Polynomial Algebra by Values

Robert M. Corless
ORCCA & Dept. Applied Mathematics
University of Western Ontario
London, Canada

1. Introduction & Properties of various polynomial bases

2. Nearest polynomial with a given zero, weighted p-norm

3. Bézout Matrices in the Lagrange Basis

4. Nearest singular polynomial

5. Lots more work to do

Introduction

1 The sensitivity of a problem can depend on its *representation*. Incautious change of basis may take a well-conditioned problem into an ill-conditioned one. [Any algorithm that does so is numerically unstable.]

2 Representing a polynomial in one basis rather than another may be mathematically equivalent but numerically superior, for some purposes. For example, $f(x) = x^2 + 3x + 2 = -10000000199999999 + 200000005\,x + (x - 100000001)^2$ but trying to find the roots of the second form, using 14 decimal digits of precision, yields no correct figures, even though the rounding of the input is equivalent to an error of only one part in 10^{16}.

3 The conditioning of a problem only makes sense with respect to perturbations or errors in primary data.

This series of papers (and today's talk) looks at working with polynomials (univariate and multivariate) given only by sampled values. *We do not change bases*, but work with the primary data (equivalently in the univariate case, with the *Lagrange basis*).

References can be found at
http://www.apmaths.uwo.ca/~rcorless/frames/PAPERS/PABV

Representation of polynomials in different bases

We begin with univariate polynomials. Let $\mathbb{P}^n$ be the space of polynomials in x of degree at most n with complex coefficients. Then as usual $p(x) \in \mathbb{P}^n$ may be expressed as a linear combination of a convenient basis $\phi_k(x)$ $(0 \leq k \leq n)$ of $\mathbb{P}^n$ by

$$p(x) = \sum_{k=0}^{n} c_k \phi(x)$$

The monomial (Taylor) basis $\phi_k(x) = (x - \alpha)^k$ is of course the most familiar, but there are an infinite number of others: Bernstein, Lagrange, Newton (divided difference), and of course orthogonal polynomial bases such as Chebyshev, Legendre, and others.

If the c_k are $s \times s$ matrices, then this is a *matrix polynomial*.

Question implicit assumptions: n versus $n+1$
Notice that $n+1$ coefficients are used. "Of course" we may insist on polynomials being *monic*, which reduces that to n. Here are some reasons that's a bad idea.

1. If the leading coefficient is a singular matrix, this is not possible at all.

2. Even if c_n is a nonzero scalar, making things monic affects the metric and may turn a linear problem into a nonlinear problem.

Metric? Isn't this a conference in algebra? What's a metric doing here? We will return to this question.

Most familiar bases are *degree-graded*: $\deg \phi_k(x) = k$, and are independent of n. However, both the Bernstein bases and the Lagrange bases do depend on n, and are not degree-graded.

In particular, the degree of a polynomial expressed in either the Lagrange or the Bernstein bases may not be immediately evident.

Example: Consider the polynomial $p(x) = 4x^3 - 3x$ expressed using the monomial basis. This can be expressed in the Lagrange basis on the distinct nodes $x_0 = -1$, $x_1 = -\frac{1}{2}$, $x_2 = \frac{1}{2}$, and $x_3 = 1$ as (the values $p_k = p(x_k)$ at the nodes are -1, 1, -1, and 1)

$$p(x) = \ell(x) \sum_{k=0}^{3} \frac{w_k p_k}{x - x_k} = -L_0(x) + L_1(x) - L_2(x) + L_3(x)$$

where the L_k are the *Lagrange polynomials* on the given nodes:

$$L_k(x) = w_k \prod_{j \neq k} (x - x_j) \,,$$

the barycentric weights w_k are

$$w_k = \frac{1}{\prod_{j \neq k}(x_k - x_j)}$$

and $\ell(x) = (x+1)(x+1/2)(x-1/2)(x-1)$ is zero on all the nodes.

Barycentric form (Berrut & Trefethen 2004, Higham 2004)

The first form written above,

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

is called the barycentric form of the interpolant. Once the barycentric weights w_k are computed, which takes $O(n^2)$ work (independent of the polynomial values), then the polynomial $p(x)$ can be evaluated stably in $O(n)$ time (Higham, 2004).

Remark. We assume henceforth that the nodes $\{x_k\}$ are distinct. We have some formulae for the confluent case but they add only complication to this presentation.

Nearest polynomial with a given zero, Revisited

1. A recent paper with Nargol Rezvani, with more material yet in her Masters' thesis (Dec. 2005), fills in some details of a paper of H.J. Stetter (Sigsam Bulletin, 1999) showing an explicit solution of this problem, using a witness vector for the Hölder inequality (and the converse of the Hölder inequality) as the theoretical basis for the solution of the problem (dual norms).

2. Extensions in the thesis: arbitrary polynomial bases, including the Lagrange basis; weighted p-norms and their duals.

3. One more extension (new to this talk): The nearest polynomial of lower degree. This question makes most sense in the Lagrange or Bernstein basis.

We state the problem in four forms. We suppose that we are given $f \in \mathbb{P}^n$ where

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

where the $\phi_k(x)$ form a basis for the set of polynomials of degree less than $n+1$ with complex coefficients. We also suppose that we are given a metric $d : \mathbb{P}^n \times \mathbb{P}^n \longrightarrow \mathbb{R}^+ \cup \{0\}$ measuring the distance between polynomials, and given a complex number r that is the desired zero.

Then the problem may be stated as one of the following:

1. Find $\tilde{f} \in \mathbb{P}^n$ with $\tilde{f}(r) = 0$ and $d(f, \tilde{f})$ minimal.

2. Find $\tilde{f}$ as above but also insist that $\deg(f) = \deg(\tilde{f})$.

3. Given a *monic* f (in a basis for which this makes sense), find *monic* $\tilde{f}$ as above.

4. Given a polynomial with some *intrinsic* (i.e. fixed) coefficients, and some *empiric* (i.e. contaminated by data error) coefficients, find the nearest $\tilde{f}$ with the same intrinsic coefficients and $\tilde{f}(r) = 0$.

The solution to the third problem was given in a lecture by E. Kaltofen in August 1999, based on work by M. Hitz. It was further studied in Stetter's 1999 paper in the case $\phi_k(x) = x^k$, the monomial basis, and $d(f_1, f_2) = \|f_1 - f_2\|_p$, the p-norm of the vector of coefficients, $\mathbf{v}$, of the difference between f_1 and f_2.

Consider the weighted p-norm with the following definition.

$$\|\mathbf{u}\|_{p,w} = \begin{cases} \left(\sum_{i=0}^{n} |w_i u_i|^p\right)^{1/p} & \text{if } p \neq \infty \\ \max_{0 \leq i \leq n} w_i |u_i| & \text{if } p = \infty \end{cases} \tag{1}$$

where $\mathbf{w} = [w_0, w_1, \cdots, w_n]$ is the vector of positive weights.

It is essential that the weights go *inside* the power p. Otherwise, our results are not continuous in p.

Hölder's inequality (complex case)

$$\left|\sum_{k=0}^{n} a_k b_k\right| \leq \left(\sum_{k=0}^{n} |a_k|^p\right)^{1/p} \left(\sum_{k=0}^{n} |b_k|^q\right)^{1/q}$$

with equality if and only if there exist constants λ and C such that

$$\lambda |a_k|^{1/q} = |b_k|^{1/p}$$

and

$$\arg a_k b_k = C$$

whenever this argument is defined (i.e. when both of a_k and b_k are nonzero).

The *converse* of this inequality states that if there exists K such that

$$\left|\sum_{k=0}^{n} a_k b_k\right| \leq K \|\mathbf{b}\|_p$$

for all $\mathbf{b}$, then $\|\mathbf{a}\|_q \leq K$. The *witness vector* for equality (that occurs when $\lambda|a_k|^{1/q} = |b_k|^{1/p}$, together with the condition on the argument) will provide the vector of coefficients for our globally nearest polynomial.

Dual of a weighted p-norm

It can be shown (by rescaling and using Hölder's inequality and its converse) that the dual of the weighted p-norm is

$$\|\mathbf{v}\|_{p,w}^{*} = \|\mathbf{v}\|_{q,w^*}$$

where q is given by $1/p + 1/q = 1$ and $w^* = (w_0^{-1}, \cdots, w_n^{-1})$ and if any $w_i = 0$, then unless $v_i = 0$ as well, $\|\mathbf{v}\|_{p,w}^{*} = \infty$. That is, the subordinate dual norm

$$\|\mathbf{v}\|_{p,w}^{*} := \max_{\|\mathbf{u}\|_{p,w}=1} |\mathbf{v} \cdot \mathbf{u}|$$

can be computed by the formula above. Moreover, the witness vector for the Hölder inequality can be used to solve minimization problems.

A note on input of weights

We felt that it would be confusing to force a user to specify $w_i = \infty$ to force a zero change in the ith coefficient. Therefore, our implementation asks the user to specify the *dual* weights, with zeros corresponding to where the user wants no change in the coefficient.

Surprises.

1. The polynomial having a zero at $z = r$ and being nearest to the given polynomial might be identically zero: $f(z) = 1 + z + z^2$, $p = \infty$, $r = 1/2$, for example.

2. Insisting that the perturbed polynomial have the same degree makes the problem (in some cases) fail to have a solution at all. There is no degree two polynomial that has a zero at $z = 1/2$ that is *closest* to $f(z)$ above: we can get arbitrarily close to the zero polynomial. This was observed already by Hitz, Kaltofen & Lakshman (ISSAC 1999).

3. Insisting that the resulting polynomial be monic allows a solution, but potentially changes the problem (the metric is different); and moreover this does not always make sense in the case of the Lagrange basis or the Bernstein basis.

4. The ambiguity of the choice of k_0 in the event that more than one u_k has unit magnitude allows for the possibility of multiple optima, in the ∞-norm case.

5. Defining a weighted norm which is continuous in p and applying it for insisting on sparsity or monicity, and allowing the weight vectors to specify which coefficients change and which remain the same, leads us to some interesting results.

6. The cost of the routine NPGZ is linear in the number of nonzero coefficients, and is thus efficient.

The Bezout matrix in the Lagrange basis

This was reported on briefly by my student Azar Shakoori at EACA 2004 in Santander.

Suppose $f(x)$ and $g(x)$ are given on the distinct nodes $[x_0, x_1, \ldots, x_n]$, and each are degree n. Then one may form the $n+1$ by $n+1$ matrix

$$B_{ij} = \begin{cases} \frac{f(x_i)g(x_j) - f(x_j)g(x_i)}{x_i - x_j} & i \neq j \\ f'(x_i)g(x_i) - f(x_i)g'(x_i) & i = j \end{cases}$$

One needs to compute derivatives, for example by the formula

$$f'(x_i) = \frac{-1}{w_i} \sum_{j=0, j\neq i}^{n} \frac{w_j(f_i - f_j)}{x_i - x_j},$$

but this is no hardship.

Then each common root r of $f(x)$ and $g(x)$ gives a vector $[L_0(r), L_1(r), \ldots, L_n(r)]^T$ in the null space of B (multiple roots give null vectors that are evaluations of the derivatives of the Lagrange polynomials).

Moreover, because the polynomials are degree n, this $n+1$ by $n+1$ matrix also has a null vector corresponding to the *common root at infinity*, namely the vector of leading coefficients of the Lagrange polynomials as $r \to \infty$:

$$[w_0, w_1, \ldots, w_n]^T$$

which are the barycentric weights (depending only on the nodes).

Dependence on the nodes has not been included in the notation.

We could say $w_k(x_0, x_1, \ldots, x_n)$, or for short $w_k^{(n)}$, and similarly $L_k(x; x_0, x_1, \ldots, x_n)$.

Theorem If $f(x)$ has degree $m \leq n$ then each principal minor of B of order $k+1$ $(m \leq k \leq n)$ has a null vector of the form

$$[w_0^{(k)}, w_1^{(k)}, \ldots, w_k^{(k)}]\,.$$

Proof Recursive descent.

Example. Suppose $f(x) = [-1, 1, -1, 1]$ on the nodes $[-1, -1/3, 1/3, 1]$, and $g(x) = [1, 1, 1, 1]$ is identically 1. Then the Bezout matrix of these two polynomials is

$$\begin{bmatrix} 10 & 3 & 0 & 1 \\ 3 & -2 & -3 & 0 \\ 0 & -3 & -2 & 3 \\ 1 & 0 & 3 & 10 \end{bmatrix}$$

and the vector of barycentric weights is $[-\frac{9}{16}, \frac{27}{16}, -\frac{27}{16}, \frac{9}{16}]$. It is easy to see that this is a null vector of this matrix.

Moreover, the principal 3 by 3 minor is nonsingular; hence the polynomial f is of degree 3.

Nearest polynomial of lower degree

Note that

$$\sum_{j=0}^{n} w_j f_j = f^{(n)}(x)$$

is a constant, being the nth Taylor coefficient of the degree n interpolating polynomial

$$f(z) = \ell(z) \sum_{j=0}^{n} \frac{w_j f_j}{z - x_j} .$$

Setting this constant to zero forces the degree to drop. This is equivalent to forcing the polynomial to have a *given zero* at infinity. The same Hölder inequality approach used for finite zeros works here as well. (Not yet implemented: we just do this "by hand").

Aren't these finite differences? Aren't they dangerous? Differentiation is an *ill-posed* problem. Even if $f(z)$ and $g(z)$ are very close, their derivatives $f'(z)$ and $g'(z)$ may be arbitrarily far apart. Therefore approximate differentiation may easily reveal errors made in evaluating f.

But here, we are *using finite differences to enforce constraints*, and in effect using D^{-1} in computation. This is a *smoothing* operation.

Example. Consider again $f(x) = [-1, 1, -1, 1]$ on $x = [-1, -1/3, 1/3, 1]$. Take $p = 2$, hence $q = 2$. Then the nearest polynomial in the 2 norm that is of lower degree is $\tilde{f}(x) = [-3/5, -1/5, 1/5, 3/5]$, which is $f(t) = 3t/5$ in the monomial basis. By virtue of symmetry, we have actually lowered the degree by two.
If $p = \infty$, then $\tilde{f}(x) = [0, 0, 0, 0]$, identically zero. If $p = 1$, then $\tilde{f}(x) = [-1, -5/3, -1, 1]$, corresponding to $f(t) = 3(t^2 - 1)/2 + t$, degree two. There is another optimal solution that is symmetric to this one, altering the 3rd value, not the 2nd, gives us $[-1, 1, 5/3, 1]$ or $f(t) = -3(t^2 - 1)/2 + t$.

Nearest polynomial of degree k

This problem is similar, but without an explicit solution: Given a metric $d(f,g)$ and a polynomial f, find $\tilde{f}$ with

$$\min d(\tilde{f}, f)$$

and

$$\begin{bmatrix} w_0^{(k)} & w_1^{(k)} & \cdots & w_k^{(k)} & & & \\ & w_0^{(k)} & w_1^{(k)} & \cdots & w_k^{(k)} & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & w_0^{(k)} & w_1^{(k)} & \cdots & w_k^{(k)} \end{bmatrix} \begin{bmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_n \end{bmatrix} = 0$$

which, if d is a nice metric (say difference of weighted 2-norms), gives a linear system for the $\tilde{f}_k$. Notice the barycentric weights in each row differ (the notation does not indicate the nodes used).

Nearest singular polynomial

Nargol Rezvani is currently working on this. So far we have 'proof-of-concept': we have adapted the Karmarkar-Lakshman approach, and have used this to successfully solve *one* problem, in the Lagrange basis.

Multivariate at last
What we have done:

1. Used bivariate Bézout matrices in the Lagrange basis on rectangular (separable) grids to solve some bivariate systems. (Shakoori)

2. Used multivariate division to set up some resultant-like matrices that we have then used to solve some systems of equations by the eigenvalue method. (Amiraslani)

3. Investigated faster methods for solving the companion-like eigenproblems that arise (Amiraslani, Aruliah). The methods are variants on Rayleigh Quotient Iteration.

What we have yet to do:

1. Move to general minimal-degree multivariate interpolants. (Sauer & co-authors, Olver (2006), de Boor, Stetter, ...)

2. Understand how to set up the matrices that enforce degree constraints on a general irregular (though poised) grid.

3. Understand the numerical stability issues

4. Understand the effect of the geometry of the nodes.

Tasks for the workshop

1. (with E. Kaltofen) Formulate the Ruppert matrix in the Lagrange basis, for factorization

2. (with H.-M. Möller and T. Sauer) Look at the use of division to construct multiplication matrices directly from the data, in the non-separable case, Look at interpolation and uniqueness of derivatives (for degree constraints)

3. (with L. Zhi) Look at improvements in the nearest singular polynomial algorithms.

Thank You for your attention.

Acknowledgements.
Many people have contributed to my understanding of this area, including (but not limited to) My students: Amirhossein Amiraslani, Ashley Pitcher, Nargol Rezvani, Azar Shakoori, Jichao Zhao. My coauthors: Dhavide Aruliah, Mark Giesbrecht, Laureano Gonzalez-Vega, David Jeffrey, Erich Kaltofen, Stephen Watt, Lihong Zhi. Finally, there are many people whose works I have read or heard and appreciated: Jim Demmel, Nick Higham, William Kahan, Hans-Michael Möller, Peter Olver, Hans J. Stetter, Tomas Sauer, Nick Trefethen, Zhonggang Zeng, and many others. I also thank the organizers of this workshop for the chance to participate.