IMA Journal of Numerical Analysis (2016) **36**, 421–436
doi:10.1093/imanum/drv005
Advance Access publication on March 18, 2015

Accuracy and stability of inversion of power series

Raymundo Navarrete and Divakar Viswanath*
Department of Mathematics, University of Michigan, Ann Arbor, MI, USA
*Corresponding author: divakar@umich.edu raymundo@umich.edu

[Received on 11 August 2014; revised on 7 January 2015]

This article considers the numerical inversion of the power series $p(x) = 1 + b_1x + b_2x^2 + \cdots$ to compute the inverse series $q(x)$ satisfying $p(x)q(x) = 1$. Numerical inversion is a special case of triangular back-substitution, which has been known for its beguiling numerical stability since the classic work of Wilkinson (1961, Error analysis of direct methods of matrix inversion. *J. Assoc. Comput. Mach.*, **8**, 281–330). We prove the numerical stability of inversion of power series and obtain bounds on numerical error. A range of examples show that these bounds overestimate the error by only a few digits. When $p(x)$ is a polynomial and $x = a$ is a root with $p(a) = 0$, we show that root deflation via the simple division $p(x)/(x - a)$ can trigger instabilities relevant to polynomial root finding and computation of finite-difference weights. When $p(x)$ is a polynomial, the accuracy of the computed inverse $q(x)$ is connected to the pseudozeros of $p(x)$.

Keywords: power series; root deflation; rounding errors; pseudozeros.

1. Introduction

Suppose $p(x)$ is the power series $1 + b_1x + b_2x^2 + \cdots$. We consider the numerical accuracy and stability of computing its multiplicative inverse $q(x) = 1 + c_1x + c_2x^2 + \cdots$, which satisfies $q(x) = 1/p(x)$. No assumption is made regarding the convergence of either series. It is only required that the Cauchy product $p(x)q(x) = 1$.

The algorithm for inverting power series is especially simple. It is a specialized form of triangular back-substitution. To find c_k, we use $c_k = -b_k - \sum_{j=1}^{k-1} c_jb_{k-j}$ in the order $k = 1, 2, 3, \ldots$

Inversion of power series arises as an auxiliary step in polynomial algebra, Hermite interpolation and computations related to Padé approximation (Butcher *et al.*, 2011; Sadiq & Viswanath, 2013), where a knowledge of its numerical properties would be useful. Yet the algorithm itself is so simple that it appears appropriate to state that the numerical properties of triangular back-substitution are especially subtle. In his classic paper (Wilkinson, 1961), Wilkinson provided a rounding error analysis of triangular back-substitution and remarked that the algorithm itself appeared more accurate than the error bounds. In particular, while the bounds predict relative error proportional to the condition number, the actual errors appear independent of condition numbers. Higham (1989, 2002) has refined and extended Wilkinson's analysis.

In Section 2.1, we consider the following calculation: a polynomial $p(x)$, for which $x = a$ is a root satisfying $p(a) = 0$, is deflated to compute $q(x) = p(x)/(x - a)$. This step arises in polynomial root finding as well as the computation of finite-difference weights. We show that an obvious method for deflating by a root has a catastrophic numerical instability. Indeed, a general method for calculating spectral differentiation matrices, implemented by Weideman & Reddy (2000), suffers from this instability as the order of the derivative increases, as shown earlier in Sadiq & Viswanath (2014). In Section 2.1, we show why the instability arises in a seemingly harmless situation.

It is well known that some natural and obvious methods for basic tasks such as computing the standard deviation or solving a quadratic equation are numerically unstable (Higham, 2002). Deflating a polynomial by a root is another example where an obvious method develops a numerical instability.

The problem of deflating by a root is related to, but not exactly the same as that of inverting a power series. In Section 2.2, we consider the special case of inverting a quadratic. These two problems of Section 2 bring to light some of the issues that arise in inverting power series in a relatively transparent manner.

The notion of pseudozeros due to Mosier (1986) (who called them root neighbourhoods) and in greater generality to Toh & Trefethen (1994) may be invoked to shed further light on rounding errors that arise during inversion of polynomials. The rounding errors in coefficients of the inverse series are eventually dominated by the polynomial root closest to the origin. However, the bounds based on pseudozeros and condition numbers are not good. Condition numbers are derived for each root separately. In contrast, the perturbative errors in the roots are finely correlated, and the correlation in errors leads to much better accuracy than the bounds indicate.

In Section 3, we give better bounds for the rounding errors that arise while inverting power series. These bounds imply the numerical stability of power series inversion. Computations that utilize extended precision arithmetic (with 100 digits of precision) show that the bounds are quite good. There is no significant gap between numerical condition and actual errors, unlike the situation with triangular matrices.

A significant contribution to explain the puzzle raised by Wilkinson (1961, p. 320), namely the observed independence of relative errors from condition numbers in triangular back-substitution, was made by Stewart (1997). Stewart has noted that triangular matrices that arise from Gaussian elimination or QR factorization are likely to be rank-revealing (in a sense explained in Section 3). For such matrices, Stewart has proved that the ill-conditioning can be eliminated using row scaling, thus partially explaining Wilkinson's observation. The triangular Toeplitz matrices associated with power series are typically not rank-revealing, but can be so in some situations, as shown in Section 3, but in these situations power series inversion is well-conditioned. Thus bounds for power series inversion are generally quite good, unlike the situation with triangular matrices.

2. Inversion of polynomials

In this section, we first consider deflating a polynomial $p(x)$ by factoring out $(x-a)$, where a is a root satisfying $p(a)=0$. Next, we look at the calculation of the multiplicative inverse of a quadratic polynomial and the theory of pseudozeros.

Following Higham (2002), but with some modifications, we set down the basic properties of floating-point arithmetic. The floating-point axiom is $\mathrm{fl}(\mathrm{x.op.y}) = (\mathrm{x.op.y})(1+\delta)$, where $|\delta| \leqslant u$. We may also write

$$\mathrm{fl}(\mathrm{x.op.y}) = (\mathrm{x.op.y})/(1+\delta),$$

where again $|\delta| \leqslant u$. Here, u is the unit round-off ($u = 2^{-53}$ for double precision arithmetic) and op may be addition, subtraction, division or multiplication.

To handle the accumulation of relative error through a succession of operations, it is helpful to introduce θ_n, which is any quantity that satisfies

$$1+\theta_n = (1+\delta_1)^{\rho_1}(1+\delta_2)^{\rho_2}\cdots(1+\delta_n)^{\rho_n}$$

for $|\delta_i| \leqslant u$ and with each ρ_i being $+1$, -1 or 0. In our usage, the θ variables are local to each usage. So, for example, if θ_3 occurs in two different equations or in two different places in the same equation, it is not the same θ_3, but each θ_3 is a possibly different relative error equal to the relative error from three (or fewer) operations. If a and b are of the same sign, we may write $a(1+\theta_n)+b(1+\theta_n) = (a+b)(1+\theta_n)$, but not if they are of opposite signs.

It may be shown (see Higham, 2002) that $|\theta_n| \leqslant \gamma_n$, where $\gamma_n = nu/(1-nu)$, if $nu < 1$. Unlike θ_n, γ_n stands for the same quantity in every occurrence. Whenever γ_n is used, the assumption $nu < 1$ is made implicitly. Another useful bound is $(1+\gamma_k)(1+\gamma_l) \leqslant 1+\gamma_{k+l}$.

2.1 *Deflation by $x - a$*

Let $p(x) = x^n + b_{n-1}x^{n-1} + \cdots + b_0$ and $p(a) = 0$. Consider

$$\frac{x^n + b_{n-1}x^{n-1} + \cdots + b_1 x + b_0}{x-a} = x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_1 x + c_0.$$

Equating coefficients, we get the equations

$$\begin{aligned} -ac_0 &= b_0, \\ -ac_1 + c_0 &= b_1, \\ &\ \vdots \\ -ac_{n-2} + c_{n-3} &= b_{n-2}, \\ -a + c_{n-2} &= b_{n-1}. \end{aligned} \tag{2.1}$$

We consider the accumulation of rounding error when these equations are solved for c_i in the order $c_0, c_1, \ldots, c_{n-2}$ using $c_k = (c_{k-1} - b_k)/a$ for $k > 1$. If $\hat{c}_k$ is the computed quantity in floating-point arithmetic, we assume inductively that

$$\hat{c}_{k-1} = -\frac{b_{k-1}}{a}(1+\theta_2) - \frac{b_{k-2}}{a^2}(1+\theta_4) - \cdots - \frac{b_1}{a^{k-1}}(1+\theta_{2k-2}) - \frac{b_0}{a^k}(1+\theta_{2k-2}).$$

Following the convention stated earlier, the two occurrences of θ_{2k-2} above are possibly different relative errors, each resulting from $2k-2$ or fewer operations. Since the recurrence $c_k = (c_{k-1} - b_k)/a$ involves two operations, we have

$$\hat{c}_k = -\frac{b_k}{a}(1+\theta_2) - \frac{b_{k-1}}{a^2}(1+\theta_4) - \cdots - \frac{b_1}{a^k}(1+\theta_{2k}) - \frac{b_0}{a^{k+1}}(1+\theta_{2k}).$$

Using $c_k = -\sum_{j=0}^{k} b_j/a^{k+1-j}$, we have the following bound for the rounding error in c_k.

THEOREM 2.1 If Equations (2.1) are solved for c_i in the order $c_0, c_1, \ldots, c_{n-2}$, and $\hat{c}_k$ is the computed value of c_k in floating-point arithmetic, we have

$$|\hat{c}_k - c_k| \leqslant \gamma_2 \left|\frac{b_k}{a}\right| + \cdots + \gamma_{2k}\left|\frac{b_1}{a^k}\right| + \gamma_{2k}\left|\frac{b_0}{a^{k+1}}\right| \leqslant \gamma_{2k} \sum_{j=0}^{k} \left|\frac{b_j}{a^{k+1-j}}\right|.$$

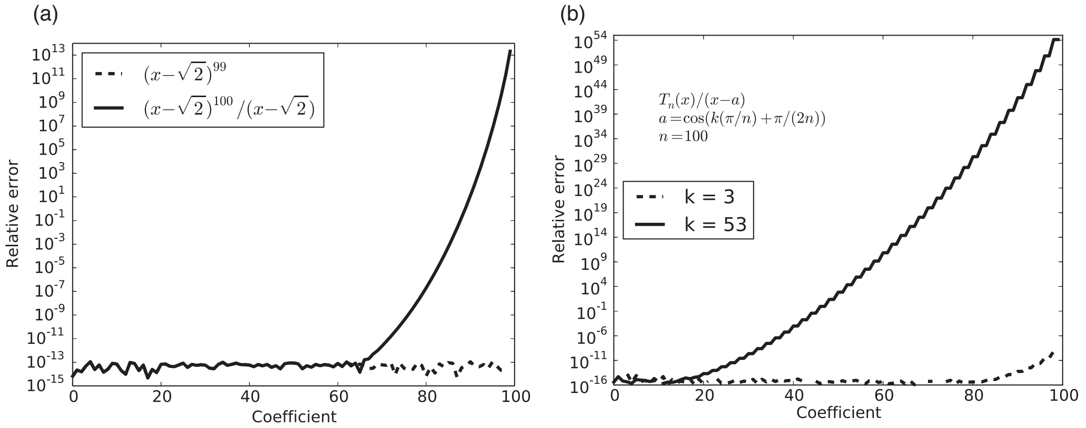

FIG. 1. Accumulation of rounding error in the coefficients when the polynomial $p(x)$ is deflated by $(x-a)$, a being a root.

Within the error bound of Theorem 2.1, there are two different mechanisms for large rounding errors. These two mechanisms are illustrated in Fig. 1. Figure 1(a) shows the relative errors in the coefficients of $(x-\sqrt{2})^{99}$ computed in two ways. The first computation begins with $(x-\sqrt{2})^{100}$ and then divides by $x-\sqrt{2}$. In the second computation, 99 factors $x-\sqrt{2}$ are multiplied. In the first computation, it is seen that the relative errors are initially small, but begin to explode after the halfway mark. In contrast, the relative errors remain small throughout in the second computation.

The error bound in Theorem 2.1 corresponds to the exact formula $c_k = -\sum_{j=0}^{k} b_j/a^{k+1-j}$. The relative error in c_k will be large if some of the terms of this sum are much larger than c_k. In the binomial expansion, the coefficients at the edges are much smaller than the ones in the middle. Thus deflation, using the method of Theorem 2.1, leads to large errors once we get past the middle. This is the first mechanism for large rounding errors.

The Chebyshev polynomial $T_n(x)$ is defined as $\cos(n \arccos x)$ for $x \in [-1,1]$. All its n roots are in the interval $[-1,1]$. Figure 1(b) shows the errors in the coefficients of $T_n(x)/(x-a)$, where a is a root close to 0 and when a is a root close to 1. The errors grow explosively for $a \approx 0$ ($k = 53$ in the plot), but are quite mild when $a \approx 1$ ($k = 3$ in the plot). Here too, as indicated by Theorem 1, there must be cancellations between the terms of $-\sum_{j=0}^{k} b_j/a^{k+1-j}$ for large relative errors. The cancellations can be particularly severe when a is small. This is the second mechanism for large rounding errors.

One of the methods for computing spectral differentiation matrices (Welfert, 1997; Weideman & Reddy, 2000) suffers from an instability related to the second mechanism. This instability has been completely fixed (Sadiq & Viswanath, 2014), yet we explain exactly how it comes about. In the original formulation (Welfert, 1997; Weideman & Reddy, 2000), the connection to polynomials and root deflation is not transparent.

Equation (7) of Welfert (1997), which is the heart of the algorithm in that paper, reads as follows:

$$(D_{p+1})_{k,j} = \frac{p+1}{x_k - x_j}\left(\frac{c_k}{c_j}(D_p)_{k,k} - (D_p)_{k,j}\right).$$

The grid points $x_0, x_1, \ldots, x_n$ figure in the denominator and are assumed to be distinct. Here, $(D_p)_{k,j}$ denotes the coefficient at x_j of the pth derivative at x_k. More specifically, if we seek to approximate the pth derivative of the function f at x_k using the function values at the grid points, the finite-difference formula is

$$f^{(p)}(x_k) = \sum_{j=0}^{n} (D_p)_{k,j} f(x_j) + \text{error}.$$

The c_i in the recurrence formula for $(D_p)_{k,j}$ are normalizing constants extraneous to the discussion here and will be ignored. They are unrelated to the c_i that appear in (2.1).

Let $l_j(x)$ denote the Lagrange cardinal function which is equal to 1 at x_j and 0 at the other grid points. The coefficients of u^p and u^{p+1} in the polynomial $l_j(u + x_k)$

$$\frac{(u + x_k - x_1)(u + x_k - x_2)\cdots(u + x_k - x_n)}{u + x_k - x_j},$$

multiplied by normalizing constants which we ignore, are equal to the pth and $(p+1)$st derivatives of the Lagrange cardinal function $l_j(x)$ evaluated at x_k, respectively (see Sadiq & Viswanath, 2014). Similarly, the coefficient of u^{p+1} of $\prod_{i=1}^{n}(u + x_k - x_i)$, multiplied by a normalizing constant which we ignore, is equal to the pth derivative of the Lagrange cardinal function $l_k(x)$ evaluated at x_k (Sadiq & Viswanath, 2014). Finite-difference weights are nothing but the coefficients of Lagrange cardinal functions, suitably normalized. It follows that Equation (7) of Welfert (1997) is using exactly the same recurrence as in Theorem 2.1, and is therefore susceptible to the instability exhibited above.

Root deflation is a part of the polynomial root finding algorithm due to Jenkins & Traub (1970). In contrast, the commonly used root finding algorithm based on companion matrices involves only implicit deflation. The Ehrlich–Aberth algorithm (Ehrlich, 1967; Aberth, 1973) iterates simultaneously for all the roots. In the Jenkins–Traub algorithm, Equations (2.1) are solved for c_i in the order $c_{n-2}, c_{n-3}, \ldots, c_0$. The bound in the following theorem is proved in much the same way as the bound in Theorem 2.1.

THEOREM 2.2 If Equations (2.1) are solved for c_i in the order $c_{n-2}, c_{n-3}, \ldots, c_0$ (bottom to top in (2.1)), the error in the computed quantity $\hat{c}_k$ satisfies the bound

$$|\hat{c}_k - c_k| \leqslant |b_{k+1}|\gamma_1 + |ab_{k+2}|\gamma_3 + \cdots + |a|^{n-k-1}\gamma_{2n-2k+1}.$$

The computation in Theorem 2.2 corresponds to the formula $c_k = a^{n-k-1} + \sum_{j=k+1}^{n-1} b_j a^{j-k-1}$. This appears a safer method because it is not vulnerable to the second mechanism when $a \approx 0$, and if the coefficients are well-scaled, we may assume that the roots a are not too large. However, it is still vulnerable to the first mechanism. For example, if this algorithm is applied to deflate a factor of $(x-a)^n$, large errors in the coefficients will occur for powers lower than $x^{n/2}$.

In the computation of finite-difference weights, both instability mechanisms are avoided by the method of partial products (Sadiq & Viswanath, 2014). In that method, the operation of deflating a polynomial by a factor is not employed. By analogy, it is natural to make the suggestion that polynomial root finding algorithms that avoid root deflation may be more accurate for each individual root. The operation count may be higher, but the polynomial root finding problems are puny compared to the power of modern computers. Thus accuracy is of greater concern.

2.2 *Inversion of a quadratic*

In a quadratic $ay^2 + by + c$ with $ac \neq 0$, we may make the change of variables $x = sy$, and choose the scale factor s to make the coefficients of x^2 and x^0 equal in magnitude. If the coefficient of x^2 is factored out, we are left with a quadratic of the form $x^2 + bx \pm 1$. The operations of factoring out the leading coefficient and rescaling the variable induce minimal relative error in the computed coefficients. Therefore, as far as the accumulation of error in the coefficients of the multiplicative inverse is concerned, we are left with only two cases:

$$\frac{1}{x^2 + bx \pm 1} = \pm 1 + c_1 x + c_2 x^2 + \cdots .$$

In the -1 case, we have $c_1 = -b$, $c_2 = bc_1 - 1$ and $c_{n+1} = bc_n + c_{n-1}$. It follows that c_i has the opposite sign to b if i is odd and is negative if i is even. There are no cancellations and all coefficients are computed with excellent relative accuracy. Both roots of the quadratic equation $x^2 + bx - 1 = 0$ are real.

The other case is with $+1$. In this case, we have

$$\begin{aligned} c_1 &= -b, \\ c_2 &= b^2 - 1, \\ c_3 &= -b^3 + 2b, \\ &\vdots \end{aligned}$$

In general, $c_{n+1} = -bc_n - c_{n-1}$. Each c_n is a polynomial in b: $c_n = F_n(b)$, where F_n is a polynomial of degree n. If α and β are the two distinct roots of $x^2 + bx + 1 = 0$, it follows that

$$c_n = F_n(b) = \frac{1}{\beta - \alpha}\left(\frac{1}{\beta^{n+1}} - \frac{1}{\alpha^{n+1}}\right). \tag{2.2}$$

To keep the discussion simple, we omit the cases $b = \pm 2$ with repeated roots. The polynomials F_n are a version of Fibonacci polynomials (Hoggatt & Bicknell, 1973).

An easy induction argument using the recurrence $c_{n+1} = -bc_n - c_{n-1}$ proves that the polynomial $c_{2n} = F_{2n}(b)$ has only even degree terms and that the coefficients alternate in sign beginning with b^{2n}. Similarly, $c_{2n+1} = F_{2n+1}(b)$ has only odd-degree terms and the coefficients alternate in sign beginning with $-b^{2n+1}$.

We write $c_n = F_n(b) = \sum_{k=0}^{n} C_{n,k} b^k$ and prove inductively that $\hat{c}_{n+1} = \sum_{k=0}^{n+1} C_{n+1,k} b^k (1 + \theta_{2n+2})$. The proof relies on the alternation in sign of the coefficients of $F_n(b)$ mentioned in the previous paragraph.

We may inductively assume that the computed quantity $\hat{c}_{n-1}$ is given by $\sum_{k=0}^{n-1} C_{n-1,k} b^k (1 + \theta_{2n-2})$ and that $\hat{c}_n = \sum_{k=0}^{n} C_{n,k} b^k (1 + \theta_{2n})$. The recurrence $c_{n+1} = -bc_n - c_{n-1}$ implies that $C_{n+1,k} b^k = -(C_{n,k-1} b^{k-1}) b - C_{n-1,k} b^k$. Crucially, $C_{n,k-1}$ and $C_{n-1,k}$ have the same sign, owing to the pattern in the signs of the coefficients of $F_n(b)$ and $F_{n-1}(b)$. Therefore

$$-C_{n,k-1} b^k (1 + \theta_{2n})(1 + \theta_2) - C_{n-1,k} b^k (1 + \theta_{2n-2})(1 + \theta_1) = C_{n+1,k} b^k (1 + \theta_{2n+2}),$$

and we may infer that $\hat{c}_{n+1} = \sum_{k=0}^{n+1} C_{n+1,k} b^k (1 + \theta_{2n+2})$, completing the induction.

The error bound

$$\frac{|c_n - \hat{c}_n|}{|c_n|} \leqslant \frac{|F_n|(|b|)}{|F_n(b)|}\gamma_{2n},$$

where $|F_n|$ is the polynomial with all coefficients of F_n replaced by their absolute values, follows immediately. If we go back to formula (2.2) for c_n, we get a sense of when the relative errors in the computed coefficients may be large. If $|b| < 2$, both roots α and β of $x^2 + bx + 1$ are complex of magnitude 1 and conjugates of each other. For certain values of n, the arguments of α^{n+1} and β^{n+1} will differ very nearly by a multiple of 2π and formula (2.2) implies a cancellation making $F_n(b)$ much smaller in magnitude than $|F_n|(|b|)$. The corresponding coefficients c_n will have large relative errors.

2.3 *Connection to pseudozeros*

Let $p = p_0 + p_1 z + p_{n-1} z^{n-1} + z^n$ be a monic polynomial and let $Z(p) = \{a_1, a_2, \ldots, a_n\}$ be the set of roots of p. We assume $p_0 \neq 0$. We shall connect the errors in computing the inverse series $q(z) = 1/p(z)$ to the pseudozeros of $p(z)$. The analysis here is of conditioning, not of rounding errors. We consider another monic polynomial $\hat{p}$ close to p and bound the errors in $\hat{q} = 1/\hat{p}$ using the pseudozero sets of p. The subscripted variable p_i denotes the coefficient of z^i in $p(z)$. Similarly, q_i denotes the coefficient of z^i in $q(z)$.

Pseudozero sets have been defined using the infinity norm (Mosier, 1986) or more general norms (Toh & Trefethen, 1994). Here, we define pseudozero sets using the maximum coefficient-wise relative error. Our definition is close to that of Mosier (1986). Let

$$e(\hat{p}) := \max_{i, p_i \neq 0} \frac{|p_i - \hat{p}_i|}{|p_i|}$$

be the maximum coefficient-wise error in $\hat{p}$ relative to p. The ϵ-pseudozero set of p in the complex plane is given by

$$Z_\epsilon(p) := \{z \in \mathbb{C} : z \in Z(\hat{p}),\ e(\hat{p}) \leqslant \epsilon\}.$$

An argument in Mosier (1986) (also see Toh & Trefethen, 1994) implies that

$$Z_\epsilon(p) = \left\{ z \in \mathbb{C} : \frac{|p(z)|}{|p|(|z|)} \leqslant \epsilon \right\},$$

where $|p|$ is the polynomial with all coefficients of p replaced by their absolute values.

Suppose $\hat{a} \in Z_\epsilon(p)$ and let $a \in Z(p)$, with $a = a_i$ for some i, be the root closest to $\hat{a}$. All the roots a_i of $p(x) = 0$ are assumed to be distinct, to avoid technicalities of no value for the discussion here. Then,

$$|a - \hat{a}|^n \leqslant \prod_{i, p(a_i)=0} |\hat{a} - a_i| = |p(\hat{a})| \leqslant \epsilon |p|(|\hat{a}|)$$

since p is a monic polynomial. We have

$$|a - \hat{a}| \leqslant \sqrt[n]{\epsilon |p|(|\hat{a}|)},$$

but this bound on the error is highly pessimistic. This bound is reasonably good only if $|\hat{a} - a| \approx |\hat{a} - a_i|$ for every i, which is very seldom the case.

Condition numbers of polynomial roots (Gautschi, 1984; Toh & Trefethen, 1994) may be used to derive tighter bounds. If a_j is a simple root of p, we may define

$$\kappa(a_j, p) := \lim_{e(\hat{p})\to 0} \sup_{\hat{p}} \frac{|a_j - \hat{a}_j|}{e(\hat{p})},$$

where $\hat{a}_j$ is the root of $\hat{p}$ corresponding to a_j and $e(\hat{p})$ is the maximum relative coefficient-wise distance of $\hat{p}$ from p defined earlier. If $e(\hat{p}) < \epsilon$ and $\epsilon \to 0$, we have

$$\frac{p(\hat{a}_j)}{p'(a_j)(\hat{a}_j - a_j)} = \frac{(\hat{a}_j - a_1)\cdots(\hat{a}_j - a_{j-1})(\hat{a}_j - a_{j+1})\cdots(\hat{a}_j - a_n)}{(a_j - a_1)\cdots(a_j - a_{j-1})(a_j - a_{j+1})\cdots(a_j - a_n)} \to 1$$

implying $\hat{a}_j - a_j \approx p(\hat{a}_j)/p'(a_j)$. Therefore, we have

$$\kappa(a_j, p) = \lim_{\epsilon\to 0} \sup_{\hat{p}, e(\hat{p})\leqslant\epsilon} \frac{|p(\hat{a}_j)|/|p'(a_j)|}{e(\hat{p}_j)} = \frac{|p|(|a_j|)}{|p'(a_j)|}$$

noting that the inequality $|p(\hat{a})| \leqslant \epsilon |p|(|\hat{a}|)$ is sharp for some polynomial $\hat{p}$ with $e(\hat{p}) = \epsilon$ (see Mosier, 1986).

If p has only distinct roots as assumed, we have

$$q(z) = \frac{\operatorname{Res}(q, a_1)}{(z - a_1)} + \cdots + \frac{\operatorname{Res}(q, a_n)}{(z - a_n)},$$

where the residue of q at one of its simple poles a_j is given by $\operatorname{Res}(q, a_j) = 1/[(a_j - a_1)\cdots(a_j - a_{j-1})(a_j - a_{j+1})(a_j - a_n)]$. We may expand q as

$$q(z) = \sum_{j=1}^{n} \operatorname{Res}(q, a_j)\left(\frac{-1}{a_j}\right)\sum_{k=0}^{\infty}\left(\frac{z}{a_j}\right)^k = \sum_{k=0}^{\infty}\left(\sum_{j=1}^{n}\frac{-\operatorname{Res}(q, a_j)}{a_j^{k+1}}\right) z^k,$$

with the infinite sum being convergent if and only if $|z| < \min_j |a_j|$. Let $\hat{q} = 1/\hat{p}$, where $e(\hat{p}) \leqslant \epsilon$, and let $Z(\hat{p}) = \{\hat{a}_1, \ldots, \hat{a}_n\}$ with $\hat{a}_i$ corresponding to a_i, with ϵ assumed small enough that the correspondence may be set up. The error in the coefficient of z^k is

$$(q - \hat{q})_k = \sum_{j=1}^{n}\left(\frac{\operatorname{Res}(\hat{q}, \hat{a}_j)}{\hat{a}_j^{k+1}} - \frac{\operatorname{Res}(q, a_j)}{a_j^{k+1}}\right).$$

A perturbative calculation of error, assuming ϵ so small that $\Delta a_i = \hat{a}_i - a_i$ satisfies $|\Delta a_i| \ll |a_j - a_k|$ for any i, j, k, follows. The perturbative calculation is based on

$$\operatorname{Res}(\hat{q}, \hat{a}_j) = \operatorname{Res}(q, a_j)\left(1 - \sum_{i \neq j}\frac{\Delta a_j - \Delta a_i}{a_j - a_i}\right) + \mathcal{O}(\Delta a^2)$$

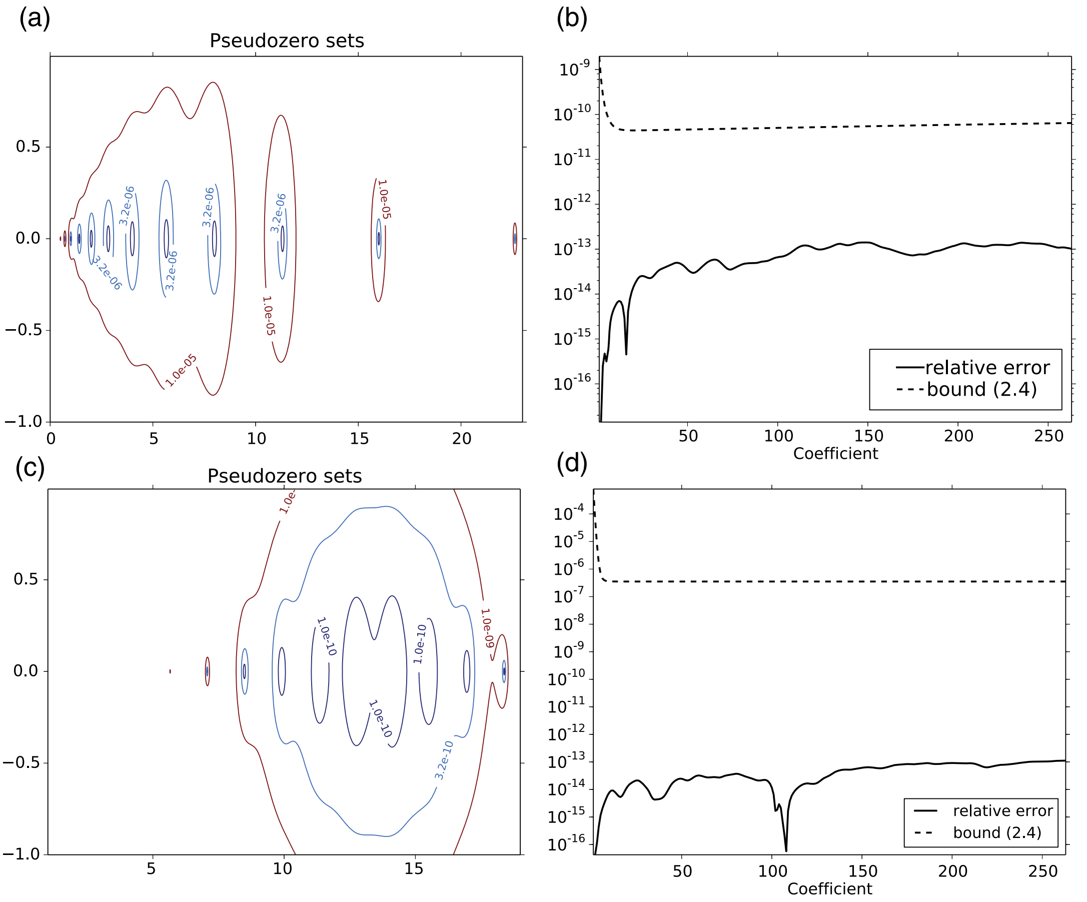

FIG. 2. Pseudozero sets and plots of relative error vs. coefficient for a well-conditioned polynomial (a and b) and an ill-conditioned one (c and d), both of degree 13. The bound (2.4) bounds absolute errors. It is converted to a bound on relative errors in the plots.

and

$$\frac{1}{\hat{a}_j^{k+1}} = \frac{1}{a_j^{k+1} + k a_j^k \Delta a_j + \mathcal{O}(\Delta a_j^2)} = \frac{1}{a_j^{k+1}} - \frac{(k+1) a_j^k \Delta a_j}{a_j^{2(k+1)}} + \mathcal{O}(\Delta a_j^2).$$

These complete the first-order perturbative calculation by implying

$$(q-\hat{q})_k = -\sum_{j=1}^{n} \frac{\mathrm{Res}(q,a_j)}{a_j^{k+1}} \left(\frac{(k+1)\Delta a_j}{a_j} + \sum_{i \neq j} \frac{\Delta a_j - \Delta a_i}{a_j - a_i} + \mathcal{O}(\Delta a^2) \right). \tag{2.3}$$

Turning to condition number of roots of $p(z) = 0$, we get the asymptotic bound

$$|(q-\hat{q})_k| \lesssim \epsilon \sum_{j=1}^{n} \left| \frac{\mathrm{Res}(q,a_j)}{a_j^{k+1}} \right| \left(\frac{(k+1)\kappa(a_j,p)}{|a_j|} + \sum_{i \neq j} \frac{\kappa(a_i,p) + \kappa(a_j,p)}{|a_j - a_i|} \right). \tag{2.4}$$

If the jth term of this bound is assigned to the jth root a_j, the term that begins to dominate as k increases is the term with $|a_j|$ smallest. That is because the parenthesized factor increases only linearly with k, while $1/a_j^{k+1}$ varies exponentially with k. Therefore, the bound suggests that the error in the kth coefficient is dominated by the root closest to 0 in the limit $k \to \infty$. In the transient phase, it suggests that the dominant contribution to the error is from either the smallness of $|a_j|$ or the smallness of $|a_j - a_i|$, owing to the proximity of two roots or the largeness of the condition number of a root.

Figure 2 compares the bound (2.4) (dashed line) to actual errors (solid line) for two examples. The first example is $\prod_{i=-3}^{9}(x - 2^{i/2})$, implying well-conditioned roots, and the second example is $\prod_{i=1}^{13}(x - i\sqrt{2})$, implying ill-conditioned roots. Both examples are based on Wilkinson (1984). In both examples, the bound (2.4) suggests transient errors at the beginning which never materialize. The bound is highly pessimistic for the ill-conditioned example.

Part of the problem with the bound (2.4) is that the condition numbers $\kappa(a_j, p)$ can overestimate the perturbation to the roots. But a more serious problem is that the errors Δa_i in the first-order error estimate (2.3) are highly correlated, and this correlation is lost when they are bounded separately using $\kappa(a_j, p)$. Since p and $\hat{p}$ are both monic polynomials, the negative sums of their roots must equal p_{n-1} and $\hat{p}_{n-1}$, respectively. Therefore, no matter how large each perturbation Δa_i may be, their sum $\sum \Delta a_i$ must be of the order of machine precision, implying correlation between the errors.

Such correlation between the errors Δa_i is lost in the asymptotic bound $|\Delta a_i| \lesssim \kappa(a_i, p)$. Whether the pseudozero plots contain information about correlations in the errors is unknown.

It is reasonable to expect a numerically stable algorithm for finding roots to reproduce elementary symmetric functions, such as the sum of all the roots or the product of all the roots, accurately. However, the Jenkins–Traub algorithm progresses from root to root, deflating the polynomial every time a root is found. Perhaps, for that reason it does not seem to have this property. If deflation uses the method of Theorem 2.2, the Jenkins–Traub algorithm will reproduce the sum of the roots with accuracy, but not the product of the roots. The Ehrlich–Aberth method (Ehrlich, 1967; Aberth, 1973) iterates simultaneously for all the roots. Whether the roots computed by the Ehrlich–Aberth method accurately reproduce all the elementary symmetric functions, with values corresponding to the coefficients of the polynomial, remains to be investigated.

3. Error bounds and numerical stability

The analysis given in this section uses techniques pioneered by Wilkinson (1961) and refined by Higham (1989, 2002). The application of the techniques is specialized to the inversion of power series. Near the end of this section, we discuss the work of Stewart (1997) when comparing the errors that are realized with the error bounds.

3.1 *Rounding error analysis*

To invert a power series as in

$$\frac{1}{1 + b_1 x + b_2 x^2 + \cdots} = 1 + c_1 x + c_2 x^2 + \cdots,$$

the coefficients c_i may be computed using

$$\begin{aligned} c_1 &= -b_1, \\ c_2 &= -b_2 - c_1 b_1, \\ &\vdots \\ c_k &= -b_k - c_1 b_{k-1} - \cdots - c_{k-1} b_1. \end{aligned} \tag{3.1}$$

The subtractions here are assumed to be left to right associative, unlike Wilkinson's analysis of triangular back-substitution (Wilkinson, 1961), which assumes the opposite. Left to right associativity has the advantage of preserving the Toeplitz structure of the matrices that arise in error bounds.

If we define C_n and T_n as

$$C_n = \begin{pmatrix} 1 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}, \quad T_{n=} \begin{pmatrix} 1 & & & & \\ b_1 & 1 & & & \\ b_2 & b_1 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ b_n & b_{n-1} & b_{n-2} & \dots & 1 \end{pmatrix}, \text{ then } T_n^{-1} = \begin{pmatrix} 1 & & & & \\ c_1 & 1 & & & \\ c_2 & c_1 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_n & c_{n-1} & c_{n-2} & \dots & 1 \end{pmatrix}. \tag{3.2}$$

Here, T_n^{-1} is, like T_n, a Toeplitz matrix. In addition, we have $T_n C_n = \mathbf{e}_1$, where $\mathbf{e}_1$ is the vector whose first component is 1 and all others are 0. In the recursion (3.1) for computing c_k, the last term $c_{k-1}b_1$ participates in only two arithmetic operations, namely the multiplication of c_{k-1} and b_1, and the subtraction of that product. Earlier terms participate in more subtractions and the second term, which is $-c_1 b_{k-1}$, participates in k subtractions. If the computed quantity is denoted by $\hat{c}_k$, we may write

$$\hat{c}_k = -b_k(1+\theta_{k+1}) - \hat{c}_1 b_{k-1}(1+\theta_k) - \cdots - \hat{c}_{k-1} b_1 (1+\theta_2).$$

In other words, if $\hat{C}_n$ is the vector made up of $\hat{c}_1, \ldots, \hat{c}_n$, we have $(T_n + \Delta T_n)\hat{C}_n = \mathbf{e}_1$ with $|\Delta T_n| \leqslant E_n$, where

$$E_n = \begin{pmatrix} 0 & & & & \\ \gamma_2|b_1| & 0 & & & \\ \gamma_3|b_2| & \gamma_2|b_1| & 0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \gamma_{n+1}|b_n| & \gamma_n|b_{n-1}| & & \dots & 0 \end{pmatrix}. \tag{3.3}$$

The identity

$$(\hat{C}_n - C_n) = -T_n^{-1}\Delta T_n(\hat{C}_n - C_n) - T_n^{-1}\Delta T_n C_n \tag{3.4}$$

is the basis of the error bounds.

We may take norms of either side of (3.4) and obtain

$$|\hat{c}_n - c_n| \leqslant ||C_n - \hat{C}_n||_\infty \leqslant \frac{|||T_n^{-1}|E_n|C_n|||_\infty}{1 - |||T_n^{-1}|E_n||_\infty}. \tag{3.5}$$

However, this bound is very poor. The coefficients of power series are typically scaled badly, with terms increasing or decreasing at a rapid rate. Norm-wise bounds are not of much use.

To get a component-wise bound, we go back to (3.4) and take absolute values of both sides.

$$\begin{aligned}|\hat{C}_n - C_n| &\leqslant |T_n^{-1}|E_n|\hat{C}_n - C_n| + |T_n^{-1}C_n|E,\\ (I - |T_n^{-1}|E_n)|\hat{C}_n - C_n| &\leqslant |T_n^{-1}C_n|E_n.\end{aligned}$$

Noting that the matrix $(I - |T_n^{-1}|E_n)$ is lower triangular with a non-negative inverse, we have the following theorem.

THEOREM 3.1 If a power series is inverted using the recurrence (3.1) and left to right associativity, we have the error bound

$$|\hat{C}_n - C_n| \leqslant (I - |T_n^{-1}|E_n)^{-1}|T_n^{-1}C_n|E_n. \tag{3.6}$$

3.2 *Condition analysis and numerical stability*

If p is a power series, $|p|$ denotes the power series with coefficients replaced by their absolute values. Let p and q be power series with constant terms equal to 1 and

$$pq = 1.$$

If p is perturbed to $p + \Delta p$, where the constant term of Δp is 0, suppose that q gets perturbed to $q + \Delta q$. We have

$$(p + \Delta p)(q + \Delta q) = 1.$$

It follows that

$$\begin{aligned}p\Delta q &= -q\Delta p - \Delta p\Delta q,\\ \Delta q &= -q^2\Delta p - q\Delta p\Delta q,\\ |\Delta q| &\leqslant |q^2\Delta p| + |q\Delta p||\Delta q|,\\ (1 - |q\Delta p|)|\Delta q| &\leqslant |q^2\Delta p|.\end{aligned}$$

All the coefficients of the power series $1/(1 - |q\Delta p|)$ are positive. Therefore, we may multiply by that power series to get the bound

$$|\Delta q| \leqslant \frac{|q^2\Delta p|}{1 - |q\Delta p|} \leqslant \frac{|q^2||\Delta p|}{1 - |q||\Delta p|}. \tag{3.7}$$

We may take $|\Delta p|$ to be

$$\sum_{j=1}^{\infty} u|p_j|x^j, \tag{3.8}$$

where u is the unit round-off, to obtain a bound on each entry of q using (3.7). Here, it is significant that the constant term of Δp is zero. The conditioning bound (3.7), with $|\Delta p|$ given by (3.8), is sharp up to first-order for each coefficient of Δq with a suitable choice of the signs of the coefficients of $|\Delta p|$.

Armed with this conditioning bound, we may consider the numerical stability of the inversion of power series using the recurrence (3.1). Theorem 3.1 states that

$$|\hat{C}_n - C_n| \leqslant (I - |T_n^{-1}|E_n)^{-1}|T_n^{-1}C_n|E_n.$$

From the definitions of C_n and T_n^{-1} in (3.2) as well as that of E_n in (3.3), we obtain

$$|C_n - \hat{C}_n| \leqslant \frac{2(n+1)|q^2||\Delta p|}{1 - 2(n+1)|q||\Delta p|}.$$

Here, we have used $\gamma_k < \gamma_{n+1}$ for $k \leqslant n$ and $\gamma_{n+1} \leqslant 2(n+1)u$, which assumes $(n+1)u < 1/2$. This bound differs from the conditioning bound (3.7) for each coefficient by only a polynomial factor in n. Therefore, inversion of power series using back-substitution is numerically stable.

3.3 *Discussion of Stewart's lower bound for the least singular value*

Figure 3 shows that the bounds of Sections 3.1 and 3.2 do quite well on four different examples. The bounds themselves were computed using extended precision of 100 digits. The actual relative error was computed by comparing the double precision answers with extended precision answers. For inversion of cosine, $1/\cos x$ in Fig. 3(b), the odd terms were ignored. It may be noted that the inverse cosine series is one of the ways of defining Euler numbers. In the 'randn' series, each b_i in $p(x) = 1 + \sum_{i=1}^{\infty} b_i x^i$ is an independent standard normal variable.

Error bounds for inversion of triangular matrices are similar to that of Theorem 3.1. However, they often overestimate the error greatly (Wilkinson, 1961). In particular, for many triangular matrices the relative error in the inverse appears independent of the condition number. Here, we discuss the work of Stewart (1997) and connect it to the inversion of power series.

Consider the upper triangular matrix

$$\begin{pmatrix} R & r \\ 0 & \delta \end{pmatrix}. \tag{3.9}$$

If σ is its smallest singular value, suppose $\sigma \geqslant \beta\delta$, where $\beta \in [0, 1]$ must hold. If β is not too tiny, the matrix is rank-revealing in the sense of Stewart. The last row of this matrix may be rescaled to obtain

$$\begin{pmatrix} R & r \\ 0 & 1 \end{pmatrix},$$

whose least singular value is denoted by $\hat{\rho}$. If the least singular value of R is ρ, Stewart (1997) has proved that

$$\hat{\rho} \geqslant \frac{\beta\rho}{\sqrt{\beta^2 + \rho^2}}.$$

This bound may be interpreted as follows. If the matrix (3.9) is rank-revealing with a β that is not too tiny, any significant fall in the least singular value when we move from R to that matrix must be due to the smallness of δ. The smallness of δ can be easily eliminated by rescaling the last row to get a matrix whose condition number $\hat{\rho}$ is only moderately smaller than ρ, the condition number of R. On the other hand, if the best possible β is quite tiny, it may mean that the ill-conditioning of the matrix (3.9) is hidden within the correlations between rows in a way that may not be eliminated so easily. If each one of the principal submatrices of a matrix is rank-revealing, any ill-conditioning is almost entirely

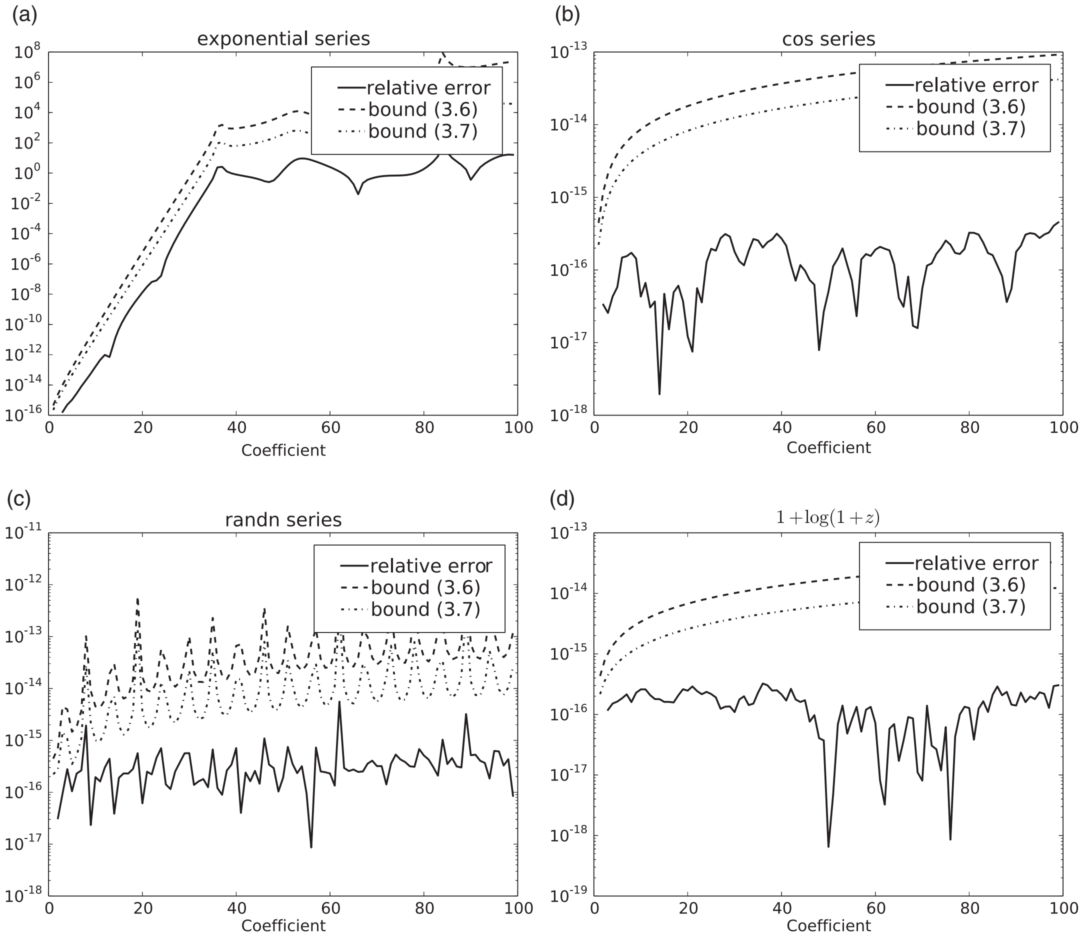

FIG. 3. Rounding error bounds and actual rounding errors for four examples. The bound of Theorem 3.1 is on absolute error. That bound is converted to a bound on relative error and labelled as 'bound (3.6)' in each of the plots. Likewise, the bound of (3.7), with $|\Delta p|$ given by (3.8), is converted to a bound on relative error and labelled as 'bound (3.7)' in each of the plots. Each plot graphs relative error in the nth coefficient vs. n.

removed by rescaling rows explaining Wilkinson's observation that rounding errors that arise during back-substitution are far smaller than bounds implied by the condition number of the triangular matrix.

Many triangular matrices are not rank-revealing. For example, random triangular matrices are not rank-revealing with probability 1 as proved in Viswanath & Trefethen (1998). However, Stewart (1997) argues intuitively that the triangular matrices that arise in Gaussian elimination and QR factorization with pivoting are likely to be rank-revealing. His argument is that if a matrix is rank deficient, Gaussian elimination and QR will break down with a 0 on the diagonal. If it is nearly rank deficient, continuity suggests that a very small entry must appear on the diagonal indicating its rank deficiency and especially so if pivoting is employed. Thus, the rank-revealing property of pivoted Gaussian elimination and QR factorization, if rigorously established, would explain why the triangular matrices that arise during pivoted Gaussian elimination and QR factorization do not have tiny βs.

To connect Stewart's analysis to power series, we shall assume that $p(x) = 1 + \sum b_i x^i$ has a radius of convergence R equal to 1. Any finite radius of convergence can be turned into 1 by the change of variables $x \leftarrow x/R$. Assuming $R = 1$, the matrix T_n of (3.2) is rank-revealing if and only if its least singular value is $\mathcal{O}(1)$. The least singular value of T_n is $\mathcal{O}(1)$ if and only if the greatest singular value of $T_n^{-1}$is $\mathcal{O}(1)$, which is true if and only if the entries c_i of T_n^{-1} in (3.2) are $\mathcal{O}(1)$. Since c_i are the coefficients of the power series of $1/p(x)$, we have that T_n is rank-revealing in the sense of Stewart if and only if the radius of convergence of $1/p(x)$ is 1 or greater.

If the equation $p(z) = 0$ has a solution with $|z| < 1$ in the complex plane, the matrix T_n will not be rank-revealing. The example of Fig. 3(d), $p(z) = 1 + \log(1 + z)$ has a zero at $z = 1 - 1/e$ and the corresponding matrix T_n is not rank-revealing. If in fact the radius of converge of $p(z)$ is 1 and there is no zero with $|z| < 1$, the matrix T_n will be rank-revealing, but its condition number will be $\mathcal{O}(1)$. Within the scope of the analysis given by Stewart, the situation where the actual relative errors are much smaller than the conditioning bound appears unlikely. The good agreement between the bounds and the actual errors in Fig. 3 is the rule rather than the exception.

4. Conclusions

In this article, we have considered the inversion of power series with particular attention to the special case of inverting polynomials. Essential background is provided by the classic work of Wilkinson (1961) on inversion of triangular systems.

We found and explicated a subtle numerical instability that arises when factors corresponding to known roots are deflated from polynomials. This instability has occurred in the computation of spectral differentiation matrices. The suggestion that polynomial root finding algorithms such as Jenkins–Traub may be more accurate without the deflation step merits further investigation.

The rounding error analysis and the condition analysis of power series inversion imply numerical stability. In addition, the error bounds that result from the analysis are not unduly pessimistic, as happens for certain other triangular systems.

Acknowledgement

Thanks to both the referees for valuable suggestions and comments. Thanks to Jervis Muindi for sharing his Python implementation of Jenkins–Traub on GitHub and fixing a bug at a moment's notice.

Funding

This research was partially supported by NSF grants DMS-1115277 and SCREMS-1026317.

References

Aberth, O. (1973) Iteration methods for finding all zeros of a polynomial simultaneously. *Math. Comp.*, **27**, 339–344.

Butcher, J. C., Corless, R. M., Gonzalez-Vega, L. & Shakoori, A. (2011) Polynomial algebra for Birkhoff interpolants. *Numer. Algorithms*, **56**, 319–347.

Ehrlich, L. W. (1967) A modified Newton method for polynomials. *Commun. ACM*, **10**, 107–108.

Gautschi, W. (1984) Questions of numerical condition related to polynomials. *Studies in Numerical Analysis* (G. H. Golub ed.). MAA Studies in Mathematics, vol. 24. MAA, pp. 140–177.

Higham, N. J. (1989) The accuracy of solutions to triangular systems. *SIAM J. Numer. Anal.*, **26**, 1252–1265.

Higham, N. J. (2002) *Accuracy and Stability of Numerical Algorithms*, 2nd edn. Philadelphia: SIAM.

Hoggatt Jr., V. E. & Bicknell, M. (1973) Roots of Fibonacci polynomials. *Fibonacci Quart.*, **11**, 271–274.

Jenkins, M. A. & Traub, J. F. (1970) A three stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration. *Numer. Math.*, **14**, 252–263.

Mosier, R. G. (1986) Root neighborhoods of a polynomial. *Math. Comput.*, **47**, 265–273.

Sadiq, B. & Viswanath, D. (2013) Barycentric Hermite interpolation. *SIAM J. Sci. Comput.*, **35**, A1254–A1270.

Sadiq, B. & Viswanath, D. (2014) Finite-difference weights, spectral differentiation, and superconvergence. *Math. Comput.*, **83**, 2403–2427.

Stewart, G. W. (1997) The triangular matrices of Gaussian elimination and related decomposition. *IMA J. Numer. Anal.*, **17**, 7–16.

Toh, K.-C. & Trefethen, L. N. (1994) Pseudozeros of polynomials and pseudospectra of companion matrices. *Numer. Math.*, **68**, 403–425.

Viswanath, D. & Trefethen, L. N. (1998) Condition numbers of random triangular matrices. *SIAM J. Matrix Anal. Appl.*, **19**, 564–581.

Weideman, J. A. C. & Reddy, S. C. (2000) A MATLAB differentiation matrix suite. *ACM Trans. Math. Softw.*, **26**, 465–519.

Welfert, B. D. (1997) Generation of pseudospectral differentiation matrices I. *SIAM J. Numer. Anal.*, **34**, 1640–1657.

Wilkinson, J. H. (1961) Error analysis of direct methods of matrix inversion. *J. Assoc. Comput. Mach.*, **8**, 281–330.

Wilkinson, J. H. (1984) The perfidious polynomial. *Studies in Numerical Analysis* (G. H. Golub ed.). MAA Studies in Mathematics, vol. 24. MAA, pp. 1–28.