EXTENDING THE QD-ALGORITHM TO TACKLE MULTIVARIATE PROBLEMS

Annie Cuyt [1]

and Brigitte Verdonk [2]

Abstract. A lot has been said and done about the *qd*-algorithm [13, 9, 7, 5, 14]. Our main interest is to analyze how the original algorithm and its various improvements can be generalized for use in several multivariate applications. The present paper recalls the known univariate results in the sections 2.1 and 3, and discusses their multivariate generalization in the sections 2.2 and 4, but without going into all the multivariate details. We just make everything multivariate-ready for implementation in floating-point polynomial arithmetic (covering additional difficulties not encountered in exact polynomial arithmetic). The reader who is only familiar with the properties of the univariate *qd*-algorithm and does not have an extensive knowledge of the multivariate theory, can easily follow the analysis.

§1 Introduction: A state of the art in *qd*-algorithms

When studying the literature on the *qd*-algorithm, the results mainly have to be classified into two groups. A number of papers discuss the application of the *qd*-algorithm to the moments of a meromorphic function, for the calculation of its poles and multiplicities. This technique will be summarized in section 2. Another collection of publications discuss the use of the *qd*-algorithm applied to an underlying rational function, for the computation of the eigenvalues or singular values of certain tri- or bidiagonal matrices. These computational schemes are recalled in section 3. We stress that the main difference lies in the fact that the latter applications rely on the fact that the underlying function is rational and hence has a terminating continued fraction expansion.

[1]Research Director FWO-Vlaanderen, Department of Mathematics and Computer Science, Universiteit Antwerpen (UIA), Universiteitsplein 1, B–2610 Wilrijk, Belgium, email: cuyt@uia.ua.ac.be

[2]Postdoctoral Fellow FWO-Vlaanderen, Department of Mathematics and Computer Science, Universiteit Antwerpen (UIA), Universiteitsplein 1, B–2610 Wilrijk, Belgium, email: verdonk@uia.ua.ac.be

Computational Methods and Function Theory (CMFT'97)
N. Papamichael, St. Ruscheweyh and E. B. Saff (Eds.), pp. 1–??

ISBN 000-00-0000-0

In [11] the authors describe an ingenious way to transform a problem of the first kind into one of the second kind, at the expense of great computational complexity though. Our interest is in problems of the first kind, mainly with the ambition to generalize the application of the qd-algorithm to multivariate functions in order to obtain information on the location of their pole curves. Such methods, that decipher information on the poles of a function from its Taylor series development, are also closely related to the computation of Padé approximants or the formally orthogonal Hadamard polynomials. The Padé approximant $r_{n,m}(z)$ for a function $f(z)$ is known to be the irreducible form of a rational function $p(z)/q(z)$ of degree n in the numerator and m in the denominator satisfying $(fq-p)(z) = O(z^{n+m+1})$. Here we normalize the Padé approximant such that the denominator is monic.

By performing an equivalence transformation on the underlying continued fraction, the computation of the qd-table can be combined with that of the Hadamard polynomials into one scheme [7]. The main interest of this rewrite is in the fact that this last form of the qd-algorithm can also be applied to multivariate functions for the computation of their poles [6]. In section 4 we describe a new implementation of this symbolic qd-algorithm in floating-point polynomial arithmetic, instead of exact polynomial arithmetic, thus creating the possibility to work with real-life numeric data. This gives rise to a lot of new applications that we intend to discuss in future papers, such as numeric–symbolic factorization techniques for multivariate polynomials, numeric pole estimation of multivariate meromorphic functions etc.

§2 Meromorphic function case

2.1 The progressive qd-algorithm.

We immediately present the more stable progressive form of the qd-algorithm [10, p. 614]. Let the function $f(z)$ be known by its formal series expansion

$$f(z) = \sum_{i=0}^{\infty} c_i z^i \tag{2.1.1}$$

For arbitrary integers n and for integers $m \geq 0$ we define the Hankel determinants

$$H_m^{(n)} = \begin{vmatrix} c_n & c_{n+1} & \cdots & c_{n+m-1} \\ c_{n+1} & c_{n+2} & \cdots & c_{n+m} \\ \vdots & & & \vdots \\ c_{n+m-1} & c_{n+m} & \cdots & c_{n+2m-2} \end{vmatrix}$$

with $c_i = 0$ for $i < 0$ and $H_0^{(n)} = 1$. The series (2.1.1) is termed k-normal if $H_m^{(n)} \neq 0$ for $m = 0, 1, \ldots, k$ and $n \geq 0$. It is called ultimately k-normal if for every $0 \leq m \leq k$ there exists an $n(m)$ such that $H_m^{(n)} \neq 0$ for $n > n(m)$. With (2.1.1) as input we can define the qd-scheme as follows.

(a) The start values are given by

$$\begin{aligned}
q_k^{(-k+1)} &= 0 \qquad k = 2, 3, \ldots \\
e_k^{(-k+1)} &= \frac{w_{k+1}}{w_k} \qquad k = 1, 2, \ldots \\
e_0^{(k)} &= 0 \qquad k = 1, 2, \ldots \\
q_1^{(0)} &= \frac{c_1}{c_0}
\end{aligned}$$

where $w_0, w_1, \ldots$ are the Taylor coefficients of the function $1/f$. In other words,

$$\begin{aligned}
w_0 &= \frac{1}{c_0} \\
w_k &= -\frac{(c_1 w_{k-1} + \ldots + c_k w_0)}{c_0} \qquad k = 1, 2, \ldots
\end{aligned}$$

(b) The rhombus rules for continuation of the scheme are given by

$$\begin{aligned}
q_m^{(-m+n)} &= e_m^{(-m+n-1)} - e_{m-1}^{(-m+n)} + q_m^{(-m+n-1)} \\
&\qquad m = 1, 2, \ldots \quad n = 2, 3, \ldots \\
e_m^{(-m+n)} &= \frac{q_{m+1}^{(-m+n-1)}}{q_m^{(-m+n)}} e_m^{(-m+n-1)}
\end{aligned}$$

Usually the values $q_m^{(n)}$ and $e_m^{(n)}$ are arranged in a table where subscripts indicate columns and superscripts downward sloping diagonals and each continuation rule links four elements in a rhombus:

$$
\begin{array}{ccccccccc}
 & q_1^{(0)} & & q_2^{(-1)} & & q_3^{(-2)} & \cdots & & \\
e_0^{(1)} & & e_1^{(0)} & & e_2^{(-1)} & & e_3^{(-2)} & \cdots & \\
 & q_1^{(1)} & & q_2^{(0)} & & q_3^{(-1)} & \cdots & & \\
e_0^{(2)} & & e_1^{(1)} & & e_2^{(0)} & & e_3^{(-1)} & \cdots & \\
 & q_1^{(2)} & & q_2^{(1)} & & q_3^{(0)} & \cdots & & \\
e_0^{(3)} & & e_1^{(2)} & & e_2^{(1)} & & e_3^{(0)} & \cdots & \\
 & q_1^{(3)} & & q_2^{(2)} & & \ddots & & & \\
e_0^{(4)} & & e_1^{(3)} & & e_2^{(2)} & & & & \\
 & q_1^{(4)} & & q_2^{(3)} & & \ddots & & & \\
e_0^{(5)} & \vdots & e_1^{(4)} & \vdots & e_2^{(3)} & & & & \\
\vdots & & \vdots & & \vdots & & & &
\end{array}
$$

The following property can be found in [10].

Theorem 1. [10, pp. 612–613] *Let (2.1.1) be the Taylor series at $z = 0$ of a function f meromorphic in the disk $B(0, R) = \{z : |z| < R\}$ and let the poles z_i of f in $B(0, R)$ be numbered such that*

$$z_0 = 0 < |z_1| \leq |z_2| \leq \ldots < R$$

each pole occuring as many times in the sequence $\{z_i\}_{i \in \mathbb{N}}$ as indicated by its order. If f is ultimately k-normal for some integer $k > 0$, then the qd-scheme associated with f has the following properties (put $z_{k+1} = \infty$ if f has only k poles):

(a) For each m with $0 < m \leq k$ and $|z_{m-1}| < |z_m| < |z_{m+1}|$,

$$\lim_{n \to \infty} q_m^{(n)} = z_m^{-1} \tag{2.1.2}$$

(b) For each m with $0 < m \leq k$ and $|z_m| < |z_{m+1}|$,

$$\lim_{n \to \infty} e_m^{(n)} = 0 \tag{2.1.3}$$

Any index m such that the strict inequality

$$|z_m| < |z_{m+1}|$$

holds, is called a critical index. It is clear that the critical indices of a function do not depend on the order in which the poles of equal modulus are numbered. The theorem above states that if m is a critical index and f is ultimately m-normal, then

$$\lim_{n\to\infty} e_m^{(n)} = 0$$

Thus the qd-table of a meromorphic function is divided into subtables by those e-columns tending to zero. This property motivated Rutishauser [13] to apply the rhombus rules satisfied by the q- and e-values, namely

$$\begin{aligned} q_m^{(n+1)} e_m^{(n+1)} &= e_m^{(n)} q_{m+1}^{(n)} \\ e_{m-1}^{(n+1)} + q_m^{(n+1)} &= q_m^{(n)} + e_m^{(n)} \end{aligned}$$

in their progressive form as formulated above. When computing the q-values from the top down rather than from left to right, one avoids divisions by possibly small e-values that can inflate rounding errors. Any q-column corresponding to a simple pole of isolated modulus is flanked by such e-columns and converges to the reciprocal of the corresponding pole. If a subtable contains $j > 1$ columns of q-values, the presence of j poles of equal modulus is indicated. In [10, p. 642] it is also explained how to determine these poles if $j > 1$.

Theorem 2. [10, p. 642] *Let m and $m+j$ with $j > 1$ be two consecutive critical indices and let f be $(m+j)$-normal. Let the polynomials $v_k^{(n)}$ be defined by*

$$\begin{aligned} v_0^{(n)}(z) &= 1 \\ v_{k+1}^{(n)}(z) &= z v_k^{(n+1)}(z) - q_{m+k+1}^{(n)} v_k^{(n)}(z) \qquad n \geq 0 \qquad k = 0, 1, \ldots, j-1 \end{aligned}$$

Then there exists a subsequence $\{n(\ell)\}_{\ell \in I\!N}$ such that

$$\lim_{\ell\to\infty} v_j^{(n(\ell))}(z) = (z - z_{m+1}^{-1}) \ldots (z - z_{m+j}^{-1})$$

The polynomials $v_k^{(n)}$ are closely related to the formally orthogonal Hadamard polynomials [10, p. 629] which can be defined as follows. Let $A_m^{(n)}$ denote the matrix

$$A_m^{(n)} = \begin{pmatrix} q_1^{(n)} + e_0^{(n)} & q_1^{(n)} e_1^{(n)} & & & 0 \\ 1 & q_2^{(n)} + e_1^{(n)} & q_2^{(n)} e_2^{(n)} & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & q_{m-1}^{(n)} + e_{m-2}^{(n)} & q_{m-1}^{(n)} e_{m-1}^{(n)} \\ 0 & & & 1 & q_m^{(n)} + e_{m-1}^{(n)} \end{pmatrix}$$

Then the Hadamard polynomials $p_m^{(n)}$ are given by

$$p_0^{(n)}(z) = 1 \qquad p_m^{(n)}(z) = \det(zI - A_m^{(n)}) \qquad m = 1, 2, \ldots \tag{2.1.4}$$

and under the conditions of theorem 1 they satisfy, for each critical index m,

$$\lim_{n\to\infty} p_m^{(n)} = (z - z_1^{-1})(z - z_2^{-1}) \ldots (z - z_m^{-1}) \tag{2.1.5}$$

From the above theorems the qd-scheme seems to be an ingenious tool for determining, under certain conditions, the poles of a meromorphic function f directly from its Taylor series at the origin. This result is very much related to the convergence theorem of de Montessus de Ballore for Padé approximants of meromorphic functions [1, pp. 246–254]. The reason is that the q- and e-values appear in the partial numerators and denominators of the corresponding continued fraction for f

$$f(z) = c_0 + \sum_{i=1}^{\infty} \left(\frac{-q_i^{(1)} z \rfloor}{\lfloor 1} + \frac{-e_i^{(1)} z \rfloor}{\lfloor 1} \right) \tag{2.1.6}$$

The $(2m)^{th}$ convergent of (2.1.6) is the Padé approximant $r_{m,m}(z)$ for f of degree m in numerator and denominator. The separate treatment of poles of equal modulus is now easy to understand from the theorem of de Montessus de Ballore. Under the conditions of theorem 1, the poles of the Padé approximants with denominator degree m are approximations for the poles z_i of the function f. Now the qd-algorithm computes approximations for the values $1/z_i$ which are, again in the limit, the zeroes of the Hadamard polynomials of degree m. Since convergence of Padé approximants is obtained on compact subsets of a disk of meromorphy, poles equidistant from the origin (or the point around which f is developed into a Taylor series) cannot be treated separately. Increasing the radius of the disk (in qd-terminology moving to the next critical index) includes all the next poles of equal modulus simultaneously.

The next lemma gives an explicit formula for the q- and e-values in terms of the determinants $H_m^{(n)}$ which apparently play a crucial role.

Lemma 1. [10, pp. 610–613] *Let f be given by its formal Taylor series expansion (2.1.1). If there exists a positive integer k such that f is k-normal, then the values $q_m^{(n)}$ and $e_m^{(n)}$ exist for $m = 1, \ldots, k$ and $n \geq 0$ and they are given by*

$$q_m^{(n)} = \frac{H_m^{(n+1)} H_{m-1}^{(n)}}{H_m^{(n)} H_{m-1}^{(n+1)}}$$

$$e_m^{(n)} = \frac{H_{m+1}^{(n)} H_{m-1}^{(n+1)}}{H_m^{(n+1)} H_m^{(n)}}$$

2.2 The symbolic qd-algorithm.

By imposing another form than (2.1.6) for the corresponding continued fraction for f,

$$f(z) = c_0 + \sum_{i=1}^{\infty} \left(\frac{-Q_i^{(1)}(z) \rfloor}{\lfloor 1 + Q_i^{(1)}(z)} + \frac{-E_i^{(1)}(z) \rfloor}{\lfloor 1 + E_i^{(1)}(z)} \right) \qquad (2.2.1)$$

we obtain slightly different computation rules for the $Q_m^{(n)}(z)$ and $E_m^{(n)}(z)$ which are now rational functions of z [7]:

(a) help entries $g_{0,m}^{(n)}$ are given by

$$g_{0,m}^{(n)} = -c_{n-m+1} z^{n-m+1} \qquad m \geq 1 \quad n \geq 1 \qquad (2.2.2a)$$

$$g_{r,m}^{(n)} = \frac{g_{r-1,m}^{(n)} g_{r-1,r}^{(n+1)} - g_{r-1,m}^{(n+1)} g_{r-1,r}^{(n)}}{g_{r-1,r}^{(n+1)} - g_{r-1,r}^{(n)}} \qquad (2.2.2b)$$

$$1 \leq r \leq m-1 \quad m \geq 2 \quad n \geq 1$$

where the values $g_{r,m}^{(n)}$ are stored as in table 1

(b) and the symbolic qd-like algorithm is defined by

$$Q_1^{(n)}(z) = \frac{c_{n+1}}{c_n} \, z \, \frac{g_{0,1}^{(n)}}{g_{0,1}^{(n)} - g_{0,1}^{(n+1)}} \qquad n \geq 1 \qquad (2.2.3a)$$

$$E_m^{(n)}(z) + 1 = \frac{g_{m-1,m}^{(n+m-1)} - g_{m-1,m}^{(n+m)}}{g_{m-1,m}^{(n+m-1)}} \left(Q_m^{(n+1)}(z) + 1 \right) \qquad (2.2.3b)$$

$$m \geq 1 \quad n \geq 1$$

$$Q_m^{(n)}(z) = \frac{E_{m-1}^{(n+1)}(z) Q_{m-1}^{(n+1)}(z)}{E_{m-1}^{(n)}(z)} \frac{g_{m-2,m-1}^{(n+m-2)} - g_{m-2,m-1}^{(n+m-1)}}{g_{m-2,m-1}^{(n+m-2)}} \frac{g_{m-1,m}^{(n+m-1)}}{g_{m-1,m}^{(n+m-1)} - g_{m-1,m}^{(n+m)}}$$

$$m \geq 2 \qquad n \geq 1 \qquad (2.2.3c)$$

$g_{0,1}^{(1)}$	$g_{0,2}^{(1)}$			$g_{0,r}^{(1)}$				$g_{0,m}^{(1)}$	
		$g_{1,2}^{(1)}$			$g_{1,r}^{(1)}$				$\ddots$
$g_{0,1}^{(2)}$	$g_{0,2}^{(2)}$			$g_{0,r}^{(2)}$		$\ddots$		$g_{0,m}^{(2)}$	$g_{m-1,m}^{(1)}$
		$g_{1,2}^{(2)}$			$g_{1,r}^{(2)}$	$g_{r-1,r}^{(1)}$		$\vdots$	$\vdots$
$g_{0,1}^{(3)}$	$g_{0,2}^{(3)}$	$\vdots$	$\cdots$	$g_{0,r}^{(3)}$	$\vdots$	$\vdots$	$\cdots$		
$\vdots$	$\vdots$			$\vdots$		$g_{r-1,r}^{(n+m-r+1)}$			$g_{m-1,m}^{(n+1)}$
						$\iddots$			
		$g_{1,2}^{(n+m-1)}$			$g_{1,r}^{(n+m-1)}$				$\iddots$
$g_{0,1}^{(n+m)}$	$g_{0,2}^{(n+m)}$			$g_{0,r}^{(n+m)}$				$g_{0,m}^{(n+m)}$	

Table 1

The continued fraction (2.2.1) is especially interesting because it has the same form as the one underlying the general multivariate Padé approximation theory. The recursive computation rules (2.2.2) and (2.2.3) are no more than the univariate special case of the multivariate *qd*-like algorithm that was presented in [5]. The functions $Q_m^{(n)}(z)$ and $E_m^{(n)}(z)$ can be arranged in a table in the same way as the values $q_m^{(n)}$ and $e_m^{(n)}$ above. Symbolic or polynomial computation is necessary here because the $Q_m^{(n)}$ and $E_m^{(n)}$ in (2.2.1) are not merely coefficients anymore but rational functions. A progressive form of the computation rules (2.2.3) can be given but does not seem useful at first sight because (2.2.3c) retains the rational function $Q_m^{(n)}(z)$ as the quotient of two polynomials. As can be deduced from the next lemma and as will be detailed in section 4.2, a straightforward implementation of (2.2.3) involves a lot of greatest common divisor (GCD) computations and is not very economical. The implementation of (2.2.3) presented in section 4 takes into account all possible simplifications that remain valid in the multivariate version of this *qd*-like algorithm. Further simplifications that only hold in the univariate case would of course bring us back to the classical progressive *qd*-algorithm, and that is not the issue here.

Let us now first generalize lemma 1 and give explicit determinant representations for the rational expressions $Q_m^{(n)}(z)$ and $E_m^{(n)}(z)$. In addition to $H_m^{(n)}$ we define the determinants

$$H_{1,0}^{(n)} = 1 \qquad H_{1,m}^{(n)}(z) = \begin{vmatrix} z^m & \dots & 1 \\ c_n & \dots & c_{n+m} \\ \vdots & & \vdots \\ c_{n+m-1} & \dots & c_{n+2m-1} \end{vmatrix} \quad m \geq 1 \quad (2.2.4a)$$

$$H_{3,0}^{(n)}(z) = z^{n-1} \qquad H_{3,m}^{(n)}(z) = \begin{vmatrix} z^m & \dots & 1 \\ z^m \sum_{k=0}^{n-1} c_k z^k & \dots & \sum_{k=0}^{n+m-1} c_k z^k \\ c_n & \dots & c_{n+m} \\ \vdots & & \vdots \\ c_{n+m-2} & \dots & c_{n+2m-2} \end{vmatrix}$$

$$m \geq 1 \qquad (2.2.4b)$$

and, for $s \geq 1$,

$$\begin{aligned} H_m^{(n,s)} &= \begin{vmatrix} c_{n-s} & \dots & c_{n+m-1-s} \\ c_n & \dots & c_{n+m-1} \\ \vdots & & \vdots \\ c_{n+m-2} & \dots & c_{n+2m-3} \end{vmatrix} \qquad m \geq 2 \qquad (2.2.4c) \\ &= \begin{vmatrix} c_{n-s} & c_{n+1-s} & \dots & c_{n+m-1-s} \\ c_n & & & \\ \vdots & & H_{m-1}^{(n+1)} & \\ c_{n+m-2} & & & \end{vmatrix} \end{aligned}$$

It is easy to verify the following relations between the $H_{3,m}^{(n)}(z)$, the $H_m^{(n,s)}$ and the Hankel determinants $H_m^{(n)}$:

$$\begin{aligned} H_m^{(n,1)} &= H_m^{(n-1)} \\ H_{3,m}^{(n)}(z) &= z^{n+2m-1} H_m^{(n)} \end{aligned} \qquad (2.2.5)$$

By means of Sylvester's and Schweins' determinant identities we can prove the following lemma.

Lemma 2. [5] *Let f be given by its formal Taylor series expansion* (2.1.1). *If there exists a positive integer k such that f is k-normal, then for $m = 1, \dots, k$ and $n \geq 0$ the following values exist and are given by:*

$$Q_m^{(n)}(z) = -\frac{H_m^{(n+1)} H_{1,m-1}^{(n)}(z) H_{3,m}^{(n)}(z)}{H_m^{(n)} H_{1,m}^{(n)}(z) H_{3,m-1}^{(n+1)}(z)} \qquad (2.2.6a)$$

$$E_m^{(n)}(z) = -\frac{H_{m+1}^{(n)} H_{1,m-1}^{(n+1)}(z) H_{3,m}^{(n+1)}(z)}{H_m^{(n+1)} H_{1,m}^{(n+1)}(z) H_{3,m}^{(n)}(z)} \tag{2.2.6b}$$

$$g_{m-1,m}^{(n)}(z) = -z^n \frac{H_m^{(n-m+1)}}{H_{1,m-1}^{(n-m+2)}(z)} \tag{2.2.6c}$$

$$g_{r,m}^{(n)}(z) = -z^{n+1-m+r} \frac{H_{r+1}^{(n-r+1,m-r)}}{H_{1,r}^{(n-r+1)}(z)} \quad 1 \le r \le m-1 \tag{2.2.6d}$$

Let us now take a look at how the theorems 1 and 2 generalize into one powerful theorem for the new *qd*-like algorithm (2.2.3). To this end we introduce the monomial

$$\tilde{E}_m^{(n)}(z) = -\frac{H_{m+1}^{(n)} H_{3,m}^{(n+1)}(z) H_{m-1}^{(n+2)}}{H_m^{(n+1)} H_{3,m}^{(n)}(z) H_m^{(n+2)}} \tag{2.2.7}$$

that can be extracted from $E_m^{(n)}(z)$. We don't simplify the expression (2.2.7) for $\tilde{E}_m^{(n)}(z)$ further using (2.2.5), because in the multivariate case this is the way to specify $\tilde{E}_m^{(n)}(z)$. We also draw the reader's attention to the denominator factor $H_{1,m}^{(n)}(z)$ in $Q_m^{(n)}(z)$.

Theorem 3. [7] *Let (2.1.1) be the Taylor series at $z = 0$ of a function f meromorphic in the disk $B(0,R) = \{z : |z| < R\}$ and let the poles z_i of f in $B(0,R)$ be numbered such that*

$$z_0 = 0 < |z_1| \le |z_2| \le \ldots < R$$

each pole occuring as many times in the sequence $\{z_i\}_{i\in\mathbb{N}}$ as indicated by its order. If f is ultimately k-normal for some integer $k > 0$, then the symbolic qd-like scheme (2.2.3) associated with f has the following properties (put $z_{k+1} = \infty$ if f has only k poles):

(a) for each m with $0 < m \le k$ and $|z_m| < |z_{m+1}|$,

$$\lim_{n\to\infty} \frac{H_{1,m}^{(n)}(z)}{H_m^{(n+1)}} = (z - z_1)\ldots(z - z_m)$$

(b) for each m with $0 < m \le k$ and $|z_m| < |z_{m+1}|$,

$$\lim_{n\to\infty} \tilde{E}_m^{(n)}(z) = 0$$

both uniformly on compact subsets of $B(0,|z_{m+1}|)$.

From the above it is clear that, when trying to decipher pole information from Taylor coefficients, one is in fact interested in the functions $H_{1,m}^{(n)}(z)/H_m^{(n+1)}$ each time $|z_m| < |z_{m+1}|$, or in other words, for each m for which $\tilde{E}_m^{(n)}(z) \to 0$. We therefore still call every index m for which $\tilde{E}_m^{(n)}(z) \to 0$ a critical index. The information contained in $1/q_m^{(n)}$ in the classical qd-algorithm is now contained in the polynomial $H_{1,m}^{(n)}(z)$, which can be found in the denominator of the symbolic expression $Q_m^{(n)}(z)$. Similarly, the information contained in $e_m^{(n)}$ in the classical qd-algorithm is contained in $\tilde{E}_m^{(n)}(z)$ in the symbolic qd-algorithm. In fact it is easy to verify that

$$\tilde{E}_m^{(n)}(z) = -\frac{e_m^{(n)}}{q_m^{(n+1)}} z$$

It is also apparent that equimodular poles do not need a separate treatment such as in theorem 2 anymore: they are automatically retrieved in $H_{1,m}^{(n)}(z)/H_m^{(n+1)}$ from one critical index to another.

§3 Rational function case

If f is a rational function, even more properties of the qd-algortihm can be proven than is the case for a meromorphic f. Assume f has numerator degree n and denominator degree $m \leq n$, with finite poles $z_1, \ldots, z_m$ each pole occurring as many times as indicated by its multiplicity, just like above. Assume also that the series (2.1.1) is m-normal.

Theorem 4. *Let* (2.1.1) *be the Taylor series at $z = 0$ of a rational function of degree n in the numerator and $m \leq n$ in the denominator. Then if the series f is m-normal,*

(a) all the Hadamard polynomials $p_m^{(n-m+h)}(z)$ are identical:

$$p_m^{(n-m+h)}(z) = (z - z_1^{-1})(z - z_2^{-1}) \ldots (z - z_m^{-1}) \qquad h > 0$$

(b) and the m^{th} e-column equals zero:

$$e_m^{(n-m+h)} = 0 \qquad h > 0$$

thus terminating the computation of further q- and e-values.

The above theorem is again easy to understand when linking it to the theory of Padé approximants. For a rational function f of degree n in the numerator and m in the denominator, all Padé approximants $r_{n+h,m+h}(z)$ with

$h \geq 0$ are equal [8, p. 68–72]. This property translates to the normalized Padé denominators of $r_{n+h,m}(z)$ being equal. As has already been hinted at in (2.1.5) and as will become clear from lemma 3 and (4.1.3), there is a one-to-one link between the normalized Padé denominators and the Hadamard polynomials. Hence also all the Hadamard polynomials $p_m^{(n-m+h)}(z), h \geq 1$, are equal. Moreover, the consistency property of Padé approximants states that if f is a rational function of degree n in the numerator and m in the denominator, all $r_{n+h,m+h}(z) = f(z)$ [2, p. 30]. This then implies that the corresponding continued fraction (2.1.6) terminates, because it is a rational function, and hence that from a certain column on all q- and e-values are zero.

Since the polynomial $p_m^{(n-m+h)}(z)$, defined by (2.1.4), is the characteristic polynomial of the matrix $A_m^{(n-m+h)}$, the theorem above tells us that all the matrices $A_m^{(k)}$ with $k > n-m$, have the same eigenvalues. It is also easy to prove that the matrix $A_m^{(n-m+h)}$ has the same eigenvalues as the matrix $B_m^{(n-m+h)}$ where $B_m^{(n)}$ is defined by

$$B_m^{(n)} = \begin{pmatrix} q_1^{(n)} + e_0^{(n)} & -q_1^{(n)} & & & 0 \\ -e_1^{(n)} & q_2^{(n)} + e_1^{(n)} & -q_2^{(n)} & & \\ & \ddots & \ddots & \ddots & \\ & & -e_{m-2}^{(n)} & q_{m-1}^{(n)} + e_{m-2}^{(n)} & -q_{m-1}^{(n)} \\ 0 & & & -e_{m-1}^{(n)} & q_m^{(n)} + e_{m-1}^{(n)} \end{pmatrix}$$

Hence the qd-algorithm is an ingenious way to compute the eigenvalues of tridiagonal matrices $A_m^{(n-m+h)}$ or $B_m^{(n-m+h)}$. For distinct eigenvalues for instance [13, pp. 466–469], all indices $j = 1, \ldots, m-1$ are critical and hence $\lim_{h\to\infty} e_j^{(n-m+h)} = 0$ for $j = 1, \ldots, m-1$. In that case the values $q_j^{(n-m+h)}$ converge to the zeroes $1/z_j$ of the polynomial in theorem 4(a) and the eigenvalues can be read from the diagonal of $B_m^{(n-m+h)}$, for h sufficiently large.

In addition, if all q- and e-values in $A_m^{(n)}$ are positive, then computing the eigenvalues of $A_m^{(n)}$ is also equivalent to computing the eigenvalues of

$T_m^{(n)}$ where

$$T_m^{(n)} = \begin{pmatrix} q_1^{(n)} + e_0^{(n)} & \sqrt{q_1^{(n)} e_1^{(n)}} & & & 0 \\ \sqrt{q_1^{(n)} e_1^{(n)}} & q_2^{(n)} + e_1^{(n)} & \sqrt{q_2^{(n)} e_2^{(n)}} & & \\ & \ddots & \ddots & \ddots & \\ & & & q_{m-1}^{(n)} + e_{m-2}^{(n)} & \sqrt{q_{m-1}^{(n)} e_{m-1}^{(n)}} \\ 0 & & & \sqrt{q_{m-1}^{(n)} e_{m-1}^{(n)}} & q_m^{(n)} + e_{m-1}^{(n)} \end{pmatrix}$$

This is in turn equivalent to computing the singular values of the bidiagonal matrix

$$R_m^{(n)} = \begin{pmatrix} \sqrt{q_1^{(n)}} & \sqrt{e_1^{(n)}} & & & 0 \\ 0 & \sqrt{q_2^{(n)}} & \sqrt{e_2^{(n)}} & & \\ & \ddots & \ddots & \ddots & \\ & & & \sqrt{q_{m-1}^{(n)}} & \sqrt{e_{m-1}^{(n)}} \\ 0 & & & 0 & \sqrt{q_m^{(n)}} \end{pmatrix}$$

where $T_m^{(n)} = {R_m^{(n)}}^T R_m^{(n)}$ is the Cholesky decomposition of $T_m^{(n)}$.

Several variants of the progressive *qd*-algorithm have been developed in the framework of computing the eigenvalues of a tridiagonal matrix and the singular values and vectors of a bidiagonal matrix. We refer among others to the differential *qd*-algorithm [13, pp. 505–506], the *qd*-algorithm with shift [13, pp. 472–474], the stationary *qd*-algorithm [13, pp. 508–513] and the orthogonal *qd*-algorithm [9, 14]. The main aim of these variants is either to avoid the pitfalls of a floating-point implementation or to accelerate the convergence of the algorithm or both. These improved schemes, however, rely on the fact that the matrix elements satisfy certain properties (such as positivity). Such properties can not be assumed in general when using the *qd*-algorithm to locate the poles of a meromorphic function from its Taylor series expansion. The generalization of these variants will be the subject of future investigation in the framework of applying multivariate *qd*-schemes to the solution of parameterized eigenvalue problems. For the moment there is no need to incorporate these improvements in the numeric-symbolic *qd*-algorithm which is implemented in the next section for the detection of pole curves of meromorphic functions.

§4 Towards a numeric-symbolic implementation

4.1 More symbolic computation rules.

Let us first explicit the relation between the Hadamard polynomials from section 2.1 that play an important role through theorem 2, and the polynomials $H_{1,m}^{(n)}(z)/H_m^{(n+1)}$ from section 2.2 that according to theorem 3 contain valuable information.

Lemma 3. *The Hadamard polynomial* $p_m^{(n)}(z)$ *equals*

$$p_m^{(n)}(z) = (-1)^m z^m H_{1,m}^{(n)}(1/z)/H_m^{(n)} \qquad m \geq 0 \qquad n \geq 0$$

Proof: The proof is very easy if we make use of the fact that, as indicated in [10, p. 625], the Hadamard polynomials are monic and enjoy the determinant representation

$$\begin{aligned} p_0^{(n)} &= 1 \\ p_m^{(n)}(z) &= \frac{1}{H_m^{(n)}} \begin{vmatrix} c_n & \dots & c_{n+m-1} & 1 \\ \vdots & & \vdots & z \\ c_{n+m} & \dots & c_{n+2m-1} & z^m \end{vmatrix} \qquad n = 0, 1, 2, \dots \qquad m = 1, 2, \dots \end{aligned}$$

Rewriting this determinant formula as

$$\begin{aligned} p_m^{(n)}(z) &= \frac{(-1)^m z^m}{H_m^{(n)}} \begin{vmatrix} z^{-m} & c_n & \dots & c_{n+m-1} \\ z^{-m+1} & & & \\ \vdots & \vdots & & \vdots \\ 1 & c_{n+m} & \dots & c_{n+2m-1} \end{vmatrix} \\ &= \frac{(-1)^m z^m}{H_m^{(n)}} H_{1,m}^{(n)}(1/z) \end{aligned}$$

concludes the proof. ■

From theorem 3 in section 2.2 it is apparent that in the symbolic version of the *qd*-algorithm, one is mainly interested in the functions $H_{1,m}^{(n)}(z)$ (for pole information) and $\tilde{E}_m^{(n)}(z)$ (for critical index information). Extracting this information from the $Q_m^{(n)}(z)$ and $E_m^{(n)}(z)$ is only possible in exact polynomial arithmetic [6, 7], whereas here we are interested in obtaining this information also when dealing with inexact data. It can be observed from (2.2.6c) that the pole information $H_{1,m}^{(n)}(z)$ can be directly found in the denominator of $g_{m,m+1}^{(n+m-1)}(z)$. In order to obtain the $\tilde{E}_m^{(n)}(z)$ with as little computational effort as possible, we introduce the functions

$$\Delta_m^{(n)}(z) = \frac{H_{m+1}^{(n)} H_{3,m}^{(n+1)}(z)}{H_{1,m}^{(n)}(z) H_{1,m}^{(n+1)}(z)} \tag{4.1.1}$$

These satisfy a simpler recurrence relation than the $E_m^{(n)}(z)$.

Theorem 5.
$\Delta_m^{(n)}(z)$ *satisfies the recurrence relation*

$$\Delta_0^{(n)}(z) = c_n z^n \quad n \geq 1 \tag{4.1.2a}$$

$$\Delta_m^{(n)}(z) = g_{m-1,m}^{(n+m)} \left(\frac{\Delta_{m-1}^{(n+1)}(z)}{g_{m-1,m}^{(n+m)} - g_{m-1,m}^{(n+m-1)}} - \frac{\Delta_{m-1}^{(n+2)}(z)}{g_{m-1,m}^{(n+m+1)} - g_{m-1,m}^{(n+m)}} \right) \quad n \geq 1 \quad m \geq 1 \tag{4.1.2b}$$

Proof: A proof can again be obtained through the link with Padé approximation theory. Under the condition of m-normality, we can give a determinant representation of the Padé approximants for $f(z)$ in terms of the determinants $H_{1,m}^{(n)}(z)$ and the determinants $H_{2,m}^{(n)}(z)$ defined by

$$H_{2,0}^{(n)} = \sum_{k=0}^{n-1} c_k z^k \qquad H_{2,m}^{(n)}(z) = \begin{vmatrix} z^m \sum_{k=0}^{n-1} c_k z^k & \dots & \sum_{k=0}^{n+m-1} c_k z^k \\ c_n & \dots & c_{n+m} \\ \vdots & & \vdots \\ c_{n+m-1} & \dots & c_{n+2m-1} \end{vmatrix} \qquad m \geq 1$$

The Padé approximant $r_{n,m}(z)$ for f is given by [8]

$$r_{n,m}(z) = \frac{H_{2,m}^{(n-m+1)}(z)/H_m^{(n-m+2)}}{H_{1,m}^{(n-m+1)}(z)/H_m^{(n-m+2)}} \tag{4.1.3}$$

An application of Sylvester's determinant identity shows that

$$\Delta_m^{(n)}(z) = r_{n+m,m}(z) - r_{n+m-1,m}(z) \tag{4.1.4}$$

Using the E-algorithm [4, 3] one can show that

$$r_{n,m}(z) = \frac{r_{n,m-1}(z) g_{m-1,m}^{(n+1)} - r_{n+1,m-1}(z) g_{m-1,m}^{(n)}}{g_{m-1,m}^{(n+1)} - g_{m-1,m}^{(n)}}$$

Substituting this into (4.1.4) easily yields the desired recurrence relation for the $\Delta_m^{(n)}(z)$. ■

4.2 A numeric-symbolic GCD–free implementation.

The computation rules (2.2.2) and (4.1.2) will allow us to compute pole information and critical indices for a function f given by its Taylor series expansion. In order to retrieve this information from the rational functions $g^{(n)}_{m-1,m}$ and $\Delta^{(n)}_m$, the computed rational functions must be irreducible and normalized. Let us introduce the following notations to denote the numerator and denominator of the normalized irreducible rational functions $g^{(n)}_{r,m}$ and $\Delta^{(n)}_m$, where the normalization is such that the denominator polynomials of $g^{(n)}_{r,m}$ and $\Delta^{(n)}_m$ are monic:

$$\hat{g}^{(n)}_{r,m}(z) = -z^{n+1-m+r}\frac{H^{(n-r+1,m-r)}_{r+1}}{H^{(n-r+2)}_r} \tag{4.2.1a}$$

$$\overline{g}^{(n)}_{r,m}(z) = \frac{H^{(n-r+1)}_{1,r}(z)}{H^{(n-r+2)}_r} \tag{4.2.1b}$$

$$\hat{\Delta}^{(n)}_m(z) = \frac{H^{(n)}_{m+1}H^{(n+1)}_{3,m}(z)}{H^{(n+1)}_m H^{(n+2)}_m} \tag{4.2.1c}$$

$$\overline{\Delta}^{(n)}_m(z) = \frac{H^{(n)}_{1,m}(z)H^{(n+1)}_{1,m}(z)}{H^{(n+1)}_m H^{(n+2)}_m} \tag{4.2.1d}$$

The computation of pole and critical index information is reduced to computing $\hat{g}^{(n)}_{m-1,m}, \overline{g}^{(n)}_{m-1,m}$ and $\hat{\Delta}^{(n)}_m(z)$ as the following lemma indicates.
Lemma 4.

$$\frac{H^{(n)}_{1,m}(z)}{H^{(n+1)}_m} = \overline{g}^{(n+m-1)}_{m,m+1}(z)$$

$$\tilde{E}^{(n)}_m(z) = \frac{\hat{\Delta}^{(n)}_m(z)}{z^m \hat{g}^{(n+m-1)}_{m-1,m}(z)}$$

Proof: The result is an obvious consequence of (4.2.1) and (2.2.7), where we have used (2.2.5). ■

Our aim is now to implement the recursive computation rules (2.2.2) for $g^{(n)}_{m-1,m}(z)$ and (4.1.2) for $\Delta^{(n)}_m(z)$ in such a way that the computed rational functions are, as required, irreducible and normalized. Moreover, this will be achieved without GCD computations, in contrast to a straightforward symbolic implementation of (2.2.2) and (2.2.3). Using the notations (4.2.1), it is easy to rewrite the recursive rule (2.2.2) as

$$\frac{\hat{g}^{(n)}_{r,m}}{\overline{g}^{(n)}_{r,m}} = \frac{\hat{g}^{(n)}_{r-1,m}\hat{g}^{(n+1)}_{r-1,r} - \hat{g}^{(n+1)}_{r-1,m}\hat{g}^{(n)}_{r-1,r}}{\hat{g}^{(n+1)}_{r-1,r}\overline{g}^{(n)}_{r-1,r} - \hat{g}^{(n)}_{r-1,r}\overline{g}^{(n+1)}_{r-1,r}}$$

Let us point out that the left hand side of this equation is relatively prime, while the right hand side is not. By applying Sylvester's identity to the numerator and denominator of the right hand side, one can verify that the GCD of this rational function is $\hat{g}^{(n+1)}_{r-2,r-1}/z$. Hence

$$\frac{\hat{g}^{(n)}_{r,m}}{\overline{g}^{(n)}_{r,m}} = \frac{z(\hat{g}^{(n)}_{r-1,m}\hat{g}^{(n+1)}_{r-1,r} - \hat{g}^{(n+1)}_{r-1,m}\hat{g}^{(n)}_{r-1,r})/\hat{g}^{(n+1)}_{r-2,r-1}}{z(\hat{g}^{(n+1)}_{r-1,r}\overline{g}^{(n)}_{r-1,r} - \hat{g}^{(n)}_{r-1,r}\overline{g}^{(n+1)}_{r-1,r})/\hat{g}^{(n+1)}_{r-2,r-1}}$$

Let us also point out that in the left hand side the denominator $\overline{g}^{(n)}_{r,m}$ is monic while the denominator in the right hand side is not. We shall denote by $\gamma^{(n)}_r$ the highest degree coefficient of

$$z\frac{\hat{g}^{(n+1)}_{r-1,r}\overline{g}^{(n)}_{r-1,r} - \hat{g}^{(n)}_{r-1,r}\overline{g}^{(n+1)}_{r-1,r}}{\hat{g}^{(n+1)}_{r-2,r-1}}$$

Then

$$\begin{aligned}(\hat{g}^{(n)}_{r,m}, \overline{g}^{(n)}_{r,m}) = \texttt{Normalize}\Bigg(&\frac{z}{\hat{g}^{(n+1)}_{r-2,r-1}}\left(\hat{g}^{(n)}_{r-1,m}\hat{g}^{(n+1)}_{r-1,r} - \hat{g}^{(n+1)}_{r-1,m}\hat{g}^{(n)}_{r-1,r}\right), \\ &\frac{z}{\hat{g}^{(n+1)}_{r-2,r-1}}\left(\hat{g}^{(n+1)}_{r-1,r}\overline{g}^{(n)}_{r-1,r} - \hat{g}^{(n)}_{r-1,r}\overline{g}^{(n+1)}_{r-1,r}\right)\Bigg)\end{aligned} \tag{4.2.2}$$

The function `Normalize` is applied to make the denominator polynomial monic. In other words, it divides both its arguments by the highest degree coefficient of the second argument. In a similar way, we can deduce from (4.1.2) separate formulas for the numerator and denominator of $\Delta^{(n)}_m$. Starting from (4.1.2), we find

$$\begin{aligned}\frac{\hat{\Delta}^{(n)}_m}{\overline{\Delta}^{(n)}_m} &= \hat{g}^{(n+m)}_{m-1,m}\times \\ &\times\frac{\left(\hat{\Delta}^{(n+1)}_{m-1}\overline{g}^{(n+m)}_{m,m+1}\frac{\hat{g}^{(n+m+1)}_{m-2,m-1}}{z}\gamma^{(n+m)}_m - \hat{\Delta}^{(n+2)}_{m-1}\overline{g}^{(n+m-1)}_{m,m+1}\frac{\hat{g}^{(n+m)}_{m-2,m-1}}{z}\gamma^{(n+m-1)}_m\right)}{\overline{g}^{(n+m)}_{m-1,m}\overline{g}^{(n+m)}_{m,m+1}\frac{\hat{g}^{(n+m+1)}_{m-2,m-1}}{z}\gamma^{(n+m)}_m\overline{g}^{(n+m-1)}_{m,m+1}\frac{\hat{g}^{(n+m)}_{m-2,m-1}}{z}\gamma^{(n+m-1)}_m}\end{aligned}$$

Using (4.2.1d), we can deduce that the GCD of the rational function in the right-hand side of the above expression is

$$\frac{\overline{g}^{(n+m)}_{m-1,m}\hat{g}^{(n+m+1)}_{m-2,m-1}\gamma^{(n+m)}_m\hat{g}^{(n+m)}_{m-2,m-1}\gamma^{(n+m-1)}_m}{z^2}$$

and hence

$$(\hat{\Delta}_m^{(n)}, \overline{\Delta}_m^{(n)}) = \texttt{Normalize}\left(\frac{z^2 \hat{g}_{m-1,m}^{(n+m)}}{\overline{g}_{m-1,m}^{(n+m)} \hat{g}_{m-2,m-1}^{(n+m+1)} \gamma_m^{(n+m)} \hat{g}_{m-2,m-1}^{(n+m)} \gamma_m^{(n+m-1)}} \times \right.$$
$$(\hat{\Delta}_{m-1}^{(n+1)} \overline{g}_{m,m+1}^{(n+m)} \frac{\hat{g}_{m-2,m-1}^{(n+m+1)}}{z} \gamma_m^{(n+m)} - \hat{\Delta}_{m-1}^{(n+2)} \overline{g}_{m,m+1}^{(n+m-1)} \frac{\hat{g}_{m-2,m-1}^{(n+m)}}{z} \gamma_m^{(n+m-1)}),$$
$$\left. \overline{g}_{m,m+1}^{(n+m)} \overline{g}_{m,m+1}^{(n+m-1)} \right)$$
$$= \left(\frac{z^2 \hat{g}_{m-1,m}^{(n+m)}}{\overline{g}_{m-1,m}^{(n+m)} \hat{g}_{m-2,m-1}^{(n+m+1)} \gamma_m^{(n+m)} \hat{g}_{m-2,m-1}^{(n+m)} \gamma_m^{(n+m-1)}} \times \right.$$
$$(\hat{\Delta}_{m-1}^{(n+1)} \overline{g}_{m,m+1}^{(n+m)} \frac{\hat{g}_{m-2,m-1}^{(n+m+1)}}{z} \gamma_m^{(n+m)} - \hat{\Delta}_{m-1}^{(n+2)} \overline{g}_{m,m+1}^{(n+m-1)} \frac{\hat{g}_{m-2,m-1}^{(n+m)}}{z} \gamma_m^{(n+m-1)}),$$
$$\left. \overline{g}_{m,m+1}^{(n+m)} \overline{g}_{m,m+1}^{(n+m-1)} \right) \tag{4.2.3}$$

The main outcome of implementing the computation rules (2.2.2) and (4.1.2) as (4.2.2) and (4.2.3) is that the computation of GCDs at each application of the rules (2.2.2) and (4.1.2), in order to obtain the irreducible forms $\hat{g}_{r,m}^{(n)}/\overline{g}_{r,m}^{(n)}$ and $\hat{\Delta}_m^{(n)}/\overline{\Delta}_m^{(n)}$, is replaced by mere polynomial divisions in (4.2.2) and (4.2.3). By the analysis above we have shown that we know beforehand what the GCDs are, from the use of some determinant identities. We cannot overemphasize enough the reduction in computational complexity between a true GCD computation and a mere polynomial division, where it is moreover theoretically known that the polynomial remainder equals zero.

Let us take a closer look at the degrees and orders of the polynomials occurring in the right hand side of the equations (4.2.2) and (4.2.3).

Lemma 5. *Let f be given by its formal Taylor series expansion* (2.1.1). *If there exists a positive integer k such that f is k-normal, then for $m = 1, \ldots, k-1$ and $n \geq 0$ the following values exist and we have*

$\hat{\Delta}_m^{(n)}$ is a monomial of degree $n + 2m$
$\overline{\Delta}_m^{(n)}$ is a polynomial of degree $2m$
$\hat{g}_{m-1,m}^{(n)}$ is a monomial of degree n
$\overline{g}_{m-1,m}^{(n)}$ is a polynomial of degree $m - 1$

Proof: The proof is an obvious consequence of the determinant formulas (2.2.4) and definition (4.2.1), where we have also used the fact that

$$H_{3,m}^{(n)}(z) = z^{n+2m-1} H_m^{(n)}$$

■

Information on the order and the degree of the polynomials occurring in the right-hand side of the computation rules (4.2.2) and (4.2.3) is important for error correction and validation when computing in floating-point polynomial arithmetic, as we shall indicate now. Since the univariate $\hat{g}_{m-1,m}^{(n)}$ are monomials of degree n, it appears that only one polynomial division must be applied in the above computation rules for $(\hat{\Delta}_m^{(n)}, \overline{\Delta}_m^{(n)})$ and $(\hat{g}_{r,m}^{(n)}, \overline{g}_{r,m}^{(n)})$, namely the division by $\overline{g}_{m-1,m}^{(n+m)}$ for the computation of $\hat{\Delta}_m^{(n)}$. We shall now see that, moreover, this polynomial division further reduces to the simple division of highest degree coefficients. Using lemma 5, it can easily be verified that the first argument of `Normalize` in (4.2.3) can be written as

$$\frac{\rho_m^{(n)}(z)}{\overline{g}_{m-1,m}^{(n+m)}(z)} \tag{4.2.4}$$

where $\rho_m^{(n)}(z)$ is of the form

$$\rho_m^{(n)}(z) = \sigma_m^{(n)} z^{n+2m-1} \sum_{i=0}^{m} \tau_i z^i$$

while, also according to lemma 5, the expression (4.2.4) for $\hat{\Delta}_m^{(n)}$ has to reduce to a monomial of degree $n+2m$. Taking into account that $\overline{g}_{m-1,m}^{(n+m)}$ is a polynomial of degree $m-1$, this implies

$$\tau_0 = 0 \qquad \text{and} \qquad \overline{g}_{m-1,m}^{(n+m)}(z) | \sum_{i=1}^{m} \tau_i z^i \tag{4.2.5}$$

and hence

$$\frac{\rho_m^{(n)}(z)}{\overline{g}_{m-1,m}^{(n+m)}(z)} = \sigma_m^{(n)} z^{n+2m-1} \frac{\sum_{i=1}^{m} \tau_i z^i}{\overline{g}_{m-1,m}^{(n+m)}(z)} = \sigma_m^{(n)} \tau_m z^{n+2m} \tag{4.2.6}$$

In other words, the polynomial division by $\overline{g}_{m-1,m}^{(n+m)}(z)$ has reduced to determining the highest degree coefficient of a polynomial, where we have taken into account that $\overline{g}_{m-1,m}^{(n+m)}(z)$ is monic. We shall denote by `HDC` the function which, when given a polynomial as input, returns the highest degree coefficient of that polynomial.

Whereas (4.2.5) certainly holds when all computations are performed in exact polynomial arithmetic, this may no longer be the case when dealing with real-life numeric data or when performing the computations in floating-point polynomial arithmetic. Data and rounding errors may have as effect that

$$\tau_0 \neq 0 \qquad \text{or} \qquad \overline{g}_{m-1,m}^{(n+m)}(z) \not| \sum_{i=1}^{m} \tau_i z^i$$

Applying (4.2.6) to compute $\hat{\Delta}_m^{(n)}$ then in fact amounts to approximating the computed polynomial $\sum_{i=0}^{m} \tau_i z^i$ by its best polynomial approximation of degree m and order one in the ℓ_1–sense, in other words setting $\tau_0 = 0$.

Taking the above into account, we obtain the following algorithm in floating-point polynomial arithmetic for the computation of the poles and critical indices of the function f.

`Given :`

$$f(z) = \sum_{i=0}^{\infty} c_i z^i$$

`Initialization :`

$$\left.\begin{aligned} (\hat{g}_{0,1}^{(n)}, \overline{g}_{0,1}^{(n)}) &= (-c_n z^n, 1) \\ \hat{\Delta}_0^{(n)} &= c_n z^n \end{aligned}\right. \qquad \texttt{n = 1,...,nmax}$$

`For m = 1,2, ...`

- `Initialization of` $(\hat{g}_{0,m+1}^{(n)}, \overline{g}_{0,m+1}^{(n)})$:

$$(\hat{g}_{0,m+1}^{(n)}, \overline{g}_{0,m+1}^{(n)}) = (-c_{n-m} z^{n-m}, 1) \qquad \texttt{n = 1,...,nmax}$$

- `Computation of` $\hat{g}_{m,m+1}^{(n)}$ `and` $\overline{g}_{m,m+1}^{(n)}$:

$$(\hat{g}_{r,m+1}^{(n)}, \overline{g}_{r,m+1}^{(n)}) =$$
$$\texttt{Normalize}\left(\frac{z}{\hat{g}_{r-2,r-1}^{(n+1)}} \left(\hat{g}_{r-1,m+1}^{(n)} \hat{g}_{r-1,r}^{(n+1)} - \hat{g}_{r-1,m+1}^{(n+1)} \hat{g}_{r-1,r}^{(n)} \right), \right.$$
$$\left. \frac{z}{\hat{g}_{r-2,r-1}^{(n+1)}} \left(\hat{g}_{r-1,r}^{(n+1)} \overline{g}_{r-1,r}^{(n)} - \hat{g}_{r-1,r}^{(n)} \overline{g}_{r-1,r}^{(n+1)} \right) \right)$$

yielding

$$\gamma_r^{(n)} = \texttt{HDC}\left[\frac{z}{\hat{g}_{r-2,r-1}^{(n+1)}}(\hat{g}_{r-1,r}^{(n+1)}\overline{g}_{r-1,r}^{(n)} - \hat{g}_{r-1,r}^{(n)}\overline{g}_{r-1,r}^{(n+1)})\right]$$

r = 1,...,m and n = 1,...,nmax-r

- Computation of $\hat{\Delta}_m^{(n)}$:

$$\hat{\Delta}_m^{(n)} = \frac{z^{2n+3m}\hat{g}_{m-1,m}^{(n+m)}}{\hat{g}_{m-2,m-1}^{(n+m+1)}\gamma_m^{(n+m)}\hat{g}_{m-2,m-1}^{(n+m)}\gamma_m^{(n+m-1)}} \times \texttt{HDC}\left[\hat{\Delta}_{m-1}^{(n+1)}\overline{g}_{m,m+1}^{(n+m)}\frac{\hat{g}_{m-2,m-1}^{(n+m+1)}}{z}\gamma_m^{(n+m)} - \hat{\Delta}_{m-1}^{(n+2)}\overline{g}_{m,m+1}^{(n+m-1)}\frac{\hat{g}_{m-2,m-1}^{(n+m)}}{z}\gamma_m^{(n+m-1)}\right]$$

n = 1,...,nmax-2m

- Critical index information:

$$\tilde{E}_m^{(n)}(z) = \frac{\hat{\Delta}_m^{(n)}(z)}{z^m \hat{g}_{m-1,m}^{(n+m-1)}(z)} \qquad \texttt{n=0,1,..., nmax-2m}$$

- Pole information (if m is a critical index):

$$\overline{g}_{m,m+1}^{(n+m-1)}(z) = \frac{H_{1,m}^{(n)}(z)}{H_m^{(n+1)}} \qquad \texttt{n=0,1,..., nmax-2m+1}$$

All essential ideas discussed here for the univariate case can be carried over to the multivariate case. The multivariate algorithm analogously consists of an initialization phase with terms from the multivariate Taylor series expansion, followed by the computation of the multivariate $\overline{g}_{m,m+1}^{(n)}, \hat{g}_{m,m+1}^{(n)}$ and $\hat{\Delta}_m^{(n)}$. The latter two give information on which indices are critical in the multivariate case, while the first gives information on the pole curves of the multivariate function if the index m is critical. The multivariate exposition of the algorithm is burdened by the fact that what are monomials in the univariate case, become multivariate polynomials of homogeneous degree in the multivariate case. Also, it is necessary in the multivariate case to work with the notion of partial degree, which indicates the degree of the multivariate polynomial per variable. The reader interested in the multivariate case is referred to [5] for the multivariate equivalent of all the determinants given here while the actual multivariate algorithm will be described in a future paper.

§5 Numerical example

We conclude the paper by applying the classical qd-algorithm, the symbolic qd-algorithm and the numeric–symbolic qd-algorithm to compute the poles (with multiplicities) of the meromorphic function

$$f(z) = \frac{e^z}{(z-1)(z-2)(z+2)}$$

given by its Taylor series expansion.

`INPUT` classical qd-algorithm:

$$\text{floating} - \text{point representation of } (c_0, c_1, \ldots, c_{18})$$

`OUTPUT` :

$$\begin{aligned}
q_1^{(17)} &= 0.1000004\text{E}{+}01 \\
e_1^{(16)} &= -0.3674957\text{E}{-}05 \\
q_2^{(15)} &= 0.4479084\text{E}{+}00 \\
e_2^{(14)} &= -0.1102231\text{E}{+}00 \\
q_3^{(13)} &= -0.5581391\text{E}{+}00 \\
e_3^{(12)} &= -0.3003665\text{E}{-}07
\end{aligned}$$

`CONCLUSION` (see theorems 1 and 2):

$$\begin{aligned}
z_1 &\approx \frac{1}{q_1^{(17)}} = 0.9999960\text{E}{+}00 \\
z_i^{-2} &- (q_2^{(14)} + q_3^{(13)}) z_i^{-1} + q_2^{(13)} q_3^{(13)} \approx 0 \qquad i = 2,3 \\
z_2 &\approx 0.2000095\text{E}{+}01 \\
z_3 &\approx -0.2000032\text{E}{+}01
\end{aligned}$$

We now display the output of the symbolic qd-like algorithm (2.2.3), which relies on exact polynomial arithmetic and involves many GCD computations, for the same meromorphic function. Here we will denote by $\tilde{H}_{1,m}^{(n)}(z)$ the monic polynomial $H_{1,m}^{(n)}(z)/H_m^{(n+1)}$.

`INPUT` symbolic qd-like algorithm :

$$\sum_{i=0}^{18} c_i z^i \text{ with exact } c_i$$

`OUTPUT` :

$$\begin{aligned}
Q_1^{(17)}(z) &= \frac{-z}{z - 0.9999960} \\
E_1^{(16)}(z) &= \frac{3.674943\text{E}{-}06 * z}{z - 0.9999960} \\
Q_2^{(15)}(z) &= \frac{-z * (z - 0.9999842)}{(z - 2.232566) * (z - 0.9999990)} \\
E_2^{(14)}(z) &= \frac{0.2460840\text{E}{+}00 * z * (z - 0.9999842)}{(z - 2.232566) * (z - 0.9999990)} \\
Q_3^{(13)}(z) &= \frac{-z * (z - 0.9999961) * (z - 2.232565)}{(z - 1.000000) * (z - 2.000000) * (z + 2.000000)} \\
E_3^{(12)}(z) &= \frac{-5.387212\text{E}{-}08 * z * (z - 0.9999961) * (z - 2.232565)}{(z - 1.000000) * (z - 2.000000) * (z + 2.000000)}
\end{aligned}$$

`CONCLUSION` (see theorem 3):

$$\begin{aligned}
\tilde{E}_1^{(16)}(z) &= 3.674943\text{E}{-}06 * z \\
\tilde{H}_{1,1}^{(17)}(z) &= z - 0.9999960 \\
\tilde{E}_2^{(14)}(z) &= 0.2460840\text{E}{+}00 * z \\
\tilde{E}_3^{(12)}(z) &= -5.387212\text{E}{-}08 * z \\
\tilde{H}_{1,3}^{(13)}(z) &= (z - 1.000000) * (z - 2.000000) * (z + 2.000000)
\end{aligned}$$

It is important to emphasize that in the above algorithm all computations are performed in exact (rational) polynomial arithmetic. Only the final display of the results is done in floating-point format. This is in contrast to the new numeric–symbolic *qd*-algorithm presented in this paper, of which we display the output next.

`INPUT` numeric–symbolic *qd*-algorithm:

$$\text{floating} - \text{point representation of } (c_0, c_1, \ldots, c_{18})$$

`OUTPUT`:

$$\begin{aligned}
\frac{\hat{\Delta}_1^{(16)}}{z\hat{g}_{0,1}^{(16)}} &= \frac{3.32979\text{E}{-}07 * z^{18}}{0.90608 * z^{17}} \\
\overline{g}_{1,2}^{(17)} &= z - 0.9999960 \\
\frac{\hat{\Delta}_2^{(14)}}{z^2\hat{g}_{1,2}^{(15)}} &= \frac{3.27768\text{E}{-}06 * z^{18}}{1.33193\text{E-}05 * z^{17}}
\end{aligned}$$

$$\frac{\hat{\Delta}_3^{(12)}}{z^3\hat{g}_{2,3}^{(14)}} = \frac{-7.06301\text{E}{-13} * z^{18}}{1.31107\text{E}{-05} * z^{17}}$$

$$\overline{g}_{3,4}^{(15)} = 4.0000 - 4.0000z - z^2 + z^3$$

`CONCLUSION` (see lemmas 4 and 5):

$$\tilde{E}_1^{(16)}(z) = 3.674940\text{E}{-06} * z$$
$$\tilde{H}_{1,1}^{(17)}(z) = z - 0.9999960$$
$$\tilde{E}_2^{(14)}(z) = 0.2460840\text{E}{+00} * z$$
$$\tilde{E}_3^{(12)}(z) = -5.387212\text{E}{-08} * z$$
$$\tilde{H}_{1,3}^{(13)}(z) = 4.0000 - 4.0000 * z - z^2 + z^3$$

References

[1] G.A. Baker and P. Graves-Morris. *Padé approximants part I: basic theory.* Encycl. Math. 13, Addison Wesley, London, 1981.

[2] G.A. Jr. Baker and P. Graves-Morris. *Padé approximants (2nd Ed.).* Cambridge University Press, 1996.

[3] C. Brezinski. A general extrapolation algorithm. *Numer. Math.*, 35:175–187, 1980.

[4] A. Cuyt. A recursive computation scheme for multivariate rational interpolants. *SIAM J. Numer. Anal.*, 24:228–238, 1987.

[5] A. Cuyt. A multivariate qd-like algorithm. *BIT*, 28(86-15):98–112, 1988.

[6] A. Cuyt. Floating-point versus symbolic computations in a multivariate *qd*-like algorithm. *Communications Anal. Th. Cont. Fr.*, IV:84–90, 1995.

[7] A. Cuyt. Floating-point versus symbolic computations in the qd-algorithm. *J. Symbolic Computation*, 24:695–703, 1997.

[8] A. Cuyt and L. Wuytack. *Nonlinear methods in numerical analysis.* North-Holland, Amsterdam, 1987.

[9] K. Vince Fernando and Beresford N. Parlett. Accurate singular values and differential *qd*-algorithms. *Numer. Math.*, 67:191–229, 1994.

[10] P. Henrici. *Applied and computational complex analysis I.* John Wiley, New York, 1974.

[11] P. Kravanja and A. Haegemans. Computing zeros of analytic functions via the qd-algorithm. Technical Report TW 233, KU Leuven, 1995.

[12] O. Perron. *Die Lehre von den Kettenbruchen II.* Teubner, Stuttgart, 1977.

[13] H. Rutishauser. *Lectures on Numerical Mathematics.* Birkhaüser, Boston, 1990.

[14] U. Von Matt. The orthogonal QD-algorithm. *SIAM J. Sci. Statist. Comput.*, 18:1163–1186, 1997.