Bernstein Bases are Optimal, but, sometimes, Lagrange Bases are Better

Robert M. Corless and Stephen M. Watt

Ontario Research Centre for Computer Algebra
Department of Applied Mathematics
Department of Computer Science
University of Western Ontario
London, Canada
{Rob.Corless,Stephen.Watt}@uwo.ca

Abstract. Experimental observations of rootfinding by generalized companion matrix pencils expressed in the Lagrange basis show that the method can sometimes be numerically stable, and indeed sometimes be much more stable than rootfinding of polynomials expressed in even the Bernstein basis. This paper details some of those experiments and provides a theoretical justification for this. We prove that a new condition number, defined for points on a set containing the interpolation points, is never larger than the rootfinding condition number for the Bernstein polynomial; and computation shows that sometimes it can be much smaller. This result may be of interest for those who wish to find the zeros of polynomials given simply by values.

1 Introduction

There are two parallel threads of recent research into polynomial computation using alternative bases, other than the power basis. The motivation for both threads is that *conversion between bases can be unstable*, and the instability increases with the degree [11]. These threads are of interest both to the computer algebra community and to the numerical analysis community because both threads involve hybrid symbolic-numeric computation.

One thread, exemplified by the papers [3, 8, 9, 16, 17] investigates polynomial computation via the Bernstein basis, which is well-suited to applications in computer-aided geometric design. Indeed, the papers [7, 13] prove that in a certain sense Bernstein bases are optimal both for evaluation and for rootfinding.

The other parallel thread, exemplified by the papers [1, 2, 5, 14], investigates computation with polynomials expressed in the Lagrange basis, or in other words directly by values. Related works include [10, 15], which use the Lagrange basis as an intermediate step in the computation and analysis of polynomial roots.

This present paper imitates the proofs of [7, 13] to show that while Bernstein bases are optimal in the class of bases nonnegative on the interval $[0, 1]$, Lagrange bases can sometimes be better (even though they can be negative on the interval). The motivation for these proofs is given by the following examples, which show good numerical behaviour of algorithms for polynomials expressed by values.

We assume henceforth that all polynomials under consideration have only simple roots. In one example we see what can happen if that assumption is violated.

1.1 Definitions

Following [5], the *generalized companion matrix* of the polynomial given by its values is defined below. If the values of the (matrix) polynomial at $x = x_0$, $x = x_1$, ..., and x_n are (the matrices) $\mathbf{P_0}$, $\mathbf{P_1}$, ... and $\mathbf{P_n}$, then the generalized companion matrix pencil is

$$\mathbf{C_0} = \begin{bmatrix} x_0 I & & & & \mathbf{P_0} \\ & x_1 I & & & \mathbf{P_1} \\ & & \ddots & & \vdots \\ & & & x_n I & \mathbf{P_n} \\ -\ell_0 I & -\ell_1 I & \cdots & -\ell_n I & 0 \end{bmatrix} \tag{1}$$

where the $\ell_k = 1/\prod_{j \neq k}(x_k - x_j)$ are the (scalar) normalization factors of the Lagrange polynomials $L_k(x) = \ell_k \prod_{j \neq k}(x - x_j)$, and

$$\mathbf{C_1} = \begin{bmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & 0 \end{bmatrix} \tag{2}$$

where the blocks I are conformal with the square blocks $\mathbf{P_k}$. In this paper we use only the scalar case: each matrix $\mathbf{P_k}$ is just 1×1. Then[1] $\det(x\mathbf{C_1} - \mathbf{C_0}) = \det \mathbf{P(x)} = \det(L_0(x)\mathbf{P_0} + L_1(x)\mathbf{P_1} + \cdots + L_n(x)\mathbf{P_n})$. Here, the determinant of a 1×1 matrix is of course just the entry, and hereafter we will write $p(x)$ instead of $\mathbf{P(x)}$. Thus the eigenvalues of the pencil $(\mathbf{C_0}, \mathbf{C_1})$ are, aside from an extraneous double root at infinity, exactly the roots of $p(x)$. Notice that the Lagrange polynomial is not converted to monomial form (or indeed even formed explicitly, though the normalization factors are).

The *conditioning* of a problem measures sensitivity of the solution to changes in the problem data. This notion is extremely useful in applied mathematics and in numerical analysis, because it can also be used to estimate the sensitivity of the solution to changes in the problem *formulation* and, by virtue of *backward error analysis*, the sensitivity of the problem to numerical errors. There are hierarchies of condition numbers for several problems, and in [13] we find an analogue of the *structured* condition number of linear algebra (see [12] for an overview) defined

[1] Note that [5] consistently had the wrong sign, writing instead $\det(C_0 - xC_1)$.

and used for evaluation of polynomials; the paper [7] shows that dividing an evaluation condition number by $|p'(r)|$ gives a rootfinding condition number. We summarize the analogues in the context of Lagrange bases, here.

Let U be a finite-dimensional vector space of functions defined on $\Omega \in \mathbb{R}^s$ and let $b = (b_0, b_1, \ldots, b_n)$ be a basis for U. Let $T \in \Omega$ be a (possibly finite) set which we will use to characterize nonnegativity of the basis: both [7] and [13] take $T = \Omega$ but we will occasionally take $T \subset \Omega$ to be a set containing the interpolation points. We will also take $s = 1$ in this paper. If $f \in U$ has the expansion $f(x) = \sum_{i=0}^{n} c_i b_i(x)$, we consider the relatively perturbed function $g = \sum_{i=0}^{n} (c_i + \delta_i c_i) b_i(x)$ and look at the differences between the roots of g and those of the unperturbed f. Similar considerations hold for the values of g compared to the values of f.

$$C_b(f,x) := \frac{1}{|p'(x)|} \sum_{i=0}^{n} |c_i b_i(x)| \tag{3}$$

$$\operatorname{cond}(b; f, x) := \frac{C_b(f,x)}{\|f\|_\infty} \tag{4}$$

$$\operatorname{cond}_T(b, f) := \sup_{x \in T} \operatorname{cond}(b; f, x) \tag{5}$$

With $\varepsilon = \|\delta\|_\infty$ we have (asymptotically as $\varepsilon \to 0$)

$$|r - \operatorname{RootOf}(g, x = r)| = C_b(f, r)\varepsilon + O(\varepsilon^2)$$

where $\operatorname{RootOf}(g, x = r)$ means the root of g closest to r.

2 Examples

2.1 The Wilkinson polynomial

In [7] we find the scaled Wilkinson polynomial

$$W_1 = \prod_{k=1}^{20} \left(x - \frac{k}{20} \right)$$

used as a test example. They show in their Fig. 1 that $C_B(W_1, r)$ where r runs through the set of roots $\{k/20\}$ of the polynomial is smaller than the condition number of either the power basis or the Ball basis.

We now take a random set of interpolation points on $(0, 1)$ (that happen to be approximations of random rationals with denominator 65537, sorted into increasing order)

$$\begin{array}{lll}0.066328943955323, & 0.19096083128614, & 0.25790011749088, \\ 0.26241665013656, & 0.31928528922593, & 0.33574927140394, \\ 0.34287501716588, & 0.34743732548026, & 0.45806185818698, \\ 0.48554251796695, & 0.49897004745411, & 0.54495933594763, \\ 0.57471352060668, & 0.62024505241314, & 0.74831011489693, \\ 0.75400155637274, & 0.77173199871828, & 0.80081480690297, \\ 0.81007675053786, & 0.85061873445535, & 0.97093245037155\end{array} \quad (6)$$

and compute W_1 for each of those floating-point values:

$$\begin{array}{rrr}-0.000000000010728612488953, & -2.7270553171585 \times 10^{-13}, & -3.7461930569485 \times 10^{-14} \\ -4.9745762165201 \times 10^{-14}, & 2.0845164306348 \times 10^{-14}, & 1.3201856930612 \times 10^{-14} \\ 6.5526224274051 \times 10^{-15}, & 2.2658972444804 \times 10^{-15}, & -2.2636597596148 \times 10^{-15} \\ -3.2717855666580 \times 10^{-15}, & -2.5904979790280 \times 10^{-16}, & 1.2337371879460 \times 10^{-15} \\ -4.3034948544772 \times 10^{-15}, & 5.3662056027121 \times 10^{-15}, & 3.2576970996232 \times 10^{-15} \\ -8.5440787103611 \times 10^{-15}, & -4.8509313053779 \times 10^{-14}, & 4.9746456902230 \times 10^{-15} \\ 7.2724485639224 \times 10^{-14}, & -1.5098138137091 \times 10^{-14}, & -5.7142694218465 \times 10^{-11}\end{array} \quad (7)$$

The generalized companion matrix of [5] was constructed and its eigenvalues found; the two extraneous infinite roots were discarded, and we were left with approximations to the roots $r = \{k/20\}$. *The maximum error in any approximation was* 7.1×10^{-12}. This contrasts with a maximum error in the Bernstein basis of approximately 10^{-7}. Thus we see that this (random) Lagrange basis is about four orders of magnitude better than the (optimal) Bernstein basis. We graph the condition numbers $C_B(W_1, r)$ and $C_L(W_1, r)$ in Fig. 1.

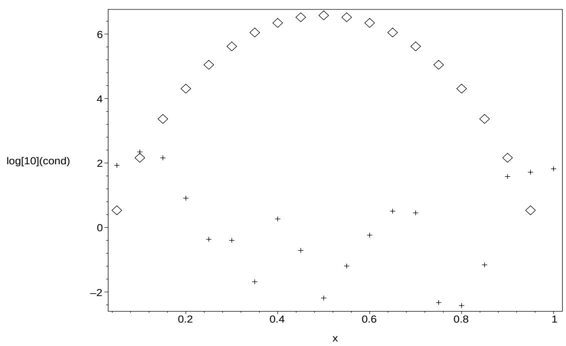

Fig. 1. Root condition of the Wilkinson polynomial $(x-1/20)(x-2/20)\cdots(x-20/20)$ expressed in the Bernstein basis (diamonds, C_B) and a random Lagrange basis (crosses, C_L). We see that the maximum condition of the polynomial in the Lagrange basis is four orders of magnitude lower than the maximum condition in the Bernstein basis, but that the Lagrange basis is not systematically better.

2.2 The second Wilkinson polynomial

Wilkinson also used another polynomial,

$$W_2 = \prod_{k=0}^{19} \left(x - 2^{-k}\right) \ ,$$

to investigate stability.[2] The paper [7] confirms in their Fig. 2 that the power basis is good, but the Bernstein basis is slightly better. They also remark that the Chebyshev basis is particularly bad, having root condition numbers as large as 10^{55}.

We can do 'better'. Taking the same random x-samples, and evaluating W_2 there, and computing the generalized companion matrix and finding the roots gives us 6-place accuracy for only the three largest roots 1, 1/2 and 1/4, and no accuracy at all for any smaller root (except that, perhaps accidentally, we get the root 1/16 to about 6% accuracy—but we do not find any approximation to the root 1/8 accurate to even one figure). In fact, the root condition numbers for this Lagrange basis are as large as 10^{63}.

But when instead of uniformly random interpolation points on $[0, 1]$ we choose a random point in each interval $[2^{-k-1}, 2^{-k}]$ for $k = 0 \ldots 19$, (a reasonable thing to do if we suspect the roots are near the origin) then the story is quite different. This time, we get all roots with relative accuracy at least as good as 1.3×10^{-10}, and the accuracy is as good for small roots as for large: the smallest root is accurate to eleven places. The root condition numbers $C_B(f, r)$ and $C_L(f, r)$ are plotted in Fig. 2.

2.3 Interpolation of analytic $f(z)$ around a circle

Consider the analytic function

$$y = f(z) = ze^z + e^{-1}$$

and evaluate it at $N+1$ points equally spaced around the circle $|z| = R$. Then the eigenvalues of the companion matrix pencil with these (z, y) values will give the zeros of the polynomial interpolating $f(z)$ at these points. One wonders whether or not zeros of $f(z)$ may be computed in this way. Of course, for this example, the zeros of $f(z)$ are just $W_k(-e^{-1})$, the values of the Lambert W function at $-e^{-1}$ (see [6] for a description of this function). Two branches of the function take on the values $W_0(-e^{-1}) = W_{-1}(-e^{-1}) = -1$ so this function has a double root. This may cause difficulty for the companion matrix method.

[2] He was surprised that his first polynomial (not intended to be difficult at all) turned out to have poor root conditioning; he was also surprised that this polynomial (intended to be difficult because the roots cluster at 0) turned out to have good root conditioning in the power basis. See his Chauvenet prize paper, "The Perfidious Polynomial".

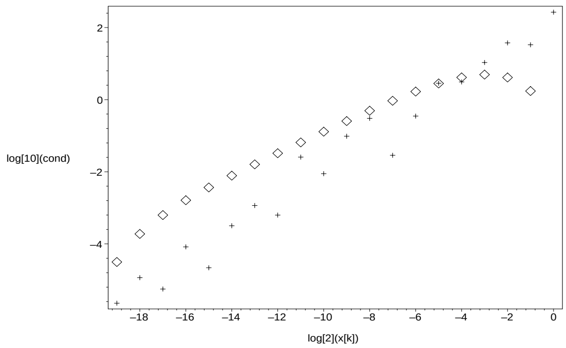

Fig. 2. Root condition of the second Wilkinson polynomial $(x-1)(x-1/2)\cdots(x-2^{-19})$ expressed in the Bernstein basis (diamonds) and a nonuniform random Lagrange basis (crosses).

We see in Fig. 3 that using $R = 10$, $N = 80$ gives an interesting picture. All the roots inside $R = 10$ of $f(z) = 0$ are computed to visual accuracy. The roots outside the circle are not. There are many roots of the interpolating polynomial that are not approximations to roots of $f(z)$. Nonetheless this provides good evidence that the polynomial rootfinding is working. The double root is computed as two simple roots at $-0.99997503474322 - 4.9463906 \times 10^{-5}i$ and $-1.0000249642602 + 4.9465335 \times 10^{-5}i$. Note that their average is accurately -1 to eight places, improving the accuracy attained as is usual with double roots.

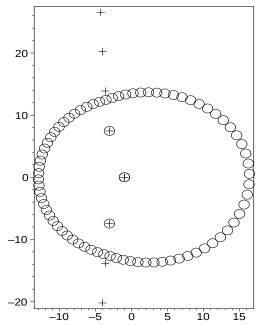

Fig. 3. Some computed roots of $f(z) = z\exp(z) + \exp(-1)$ (crosses), compared with (circles) $N = 80$ roots of a polynomial interpolating $f(z)$ on $|z| = R = 10$ at points equally spaced around the circle.

2.4 Visible Structures in Number Theory

In [4] we find an experimental exploration of the zeros of polynomials that have coefficients either 0 or 1, when expressed in the monomial basis. Similarly, they explore polynomials that have coefficients only taken from -1, 0, and 1. We may do the same thing here, in the Lagrange basis, and look at the resulting patterns to see if we can detect any numerical anomalies.

We choose to set the polynomials to be either $+1$ or -1 at roots of unity for our first example. Then for the degree N case there are $N+1$ points, but the polynomial roots are invariant if we multiply all polynomial values by -1 and hence there are only 2^N such polynomials with different root sets. This gives a plot with $N \cdot 2^N$ points. See the Figures.

Each figure displays certain symmetries that we would expect, and the symmetries lend confidence to our belief that the roots are accurate. We lose confidence in the roots as they get larger (and therefore farther from the interpolation points) because there seems to be no symmetry (especially in Fig. 5).

Checking one random polynomial from the 2^{15} polynomials from Fig. 5 we find, however, that the computed roots (even the larger ones, about magnitude 1.65) are accurate to all but one or two units in the last place. The polynomial that is 1 at all points has only spurious computed roots, however; the pencil is singular and the computation of the eigenvalues is simply erroneous (and invisibly so—the putative roots are ordinary looking complex numbers of size about 20 or so).

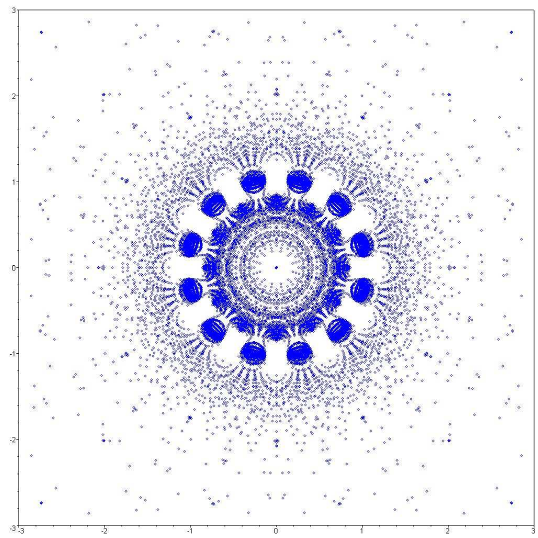

Fig. 4. All roots of polynomials taking on values ± 1 at the 12th roots of unity.

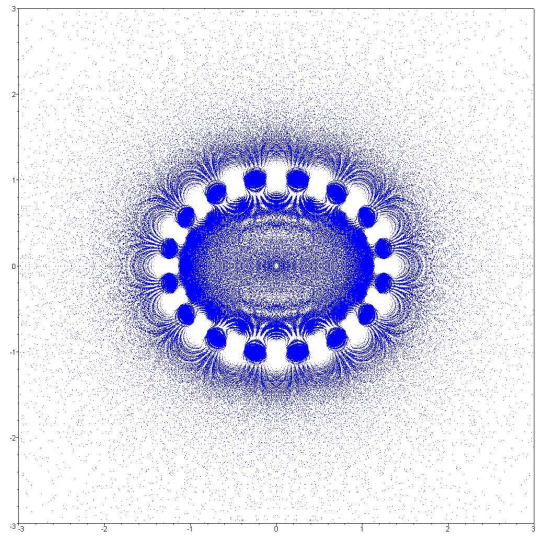

Fig. 5. All roots of polynomials taking on values ± 1 at 16 points equally-spaced on a parameterized ellipse. Computing these (over 400,000) zeros (actually, twice each) took about 5 hours on a notebook computer.

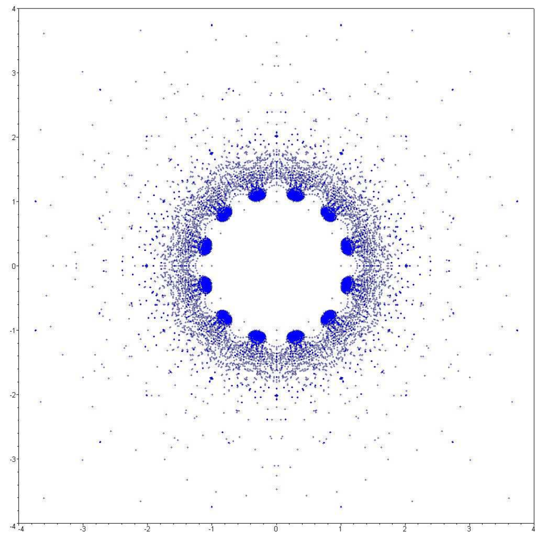

Fig. 6. All roots of polynomials taking on values either 0 or 1 at the 12th roots of unity.

3 Theoretical Analysis

These experiments, and others not presented here, help to convince us that sometimes this approach is very accurate (and, incidentally, that the approach is reasonably efficient, even with just using off-the-shelf software for computing the generalized eigenvalues). We present here some theorems that justify the (occasional) successes of this method.

For the moment we consider only polynomials defined on the interval $[0, 1]$.

Lemma 1. *Both the Bernstein and the power basis functions can be expressed as a nonnegative combination of any Lagrange basis with interpolation points taken on* $[0, 1]$*. By nonnegative combination we mean that each coefficient in the combination is nonnegative. Of course there must be some positive coefficients.*

Proof. Elements x^k of the power basis may be written as

$$x^k = \sum_{i=0}^{n} x_i^k L_i(x)$$

and the coefficients x_i^k are obviously nonnegative. Similarly, elements $b_k^n(x)$ of the Bernstein basis may be written as

$$b_k^n(x) = \binom{n}{k} (1-x)^{n-k} x^k = \sum_{i=0}^{n} \binom{n}{k} (1-x_i)^{n-k} x_i^k L_i(x)$$

and again the coefficients are obviously nonnegative since $0 \le x_i \le 1$.

Lemma 2. *The Lagrange polynomials* $L_i(x)$ *are nonnegative on the interpolation points.*

Proof. This is obvious: they take on only the values 0 or 1 on the interpolation points. However, we would like nonnegativity in an open set around the interpolation points, which we do not have.

Proposition 1. *Fix a set of interpolation points* $[x_0, x_1, \ldots, x_n]$*. If any basis* B *can be expressed as a nonnegative combination of the Lagrange basis on this set of points, then there exists a set* T*, depending on* f *and containing the interpolation points, in which* $C_L(f, T) \le C_B(f, T)$*. If further the inequality is strict on an interpolation point, that is* $C_L(f, t) < C_B(f, t)$*, then the set* T *has a non-empty interior.*

Proof. As in [7, 13], this begins as a simple consequence of the triangle inequality. Let A be the (nonnegative) matrix of change of basis from Lagrange to $B = LA$. Then since A, B and L are nonnegative on the interpolation points we have for every x_k

$$\sum_{j=0}^{n} |c_j B_j(x)| = \sum_{j=0}^{n} |c_j| B_j(x) = \sum_{i=0}^{n} \left(\sum_{j=0}^{n} |c_j| a_{ij} \right) L_i(x) \ge \sum_{i=0}^{n} \left| \sum_{j=0}^{n} c_j a_{ij} \right| L_i(x) \,.$$

Therefore T is not empty, containing at least all x_k.

If for some interpolation point, say x_k, the inequality is strict, then we observe that points near to x_k also belong to T, because all the terms in the inequality are continuous. This establishes that the set T has nonempty interior if the inequality is strict at any interpolation point.

Remark. The relative size of T compared to Ω is of immediate practical interest. In Fig. 7 we plot the sign of the difference $C_B(W_1,t) - C_L(W_1,t)$ for the random Lagrange basis used in the first example. The set T is exactly the set where this graph is nonnegative. Note that the set contains a large region around the interior interpolation points, but only a small region around each of the two points near the edge of the interval.

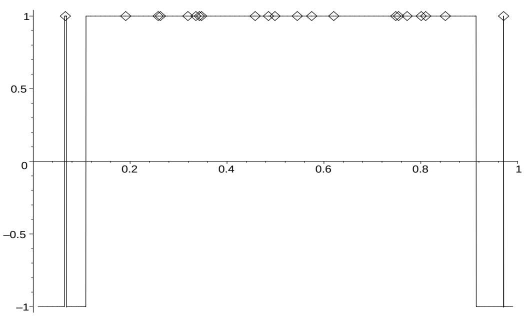

Fig. 7. The sign of the difference between the condition number in the Bernstein basis and in a Lagrange basis for the first Wilkinson polynomial. The set T where the Lagrange basis is better than the Bernstein basis is precisely the set of x-values where the sign is positive.

Proposition 2. *If we choose n of our $n+1$ interpolation points to be the roots, then $C_L(f,r) = 0$. That is, if we are lucky enough to interpolate at all the roots, the conditioning is perfect.*

Proof. This is a simple computation. The coefficients of the expansion in the Lagrange basis are, except for one coefficient (say y_0), all zero: $y_k = 0$ for $1 \leq k \leq n$. Therefore the expression for the condition of any x becomes $C_L(f,x) = |y_0 \ell_0 (x-r_1)(x-r_2)\cdots(x-r_n)|$, and this is obviously zero at each root r_k.

Remark. This implies that the convergence of the iteration that Fortune used [10] to find roots is superlinear.

4 Concluding Remarks

The Lagrange basis may sometimes become negative, and this may cause some numerical difficulty. The nonnegativity of the Bernstein basis is a genuine advantage. However, for rootfinding, the Lagrange basis may in practice be superior sometimes, if the interpolation points are 'close enough' to the roots. We have seen examples where this was so. We have also seen examples where if the interpolation points are far from the roots, then the roots are outside the set T on which the Lagrange basis is superior to the Bernstein basis. Sometimes the set T can be small; other times it can be surprisingly large.

The results of this paper offer some theoretical justification for using polynomials expressed directly by values for rootfinding, and suggest some strategies for selecting evaluation points (if that is possible).

The Lagrange basis is quite flexible. We may use it to find roots in a domain Ω from samples that 'represent well' the domain. The accuracy of the rootfinding degrades in areas that are not well-sampled, however. The results of this paper extend to any domain, because the Lagrange basis is nonnegative at the interpolation points. It is clear that any nonnegative basis in Ω may be written as a nonnegative combination of a Lagrange basis at interpolation points in Ω, and therefore on a (possibly small) set T containing the interpolation points the Lagrange basis will be optimal. [This is the analogue of the theorem from [13] stating Bernstein bases are optimal over all bases nonnegative on Ω, but here the proof is just that one sentence.]

One final observation is that we may *oversample*, and thereby cover the region of interest with enough points to be sure that we may accurately find all roots in the region. This is so because the condition of a root is improved dramatically even if only one interpolation point is nearby (all previous terms in the condition number formula become smaller, and the new one is small if y_{new} is small). In practice this will be limited by the extraneous roots at infinity becoming more and more multiple (sampling a degree 4 polynomial with 25 samples means that there are 20 extraneous eigenvalues at infinity, besides the two that are there naturally in this formulation).

We plan to look at the question of pejorative manifolds and multiple roots in a future paper. We also plan to investigate fast special-purpose methods to compute the eigenvalues of the companion matrix pencil.

Acknowledgements

Discussions on this topic with Laureano Gonzalez-Vega, Daniel Lazard, Erich Kaltofen, Amirhossein Amiraslani and Azar Shakoori were very helpful. This work was carried out with the support of the Natural Sciences Research Council of Canada.

References

1. Amirhossein Amiraslani. Dividing polynomials when you only know their values. In Laureano Gonzalez-Vega and Tomas Recio, editors, *Proceedings EACA*, pages 5–10, June 2004.
2. Amirhossein Amiraslani, Robert M. Corless, Laureano Gonzalez-Vega, and Azar Shakoori. Polynomial algebra by values. Technical Report TR-04-01, Ontario Research Centre for Computer Algebra, http:// www.orcca.on.ca/ TechReports, January 2004.
3. D. A. Bini and L. Gemignani. Bernstein-Bezoutian matrices. *Theor. Comput. Sci.*, 315(2-3):319–333, 2004.
4. Peter Borwein and Loki Jörgenson. Visible structures in number theory. Technical report, Centre for Experimental and Constructive Mathematics, http://www.cecm.sfu.ca/ loki/Papers/Numbers/, 1996.
5. Robert M. Corless. Generalized companion matrices in the Lagrange basis. In Laureano Gonzalez-Vega and Tomas Recio, editors, *Proceedings EACA*, pages 317–322, June 2004.
6. Robert M. Corless, Gaston H. Gonnet, D. E. G. Hare, David J. Jeffrey, and Donald E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5:329–359, 1996.
7. R. T. Farouki and T. N. T. Goodman. On the optimal stability of the Bernstein basis. *Math. Comput.*, 65(216):1553–1566, 1996.
8. R. T. Farouki and V. T. Rajan. On the numerical condition of polynomials in Bernstein form. *Comput. Aided Geom. Des.*, 4(3):191–216, 1987.
9. R. T. Farouki and V. T. Rajan. Algorithms for polynomials in Bernstein form. *Comput. Aided Geom. Des.*, 5(1):1–26, 1988.
10. Steven Fortune. Polynomial root finding using iterated eigenvalue computation. In Bernard Mourrain, editor, *Proc. ISSAC*, pages 121–128, London, Canada, 2001. ACM.
11. T. Hermann. On the stability of polynomial transformations between Taylor, Bézier, and Hermite forms. *Numerical Algorithms*, 13:307–320, 1996.
12. Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2002.
13. T. Lyche and J. M. Peña. Optimally stable multivariate bases. *Advances in Computational Mathematics*, 20:149–159, January 2004.
14. Azar Shakoori. The Bézout matrix in the Lagrange basis. In Laureano Gonzalez-Vega and Tomas Recio, editors, *Proceedings EACA*, pages 295–299, June 2004.
15. Brian T. Smith. Error bounds for zeros of a polynomial based upon Gerschgorin's theorem. *Journal of the Association for Computing Machinery*, 17(4):661–674, October 1970.
16. Yi-Feng Tsai and Rida T. Farouki. Algorithm 812: BPOLY: An object-oriented library of numerical algorithms for polynomials in Bernstein form. *ACM Trans. Math. Softw.*, 27(2):267–296, 2001.
17. Joab R. Winkler. The transformation of the companion matrix resultant between the power and Bernstein polynomial bases. *Appl. Numer. Math.*, 48(1):113–126, 2004.