STABILITY OF BARYCENTRIC INTERPOLATION FORMULAS

MARCUS WEBB*, LLOYD N. TREFETHEN†, AND PEDRO GONNET‡

Abstract. The barycentric interpolation formula defines a stable algorithm for evaluation at points in $[-1,1]$ of polynomial interpolants through data on Chebyshev grids. Here it is shown that for evaluation at points in the complex plane outside $[-1,1]$, the algorithm becomes unstable and should be replaced by the alternative modified Lagrange or "first barycentric" formula dating to Jacobi in 1825. This difference in stability confirms the theory published by N. J. Higham in 2004 (*IMA J. Numer. Anal.*, v. 24) and has practical consequences for computation with rational functions.

Key words. Barycentric interpolation, Chebfun, rational approximation, Bernstein ellipse, Chebfun ellipse

AMS subject classifications. 65D05, 41A20

1. The effect.

In experiments involving numerical analytic continuation in the complex plane, we encountered an instability that surprised us [18]. On investigation it proved to have an elegant explanation, and the effect is worth knowing about for those who compute with polynomials or rational functions. It has led to a significant correction in Chebfun [17].

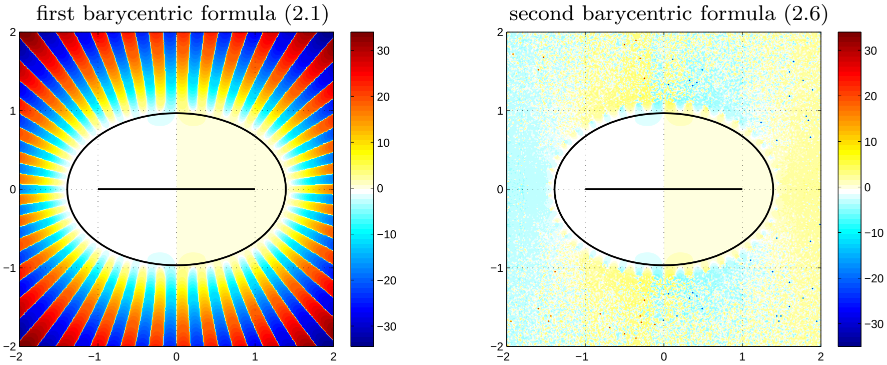

Fig. 1.1. *Numerically computed values* $\mathrm{Re}\,p(x)$ *in the complex* x*-plane, where* p *is the polynomial interpolant in* 42 *Chebyshev points in* $[-1,1]$ *to* $f(x)=\tanh(\pi x/2)$. *Since the aim is to show both sign and magnitude of* $\mathrm{Re}\,p(x)$, *the precise quantity plotted is* $\mathrm{sign}(\mathrm{Re}\,p(x))\cdot\log(1+|\mathrm{Re}\,p(x)|)$. *On the left, the computation uses the formula* (2.1) *and the picture is correct, with oscillatory behavior outside a certain ellipse enclosing* $[-1,1]$. *On the right, the computation is based on the more usual formula* (2.6) *and the picture is entirely wrong outside approximately the same ellipse, showing seemingly random values that fail to grow in magnitude as* $|x|\to\infty$. *The errors are significant inside the ellipse, too, though this is not apparent in the plot.*

Figure 1.1 illustrates the effect. The function $f(x)=\tanh(\pi x/2)$ has been approximated to 16 digit accuracy on $[-1,1]$ by interpolation by a polynomial $p(x)$ of degree n in $n+1$ Chebyshev points $x_j=\cos(\pi j/n)$, $0\le j\le n$; the value that achieves

*Worcester College, Oxford OX1 2HB, UK (webb@maths.ox.ac.uk)

†Oxford University Mathematical Institute, 24–29 St Giles', Oxford OX1 3LB, UK (trefethen@maths.ox.ac.uk, http://people.maths.ox.ac.uk/trefethen)

‡Oxford University Mathematical Institute, 24–29 St Giles', Oxford OX1 3LB, UK (gonnet@maths.ox.ac.uk, http://people.comlab.ox.ac.uk/gonnet)

the prescribed accuracy is $n = 41$. The figure shows contour plots of the numerically computed function $\mathrm{Re}\,p(x)$ in the complex x-plane. The result labeled "first barycentric formula" is correct, but the result labeled "second barycentric formula" is entirely wrong outside an ellipse enclosing $[-1, 1]$, and also quite inaccurate for values of x inside the ellipse. The ellipse in question is a Bernstein ellipse, that is, with foci ± 1, the largest such ellipse inside of which f is analytic.

2. Two variants of the barycentric formula.

By the first barycentric formula, we mean the modified Lagrange interpolation formula

$$p(x) = \ell(x) \sum_{j=0}^{n} \frac{\lambda_j f_j}{x - x_j}, \tag{2.1}$$

where $\ell(x)$ is the node polynomial

$$\ell(x) = \prod_{k=0}^{n} (x - x_k) \tag{2.2}$$

and the numbers λ_j are the barycentric weights

$$\lambda_j = \frac{1}{\ell'(x_j)}. \tag{2.3}$$

For the Chebyshev points, the weights turn out to be

$$\lambda_j = \frac{2^{n-1}}{n}(-1)^j, \quad 1 \le j \le n-1, \tag{2.4}$$

and half these values for $j = 0$ and n. For derivations, see [1] or [16]. Formulas (2.1)–(2.3) originate with Jacobi in 1825, and (2.4) is due to Marcel Riesz in 1916 [6, 11]. We are grateful to Folkmar Bornemann of the Technical University of Munich for drawing our attention to these early papers.

The second barycentric formula is obtained as follows. If we represent the constant function $f(x) = 1$ by (2.1), we get

$$1 = \ell(x) \sum_{j=0}^{n} \frac{\lambda_j}{x - x_j}, \tag{2.5}$$

and dividing (2.1) by (2.5) gives

$$p(x) = \sum_{j=0}^{n} \frac{\lambda_j f_j}{x - x_j} \Bigg/ \sum_{j=0}^{n} \frac{\lambda_j}{x - x_j}. \tag{2.6}$$

Equation (2.6) originates with Taylor and Dupuy in the 1940s, and the special case of Chebyshev points was first treated by Salzer in 1972 [2, 13, 15]. The terms "first" and "second" come from Rutishauser [12]. Formula (2.6) has a special elegance about it, since the factor $\ell(x)$ of (2.1) has dropped out, and this feature has practical consequences. That factor has size approximately 2^{-n} for $x \in [-1, 1]$, making (2.2) susceptible to underflow in floating point arithmetic for large n; similarly (2.4) shows that the weights λ_j have size approximately 2^n, leading to a risk of overflow when λ_j is defined by (2.4). Moreover, all these numbers change with nth powers if $[-1, 1]$ is rescaled to a general interval $[a, b]$; the number 2 arises for $[-1, 1]$ because the logarithmic capacity of this interval is 1/2. With the formula (2.6), however, we can cancel the common factor $2^{n-1}/n$ from (2.4), taking the weights in both numerator and denominator to be simply ± 1 in the interior and $\pm\frac{1}{2}$ at the endpoints. This makes (2.6) scale-invariant and circumvents all problems of underflow and overflow.

3. Explanation of the instability. Both formulas (2.1) and (2.6) work beautifully for interpolating a function on a Chebyshev grid in $[-1, 1]$ and evaluating the interpolant at points in $[-1, 1]$ (assuming the underflow/overflow problem of (2.1) is addressed for large n, which can be done by reformulating it via logarithms). This numerical stability of (2.6) has been emphasized over the years by Salzer, Rutishauser, and other authors, including Henrici [4], and it is relied upon by Chebfun for polynomial interpolants even in millions of points. In terms of theoretical support, two important contributions are a paper by Rack and Reimer in 1982 [10] and a definitive work by N. J. Higham in 2004 [5].

Yet Higham's analysis identifies an Achilles heel in (2.6). He shows that (2.1) has the gold standard property of backward stability: under standard assumptions of floating point arithmetic, the computed result $\hat{p}(x)$ delivered by (2.1) is the exactly correct value for a set of data $\{\hat{f}_j\}$ that differ from $\{f_j\}$ by relative perturbations no greater than about $5nu$, where u is the unit roundoff (typically $u \approx 10^{-16}$). Formula (2.6), on the other hand, is not backward stable, satisfying only a more restrictive forward stability bound. For $x \in [-1, 1]$ and interpolation in Chebyshev points, which is the most familiar case from applications such as ordinary Chebfun computations, there is little difference in the two bounds, and (2.6) is stable. As x moves away from from $[-1, 1]$, however, the forward stability bound grows rapidly, and (2.6) becomes unstable.

Like many numerical instabilities, this one is a consequence of cancellation. The accuracy of (2.6) depends on the accuracy of (2.5), which we can rewrite as

$$\sum_{j=0}^{n} \frac{\lambda_j}{x - x_j} = \frac{1}{\ell(x)}. \tag{3.1}$$

Higham calls this "a mathematical identity that does not necessarily hold in floating point arithmetic." The problem is that as x moves farther from $[-1, 1]$, $1/\ell(x)$ shrinks rapidly, so the validity of (3.1) relies increasingly on cancellation. In floating point arithmetic, the result is rapid loss of accuracy, until soon there are no accurate digits at all.

Making the argument more quantitative explains why ellipses arise. The weights λ_j on the left of (3.1) have size approximately 2^n, and for $x \in [-1, 1]$, the right-hand side is of the same order since $\ell(x)$ is approximately of size 2^{-n}. Thus cancellation is not an issue. As x moves away from $[-1, 1]$, however, $\ell(x)$ grows to order approximately $2^{-n}\rho^n$, where ρ is the parameter of the Bernstein ellipse passing through x. (That is, $x = (w + w^{-1})/2$ for some w with $|w| = \rho$ [16].) This means that (3.1) relies on cancellation of magnitude ρ^n to occur, so we must expect loss of accuracy by approximately this factor. If ρ^n is of size u^{-1} or larger, all the accuracy will be lost. If $|x|$ is increased still further, so that ρ^n grows beyond u^{-1}, then (3.1) will fail to decrease any further and the computed $p(x)$ from (2.6) will fail to increase any further. As is apparent in the second image of Figure 1.1, we find ourselves with a polynomial $\hat{p}(x)$ that is bounded as $|x| \to \infty$!

Suppose a function f of size approximately 1 is interpolated in Chebyshev points to machine precision on $[-1, 1]$, the basic operation of Chebfun. It is customary to

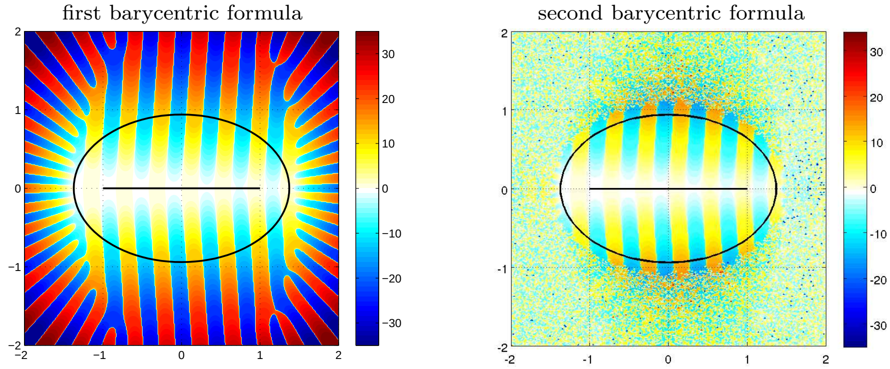

Fig. 3.1. *Repetition of Figure* 1.1 *for* $f(x) = \exp(x)\sin(15x)$*. Again the erroneous results appear approximately outside an ellipse, the Chebfun ellipse for this function.*

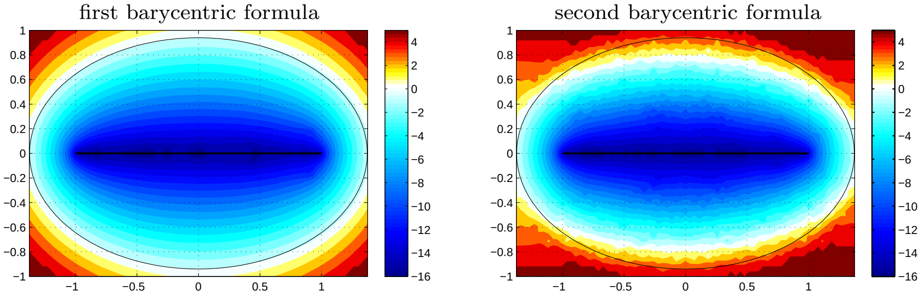

Fig. 3.2. *Contour plot of numerically computed values* $\log_{10}|f(x) - p(x)|$ *for formulas* (2.1) *and* (2.6) *with* $f(x) = \exp(x)\sin(15x)$*. The first plot is unaffected by rounding errors, so it is a correct picture of interpolation errors only, showing accuracy precisely within the Chebfun ellipse. The errors in the second image are significantly larger because of rounding errors amplified by instability.*

define the *Chebfun ellipse* for f as that Bernstein ellipse whose parameter ρ satisfies $\rho^n = u^{-1}$, where n is the degree of p, and it is inside this ellipse that p can be expected to have some accuracy as an approximation to f. Because the Chebfun ellipse is determined by the factor ρ^n, we can expect the numerical accuracy of the barycentric formula to be lost entirely, approximately, if x lies outside it. In other words, for just the values of x for which $p(x)$ has no accuracy as an approximation of f, the evaluation of $p(x)$ by (2.6) is also completely inaccurate because of numerical instability.

For many functions, the Chebfun ellipse is approximately the Bernstein ellipse that passes through the closest singularity of f to $[-1, 1]$, since that singularity controls the value of n needed to represent f to machine precision. However, the Chebfun ellipse may be smaller than this, for example if f is an entire function (analytic throughout the complex plane). Figure 3.1 shows such a case, repeating Figure 1.1 for the entire function $f(x) = \exp(x)\sin(15x)$.

Figure 3.2 compares the two barycentric formulas from another angle, looking at the computed differences $|f(x) - p(x)|$ between the function and its interpolant.

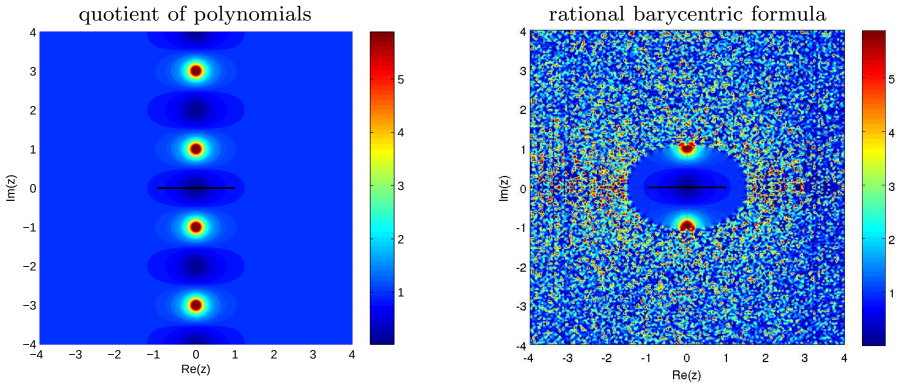

Fig. 4.1. *Contour plots of* $|r(z)|$ *for type* $(9,6)$ *rational linearized least-squares fits in* 42 *Chebyshev points in* $[-1,1]$ *to* $\tanh(\pi x/2)$*, based on calling Chebfun's* `ratinterp` *command with* $(m,n) = (10,10)$*. The unstable computation on the right uses a rational barycentric formula, whereas on the left* $r(x) = p(x)/q(x)$ *is computed by applying* (2.1) *separately to* $p(x)$ *and* $q(x)$*.*

4. Practical implications.

Confirming Higham's theory of [5], we have shown that the standard barycentric interpolation formula (2.6) should not be used for evaluating a polynomial outside the interval of interpolation. One should use (2.1) instead, after reformulating it via logarithms to avoid over- and underflow. We close by mentioning two contexts in which this and a related observation may have practical importance.

First, suppose one wants to evaluate a rational function r in the complex plane. This task arises frequently, because Padé approximations and their relatives offer the best known techniques for many problems of extrapolation and analytic continuation. One of the attractions of the barycentric formula (2.6) is that it generalizes to rational functions, as pointed out first by Schneider and Werner [1, 14], and this is the method used by Chebfun up till now to evaluate the rational functions that result from interpolation and least-squares, minimax, Chebyshev–Padé, and Carathéodory–Fejér approximation (Chebfun commands `ratinterp`, `remez`, `chebpade`, `cf`). We find that the instability described in this paper applies to the rational analogues of (2.6) too, and so it is not a good idea to evaluate $r(x)$ in this way. Since it is not clear that a rational analogue of (2.1) exists, we recommend evaluating $r(x) = p(x)/q(x)$ by treating the numerator and denominator separately with (2.1) and then just taking the quotient. This adjustment was crucial to the success of the explorations reported in [18]. An example of stable and unstable evaluation of a rational function is shown in Figure 4.1.

We will not give details for rational approximants in this paper, because this topic is considerably more involved than the polynomial case and we are currently investigating how best to treat it. The rational interpolation scheme used for Figure 4.1, implemented in Chebfun's `ratinterp` command, is an analogue for Chebyshev points in $[-1,1]$, as in [8], of the scheme presented in [3] for robust rational interpolation and least-squares fitting in roots of unity on the unit disk.

Second, questions arise about evaluation of both polynomial rational functions even within $[-1,1]$ when the interpolation points are far from the Chebyshev distribution. For polynomials, this was the example focused on by Higham to illustrate the possible instability of (2.6). As it happens, barycentric formulas have been used apparently to great advantage by Pachón for working with the irregular reference sets that arise in the Remez algorithm for computing best polynomial and rational approximations [7, 9]. It would appear that it may be worth revisiting these algorithms in the light of our new practical experience of the instability of (2.6).

Acknowledgments. This research originated with a EPSRC Vacation Bursary awarded to MW to support a summer undergraduate research project.

REFERENCES

[1] J.-P. Berrut and L. N. Trefethen, Barycentric Lagrange interpolation, *SIAM Rev.* 46 (2004), 501–517.

[2] M. Dupuy, Le calcul numérique des fonctions par l'interpolation, *C. R. Acad. Sci.* 226 (1948), 158–159.

[3] P. Gonnet, R. Pachón and L. N. Trefethen, Robust rational interpolation and least-squares, *Elect. Trans. Numer. Anal.* 38 (2011), 146–167.

[4] P. Henrici, *Essentials of Numerical Analysis,* Wiley, 1982.

[5] N. J. Higham, The numerical stability of barycentric Lagrange interpolation, *IMA J. Numer. Anal.* 24 (2004), 547–556.

[6] C. G. J. Jacobi, *Disquisitiones Analyticae de Fractionibus Simplicibus*, thesis, Berlin, 1825.

[7] R. Pachón, *Algorithms for Polynomial and Rational Approximation in the Complex Domain,* DPhil thesis, Oxford University Mathematical Institute, 2010.

[8] R. Rachón, P. Gonnet and J. van Deun, Fast and stable rational interpolation in roots of unity and Chebyshev points, *SIAM J. Numer. Anal.,* submitted.

[9] R. Pachón and L. N. Trefethen, Barycentric-Remez algorithms for best polynomial approximation in Chebfun, *BIT Numer. Math.* 49 (2009), 721–741.

[10] H.-J. Rack and M. Reimer, The numerical stability of evaluation schemes for polynomials based on the Lagrange interpolation form, *BIT* 22 (1982), 101–107.

[11] M. Riesz, Eine trigonometrische Interpolationsformel und einige Ungleichungen für Polynome, *Jahresber. Deutsch. Math.-Ver.* 23 (1914), 354–368.

[12] H. Rutishauser, *Lectures on Numerical Mathematics,* M. H. Gutknecht et al, eds., Birkhäuser, 1990.

[13] H. E. Salzer, Lagrangian interpolation at the Chebyshev points $x_{n,\nu} = \cos(\nu\pi/n)$, $\nu = 0(1)n$; some unnoted advantages, *Computer J.* 15 (1972), 156–159.

[14] C. Schneider and W. Werner, Some new aspects of rational interpolation, *Math. Comp.* 47 (1986), 285–299.

[15] W. J. Taylor, Method of Lagrangian curvilinear interpolation, *J. Res. Nat. Bur. Stand.* 35 (1945), 151–155.

[16] L. N. Trefethen, *Approximation Theory and Approximation Practice*, book in preparation.

[17] L. N. Trefethen and others, Chebfun Version 4.1, The Chebfun Development Team, 2011, `http://www.maths.ox.ac.uk/chebfun/`.

[18] M. Webb, article in preparation to be submitted to *SIAM Undergraduate Research Online.*