ABSTRACTS

International Conference
on Interval and Computer-Algebraic Methods
in Science and Engineering ♦ Interval'94
March 7–10, 1994 ♦ St-Petersburg, Russia

Organized by:

International Journal "Interval Computations" ♦
St-Petersburg State University ♦
Faculty of Applied Mathematics and Control Processes
Institute of Computer Mathematics and Control Processes
Institute of New Technologies ♦
Matrix Computation of Subresultant Polynomial Remainder Sequences in Integral Domains

Alkiviadis G. Akritas\textsuperscript{1}, Evgenia K. Akritas\textsuperscript{1} and Genadii I. Malaschonok\textsuperscript{2}

\textsuperscript{1} — University of Kansas, Computer Science, Lawrence, KS 66045-2192, USA
\textsuperscript{2} — Kiev University, Department of Cybernetics, Vladimiriska 64, 252017 Kiev, Ukraine

1. Introduction

Let $I$ be an integral domain, and let

$$A_i = \sum_{j=1}^{m} a_{ij} x^{m-j},$$

where $a_{ij} \in I$, $i = 1, 2, \ldots, n$; then

$$\text{mat}(A_1, A_2, \ldots, A_n)$$

denotes the matrix $(a_{ij})$ of order $n \times m$. Moreover, let $A, B \in I[x], \deg A = m, \deg B = n$ and let

$$M_k = \text{mat}(x^{n-k-1}A, x^{n-k-2}A, \ldots, A, x^{m-k-1}B, x^{m-k-2}B, \ldots, B),$$

$$0 \leq k < \min(m, n)$$

be the matrix of order $(m + n - 2k) \times (m + n - k)$, where $M_0$ is the well-known Sylvester's matrix. Then, kth subresultant polynomial of $A$ and $B$ is called the polynomial

$$S_k = \sum_{i=0}^{k} M_i^k x^i,$$

of degree $\leq k$, where $M_i^k$ is a minor of the matrix $M_k$ of order $m + n - 2k$, formed by the elements of columns $1, 2, \ldots, m + n - 2k - 1$ and column $m + n - k - i$. Habicht's known theorem [4] establishes a relation between the subresultant polynomials $S_0, S_1, \ldots, S_{\min(m,n)-1}$ and the polynomial remainder sequence(prs) of $A$ and $B$, and also demonstrates the so-called gap structure.

According to the matrix-triangulization subresultant prs method (see [1] or [2]) all the subresultant polynomials of $A$ and $B$ can be computed within sign by transforming the matrix (suggested by Sylvester [7])

$$\text{mat}(x^{\max(m,n)-1}A, x^{\max(m,n)-1}B, x^{\max(m,n)-2}A, x^{\max(m,n)-2}B, \ldots, A, B),$$

of order $2 \cdot \max(m,n)$, into its upper triangular form with the help of Dodgson's integer preserving transformations [3]; they are then located using a theorem by Van Vleck [8] and its extension [2]. (We depart from established practice and we give credit to Dodgson, and not to Bareiss, for the integer preserving transformations [6]. Charles Lutwidge Dodgson (1832-1898) is the same person widely known for his writing *Alice in Wonderland* under the pseudonym Lewis Carroll.)

Below we propose a matrix-triangulization subresultant prs method allowing us to exactly compute and locate the members of the prs (without using Van Vleck's theorem [8]) by applying Dodgson's integer preserving transformations to a matrix of order $m + n$.

2. The Method and Related Theorems

We assume that $\deg A = m \geq \deg B = n$ and we denote by $M$, the following matrix

$$M = \text{mat}(x^{m-1}B, x^{m-2}B, \ldots, x^{n-1}B, x^{n-1}A, x^{n-2}B, x^{n-2}A, \ldots, B, A)$$

of order $m + n$ with elements $a_{ij}(j, i = 1, 2, \ldots, m + n)$. (This matrix can be obtained from Sylvester's matrix $M_0$ after a rearrangement of its rows.)

Dodgson's integer preserving transformations

$$a_{ij}^{k+1} = \left(\frac{\alpha^k_{ij}a_{ij}^k - \alpha^k_{ij}a_{ij}^k}{a_{k-1,i}a_{j,k-1}}\right)$$

(1)
(see [2], [3], [5] or [6]) where we set \( a_{00} = 1 \) and it is assumed that \( a_{kk} \neq 0, k = 1, 2, \ldots, m + n \), are applied to the matrix \( M = (a_{ij}) \) and transform it to the upper-triangular matrix \( M_D = (b_{ij}), (i, j = 1, 2, \ldots, m + n) \), where

\[
b_{ij} = \begin{cases} 
0 & \text{for } i > j \\
a_{ij} & \text{for } i \leq j 
\end{cases}
\]

and, in general,

\[
a_{ij}^k = \begin{pmatrix} 
a_{11} & \ldots & a_{1,k-1} & a_{1j} \\
\vdots & \ddots & \vdots & \vdots \\
a_{k-1,1} & \ldots & a_{k-1,k-1} & a_{k-1,j} \\
a_{i1} & \ldots & a_{i,k-1} & a_{ij} 
\end{pmatrix}
\]

with \( 1 \leq k \leq m + n \), and \( k \leq i, j \leq m + n \).

The following two theorems can be used to locate the members of the prs in the rows of \( M_D \); proofs will be presented elsewhere. The correct sign is computed.

Case 1: If none of the diagonal minors of the matrix \( M \) is equal to zero, then we have:

**Theorem 1.** Dodgson’s integer preserving transformation will transform matrix \( M \) to the upper triangular matrix \( M_D \), which contains all \( n \) subresultants (located in rows \( m + n - 2k, k = 0, 1, 2, \ldots, n - 1 \))

\[
S_k = \sum_{i=0}^{k} M_k^i x_i,
\]

where

\[
M_k^i = (-1)^{\sigma(k)} a_{i,j+k-i}^{m+n-2k}
\]

and

\[
\sigma(k) = (n - k + 1) + \cdots + (m - k) = \frac{(n - k)(2m - n - k + 1)}{2},
\]

\( k = 0, 1, \ldots, n - 1 \).

Case 2: If not all diagonal minors of the matrix \( M \) are nonzero, then we have:

**Theorem 2.** Dodgson’s integer preserving transformations with bubble pivot and choice of the pivot element by column, will transform matrix \( M \) to the upper triangular matrix \( M_D \), and at the same time will compute all subresultants \( S_k \); if, in the process, \( s \) row replacements take place, namely row \( j_1 \) replaces row \( i_1, j_2 \) replaces \( i_2, \ldots, j_s \), replaces \( i_s \), (and after each replacement row \( i_p \) is immediately below row \( j_p, p = 1, 2, \ldots, s \)), then (a) \( S_k = 0 \), for all \( k \) such that \( \frac{(m+n-j_p)}{2} > k > \frac{(m+n-i_p)}{2} \) and for all \( p = 1, 2, \ldots, s \). (b) for all \( p = 1, 2, \ldots, s \), if \( k = \frac{(m+n-i_p)}{2} \) is an integer number not in (a), \( S_k \) is located in row \( i_p \) before it is replaced by row \( j_p \). (c) for the remaining (c) \( k = 0, 1, \ldots, n - 1 \) and those not in (a) or (b) \( S_k \) is located in row \( j = m + n - 2k \).

Moreover, in (b) and (c) the subresultant \( S_k = \sum_{i=0}^{k} M_k^i x_i \), is located in row \( j \) in such a way that

\[
M_k^i = (-1)^{\sigma(k) + \sigma(j)} a_{i,j+k-i}
\]

where

\[
\sigma(k) = \frac{(n - k)(2m - n - k + 1)}{2},
\]

\[
\sigma(j) = \sum_{j=1}^{k} j_p - \sum_{p=1}^{s} i_p, j_p \leq j, i_p \leq j.
\]

Note that in cases (b) and (c) Theorem 2 reduces to Theorem 1 in the case of a complete prs, and due to the fact that rows above row \( j \) change places, the sign changes by a factor \((-1)^{\sigma(j)}\).

**References**

Implementation of Real Root Isolation Algorithms in Mathematica

Alkiviadis G. Akritas¹, Alexei V. Bocharov², Adam W. Strzeboński³

¹ — University of Kansas, Computer Science, Lawrence, KS, USA
² — Wolfram Research Inc., USA and Russian Academy of Science, Russia
³ — Wolfram Research Inc., USA and Jagiellonian University, Kraków, Poland

In this paper we compare two real root isolation methods (both derivatives of Vincent’s theorem of 1836) using Descartes’ Rule of Signs: the Interval Bisection method, and the Continued Fractions method. We present some time-saving improvements to both methods. Comparing computation times we conclude that the Continued Fractions method works much faster save for the case of very many very large roots.

1. Introduction

Isolation of real roots of univariate polynomials is the time-critical part of any algorithm for complex root isolation. Therefore the efficiency of the real root isolation algorithm is essential for developing efficient, guaranteed and precise root-finding strategies.

The present paper contains an analysis of 2 different techniques for real roots isolation; both are based on Vincent’s theorem of 1836 (see [1] and [2]) and Descartes’ rule of signs, and were proposed by the first named author (see [3], [1], [2] and article [4] for a survey of various techniques).

We discuss the behavior of the actual root isolation programs using the basic techniques along with the behaviour of a specific implementational variations thereof, using interval arithmetic for preprocessing the data.

The timing statistics over a wide variety of polynomials and segments seem to suggest that the Continued Fractions method of root isolation works significantly faster in most cases but that an actual implementation should be also able to switch to the Interval Bisection in a small class


