成分と対応する右辺の値を記号a,bと置き換え、以下のような行列を構成する。

$$\left[\begin{array}{cccc|ccc|c} 1 & -1 & 1 & -1 & \frac{1}{26} & -\frac{1}{26} & \frac{1}{26} & \frac{1}{26} \\ 0 & 2 & 0 & 2 & -\frac{1}{13} & 0 & -\frac{1}{13} & 0 \\ 0 & 0 & -1 & 0 & 0 & \frac{1}{26} & 0 & \frac{25}{26} \\ 0 & 0 & 0 & -\frac{10}{27} & -\frac{125}{4251} & -\frac{250}{12753} & \frac{5}{4251} & -\frac{6250}{12753} \\ 0 & 0 & 0 & 0 & -\frac{1250}{24089} & -\frac{2050}{216801} & \frac{50}{24089} & -\frac{51250}{216801} \\ 0 & 0 & 0 & 0 & 0 & -\frac{100}{1989} & 0 & -\frac{2500}{1989} \\ 0 & 0 & 0 & 0 & 0 & 0 & a & b \end{array}\right]$$

   これに対して後退代入を行い、有理関数の係数を求めると次のような有理関数が得られる。

$$\begin{aligned} r_{3,3}(x) &= \frac{1+\frac{b}{25a}x}{1+\frac{b}{25a}x+25x^2+\frac{b}{a}x^3} \\ &= \frac{25a+bx}{(1+25x^2)(25a+bx)} \end{aligned}$$

記号 a, b が含まれる項は、分子分母の共通因子となっており、簡約化すれば元の Runge の例の関数と一致することが分かる。

2. 数値計算を行った場合

数値計算によって $r_{3,3}(x)$ を求めようとすると、浮動小数近似による誤差から、本来はランク落ちを起こすはずの成分に微小な値が残り、一意に解を求めることができる。有効桁 5 桁で計算した場合は、次のようになる。

$$\left[\begin{array}{ccccccc|c} 1.0 & -1.0 & 1.0 & -1.0 & 0.038462 & -0.038462 & 0.038462 & 0.038462 \\ 0 & 2.0 & 0 & 2.0 & -0.076924 & 0 & -0.076924 & 0 \\ 0 & 0 & -1.0 & 0 & 0 & 0.038462 & 0 & 0.96154 \\ 0 & 0 & 0 & -0.3704 & -0.029402 & -0.019602 & 0.0011756 & -0.49004 \\ 0 & 0 & 0 & 0 & -0.051899 & -0.0094565 & 0.0020770 & -0.23643 \\ 0 & 0 & 0 & 0 & 0 & -0.050274 & 0.179\ \text{e-}6 & -1.2569 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.880\ \text{e-}5 & 0.539\ \text{e-}4 \end{array}\right]$$

ランク落ちを起こすはずの最後の行に、有効桁 5 桁に対して微小な値 -0.88008 e-6、0.53991 e-4 が含まれている。これによって、有理関数の係数は一意に求まり、次のようになる。

$$\begin{aligned} r_{3,3}(x) &\simeq \frac{1.0000 - 2.4541x + 0.5\ \text{e-}4\, x^2 + 0.9\ \text{e-}4\, x^3}{1 - 2.4550x + 25.001x^2 - 61.348x^3} \\ &= 0.1467\ \text{e-}5\, \frac{(x - 0.40749)(x + 165.61)(x - 164.65)}{(x - 0.40749)(x^2 - 0.3798\ \text{e-}4\, x + 0.040004)} \end{aligned}$$

$r_{3,3}(x)$ の分子分母には、$(x - 0.40749)$ という近似的な共通因子が存在していることがわかる。これが、不必要な極を生じる原因となる近似的な共通因子である。近似 GCD を用いてこれを除去すれば、

$$\tilde{r}(x) \simeq \frac{-2.4541}{-61.348x^2 - 2.4550} = \frac{1.0}{24.998x^2 + 1.0004}$$

となり、元の Runge の例の関数 $1/(25x^2+1)$ に対して高精度な近似になっていることが分かる。

次に、厳密計算の場合と同様にランク落ちを引きおこすはずの成分に a, b という記号を代入することを考える。ここでは、-0.88008 e-6,0.53991 e-4 をそれぞれ a, b で置き換えることになる。

$$\left[\begin{array}{ccccccc|c} 1.0 & -1.0 & 1.0 & -1.0 & 0.038462 & -0.038462 & 0.038462 & 0.038462 \\ 0 & 2.0 & 0 & 2.0 & -0.076924 & 0 & -0.076924 & 0 \\ 0 & 0 & -1.0 & 0 & 0 & 0.038462 & 0 & 0.96154 \\ 0 & 0 & 0 & -0.3704 & -0.029402 & -0.019602 & 0.0011756 & -0.49004 \\ 0 & 0 & 0 & 0 & -0.051899 & -0.0094565 & 0.0020770 & -0.23643 \\ 0 & 0 & 0 & 0 & 0 & -0.050274 & 0.179\ \text{e-}6 & -1.2569 \\ 0 & 0 & 0 & 0 & 0 & 0 & a & b \end{array}\right]$$

これに対して後退代入を行うと、以下のような有理関数が得られる。

$$\frac{a + 0.107\ \text{e-3}\ ax + 0.04bx + 0.5\ \text{e-4}\ ax^2 + 0.137\ \text{e-6}\ bx^2 - 0.1\ \text{e-3}\ ax^3 - 0.3\ \text{e-5}\ bx^3}{a + 0.0002ax + 0.040bx + 25.001ax^2 + 0.35772\ \text{e-5}\ bx^2 + bx^3}$$

このままでは分子分母に共通因子は存在しない。ここで、有効桁5桁であることを考慮して微小な係数を無視すると、

$$\tilde{r}_{3,3}(x) \simeq \frac{a + 0.04bx}{a + 0.04bx + 25ax^2 + bx^3} = \frac{(25.0a + 1.0bx)}{(1.0 + 25.0x^2)(25.0a + 1.0bx)}$$

となり、厳密計算を行った場合と同じ結果が得られる。また、ここで適当な値、例えば $a = 0.01, b = 50$ を代入すると、

$$\begin{aligned} r_{3,3}(x) &\simeq \frac{1. + 200.02x + 0.73795\ \text{e-3}\ x^2 - 0.015100x^3}{1 + 200.10x + 25.019x^2 + 5000.0x^3} \\ &= -0.302\ \text{e-5}\ \frac{(x + 0.0049995\ \text{e-2})(x + 115.07)(x - 115.12)}{(x + 0.0049975)(x^2 + 0.62948\ \text{e-6}\ x + 0.040020)} \end{aligned}$$

このままでは不必要な極を生じるが、近似GCDを取り除くと、

$$\tilde{r}(x) \simeq \frac{1.0}{24.998x^2 + 1.0004}$$

となり元の関数に近い有理関数が得られることが分かる。

同じことが、$r_{4,4}(x), r_{5,5}(x), \cdots$ にも見られる。$r_{4,4}(x)$ では共通因子は2次の多項式となり、$r_{5,5}(x)$ では共通因子は3次の多項式となる。そして、ハイブリッド有理関数近似を行うことで分子分母が2次の元の有理関数に一致するような有理関数に変換される。

4.4.2 $\log(x+2)$ の場合

このような入力の誤差やGauss消去法の計算過程で生じる誤差は、補間区間内で対称関数でないlogのような関数を近似した場合にも影響してくると考えられる。しかしこのような関数では、有理関数の分子分母の次数が何次であっても、厳密計算を行った場合にランク落ちが生じない。しかし、同じ有理関数による近似でも、有効桁数によってGauss消去法の計算過程には変化がみられる。例として、$\log(x+2)$ を補間区間 $[-1, 1]$ において $r_{3,3}(x)$ で近似することを考える。

1. 有効桁数が小さい場合
   有効桁3桁で計算を行ったとする。条件数の部分でも述べたように、条件数は残差が何倍拡大されて相対誤差に反映するかの倍率となっている。したがって、得られた近似解にある程度の精度を持たせるためには、条件数に対して十分な大きさの有効桁を用意して計算を行う必要がある。ここで、有効桁3桁というのは、$r_{3,3}(x)$ での有理関数近似の問題の条件数が 10^7 であることを考えれば、明らかに

精度が不足しているといえる。このように、条件数に対して不十分な精度で計算を行うと、次のような結果が得られる。

有効桁3桁でGauss消去法を適用した場合、三角化された行列は次のようになる。

$$\left[\begin{array}{ccccccc|c} 1.0 & -1.0 & 1.00 & -1.00 & 0 & 0 & 0 & 0 \\ 0 & 2.00 & 0 & 2.00 & -1.10 & -1.10 & -1.10 & 1.10 \\ 0 & 0 & -1.00 & 0.00100 & 0.550 & 0.549 & 0.549 & 0.144 \\ 0 & 0 & 0 & -0.371 & -0.0421 & 0.176 & 0.321 & -0.00158 \\ 0 & 0 & 0 & 0 & 0.0259 & -0.0729 & 0.283 & 0.00639 \\ 0 & 0 & 0 & 0 & 0 & 0.00362 & -0.0109 & 0.00279 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.00633 & -0.000298 \end{array}\right]$$

最後の行をみると、有効桁3桁に対して微小な値を持つ成分 -0.000298 が存在する。ここで、後退代入によって得られる有理関数は、次のようになる。

$$\begin{aligned} r_{3,3}(x) &\simeq \frac{0.693 + 2.25x + 1.57x^2 + 0.0133x^3}{1 + 2.53x + 0.629x^2 - 0.0471x^3} \\ &= 0.282\,\frac{(x + 117.)(x + 1.00)(x + 0.447)}{(x + 2.86)(x - 16.7)(x + 0.446)} \end{aligned}$$

最後の項 $(x+0.447)$ と $(x+0.446)$ が近似的な共通因子になっており、これによって得られた有理関数 $r_{3,3}(x)$ は不必要な極を生じる。そこで、近似GCDを用いてこの共通因子を除去する必要がある。ここで、先ほど示した三角化された行列の最後の行の微小な値となっている成分 -0.000298 を記号 a で置き換えて同様に後退代入を行うと次のような関数が得られる。

$$r_{3,3}(x) \simeq \frac{0.695 + (2.17 - 279.a)x + (1.61 + 128.a)x^2 + (0.135 + 407.a)x^3}{1 + (2.42 - 400.a)x + (0.771 + 475.a)x^2 + 158.ax^3}$$

このままでは、近似的な共通因子は存在しない。ところが、この a に適当な値、例えば $a = 10$ を代入すると、次のような共通因子を生じる。

$$\begin{aligned} r_{3,3}(x) &\simeq \frac{0.695 - 2790.x + 1280.x^2 + 4070.x^3}{1 - 4000.x + 4750.x^2 + 1580.x^3} \\ &= 2.58\,\frac{(x + 1.00)(x - 0.000249)(x - 0.685)}{(x + 3.69)(x - 0.000250)(x - 0.685)} \end{aligned}$$

このままでは不必要な極を生じるが、ハイブリッド有理関数近似を行うと分子分母が1次の有理関数になり、元の関数を近似することができる。

2. 有効桁数を大きくした場合

次に、条件数の大きさに対して十分な精度の有効桁を用意して上記のような計算を行う。$r_{3,3}(x)$ の場合の条件数は 10^7 なので、有効桁15桁で計算を行うことにする。すると、Gauss消去法で三角化された行列において、先ほど微小な値となっ

ていた成分は、$-0.166227168749054 \times 10^{-5}$ という有効桁 15 桁に対して十分な大きさを持った値となる。このとき、後退代入によって得られる有理関数は、分子分母に近似的な共通因子を持たない。ここで、この成分を a という記号に置き換えて同様の実験を行うと、a にどのような値を代入しても、分子分母には近似的な共通因子が現れない。

これらのことから分かるように、近似的な共通因子には、

- 本来はランク落ちを生じるような場合
- 条件数の大きさに対して計算精度が足りない場合

に現れることが分かる。そして、Runge の例のような場合には、近似的な共通因子は、元の関数に近づくために余分な多項式の次数を落す役割を果たしているようにみえる。

5 安定化理論を用いた有理関数補間

5.1 安定化理論

安定化理論 [4] は、不安定なアルゴリズムを安定なアルゴリズムに変換する手法として提案されているものである。代数的アルゴリズムは厳密計算での実行を前提としているため、入力に浮動小数近似した値を利用することで、真の解とは全く異なる値を導出してしまう場合がある。このようなアルゴリズムが不安定なアルゴリズムと呼ばれるもので、代表的なものとしては 0 判定の条件分岐を持つようなアルゴリズムがあげられる。

不安定なアルゴリズムの例を図 7 に示す。図 7 のアルゴリズムに対して $X = 1/3$ を

[入力] : X
[出力] : Zero または NonZero

[アルゴリズム] :
$Y = 3X - 1$
if $Y = 0$ then return Zero
else return NonZero

図 7: 不安定なアルゴリズム

入力すれば、正しく Zero が返されるが、$X = 0.3333...$ のような近似した値を入力すると NonZero が返される。これは、近似の精度をいくら上げても正しい出力 Zero を得ることができない。このように不安定なアルゴリズムでは、浮動小数近似した値を用いることによって条件分岐で誤った方向に進み、真の解に近づくことができなくなってしまうようなアルゴリズムである。

安定化理論は、このような不安定なアルゴリズムを入力に浮動小数近似した値を用いても真の解に近づくことができるような安定なアルゴリズムに変換する手法である。安定化の方法は次のようなものである。

1. アルゴリズムの構造は変えない。
2. データの係数を「区間係数」に置き換える。
3. ゼロ判定の条件文で、「区間係数のゼロ書き換え」を行う。
   (区間数が0を含むであれば、その区間を0に書き換える)

一般の精度保証付き数値計算に使用される区間演算では、区間数のゼロ書き換えは行われないが、安定化理論では敢えて区間数のゼロ書き換えを行うことで、条件分岐文で正しい方向に進むことができるようになっている。安定化理論では、真の解に近付くまで有効桁数を上げながら再計算を行う必要がある。有効桁数が小さい場合には、書き換える必要がない区間数までゼロ書き換えしてしまうことで正しい解が得られない場合があるが、有効桁数を上げて繰り返し計算を行うことで、必ず真の解に近づくことが理論的に証明されている[4]。

有理関数近似では、Gauss消去法のアルゴリズムを安定化することを考える。Gauss消去法では、明示的なゼロ判定の条件分岐は存在しないが、ランク落ちの判定をするためのゼロ判定が存在しており、安定化が有効なアルゴリズムであると考えられる。

5.2 結果

表 4: 安定化理論を用いた有理関数近似 ($\log(x+2)$, 補間区間 $[-1,1]$)
○... 不必要な極のない有理関数が得られる
×... 計算過程でランク落ちが生じる

有効桁	$r(5,5)$	$r(10,10)$	$r(15,15)$	$r(20,20)$
19桁	○	×	×	×
28桁	○	×	×	×
38桁	○	不正確	×	×
48桁	○	○	×	×
57桁	○	○	不正確	×
67桁	○	○	○	×
77桁	○	○	○	不正確
86桁	○	○	○	不正確
96桁	○	○	○	○

表 5: 安定化理論を用いた有理関数近似 ($\cos(5x^2)$, 補間区間 $[-1,1]$)

有効桁	$r(10,10)$	$r(15,15)$	$r(20,20)$	$r(25,25)$	$r(30,30)$
19 桁	×	×	×	×	×
28 桁	○	×	×	×	×
38 桁	○	×	不正確	×	×
48 桁	○	×	○	×	×
57 桁	○	×	○	×	×
67 桁	○	×	○	×	不正確
77 桁	○	×	○	×	○
86 桁	○	×	○	×	○
96 桁	○	×	○	×	○

安定化理論を用いた有理関数近似を行った場合の不必要な極のふるまいを表 4、表 5 に示す。安定化した Gauss 消去法を利用して有理関数近似を行った場合、有効桁数が小さいときにはゼロ書き換えの必要がない区間数までゼロ書き換えされてしまい、計算過程でランク落ちが生じる。一方、有効桁数を上げていくと、解が求まるようになる。始めのうちは求まった解は真の解とは離れた値であり、正しく元の関数を近似することができないが、さらに有効桁を上げれば真の解に収束する。結果としては、このときの有効桁が条件数が安定する有効桁数よりもを大きくなっているため、求まった解の値は安定しており、関数近似を行うと不必要な極を生じないような有理関数が得られる。また、表 5 のように厳密には行列式の値が 0 になる場合には、安定化手法を用いると必ずランク落ちが生じ、解が一意に求まらない。これらの結果から、安定化手法を利用すれば、厳密計算の性質に合った結果を得ることができるといえる。

6 まとめ

素朴な有理関数近似を行うことを考えた場合、得られた有理関数には不必要な極が生じることがある。そこで、不必要な極の出現に関する問題について詳しく検討し、具体的に計算を通じて以下のようなことが明らかになった。

1. 有理関数近似で現れる不必要な極は、この極に対応する分子の零点を近似 GCD によって取り除くと高精度な有理関数近似を得ることができる。
2. しかし、不必要な極と対応する零点の位置は有効桁数によって変化する。
3. 有理関数の係数を決定するための係数行列は、極めて悪条件である。
4. 悪条件行列を Gauss 消去法で解いているが、上のような不必要な極を生じる部分以外では極めて良好な近似ができる。

このような状況を検討し、問題を3種に分類することができた。

1. 補間区間内で対称な有理関数から得られるデータ列を厳密計算で近似する場合
   Runge の例 $1/(25x^2+1)$ を $r_{m,n}(x)(m>1,\ n>2)$ の有理関数で近似する問題について検討したところ、係数行列には全ての要素が0となる行が生じる。すなわち、ランク落ちを起こす。このようなランク落ちした行列の対角要素に記号を代入して、有理関数の係数を決定すると、得られる有理関数はこの記号を含むような分子分母の共通因子が生じ、これを簡約化すれば正しく元の有理関数になる。

2. 上の問題を浮動小数計算で近似する場合
   厳密な計算では全ての要素が0になる行が存在するが、浮動小数計算では誤差が混入するため、微小な値を持つ要素が現れるためランク落ちせず、係数を一意に決定することができる。有理関数の分子分母の係数は、この微小な値の関数になるが、これが不必要な極と零点の出現に大きく関与する。この微小な値は浮動小数計算の誤差であり、その値は一定ではないが、いずれの場合でもこの値は近似GCD で取り除き得る不必要な極と零点に対応しており、結果の近似には影響を及ぼさない。

3. 一般の関数から得られるデータを浮動小数計算で近似する場合
   本論では、$\log(x+2)$ について検討を行った。結果として、2. と非常に近いふるまいをしていることがわかった。不必要な極と零点の出現、またこれらの結果への関与については 2. と同様である。しかし、ここで大きな問題は、これら微小な値が単なる浮動小数計算における近似ではなく、有意な値となっている点である。したがって、有効桁を大きく増加させる場合、2. の微小な値は0に近づくか、一定の値として意味を持ち続けることになる。非常に高い精度で計算を行うと、2. の場合は0に近付くが、この場合は微小な値が意味を持ち始め、不必要な極を持たない有理関数をとなる。

以上のような問題に対して、安定化理論を用いて計算を行った。この場合には、解が得られるまで有効桁を増加させながら再計算を行うことになる。ここで、解が求まり、それが元の関数の近似になっている場合には、必ず不必要な極は生じない近似になっている。そのため、解を得るためには大きな有効桁が必要となるが、不必要な極のない高精度な近似であることを信頼することができる。また、厳密には行列式が0ならば、解が一意に定まらないことから、厳密計算の性質にあった結果を得ることができる手法であるといえる。

参考文献

[1] 甲斐 博：ハイブリッド有理関数近似の誤差評価, 情報処理学会論文誌 , Vol.40, No.4, pp.1754–1759, Apr.1999

[2] M.T.Noda and H.Kai: Hybrid rational function approximation and its accuracy analysis, *Reliable Computing* 6,pp.429–438,2000

[3] M.T.Noda, E.Miyahiro and H.Kai: Hybrid rational function approximation and its use in the hybrid integration, in "Advances in Computer Methods for Partial Differential Equations VII",eds.R. Vichnevetsky, D.Knight and G.Richter, IMACS, pp.565–571,1992

[4] K.Shirayanagi and M.Sweedler: A Theory of Stabilizing Algebraic Algorithms, *Technical Report* 95-28, Cornell University, 1995, pp.1-92.