Polynomial Algebra by Values

Amirhossein Amiraslani Robert M. Corless
Laureano Gonzalez-Vega Azar Shakoori

January 2004

1 Introduction

This paper outlines some new algorithms for operations on multivariate polynomials. The point of view taken in the paper is that all polynomials considered are given by their values at known points, and that the degrees of the polynomials are known or can be deduced.

Of course, in this situation we may in principle apply an interpolation algorithm [5, 9] and convert to a monomial basis, and then use standard algorithms for all purposes, and at the end evaluate the result on the desired grid. It is the point of view of this paper that we choose not to carry out these conversions, but rather to perform all operations directly on the values on the grid. See Figure 1.

Direct methods have some disadvantages, notably the density of the representation and the resulting complexity of the algorithms. However, in the approximate polynomial case, the conversion we avoid is usually ill-conditioned,

$$\begin{array}{ccc} P & \dashrightarrow & S \\ \Downarrow & & \Uparrow \\ P_M & \Rightarrow & S_M \end{array}$$

Figure 1: This paper attempts the dashed road. The symbols P, S, P_M, and S_M stand for the Problem to be solved in an arbitrary basis, the Solution in the same basis, the Polynomial after conversion to Monomial basis, and the Solution in the Monomial basis, respectively.

and thus by avoiding conversion we may possibly increase the numerical stability of our algorithms as a whole.

We also give an apparently novel method to solve systems of polynomial equations, by giving a novel construction of eigenvalue problem that contain the (zero-dimensional) variety in the spectrum, without first computing Gröbner bases or resultants. The method is based on multivariate division. This method may or may not be competitive with standard methods; its advantage is that it avoids Gröbner basis computation, but its disadvantage, like that of resultants, is that it introduces extraneous roots.

The viewpoint of the paper is that the polynomial values given are floating point numbers and hence inexact. The precise amount of inexactness is left vague for the moment. The constructions of the paper will, however, carry over (albeit relatively inefficiently) to the exact arithmetic case.

2 Notation

We assume that the polynomials are given as values on a common rectangular grid $x_{0,k} < x_{1,k} < \cdots < x_{N_k,k}$ for $1 \le k \le s$. The subscript notation $p_{i_1,i_2,\ldots,i_s}$ means $p(x_{i_1,1}, x_{i_2,2}, \ldots, x_{i_s,s})$. There are $N = N_1 N_2 \ldots N_s$ points. In the common case of two variables we will use x_j, y_j instead of $x_{j,1}$ and $x_{j,2}$.

Unless otherwise stated, we assume that the degrees of the given polynomials are also known. If necessary, one may compute the degree (or a suitable degree) by fitting a polynomial to the data.

3 Operations Considered

The operators $p \pm q$, pq and p^m are trivial to define: we perform these directly on the N values as expected, and the cost is $O(N)$. Note that in the case of $+$ and $-$ the degree cannot increase, but in the case of multiplication and powering it may.

The paper is organized as follows: we show how to compute derivatives of polynomials given by values in section 4. We then show how to construct the Bézout matrix for polynomials given by values, in section 5. We outline a method to use the Bézout matrix via Barnett's Theorems to compute the GCD in section 6. We then show how to solve bivariate systems given by values by conversion of the Bézout matrix to a generalized eigenproblem. In that section we also give a new construction of the generalized companion matrix for (matrix) polynomials given by values that is simpler and more efficient than that of Smith [7] or [2].

We then move on to division of polynomials, and the use of division. In section 8 we cover the univariate case, and extend it to the multivariate case in section 9. In section 10 we use these results in a new algorithm to solve multivariate systems of polynomial equations given by values on a rectangular grid.

We end the paper with a brief discussion of the problems remaining: factoring, decomposition and composition, degree, and others.

4 How to compute Derivatives

If polynomial $p(x)$ of degree d is given by its values on distinct points $x_0, x_1, \ldots, x_n$, then the values of the derivatives can be calculated by means of a particular matrix $\mathbf{D}$ for which

$$\mathbf{D}\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} p'_0 \\ p'_1 \\ p'_2 \\ \vdots \\ p'_n \end{bmatrix}, \tag{1}$$

where $p_i = p(x_i)$ and $p'_i = p'(x_i)$, for $1 \le i \le n$.

The matrix $\mathbf{D}$ depends only on the distinct interpolation points x_i, and can be constructed in $O(n^2)$ operations. To find $\mathbf{D}$, consider the Lagrange polynomials $L_i(x)$ and $p(x) = \sum p_j L_j(x)$. Then $p'(x) = \sum p_j L'_j(x)$ and $p'(x_i) = \sum p_j L'_j(x_i)$ and we need to evaluate $L'_j(x_i)$ to find the (i,j) entry of $\mathbf{D}$. A short computation shows that if $i = j$ then

$$D_{ii} = \sum_{j \ne i} \frac{1}{x_i - x_j} \tag{2}$$

and if $i \ne j$ then

$$D_{ij} = \prod_{\substack{k \ne j \\ k \ne i}} (x_i - x_k) / \prod_{k \ne i} (x_i - x_k)\,. \tag{3}$$

This matrix may be factored into

$$\mathbf{D} = \mathbf{D}_1\mathbf{D}_2\mathbf{D}_1^{-1} + \mathbf{D}_3 \tag{4}$$

where the entries of the diagonal matrix $\mathbf{D}_1$ are $d_{ii} = \prod(x_i - x_j)$, where $j \neq i$, and the matrix $\mathbf{D}_2$ has entries $d_{ij} = 1/(x_i - x_j)$ if $i \neq j$ and $d_{ii} = \sum d_{ij}$, for $j \neq i$(the row sum of the other elements). Finally the entries of the diagonal matrix $\mathbf{D}_3$ are $d_{ii} = \sum d_{ij}$, where $j \neq i$, and d_{ij} are from matrix $\mathbf{D}_2$. We clarify by an example. For the case $n = 3$ on the grid x_0, x_1, x_2, and x_3 we have

$$\mathbf{D}_1 = \begin{bmatrix} (x_0-x_1)(x_0-x_2)(x_0-x_3) & & & \\ & (x_1-x_0)(x_1-x_2)(x_1-x_3) & & \\ & & (x_2-x_0)(x_2-x_1)(x_2-x_3) & \\ & & & (x_3-x_0)(x_3-x_1)(x_3-x_2) \end{bmatrix}, \tag{5}$$

and

$$\mathbf{D}_2 = \begin{bmatrix} 0 & \frac{1}{x_0-x_1} & \frac{1}{x_0-x_2} & \frac{1}{x_0-x_3} \\ -\frac{1}{x_0-x_1} & 0 & \frac{1}{x_1-x_2} & \frac{1}{x_1-x_3} \\ -\frac{1}{x_0-x_2} & -\frac{1}{x_1-x_2} & 0 & \frac{1}{x_2-x_3} \\ -\frac{1}{x_0-x_3} & -\frac{1}{x_1-x_3} & -\frac{1}{x_2-x_3} & 0 \end{bmatrix}$$

Clearly $\mathbf{D}_1$ and $\mathbf{D}_2$ may be computed in $O(n^2)$ operations, and (if desired) $\mathbf{D}$ may be explicitly constructed in $O(n^2)$ more, although of course the product $\mathbf{D}p$ may be carried out in $O(n^2)$ operations as $t_1 = \mathbf{D}^{-1}\,p$, $t_2 = \mathbf{D}_2\,t_1$, $t_3 = D_1\,t_2$, $t_4 = t_3 + \mathbf{D}_3\,p$

5 The Bézout Matrix in the Lagrange Basis

We begin with the familiar construction of the Bézout matrix in the monomial basis from the Cayley quotient (see e.g. [1]):

$$C(x, \eta) = \frac{f(x)g(\eta) - f(\eta)g(x)}{x - \eta}. \tag{6}$$

If $\max(\deg(f), \deg(g))$ is n, then $C(x, \eta)$ is polynomial in x and η of degree at most $n-1$, and can be written as a quadratic form defining the symmetric

Bézout matrix $\mathbf{B}^{(M)}$:

$$C = \begin{bmatrix} 1 & \eta & \ldots \eta^{n-1} \end{bmatrix} \mathbf{B}^{(M)} \begin{bmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{bmatrix}, \tag{7}$$

Now consider $n+1$ distinct sample points x_i, $1 \le i \le n+1$, and $n+1$ distinct sample points η_i, $1 \le i \le n+1$. For now, each η_i is different from each x_i though later we will let $\eta_i \to x_i$. Each set of sample points defines a Lagrange basis: $L_i(x; x_1, x_2, \ldots, x_{n+1})$, or $L_i^{n+1}(x)$ for short, and $L_i(\eta; \eta_1, \eta_2, \ldots, \eta_{n+1})$ or $L_i^{n+1}(\eta)$ for short.

We will also consider the bases constructed by omitting one point, say x_{n+1}; call these $L_i^n(x)$ and $L_i^n(\eta)$. In fact these will be used more often. Note that

$$\begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_1 & x_2 & x_3 & \ldots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \ldots & x_n^2 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \ldots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} L_1^n(x) \\ L_2^n(x) \\ L_3^n(x) \\ \vdots \\ L_n^n(x) \end{bmatrix} \tag{8}$$

because $\sum_{i=1}^n x_i^k L_i(x)$ interpolates x^k on these n points uniquely if $0 \le k \le n-1$; similarly for the $n+1$ vectors $\begin{bmatrix} 1 & x & x^2 & \ldots & x^n \end{bmatrix}$, and $\begin{bmatrix} 1 & \eta & \eta^2 & \ldots & \eta^n \end{bmatrix}$. This matrix is the transpose of the Vandermonde matrix $\mathbf{V}_n$. We may therefore write (the first $\mathbf{V}_n$ depends on η_j, the second on x_j):

$$C(x, \eta) = \begin{bmatrix} L_1^n(\eta), L_2^n(\eta), \ldots, L_n^n(\eta) \end{bmatrix} \mathbf{V}_n \mathbf{B}^{(M)} \mathbf{V}_n^T \begin{bmatrix} L_1^n(x) \\ L_2^n(x) \\ \vdots \\ L_n^n(x) \end{bmatrix} = \sum_{i=1}^n \sum_{j=1}^n c_{ij} L_i^n(x) L_j^n(\eta). \tag{9}$$

defining the Lagrange basis Bézout matrix $\mathbf{B}^{(L)} = \mathbf{V}\mathbf{B}^{(M)}\mathbf{V}^T$. We now have, by evaluation at x_i and η_j,

$$\mathbf{B}_{ij}^{(L)} = \frac{f(x_i)g(\eta_j) - f(\eta_j)g(x_i)}{x_i - \eta_j}.$$

We now let $\eta_j \to x_j$ for all $1 \le j \le n+1$. This does no harm to the above formula except when $i = j$. In this case a limiting argument, adding

$-f(\eta_j)g(\eta_j) + f(\eta_j)g(\eta_j)$ to the numerator, shows that

$$\mathbf{B}^{(L)}_{ii} = f'_i\, g_i - f_i\, g'_i\,,$$

where $f_i = f(x_i), g_i = g(x_i), f'_i = f'(x_i)$ and $g'_i = g'(x_i)$. Note that to calculate f'_i and g'_i we need to use all f_i, g_i on $1 \le i \le n+1$. We use the method of section 4 to do this efficiently. Since $\mathbf{V}_n$ is nonsingular (all x_j are distinct) this matrix $\mathbf{B}^{(L)}$ is singular exactly when $\mathbf{B}^{(M)}$ is; that is, when f and g have a common zero.

5.1 The null spaces determine common roots

We can say more. Suppose first that f and g have exactly one common zero x^*. Then it is well known that $\mathbf{B}^{(M)}$ will have its nullspace spanned by the vector $X = \left[\begin{array}{cccccc} 1 & x^* & (x^*)^2 & \ldots & (x^*)^{n-1} \end{array}\right]^T$. Therefore $\mathbf{B}^{(L)}$ will have its nullspace spanned by $(V^T)^{-1}X = [L_1(x^*), L_2(x^*), \ldots, L_{n-1}(x^*)]$.

We may use any vector U in this nullspace to compute x^* as follows, by a procedure that is referred to as "taking moments":

$$x^* = \frac{\sum_{i=1}^n x_i u_i}{\sum_{i=1}^n u_i} \tag{10}$$

Proof:
By hypothesis $u_i = \alpha L_i(x^*)$ for some α. Then $\sum_{i=1}^n u_i = \alpha \sum_{i=1}^n L_i(x^*) = \alpha$, because the sum interpolates 1 on this set of points; likewise $\sum x_i u_i = \alpha \sum x_i L_i(x^*) = x^* \alpha$ because the sum interpolates x.

We will try to ensure in our main application that there will generically only be one common root, but it is of interest to use what happens in general. We consider first the case of only one root, but a multiple one. If there is one common root of multiplicity m, then it is well-known that the null space of $\mathbf{B}^{(M)}$ is spanned by (each column is the derivative of the previous one)

$$\underbrace{\left[\begin{array}{cccc} 1 & 0 & 0 & \ldots \\ x^* & 1 & 0 & \ldots \\ (x^*)^2 & 2x^* & 2 & \ldots \\ \vdots & \vdots & \vdots & \ldots \\ (x^*)^{n-1} & (n-1)(x^*)^{n-2} & \ldots & \ldots \end{array}\right]}_{m\,\textbf{columns}} \tag{11}$$

Thus the null space of $\mathbf{B}^{(L)}$ is spanned by derivatives of Lagrange polynomials:

$$\begin{bmatrix} L_1(x^*) & L_1'(x^*) & \dots & L_1^{(m-1)}(x^*) \\ L_2(x^*) & L_2'(x^*) & \dots & L_2^{(m-1)}(x^*) \\ \vdots & \vdots & \vdots & \vdots \\ L_n(x^*) & L_n'(x^*) & \dots & L_n^{(m-1)}(x^*) \end{bmatrix} \tag{12}$$

Identification of x^* from these null spaces is possible (in either case) but requires computational effort because the null space as computed will not, in general, be given in this form but rather as a collection of m independent linear combinations of these vectors. One has to solve for the combining coefficients at the same time as solving for x^*. In the monomial basis case this leads to a triangular system of equations (that can be used as a starting point for a nonlinear least squares fit in the case of a null space contaminated by noise, e.g data error and rounding error). In the Lagrange basis case again we take moments, as follows. Suppose:

$$N = N_c \mathbf{A} \tag{13}$$

is given, where we need to find the $m \times m$ matrix $\mathbf{A}$ as well as finding x^*. Then we compute

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^{(m-1)} & x_2^{(m-1)} & \dots & x_n^{(m-1)} \end{bmatrix} N = \begin{bmatrix} 1 & 0 & \dots & 0 \\ x^* & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & 0 \\ (x^*)^{(m-1)} & \dots & \dots & n! \end{bmatrix} \mathbf{A} \tag{14}$$

and we may compare the entries in order to give triangular equations to solve for the entries of $\mathbf{A}$ and for x^*.

The case of two or more common roots that are different is harder but still soluble. We illustrate with two common roots x^* and y^*. We suppose that two non-parallel vectors $a = \alpha_1 L(x^*) + \alpha_2 L(y^*)$ and $b = \beta_1 L(x^*) + \beta_2 L(y^*)$ are given (α_1,α_2, x^*, y^*, and β_1, β_2 are unknown). Taking moments (i.e. multiplying by the interpolation points x_i and summing over all of them) we find

$$\begin{aligned} a_1 &= \alpha_1 + \alpha_2 & b_1 &= \beta_1 + \beta_2 \\ a_2 &= \alpha_1 x^* + \alpha_2 y^* & b_2 &= \beta_1 x^* + \beta_2 y^* \\ a_3 &= \alpha_1 (x^*)^2 + \alpha_2 (y^*)^2 & b_3 &= \beta_1 (x^*)^2 + \beta_2 (y^*)^2 \end{aligned} \tag{15}$$

because, for example, $\sum x_i^2(\alpha_1 L_i(x^*) + \alpha_2 L_i(y^*)) = \alpha_1(x^*)^2 + \alpha_2(y^*)^2$.

Eliminating α_1, α_2, β_1 and β_2 by the first two sets of equations gives, if $x^* \neq y^*$,

$$\begin{bmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{bmatrix} = \frac{1}{y^* - x^*} \begin{bmatrix} y^* & -1 \\ -x^* & 1 \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \tag{16}$$

and the last pair of equations yields

$$\begin{aligned} x + y &= \tfrac{a_1 b_3 - a_3 b_1}{a_1 b_2 - b_1 a_2} = t_1 \\ xy &= a_2 b_3 - a_3 b_2 = t_2 \end{aligned} \tag{17}$$

The denominator for $x + y$ is not zero if the initial vectors are not parallel. These equations can be solved to get $x = (t_1 + \sqrt{t_1^2 - 4t_2})/2$, and $y = (t_1 - \sqrt{t_1^2 - 4t_2})/2$. Interchanging the signs just interchanges x and y. Larger null spaces are harder to deal with, but the symmetry just alluded to allows reasonable solution, especially if there are not too many common roots.

6 Computation of the GCD

The Bézout matrix (see section 5) allows one to compute the coefficients of the GCD, as a polynomial expressed in a certain basis, by solving one linear system of equations and taking linear combinations of the columns. But this is only true if the basis (in the vector space of polynomials with degree smaller than n) used are triangular, i.e. $\deg \phi_k = \deg \phi_{k-1} + 1$, which is not the case for the Lagrange Basis but is the case for the basis coming from Newton Interpolation (see section 6.2). Thus a change of basis moving from Lagrange to Newton will produce the matrix with the right property. This represents a compromise between pure "polynomials by values" and the standard monomial basis. Note that in the standard theory this "right property" produces the coefficients of the gcd with respect to the monomial basis but it is not complicated to get the coefficients in the Newton basis instead, and from there evaluate the GCD at the given points.

6.1 Barnett's Theorems through Bezout matrices.

Since $p_0^m \mathrm{Bez}(P, Q)$ (p_0 is the leading coefficient of P) is the matrix associated to the linear mapping Φ_P^Q considering the Horner basis in the initial vector space and the Standard basis in the final one. Barnett's Theorems I and II [1] can be rewritten as follows with Bezout matrices (note that the polynomials which define B_{Ho} are arranged in decreasing degree order).

Theorem 6.1.
The degree of the greatest common divisor of $P(x), Q_1(x), \ldots, Q_t(x)$ verifies the following formula:

$$\deg(\gcd(P, Q_1, \ldots, Q_t)) = n - \mathrm{rank}(\mathcal{B}_P(Q_1, \ldots, Q_t))$$

where

$$\mathcal{B}_P(Q_1, \ldots, Q_t) = \begin{pmatrix} \mathrm{Bez}(P, Q_1) \\ \vdots \\ \mathrm{Bez}(P, Q_t) \end{pmatrix}.$$

Theorem 6.2.
If $c_1, \ldots, c_n$ are the columns of the matrix $\mathcal{B}_P(Q_1, \ldots, Q_t)$ and its rank is $n - k$ then the last $n - k$ columns $c_{k+1}, \ldots, c_n$ are linearly independent and each c_i $(1 \leq i \leq k)$ can be written as a linear combination of $c_{k+1}, \ldots, c_n$.

Finally it is shown how to use the matrix $\mathcal{B}_P(Q_1, \ldots, Q_t)$ in order to get the coefficients of the greatest common divisor of $P(x), Q_1(x), \ldots, Q_t(x)$.

Theorem 6.3.
If $c_1, \ldots, c_n$ are the columns of the matrix $\mathcal{B}_P(Q_1, \ldots, Q_t)$, $n - k$ is its rank,

$$c_{k-i} = h_{k-i}^{k+1} c_{k+1} + \sum_{j=k+2}^{n} h_{k-i}^{j} c_j, \quad i = 0, \ldots, k-1,$$

$\{d_1, \ldots, d_k\}$ given by

$$d_j = d_0 h_{k-j+1}^{k+1}$$

and $d_0 \in F$ then:

$$D(x) = d_0 x^k + d_1 x^{k-1} + \ldots + d_{k-1} x + d_k$$

is a greatest common divisor for the polynomials $P(x), Q_1(x), \ldots, Q_t(x)$.

6.2 Other bases

We know that $P(x)$ is of degree n and the values

$$\{P(x_0), \ldots, P(x_n) : x_i \in D, x_i \neq x_j\}$$

Then the set

$$C_{\mathrm{N}} = \{1, (x - x_1), (x - x_1)(x - x_2), \ldots, (x - x_1)(x - x_2) \ldots (x - x_{n-1})\}$$

is another basis of $F_n[x]$ coming from Newton's Interpolation algorithm. Thus:

$$P(x) = \nu_0 + \nu_1(x - x_1) + \ldots + \nu_n \prod_{i=1}^{n}(x - x_i)$$

with

$$\nu_0 = P(x_1), \quad \nu_1 = \frac{P(x_2) - \nu_0}{x_2 - x_1},$$

y

$$\nu_i = \frac{P(x_{i+1}) - \nu_0 - \ldots - \nu_{i-1}(x_{i+1} - x_1) \cdot \ldots \cdot (x_{i+1} - x_{i-1})}{(x_{i+1} - x_1) \ldots (x_{i+1} - x_i)}.$$

Then the matrix of multiplying times x in the quotient by P is:

$$\Lambda_P = \begin{bmatrix} \nu_n x_1 & 0 & \cdots & \cdots & 0 & -\nu_0 \\ \nu_n & \nu_n x_2 & \cdots & \cdots & 0 & -\nu_1 \\ 0 & \nu_n & \cdots & \cdots & 0 & -\nu_2 \\ 0 & 0 & \cdots & \cdots & 0 & -\nu_3 \\ \vdots & \vdots & & & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \nu_n x_{n-1} & -\nu_{n-2} \\ 0 & 0 & 0 & 0 & \nu_n & (-\nu_{n-1} + x_n \nu_n) \end{bmatrix}.$$

and if Q is monic (if not, we may instead construct a generalized matrix pencil) then $Q(\Lambda_P)$ has the right properties with respect to the computation of the coefficients of the GCD.

7 Solving bivariate polynomial systems using the Bézout matrix in the Lagrange basis

If we suppose that f and g are given on the rectangular grid $X \times Y$ when $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$, and that these are enough points to specify f and g both, then for each y_k, $1 \le k \le n$, we may form the Bézout matrix $\mathbf{B}_k^{(L)}$ of $f(x, y_k)$ and $g(x, y_k)$ by the method of the previous section. Then the Bezoutian is

$$\mathbf{B}^{(L)}(y) = \sum_{k=0}^{n} \mathbf{B}_k^{(L)} L_k(y), \tag{18}$$

and for this to be singular we must have the generalized companion matrix $\mathbf{C}_0 - y\mathbf{C}_1$ also be singular where $\mathbf{C}_0, \mathbf{C}_1$ were given, for example, in [2] (though we will see an improved method shortly). Thus to find the y^* values for which $f(x, y^*)$ and $g(x, y^*)$ have common roots, we may numerically solve the generalized eigenproblem $\mathbf{C}_0 V = y\mathbf{C}_1 V$. Moreover, as stated in [2], the null vectors of $\mathbf{B}^{(L)}(y^*)$ are of the form $L = [L_1(x^*), L_2(x^*), \ldots, L_m(x^*)]^T$, and the resulting generalized eigenvectors V of the pencil $\mathbf{C}_0 - \lambda\mathbf{C}_1$ are of the form $[y^{n-1}L, y^{n-2}L, \ldots, yL, L]^T$. This mixture of monomial bases and Lagrange bases is not entirely satisfactory, and we improve on it below, but it succeeds in solving some bivariate system given on rectangular grids. One significant problem with it is the problem of spurious multiplicity: for a given y^*, there may be multiple x-roots (and vice versa) purely due to symmetry. The way to get around this problem in monomial basis expressions is to perform a random change of coordinate [6]. In this context, this is tantamount to "regridding", an expensive process that may, as well, introduce errors.

Another problem is the possibility of spurious roots, or even multiple roots, at infinity. Again in the monomial case the cure is a random change of projective coordinates.

7.1 An improved Generalized Companion Matrix Pencil for polynomials given by values

The paper [2] gave an n by n companion matrix pencil C_0, C_1 such that if $p(x)$ was given by its values p_1, p_2, up to p_{n+1} on the $n+1$ points x_1, x_2, up to x_{n+1} then $p(x) = \det(C_0 - xC_1)$. Thus finding the roots of p could be done by finding the generalized eigenvalues of this pencil. Other methods for doing this also exist: see for example [4], who uses a method of Smith [7] (that is quite general and can handle multiple roots). The method of Smith as used in [4] has not yet been generalized to matrix polynomials, however, and requires a rank-one outer product in its construction.

The construction of [2] required $O(n^3m^2)$ work (even more work than the rank-one outer product in the method of Smith) to construct the two most complicated blocks in the generalized companion matrix. We present a cheaper alternative now. Consider for expository purposes the scalar degree 3 Lagrange polynomial on 4 points x_1, x_2, x_3, and x_4:

$$p(x) = p_1 L_1(x) + p_2 L_2(x) + p_3 L_3(x) + p_4 L_4(x)\ .$$

Consider the matrix

$$C_0 = \begin{bmatrix} x_1 & & & & p_1 \\ & x_2 & & & p_2 \\ & & x_3 & & p_3 \\ & & & x_4 & p_4 \\ -\ell_1 & -\ell_2 & -\ell_3 & -\ell_4 & 0 \end{bmatrix} \tag{19}$$

where $\ell_1 = 1/((x_1 - x_2)(x_1 - x_3)(x_1 - x_4))$ is the first Lagrange polynomial coefficient, ℓ_2 is the second, and so on. Then let

$$C_1 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 0 \end{bmatrix} \tag{20}$$

be the identity matrix *except that the final entry is zero*. Then expansion by minors shows that $\det(C_0 - x_k C_1) = p_k$ for $1 \le k \le 4$, and moreover that $\det(C_0 - xC_1)$ has degree 3, even though C_1 and C_0 are both 5 by 5 matrices. We have just shown that the characteristic polynomial of this pencil is exactly the unique degree 3 polynomial that interpolates p_k on the 4 given points. This construction is perfectly general, and moreover generalizes to matrix polynomials, as we will see shortly.

Finally, the right and left eigenvectors corresponding to a simple eigenvalue r (root of the interpolating polynomial) are easily seen to be

$$R = \begin{bmatrix} p_1/(r - x_1) \\ p_2/(r - x_2) \\ p_3/(r - x_3) \\ p_4/(r - x_4) \\ 1 \end{bmatrix} \tag{21}$$

and $L = \left[L_1(r), L_2(r), L_3(r), L_4(r), \prod_{i=1}^{4}(r - x_i)\right]$, where $L_i(r)$ are the Lagrange polynomials (on the given four points) evaluated at $x = r$. Thus, as before, the roots r may be recovered from the left eigenvectors by taking moments (ignoring the final entry).

There is also an infinite eigenvalue of this matrix pencil, with left eigenvector $[0,0,0,0,1]$, and its transpose is the corresponding right eigenvector.

7.2 Extension to Matrix Polynomials

Extension to matrix polynomials is straightforward. If the values of the matrix polynomial at $x = x_1$, x_2, x_3 and x_4 are the matrices $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_3}$ and $\mathbf{P_4}$, then the generalized companion matrix pencil is

$$C_0 = \begin{bmatrix} x_1 I & & & & \mathbf{P_1} \\ & x_2 I & & & \mathbf{P_2} \\ & & x_3 I & & \mathbf{P_3} \\ & & & x_4 I & \mathbf{P_4} \\ -\ell_1 I & -\ell_2 I & -\ell_3 I & -\ell_4 I & 0 \end{bmatrix} \tag{22}$$

where as before the ℓ_k are the (scalar) Lagrange polynomial coefficients, and

$$C_1 = \begin{bmatrix} I & & & & \\ & I & & & \\ & & I & & \\ & & & I & \\ & & & & 0 \end{bmatrix} \tag{23}$$

where the blocks I are conformal with the square blocks P_k. Then $\det(C_0 - xC_1) = \det \mathbf{P}(\mathbf{x}) = \det(L_1(x)\mathbf{P_1} + L_2(x)\mathbf{P_2} + L_3(x)\mathbf{P_3} + L_4(x)\mathbf{P_4})$, and if $\mathbf{v}$ is any (left if transposed) null vector of $\mathbf{P}(\mathbf{x})$ then the eigenvectors of this pencil corresponding to the finite eigenvalue r are

$$R = \begin{bmatrix} \mathbf{P_1}\mathbf{v}/(r - x_1) \\ \mathbf{P_2}\mathbf{v}/(r - x_2) \\ \mathbf{P_3}\mathbf{v}/(r - x_3) \\ \mathbf{P_4}\mathbf{v}/(r - x_4) \\ \mathbf{v} \end{bmatrix} \tag{24}$$

and on the left $[L_1(r)\mathbf{v^T}, L_2(r)\mathbf{v^T}, L_3(r)\mathbf{v^T}, L_4(r)\mathbf{v^T}, (r - x_1)(r - x_2)(r - x_3)(r - x_4)\mathbf{v^T}]$. The infinite eigenvalues are analogous.

The generalization from 4 points to n points is clear, as is the use of this generalized companion matrix to find the eigenvalues y that make the Bézout matrix singular when the Bézout matrix is expressed by its (matrix) values at the sample points y_k.

8 Univariate division

The univariate case is substantially easier than the multivariate case, in that remainders are (however computed) automatically unique, in any basis.

We wish to divide the polynomial $A(x)$, given by its values a_j, $1 \leq j \leq N$ on the ordered grid x_j by the polynomial $B(x)$ given by its values b_j on the same grid. We assume $\deg(A) = n$ and $\deg(B) = m$ are known, and $n \geq m$, else the problem is trivial. We seek the values Q_j and R_j such that

$$a_j = Q_j b_j + R_j \qquad 1 \leq j \leq N, \tag{25}$$

where the Q_j are the values of a polynomial of degree $n - m$, whilst the R_j are the values of a polynomial of degree at most $m - 1$. We have $2N$ unknowns, and only N equations. We now impose the degree constraints. We could require that derivatives $Q^{(n-m+1)}(x_j) = 0$ and $R^{(m)}(x_j) = 0$ for all x_j, which gives us $2N$ more equations, overconstraining the system. By using the techniques of section 4, we may express these derivatives as a linear combination of the values Q_j and of the values R_j. However, we may also impose these degree constraints by using differences, which makes the result simpler in the case of equally spaced grids.

Lemma 8.1. If the values of the degree at most ℓ polynomial $p(x)$ are given at the $\ell + 1$ distinct points $x_1, x_2, \ldots, x_{\ell+1}$ then the values of the(constant) ℓ^{th} derivative at each of these $\ell + 1$ points is given by

$$p^{(\ell)}(x) = \sum_{j=1}^{\ell+1} \frac{\ell!\, p_j}{\prod_{k \neq j}(x_j - x_k)}. \tag{26}$$

Proof:
Interpolate $p(x)$ by Lagrange polynomials;

$$p(x) = \sum_{j=1}^{\ell+1} p_j\, L_j(x)\,,$$

where

$$L_j(x) = \prod \frac{(x - x_k)}{(x_j - x_k)} \quad k \neq j \tag{27}$$

The ℓ^{th} derivative of $\prod_{k \neq j}(x - x_k)$ is $\ell!$.

Lemma 8.2. If $p(x)$ has degree larger than ℓ, then there exists a point ξ somewhere in $\min x_j < \xi < \max x_j \quad, 1 \le j \le \ell+1$, where

$$p^{(\ell)}(\xi) = \ell! \sum_{j=1}^{\ell+1} \frac{p_j}{\prod_{k \neq j}(x_j - x_k)}.$$

Proof:
Use the extended mean value theorem: the lemma is true for $\ell = 1$ and by induction for all ℓ.

Thus, according to the lemma, insisting that

$$\sum_{j=1}^{\ell+1} \frac{p_j}{\prod_{k \neq j}(x_j - x_k)} = 0$$

imposes at least one condition $p^{(\ell)}(\xi) = 0$ for some $\xi \in (\min x_j, \max x_j)$. If we impose N such independent conditions then either $p^{(\ell)}(x) \equiv 0$ or $\deg(p) > N+\ell-1$. But by assumption $\deg p \le N-1$(else we would need more points). Taking $\ell + 1 = N$ makes the derivative exact but possibly trivial.

We may also impose the further conditions that $p^{(\ell)}(\xi) = 0$ for all $\ell \ge n-m+1$ in the case of Q and all $\ell \ge m$ in the case of R. This gives $N-(n-m)+N-(m)$ further, possibly redundant, conditions.

An example at this point will clarify the process. Consider $A(x)$ of degree 3 given on the grid of 5 points $x_1 < x_2 < x_3 < x_4 < x_5$, and $B(x)$ of degree 2 given on the same grid. Then division requires that we find Q_j and R_j such that

$$A_j = Q_j B_j + R_j \qquad 1 \le j \le 5.$$

Q_j must be of degree 1. Therefore (assuming the x_k are equally spaced in this case to make the presentation easier)

$$\begin{aligned} Q_1 - 2Q_2 + Q_3 &= 0 \\ Q_2 - 2Q_3 + Q_4 &= 0 \\ Q_3 - 2Q_4 + Q_5 &= 0, \end{aligned}$$

The ξ from (x_1, x_2, x_3) must be different than any form (x_3, x_4, x_5) because these intervals are disjoint. Hence $Q''(\xi_1) = Q''(\xi_2) = 0$ and Q must be degree 1 or else of degree larger than 4.

$R(x)$ must also in this case be of degree 1. In a similar way we see

$$\begin{aligned} R_1 - 2R_2 + R_3 &= 0 \\ R_2 - 2R_3 + R_4 &= 0 \\ R_3 - 2R_4 + R_5 &= 0, \end{aligned}$$

and we now have 11 equations in the 10 unknowns Q_j, R_j.

We may also (or instead) impose

$$A(\alpha) = Q(\alpha)B(\alpha) + R(\alpha)$$

at any random point $\alpha \notin \{x_j\}$ by use of the Lagrange interpolation formula: this gives

$$A(\alpha) = \sum_{j=1}^{N} Q_j \, L_j(\alpha) \, B(\alpha) + \sum_{j=1}^{N} R_j \, L_j(\alpha) \; ,$$

a linear equation in Q_j, R_j.

We may also impose third difference conditions:

$$\begin{aligned} Q_1 - 3Q_2 + 3Q_3 - Q_4 &= 0 \\ Q_2 - 3Q_3 + 3Q_4 - Q_5 &= 0 \end{aligned}$$

$$\begin{aligned} R_1 - 3R_2 + 3R_3 - R_4 &= 0 \\ R_2 - 3R_3 + 3R_4 - R_5 &= 0 \end{aligned}$$

However, these are redundant. Note that

$$Q_1 - 3Q_2 + Q_3 = 0, \qquad Q_2 - 2Q_3 + Q_4 = 0$$

implies

$$Q_1 - 3Q_2 + 3Q_3 - Q_4 = 0 \, . \tag{28}$$

Using second differences only, our matrix is:

$$
\begin{bmatrix}
B_1 & & & & & 1 & & & & \\
& B_2 & & & & & 1 & & & \\
& & B_3 & & & & & 1 & & \\
& & & B_4 & & & & & 1 & \\
& & & & B_5 & & & & & 1 \\
1 & -2 & 1 & & & & & & & \\
& 1 & -2 & 1 & & & & & & \\
& & 1 & -2 & 1 & & & & & \\
& & & & & 1 & -2 & 1 & & \\
& & & & & & 1 & -2 & 1 & \\
& & & & & & & 1 & -2 & 1
\end{bmatrix}
\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{bmatrix}
=
\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (29)
$$

which, it turns out, may be solved for the quotient and remainder even though it is singular. The example matrix above is rank 8, but the vector on the right hand side is in the range of the matrix; indeed, there is a two dimensional space of possible solutions. To force a unique choice of solution, we *minimize the norm of the solution vector* by using the SVD or equivalently the Moore-Penrose inverse. As is well-known, using the SVD will minimize the norm of the solution in the case of column-rank deficient matrix problems.

This has the incidental benefit of allowing this division algorithm to answer the boolean "`divides?`" question. We say that $a(x)$ exactly divides $b(x)$ if its remainder is "small enough", that is, less than some pre-assigned tolerance (to within an acceptable factor). If the polynomials are given only on a grid, then what we are saying is that one polynomial divides another if the values of the remainder on the grid are small enough. This is an apparently sensible description in the face of data error.

The generalization of this example to a full algorithm is clear: set up the division matrix, including the degree constraints, and solve the system using the SVD (or a more special-purpose direct minimization scheme, perhaps minimizing the maximum value of the remainder on the grid).

9 Multivariate Division

The generalization of the previous section to the multivariate case is reasonably straightforward. One chooses an ordering for the basis to express remainders, say $[\phi_1(x), \phi_2(x), \ldots, \phi_N(x)]$, in the Lagrange basis case (when this corresponds to a vector of polynomial values on the grid) for example by traversing the grid in a definite order, and this specifies the placement of the entries in the division matrix. The degree constraint rows now must relate the values of the polynomial given at points recorded at possibly distant places in the vector of polynomial values, but this is just a matter of bookkeeping.

One important refinement is that the degree constraints must now express the idea that "no leading term of the divisors may divide any term in the remainder". Typically this can be imposed by requiring that (in the two-dimensional case, with an obvious generalization to m dimensions)

$$\frac{\partial^d}{\partial x^k \partial y^{d-k}}\, p(x,y) = 0 \tag{30}$$

and this can be specified as usual by a linear constraint. Imposing this on every available point on the grid usually gives enough constraints to specify the problem.

Example 9.1. Consider the following two bivariate polynomials:

$$\{f = x^3 + 2y^2 - 6x^2y - 2x + 7,\ g = x^2y^2 + y^2 - 1\} \tag{31}$$

We artificially construct a Lagrange basis example by sampling these polynomials at the sixth and fifth roots of unity respectively:

$$x_m = \exp\,(\frac{2\,i\pi\,m,}{6}), \qquad m = 0\ldots 5$$

$$y_n = \exp\,(\frac{2\,i\pi\,n}{5}), \qquad n = 0\ldots 4$$

The values shown in Table 1 are the Lagrange coefficients of f and g with respect to the given sample points. At this point, we pretend we know f and g only by these values and that we know their degrees $[3,0]$ and $[2,2]$. Now if we want to divide any bivariate polynomial, say p, of known degree, whose Lagrange coefficients at the same sample points are given as the vector $\mathbf{p}$, by f and g we must solve the following linear system:

$$\mathbf{M}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\\ \mathbf{r}\end{bmatrix} = \begin{bmatrix}\mathbf{p}\\ \mathbf{0}\end{bmatrix}, \tag{32}$$

Table 1: f and g evaluated at the sample points

f	g
2.0	1.0
$2.527864045 - 4.530768593\,i$	$-2.618033988 + 1.175570504\,i$
$11.47213595 - 5.428824546\,i$	$-0.3819660112 - 1.902113032\,i$
$11.47213595 + 5.428824546\,i$	$-0.3819660112 + 1.902113032\,i$
$2.527864045 + 4.530768593\,i$	$-2.618033988 - 1.175570504\,i$
$10.0 - 6.928203230\,i$	$-0.5000000000 + 0.8660254037\,i$
$\vdots$	$\vdots$
$0.136761242 - 2.610482083\,i$	$-1.669130606 - 0.7431448254\,i$
$6.245204768 - 2.332967532\,i$	$-0.0218523992 + 0.2079116908\,i$
$9.250851615 - 0.690989841\,i$	$-1.913545457 + 0.4067366430\,i$

Where $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{r}$ are the Lagrange coefficients of the polynomials a, b and r (the remainder) respectively in:

$$p(x, y) = a(x, y)f(x, y) + b(x, y)g(x, y) + r(x, y) \tag{33}$$

$\mathbf{M}$ is analogous to the matrix in Equation 29. In Figure 2 an image of $\mathbf{M}$ has been shown, with nonzero entries represented by black dots.

We see that $\mathbf{M}$ is a sparse matrix whose first 30 rows show the equations related to evaluation at the grid points. These rows do not depend on p.However, the next rows which show the degree constraints imposed on a, b and r, depend on the degrees of p. Notice that the constraint blocks each have analogous, but not identical, structure.

After solving for the coefficients of r, we may separately test to ensure that the degree constraints (coming from the fact that no term in the remainder may be divisible by a leading term of f or g) are satisfied.

10 Solving Systems of Polynomials by Division

Suppose that we are given $I = \{f_i(x)\}$, $1 \leq i \leq n$, $x = (x_1, x_2, \ldots, x_s)$ and assume that $V(I)$ is zero-dimensional. Suppose also that we are given

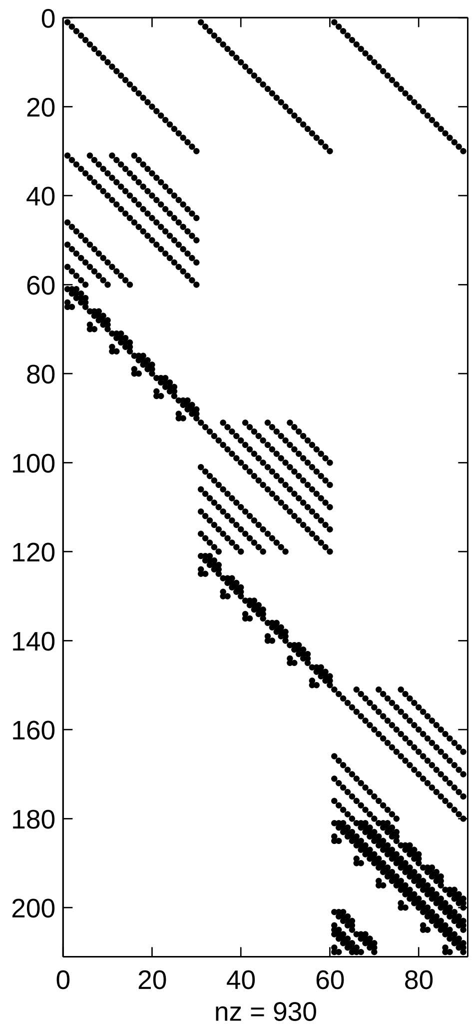

Figure 2: The Image of the Sparse Matrix $\mathbf{M}$

a basis for $K[x]$, perhaps the monomial basis $1, x_1, \ldots, x_s, x_1^2, \ldots$, or perhaps the family of Lagrange Polynomials, and an ordering. Take $\alpha \in K$ randomly, and consider the remainder of a random polynomial ρ mod I, say $\rho = \sum_{i=1}^{N} r_i \phi_i(x)$ where no leading term of any f_i divides any ϕ_i . By the randomness of ρ we conclude that generically all the remainder basis elements ϕ_i that we need are present(this need not be so but we will plan for the contingency). Now for each $\phi_i(x)$, compute

$$(x_1 - \alpha)\phi_i(x) \quad \mathtt{mod}\, I \tag{34}$$

and, if no new $\tilde{\phi}_j$ are introduced by these divisions, express each as

$$(x_1 - \alpha)\phi_i(x) \equiv \sum_{j=1}^{N} r_{ij}\phi_j. \tag{35}$$

If for some ϕ_i we need some new $\tilde{\phi}_j$ to express ϕ_i mod I, we may add each such $\tilde{\phi}_j$ to the list of ϕ_j at the end, say $\phi_{N+1} = \tilde{\phi}_j$, and extend each previously computed remainder

$$(x_1 - \alpha)\phi_i = \sum_{j=1}^{N} r_{ij}\phi_j + 0 \cdot \phi_{N+1} \tag{36}$$

Will this process terminate? If $\{f_i\}$ is a Gröbner basis already, then with probability 1 it will terminate on the first step, but in any case the number of possible $\tilde{\phi}_j$ to add is finite (by hypothesis I is zero-dimensional). But, if f is not a Gröbner basis, then we do not as yet have reason to expect it to terminate.

Theorem 10.1. If a total-degree ordering is used, then this process terminates.

Proof:
Any remainder terms can not be higher total degree than the initial polynomials $(x_1 - \alpha)\phi_j$. There are only finitely many such.

From now on we suppose that the set $\{\phi_i\}_{i=1}^{N}$ is fixed, and closed under the operation $(x_1 - \alpha)\phi_i$ mod $I = \sum_{j=1}^{N} r_{ij}\phi_j$.

In the case of Lagrange bases, we presume that our totality of sample points determines such a closed remainder space basis.

Corollary 10.1. If

$$(x_1 - \alpha)\phi_i(x) \equiv \sum_{j=1}^{N} r_{ij}\phi_j(x) \mod I, \tag{37}$$

then each $x^* \in V(I)$ parameterizes an eigenvector of the matrix $[r_{ij}]$ because at the variety, the congruence above becomes equality: thus

$$(x_1^* - \alpha)\begin{bmatrix} \phi_1(x^*) \\ \phi_2(x^*) \\ \vdots \\ \phi_N(x^*) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1N} \\ r_{21} & r_{22} & \dots & r_{2N} \\ \vdots & \vdots & \dots & \vdots \\ r_{N1} & \dots & \dots & r_{NN} \end{bmatrix} \begin{bmatrix} \phi_1(x^*) \\ \phi_2(x^*) \\ \vdots \\ \phi_N(x^*) \end{bmatrix}. \tag{38}$$

In the monomial basis, we know how to find x^* given such an eigenvector [8]. For the Lagrange basis case, we may find each coordinate x_k^* of x^* from the eigenvector $U = \alpha[L_1(x^*), L_2(x^*), \dots, L_N(x^*)]$ by taking moments: if the interpolation points have coordinates $(y_1^{(i)}, y_2^{(i)}, \dots, y_s^{(i)})$ for $1 \le i \le N$, then

$$x_k^* = \sum_{i=1}^{N} y_k^{(i)} u_i \Big/ \sum_{i=1}^{N} u_i, \tag{39}$$

because the linear polynomial x_k is interpolated by $\sum y_k^{(i)} L_i(x)$.

Corollary 10.2. Every root x^* of $V(I)$ may be found by searching the eigenvectors of R, taking moments, and verifying afterwards that $f_i(x^*) = 0$ for $1 \le i \le N$.

Remark 10.1. Of course, there are eigenvectors of R that do not parameterize by roots(unless $\{f_i\}$ a Gröbner basis). So this approach shares, with the method of resultants, the drawback of introducing extraneous roots.

It is hoped that fast black-box methods for solving structured eigenproblems may allow this method to be useful in some circumstances. For now, we note that the method computes pseudozeros—that is, points x^* for which the $|f_i(x^*)|$ are numerically small. One expects the numerical characteristics of such a method to be more faithful to the uncertainty model of the data, because no artificial bases are introduced, potentially transforming a well-conditioned problem to an ill-conditioned one.

Example 10.1. Consider the Lagrange coefficients of f and g given in Example 9.1 with the same x-y sample points. We observe that:

$$x\begin{bmatrix} L_{11} \\ L_{12} \\ L_{13} \\ \vdots \\ L_{64} \\ L_{65} \end{bmatrix} = \mathbf{R_x}\begin{bmatrix} L_{11} \\ L_{12} \\ L_{13} \\ \vdots \\ L_{64} \\ L_{65} \end{bmatrix} \tag{40}$$

where $\mathbf{R_x}$ is the remainder matrix. $\mathbf{R_x}$ is a 30×30. Therefore, it has 30 eigenvalues. However, it turns out that the system has only 10 common roots. The following table shows the (x, y) pairs found through the algorithm. The first 10 pairs are what we discover to be valid common roots of the system.

Table 2: possible x-y roots

x	y
$-3.161100567 - 6.965879443 \times 10^{-13}\, i$	$-0.3016134059 - 1.395122128 \times 10^{-13}\, i$
$0.5297822533 + 2.311433850\, i$	$-0.1230396429 + 0.4423651778\, i$
$0.5297822533 - 2.311433850\, i$	$-0.1230396429 - 0.4423651778\, i$
$1.729089767 + 0.2047225155\, i$	$0.4973613007 - 0.04424307457\, i$
$1.729089767 - 0.20472251550\, i$	$0.4973613007 + 0.04424307457\, i$
$-1.478905530 + 2.694675942 \times 10^{-14}\, i$	$0.5601417707 + 4.705630138 \times 10^{-15}\, i$
$0.1281489965 + 1.013629309\, i$	$-1.356975802 + 1.415769261\, i$
$-0.06701796811 + 0.9844345006\, i$	$-2.146610037 - 1.646858440\, i$
$0.12814899659 - 1.013629309\, i$	$-1.356975802 - 1.415769261\, i$
$-0.06701796811 - 0.9844345006\, i$	$-2.146610037 + 1.646858440\, i$
$0.002760222989 + 0.3577555685\, i$	$1.456477664 + 0.6252671799\, i$
$0.2058182684 - 0.01451830109\, i$	$0.2609557868 + 0.4283966344\, i$
$\vdots$	$\vdots$
$0.8234154099 - 2.000714408\, i$	$0.1156482317 - 0.7991292812\, i$
$0.09553911503 - 0.4260017074\, i$	$-0.1720274879 + 0.1044580018\, i$

To find the valid common roots of f and g as measure the accuracy of the solution, we substitute each x-y pairs from Table 2 into the polynomials

f and g. The results are tabulated in Table 3. We see that the first ten residuals are small enough that we may be justified in accepting them as valid roots.

Table 3: Verifying the actual roots

$f(x,y)$	$g(x,y)$
$8.6 \times 10^{-12} - 2.986313919 \times 10^{-12}\,i$	$-3.555 \times 10^{-12} + 1.325738096 \times 10^{-12}\,i$
$2.00 \times 10^{-12} + 6.0 \times 10^{-13}\,i$	$-4.6 \times 10^{-13} + 2.5 \times 10^{-13}\,i$
$-2.5 \times 10^{-12} + 1.8 \times 10^{-12}\,i$	$3.1 \times 10^{-13} - 6.2 \times 10^{-13}\,i$
$-1.00 \times 10^{-12} + 9.2 \times 10^{-13}\,i$	$1.00 \times 10^{-13} - 2.15 \times 10^{-13}\,i$
$-1.2 \times 10^{-12} + 6.00 \times 10^{-13}\,i$	$2.3 \times 10^{-13} - 5.6 \times 10^{-14}\,i$
$1.9 \times 10^{-12} + 3.395799699 \times 10^{-13}\,i$	$-3.1 \times 10^{-13} - 8.206082248 \times 10^{-15}\,i$
$1.20 \times 10^{-12} - 8.0 \times 10^{-13}\,i$	$7.1 \times 10^{-13} - 1.0 \times 10^{-13}\,i$
$5.14 \times 10^{-12} + 7.6 \times 10^{-12}\,i$	$2.0 \times 10^{-13} - 3.0 \times 10^{-13}\,i$
$-1.2 \times 10^{-13} + 1.3 \times 10^{-12}\,i$	$-1.00 \times 10^{-13} - 4.0 \times 10^{-13}\,i$
$6.39 \times 10^{-12} - 5.0 \times 10^{-13}\,i$	$1.5 \times 10^{-12} + 1.5 \times 10^{-12}\,i$
$11.57997787 + 3.344335491\,i$	$0.505315991 + 1.591690569\,i$
$6.284742931 + 0.3753797104\,i$	$-1.118954789 + 0.2336991755\,i$
$\vdots$	$\vdots$
$12.87802872 - 6.084831261\,i$	$-0.1554338037 + 2.489756422\,i$
$6.566232784 + 0.8697712395\,i$	$-0.9874632983 - 0.03126587424\,i$

Therefore, there are 10 actual roots.

11 Other operations

Other operations to be considered in this context are `degree`, which may be answered by approximate implicitization; composition (which seems to require regridding to align the given f values with the needed domain values for g in $g(f(x))$; decomposition, which may prove easier; factoring, and many others. Some of these have been done: we know how to construct resultants by polynomial values, and how to construct Sylvester matrices, for example. Others are ripe for approach.

12 Concluding Remarks

The algorithms in this paper are not completely studied yet. In particular, the numerical stability of the solution-by-division algorithm has not been elucidated; it very probably depends on the interpolation points in a strong way. The complexity of the algorithms are relatively straightforward: the division matrices are sparse and their construction cost can be counted, and the solution cost can be bounded as usual; though as the matrices are singular an iteratively improved SVD for singular systems as in [3] might be an interesting approach for efficiency, especially if it is modified to account for structure. We believe that the main contribution of this present paper is to give the idea that direct algorithms for polynomials expressed as values may well be interesting.

A fully detailed version of this paper is planned, with detailed explorations of the effects of multiplicity to be studied by Shakoori in her Ph.D. thesis, and of the benefits of using structure, by Amiraslani in his Ph.D. thesis.

References

[1] Barnett, S. *Matrices: Methods and Applications.* Clarendon, Oxford, 1990.

[2] Corless, R. M., and Litt, G. Generalized companion matrices for polynomials not expressed in monomial bases. unpublished.

[3] Corless, R. M., and Schicho, J. Iterative improvement and the SVD for singular systems. Tech. Rep. TR-00-09, at http://www.orcca.on.ca/TechReports, Ontario Research Centre for Computer Algebra, 2000.

[4] Fortune, S. Polynomial root finding using iterated eigenvalue computation. In *Proc. ISSAC* (London, Canada, 2001), B. Mourrain, Ed., ACM, pp. 121–128.

[5] Giesbrecht, M. W., Labahn, G., and shin Lee, W. Symbolic-numeric sparse interpolation of multivariate polynomials. preprint.

[6] Shakoori, A. Solving bivariate polynomials by eigenvalues. Master's thesis, University of Western Ontario, 2003. Department of Applied Mathematics.

[7] Smith, B. T. Error bounds for zeros of a polynomial based upon Gerschgorin's theorem. *Journal of the Association for Computing Machinery 17*, 4 (October 1970), 661–674.

[8] Stetter, H. J. *Numerical Polynomial Algebra.* SIAM, 2004.

[9] Zippel, R. *Effective Polynomial Computation.* Kluwer Academic, Boston, 1993.