Chapter 1

PROPERTIES OF THE COMPANION MATRIX RESULTANT FOR BERNSTEIN POLYNOMIALS

Joab R. Winkler
Department of Computer Science
The University of Sheffield
Regent Court
211 Portobello Street
Sheffield S1 4DP
United Kingdom
j.winkler@dcs.shef.ac.uk

Abstract The computational implementation in a floating point environment of the companion matrix resultant is considered and it is shown that the numerical condition of the resultant matrix is strongly dependent on the basis in which the polynomials are expressed. In particular, a companion matrix of a Bernstein polynomial is derived and this is used to construct a resultant matrix for two Bernstein polynomials. A measure of the numerical condition of a resultant matrix is developed and then used to compare the stability of the resultant matrices of the same polynomials that are expressed in different bases. It is shown that it is desirable to express the polynomials in the Bernstein basis, but since the power basis is the natural choice in many applications, a transformation of the resultant matrix between these bases is required. It is shown that this transformation of the resultant matrix between the bases cannot be achieved by performing a basis transformation of each polynomial. Rather, the equation that defines the transformation of the companion matrix resultant between the bases is derived by considering the eigenvectors of the companion matrix of a polynomial in each basis. The numerical condition of this equation is considered and it is shown that it is ill–conditioned, even for polynomials of low degree.

Keywords: Resultants, Bernstein basis, numerical condition

Introduction

The discovery of Gröbner bases and the numerous applications of resultants, for example, robot motion planning [1], computer graphics [3], inverse kinematics for serial mechanisms [4], computer vision [6] and computer–aided geometric design [7], have caused renewed interest in computational algebraic geometry. This revival has yielded new theoretical results and generated interest in the computational implementation of resultants, and this paper addresses some issues in the latter topic. In particular, it is shown that the implementation of resultants in a floating point environment poses significant challenges; the polynomial basis may significantly affect the accuracy of computed solution, and advanced methods from computational linear algebra are required in order to estimate the numerical condition of a resultant matrix, and this measure may be used to determine if the computed solution is acceptable.

A resultant of two polynomials is a scalar function of their coefficients that is exactly zero if and only if they have a non–constant common divisor. The resultant is evaluated as the determinant of a matrix, called the resultant matrix, and the rank deficiency of this matrix is equal to the degree of the greatest common divisor (GCD) of the polynomials. Furthermore, the coefficients of the GCD may be evaluated by reducing the resultant matrix to row echelon form. The numerical computation of the resultant of two (or more) polynomials is difficult because of the requirement that the determinant of the resultant matrix be *exactly* zero; since the roots of a polynomial are unaltered by scaling its coefficients by an arbitrary non–zero constant, the resultant matrix and its determinant can also be scaled arbitrarily, and thus a small value of the determinant of this matrix does not imply that there exist one or more roots from each polynomial that are close.

The practical applications of resultants generate entries of the resultant matrix that are specified within a tolerance; they must be interpreted as typical values, and another set of data that lie within the tolerance is also permissible. This family of permissible input data yields a family of solutions, and this leads naturally to the numerical condition of the equation that relates the sets of data and solutions. This concept is well–developed for the standard linear algebraic equation $Ax = b$, such that a computationally reliable solution can be obtained even if this equation is ill–conditioned. Since the solution is a continuous function of the input, the perturbed input $b + \delta b$, which also lies in the family of permissible data, will generate a change δx in the solution. The perturbed solution $x + \delta x$ is acceptable if the equation is well–conditioned.

The situation is more involved when resultants are implemented in a floating point environment because, as noted above, the degree of the GCD is equal to the rank deficiency of the resultant matrix, and the rank of a matrix (an integer quantity) is not a continuous function of the data (which are floating point numbers). Since the rank of a matrix is not defined in a floating point environment, the concepts that are applicable to the linear algebraic equation $Ax = b$ must be discarded or modified. If the data are perfect and the polynomials have a non–constant common divisor, the effect of roundoff error due to floating point may cause the *numerical rank* of the resultant matrix to be ill–defined. This problem is compounded when there is an error tolerance on the polynomial coefficients because the family of permissible data may allow the rank of the resultant matrix to take one of several values, depending on the exact perturbation of the polynomial coefficients. Moreover, if the change in the polynomial coefficients is small and causes a reduction in the rank of the resultant matrix, then this matrix is ill–conditioned because it is 'near' a matrix of lower rank.

This paper considers the situation that arises when the data are perfect, and thus consideration is restricted to the effect of roundoff error on the computation of the resultant of two polynomials. It is shown that the numerical condition of the resultant matrix may be significantly improved by representing the polynomials in the Bernstein basis rather than the power basis. There exist several different resultants, including the Sylvester, Bézout and companion matrix resultants, but only the companion matrix resultant is considered in this paper. Some of the results, for example, the expression for the condition number of a resultant matrix, may be applied to the other resultants, and preliminary computational results show that the numerical condition of the Bézout resultant matrix may be significantly improved by expressing the polynomials in the Bernstein basis.

A review of previous work is considered in section 1 and the method that is currently used to calculate the resultant of two Bernstein polynomials is described. In particular, it is shown that this method reduces to the calculation of the resultant of two power basis polynomials. Although this technique is theoretically correct, it cannot be justified numerically because one of the advantages of the Bernstein basis – its enhanced numerical stability – is lost. It is therefore advantageous to use the Bernstein basis exclusively, such that the power basis is not utilised. This objective is achieved in section 2 because a companion matrix of a Bernstein polynomial is developed and then used to construct a resultant matrix for two Bernstein polynomials.

A comparison of the numerical condition of two resultant matrices requires that an expression for the condition number of a resultant matrix for two polynomials that are expressed in an arbitrary basis be defined. This is considered in section 3 and it is shown that this measure is easy to compute and yields a simple geometric interpretation. This measure is used in section 4 to compare the numerical condition of the companion matrix resultant C_B of two Bernstein polynomials, and the numerical condition of the resultant matrix C_P that is obtained when a parameter substitution is used to transform, in a weak sense, each Bernstein polynomial to its power basis form. The computational results show that C_B is better conditioned than C_P, possibly by several orders of magnitude, and it may therefore be desirable to transform the power basis resultant matrix to its Bernstein form if the polynomials are specified in the power basis. This topic is considered in section 5, and the numerical condition of the equation that defines the transformation is reviewed. A discussion of future work is contained in section 6.

1. Previous work

The Bernstein basis is used for the representation of curves and surfaces in computer–aided geometric design because of its elegant geometric properties and simple algorithms that are available for processing it [7], and it is therefore necessary to develop resultants for Bernstein polynomials. This has been achieved hitherto by a simple parameter transformation that enables a Bernstein polynomial to be transformed, in a weak sense, to a power basis polynomial, such that a resultant matrix for two power basis polynomials may be used.

The parameter substitution

$$t = \frac{x}{1-x}, \qquad x \neq 1, \tag{1.1}$$

transforms the Bernstein polynomial

$$p(x) = \sum_{i=0}^{n} a_i \binom{n}{i} (1-x)^{n-i}\, x^i,$$

to the power basis polynomial

$$q(t) = (1+t)^n\, p\left(\frac{t}{1+t}\right) = \sum_{i=0}^{n} c_i t^i, \qquad c_i = a_i \binom{n}{i}, \qquad t \neq -1,$$

and hence the resultant of the Bernstein polynomials $r(x)$ and $s(x)$, with coefficients $\{r_i\}_{i=0}^{n}$ and $\{s_i\}_{i=0}^{m}$, respectively, may be determined by computing the resultant of the power basis polynomials whose coefficients

are $\{r_i\binom{n}{i}\}_{i=0}^{n}$ and $\{s_i\binom{m}{i}\}_{i=0}^{m}$. Clearly, if $x = x_0$ is a comon root of the Bernstein polynomials $r(x)$ and $s(x)$, then $t = t_0 = \frac{x_0}{1-x_0}$ is a common root of the power basis polynomials whose coefficients are $\{r_i\binom{n}{i}\}_{i=0}^{n}$ and $\{s_i\binom{m}{i}\}_{i=0}^{m}$. This method may be used for theoretical investigation, but the poor numerical condition of the power basis suggests that the change of polynomial basis that is implied by the substitution (1.1) will yield unsatisfactory results in a floating point environment. It is therefore desirable to develop a resultant matrix for Bernstein polynomials, and thus retain the numerical superiority of the Bernstein basis, and this is considered in the next section.

2. A companion matrix resultant for two Bernstein polynomials

A companion matrix C of a polynomial $p(x)$, expressed in an arbitrary basis, satisfies

$$\det(C - \lambda I) = p(\lambda),$$

where the structure of C is explicitly defined for each basis. Clearly, the eigenvalues of C are identically equal to the roots of $p(x)$. It is shown in this section that a resultant matrix for two Bernstein polynomials can be developed from the companion matrix of one of these polynomials, and the coefficients of the other polynomial. The matrix C assumes a particularly simple form when $p(x)$ is expressed in the power basis, and the following theorem [8] shows that the companion matrix of a Bernstein polynomial has a more complex structure.

Theorem 1 *Consider the matrices A and E, both of order n,*

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix}, \qquad (1.2)$$

and

$$E = \begin{bmatrix} \frac{\binom{n}{1}}{\binom{n}{0}} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{\binom{n}{2}}{\binom{n}{1}} & 1 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & \frac{\binom{n}{n-1}}{\binom{n}{n-2}} & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \cdots & -a_{n-2} & -a_{n-1} + \frac{\binom{n}{n}}{\binom{n}{n-1}} \end{bmatrix}. \qquad (1.3)$$

Then

$$\det(A - \lambda E) = (-1)^n \sum_{i=0}^{n} a_i \binom{n}{i} (1-\lambda)^{n-i} \lambda^i, \qquad a_n = 1,$$

and thus if $\det E \neq 0$*, then* $M = E^{-1}A$ *is a companion matrix of the Bernstein polynomial*

$$p(\lambda) = (-1)^n \sum_{i=0}^{n} a_i \binom{n}{i} (1-\lambda)^{n-i} \lambda^i, \qquad a_n = 1.$$

The condition $a_n = 1$ *may be generalised to* $a_n \neq 0$*, and it is therefore assumed that* $\lambda_0 = 1$ *is not a root of* $p(\lambda)$*.*

It follows from (1.2) and (1.3) that $M = E^{-1}A = (F+A)^{-1}A$ where

$$F = \operatorname{diag}\left[\begin{array}{ccccc} \frac{\binom{n}{1}}{\binom{n}{0}} & \frac{\binom{n}{2}}{\binom{n}{1}} & \cdots & \frac{\binom{n}{n-1}}{\binom{n}{n-2}} & \frac{\binom{n}{n}}{\binom{n}{n-1}} \end{array} \right],$$

and a closed form expression for E^{-1} is developed in [8]. This yields

$$M = E^{-1}A = (F+A)^{-1} A = \left(I + \frac{De_n a^T}{\tau} \right) DA,$$

where the elements of D are

$$d_{i,k+i} = \begin{cases} \frac{(-1)^k}{\prod_{m=i}^{i+k} c_{mm}} & 1 \leq i \leq n, \quad 0 \leq k \leq n-i, \\ 0 & \text{otherwise,} \end{cases}$$

the elements c_{ij} are

$$c_{ij} = \begin{cases} \frac{\binom{n}{i}}{\binom{n}{i-1}} & \text{if } i = j, \\ 1 & \text{if } i+1 = j, \\ 0 & \text{otherwise,} \end{cases}$$

e_n is the nth standard basis vector, $\tau = \det E$ and

$$a^T = \left[\begin{array}{ccccc} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \end{array} \right].$$

The following two theorems show that a companion matrix of a polynomial may be used to construct a resultant matrix for a pair of polynomials.

Theorem 2 *Let $r(x)$ and $s(x)$ be two Bernstein polynomials with coefficients $\{r_j\}_{j=0}^{n}, r_n = 1$, and $\{s_j\}_{j=0}^{m}$, respectively,*

$$r(x) = \sum_{j=0}^{n} r_j \binom{n}{j} (1-x)^{n-j} x^j, \quad s(x) = \sum_{j=0}^{m} s_j \binom{m}{j} (1-x)^{m-j} x^j. \tag{1.4}$$

If M is a companion matrix of the polynomial $r(x)$ and the eigenpairs of M are $\{\lambda_i, x_i\}_{i=1}^{n}$, then

$$\det(s(M)) = \prod_{i=1}^{n} s(\lambda_i), \tag{1.5}$$

and thus the determinant of $s(M)$ is equal to zero if and only if λ_i is a root of $s(x)$. Since the eigenvalues $\{\lambda_i\}_{i=1}^{n}$ are the roots of $r(x)$, it follows that $s(M)$ is a resultant matrix for the polynomials $r(x)$ and $s(x)$.

Proof Consider the matrix polynomial

$$s(M) = \sum_{j=0}^{m} s_j \binom{m}{j} (I-M)^{m-j} M^j.$$

Since the eigenpairs of M are $\{\lambda_i, x_i\}_{i=1}^{n}$, the eigenpairs of

$$\binom{m}{j} (I-M)^{m-j} M^j,$$

are

$$\left\{ \binom{m}{j} (1-\lambda_i)^{m-j} \lambda_i^j, x_i \right\}_{i=1}^{n},$$

and thus

$$\binom{m}{j} (I-M)^{m-j} M^j x_i = \binom{m}{j} (1-\lambda_i)^{m-j} \lambda_i^j x_i, \qquad i = 1, \ldots, n.$$

It follows that for $i = 1, \ldots, n$,

$$\sum_{j=0}^{m} s_j \binom{m}{j} (I-M)^{m-j} M^j x_i = \sum_{j=0}^{m} s_j \binom{m}{j} (1-\lambda_i)^{m-j} \lambda_i^j x_i,$$

and thus the eigenvalues of $s(M)$ are

$$s(\lambda_i) = \sum_{j=0}^{m} s_j \binom{m}{j} (1-\lambda_i)^{m-j} \lambda_i^j, \qquad i = 1, \ldots, n,$$

and thus (1.5) is obtained. □

The following theorem enables the degree and coefficients of the GCD of $r(x)$ and $s(x)$ to be calculated [2, 5].

Theorem 3 *Let $w(x)$ be the GCD of $r(x)$ and $s(x)$, which are defined in (1.4), and let M be the companion matrix of $r(x)$. Then*

1. *The degree of $w(x)$ is equal to $n - \text{rank } s(M)$.*
2. *The coefficients of $w(x)$ are proportional to the last row of $s(M)$ after it has been reduced to row echelon form.*

3. A condition number of a resultant matrix

A normwise condition number of the resultant matrix $s(M)$ is developed and it is shown that it is independent of the arbitrary scale factor that can be applied to the polynomial $s(x)$. An expression for this condition number is stated in the following theorem. The result has been specialised to square matrices because a resultant matrix is always of this form.

Theorem 4 *Let $X \in \mathbb{R}^{n\times n}$ be of rank r, and let USV^T be its singular value decomposition. Let $X_k = US_kV^T$ where $k < r$, and let $S_k \in \mathbb{R}^{n\times n}$ be the diagonal matrix with elements*

$$\begin{pmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_k & 0 & 0 & \cdots & 0 \end{pmatrix}, \qquad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0.$$

Then the rank of X_k is k, and

$$\sigma_{k+1} = \|X - X_k\|_2 = \min_{\text{rank }(Y)=k} \|X - Y\|_2 .$$

This theorem states that if a square matrix X is of order n and rank $r \leq n$, then the nearest matrix Y, of rank $k < r$, to X is obtained by setting $\sigma_i = 0, i = k+1, \ldots, r$, where σ_i is the ith singular value of X. Furthermore, the minimum normwise distance between X and Y is σ_{k+1}.

This expression for the minimum normwise distance between a matrix X and the nearest matrix Y of specified lower rank cannot be used to determine the numerical condition of $s(M)$ because it is not scale–invariant. In particular, the polynomial $r(x)$ is monic and therefore normalised, but the polynomial $s(x)$ can be scaled arbitrarily. It follows that if $s(x)$ is scaled to $\alpha s(x)$, then the singular values of $s(M)$ are also scaled by α, and thus the minimum normwise distance between $s(M)$ and the nearest matrix of lower rank is also scaled by α. This is

unsatisfactory, but this deficiency is easily overcome by defining a scale invariant measure, called the *normalised distance to singularity.* The normalised distance to singularity between the matrices X and X_k that are defined in theorem 4 is obtained by normalising σ_{k+1} by $\|X\|_2 = \sigma_1$,

$$\frac{\sigma_{k+1}}{\sigma_1}, \tag{1.6}$$

which may be defined as the reciprocal of the condition number of $X_{k+1} = US_{k+1}V^T$.

The application of theorem 4 to the determination of the numerical condition of a resultant matrix requires that a unit loss of rank be considered, and thus $k = r - 1$. It follows from (1.6) that if $s(M)$ has rank r, the normalised distance to singularity of $s(M)$ is

$$d(s(M)) = \frac{\sigma_r\left(s(M)\right)}{\sigma_1\left(s(M)\right)}. \tag{1.7}$$

The parameter substitution (1.1) may be used to transform (in a weak sense) a Bernstein polynomial to the power basis, and its application to the polynomials (1.4) yields the polynomials

$$\tilde{r}(x) = \sum_{j=0}^{n} \tilde{r}_j x^j, \qquad \tilde{r}_j = r_j \binom{n}{j},$$

and

$$\tilde{s}(x) = \sum_{i=0}^{m} \tilde{s}_i x^i, \qquad \tilde{s}_i = s_i \binom{m}{i},$$

respectively. If P is a companion matrix of $\tilde{r}(x)$, then

$$\tilde{s}(P) = \sum_{i=0}^{m} \tilde{s}_i P^i,$$

is a resultant matrix for $\tilde{s}(x)$ and $\tilde{r}(x)$, and it follows from (1.7) and its equivalent for $\tilde{s}(P)$ that the ratio of the normalised distance to singularity of $\tilde{s}(P)$ to the normalised distance to singularity of $s(M)$ is

$$d(\tilde{s}(P), s(M)) = \frac{\sigma_r\left(\tilde{s}(P)\right)}{\sigma_1\left(\tilde{s}(P)\right)} \frac{\sigma_1\left(s(M)\right)}{\sigma_r\left(s(M)\right)}. \tag{1.8}$$

This measure is used in section 4 to compare the normalised distance to singularity of $\tilde{s}(P)$ and $s(M)$.

4. Computational experiments

This section contains several examples that enable a comparison to be made of the resultant matrix of the Bernstein polynomials $r(x)$ and $s(x)$, and the power basis resultant matrix that is obtained when the substitution (1.1) is used to transform $r(x)$ and $s(x)$ to $\tilde{r}(x)$ and $\tilde{s}(x)$, respectively. The ratio (1.8) is used to calculate the ratio of the normalised distance to singularity of $\tilde{s}(P)$ and $s(M)$. Since use of the Bernstein basis necessarily implies that interest is restricted to the unit interval $I = \{x : 0 \leq x \leq 1\}$, the polynomials $r(x)$ and $s(x)$ are chosen so that they have at least one common root in this interval.

In all the examples, the polynomial $r(x)$ is defined as the Wilkinson polynomial,

$$r(x) = \prod_{i=1}^{19} \left(x - \frac{i}{20} \right) = \sum_{i=0}^{19} r_i \binom{19}{i} (1-x)^{19-i} x^i, \tag{1.9}$$

and different polynomials $s_i(x)$ are used. The upper index is 19 and not its more usual value of 20 because, as noted in theorem 1, the companion matrix M must not have an eigenvalue $\lambda_0 = 1$. The matrices $s_i(M)$ and $\tilde{s}_i(P)$ were computed for several polynomials $s_i(x)$, and the ratio of the normalised distances to singularity $d(\tilde{s}_i(P), s_i(M))$ was calculated. The results are shown in table 1.1.

Table 1.1. The ratio $d(\tilde{s}_i(P), s_i(M))$ for the polynomials (1.9) and $s_i(x)$.

Experiment	*Polynomial* $s_i(x)$	$d(\tilde{s}_i(P), s_i(M))$
1	$s_1(x)$	2.728×10^{-2}
2	$s_2(x)$	2.194×10^{-15}
3	$s_3(x)$	7.038×10^{-13}
4	$s_4(x)$	1.828×10^{-4}
5	$s_5(x)$	1.434×10^{-13}
6	$s_6(x)$	1.747×10^{-12}
7	$s_7(x)$	2.235×10^{-12}
8	$s_8(x)$	2.094×10^{-9}
9	$s_9(x)$	8.449×10^{-1}
10	$s_{10}(x)$	6.785×10^{1}

The polynomials $s_i(x), i = 1, \ldots, 10$, in this table are :

$$\begin{aligned}
s_1(x) &= (x-0.80)(x-0.90)(x-0.95)(x-0.99)(x-1.01) \\
s_2(x) &= (x+0.05)(x+0.025)(x-0.05)^3 \\
s_3(x) &= (x-0.20)^2(x-0.40)^2(x-0.60)^2(x-0.79)(x-0.80)
\end{aligned}$$

$$
\begin{aligned}
s_4(x) &= (x+2.0)(x+1.0)(x-0.30)(x-0.50)(x-0.70)(x-0.80)\times \\
&\quad (x-0.85)(x-0.86)(x-0.87)(x-0.95)^2(x-1.01) \\
s_5(x) &= (x-0.02)(x-0.04)(x-0.06)(x-0.08)(x-0.10)^2 \\
s_6(x) &= (x-0.05)^3(x-0.10)^3(x-0.15)^3 \\
s_7(x) &= (x-0.40)^2(x-0.425)(x-0.45)^2(x-0.475)^2(x-0.50)^3 \\
s_8(x) &= (x+3.0)(x+1.0)(x-0.40)^2(x-0.41)^2(x-0.94)\times \\
&\quad (x-0.95)^2(x-0.96)(x-1.50)(x-1.60) \\
s_9(x) &= (x-0.50)(x-0.55)(x-0.60)(x-0.65)(x-0.70)\times \\
&\quad (x-0.75)(x-0.80)(x-0.85)(x-0.90)(x-0.95) \\
s_{10}(x) &= (x-0.95)^2(x-0.97)^2(x-0.99)^2(x-1.01)^2
\end{aligned}
$$

The results in table 1.1 show that the Bernstein basis resultant matrix $s(M)$ is, in most situations, better conditioned than the power basis resultant matrix $\tilde{s}(P)$ because it is further away from singularity, but the ratio $d(\tilde{s}_{10}(P), s_{10}(M))$ shows that $s(M)$ is not guaranteed to be better conditioned than $\tilde{s}(P)$. More computational results are in [8], and these two sets of results show that, in general, the parameter substitution (1.1) is not recommended because it causes the resultant matrix to be ill–conditioned. It must be noted that these results do not define an exact comparison between the power and Bernstein bases because this parameter substitution is not a basis transformation in the strict sense. A computational experiment that compares the numerical condition of the power and Bernstein resultant matrices when the basis transformation is performed exactly is in [8], and it is noted that the Bernstein basis resultant matrix is much better conditioned.

5. The basis transformation of the companion matrix resultant

The results of section 4 show that it is numerically advantageous to compute the resultant of two polynomials when they are expressed in the Bernstein basis rather than when they are expressed in the power basis. The power basis is the natural choice for the representation of curves and surfaces in the applications (apart from computer graphics) that are mentioned in the introduction, and the computation of the resultant of two polynomials in a floating point environment therefore requires that a basis transformation of the resultant matrix be performed.

Consider two power basis polynomials $f(x)$ and $g(x)$, and let $r(x)$ and $s(x)$ be, respectively, their Bernstein forms,

$$r(x) = \sum_{i=0}^{n} r_i \binom{n}{i} (1-x)^{n-i} x^i, \qquad f(x) = \sum_{i=0}^{n} f_i x^i,$$

where $r(x) \equiv f(x)$, and

$$s(x) = \sum_{i=0}^{m} s_i \binom{m}{i} (1-x)^{m-i} x^i, \qquad g(x) = \sum_{i=0}^{m} g_i x^i,$$

where $s(x) \equiv g(x)$. It is important to note that if $f(x)$ and $g(x)$ are given, the resultant of $r(x)$ and $s(x)$ cannot be computed in a floating point environment by performing a basis transformation on $f(x)$ and $g(x)$ and then using the theory in section 2 to compute $s(M)$ where M is the companion matrix of $r(x)$. This method decouples the polynomials – the pairs of polynomials $(r(x), f(x))$ and $(s(x), g(x))$ are treated independently, but the resultant matrix $s(M)$ is a function of r_i and s_j, and thus the error that arises from the cross product terms $r_i s_j$ in $s(M)$ must be considered for a complete error analysis.

Let P be the companion matrix of the power basis polynomial $f(x)$, such that it has the same form as A in (1.2). An identical proof to that of theorem 2 shows that $g(P)$ is a companion matrix resultant for the power basis polynomials $f(x)$ and $g(x)$. It is shown in [9] that the resultant matrices $s(M)$ and $g(P)$ are related by the similarity transformation

$$g(P) = B s(M) B^{-1}, \tag{1.10}$$

where the entries $\{b_{jk}\}_{j,k=0}^{n-1}$ of B and $\left\{b_{jk}^{-1}\right\}_{j,k=0}^{n-1}$ of B^{-1} are given by, respectively,

$$b_{jk} = \begin{cases} \frac{n-k}{n} \frac{\binom{k}{j}}{\binom{n-1}{j}}, & j = 0, \ldots, n-1;\ k = j, \ldots, n-1, \\ 0, & k < j, \end{cases} \tag{1.11}$$

and

$$b_{jk}^{-1} = \begin{cases} (-1)^{k-j} \binom{n}{j} \binom{n-1-j}{k-j}, & j = 0, \ldots, n-1;\ k = j, \ldots, n-1, \\ 0, & k < j. \end{cases} \tag{1.12}$$

The eigenvalues of P are equal to the eigenvalues of M since these are companion matrices of the same polynomial but expressed in different

bases. The transformation (1.10) is obtained by deriving an expression for the eigenvectors $y_i, i = 1, \ldots, n$, and $x_i, i = 1, \ldots, n$, of, respectively, P and M, and noting that the elements of each of the eigenvectors y_j and x_j form a basis for polynomials of degree $n-1$. It follows, therefore, that there exists a square non–singular matrix B such that $y_j = Bx_j$, and this equation allows the closed form expression (1.11) to be derived. It is shown in [9] that B is a totally non–negative matrix, and BB^T and B^TB are oscillation matrices.

The motivation for the transformation (1.10) of the power basis resultant matrix $g(P)$ to its Bernstein form $s(M)$ is the improvement in the numerical condition, as noted in the examples in section 4. This advantage of performing all computations in the Bernstein basis is retained only if (1.10) is well–conditioned, such that errors due to floating point computations and uncertainties in the data are not magnified during the computation of $s(M)$ from $g(P)$. If, however, this equation is ill–conditioned, then the advantages of transforming to the Bernstein basis may be cancelled by the potentially large magnification of these errors. It is therefore necessary to determine the numerical condition of (1.10), and this is considered in detail in [9]. It is shown that this equation can be reduced to a linear algebraic equation by the Kronecker product, and this allows a sharp upper bound of the condition number of this equation to be obtained[1]. It is shown in [9] that this condition number increases rapidly with n, and thus the equation may be ill–conditioned, even for matrices B of low order. It is therefore concluded that the resultant matrix should be formed *a priori* in the Bernstein basis, such that the power basis is not used.

The Bernstein basis is the common choice for the representation of polynomials in computer graphics and computer–aided geometric design, but the power basis is commonly used in other applications. The results in [8] and [9] show that improved numerical answers will be obtained if the Bernstein basis were also the natural choice for the representation of polynomials in these other applications.

6. Discussion

This paper has considered some issues that arise during the computational implementation of resultants in a floating point environment. It was shown that it is desirable to compute the resultant of two polynomials that are expressed in the Bernstein basis rather than perform

[1]The author wishes to thank Professor N. Higham of Manchester University for introducing him to the Kronecker product.

a parameter substitution that implements a basis transformation, in a weak sense, to the power basis. Attention was restricted to the companion matrix resultant, and it is therefore desirable to construct other resultant matrices for the Bernstein basis and determine if they are also numerically better conditioned than their power basis equivalents. Also, the numerical condition of the Bernstein form of these types of resultant matrix must be compared in order that the most stable resultant matrix be used.

The results that were developed in this paper for the companion matrix resultant are also applicable to other resultant matrices. For example, the normalised distance to singularity (1.7) can be applied to the Bézout resultant matrix, but this measure is not scale invariant for the Sylvester resultant matrix because the coefficients of the polynomials are decoupled in this matrix [10]. Also, the basis transformation equation (1.10) generalises to $S = XZY$ for the Sylvester and Bézout resultant matrices, where S and Z are the resultant matrices in the power and Bernstein bases, and X and Y are transformation matrices whose elements are given by formulae that are similar to those in (1.11) and (1.12).

References

[1] J. F. Canny. *The Complexity of Robot Motion Planning.* The MIT Press, Cambridge, USA, 1988.

[2] R. N. Goldman, T. W. Sederberg, and D. C. Anderson. Vector elimination : A technique for the implicitization, inversion and intersection of planar parametric rational polynomial curves. *Computer Aided Geometric Design*, 1:327–356, 1984.

[3] J. T. Kajiya. Ray tracing parametric patches. *Computer Graphics*, 16:245–254, 1982.

[4] D. Manocha. Numerical methods for solving polynomial equations. In D. Cox and B. Sturmfels, editors, *Proceedings of Symposia in Applied Mathematics, volume 53*, Applications of Computational Algebraic Geometry, pages 41–66. AMS, Rhode Island, USA, 1998.

[5] Y. De Montaudouin and W. Tiller. The Cayley method in computer aided geometric design. *Computer Aided Geometric Design*, 1:309–326, 1984.

[6] S. Petitjean. Algebraic geometry and computer vision : Polynomial systems, real and complex roots. *Journal of Mathematical Imaging and Vision*, 10:191–220, 1999.

[7] T. Sederberg. Applications to computer aided geometric design. In D. Cox and B. Sturmfels, editors, *Proceedings of Symposia in Applied Mathematics, volume 53*, Applications of Computational Algebraic Geometry, pages 67–89. AMS, Rhode Island, USA, 1998.

[8] J. R. Winkler. A companion matrix resultant for Bernstein polynomials, 2002. Submitted.

[9] J. R. Winkler. The transformation of the companion matrix resultant between the power and Bernstein polynomial bases, 2002. Submitted.

[10] J. R. Winkler and R. N. Goldman. The Sylvester resultant matrix for Bernstein polynomials, 2002. Submitted.