Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 205 (2007) 281 – 295
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Numerical experiments on the accuracy of the Chebyshev–Frobenius companion matrix method for finding the zeros of a truncated series of Chebyshev polynomials ☆

John P. Boyd*, Daniel H. Gally

Department of Atmospheric, Oceanic and Space Science, University of Michigan, 2455 Hayward Avenue, Ann Arbor, MI 48109, USA

Received 12 May 2005; received in revised form 6 May 2006

Abstract

For a function $f(x)$ that is smooth on the interval $x \in [a, b]$ but otherwise arbitrary, the real-valued roots on the interval can always be found by the following two-part procedure. First, expand $f(x)$ as a Chebyshev polynomial series on the interval and truncate for sufficiently large N. Second, find the zeros of the truncated Chebyshev series. The roots of an arbitrary polynomial of degree N, when written in the form of a truncated Chebyshev series, are the eigenvalues of an $N \times N$ matrix whose elements are simple, explicit functions of the coefficients of the Chebyshev series. This matrix is a generalization of the Frobenius companion matrix. We show by experimenting with random polynomials, Wilkinson's notoriously ill-conditioned polynomial, and polynomials with high-order roots that the Chebyshev companion matrix method is remarkably accurate for finding zeros on the target interval, yielding roots close to full machine precision. We also show that it is easy and cheap to apply Newton's iteration directly to the Chebyshev series so as to refine the roots to full machine precision, using the companion matrix eigenvalues as the starting point. Lastly, we derive a couple of theorems. The first shows that simple roots are stable under small perturbations of magnitude ε to a Chebyshev coefficient: the shift in the root x_* is bounded by $\varepsilon/\mathrm{d}f/\mathrm{d}x(x_*) + \mathrm{O}(\varepsilon^2)$ for sufficiently small ε. Second, we show that polynomials with definite parity (only even or only odd powers of x) can be solved by a companion matrix whose size is one less than the number of nonzero coefficients, a vast cost-saving.

MSC: 65H05; 42C10; 65D05

Keywords: Kepler's equation; Chebyshev polynomials; Rootfinding; Chebyshev-Frobenius companion matrix

1. Introduction

A polynomial is usually written in the "power form", also known as the "monomial form",

$$f_N(x) = \sum_{j=0}^{N} b_j x^j. \tag{1}$$

☆ This work was supported by the National Science Foundation through grant OCE9986368.
* Corresponding author. Tel.: +1 734 7643338.
E-mail addresses: jpboyd@engin.umich.edu, j.p.boyd@comcast.net (J.P. Boyd), dgally@umich.edu (D.H. Gally).
URL: http://www-personal.engin.umich.edu:/~jpboyd/.

0377-0427/$ - see front matter
doi:10.1016/j.cam.2006.05.006

However, the power form is dreadfully ill-conditioned. Least-squares library software therefore computes polynomial fits as sums of Legendre polynomials, which are close cousins of the Chebyshev polynomials employed here [4]. Any polynomial can alternatively be written in the "Chebyshev form", a truncated series of Chebyshev polynomials:

$$f_N(x) = \sum_{j=0}^{N} a_j T_j(x). \tag{2}$$

On its "canonical interval", $x \in [-1, 1]$, the Chebyshev form is extremely stable and well-conditioned. Nonetheless, library software for polynomial rootfinding invariably expects as input the coefficients b_j of the power form.

The lack of software to find roots of the Chebyshev form is particularly unfortunate because many applications—perhaps most—naturally generate polynomials in this form.

Boyd has shown [5,7] that to find the real zeros of a transcendental function $f(x)$ on an interval, f can be replaced by its Chebyshev interpolant f_N, whose roots are then very good approximations to those of $f(x)$. This Chebyshev-proxy strategy may be an order of magnitude cheaper than finding the roots of $f(x)$ directly if f is expensive to evaluate, such as the determinant of a large matrix.

Battles and Trefethen [2] have extended Matlab from vectors and matrices to functions and operators by replacing functions by their approximations as Chebyshev interpolants—i.e., polynomials in Chebyshev form where the degree N may be as high as the thousands. This is "Chebyshevization" on a grand scale, and rootfinding-in-Chebyshev-form is an essential component [2].

The appendix describes the simple procedures for robust adaptive Chebyshev interpolation.

Chebyshev spectral methods solve differential equations by approximating the unknown $u(x)$ as a polynomial of some large degree N in Chebyshev form [6]. The book review [8] catalogues eighteen books on Chebyshev spectral algorithms. To find the roots of such a spectrally computed $u(x)$, one must find the zeros of a polynomial in Chebyshev form. The maxima and minima of $u(x)$ are the roots of its derivative $\mathrm{d}u/\mathrm{d}x$. In the appendix, we give a recurrence for computing the coefficients of the derivative from those of $u(x)$ itself. Thus, computing extrema is also an exercise in Chebyshev root-finding.

It is therefore useful to devise algorithms that can find the real roots of a polynomial in Chebyshev form on the canonical interval $x \in [-1, 1]$. (Outside this interval, including complex x, the Chebyshev polynomials are ill-conditioned and the accuracy of Chebyshev approximation degenerates very rapidly with distance from the canonical interval, so it is reasonable to restrict attention to those roots that lie on the real interval where the Chebyshev polynomials are well-behaved; for Chebyshev spectral methods, this "canonical" interval is also the whole of the physical domain.)

A variety of different methods for "Chebyshev rootfinding" are described in [5,7,9,10]. However, a very useful and general method is the companion matrix method. For general orthogonal polynomials, the companion matrix was first discovered by Specht [15,16] and independently rediscovered several times since [1,14,13,17]. The roots of the truncated Chebyshev series are the eigenvalues of a matrix whose elements are simple, explicit functions of the Chebyshev coefficients a_j.

When the eigenvalues are found by the QR algorithm [19], the cost is about $10N^3$. This is rather expensive for very large N as in some cases of [2], which has inspired the search for faster algorithms for large N. For $N \leqslant 100$, however, the cost is a hundredth of a second or less on a fast personal computer.

An important issue is: How stable and accurate is the companion matrix algorithm? The experiments in [13] are encouraging; our goal is to provide a much more comprehensive set of tests.

In the next section, we briefly describe the companion matrix algorithm. For expository simplicity, we will assume that the interval of interest (and of accurate Chebyshev approximation) is the canonical interval $x \in [-1, 1]$. It is trivial to generalize this to an arbitrary interval $x \in [a, b]$ merely by making a linear change-of-coordinates,

$$y = \frac{2x - (b + a)}{b - a} \tag{3}$$

and performing all Chebyshev calculations in the stretched variable $y \in [-1, 1]$. The formulas of the appendix allow for arbitrary a, b.

Before discussing our numerical experiments on the companion matrix method, we offer some new theorems in Sections 3 and 5 that help to explain why the method is so robust, and how to apply it more cheaply to special cases. The rate of convergence of Chebyshev series is discussed in Section 4 and illustrated with a numerical example.

2. Companion matrix methods

More than a century ago, Georg Frobenius showed that the roots of a polynomial in monomial form,

$$f_N(x) = \sum_{j=0}^{N} b_j x^j \tag{4}$$

are also the eigenvalues of the matrix which is now called the "Frobenius companion matrix" of the polynomial. For $N = 5$, the matrix is

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ (-1)\dfrac{b_0}{b_5} & (-1)\dfrac{b_1}{b_5} & (-1)\dfrac{b_2}{b_5} & (-1)\dfrac{b_3}{b_5} & (-1)\dfrac{b_4}{b_5} \end{vmatrix} \tag{5}$$

with an obvious generalization to arbitrary N.

Boyd [5,7] shows that the Chebyshev coefficients $\{a_j\}$ can be converted to the power coefficients $\{b_j\}$ by a vector–matrix multiplication where the elements of the conversion matrix can be computed by a simple recurrence. However, the condition number of the conversion matrices grows as roughly 2.4^N so that the "convert-to-powers" method is safe only when $N < 17$ (or so). The Chebyshev companion matrix is therefore a valuable alternative.

For the $N = 5$ case, the Chebyshev companion matrix is

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 0 \\ (1/2) & 0 & (1/2) & 0 & 0 \\ 0 & (1/2) & 0 & (1/2) & 0 \\ 0 & 0 & (1/2) & 0 & (1/2) \\ (-1)\dfrac{a_0}{2a_5} & (-1)\dfrac{a_1}{2a_5} & (-1)\dfrac{a_2}{2a_5} & (-1)\dfrac{a_3}{2a_5} + (1/2) & (-1)\dfrac{a_4}{2a_5} \end{vmatrix}. \tag{6}$$

For general N, the elements of the Chebyshev–Frobenius companion matrix are, with δ_{jk} the usual Kronecker delta-function such that $\delta_{jj} = 1$ while $\delta_{jk} = 0$ if $j \neq k$,

$$A_{jk} = \begin{cases} \delta_{2,k}, & j = 1, \quad k = 1, 2, \ldots, N, \\ \frac{1}{2}\{\delta_{j,k+1} + \delta_{j,k-1}\}, & j = 2, \ldots, (N-1), \quad k = 1, 2, \ldots, N, \\ (-1)\frac{a_{j-1}}{2a_N} + \frac{1}{2}\,\delta_{k,N-1}, & j = N, \quad k = 1, 2, \ldots, N, \end{cases} \tag{7}$$

Computing the eigenvalues of the companion matrix will return all N roots of the polynomial. However, Chebyshev series are accurate approximations only on the canonical interval $x \in [-1, 1]$. As explained in [6], the error $f(x) - f_N(x)$ grows *exponentially* fast as x moves from this interval for either real or complex x. Consequently, the only roots of interest are those that either lie on the canonical interval or are extremely close.

It follows that the computation of the eigenvalues of the companion matrix is only a first step. The second step is to accept only those roots that satisfy the inequalities

$$|\Re(x)| \leqslant 1 + \tau, \quad |\Im(x)| \leqslant \tau, \tag{8}$$

where $\tau > 0$ is a user-specified "interval tolerance". Often it is sufficient to set $\tau = 0$ and merely test for real-valued eigenvalues on $x \in [-1, 1]$. However, roots at the endpoints or multiple roots can easily be perturbed off the canonical interval by the numerical errors which are inevitable in any floating point computation. A tiny but nonzero τ ensures that no real-valued roots on the interval are missed.

The optional third step is to refine the Chebyshev eigenvalues by one or two iterations of Newton's method:

$$x^{(n+1)} = x^{(n)} - \frac{f_N(x^{(n)})}{\mathrm{d}f_N/\mathrm{d}x(x^{(n)})}. \tag{9}$$

It is easy to compute the coefficients of $\mathrm{d}f_N/\mathrm{d}x$ from those of f_N by means of the recurrence relation (45). A Chebyshev series can be summed in $\mathrm{O}(N)$ operations by using the Clenshaw–Horner recurrence (43).

3. Chebyshev coefficient perturbation theorem

Before describing the numerical experiments, it is desirable to note that the Chebyshev form of a polynomial is very stable to perturbations (unlike the power form), at least for simple roots.

Theorem 1 (*Chebyshev perturbation theorem*). *Let x_* denote a root of $f_N(x)$. Suppose that $f_N(x)$ is perturbed by an amount ε in the coefficient of $T_j(x)$:*

$$g(x) \equiv f_N(x) + \varepsilon T_j(x). \tag{10}$$

Let $x_g(\varepsilon)$ denote the root of $g(x;\varepsilon)$ which is nearest x_*. Then

1.

$$\lim_{\varepsilon\to 0} \frac{x_g(\varepsilon) - x_*}{\varepsilon} = -\frac{T_j(x_*)}{\mathrm{d}f_N/\mathrm{d}x(x_*)}. \tag{11}$$

This statement is true even for roots located off the canonical interval.

2. *If x_* is on the canonical interval, $x_* \in [-1, 1]$, then*

$$|x_g(\varepsilon) - x_*| = \leqslant |\varepsilon| \frac{1}{|\mathrm{d}f_N/\mathrm{d}x(x_*)|} + \mathrm{O}(\varepsilon^2) \tag{12}$$

for sufficiently small ε.

Proof. The first proposition follows from applying Newton's iteration to $g(x)$ from the initial iterate $x = x_*$, and exploiting the fact that $f_N(x_*) = 0$ by definition. The second proposition follows from the well-known Chebyshev bound [6, p. 47], $|T_j(x)| \leqslant 1$ for all $x \in [-1, 1]$. □

4. Chebyshev convergence theory with a numerical example of rootfinding by Chebyshev-series proxy

4.1. The ellipse of convergence

If $f(x)$ is an analytic function on $x \in [-1, 1]$, then a well-known theorem [6, p. 48, 18] asserts that its Chebyshev series will converge within the largest ellipse with foci at $x = \pm 1$ which is free of singularities. Any of these confocal ellipses can be parameterized in the form, with $\eta \in [0, 2\pi]$,

$$\Re(x) = \cosh(\mu)\,\cos(\eta), \quad \Im(x) = \sinh(\mu)\,\sin(\eta), \tag{13}$$

where μ is constant on the ellipse and the complete curve is traced as η varies from 0 to 2π. The convergence-limiting singularity or singularities are those whose location (x_s, y_s) in the complex-plane give the smallest positive values, among all singularity locations, for μ where

$$\mu = \log\left(\alpha + \sqrt{\alpha^2 - 1}\right), \quad \alpha = \frac{1}{2}\left\{\sqrt{(x_s+1)^2 + y_s^2} + \sqrt{(x_s-1)^2 + y_s^2}\right\}. \tag{14}$$

In other words, among all ellipses that have singularities on them, the convergence-limiting ellipse is the one which is closest to the real interval $x \in [-1, 1]$. The Chebyshev series will converge at a geometric rate with coefficients a_n

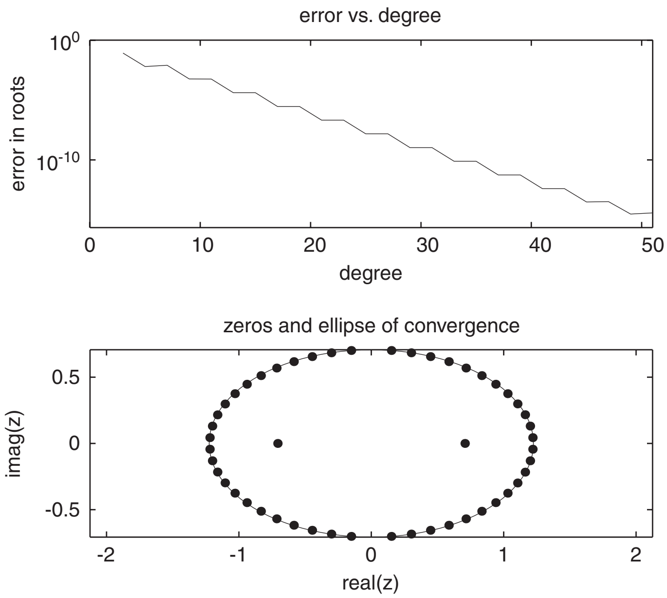

Fig. 1. Top: the errors in approximating the roots of $f(x) = (1 - 2x^2)/(1 + 2x^2)$ versus the degree N of the truncated Chebyshev series which is used to approximate $f(x)$. The bottom axis is at machine epsilon; increasing the degree further produces no improvement because of roundoff error. (The errors in each of the two roots is the same because these roots are identical except for sign.) Bottom: the 51 zeros of the approximating polynomial of degree 51. The solid curve is the ellipse of convergence.

proportional to $\exp(-n\mu)$ or equivalently, $(\exp(-\mu))^n$, where μ depends on the location of the singularities nearest the interval as in (14). For the special case of an entire function, $\mu \to \infty$ and the series decreases "supergeometrically" [6].

The only weakness of a Chebyshev expansion is that a function $f(x)$ might have singularities only a distance ε from the nearest point of $x \in [-1, 1]$ where $\varepsilon \ll 1$. In that case, the exponential dependence of the coefficients on n might be as slow as $\exp(-n\varepsilon)$ (if the singularities are on the imaginary axis, somewhat faster if $\Re(x) \neq 0$). For a "nearly singular" function in the sense that a pole or branch point is almost on the expansion interval, then many terms in the Chebyshev series will be necessary to obtain a good approximation and the Chebyshev-proxy rootfinder will be relatively costly.

The accuracy of a Chebyshev series degrades exponentially fast off the canonical interval. If we introduce elliptical coordinates in the complex x-plane through (13), it is known that on the ellipse with quasi-radial coordinate μ, the maximum of the term $a_n T_n(x)$ is larger than its maximum on $x \in [-1, 1]$ by $\exp(n\mu)$. Consequently, although the Chebyshev-proxy method is theoretically capable of finding *complex-valued* roots if they lie *within* the *ellipse of convergence* of the Chebyshev series, the accuracy will be much poorer than for the real roots. Consequently, the prudent and practical viewpoint is to accept only roots of the Chebyshev series on the real, canonical interval $x \in [-1, 1]$ and discard all complex-valued roots, or real roots with $|x| > 1$, as untrustworthy.

4.2. Numerical example

The function

$$f(x) \equiv \frac{1 - 2x^2}{1 + 2x^2}, \quad x \in [-1, 1] \tag{15}$$

is typical in that it is free of singularities on the canonical interval for Chebyshev series, $x \in -[1, 1]$, but has singularities at a not-too-small distance off the axis. For this example, the singularities are simple poles at $z = \pm i/\sqrt{2}$ and the coefficients decrease proportionally to $(0.517)^n$. Fig. 1 (upper panel) shows that because the series is converging exponentially fast to $f(x)$, the error in approximating the zeros of $f(x)$ by those of its Chebyshev series proxy are falling exponentially fast, too.

A. Hurwitz proved a theorem [12, p. 148] that if a sequence of functions (not restricted to a truncated series of Chebyshev polynomials) converges to a function $f(x)$ within some domain in the complex-plane, then the zeros of the sequence converge to the zeros of $f(x)$ within the domain while the members of the sequence may have additional zeros that cluster at the boundaries of the convergence region. The lower panel is a numerical confirmation: the truncated Chebyshev series has two real zeros that closely approximate (to within about 15 decimal places!) those of $f(x)$; it also has 49 zeros unconnected with those of $f(x)$ except that they cluster around the ellipse of convergence.

As noted in the previous section, no special rule is needed to reject these "Hurwitz zeros" as spurious approximations to the roots of $f(x)$. As noted in the introduction and again in the previous subsection, the most prudent policy is to reject all roots of the Chebyshev series that do not lie on the real interval $x \in [-1, 1]$ even though *some* complex roots—those well within the ellipse of convergence and not near it like the "Hurwitz zeros"—might be at least crude approximations to zeros of $f(x)$.

5. Chebyshev series with parity

In applications, one often encounters Chebyshev series of even degree polynomials only, which are symmetric with respect to $x = 0$, or series of odd terms only, which are antisymmetric. These special cases can always be done as general Chebyshev series, but this is wasteful; it would clearly be desirable to devise a companion matrix whose size is equal to the number of nonzero terms as in the following.

Theorem 2 (*Companion matrix for sum of even degree polynomials*). *Define*

$$S(x) \equiv \sum_{j=0}^{N} a_j T_{2j}(x) \tag{16}$$

and the unsymmetric polynomial of half the degree and the same coefficients,

$$\sigma(y) \equiv \sum_{j=0}^{N} a_j T_j(y). \tag{17}$$

Then the roots of $S(x)$ are given in terms of the roots y_j of $\sigma(y)$ by

$$x_j = \pm \cos\left(\tfrac{1}{2}\arccos(y_j)\right), \quad j = 1, 2, \ldots, N. \tag{18}$$

(*Note that symmetric polynomial S has twice as many roots as $\sigma(y)$.*) *This formula is valid both for roots on the interval $x \in [-1, 1]$ and for roots off this interval. It implies that one can find the roots of a truncated Chebyshev series with symmetry* (*i.e.*, $S(x) = S(-x) \forall x$) *by computing the eigenvalues of the Chebyshev Frobenius matrix of the associated polynomial σ and then applying the transformation* (18).

Proof. Because of the identity $T_j(\cos(t)) = \cos(jt)$ for all j, t,

$$S(\cos(t)) = \sum_{j=0}^{N} a_j \cos(2jt). \tag{19}$$

If we make the change-of-variable $s = 2t$, the series becomes

$$S(\cos(s/2)) = \sum_{j=0}^{N} a_j \cos(js). \tag{20}$$

By making the reverse change-of-variable $s = \arccos(y)$

$$S(\cos\{\arccos(y)/2\}) = \sum_{j=0}^{N} a_j T_j(y). \tag{21}$$

This is in the form of a truncated Chebyshev series without symmetry. It follows that by applying (7) and computing the eigenvalues y_j, the roots of the polynomial with parity are related to the eigenvalues of the matrix by unscrambling the changes-of-coordinate as in (18). □

An odd polynomial in Chebyshev form can always be converted into one of even parity (and lower degree) by applying the following lemma.

Lemma 1 (*Chebyshev division-by-x*). *Let*

$$A(x) = \sum_{j=0}^{N} a_j T_{2j+1}(x) \tag{22}$$

and

$$S(x) = \frac{A(x)}{x} = \sum_{j=0}^{N} b_j T_{2j}(x). \tag{23}$$

Then the coefficients of the symmetric polynomial $S(x)$ can be computed by

$$b_N = 2a_N \tag{24}$$

followed by the recurrence

$$b_j = 2a_j - b_{j+1}, \quad j = 1, \ldots, N-1 \tag{25}$$

and

$$b_0 = a_0 - (1/2)\, b_1. \tag{26}$$

Proof. Repeated use of the Chebyshev recursion relation in the form

$$xT_n(x) = \tfrac{1}{2}\{T_{n+1}(x) + T_{n-1}(x)\}. \qquad \square \tag{27}$$

Theorem 3 (*Companion matrix for sum of odd degree polynomials*). *Define*

$$A(x) \equiv \sum_{j=0}^{N} a_j T_{2j+1}(x). \tag{28}$$

Apply the recursion of the lemma to obtain

$$S(x) = \frac{A(x)}{x} = \sum_{j=0}^{N} b_j T_{2j}(x). \tag{29}$$

Define, exactly as in the previous theorem, the unsymmetric polynomial of half the degree and the same coefficients,

$$\sigma(y) \equiv \sum_{j=0}^{N} b_j T_j(y). \tag{30}$$

Then the roots of $S(x)$ are given in terms of the roots y_j of $\sigma(y)$ by

$$x = 0, \pm\cos\left(\tfrac{1}{2}\arccos(y_j)\right), \quad j = 1, 2, \ldots, N. \tag{31}$$

This formula is valid both for roots on the interval $x \in [-1, 1]$ and for roots off this interval.

Proof. Follows trivially from the lemma and previous theorem. □

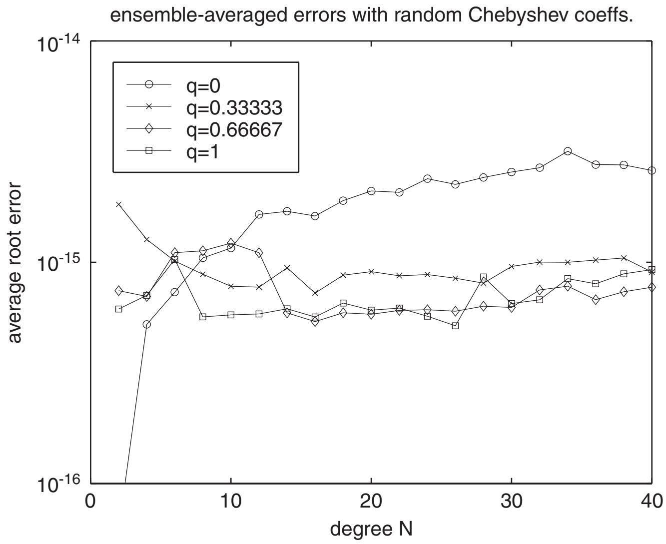

Fig. 2. Ensemble-averaged maximum error in the roots for four different decay rates: $q = 0$ (no decay) (circles), $q = \frac{1}{3}$ (x's), $q = \frac{2}{3}$ (diamonds) and $q = 1$ (squares). Each ensemble included 100 polynomials.

6. Polynomials with random coefficients

For a first set of experiments, we created an ensemble of polynomials with random coefficients chosen from the uniform distribution on $[-1, 1]$ and then averaged the error within each ensemble. The size of the ensembles was increased until the average error had converged.

Chebyshev series typically converge like a geometric series [6], so we also experimented with multiplying the random coefficients by a geometrically decreasing factor, $\exp(-qj)$ where $q \geqslant 0$ is a constant and j is the degree of the Chebyshev polynomial multiplying this factor.

Fig. 2 shows that the companion matrix algorithm is remarkably accurate. Independent of the decay rate and also of the degree of the polynomial, the maximum error in any of the roots is on average only an order of magnitude greater than machine epsilon, 2.2×10^{-16}!

These errors are so uniformly tiny that there is little point in further experiments on random polynomials. Even for N as high as 200 (not shown), errors above 10^{-14} were not observed. However, the polynomials that arise in applications are not random. We next turn to special classes of polynomials that are known to be troublesome.

7. Wilkinson polynomial

James Wilkinson showed more than 40 years ago that a polynomial with a large number of evenly spaced real roots was spectacularly ill-conditioned. Bender and Orszag [3] give a very good discussion with graphs. They note that when $N = 20$, perturbing a coefficient in the *power* form by 10^{-9} makes six roots complex-valued with imaginary parts as large as much as 4% of their unperturbed value.

For our purposes, it is convenient to shift and rescale Wilkinson's example so that the Wilkinson polynomial of degree N has its roots evenly spaced on the canonical interval $x \in [-1, 1]$:

$$W(x; N) \equiv \prod_{j=1}^{N} \left(x - \frac{2j - N - 1}{N - 1} \right). \tag{32}$$

One difficulty with the *power* form is that even for N as small as 20, the power coefficients b_j vary by nearly a factor of a billion:

$$W(x;20) = 0.11\times 10^{-7} - 0.50\times 10^{-5}x^2 + 0.33\times 10^{-3}x^4 - 0.80\times 10^{-2}x^6 + 0.092x^8 - 0.58x^{10} + 2.09x^{12} - 4.48x^{14} + 5.57x^{16} - 3.68x^{18} + x^{20}. \tag{33}$$

In contrast, the Chebyshev coefficients exhibit a much smaller range and the Chebyshev polynomials oscillate uniformly on the interval $x \in [-1, 1]$ whereas the powers of x are very nonuniform on the same range:

$$W(x;20) = \frac{1}{0.000049}\{-1 - 0.18T_2(x) - 0.12T_4(x) - 0.036T_6(x) + 0.045T_8(x) + 0.10T_{10}(x) + 0.12T_{12}(x) + 0.093T_{14}(x) + 0.054T_{16}(x) + 0.021T_{18}(x) + 0.0039T_{20}(x)\}. \tag{34}$$

Theorem 1 shows that for a function like the Wilkinson polynomial, which has only well-separated real roots, the roots are insensitive to slight perturbations in the Chebyshev coefficients. Thus, the problems of (i) a large range in the *coefficients* and (ii) great sensitivity to perturbations in the coefficients are removed by shifting to the Chebyshev form.

The second difficulty is that the function $W(x)$ (not just its coefficients) has a huge "dynamic range", that is, the maxima and minima between the roots vary by many orders of magnitude. This difficulty is displayed by the asymptotic form [9] (for even N)

$$W \sim \frac{\sin(\pi(N/2)x)}{\pi}\exp\left\{\frac{Nx^2}{2} + \frac{N}{12}x^4 + \cdots\right\}, \quad N\to\infty, \quad |x| \ll N/2 \tag{35}$$

and is also shown in Fig. 3. The absolute errors in a Chebyshev expansion are uniformly small, on the order of 10^{15}; in Fig. 3, these errors are horizontal lines very near the lower axes in both of the two upper panels as labeled. However, the *relative* errors are huge where the oscillations between the roots have tiny amplitude. In graphical terms, the relative errors are large when the graph of the Wilkinson polynomial in the upper left panel of Fig. 3 dips close to the absolute error level.

When an asymptotic form is known, accuracy can be greatly improved by multiplying $W(x)$ by a scaling function—in this case, $\exp(-Nx^2/2)$—and reexpanding. Thus,

$$f_N(x) \approx \Omega(x)W(x;N_{\text{Wilkinson}}), \tag{36}$$

where the scaling function is

$$\Omega(x;N_{\text{Wilkinson}}) \equiv \exp(-(1/2)N_{\text{Wilkinson}}x^2). \tag{37}$$

Note that after such rescaling, the Wilkinson polynomial becomes essentially a sine function with a very small dynamic range as illustrated in the right upper panel of Fig. 3.

Since an asymptotic approximation will be available only rarely, it is instructive to attack the Wilkinson polynomial *without* such scaling. Fig. 4 shows that errors grow with the degree of the Wilkinson polynomial so that it is not possible to obtain any accuracy at all for $W(x;60)$. The reason for the failure is the dynamic range problem displayed in (35). The fault lies not in the companion matrix method, but rather in the fact that the Wilkinson polynomial of large degree has oscillations between the roots around $x = 0$ which are smaller in magnitude than the product of machine epsilon with the maximum of the polynomial. In the neighborhood of $x = 0$, the Chebyshev series is essentially generating random numbers with the magnitude of machine epsilon, and there is nothing that any rootsolver can do to retrieve this situation. The only remedy is to multiply the Wilkinson polynomial by an exponential scaling function as described above.

However, the contours of the error are almost vertical. This implies that there is no *accuracy* penalty for expanding $W(x;N)$ as a truncated Chebyshev series of degree larger than N; there is only the cost of unnecessary work. The figure thus confirms that there is remarkably little roundoff error with the Chebyshev companion matrix method.

Fig. 5 shows that the scaling works as advertised. With scaling, the degree N of the Chebyshev interpolant, which is the matrix size, must be chosen *larger* than the degree $N_{\text{Wilkinson}}$ of the Wilkinson polynomial so that the scaled

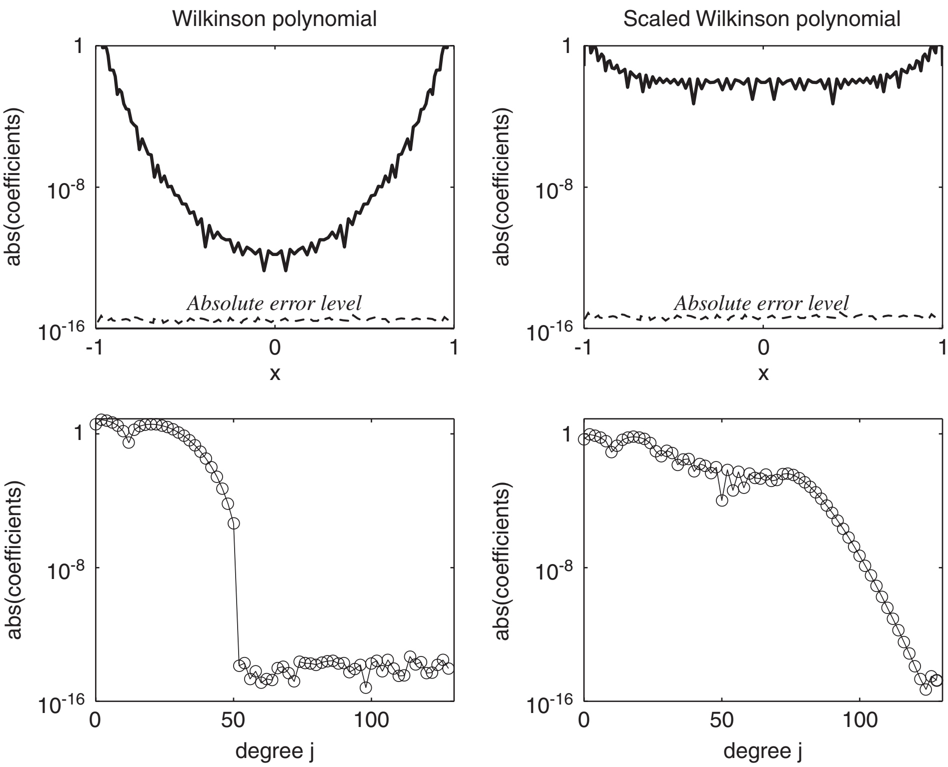

Fig. 3. Upper panels: the absolute value of the Wilkinson polynomial of degree 50 without scaling (left) and with scaling (right), both normalized to a maximum value of one. Bottom panels: the absolute values of the corresponding Chebyshev coefficients.

function—no longer a pure polynomial—is adequately approximated by a polynomial of degree N. Thus, the method is more costly with scaling than without, at least for the special case that $f(x)$ is a polynomial. Still, the reward for the extra cost in this example is that the maximum error is reduced by 10 orders of magnitude!

The errors have been plotted against the roots themselves to show that without scaling, the errors near $x = \pm 1$ where the Wilkinson polynomial has its largest amplitude, are just as tiny as with the scaling-by-Gaussian-function. However, without the scaling factor, the errors for the roots near the origin, where the unscaled polynomial is oscillating between very tiny maxima and minima, are relatively huge.

8. Multiple roots

The "power function"

$$f_{\mathrm{pow}}(x; k, x_0) = (x - x_0)^k \tag{38}$$

is useful because it allows us to examine the effects of multiple roots on the accuracy of the companion matrix algorithm. For the extreme values of $x_0 = 0$ (center of the interval) and $x_0 = 1$ (endpoint), we expanded f_{pow} as a Chebyshev series of degree N and then computed the eigenvalues for various N.

Multiple roots are very difficult because small perturbations will split a k-fold root into a cluster of k simple roots x_k, the so-called "multiple-root starburst":

$$f_{\mathrm{pow}}(x; k, x_0) - \varepsilon = 0 \quad \rightarrow \quad x_k = x_0 + \varepsilon^{1/k} \exp(\mathrm{i}2\pi j/k), \quad j = 1, \ldots, k. \tag{39}$$

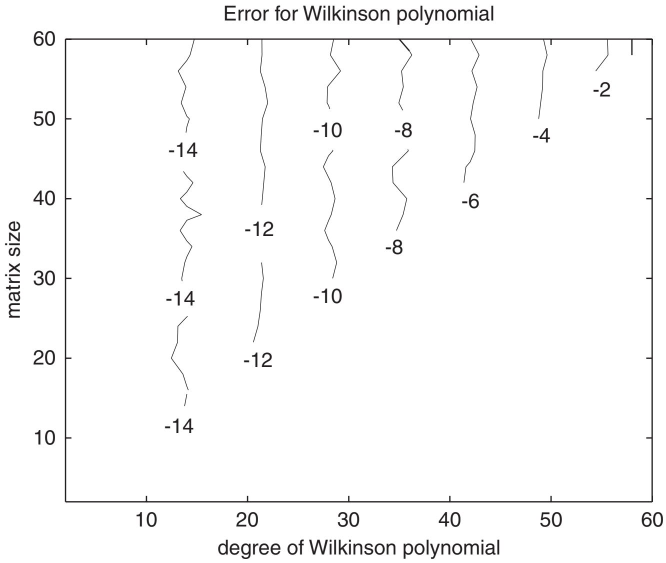

Fig. 4. A contour plot of the base-10 logarithm of the errors in computing the roots of the Wilkinson polynomial. (The contour labeled "-12" thus denotes an absolute error, everywhere along the isoline, of 10^{-12}.) The horizontal axis is the degree of the Wilkinson polynomial; the vertical axis is the degree of the interpolating polynomial and the size of the companion matrix, which may be greater than the degree of the Wilkinson polynomial. (When one computes an interpolating polynomial for the Wilkinson polynomial of higher degree N than that of the Wilkinson polynomial, $N_{\text{Wilkinson}}$, the high degree coefficients of the interpolant, i.e., those of degree $> N_{\text{Wilkinson}}$, are zero, modulo roundoff error; the graph shows that there is little penalty for choosing N larger than necessary. It is, however, more efficient to prune very tiny high degree coefficients before finding the eigenvalues of a companion matrix of smaller size.)

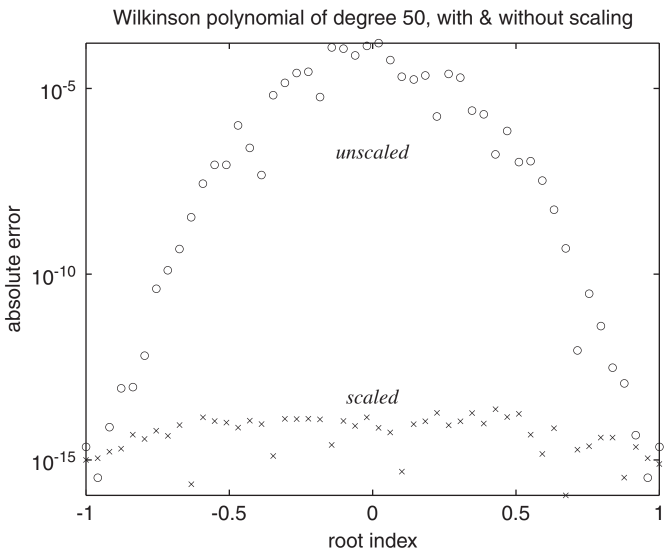

Fig. 5. Errors in computing the roots of the Wilkinson polynomial of degree $N_{\text{Wilkinson}} = 50$, with scaling (crosses) and without scaling (circles). The largest unscaled error was 1.6×10^{-4}. With scaling by multiplying by the factor $\exp(-(\frac{1}{2})N_{\text{Wilkinson}}x^2)$, the error was reduced to a maximum of 2.3×10^{-14}. The companion matrix size was $N = 130$.

Thus, if a polynomial with a triple zero ($k = 3$) is perturbed by adding a constant $\varepsilon = 10^{-15}$, the third-order zero becomes three simple roots each at a distance 10^{-5} (i.e., $\varepsilon^{1/3}$) in the complex plane from the unperturbed root. This instability to perturbations is an intrinsic property of multiple roots, and not something that one may reasonably expect a rootsolver to completely cure.

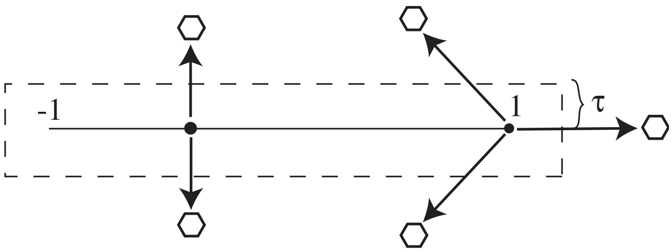

Fig. 6. Schematic of the "box of acceptance", $[-1-\tau, 1+\tau] \times [-\tau, \tau]$, about the canonical interval. The arrows show how a double root can be perturbed into a complex conjugate pair of roots and how a triple root can be perturbed into a root that is real, but off the canonical interval, plus a pair of complex roots. Roots of other orders at other locations are similarly sensitive to perturbations; the point is that roots will be missed unless the rectangle of acceptance is sufficiently large.

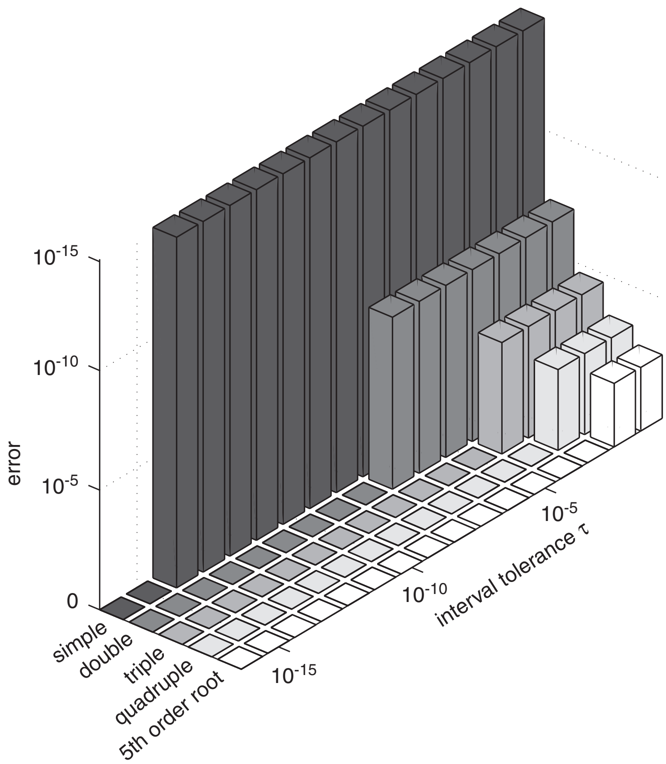

Fig. 7. The negative of the base-10 logarithm of the error in computing roots of $(x-1)^k$ for $k=1$ to 5 (labeled by "single", "double", etc., denoting the order of the roots) for various choices of the interval tolerance τ. The flat squares denote that algorithm failed for those values of k and τ.

One awkward feature of this sensitivity to roundoff errors is that tiny perturbations may move multiple roots off the real axis, or off the canonical interval $x \in [-1, 1]$ to larger real x as illustrated in Fig. 6. As noted earlier, the rootfinding algorithm should exclude roots far from the canonical interval where the Chebyshev approximation is likely inaccurate, but accept roots that lie within a small rectangle in the complex x-plane, $[-1-\tau, 1+\tau] \times [-\tau, \tau]$. Because multiple roots are easily perturbed off the interval, they may be missed if the "interval tolerance" τ is too small as shown schematically in Fig. 6.

Therefore, in Fig. 7, we have plotted errors for the power function with roots at the end of the canonical interval as functions not only of the order k of the zeros, but also of the interval tolerance τ in (8). For high-order zeros, roundoff on the order of 10^{-15} will move the roots to $x > 1 + \tau$ unless $\tau > \varepsilon^{1/k}$. When the algorithm thus failed to detect the roots, a flat square is plotted.

Clearly, for simple roots (column furthest from the viewer), the choice of τ makes little difference and the errors are around 10^{-15}. For higher-order roots, the "starburst" is great enough that the algorithm fails unless the error tolerance τ is sufficiently large. Even with an appropriate interval tolerance, the error grow to roughly $10^{-15/k}$.

9. Summary

The theorems and numerical experiments presented here show that the Chebyshev–Frobenius companion matrix method is a very reliable way to find the roots of a polynomial. One theorem shows that the roots are insensitive to perturbations in the Chebyshev coefficients: the Chebyshev form is a very stable and well-conditioned representation of a polynomial. Two other theorems show that if the polynomial is of definite parity with respect to the origin, and thus is composed of odd degree Chebyshev polynomials only or of even degree polynomials only, one can half the size of the companion matrix so that its size is one less than the number of *nonzero* Chebyshev coefficients.

For polynomials with random coefficients, with or without a superimposed exponential decay with degree, the error is $O(10^{-15})$, only an order of magnitude worse than machine epsilon, even for polynomials whose degree is in the hundreds. For special classes of polynomials which are intrinsically ill-conditioned, the companion matrix does as well as can be expected.

The error in the roots of the Wilkinson polynomial grows exponentially with degree, yielding no accuracy for $N > 60$. However, this is not the fault of the companion matrix algorithm, but is rather due to the exponential growth of the oscillations of the polynomial as $|x| \to 1$. The difficulty can be removed by scaling the Wilkinson polynomial by an exponential function of x, reexpanding the scaled function as a truncated Chebyshev expansion, and then applying the companion matrix method.

Acknowledgments

This work was supported by NSF grants OCEOCE9986368 and OCE0451951 and by the Undergraduate Research Opportunities Program (UROP) at the University of Michigan.

Appendix A. Formulas for Chebyshev manipulations

A.1. Fast summation of a truncated Chebyshev series or cosine series at an arbitrary point: Clenshaw–Horner recurrence

To evaluate

$$f_N \equiv \sum_{j=0}^{N} a_j T_j \left(\frac{2x - (b+a)}{b-a} \right) = \sum_{j=0}^{N} a_j \cos \left\{ j \arccos \left(\frac{2x - (b+a)}{b-a} \right) \right\}, \tag{40}$$

at an arbitrary point x, the fastest algorithm is the following recurrence. First, define

$$y = \frac{2x - (b+a)}{b-a}, \quad b_1 = 0, \quad b_2 = 0. \tag{41}$$

Second, apply the loop $N + 1$ times:

$$b_0 = 2yb_1 - b_2 + a_{N+1-j}, \quad b_3 = b_2, \quad b_2 = b_1, \quad b_1 = b_0, \quad j = 1, 2, \ldots, N. \tag{42}$$

The sum of the truncated series at x is then

$$f_N(x) = (1/2)(b_0 - b_3) + (1/2)a_0. \tag{43}$$

A.2. Computing the Chebyshev series for the qth derivative

Let $a_k^{(q)}$ denote the coefficients of the qth derivative:

$$\frac{\mathrm{d}^q u}{\mathrm{d}x^q} = \sum_k^N a_k^{(q)} T_k(x). \tag{44}$$

These may be computed from the Chebyshev coefficients of the $(q-1)$th derivative by the recurrence relation (in descending order)

$$a_N^{(q)} = a_{N-1}^{(q)} = 0,$$

$$a_{k-1}^{(q)} = \frac{1}{c_{k-1}}\{2ka_k^{(q-1)} + a_{k+1}^{(q)}\}, \quad k = N-1, N-2, N-3, \ldots, 1, \tag{45}$$

where $c_k = 2$ if $k = 0$ and $c_k = 1$ for $k > 0$.

A.3. Chebyshev interpolation of a function $f(x)$: Lobatto (endpoints-and-extrema) grid

Goal: to compute a Chebyshev series, including terms up to and including T_N, on the interval $x \in [a, b]$.
Step 1: Create the interpolation points (Lobatto grid):

$$x_k \equiv \frac{b-a}{2}\cos\left(\pi\frac{k}{N}\right) + \frac{b+a}{2}, \quad k = 0, 1, 2, \ldots, N. \tag{46}$$

Step 2: Compute the elements of the $(N+1) \times (N+1)$ interpolation matrix.
Define $p_j = 2$ if $j = 0$ or $j = N$ and $p_j = 1$, $j \in [1, N-1]$. Then the elements of the interpolation matrix are

$$\mathscr{I}_{jk} = \frac{2}{p_j p_k N}\cos\left(j\pi\frac{k}{N}\right). \tag{47}$$

Step 3: Compute the grid point values of $f(x)$, the function to be approximated:

$$f_k \equiv f(x_k), \quad k = 0, 1, \ldots, N. \tag{48}$$

Step 4: Compute the coefficients through a vector–matrix multiply:

$$a_j = \sum_{k=0}^N \mathscr{I}_{jk} f_k, \quad j = 0, 1, 2, \ldots, N. \tag{49}$$

The approximation is

$$f \approx \sum_{j=0}^N a_j T_j\left(\frac{2x-(b+a)}{b-a}\right) = \sum_{j=0}^N a_j \cos\left\{j \arccos\left(\frac{2x-(b+a)}{b-a}\right)\right\}. \tag{50}$$

A.4. Adaptive Chebyshev interpolation

Clenshaw and Curtis [11] developed an adaptive Chebyshev quadrature scheme. Their key observation, applicable equally well to interpolation, is that all points on the $(2N+1)$-point Chebyshev–Lobatto grid are also points on the $(N+1)$-point grid. Thus, one can approximately double N while evaluating $f(x)$, the function being interpolated, at only half the points of the new, denser grid.

To perform adaptive interpolation, specify an error tolerance ε and choose an initial N. Evaluate the errors in the $(N+1)$-point approximation at the interstitial points, the points of the $(2N+1)$-point Lobatto grid which are not also on the $(N+1)$-point grid. Define the $(N+1)$-point "interstitial error" as

$$E_N \equiv \max_k |f(x_k) - f_N(x_k)|, \quad x_k = \frac{b-a}{2}\cos\left(\pi\frac{(2k-1)}{2N}\right) + \frac{b+a}{2}, \quad k = 1, \ldots, N. \tag{51}$$

Double N and repeat the test until E_N is sufficiently small.

There are two good choices for the tolerance. The cheaper but less robust choice is to demand $E_N \leqslant \sqrt{\varepsilon}$. Because of the usual geometric convergence of spectral series, it follows that for sufficiently large N, the error in the $(2N+1)$-term Chebyshev series will be roughly ε.

A stricter criterion is to increase N by a factor of two until the error of the $(N+1)$-point approximation at the interstices is less than ε. This requires evaluating $f(x)$ at twice as many points as the degree of the final less-than-ε-in-error interpolant, but is independent of asymptotic expectations.

References

[1] S. Barnett, Companion matrix analog for orthogonal polynomials, Linear Algebra Appl. 12 (1975) 197–208.
[2] Z. Battles, L.N. Trefethen, An extension of Matlab to continuous functions and operators, SIAM J. Sci. Comput. 25 (2004) 1743–1770.
[3] C.M. Bender, S.A. Orszag, Advanced Mathematical Methods for Scientists and Engineers, McGraw-Hill, New York, 1978 594pp.
[4] A.D. Booth, Numerical Methods, Butterworths Scientific Publications, London, 1955.
[5] J.P. Boyd, A Chebyshev polynomial interval-searching method (Lanczos economization) for solving a nonlinear equation with application to the nonlinear eigenvalue problem, J. Comput. Phys. 118 (1995) 1–8.
[6] J.P. Boyd, Chebyshev and Fourier Spectral Methods, second ed., Dover, Mineola, New York, 2001 665pp.
[7] J.P. Boyd, Computing zeros on a real interval through Chebyshev expansion and polynomial rootfinding, SIAM J. Numer. Anal. 40 (2002) 1666–1682.
[8] J.P. Boyd, A review of high-order methods for incompressible fluid flow by M.O. Deville, P.F. Fischer and E.H. Mund, SIAM Rev. 46 (2004) 151–157.
[9] J.P. Boyd, Computing real roots of a polynomial in Chebyshev series form through subdivision, Appl. Numer. Math. 56 (2006) 1077–1091.
[10] J.P. Boyd, Computing real roots of a polynomial in Chebyshev series form through subdivision with linear testing and cubic solves, Appl. Math. Comput. 174 (2006) 1642–1648.
[11] C.W. Clenshaw, A.R. Curtis, A method for numerical integration on an automatic computer, Numer. Math. 2 (1960) 197–205.
[12] J.B. Conway, Functions of One Complex Variable, Springer, New York, 1973.
[13] D.M. Day, L. Romero, Roots of polynomials expressed in terms of orthogonal polynomials, SIAM J. Numer. Anal. 43 (2005) 1969–1987.
[14] E.M. Gol'berg, V.N. Malozemov, Estimates for the zeros of certain polynomials, Vestmol Leningrad Univ. Math. 6 (1979) 127–135 Translated from Vestnik Leningrad Univ. Mat. Mekh. Astronom. 7 (1973) 18–24.
[15] W. Specht, Die Lage der Nullstellen eines Polynoms III, Math. Nachr. 16 (1957) 369–389.
[16] W. Specht, Die Lage der Nullstellen eines Polynoms IV, Math. Nachr. 21 (1960) 201–222.
[17] H.J. Stetter, Numerical Polynomial Algebra, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2004.
[18] G. Szegö, Orthogonal Polynomials, second ed., American Mathematical Society, New York, 1959.
[19] L.N. Trefethen, D. Bau III, Numerical Linear Algebra, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.