Applied Mathematics and Computation 309 (2017) 245–271

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Accurate quotient-difference algorithm: Error analysis, improvements and applications

Peibing Du [a,1], Roberto Barrio [b,2], Hao Jiang [c,3,*], Lizhi Cheng [a,d,4]

[a] School of Science, National University of Defense Technology, Changsha 410073, China
[b] Departamento de Matemática Aplicada and IUMA, University of Zaragoza, Zaragoza E-50009, Spain
[c] College of Computer, National University of Defense Technology, Changsha 410073, China
[d] The State Key Laboratory of High Performance Computation, National University of Defense Technology, Changsha 410073, China

ARTICLE INFO

Keywords:
Qd algorithm
Compensated qd algorithm
Error-free transformation
Rounding error
Continued fractions
Pole detection

ABSTRACT

The compensated quotient-difference (Compqd) algorithm is proposed along with some applications. The main motivation is based on the fact that the standard quotient-difference (qd) algorithm can be numerically unstable. The Compqd algorithm is obtained by applying error-free transformations to improve the traditional qd algorithm. We study in detail the error analysis of the qd and Compqd algorithms and we introduce new condition numbers so that the relative forward rounding error bounds can be derived directly. Our numerical experiments illustrate that the Compqd algorithm is much more accurate than the qd algorithm, relegating the influence of the condition numbers up to second order in the rounding unit of the computer. Three applications of the new algorithm in the obtention of continued fractions and in pole and zero detection are shown.

1. Introduction

The quotient-difference (qd) algorithm was proposed by Rutishauser from previous works of Hadamard [1], Aitken [2,3], and Lanczos [4] (for details see [5]). This algorithm is highly related to the Padé approximation [6–8] techniques. The qd algorithm, and its variants, have numerous applications. For instance, it can be used to obtain the continuous fraction representation of meromorphic functions given by its power series development [7–9]. It is also related with complex analysis, as it provides a direct method to locate poles of complex functions [9,10] and zeros of polynomials [10,11]. Besides, in eigenvalue computation, the progressive qd algorithm [10] has a relevant role as it can be interpreted as the LR transform for a tridiagonal matrix [12–14].

Unfortunately, in finite precision arithmetic, the quotient-difference algorithm has been shown in experiments to be numerically unstable. It is overly sensitive to rounding errors. As a consequence, high-precision arithmetic or exact arithmetic

* Corresponding author.
E-mail addresses: dupeibing10@nudt.edu.cn (P. Du), rbarrio@unizar.es (R. Barrio), haojiang@nudt.edu.cn, jhnudt@163.com (H. Jiang), clzcheng@nudt.edu.cn (L. Cheng).

1 Partially supported by National Natural Science Foundation of China (No. 61571008).
2 Partially supported by the Spanish Research project MTM2015-64095-P and by the European Social Fund and Diputación General de Aragón (Grant E48).
3 Partially supported by National Natural Science Foundation of China (Nos. 61402495, 61602166, 61303189, 61402496).
4 Partially supported by Science Project of National University of Defense Technology (JC120201) and National Natural Science Foundation of Hunan Province in China (13JJ2001).

http://dx.doi.org/10.1016/j.amc.2017.04.004
0096-3003/© 2017 Elsevier Inc. All rights reserved.

are recommended to overcome such a problem [15]. In order to increase the accuracy and stability of algorithms for ill-conditioned problems, several researchers studied their corresponding accurate compensated algorithms by applying error-free transformations [16–18] which can yield, in most circumstances, a full precision accuracy in standard precision. For instance, to evaluate ill-conditioned polynomials with floating-point coefficients, Graillat et al. [19–21] proposed a compensated Horner algorithm to evaluate polynomials in monomial basis; Jiang et al. [22–24] presented compensated de-Casteljau and Clenshaw algorithms to evaluate polynomials in Bernstein, Chebyshev and Legendre basis, respectively.

In this paper, we first perform a complete analysis of the stability of the quotient-difference algorithm by providing forward rounding error bounds and we introduce condition numbers adapted to the problem that permit to give a simple error bound that helps to locate the instability problems. The bounds shown in this paper provide a theoretical statement of the numerical simulations in literatu re. To overcome, or at least, to delay the appearance of instability problems in standard precision, we introduce a new more accurate algorithm, the compensated quotient-difference algorithm. The proposed algorithm is based on error-free transformations. To obtain the compensated quotient-difference algorithm we consider, especially, the division operation in each inner loop which has never been used in previous works of compensated algorithms. Again, we perform a complete analysis of the stability and now, from the forward rounding error bounds, we observe that the condition numbers are multiplied by the square of the rounding unit, instead of the rounding unit. This result states that the proposed compensated quotient-difference algorithm is much more stable than the standard quotient-difference algorithm in working precision.

The paper is organized as follows. In Section 2, we introduce the classical `qd` algorithm, some basic notations about floating-point arithmetic and error-free transformations. Section 3 presents the error analysis of the `qd` algorithm and its condition numbers. In Section 4, the proposed new compensated `qd` algorithm, `Compqd`, is provided. Section 5 presents the forward rounding error bounds of the `Compqd` algorithm. Finally, in Section 6, we give several numerical experiments together with three practical applications to illustrate the efficiency, accuracy and stability of the new `Compqd` algorithm. In the Appendices all the algorithms are detailed, and besides, a new compensated version of the *progressive* form of the `qd` scheme (`Compproqd` algorithm) is given.

2. Preliminaries

In this section, we review the classical `qd` algorithm (Section 2.1). In order to perform the detailed error analysis of the algorithms, we give some basic notations (Section 2.2) and we present the error-free transformations (Section 2.3).

2.1. The quotient-difference algorithm

Along this paper, quotient-difference is called `qd` for short and we assume that the conditions for the existence of the `qd` scheme (also known as the `qd` table [25]) are satisfied.

Considering the formal power series

$$f(z) = c_0 + c_1 z + c_2 z^2 + \cdots \equiv \sum_{k=0}^{\infty} c_k z^k, \tag{1}$$

where $c_i \in \mathbb{R}$, we define its double sequence of Hankel determinants by

$$H_m^{(n)} = \begin{vmatrix} c_n & c_{n+1} & \cdots & c_{n+m} \\ c_{n+1} & c_{n+2} & \cdots & c_{n+m+1} \\ \vdots & \cdots & \cdots & \vdots \\ c_{n+m} & c_{n+m+1} & \cdots & c_{n+2m} \end{vmatrix}, \qquad n, m \in \mathbb{N}.$$

A remarkable connection among Hankel determinants [7] is given by

$$(H_m^{(n)})^2 + H_{m+1}^{(n-1)} H_{m-1}^{(n+1)} = H_m^{(n-1)} H_m^{(n+1)}. \tag{2}$$

If we define

$$q_m^{(n)} = \frac{H_m^{(n+1)} H_{m-1}^{(n)}}{H_m^{(n)} H_{m-1}^{(n+1)}}, \quad e_m^{(n)} = \frac{H_{m+1}^{(n)} H_{m-1}^{(n+1)}}{H_m^{(n)} H_m^{(n+1)}}, \tag{3}$$

then the previous relationship (2) can be interpreted as the following addition rhombus rule

$$q_m^{(n)} + e_m^{(n)} = q_m^{(n+1)} + e_{m-1}^{(n+1)}, \tag{4}$$

and, considering the definition (3), $q_m^{(n)}$ and $e_m^{(n)}$ give the product rhombus rule

$$q_m^{(n+1)} e_m^{(n+1)} = q_{m+1}^{(n)} e_m^{(n)}. \tag{5}$$

Hence, both rhombus relations, (4) and (5), give rise to the classical `qd` algorithm: