The Nearest Polynomial of Lower Degree

Robert M. Corless and Nargol Rezvani

Ontario Research Centre for Computer Algebra,
Department of Applied Mathematics,
University of Western Ontario,
London, Ontario, N6A 5B7, Canada.

1 Introduction

Suppose one is working with the polynomial $p(x) = -0.99B_0^3(x) - 1.03B_1^3(x) - 0.33B_2^3(x) + B_3^3(x)$ expressed in terms of degree 3 Bernstein polynomials and suppose one has reason to believe that $p(x)$ is really of degree 2. Applying the degree-reducing procedure [6] one gets $-0.99B_0^2(x) - 1.05B_1^2(x) + B_2^2(x)$. But is this correct, or have we treated $p(x)$ in a Procrustean [1] fashion? Checking by converting $p(x)$ to the power basis, we find that the coefficient of x^3 is 0.11. Is this zero or not?

Using the results of section 6, below, we find that the nearest [2] polynomial of degree 2 is in fact $\hat{p}(x) = -1.003B_0^3(x) - 1.016B_1^3(x) - 0.343B_2^3(x) + 1.013B_3^3(x)$. Then applying the degree-reducing procedure to this polynomial we arrive at $p_3(x) = -1.003B_0^2(x) - 1.022B_1^2(x) + 1.013B_2^2(x)$. In this case, since $\|p - \hat{p}(x)\| \doteq 10^{-2}$, we have some confidence that the degree-reducing did not throw away important information; indeed, the process of finding $\hat{p}(x)$ amounts to a linear filter, removing noise. If $\|p - \hat{p}\|$ had been large, we would have known that degree-reducing was not valid.

Similarly, if we are given polynomial values on a grid, if is sometimes useful to find the values of the nearest polynomial of lower degree on the same grid, which again may filter noise.

[1] *Procrustes* was a mythical villain who had a bed that was 'just the right size' for his guest/victims; just the right size, that is, after Procrustes would chop off bits of too-tall guests, or stretch too-short guests on a rack.
[2] nearest in this case means that the vector of differences in the coefficients has the smallest ∞-norm.

One thing we wish to avoid is conversion to monomial basis, wherever possible. *Conversion between bases can be unstable*, and the instability increases with the degree [7]. As we saw in the example above, the information obtained by such conversions is sometimes hard to understand, anyway: we couldn't decide if 0.11 was noise or not. As in [6] we work in the Bernstein basis if we start in the Bernstein basis. Similarly as in [1] [2][4][11], if we start in the Lagrange basis we stay in the Lagrange basis.

Remark. One of the computations below, specifically in equation (7), in effect computes the leading coefficient in the monomial basis. This is the only such coefficient computed, and moreover it is computed in a way that gives a natural scaling in terms of the *distance* to the nearest polynomial of lower degree. This is the contribution of the present paper.

2 Notation

Recall that the Bernstein basis is defined as

$$B_k^{(n)}(x) = \binom{n}{k}(x-a)^k(b-x)^{(n-k)}, \qquad a \leq x \leq b \tag{1}$$

We will usually take $(a,b) = (0,1)$.

For the Lagrange basis, recent papers by Berrut and Trefethen [3] and Higham [8] show that one may stably write the polynomial $p(x)$ that takes on the values p_k at the distinct nodes x_k in the *barycentric form*:

$$p(x) = \ell(x)\sum_{k=0}^{n} \frac{w_k p_k}{(x-x_k)} \tag{2}$$

where $\ell(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$ and the barycentric weights are $w_k = \prod_{j\neq k} \frac{1}{(x_k-x_j)}$. They show that this formula is *numerically stable*. In this paper, for representing the polynomial in the Lagrange basis, we use the barycentric form.

The metric used in [5] to compare two polynomials expressed in the Bernstein basis on $[0,1]$ is $d(f_1,f_2) = \|f_1 - f_2\|_I$ where

$$\|f\|_I^2 := \int_0^1 |f(x)|^2 \, dx \, ,$$

and where $f(x) = \sum_{k=0}^{n} C_k B_k^n(x)$. In contrast, the paper [9] uses vector norms on the coefficients:

$$\|f\|_p := \left(\sum_{k=0}^{n} |C_k|^p \right)^{1/p}$$

if $p < \infty$ and as usual $\|f\|_\infty = \max(|C_k|)$.

Lemma 1 *These norms are equivalent.*

Proof. We show that there exist constants γ_n such that

$$\|f\|_\infty \geq \|f\|_I \geq \gamma_n \|f\|_\infty \, ,$$

and then from the equivalence of the p-norms the result follows. Let C_m be a coefficient for which $\|f\|_\infty = |C_m|$ (there may be more than one). Then

$$\begin{aligned}
\int_0^1 |f(x)|^2 \, dx &= \int_0^1 \overline{f}(x) f(x) \, dx \\
&= \int_0^1 \left(\overline{C_m} B_m^n(x) + \cdots \right) \left(C_m B_m^n(x) + \cdots \right) \, dx \\
&\geq \overline{C_m} C_m \int_0^1 \left(B_m^n(x) \right)^2 \, dx \\
&\geq \|f\|_\infty^2 \gamma_n^2
\end{aligned}$$

where γ_n^2 is the integral of $B_j^n(x)^2$ that is smallest over all j (this happens for $j = \lfloor n/2 \rfloor$, and a short calculation shows $\gamma_n \sim (\pi n)^{-3/4}$). The other direction is simpler: for $C_k \in \mathbb{C}$,

$$\|f\|_I^2 \leq \int_0^1 \left| \sum_{k=0}^{n} |C_k| B_k^n(x) \right|^2 \, dx \leq \int_0^1 \|f\|_\infty^2 \left| \sum_{k=0}^{n} B_k^n(x) \right|^2 \, dx = \|f\|_\infty^2$$

because the Bernstein polynomials add to 1, and so we see that making the vector norm small also makes the integral norm small.

One can similarly prove that $\|f\|_I \leq \|f\|_1$.

3 Degree Operations in Bernstein Basis

As is well-known given $p(x) = \sum_{k=0}^{n} C_k^{(n)} B_k^{(n)}(x)$, if $p(x)$ truly is of lower degree, one can also express $p(x)$ in terms of Bernstein polynomials of lower degree, $p(x) = \sum_{k=0}^{n-1} C_k^{(n-1)} B_k^{(n-1)}(x)$. This is called *degree reduction* [6]. The process uses the recurrence in Lemma 2 below. This only works if $p(x)$ really has degree $n-1$ or less.

Given a polynomial $p(x)$ that has been computed by some process involving *errors*, $p(x)$ may not be of degree $n-1$, but only close to such a polynomial. The purpose of our paper is to introduce a procedure that can be used to guarantee lower degree, before the formal process of degree reduction is carried out.

The next lemma slightly generalizes those in section 3.2 of [6].

Lemma 2
$$C_k^{n+1} = \begin{cases} C_0^n & \text{if } k = 0 \\ \frac{1}{b-a}\left[\frac{k}{n+1} C_{k-1}^n + (1 - \frac{k}{n+1}) C_k^n\right] & \text{if } 1 \le k \le n \\ C_n^n & \text{if } k = n+1 \end{cases} .$$

Proof: exercise. **Remark.** This recurrence relation can be solved for the vector C_k^n given the vector C_k^{n+1}. This is what is meant by 'degree reduction'.

4 Summary of "Nearest Polynomial with a Given Zero, revisited"

Suppose $f \in \mathbb{P}$ where

$$\mathbb{P} \cong \mathbb{C}^{n+1}[x] = \mathrm{span}\{\phi_0(x), \phi_1(x), \ldots, \phi_n(x)\} =: \mathbf{\Phi}(\mathbf{x}) \tag{3}$$

where the $\phi_k(x)$ form a basis for the set of polynomials of degree less than $n+1$ with complex coefficients. Given the q-norm for measuring the distance between polynomials and a complex number r that is the desired zero, the problem of finding $\tilde{f} \in \mathbb{P}$ such that $\tilde{f}(r) = 0$ and $\|f - \tilde{f}\|_q$ minimal was studied in [9]. Starting with the unit vector

$$\mathbf{u} = \frac{\mathbf{\Phi}(r)}{\|\mathbf{\Phi}(r)\|_p} , \tag{4}$$

where $p = q/(q-1)$ and, using the converse of Hölder's inequality, an explicit formula for computing minimal perturbations was introduced. This same approach gives the nearest polynomial of lower degree, when we let $r \to \infty$.

5 Lower Degree

5.1 Degree-graded Bases

In this simple case, all we need to do is force the leading coefficient to be zero. This approach is used without comment in many places (for example, in Chebyshev economization [10]).

Lemma 3 *If* $f(x) = \sum_{k=0}^{n} c_k \phi_k(x)$, $\deg \phi_k = k$, *then the nearest polynomial of lower degree to* f *in any* p*-norm is* $\tilde{f}(x) = \sum_{k=0}^{n-1} c_k \phi_k(x)$.

Proof: exercise.

5.2 Lagrange Basis

The vector of leading coefficients of the Lagrange polynomials is proportional to the vector of the barycentric weights, $[w_0, w_1, \ldots, w_n]^T$, the limit of $\mathbf{u}$ from equation 4 as $r \to \infty$. We thus enforce lower degree by the linear constraint $\sum_{j=0}^{n} w_j f_j = 0$. This is mathematically equivalent to setting the leading coefficient of f in the monomial basis $\sum_{j=0}^{n} w_j f_j = \frac{f^{(n)}(0)}{n!}$ to zero.

5.3 Bernstein Basis

The vector of leading coefficients will be $[(-1)^0\binom{n}{0}, \cdots, (-1)^k\binom{n}{k}, \cdots, (-1)^n\binom{n}{n}]$. After normalization, this vector is the limit of $\mathbf{u}$ from equation 4 as $r \to \infty$. As before, we enforce lower degree by the linear constraint $\sum_{k=0}^{n} (-1)^k \binom{n}{k} C_k = 0$.

6 Algorithm of NPGZ (Nearest Polynomial with a Given Zero)

Input: Polynomial $f = \mathbf{a} \cdot \mathbf{\Phi}(x) \in \mathbb{P}$, where $\mathbf{a}$ is the vector of the coefficients and $\mathbf{\Phi}$ is the basis in question, a given zero r (infinity in our case) and the desired q-norm for minimality.

Output: The perturbed polynomial $\tilde{f}$ such that $\|f - \tilde{f}\|_q$ is minimal and $\tilde{f}(r) = 0$.

Steps:

I) Compute $p = q/(q-1)$, or $p = \infty$ if $q = 1$.

II) For $|r| < \infty$ then take $\mathbf{u} = \frac{\mathbf{\Phi}(r)}{\|\mathbf{\Phi}(r)\|_p}$. Otherwise take its limiting value as $r \to \infty$, which gives normalized versions of the vectors of leading coefficients mentioned in the previous section as the vector $\mathbf{u}$.

III) Compute the minimal perturbations:

- For $|r| < \infty$ then

$$\mathbf{\Delta a} = \frac{|f(r)|}{\|\mathbf{\Phi}(r)\|_p} \cdot \mathbf{v} \, . \tag{5}$$

where $\mathbf{v}$ is defined below. Taking $r \to \infty$ we take (note $\mathbf{v}$ is eventually constant), and using l.c.$f(x)$ to denote the leading coefficient of $f(x)$ in the monomial basis,

$$\mathbf{\Delta a} = \lim_{r\to\infty} \frac{|f(r)|}{\|\mathbf{\Phi}(r)\|_p} \cdot \mathbf{v} = \frac{|\text{l.c.}f(x)|}{\|\text{l.c.}\mathbf{\Phi}(x)\|_p} \cdot \mathbf{v} \, . \tag{6}$$

Note that these leading coefficients are known, for $\mathbf{\Phi}$, or are expressed as scalar products with the computed vector $\mathbf{u}$:

$$\text{l.c.}f(x) = \mathbf{C} \cdot (\text{l.c.}\mathbf{\Phi}) \, . \tag{7}$$

The coefficients v_k of $\mathbf{v}$ are defined by the conditions for the witness vector in Hölder's inequality, and depend on $\mathbf{u}$:

- if $1 \le p \le \infty$ then

$$v_k = \begin{cases} -\text{signum}(f(r))|u_k|^{p-2}\bar{u}_k & \text{if } u_k \neq 0 \\ 0 & \text{if } u_k = 0 \end{cases} \tag{8}$$

- if $p = \infty$ then

$$v_k = \begin{cases} 0 & \text{if } k \neq k_0 \\ -\text{signum}(f(r))\bar{u}_{k_0} & \text{if } k = k_0 \end{cases} \tag{9}$$

Note that

$$\lim_{r\to\infty} \text{signum}(f(r)) = \text{signum}(\text{l.c.}f(x)) \tag{10}$$

is ultimately constant for large enough r.

IV) Return the polynomial $\tilde{f}(x) = (\mathbf{a} + \Delta\mathbf{a}) \cdot \mathbf{\Phi}(x)$.

7 A Lagrange basis example

Suppose that we are given the values $\mathbf{f} = [-26/27, 8/9, -22/27, 28/27]$ of a polynomial on the nodes $[-1, -1/3, 1/3, 1]$. Suppose also that we apply Newton's iteration (just once, to exaggerate the effect)

$$x_{n+1} = x_n + \frac{\sum_{k=0}^{3}(w_j f_j)/(x_n - z_j)}{\sum_{k=0}^{3}(w_j f_j)/(x_n - z_j)^2}$$

to the barycentric rational function $R(z) = f(z)/\ell(z)$, starting from an initial guess $x_0 = 0$, and find the root $x^* \approx 0.01234567903$. We *deflate* the polynomial by dividing each f_j by $(x^* - z_j)$, giving a new vector of values, $\mathbf{g} = [-.951220, 2.57143, 2.53846, -1.05]$. Obviously this is supposed to be of lower degree, and so we may throw away one data point. But as is well known for the monomial basis, deflation may introduce errors, and we might be better off by first applying the NPLD (Nearest Polynomial of Lower Degree) process. When we do this, we find $\hat{\mathbf{g}} = [-.951204, 2.57141, 2.53848, -1.050015]$, which differs from the first by about $1.5 \cdot 10^{-5}$, and now we may throw away any data point without fear of loss of information because up to one ulp these are the values of a degree 2 polynomial on this set of nodes.

8 Weighted norms

We may choose to apply different weights to different coefficients in the NPLD process. One useful choice is the set of dual weights $W_k = |C_k|$, the absolute values of the coefficients. If $C_k = 0$, this process ensures that the resulting coefficient in the lower degree polynomial is also zero. The procedure will find the lower degree polynomial nearest in this weighted norm. This ensures small *relative* changes in the coefficients. For details, see [9].

For example, suppose the coefficients of a degree 5 Bernstein expansion on $[0, 1]$ are $[0., 50.097, -25.006, -25.200, 50.214, 0.]$. We suspect this is close to a degree 4 polynomial. If we simply apply the degree-reducing procedure, we get the vector $[0., 62.621, -83.423, 62.135, 0.]$. If we first apply NPLD, with weights given by the absolute values of the coefficients, then after degree-reduction we get $[0., 62.621, -83.423, 62.767, 0.]$. These results do not differ much, but they do differ, and indeed the original unfiltered polynomial has a term $2.5 \cdot x^3$ (after conversion to the monomial basis, which we avoid) which may not appear negligible unless carefully considered.

9 Numerical Stability

The barycentric weights may vary widely in magnitude, in which case it is known that interpolation is very sensitive to data errors. Likewise, the elements of the vector of leading coefficients $(-1)^k\binom{n}{k}$ for the Bernstein case also vary widely in magnitude, and the reader ought to be concerned for the numerical stability of this process.

The one-row matrix formed by $B = [(-1)^k\binom{n}{k}]$ has only one singular value σ_1, and $\sigma_1 = O(2^n)$. Thus, errors in coefficients may be amplified by as much as 2^n on multiplication by this vector. One expects, in fact, that $Bv \approx 2^n\varepsilon$. But we are actually using the reverse of this process: we are looking for a vector of deltas that satisfy $B\Delta(f) = 0$, and hence we might expect to damp errors.

What really helps, though, is the fact that we have an analytic solution to this minimization problem. The computation of the components of $\mathbf{v}$ involves only powers, absolute values, multiplication, and division, and tests for zero. The size of the resulting $\Delta(f)$ is proportional to the residual $f(r)$ (for a finite root), and this may be subject to large relative errors if the residual is small, but we are going to add this change to f, and hence the effect of the large errors is muted.

In the case $r \to \infty$, the size of the Δ is proportional to the dot product (7). If this is large, then it will be accurately computed; if it is small compared to $\|f\|$, then though it will not necessarily have high relative accuracy its effect will be small when added to f.

10 Concluding Remarks

In the case of the Lagrange basis, we may enforce several degree reducing constraints at once, to reduce from degree n to degree $n-k$. In this case, $k > 1$, we no longer have an explicit formula, but need to solve a constrained optimization problem (e.g. least squares if $p = 2$). Alternatively, we may reduce the degree one at a time, although of course this does not give the closest polynomial of degree $n-k$ for $k > 1$.

This procedure is likely to be most useful on polynomials of moderate degree, where the degree is expected to be reduced only a few times. Fitting a large amount of data by a low-degree polynomial, in contrast, is likely to be more efficient working from the 'bottom up'.

The use of NPLD prior to degree reduction is efficient (the cost is $O(n)$ flops via the explicit formula) and numerically stable. Since this approach will prevent Procrustean errors, we recommend its use in general, on polynomials of moderate degree, to increase confidence in the computed results of degree reduction.

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] Jean-Paul Berrut and Lloyd N. Trefethen. Barycentric Lagrange interpolation. *SIAM Review*, 46(3):501–517, 2004.

[4] 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.

[5] 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.

[6] R. T. Farouki and V. T. Rajan. Algorithms for polynomials in Bernstein form. *Comput. Aided Geom. Des.*, 5(1):1–26, 1988.

[7] T. Hermann. On the stability of polynomial transformations between Taylor, Bézier, and Hermite forms. *Numerical Algorithms*, 13:307–320, 1996.

[8] Nicholas J. Higham. The numerical stability of barycentric Lagrange interpolation. *IMA Journal of Numerical Analysis*, 24:547–556, 2004.

[9] Nargol Rezvani and Robert M. Corless. The nearest polynomial with a given zero, revisited. SIGSAM BULLETIN, *Communications on Computer Algebra*, 134(3):71–76, September 2005.

[10] T. Rivlin. *Chebyshev polynomials: from approximation theory to number theory.* Wiley, 1990.

[11] Azar Shakoori. The Bézout matrix in the Lagrange basis. In Laureano Gonzalez-Vega and Tomas Recio, editors, *Proceedings EACA*, pages 295–299, June 2004.