Article:

Bourne, M., Winkler, J.R. and Su, Y. (2017) A non-linear structure-preserving matrix method for the computation of the coefficients of an approximate greatest common divisor of two Bernstein polynomials. Journal of Computational and Applied Mathematics, 320. pp. 221-241. ISSN 0377-0427

https://doi.org/10.1016/j.cam.2017.01.035

A non-linear structure-preserving matrix method for the computation of the coefficients of an approximate greatest common divisor of two Bernstein polynomials

Martin Bourne, [a,⋆] Joab R. Winkler, [a] Su Yi [b]

[a] *The University of Sheffield, Department of Computer Science, Regent Court, 211 Portobello, Sheffield S1 4DP, United Kingdom*

[b] *Institute of High Performance Computing, 1 Fusionopolis Way, 16-16 Connexis North, Singapore 138632*

mbourne1@sheffield.ac.uk, j.r.winkler@sheffield.ac.uk, suyi@ihpc.a-star.edu.sg

Abstract

This paper describes a non-linear structure-preserving matrix method for the computation of the coefficients of an approximate greatest common divisor (AGCD) of degree t of two Bernstein polynomials $f(y)$ and $g(y)$. This method is applied to a modified form $S_t(f,g)Q_t$ of the tth subresultant matrix $S_t(f,g)$ of the Sylvester resultant matrix $S(f,g)$ of $f(y)$ and $g(y)$, where Q_t is a diagonal matrix of combinatorial terms. This modified subresultant matrix has significant computational advantages with respect to the standard subresultant matrix $S_t(f,g)$, and it yields better results for AGCD computations. It is shown that $f(y)$ and $g(y)$ must be processed by three operations before $S_t(f,g)Q_t$ is formed, and the consequence of these operations is the introduction of two parameters, α and θ, such that the entries of $S_t(f,g)Q_t$ are non-linear functions of α, θ and the coefficients of $f(y)$ and $g(y)$. The values of α and θ are optimised, and it is shown that these optimal values allow an AGCD that has a small error, and a structured low rank approximation of $S(f,g)$, to be computed.

Key words: Approximate greatest common divisor; Sylvester resultant matrix; structure-preserving matrix methods

[⋆] Martin Bourne was supported by a studentship from The Agency for Science, Technology and Research (A*STAR), Singapore, and The University of Sheffield.

1 Introduction

The need to calculate the points of intersection of two polynomial curves $p(x,y)=0$ and $q(x,y)=0$ arises frequently in computer aided geometric design (CAGD), and an important part of this calculation is the computation of the greatest common divisor (GCD) of $p(x,y)$ and $q(x,y)$. Resultant matrices are frequently used for this computation, and these matrices and other polynomial computations also occur in robotics [5], computer vision [6], computational geometry, for example, the implicitization of parametric curves and surfaces [9] and the construction of surfaces [10,11], control theory [13] and the computation of multiple roots of a polynomial [17,22]. There are several resultant matrices, including the Sylvester, Bézout and companion resultant matrices, of which the Sylvester matrix is the most popular, presumably because its entries are linear, even though it is larger than the Bézout and companion matrices. This property of the entries of the Sylvester matrix must be compared with the entries of the Bézout and companion matrices, which are bilinear and non-linear, respectively [1].

There has been extensive work on the theoretical and numerical properties of resultant matrices for polynomials expressed in the power basis, but much less work has been performed on resultant matrices for polynomials expressed in the Bernstein basis, which is of particular interest in CAGD because of its widespread use in this application. Explicit forms for the entries of the Bézout resultant matrix [3], the companion resultant matrix [16] and the Sylvester resultant matrix [18] of the Bernstein polynomials $\hat{f}(y)$ and $\hat{g}(y)$,

$$\hat{f}(y)=\sum_{i=0}^{m}\hat{a}_i\binom{m}{i}(1-y)^{m-i}y^i \quad \text{and} \quad \hat{g}(y)=\sum_{i=0}^{n}\hat{b}_i\binom{n}{i}(1-y)^{n-i}y^i, \qquad (1)$$

have been developed but there has been significantly less investigation into their numerical properties. These properties are worthy of consideration because these resultant matrices contain combinatorial terms, and thus even if the magnitude of the coefficients $\hat{a}_i$ and $\hat{b}_j$ is of order one, the entries of these matrices may span several orders of magnitude, which may cause numerical problems.

It was noted above that the computation of the points of intersection of two polynomial curves requires the GCD of their polynomial forms. It is necessary to distinguish between polynomials whose coefficients are, and are not, subject to error because the GCD is defined for exact polynomials only, but the coefficients of polynomials in practical problems are subject to error. Inexact (noisy) polynomials must therefore be considered, and this leads to an approximate greatest common divisor (AGCD) of two polynomials whose coefficients are subject to error. This paper considers, therefore, the computation of the coefficients of an AGCD of degree t of the noisy forms $f(y)$ and $g(y)$ of the exact polynomials $\hat{f}(y)$ and $\hat{g}(y)$,

$$f(y) = \sum_{i=0}^{m} a_i \binom{m}{i} (1-y)^{m-i} y^i \quad \text{and} \quad g(y) = \sum_{i=0}^{n} b_i \binom{n}{i} (1-y)^{n-i} y^i,$$

by applying the method of structured non-linear total least norm (SNTLN) [12] to a modified form $S_t(f,g)Q_t$ of the tth subresultant matrix $S_t(f,g)$, where Q_t is a diagonal matrix of combinatorial terms. The calculation of the degree t is considered in [4,23] and it is assumed this calculation has been performed, and the value of t is therefore known. The computation of the GCD of $\hat{f}(y)$ and $\hat{g}(y)$ by Euclid's algorithm is investigated in [14] and several stopping criteria for the termination of the divisions in finite precision arithmetic are considered. The results show that roundoff error can cause a serious deterioration in the computed GCD and that a numerically robust method is required for this computation.

One application of the work described in this paper is, as noted above, the calculation of the points of intersection of two curves. Another application is the computation of multiple roots of a polynomial, where the multiplicity of a root defines the smoothness of curves and surfaces at their intersection point. These roots are important for mechanical design because the stresses in sharp corners of an object may become very large, and much larger than in its interior, such that the object may fracture when in operation. These high stress levels can be reduced substantially by rounding off the corners of the intersecting curves and surfaces, which requires the formation of a blending surface. The simplest situation arises when the blending surface reduces to a blending curve, the formation of which requires the calculation of multiple roots a polynomial $p(y)$. Since the coefficients of this polynomial are, in practical problems, corrupted by noise, its roots are, in general, simple. This property does not, however, reflect design intent - a smooth intersection - but if the noise is sufficiently small, then $p(y)$ is near another polynomial $\tilde{p}(y)$ that has one or more multiple roots, which can therefore be used for the design of a blending curve. Structured low rank approximations of the Sylvester resultant matrices of $p^{(k)}(y)$ and $p^{(k+1)}(y)$, where $p^{(k)}(y)$ is the kth derivative of $p(y)$, $k = 0, 1, \ldots,$ allow the polynomial $\tilde{p}(y)$ and its multiple roots to be computed, and this has been considered for power basis polynomials [17,22]. The work in this paper is therefore a necessary requirement for the extension of this polynomial root solver to the Bernstein basis.

There are important differences between the GCD of $\hat{f}(y)$ and $\hat{g}(y)$, and an AGCD of $f(y)$ and $g(y)$. For example, the GCD of two exact polynomials is unique up to a non-zero constant multiplier, but an AGCD of two inexact polynomials is not unique because it can be defined in several different ways and it is a function of the relative error of the coefficients of $f(y)$ and $g(y)$ [2,24]. It cannot, however, be assumed in practical problems that this error is uniformly distributed across the coefficients, and the implications of this property for AGCD computations are considered in [4]. It is therefore assumed in this paper that the upper bound of the relative error of the coefficients of $f(y)$ and $g(y)$ is a uniformly distributed random number that spans two orders of magnitude.

The Sylvester matrix of two Bernstein polynomials is reviewed in Section 2, and the application of the method of SNTLN to the computation of an AGCD of $f(y)$ and $g(y)$ is considered in Section 3. Examples of the application of the method of SNTLN to the computation of the coefficients of an AGCD of degree t are in Section 4, and Section 5 contains a summary of the paper.

The computation of a structured low rank approximation of the Sylvester matrix of two power basis polynomials has been considered by several researchers [8,19,20,25,26]. The computation of this matrix, $S(f,g)$, for the Bernstein polynomials $f(y)$ and $g(y)$ is sufficiently different to merit a separate investigation because, apart from the importance of the Bernstein basis in CAGD, non-trivial numerical issues that do not occur with power basis polynomials must be considered. In particular, $S(f,g)$ is not Tœplitz, unlike its power basis equivalent, and the combinatorial terms in the Bernstein basis imply that the ratio of the maximum entry to the minimum entry of $S(f,g)$ may be large, even if the ratio of the maximum coefficient to the minimum coefficient of (f,g) is of order one. Also, the update formula of the QR decomposition can be applied to the Sylvester matrix and its subresultant matrices of two power basis polynomials, but the more involved structure of $S(f,g)$ and its subresultant matrices implies it cannot be used for these matrices. It therefore follows that computations with subresultant matrices of Bernstein basis polynomials are more expensive than with their power basis equivalents.

The results of this paper are now summarised:

- A modified form of $S(f,g)$ must be used for the computation of a structured low rank approximation of this matrix because the modified form minimises the adverse effects of large combinatorial terms in the Bernstein basis functions.
- Previous work [19–21] has shown that improved results are obtained when a non-linear transformation is imposed on the independent variable y. The parameters of the transformation are optimised such that a measure of the condition number of a subresultant matrix derived from $S(f,g)$ is minimised. This minimisation must be performed for each subresultant matrix, which marks a difference between these subresultant matrices and subresultant matrices of two power basis polynomials, for which the minimisation need only be performed once because the optimal values of the parameters

are independent of the order of the subresultant matrix.

- The polynomials that result from this non-linear transformation are expressed in a basis that is similar to, but distinct from, the Bernstein basis. This change in basis does not occur when the non-linear transformation is applied to a power basis polynomial.
- The method of SNTLN is applied to two different equations in order to compute two, possibly different, structured low approximations of $S(f,g)$. The first equation is derived from a subresultant matrix of $S(f,g)$ and the second equation is derived from the equation that defines an approximate factorisation of $f(y)$ and $g(y)$ into their AGCD and coprime factors. These two structured low rank approximations are very similar and both of them may therefore be used for subsequent analysis.

2 The Sylvester matrix

This section considers the Sylvester matrix and its subresultant matrices of the Bernstein polynomials $\hat{f}(y)$ and $\hat{g}(y)$ that are defined in (1). The discussion is brief and more details are in [18,23].

The Sylvester matrix $S(\hat{f},\hat{g})$ of $\hat{f}(y)$ and $\hat{g}(y)$ is a square matrix of order $m+n$,

$$S(\hat{f},\hat{g}) = D^{-1}T(\hat{f},\hat{g}), \qquad D, T(\hat{f},\hat{g}) \in \mathbb{R}^{(m+n)\times(m+n)},$$

where

$$D^{-1} = \operatorname{diag}\left[\begin{array}{cccc} \frac{1}{\binom{m+n-1}{0}} & \frac{1}{\binom{m+n-1}{1}} & \cdots & \frac{1}{\binom{m+n-1}{m+n-1}} \end{array}\right], \tag{2}$$

and $T(\hat{f},\hat{g})$ is the Sylvester matrix of $\hat{f}(y)$ and $\hat{g}(y)$ when they are expressed in the scaled Bernstein basis [15],

$$T(\hat{f},\hat{g}) = \begin{bmatrix} \hat{a}_0\binom{m}{0} & & & \hat{b}_0\binom{n}{0} & & \\ \hat{a}_1\binom{m}{1} & \ddots & & \hat{b}_1\binom{n}{1} & \ddots & \\ \vdots & \ddots & \hat{a}_0\binom{m}{0} & \vdots & \ddots & \hat{b}_0\binom{n}{0} \\ \vdots & \ddots & \hat{a}_1\binom{m}{1} & \vdots & \ddots & \hat{b}_1\binom{n}{1} \\ \hat{a}_m\binom{m}{m} & \ddots & \vdots & \hat{b}_n\binom{n}{n} & \ddots & \vdots \\ & \ddots & \vdots & & \ddots & \vdots \\ & & \hat{a}_m\binom{m}{m} & & & \hat{b}_n\binom{n}{n} \end{bmatrix}. \tag{3}$$

Consideration of the degree and coefficients of the GCD of $\hat{f}(y)$ and $\hat{g}(y)$ leads to the subresultant matrices $S_k(\hat{f},\hat{g})$, $k = 1,\ldots,\min(m,n)$, $S_1(\hat{f},\hat{g}) = S(\hat{f},\hat{g})$. The structure and dimensions of these matrices are considered in Theorem 2.1.

Theorem 2.1 *The degree $\hat{t}$ of the GCD of $\hat{f}(y)$ and $\hat{g}(y)$ is equal to the largest integer k such that $S_k(\hat{f},\hat{g})$ is singular,*

$$\begin{aligned} \textit{rank } S_k(\hat{f},\hat{g}) &< m+n-2k+2, \quad k = 1,\ldots,\hat{t}, \\ \textit{rank } S_k(\hat{f},\hat{g}) &= m+n-2k+2, \quad k = \hat{t}+1,\ldots,\min(m,n). \end{aligned} \tag{4}$$

Proof The polynomials $\hat{f}(y)$ and $\hat{g}(y)$ have common divisors of degree $k = 1,\ldots,\hat{t}$, since the degree of their GCD is $\hat{t}$. It therefore follows that there exist polynomials $\hat{d}_k(y), \hat{u}_{m-k}(y)$ and $\hat{v}_{n-k}(y)$ of degrees $k, m-k$ and $n-k$ respectively, such that

$$\hat{f}(y) = \hat{u}_{m-k}(y)\hat{d}_k(y) \qquad \text{and} \qquad \hat{g}(y) = \hat{v}_{n-k}(y)\hat{d}_k(y), \tag{5}$$

where

$$\hat{u}_{m-k}(y) = \sum_{i=0}^{m-k} \hat{u}_{m-k,i}\binom{m-k}{i}(1-y)^{m-k-i}y^i,$$

and

$$\hat{v}_{n-k}(y) = \sum_{i=0}^{n-k} \hat{v}_{n-k,i}\binom{n-k}{i}(1-y)^{n-k-i}y^i.$$

The elimination of $\hat{d}_k(y)$ between $\hat{f}(y)$ and $\hat{g}(y)$ in (5) leads to the equation $\hat{f}(y)\hat{v}_{n-k}(y) = \hat{g}(y)\hat{u}_{m-k}(y)$, which can be written in matrix form,

$$D_k^{-1}\begin{bmatrix} \hat{a}_0\binom{m}{0} & & & \hat{b}_0\binom{n}{0} & & \\ \hat{a}_1\binom{m}{1} & \ddots & & \hat{b}_1\binom{n}{1} & \ddots & \\ \vdots & \ddots & \hat{a}_0\binom{m}{0} & \vdots & \ddots & \hat{b}_0\binom{n}{0} \\ \vdots & \ddots & \hat{a}_1\binom{m}{1} & \vdots & \ddots & \hat{b}_1\binom{n}{1} \\ \hat{a}_m\binom{m}{m} & \ddots & \vdots & \hat{b}_n\binom{n}{n} & \ddots & \vdots \\ & \ddots & \vdots & & \ddots & \vdots \\ & & \hat{a}_m\binom{m}{m} & & & \hat{b}_n\binom{n}{n} \end{bmatrix} \begin{bmatrix} \hat{v}_{n-k,0}\binom{n-k}{0} \\ \hat{v}_{n-k,1}\binom{n-k}{1} \\ \vdots \\ \hat{v}_{n-k,n-k}\binom{n-k}{n-k} \\ -\hat{u}_{m-k,0}\binom{m-k}{0} \\ -\hat{u}_{m-k,1}\binom{m-k}{1} \\ \vdots \\ -\hat{u}_{m-k,m-k}\binom{m-k}{m-k} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}, \tag{6}$$

or equivalently,

$$S_k(\hat{f},\hat{g})\mathbf{p}(\hat{u}_{m-k},\hat{v}_{n-k}) = \left(D_k^{-1}T_k(\hat{f},\hat{g})\right)\mathbf{p}(\hat{u}_{m-k},\hat{v}_{n-k}) = \mathbf{0}, \tag{7}$$

where D_k^{-1} is a square diagonal matrix of order $m+n-k+1$,

$$D_k^{-1} = \text{diag}\begin{bmatrix} \frac{1}{\binom{m+n-k}{0}} & \frac{1}{\binom{m+n-k}{1}} & \cdots & \frac{1}{\binom{m+n-k}{m+n-k}} \end{bmatrix}, \qquad D_1^{-1} = D^{-1},$$

and D^{-1} is defined in (2). The matrix $T_k(\hat{f},\hat{g}) \in \mathbb{R}^{(m+n-k+1)\times(m+n-2k+2)}$ contains the coefficients of $\hat{f}(y)$ and $\hat{g}(y)$, where $T_1(\hat{f},\hat{g}) = T(\hat{f},\hat{g})$ and $T(\hat{f},\hat{g})$ is defined in (3), and $\mathbf{p}(\hat{u}_{m-k},\hat{v}_{n-k}) \in \mathbb{R}^{m+n-2k+2}$ contains the coefficients of $\hat{u}_{m-k}(y)$ and $\hat{v}_{n-k}(y)$. Equation (6) has a non-zero solution for $k = 1,\ldots,\hat{t}$, because the degree of the GCD of $\hat{f}(y)$ and $\hat{g}(y)$ is $\hat{t}$, and $\hat{u}_{m-k}(y)$ and $\hat{v}_{n-k}(y)$ are therefore non-zero polynomials for $k = 1,\ldots,\hat{t}$. It follows that $S_k(\hat{f},\hat{g})$ is singular for these values of k, and in particular, $S_k(\hat{f},\hat{g})$ has unit rank loss for $k = \hat{t}$ because the GCD of two polynomials is unique up to a non-zero scalar multiplier. The polynomials $\hat{u}_{m-k}(y)$ and $\hat{v}_{n-k}(y)$ are, however, equal to the zero polynomial for $k = \hat{t}+1,\ldots,\min(m,n)$, because the zero solution is the only solution of (6) for these values of k. □

The vector $\mathbf{p}(\hat{u}_{m-k},\hat{v}_{n-k})$ can be written as the product of a square diagonal matrix Q_k of order $m+n-2k+2$ and a vector $\mathbf{r} = \mathbf{r}(\hat{u}_{m-k},\hat{v}_{n-k}) \in \mathbb{R}^{m+n-2k+2}$,

$$\mathbf{p}(\hat{u}_{m-k},\hat{v}_{n-k}) = Q_k\mathbf{r}(\hat{u}_{m-k},\hat{v}_{n-k}), \tag{8}$$

where

$$Q_k = \text{diag} \left[\binom{n-k}{0} \, \binom{n-k}{1} \, \cdots \, \binom{n-k}{n-k} \, \binom{m-k}{0} \, \binom{m-k}{1} \, \cdots \, \binom{m-k}{m-k} \right],$$

and

$$\mathbf{r} = \left[\hat{v}_{n-k,0} \; \hat{v}_{n-k,1} \; \cdots \; \hat{v}_{n-k,n-k} \; -\hat{u}_{m-k,0} \; -\hat{u}_{m-k,1} \; \cdots \; -\hat{u}_{m-k,m-k} \right]^T.$$

The substitution of (8) into (7) yields

$$S_k(\hat{f},\hat{g})\mathbf{p}(\hat{u}_{m-k},\hat{v}_{n-k}) = \left(D_k^{-1} T_k(\hat{f},\hat{g}) Q_k \right) \mathbf{r}(\hat{u}_{m-k},\hat{v}_{n-k}) = \mathbf{0}, \tag{9}$$

and since D_k^{-1} and Q_k are non-singular, the rank of $S_k = S_k(\hat{f},\hat{g})$ satisfies

$$\text{rank } S_k = \text{rank } D_k^{-1} T_k = \text{rank } D_k^{-1} T_k Q_k = \text{rank } T_k Q_k = \text{rank } T_k, \tag{10}$$

where $T_k = T_k(\hat{f},\hat{g})$. The rank property (4) of the subresultant matrices is therefore satisfied by all the matrices in (10), and not only $S_k(\hat{f},\hat{g})$. It is shown in [4], however, that the form $S_k(\hat{f},\hat{g})Q_k = D_k^{-1}T_k(\hat{f},\hat{g})Q_k$ is preferred for AGCD computations because its condition number is smaller than the condition numbers of the other matrices in (10). The computation of its entries requires, however, the evaluation of three combinatorial terms, which is greater than the cost of the evaluation of the entries of the other matrices in (10). This disadvantage is mitigated by the simplification of the entries of $D_k^{-1}T_k(\hat{f},\hat{g})Q_k$, such that only two combinatorial terms need be computed for each value of $k = 1, \ldots, \min(m,n)$. In particular, the combinatorial terms in the entries in the first $n-k+1$ columns of $D_k^{-1}T_k(\hat{f},\hat{g})Q_k$ can be rearranged,

$$\frac{\binom{m}{i-j}\binom{n-k}{j}}{\binom{m+n-k}{i}} = \frac{\binom{m+n-k-i}{n-k-j}\binom{i}{j}}{\binom{m+n-k}{n-k}}, \quad j = 0, \ldots, n-k, \quad i = j, \ldots, m+j, \tag{11}$$

and similarly, the combinatorial terms in the entries in the last $m-k+1$ columns of $D_k^{-1}T_k(\hat{f},\hat{g})Q_k$ can be rearranged,

$$\frac{\binom{n}{i-j}\binom{m-k}{j}}{\binom{m+n-k}{i}} = \frac{\binom{m+n-k-i}{m-k-j}\binom{i}{j}}{\binom{m+n-k}{m-k}}, \quad j = 0, \ldots, m-k, \quad i = j, \ldots, n+j, \tag{12}$$

from which it is seen that, for each value of k, the cost of the evaluation of the terms on the left hand sides of (11) and (12) is greater than the cost of the evaluation of the terms on the right hand sides. It therefore follows that $D_k^{-1}T_k(\hat{f},\hat{g})Q_k$ has the best numerical properties of the matrices in (10) and the combinatorial terms in its entries can be rearranged, such that they can be computed efficiently. It is also shown in [4] that this matrix has other advantages, including simplified expressions for the geometric means of the entries that contain the coefficients of $\hat{f}(y)$ and $\hat{g}(y)$. This operation will be required in Section 3 because the non-zero entries in the first $n-k+1$ columns, and the non-zero entries in the last $m-k+1$ columns, of $D_k^{-1}T_k(\hat{f},\hat{g})Q_k$ must be normalised by their geometric means before computations can be performed on this matrix.

It is assumed the value of t has been computed [4], and thus the computation of the coefficients of an AGCD, of degree t, of $f(y)$ and $g(y)$ is performed on $D_t^{-1}T_t(f,g)Q_t$. The next section considers the application of the method of SNTLN to this computation.

3 The method of SNTLN for the computation of the coefficients of an AGCD

The errors in the coefficients of $f(y)$ and $g(y)$ may not be known or they may only be known approximately, which may cause problems because several methods for the computation of an AGCD of $f(y)$ and $g(y)$ attempt to compute common divisors of degree k, $k = \min(m,n), \min(m,n)-1, \ldots, 2, 1$, and an error measure is computed for each value of k. The computations terminate at the largest (first) value of k for which the error measure is smaller than the upper bound ϵ of the relative error, from which it follows that an AGCD is a function of ϵ.

A different procedure is adopted in this paper because the computation of an AGCD of $f(y)$ and $g(y)$ is considered in two stages:

Stage 1 Compute the degree t of an AGCD of $f(y)$ and $g(y)$.
Stage 2 Compute the coefficients of an AGCD of degree t.

Stage 1 is considered in [4], where it is also shown that $f(y)$ and $g(y)$ must be processed by three operations before these two stages are implemented. These operations are:

1. The normalisation of the coefficients of $f(y)$ and $g(y)$ in $S_k(f,g)Q_k = D_k^{-1}T_k(f,g)Q_k$ by their geometric means, λ_k and μ_k respectively, for $k = 1,\ldots,\min(m,n)$.
2. The replacement of $g(y)$ by $\alpha_k g(y)$ where α_k is a non-zero parameter whose optimal value is computed for each value of $k = 1,\ldots,\min(m,n)$. These computations require that a linear programming problem be solved for each value of k.
3. The substitution

$$y = \theta_k w, \tag{13}$$

is made, where w is the new independent variable and θ_k is a non-zero parameter whose optimal value, for each value of $k = 1, \ldots, \min(m, n)$, is calculated from the linear programming problem from which the optimal value of α_k is computed.

The substitution (13) transforms $f(y)$ and $g(y)$ to polynomials that are expressed in a generalised form of the Bernstein basis, called the modified Bernstein basis, whose basis functions for a polynomial of degree m and parameter θ are

$$\binom{m}{i}(1-\theta w)^{m-i}w^i, \qquad i = 0, \ldots, m. \tag{14}$$

The transformation of the Sylvester matrix and its subresultant matrices between the Bernstein and modified Bernstein bases is considered in Section 3.4.

It follows from the discussion above that the degree and coefficients of an AGCD of two Bernstein polynomials are computed by transforming them to the modified Bernstein basis and performing all computations in this basis. The result of the three preprocessing operations on $f(y)$ and $g(y)$, and the assumption that the degree t of an AGCD has been computed using the methods in [4], is, therefore, the polynomials $\bar{f}(w)$ and $\alpha_0 \bar{g}(w)$,

$$\bar{f} = \bar{f}(w) = \sum_{i=0}^{m} \left(\bar{a}_i \theta_0^i\right) \binom{m}{i} (1-\theta_0 w)^{m-i} w^i, \qquad \bar{a}_i = \frac{a_i}{\lambda_t}, \tag{15}$$

and

$$\alpha_0 \bar{g} = \alpha_0 \bar{g}(w) = \alpha_0 \sum_{i=0}^{n} \left(\bar{b}_i \theta_0^i\right) \binom{n}{i} (1-\theta_0 w)^{n-i} w^i, \qquad \bar{b}_i = \frac{b_i}{\mu_t}, \tag{16}$$

where α_0 and θ_0 are, respectively, the optimal values of α_k and θ_k for $k = t$. The computation of the coefficients of an AGCD of $\bar{f}(w)$ and $\alpha_0 \bar{g}(w)$ is therefore determined from the tth modified Sylvester subresultant matrix,

$$S_t(\bar{f}, \alpha_0 \bar{g}) Q_t = D_t^{-1} T_t(\bar{f}, \alpha_0 \bar{g}) Q_t = \left[\begin{array}{cc} F_t(\bar{f}) & \alpha_0 G_t(\bar{g}) \end{array} \right], \tag{17}$$

where $F_t(\bar{f}) \in \mathbb{R}^{(m+n-t+1)\times(n-t+1)}$ and $\alpha_0 G_t(\bar{g}) \in \mathbb{R}^{(m+n-t+1)\times(m-t+1)}$ contain the coefficients of $\bar{f}(w)$ and $\alpha_0 \bar{g}(w)$ respectively. The coprime polynomials

$\bar{u}(w)$ and $\bar{v}(w)$, which are of degrees $m-t$ and $n-t$ respectively, associated with $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ are

$$\bar{u}(w) = \sum_{i=0}^{m-t} \left(\bar{u}_i\theta_0^i\right) \binom{m-t}{i} \left(1-\theta_0 w\right)^{m-t-i} w^i, \tag{18}$$

and

$$\bar{v}(w) = \sum_{i=0}^{n-t} \left(\bar{v}_i\theta_0^i\right) \binom{n-t}{i} \left(1-\theta_0 w\right)^{n-t-i} w^i. \tag{19}$$

Two methods for the computation of the coefficients of an AGCD of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ are considered and they require a solution of an approximate linear algebraic equation $A\mathbf{x} \approx \mathbf{b}$ where $A \in \mathbb{R}^{p\times q}, p < q$. The matrix A and vector $\mathbf{b}$ are structured because they are derived from the modified Sylvester subresultant matrix $S_t(\bar{f}, \alpha_0\bar{g})Q_t$, which is defined in (17). This approximate equation is transformed to an exact equation by the addition of a matrix E, which has the same structure as A, to the left hand side, and a vector $\mathbf{e}$, which has the same structure as $\mathbf{b}$, to the right hand side, such that the approximation $A\mathbf{x} \approx \mathbf{b}$ is transformed to the exact equation,

$$(A+E)\bar{\mathbf{x}} = \mathbf{b} + \mathbf{e}, \tag{20}$$

where E and $\mathbf{e}$ contain the coefficients of the polynomials that are added to the noisy polynomials $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$. It follows that the perturbed forms of these noisy polynomials have a GCD of degree t, that is, the given noisy polynomials that have an AGCD of degree t are perturbed to polynomials that have a GCD of degree t. The matrix E and vector $\mathbf{e}$ are not unique because the perturbations added to $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ to induce a non-constant GCD are not unique. A constraint is therefore added to (20) to impose uniqueness, and thus this equation is transformed to a least squares equality (LSE) problem,

$$\min \left\{ \|E\|^2 + \|\mathbf{e}\|^2 \right\} \quad \text{subject to} \quad (A+E)\bar{\mathbf{x}} = \mathbf{b} + \mathbf{e}, \quad \|\cdot\| \equiv \|\cdot\|_2 . \tag{21}$$

The minimisation constraint requires that, of all the matrices E and vectors $\mathbf{e}$ that satisfy (20), the given noisy polynomials are perturbed the minimum amount such that their perturbed forms have a GCD of degree t. The LSE problem (21) is non-linear because, apart from the entries of E and $\mathbf{e}$, improved values of α_0 and θ_0 are computed, and (21) is therefore solved iteratively.

Sections 3.1 and 3.2 consider, respectively, the computation of a structured low rank approximation of the Sylvester matrix $S(\bar{f}, \alpha_0\bar{g})$ and an approximate factorisation of $\bar{f}(w)$ and $\alpha_0 \bar{g}(w)$. It is shown that the computation of the coefficients of an AGCD of degree t by these methods yields an equation of the form (21), and the convergence of the iterative procedure for its solution is considered in Section 3.3.

The third preprocessing operation, which is defined in (13), shows that the AGCD computations are performed in the modified Bernstein basis. The computed AGCD can be transformed to the Bernstein basis using the inverse transformation $w = y/\theta^*$, where θ^* is the value of θ_0 at the termination of the iterative procedure for the solution of (21).

3.1 A structured low rank approximation of the Sylvester matrix

This section considers the application of the method of SNTLN to the computation of the coefficients of an AGCD of $\bar{f}(w)$ and $\alpha_0 \bar{g}(w)$ from a structured low rank approximation of $S(\bar{f}, \alpha_0 \bar{g})$.

It follows from Theorem 2.1 that, with respect to the exact polynomials $\hat{f}(y)$ and $\hat{g}(y)$, the rank of $S_{\hat{t}}(\hat{f}, \hat{g})Q_{\hat{t}}$ is $m+n-2\hat{t}+1$, that is, there is exactly one equation that defines the linear dependence of the columns of $S_{\hat{t}}(\hat{f}, \hat{g})Q_{\hat{t}}$. This exact rank loss does not exist when inexact polynomials $\bar{f}(w)$ and $\alpha_0 \bar{g}(w)$ are considered, and it is therefore necessary to consider the approximate rank loss of $S_t(\bar{f}, \alpha_0 \bar{g})Q_t$. If $\mathbf{c}_{t,k}$ is the kth column of $S_t(\bar{f}, \alpha_0 \bar{g})Q_t$, then

$$S_t(\bar{f}, \alpha_0 \bar{g})Q_t = \left[\mathbf{c}_{t,1} \; \mathbf{c}_{t,2} \; \cdots \; \mathbf{c}_{t,m+n-2t+1} \; \mathbf{c}_{t,m+n-2t+2} \right],$$

is near unit rank loss, and (9) is replaced by, for $k = t$,

$$\frac{S_t(\bar{f}, \alpha_0 \bar{g})\mathbf{p}(\bar{u}, \bar{v})}{\left\| S_t(\bar{f}, \alpha_0 \bar{g})\mathbf{p}(\bar{u}, \bar{v}) \right\|} = \frac{\left(D_t^{-1} T_t(\bar{f}, \alpha_0 \bar{g}) Q_t \right) \mathbf{r}(\bar{u}, \bar{v})}{\left\| \left(D_t^{-1} T_t(\bar{f}, \alpha_0 \bar{g}) Q_t \right) \mathbf{r}(\bar{u}, \bar{v}) \right\|} \approx \mathbf{0}, \tag{22}$$

where $\bar{u} = \bar{u}(w)$ and $\bar{v} = \bar{v}(w)$ are defined in (18) and (19) respectively. Since $S_t(\bar{f}, \alpha_0 \bar{g})Q_t$ is near unit rank loss, one of its columns is almost linearly dependent on its other columns, and it is necessary to determine the column that lies, with minimum error, in the space spanned by the other columns. The removal of the jth column of $S_t(\bar{f}, \alpha_0 \bar{g})Q_t = D_t^{-1} T_t(\bar{f}, \alpha_0 \bar{g})Q_t$ leaves a matrix $A_{t,j} = A_{t,j}(\bar{f}, \alpha_0 \bar{g})$ of order $(m+n-t+1) \times (m+n-2t+1)$,

$$A_{t,j} = \left[\mathbf{c}_{t,1} \; \mathbf{c}_{t,2} \; \cdots \; \mathbf{c}_{t,j-1} \; \mathbf{c}_{t,j+1} \; \cdots \; \mathbf{c}_{t,m+n-2t+1} \; \mathbf{c}_{t,m+n-2t+2} \right],$$

and the residual $r_{t,j}$ of the least squares solution of the approximate equation,

$$A_{t,j}\mathbf{x}_j \approx \mathbf{c}_{t,j}, \tag{23}$$

is computed for $j = 1, \ldots, m+n-2t+2$, that is, for each column of $S_t(\bar{f}, \alpha_0\bar{g})Q_t$. The optimal column to move to the right hand side is the column $c_{t,j}$ for which the residual $r_{t,j}$ is a minimum because this is the column that lies, with minimum error, in the space spanned by the columns of $A_{t,j}$. The index q of the optimal column is therefore given by

$$q = \arg\min_j \left\{ r_{t,j} \right\}, \tag{24}$$

and thus (23) becomes

$$A_{t,q}\mathbf{x}_q \approx \mathbf{c}_{t,q}. \tag{25}$$

The modified Sylvester subresultant matrix of order t of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ is $S_t(\bar{f}, \alpha_0\bar{g})Q_t = D_t^{-1}T_t(\bar{f}, \alpha_0\bar{g})Q_t$, where $T_t(\bar{f}, \alpha_0\bar{g}) \in \mathbb{R}^{(m+n-t+1)\times(m+n-2t+2)}$ is given by

$$T_t(\bar{f}, \alpha_0\bar{g}) = \begin{bmatrix} \bar{a}_0\binom{m}{0} & & & \alpha_0\bar{b}_0\binom{n}{0} & & \\ \bar{a}_1\binom{m}{1}\theta_0 & \ddots & & \alpha_0\bar{b}_1\binom{n}{1}\theta_0 & \ddots & \\ \vdots & \ddots & \bar{a}_0\binom{m}{0} & \vdots & \ddots & \alpha_0\bar{b}_0\binom{n}{0} \\ \bar{a}_m\binom{m}{m}\theta_0^m & \ddots & \bar{a}_1\binom{m}{1}\theta_0 & \alpha_0\bar{b}_n\binom{n}{n}\theta_0^n & \ddots & \alpha_0\bar{b}_1\binom{n}{1}\theta_0 \\ & \ddots & \vdots & & \ddots & \vdots \\ & & \bar{a}_m\binom{m}{m}\theta_0^m & & & \alpha_0\bar{b}_n\binom{n}{n}\theta_0^n \end{bmatrix},$$

and if $M_q \in \mathbb{R}^{(m+n-2t+2)\times(m+n-2t+1)}$ is defined as

$$M_q = \begin{bmatrix} \mathbf{e}_1 \; \mathbf{e}_2 \; \cdots \; \mathbf{e}_{q-1} \; \mathbf{e}_{q+1} \; \cdots \; \mathbf{e}_{m+n-2t+1} \; \mathbf{e}_{m+n-2t+2} \end{bmatrix},$$

where $\mathbf{e}_i \in \mathbb{R}^{m+n-2t+2}$ is the ith unit basis vector, then (25) becomes

$$\left(D_t^{-1}T_t\left(\bar{f}, \alpha_0\bar{g}\right)Q_t\right)M_q\mathbf{x}_q \approx \left(D_t^{-1}T_t\left(\bar{f}, \alpha_0\bar{g}\right)Q_t\right)\mathbf{e}_q, \tag{26}$$

where, from (22), $\mathbf{x}_q \in \mathbb{R}^{m+n-2t+1}$ is formed by the removal of the qth entry of $\mathbf{r}(\bar{u}, \bar{v})$.

It follows from the constraint in the LSE problem (21) that the inexact and coprime polynomials $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ are perturbed in order to induce a non-constant common divisor in their perturbed forms. If the polynomials added to $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ are, respectively,

$$\sum_{i=0}^{m}\left(z_i\phi^i\right)\binom{m}{i}(1-\phi w)^{m-i}w^i,$$

and

$$\beta\sum_{i=0}^{n}\left(z_{m+1+i}\phi^i\right)\binom{n}{i}(1-\phi w)^{n-i}w^i,$$

then the tth modified subresultant matrix of the perturbations is equal to $D_t^{-1}F_t(\beta,\phi,\mathbf{z})Q_t \in \mathbb{R}^{(m+n-t+1)\times(m+n-2t+2)}$ where

$$\mathbf{z}=\begin{bmatrix} z_0 & \cdots & z_m & z_{m+1} & \cdots & z_{m+n+1}\end{bmatrix}^T \in \mathbb{R}^{m+n+2}, \tag{27}$$

and $F_t(\beta,\phi,\mathbf{z})$ is given by

$$\begin{bmatrix}
z_0\binom{m}{0} & & & \beta z_{m+1}\binom{n}{0} & & \\
z_1\binom{m}{1}\phi & \ddots & & \beta z_{m+2}\binom{n}{1}\phi & \ddots & \\
\vdots & \ddots & z_0\binom{m}{0} & \vdots & \ddots & \beta z_{m+1}\binom{n}{0} \\
z_m\binom{m}{m}\phi^m & \ddots & z_1\binom{m}{1}\phi & \beta z_{m+n+1}\binom{n}{n}\phi^n & \ddots & \beta z_{m+2}\binom{n}{1}\phi \\
 & \ddots & \vdots & & \ddots & \vdots \\
 & & z_m\binom{m}{m}\phi^m & & & \beta z_{m+n+1}\binom{n}{n}\phi^n
\end{bmatrix}.$$

This perturbation of the coefficients of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ implies that the approximation (26) is replaced by the exact equation,

$$\left(D_t^{-1}\left(T_t+F_t\right)Q_t\right)M_q\mathbf{x}_q=\left(D_t^{-1}\left(T_t+F_t\right)Q_t\right)\mathbf{e}_q, \tag{28}$$

which corresponds to the constraint in (21).

A change of notation is required because (28) is a non-linear equation that is solved iteratively, and the variables to be determined include β,ϕ and $\mathbf{z}$. The initial values of these variables in the iterative procedure are $\beta^{(0)}=\alpha_0, \phi^{(0)}=$

θ_0 and $\mathbf{z}^{(0)} = \mathbf{0}$, and it is therefore appropriate to include these parameters in the arguments of the vectors and matrices in (28),

$$\left(D_t^{-1}\left(T_t(\beta,\phi) + F_t(\beta,\phi,\mathbf{z})\right)Q_t\right)M_q\mathbf{x}_q = \mathbf{c}_t(\beta,\phi) + \mathbf{h}_t(\beta,\phi,\mathbf{z}), \tag{29}$$

where

$$\mathbf{c}_t(\beta,\phi) = D_t^{-1}T_t(\beta,\phi)Q_t\mathbf{e}_q \quad \text{and} \quad \mathbf{h}_t(\beta,\phi,\mathbf{z}) = D_t^{-1}F_t(\beta,\phi,\mathbf{z})Q_t\mathbf{e}_q.$$

The scalar multiplier β is included in the arguments of $\mathbf{c}_t$ and $\mathbf{h}_t$ because this is the most general condition. These vectors may not, however, be functions of β, and this is dependent on the index q of the optimal column, which is defined in (24),

$$\begin{array}{llll} \mathbf{c}_t = \mathbf{c}_t(\phi), & \mathbf{h}_t = \mathbf{h}_t(\phi,\mathbf{z}) & \text{if} & 1 \le q \le n-t+1, \\ \mathbf{c}_t = \mathbf{c}_t(\beta,\phi), & \mathbf{h}_t = \mathbf{h}_t(\beta,\phi,\mathbf{z}) & \text{if} & n-t+2 \le q \le m+n-2t+2. \end{array} \tag{30}$$

The following theory is developed assuming $n-t+2 \le q \le m+n-2t+2$, but the dependence of $\mathbf{c}_t$ and $\mathbf{h}_t$ on β is removed if $1 \le q \le n-t+1$.

Equation (29) is non-linear and it is solved by the Newton-Raphson method. The residual of an approximate solution of this equation is

$$\begin{aligned} \mathbf{r}(\beta,\phi,\mathbf{x}_q,\mathbf{z}) = {} & \mathbf{c}_t(\beta,\phi) + \mathbf{h}_t(\beta,\phi,\mathbf{z}) \\ & -\left(D_t^{-1}\left(T_t(\beta,\phi) + F_t(\beta,\phi,\mathbf{z})\right)Q_t\right)M_q\mathbf{x}_q, \end{aligned} \tag{31}$$

and if $\tilde{\mathbf{r}}$ is defined as

$$\begin{aligned} \tilde{\mathbf{r}} := {} & \mathbf{r}(\beta+\delta\beta,\phi+\delta\phi,\mathbf{x}_q+\delta\mathbf{x}_q,\mathbf{z}+\delta\mathbf{z}) \\ = {} & \mathbf{c}_t(\beta+\delta\beta,\phi+\delta\phi) + \mathbf{h}_t(\beta+\delta\beta,\phi+\delta\phi,\mathbf{z}+\delta\mathbf{z}) \\ & -\left(D_t^{-1}\Big(T_t(\beta+\delta\beta,\phi+\delta\phi) + F_t(\beta+\delta\beta,\phi+\delta\phi,\mathbf{z}+\delta\mathbf{z})\Big)Q_t\right) \\ & \times M_q(\mathbf{x}_q+\delta\mathbf{x}_q), \end{aligned}$$

then to first order,

$$
\begin{aligned}
\tilde{\mathbf{r}} = {} & \mathbf{r}(\beta,\phi,\mathbf{x}_q,\mathbf{z}) - \left(D_t^{-1}\left(\frac{\partial T_t}{\partial \phi} + \frac{\partial F_t}{\partial \phi}\right) Q_t M_q \mathbf{x}_q - \left(\frac{\partial \mathbf{c}_t}{\partial \phi} + \frac{\partial \mathbf{h}_t}{\partial \phi}\right)\right)\delta\phi \\
& - \left(D_t^{-1}\left(\frac{\partial T_t}{\partial \beta} + \frac{\partial F_t}{\partial \beta}\right) Q_t M_q \mathbf{x}_q - \left(\frac{\partial \mathbf{c}_t}{\partial \beta} + \frac{\partial \mathbf{h}_t}{\partial \beta}\right)\right)\delta\beta \\
& - \left(D_t^{-1}(T_t + F_t) Q_t M_q \right)\delta\mathbf{x}_q + \sum_{i=0}^{m+n+1} \frac{\partial \mathbf{h}_t}{\partial z_i}\delta z_i \\
& - \left(D_t^{-1} \sum_{i=0}^{m+n+1} \frac{\partial F_t}{\partial z_i}\delta z_i \right) Q_t M_q \mathbf{x}_q,
\end{aligned}
\tag{32}
$$

and it is noted again that the forms of $\mathbf{c}_t$ and $\mathbf{h}_t$, and their derivatives, depend on the value of q, as shown in (30).

Example 3.1 If $q = n - t + 4 > n - t + 1$, then $\mathbf{c}_t = \mathbf{c}_t(\beta,\phi)$ and $\mathbf{h}_t = \mathbf{h}_t(\beta,\phi,\mathbf{z})$, and thus

$$
\begin{aligned}
Q_t\mathbf{e}_q &= \left[\mathbf{0}_{n-t+1}^T \ 0 \ 0 \ \tbinom{m-t}{2} \ \mathbf{0}_{m-t-2}^T \right]^T, \\
\mathbf{c}_t &= \binom{m-t}{2} D_t^{-1} \left[0 \ 0 \ \beta\bar{b}_0\tbinom{n}{0} \ \beta\bar{b}_1\tbinom{n}{1}\phi \ \cdots \ \beta\bar{b}_n\tbinom{n}{n}\phi^n \ \mathbf{0}_{m-t-2}^T \right]^T, \\
\frac{\partial \mathbf{c}_t}{\partial \phi} &= \binom{m-t}{2} D_t^{-1} \left[0 \ 0 \ 0 \ \beta\bar{b}_1\tbinom{n}{1} \ \cdots \ \beta n\bar{b}_n\tbinom{n}{n}\phi^{n-1} \ \mathbf{0}_{m-t-2}^T \right]^T, \\
\frac{\partial \mathbf{c}_t}{\partial \beta} &= \binom{m-t}{2} D_t^{-1} \left[0 \ 0 \ \bar{b}_0\tbinom{n}{0} \ \bar{b}_1\tbinom{n}{1}\phi \ \cdots \ \bar{b}_n\tbinom{n}{n}\phi^n \ \mathbf{0}_{m-t-2}^T \right]^T, \\
\mathbf{h}_t &= \binom{m-t}{2} D_t^{-1} \left[0 \ 0 \ \beta z_{m+1}\tbinom{n}{0} \ \beta z_{m+2}\tbinom{n}{1}\phi \ \cdots \ \beta z_{m+n+1}\tbinom{n}{n}\phi^n \ \mathbf{0}_{m-t-2}^T \right]^T, \\
\frac{\partial \mathbf{h}_t}{\partial \phi} &= \binom{m-t}{2} D_t^{-1} \left[0 \ 0 \ 0 \ \beta z_{m+2}\tbinom{n}{1} \ \cdots \ \beta n z_{m+n+1}\tbinom{n}{n}\phi^{n-1} \ \mathbf{0}_{m-t-2}^T \right]^T, \\
\frac{\partial \mathbf{h}_t}{\partial \beta} &= \binom{m-t}{2} D_t^{-1} \left[0 \ 0 \ z_{m+1}\tbinom{n}{0} \ z_{m+2}\tbinom{n}{1}\phi \ \cdots \ z_{m+n+1}\tbinom{n}{n}\phi^n \ \mathbf{0}_{m-t-2}^T \right]^T,
\end{aligned}
$$

where the zero vector $\mathbf{0}_r^T$ is of length r. The partial derivatives $\frac{\partial T_t}{\partial \phi}, \frac{\partial T_t}{\partial \beta}, \frac{\partial F_t}{\partial \phi}$ and $\frac{\partial F_t}{\partial \beta}$ are calculated in a similar manner. □

It is still assumed that $q > n - t + 1$, and thus the general expression for $\mathbf{h}_t$ is

$$
\begin{aligned}
\mathbf{h}_t &= \binom{m-t}{q-(n-t+2)} D_t^{-1} \begin{bmatrix} \mathbf{0}_{q-n+t-2} \\ \beta z_{m+1}\binom{n}{0} \\ \beta z_{m+2}\binom{n}{1}\phi \\ \vdots \\ \beta z_{m+n}\binom{n}{n-1}\phi^{n-1} \\ \beta z_{m+n+1}\binom{n}{n}\phi^{n} \\ \mathbf{0}_{m+n-2t-q+2} \end{bmatrix} \\
&= \beta \binom{m-t}{q-(n-t+2)} D_t^{-1} \begin{bmatrix} 0_{q-n+t-2,m+1} & 0_{q-n+t-2,n+1} \\ 0_{n+1,m+1} & G \\ 0_{m+n-2t-q+2,m+1} & 0_{m+n-2t-q+2,n+1} \end{bmatrix} \left[\begin{array}{c} z_0 \\ \vdots \\ z_m \\ \hline z_{m+1} \\ \vdots \\ z_{m+n+1} \end{array} \right] \\
&= \beta D_t^{-1} P_t \mathbf{z},
\end{aligned}
$$

where

$$
G = G(\phi) = \operatorname{diag} \left[\binom{n}{0} \; \binom{n}{1}\phi \; \cdots \; \binom{n}{n-1}\phi^{n-1} \; \binom{n}{n}\phi^{n} \right] \in \mathbb{R}^{(n+1)\times(n+1)},
$$

$\mathbf{z}$ is defined in (27) and $P_t = P_t(\phi) \in \mathbb{R}^{(m+n-t+1)\times(m+n+2)}$ is given by

$$
P_t = \binom{m-t}{q-(n-t+2)} \begin{bmatrix} 0_{q-n+t-2,m+1} & 0_{q-n+t-2,n+1} \\ 0_{n+1,m+1} & G \\ 0_{m+n-2t-q+2,m+1} & 0_{m+n-2t-q+2,n+1} \end{bmatrix}.
$$

It therefore follows that the penultimate term in (32) can be expressed as

$$
\sum_{i=0}^{m+n+1} \frac{\partial \mathbf{h}_t}{\partial z_i} \delta z_i = \beta D_t^{-1} P_t \delta \mathbf{z}.
$$

The last term in (32) must also be simplified, and this simplification is achieved by noting that the matrix-vector multiplication $\left(D_t^{-1} F_t\right)(Q_t M_q \mathbf{x}_q)$ represents the product of two polynomials expressed in the modified Bernstein basis. Since polynomial multiplication is commutative, it follows that there exists a matrix $Y_t = Y_t(\beta, \phi, \mathbf{x}_q) \in \mathbb{R}^{(m+n-t+1)\times(m+n+2)}$ such that

$$\left(D_t^{-1}Y_t\right)\mathbf{z} = \left(D_t^{-1}F_t\right)(Q_t M_q \mathbf{x}_q)\,,$$

for all $\beta, \phi, \mathbf{x}_q$ and $\mathbf{z}$. The differentiation of this equation with respect to $\mathbf{z}$, and keeping β and ϕ constant, yields

$$Y_t \delta\mathbf{z} = \sum_{i=0}^{m+n+1} \left(\frac{\partial F_t}{\partial z_i}\delta z_i\right) Q_t M_q \mathbf{x}_q,$$

and thus (32) simplifies to

$$\begin{aligned}\tilde{\mathbf{r}} &= \mathbf{r}(\beta+\delta\beta, \phi+\delta\phi, \mathbf{x}_q+\delta\mathbf{x}_q, \mathbf{z}+\delta\mathbf{z})\\ &= \mathbf{r}(\beta,\phi,\mathbf{x}_q,\mathbf{z}) - \left(D_t^{-1}\left(\frac{\partial T_t}{\partial\phi}+\frac{\partial F_t}{\partial\phi}\right)Q_t M_q\mathbf{x}_q - \left(\frac{\partial \mathbf{c}_t}{\partial\phi}+\frac{\partial \mathbf{h}_t}{\partial\phi}\right)\right)\delta\phi\\ &\quad - \left(D_t^{-1}\left(\frac{\partial T_t}{\partial\beta}+\frac{\partial F_t}{\partial\beta}\right)Q_t M_q\mathbf{x}_q - \left(\frac{\partial \mathbf{c}_t}{\partial\beta}+\frac{\partial \mathbf{h}_t}{\partial\beta}\right)\right)\delta\beta\\ &\quad -\left(D_t^{-1}(T_t+F_t)Q_t M_q\right)\delta\mathbf{x}_q - D_t^{-1}(Y_t-\beta P_t)\delta\mathbf{z}, \end{aligned} \tag{33}$$

to first order. The jth iteration in the Newton-Raphson method for the calculation of $\beta, \phi, \mathbf{x}_q$ and $\mathbf{z}$ is obtained from (33),

$$\left[\, H_{\mathbf{z}}\ H_{\mathbf{x}_q}\ \mathbf{H}_\beta\ \mathbf{H}_\phi \,\right]^{(j)} \begin{bmatrix} \delta\mathbf{z} \\ \delta\mathbf{x}_q \\ \delta\beta \\ \delta\phi \end{bmatrix}^{(j)} = \mathbf{r}^{(j)}, \tag{34}$$

where $\mathbf{r}^{(j)} = \mathbf{r}^{(j)}(\beta,\phi,\mathbf{x}_q,\mathbf{z})$,

$$\begin{aligned} H_{\mathbf{z}} &= D_t^{-1}(Y_t - \beta P_t) \in \mathbb{R}^{(m+n-t+1)\times(m+n+2)},\\ H_{\mathbf{x}_q} &= D_t^{-1}(T_t+F_t)Q_t M_q \in \mathbb{R}^{(m+n-t+1)\times(m+n-2t+1)},\\ \mathbf{H}_\beta &= D_t^{-1}\left(\frac{\partial T_t}{\partial\beta}+\frac{\partial F_t}{\partial\beta}\right)Q_t M_q\mathbf{x}_q - \left(\frac{\partial \mathbf{c}_t}{\partial\beta}+\frac{\partial \mathbf{h}_t}{\partial\beta}\right) \in \mathbb{R}^{m+n-t+1},\\ \mathbf{H}_\phi &= D_t^{-1}\left(\frac{\partial T_t}{\partial\phi}+\frac{\partial F_t}{\partial\phi}\right)Q_t M_q\mathbf{x}_q - \left(\frac{\partial \mathbf{c}_t}{\partial\phi}+\frac{\partial \mathbf{h}_t}{\partial\phi}\right) \in \mathbb{R}^{m+n-t+1}. \end{aligned}$$

The values of $\beta, \phi, \mathbf{x}_q$ and $\mathbf{z}$ at the jth iteration, $j = 1, 2, \ldots,$ are

$$\mathbf{y}^{(j)} = \mathbf{y}^{(j-1)} + \delta\mathbf{y}^{(j)}, \qquad \mathbf{y}^{(j)} = \begin{bmatrix} \mathbf{z} \\ \mathbf{x}_q \\ \beta \\ \phi \end{bmatrix}^{(j)}, \qquad \delta\mathbf{y}^{(j)} = \begin{bmatrix} \delta\mathbf{z} \\ \delta\mathbf{x}_q \\ \delta\beta \\ \delta\phi \end{bmatrix}^{(j)}, \tag{35}$$

where $\mathbf{y}^{(j)} \in \mathbb{R}^{2m+2n-2t+5}$, the initial value of $\mathbf{z}$ is $\mathbf{z}^{(0)} = \mathbf{0}$ because the given data is the inexact data, the initial values of β and ϕ are $\beta^{(0)} = \alpha_0$ and $\phi^{(0)} = \theta_0$, and $\mathbf{x}_q^{(0)}$, the initial value of $\mathbf{x}_q$, is calculated from (31) when $\mathbf{r} = \mathbf{h}_t = \mathbf{0}$ and $F_t = 0$,

$$\mathbf{x}_q^{(0)} = \arg\min_{\mathbf{w}} \left\| \left(D_t^{-1} T_t \left(\alpha_0, \theta_0 \right) Q_t M_q \right) \mathbf{w} - \mathbf{c}_t \left(\alpha_0, \theta_0 \right) \right\|.$$

Equation (34) is under-determined and it can be written as

$$C^{(j)} \delta\mathbf{y}^{(j)} = \mathbf{q}^{(j)}, \tag{36}$$

where $C^{(j)} \in \mathbb{R}^{(m+n-t+1)\times(2m+2n-2t+5)}$ and $\mathbf{q}^{(j)} \in \mathbb{R}^{m+n-t+1}$ are, respectively,

$$C^{(j)} = \begin{bmatrix} H_{\mathbf{z}} \; H_{\mathbf{x}_q} \; \mathbf{H}_\beta \; \mathbf{H}_\phi \end{bmatrix}^{(j)} \qquad \text{and} \qquad \mathbf{q}^{(j)} = \mathbf{r}^{(j)}.$$

A unique solution of (36) is obtained by calculating the vector $\delta\mathbf{y}^{(j)}$ of minimum magnitude that satisfies this equation, that is, the solution that is closest to the given inexact data is required. It follows from (35) that the magnitude of the difference between the solution $\mathbf{y}^{(j)}$ at the jth iteration and the initial estimate $\mathbf{y}^{(0)}$ of the solution is

$$\left\| \mathbf{y}^{(j)} - \mathbf{y}^{(0)} \right\| = \left\| \mathbf{y}^{(j-1)} + \delta\mathbf{y}^{(j)} - \mathbf{y}^{(0)} \right\| = \left\| \delta\mathbf{y}^{(j)} - \mathbf{p}^{(j)} \right\|, \tag{37}$$

where

$$\mathbf{p}^{(j)} = \mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}, \tag{38}$$

and thus the minimisation of (37) subject to (36) yields the LSE problem (21),

$$\min_{\delta\mathbf{y}^{(j)}} \left\| \delta\mathbf{y}^{(j)} - \mathbf{p}^{(j)} \right\| \quad \text{subject to} \quad C\delta\mathbf{y}^{(j)} = \mathbf{q}^{(j)}, \quad j = 1, 2, \ldots \tag{39}$$

This problem can be solved by the QR decomposition at each iteration [7], where $\delta\mathbf{y}^{(j)}$, $C^{(j)}, \mathbf{q}^{(j)}$ and $\mathbf{p}^{(j)}$ are updated between successive iterations. The convergence of the iteration (39) is considered in Section 3.3.

If the iteration (39) converges, then the vector $\mathbf{y}^{(j)}$ at termination contains the perturbations $z_i^*, i=0,\ldots,m$, and $z_i^*, i=m+1,\ldots,m+n+1$, and the parameters β^* and ϕ^*, such that the corrected polynomials

$$\begin{aligned}\tilde{f}(w) &= \sum_{i=0}^{m}\left((\bar{a}_i+z_i^*)\phi^{*^i}\right)\binom{m}{i}(1-\phi^*w)^{m-i}w^i\\ &:= \sum_{i=0}^{m}\left(\tilde{a}_i\phi^{*^i}\right)\binom{m}{i}(1-\phi^*w)^{m-i}w^i,\end{aligned} \tag{40}$$

and

$$\begin{aligned}\beta^*\tilde{g}(w) &= \beta^*\sum_{i=0}^{n}\left((\bar{b}_i+z_{m+1+i}^*)\phi^{*^i}\right)\binom{n}{i}(1-\phi^*w)^{n-i}w^i\\ &:= \beta^*\sum_{i=0}^{n}\left(\tilde{b}_i\phi^{*^i}\right)\binom{n}{i}(1-\phi^*w)^{n-i}w^i,\end{aligned} \tag{41}$$

have a GCD $\tilde{d}(w)$ of degree t,

$$\tilde{d}(w) = \sum_{i=0}^{t}\left(\tilde{d}_i\phi^{*^i}\right)\binom{t}{i}(1-\phi^*w)^{t-i}w^i, \tag{42}$$

that satisfies

$$\tilde{u}(w)\tilde{d}(w) = \tilde{f}(w) \qquad \text{and} \qquad \tilde{v}(w)\tilde{d}(w) = \beta^*\tilde{g}(w), \tag{43}$$

where the corrected coprime polynomials $\tilde{u}(w)$ and $\tilde{v}(w)$ are obtained from the least squares solution $\mathbf{x}_q$ of (29) when $\beta=\beta^*, \phi=\phi^*$ and $\mathbf{z}=\mathbf{z}^*$. The polynomial $\tilde{d}(w)$ is most easily computed by combining the polynomial equations (43) into one matrix equation,

$$H^{-1}\begin{bmatrix} D_1(\tilde{u}) \\ D_2(\tilde{v})\end{bmatrix} J\tilde{\mathbf{d}} = \begin{bmatrix}\tilde{\mathbf{f}} \\ \beta^*\tilde{\mathbf{g}}\end{bmatrix}, \tag{44}$$

where $D_1(\tilde{u}) \in \mathbb{R}^{(m+1)\times(t+1)}$ and $D_2(\tilde{v}) \in \mathbb{R}^{(n+1)\times(t+1)}$ are Tœplitz matrices that contain the coefficients of $\tilde{u}(w)$ and $\tilde{v}(w)$ respectively,

$$H^{-1} = H(m,n)^{-1} = \begin{bmatrix} H_1(m)^{-1} & 0 \\ 0 & H_1(n)^{-1} \end{bmatrix} \in \mathbb{R}^{(m+n+2)\times(m+n+2)}, \tag{45}$$

$$\begin{aligned} H_1(p)^{-1} &= \text{diag}\left[\begin{array}{cccc} \frac{1}{\binom{p}{0}} & \frac{1}{\binom{p}{1}} & \cdots & \frac{1}{\binom{p}{p}} \end{array}\right] \in \mathbb{R}^{(p+1)\times(p+1)}, \\ J &= \text{diag}\left[\begin{array}{cccc} \binom{t}{0} & \binom{t}{1} & \cdots & \binom{t}{t} \end{array}\right] \in \mathbb{R}^{(t+1)\times(t+1)}, \\ \tilde{\mathbf{f}} &= \left[\begin{array}{cccc} \tilde{a}_0 & \tilde{a}_1\phi^* & \cdots & \tilde{a}_m{\phi^*}^m \end{array}\right]^T \in \mathbb{R}^{m+1}, \\ \tilde{\mathbf{g}} &= \left[\begin{array}{cccc} \tilde{b}_0 & \tilde{b}_1\phi^* & \cdots & \tilde{b}_n{\phi^*}^n \end{array}\right]^T \in \mathbb{R}^{n+1}, \\ \tilde{\mathbf{d}} &= \left[\begin{array}{cccc} \tilde{d}_0 & \tilde{d}_1\phi^* & \cdots & \tilde{d}_t{\phi^*}^t \end{array}\right]^T \in \mathbb{R}^{t+1}, \end{aligned} \tag{46}$$

and the coefficients $\tilde{a}_i{\phi^*}^i$, $\tilde{b}_i{\phi^*}^i$ and $\tilde{d}_i{\phi^*}^i$ are defined in (40), (41) and (42) respectively. The error of the least squares solution of (44) is very small because the polynomials $\tilde{f}(w), \beta^*\tilde{g}(w), \tilde{u}(w)$ and $\tilde{v}(w)$ satisfy (43) since they are computed from the solution of (39) at convergence.

3.2 Approximate polynomial factorisation

This section considers an approximate factorisation of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$, which are defined in (15) and (16) respectively, in order to compute the coefficients of an AGCD of degree t. This factorisation also yields a non-linear equation that requires the solution of an LSE problem at each iteration.

The preprocessing operations discussed in Section 3 yield the values α_0 and θ_0, and thus if $\dot{d}(w)$ is an AGCD of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$, then

$$\bar{u}(w)\dot{d}(w) \approx \bar{f}(w) \qquad \text{and} \qquad \tilde{v}(w)\dot{d}(w) \approx \alpha_0\bar{g}(w), \tag{47}$$

where $\bar{u}(w)$ and $\bar{v}(w)$ are defined in (18) and (19) respectively, and

$$\dot{d}(w) = \sum_{i=0}^{t} \left(\dot{d}_i\theta_0^i\right)\binom{t}{i}(1-\theta_0 w)^{t-i}w^i. \tag{48}$$

The approximations (47) can be combined into one approximate equation,

$$H^{-1}\begin{bmatrix} C_1(\theta_0) \\ C_2(\theta_0) \end{bmatrix} J\dot{\mathbf{d}}(\theta_0) \approx \begin{bmatrix} \bar{\mathbf{f}}(\theta_0) \\ \alpha_0\bar{\mathbf{g}}(\theta_0) \end{bmatrix}, \tag{49}$$

where H^{-1} and J are defined in (45) and (46) respectively,

$$\begin{aligned}
\bar{\mathbf{f}}(\theta_0) &= \begin{bmatrix} \bar{a}_0 & \bar{a}_1\theta_0 & \cdots & \bar{a}_m\theta_0^m \end{bmatrix}^T \in \mathbb{R}^{m+1},\\
\bar{\mathbf{g}}(\theta_0) &= \begin{bmatrix} \bar{b}_0 & \bar{b}_1\theta_0 & \cdots & \bar{b}_n\theta_0^n \end{bmatrix}^T \in \mathbb{R}^{n+1},\\
\dot{\mathbf{d}}(\theta_0) &= \begin{bmatrix} \dot{d}_0 & \dot{d}_1\theta_0 & \cdots & \dot{d}_t\theta_0^t \end{bmatrix}^T \in \mathbb{R}^{t+1},
\end{aligned}$$

$C_1(\theta_0) \in \mathbb{R}^{(m+1)\times(t+1)}$ and $C_2(\theta_0) \in \mathbb{R}^{(n+1)\times(t+1)}$ are Tœplitz matrices,

$$C_1(\theta_0) = \begin{bmatrix}
\bar{u}_0\binom{m-t}{0} & & & \\
\bar{u}_1\binom{m-t}{1}\theta_0 & \bar{u}_0\binom{m-t}{0} & & \\
\bar{u}_2\binom{m-t}{2}\theta_0^2 & \bar{u}_1\binom{m-t}{1}\theta_0 & \ddots & \\
\vdots & \bar{u}_2\binom{m-t}{2}\theta_0^2 & \ddots & \bar{u}_0\binom{m-t}{0} \\
\vdots & \vdots & \ddots & \bar{u}_1\binom{m-t}{1}\theta_0 \\
\bar{u}_{m-t}\binom{m-t}{m-t}\theta_0^{m-t} & \vdots & \ddots & \bar{u}_2\binom{m-t}{2}\theta_0^2 \\
 & \bar{u}_{m-t}\binom{m-t}{m-t}\theta_0^{m-t} & \ddots & \vdots \\
 & & \ddots & \vdots \\
 & & & \bar{u}_{m-t}\binom{m-t}{m-t}\theta_0^{m-t}
\end{bmatrix},$$

and

$$C_2(\theta_0) = \begin{bmatrix}
\bar{v}_0\binom{n-t}{0} & & & \\
\bar{v}_1\binom{n-t}{1}\theta_0 & \bar{v}_0\binom{n-t}{0} & & \\
\bar{v}_2\binom{n-t}{2}\theta_0^2 & \bar{v}_1\binom{n-t}{1}\theta_0 & \ddots & \\
\vdots & \bar{v}_2\binom{n-t}{2}\theta_0^2 & \ddots & \bar{v}_0\binom{n-t}{0} \\
\vdots & \vdots & \ddots & \bar{v}_1\binom{n-t}{1}\theta_0 \\
\bar{v}_{n-t}\binom{n-t}{n-t}\theta_0^{n-t} & \vdots & \ddots & \bar{v}_2\binom{n-t}{2}\theta_0^2 \\
 & \bar{v}_{n-t}\binom{n-t}{n-t}\theta_0^{n-t} & \ddots & \vdots \\
 & & \ddots & \vdots \\
 & & & \bar{v}_{n-t}\binom{n-t}{n-t}\theta_0^{n-t}
\end{bmatrix}.$$

Consideration of the combinatorial terms in $H_1(m)^{-1}C_1(\theta_0)J$, where $H_1(m)^{-1}$ is defined in (45), in the coefficient matrix in (49) shows that it is necessary to evaluate the expression

$$\frac{\binom{m-t}{i-j}\binom{t}{j}}{\binom{m}{i}}, \qquad j=0,\ldots,t, \qquad i=j,\ldots,m-t+j,$$

which requires that three combinatorial terms be computed. A computationally efficient form of this expression is obtained by rearrangement of the combinatorial terms,

$$\frac{\binom{m-t}{i-j}\binom{t}{j}}{\binom{m}{i}} = \frac{\binom{i}{j}\binom{m-i}{t-j}}{\binom{m}{t}}, \qquad j=0,\ldots,t, \qquad i=j,\ldots,m-t+j,$$

which requires the evaluation of two combinatorial terms for each value of i and j. It is clear that a similar simplification is appropriate for the matrix $H_1(n)^{-1}C_2(\theta_0)J$ in the coefficient matrix in (49), and that these simplifications are identical to the simplifications (11) and (12).

The approximate equation (49) is transformed to an exact equation by the addition of a structured matrix to the left hand side and a structured vector to the right hand side, that is, the procedure used to transform (26) to (29) is used. This procedure is equivalent to the addition of polynomials $s(w)$ and $t(w)$ to $\bar{f}(w)$ and $\bar{g}(w)$ respectively,

$$s(w) = \sum_{i=0}^{m} \left(p_i\theta_0^i\right)\binom{m}{i}(1-\theta_0 w)^{m-i}w^i, \tag{50}$$

$$t(w) = \sum_{i=0}^{n} \left(q_i\theta_0^i\right)\binom{n}{i}(1-\theta_0 w)^{n-i}w^i, \tag{51}$$

and the addition of polynomials $c(w)$ and $e(w)$ to $\bar{u}(w)$ and $\bar{v}(w)$ respectively,

$$c(w) = \sum_{i=0}^{m-t} \left(z_i\theta_0^i\right)\binom{m-t}{i}(1-\theta_0 w)^{m-t-i}w^i,$$

$$e(w) = \sum_{i=0}^{n-t} \left(z_{m-t+1+i}\theta_0^i\right)\binom{n-t}{i}(1-\theta_0 w)^{n-t-i}w^i,$$

such that the approximate equations (47) become

$$(\bar{u}(w)+c(w))\bar{d}(w) = \bar{f}(w)+s(w), \tag{52}$$

$$(\bar{v}(w)+e(w))\bar{d}(w) = (\alpha_0+\beta)\left(\bar{g}(w)+t(w)\right), \tag{53}$$

where the AGCD $\dot{d}(w)$, which is defined in (48), is replaced by the GCD $\bar{d}(w)$, and β is a constant to be determined. The polynomial equations (52) and (53) are written as a matrix equation that is solved iteratively for the coefficients of the polynomials $c(w), e(w), s(w), t(w)$ and $\bar{d}(w)$, and the parameters β and θ_0. The initial values of the coefficients and parameters in the iterative procedure are $z_i = 0, i = 0, \ldots, m+n-2t+1$, $p_i = 0, i = 0, \ldots, m$, $q_i = 0, i = 0, \ldots, n$, $\beta^{(0)} = 0$ and $\phi^{(0)} = \theta_0$. Equations (52) and (53) are therefore written as

$$H^{-1}\begin{bmatrix} C_1(\phi) + E_1(\mathbf{w}_1, \phi) \\ C_2(\phi) + E_2(\mathbf{w}_2, \phi) \end{bmatrix} J\bar{\mathbf{d}}(\phi) = \begin{bmatrix} \bar{\mathbf{f}}(\phi) + \mathbf{s}(\mathbf{p}, \phi) \\ (\alpha_0 + \beta)\Big(\bar{\mathbf{g}}(\phi) + \mathbf{t}(\mathbf{q}, \phi)\Big) \end{bmatrix}, \qquad (54)$$

where H^{-1} is defined in (45), $E_1(\mathbf{w}_1, \phi) \in \mathbb{R}^{(m+1)\times(t+1)}$ and $E_2(\mathbf{w}_2, \phi) \in \mathbb{R}^{(n+1)\times(t+1)}$ are Tœplitz matrices that contain the coefficients of $c(w)$ and $e(w)$ respectively, $\mathbf{w}_1$ and $\mathbf{w}_2$ are derived from $\mathbf{z}$, which is defined in(27),

$$\mathbf{z} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix},$$

$$\mathbf{w}_1 = \begin{bmatrix} z_0\binom{m-t}{0} & z_1\binom{m-t}{1} & \cdots & z_{m-t}\binom{m-t}{m-t} \end{bmatrix}^T \in \mathbb{R}^{m-t+1},$$

$$\mathbf{w}_2 = \begin{bmatrix} z_{m-t+1}\binom{n-t}{0} & z_{m-t+2}\binom{n-t}{1} & \cdots & z_{m+n-2t+1}\binom{n-t}{n-t} \end{bmatrix}^T \in \mathbb{R}^{n-t+1},$$

$$E_1(\mathbf{w}_1, \phi) = \begin{bmatrix} z_0\binom{m-t}{0} & & \\ z_1\binom{m-t}{1}\phi & \ddots & \\ z_2\binom{m-t}{2}\phi^2 & \ddots & z_0\binom{m-t}{0} \\ \vdots & \ddots & z_1\binom{m-t}{1}\phi \\ \vdots & \ddots & z_2\binom{m-t}{2}\phi^2 \\ z_{m-t}\binom{m-t}{m-t}\phi^{m-t} & \ddots & \vdots \\ & \ddots & \vdots \\ & & z_{m-t}\binom{m-t}{m-t}\phi^{m-t} \end{bmatrix},$$

$$E_2(\mathbf{w}_2,\phi) = \left[\begin{array}{ccc} z_{m-t+1}\binom{n-t}{0} & & \\ z_{m-t+2}\binom{n-t}{1}\phi & \ddots & \\ z_{m-t+3}\binom{n-t}{2}\phi^2 & \ddots & z_{m-t+1}\binom{n-t}{0} \\ \vdots & \ddots & z_{m-t+2}\binom{n-t}{1}\phi \\ \vdots & \ddots & z_{m-t+3}\binom{n-t}{2}\phi^2 \\ z_{m+n-2t+1}\binom{n-t}{n-t}\phi^{n-t} & \ddots & \vdots \\ & \ddots & \vdots \\ & & z_{m+n-2t+1}\binom{n-t}{n-t}\phi^{n-t} \end{array}\right],$$

the vectors $\mathbf{s} = \mathbf{s}(\mathbf{p},\phi)$ and $\mathbf{t} = \mathbf{t}(\mathbf{q},\phi)$ contain the coefficients of $s(w)$ and $t(w)$, which are defined in (50) and (51) respectively,

$$\begin{aligned} \mathbf{s} &= \left[\, p_0 \; p_1\phi \; \cdots \; p_m\phi^m \,\right]^T \in \mathbb{R}^{m+1}, \\ \mathbf{t} &= \left[\, q_0 \; q_1\phi \; \cdots \; q_n\phi^n \,\right]^T \in \mathbb{R}^{n+1}, \\ \mathbf{p} &= \left[\, p_0 \; p_1 \; \cdots \; p_m \,\right]^T \in \mathbb{R}^{m+1}, \\ \mathbf{q} &= \left[\, q_0 \; q_1 \; \cdots \; q_n \,\right]^T \in \mathbb{R}^{n+1}, \end{aligned}$$

and

$$\bar{\mathbf{d}}(\phi) = \left[\, \bar{d}_0 \; \bar{d}_1\phi \; \cdots \; \bar{d}_t\phi^t \,\right]^T \in \mathbb{R}^{t+1}.$$

Equation (54) is similar to (29) because it is derived from an approximation of a polynomial decomposition and the variables to be computed include the perturbations to be added to the inexact polynomials, such that the perturbed polynomials satisfy an equation rather than an approximation. Equation (54) is solved in the same manner as (29), that is, the Newton-Raphson method is used. This yields an under-determined equation, and uniqueness is imposed by the addition of a constraint, such that it is necessary to solve an LSE problem at each iteration.

It follows from Section 3.1 and this section that the computation of the coefficients of an AGCD requires the solution of an LSE problem in which the vectors and matrices are updated between successive iterations. The next section considers the convergence of this iterative procedure.

3.3 The LSE problem

It follows from Sections 3.1 and 3.2 that the computation of the coefficients of an AGCD requires the solution of a non-linear equation that yields the LSE problem,

$$\min_{\mathbf{y}} \|\mathbf{y} - \mathbf{p}\| \qquad \text{subject to} \qquad D\mathbf{y} = \mathbf{q},$$

at each iteration, where $D \in \mathbb{R}^{r \times s}, \mathbf{y}, \mathbf{p} \in \mathbb{R}^s, \mathbf{q} \in \mathbb{R}^r$ and $r < s$. This problem, which has a unique solution if D has full row rank, can be solved by the QR decomposition, as shown in Algorithm 1 [7].

Algorithm 1: The solution of the LSE problem by the QR decomposition

(a) Compute the QR decomposition of D^T,

$$D^T = QR = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix}.$$

(b) Set $\mathbf{w}_1 = R_1^{-T}\mathbf{q}$.
(c) Partition Q into

$$Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}, \tag{55}$$

where $Q_1 \in \mathbb{R}^{s \times r}$ and $Q_2 \in \mathbb{R}^{s \times (s-r)}$.
(d) Compute $\mathbf{w}_2 = Q_2^T\mathbf{p}$.
(e) Compute the solution

$$\mathbf{y} = Q \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = Q \begin{bmatrix} R_1^{-T} \\ 0 \end{bmatrix} \mathbf{q} + Q_2 Q_2^T \mathbf{p}.$$

Consider the application of this algorithm to the solution of (39). In particular, it follows from (38) and step (e) that

$$\delta\mathbf{y}^{(j)} = Q^{(j)} \begin{bmatrix} R_1^{-T} \\ 0 \end{bmatrix}^{(j)} \mathbf{q}^{(j)} + \left(Q_2 Q_2^T\right)^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right),$$

at the jth iteration. If $\bar{\mathbf{y}}$ is the solution of (39), and $\mathbf{e}^{(j)}$ and $\mathbf{e}^{(j-1)}$ are the errors at the jth and $(j-1)$th iterations, then it follows from (35) that

$$\begin{aligned}
\mathbf{e}^{(j)} &= \mathbf{y}^{(j)} - \bar{\mathbf{y}} \\
&= \mathbf{e}^{(j-1)} - \left(Q_2 Q_2^T\right)^{(j)} \mathbf{y}^{(j-1)} + Q^{(j)} \begin{bmatrix} R_1^{-T} \\ 0 \end{bmatrix}^{(j)} \mathbf{q}^{(j)} + \left(Q_2 Q_2^T\right)^{(j)} \mathbf{y}^{(0)},
\end{aligned}$$

and thus $\mathbf{e}^{(j)} - \mathbf{e}^{(j-1)} = \mathbf{y}^{(j)} - \mathbf{y}^{(j-1)}$ is given by

$$\begin{aligned}
\mathbf{e}^{(j)} - \mathbf{e}^{(j-1)} &= \left(Q_2 Q_2^T\right)^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) + Q^{(j)} \begin{bmatrix} R_1^{-T} \\ 0 \end{bmatrix}^{(j)} \mathbf{q}^{(j)} \\
&= Q^{(j)} \left(\left(Q^T\right)^{(j)} \left(Q_2 Q_2^T\right)^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) + \begin{bmatrix} R_1^{-T} \\ 0 \end{bmatrix}^{(j)} \mathbf{q}^{(j)} \right) \\
&= Q^{(j)} \left(\begin{bmatrix} 0 \\ Q_2^T \end{bmatrix}^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) + \begin{bmatrix} R_1^{-T} \\ 0 \end{bmatrix}^{(j)} \mathbf{q}^{(j)} \right),
\end{aligned}$$

from (55) because

$$Q_1^T Q_1 = I_r, \qquad Q_2^T Q_2 = I_{s-r} \qquad \text{and} \qquad Q_1^T Q_2 = 0.$$

It follows that the iterative scheme converges if

$$\lim_{j \to \infty} \left\| \mathbf{e}^{(j)} - \mathbf{e}^{(j-1)} \right\| = \lim_{j \to \infty} \left\| \begin{bmatrix} \left(R_1^{-T} \mathbf{q}\right)^{(j)} \\ \left(Q_2^T\right)^{(j)} \left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) \end{bmatrix} \right\| = 0,$$

which is independent of Q_1. Since $R_1^{(j)}$ is square and non-singular for all values of j, it follows that $\mathbf{q}^{(j)} \to \mathbf{0}$ as $j \to \infty$ for convergence, that is, the residual of (31) approaches zero at convergence. The second condition states that the difference between the initial estimate $\mathbf{y}^{(0)}$ of the solution and the solution $\mathbf{y}^{(j-1)}$ at the $(j-1)$th iteration lies in the nullspace of $\left(Q_2^T\right)^{(j)}$ as $j \to \infty$.

The convergence of the iterations for the solution of (39) can be considered from the residual $r^{(j)}$ at the jth iteration,

$$r^{(j)} = \frac{\left\| (\mathbf{c}_t + \mathbf{h}_t)^{(j)} - \left(D_t^{-1} \left(T_t + F_t\right) Q_t M_q \mathbf{x}_q \right)^{(j)} \right\|}{\left\| (\mathbf{c}_t + \mathbf{h}_t)^{(j)} \right\|}, \qquad j = 1, 2, \ldots \quad (56)$$

3.4 The transformation of the Sylvester matrix and its subresultant matrices between the Bernstein and modified Bernstein bases

This section considers the transformation of the Sylvester matrix and its subresultant matrices between the Bernstein and modified Bernstein bases, where the basis functions of the modified Bernstein basis for a polynomial of degree m are defined in (14). If a_i and b_j are the coefficients of $f(y)$ and $g(y)$ respectively, $a_i\theta^i$ and $b_j\theta^j$ are the coefficients of $\bar{f}(w)$ and $\bar{g}(w)$ respectively, and Θ_1 and Θ_2 are the diagonal matrices,

$$\begin{aligned}
\Theta_1 &= \operatorname{diag}\left[\theta^{-1} \; 1 \; \theta \; \theta^2 \; \cdots \; \theta^{m+n-k-1}\right] \in \mathbb{R}^{(m+n-k+1)\times(m+n-k+1)}, \\
\Theta_2 &= \begin{bmatrix} \Theta_{2,1} & \\ & \Theta_{2,2} \end{bmatrix} \in \mathbb{R}^{(m+n-2k+2)\times(m+n-2k+2)}, \\
\Theta_{2,1} &= \operatorname{diag}\left[\theta \; 1 \; \theta^{-1} \; \theta^{-2} \; \cdots \; \theta^{-n+k+1}\right] \in \mathbb{R}^{(n-k+1)\times(n-k+1)}, \\
\Theta_{2,2} &= \operatorname{diag}\left[\theta \; 1 \; \theta^{-1} \; \theta^{-2} \; \cdots \; \theta^{-m+k+1}\right] \in \mathbb{R}^{(m-k+1)\times(m-k+1)},
\end{aligned}$$

then the transformation between $S_k(\bar{f}, \bar{g})$ and $S_k(f, g)$ is

$$S_k(\bar{f}, \bar{g}) = \Theta_1 S_k(f, g) \Theta_2, \qquad k = 1, \ldots, \min(m, n).$$

The transformation matrices Θ_1 and Θ_2 are ill-conditioned if θ is very small or very large, or m is large, or n is large, and this equation should not, therefore, be used to transform between $S_k(\bar{f}, \bar{g})$ and $S_k(f, g)$.

The transformation matrix $T(f)$ between the coefficients a_i of $f(y)$ and the coefficients $a_i\theta^i$ of $\bar{f}(w)$ is diagonal,

$$T(f) = \operatorname{diag}\left[1 \; \theta \; \cdots \; \theta^m\right] \in \mathbb{R}^{(m+1)\times(m+1)},$$

and the transformation matrix $T(g)$ between the coefficients b_j of $g(y)$ and the coefficients $b_j\theta^j$ of $\bar{g}(w)$ is obtained by replacing m by n. The condition numbers of these matrices are

$$\kappa(T(f)) = \max\left\{\theta^m, \theta^{-m}\right\} \qquad \text{and} \qquad \kappa(T(g)) = \max\left\{\theta^n, \theta^{-n}\right\}, \tag{57}$$

depending on whether θ is greater than or less than one. They increase with m, n and θ, and they are therefore large if m or n are large.

4 Examples

This section contains two examples that show the application of the method of SNTLN to the computation of the coefficients of an AGCD of degree t of two inexact Bernstein polynomials $f(y)$ and $g(y)$. These inexact polynomials are obtained by adding noise to their exact forms, $\hat{f}(y)$ and $\hat{g}(y)$ respectively, in the componentwise sense,

$$\delta\hat{a}_i = \hat{a}_i r_i \varepsilon_i, \quad i = 0, \ldots, m, \quad \text{and} \quad \delta\hat{b}_j = \hat{b}_j r_j \varepsilon_j, \quad j = 0, \ldots, n, \tag{58}$$

where r_i and r_j are uniformly distributed random numbers in the interval $[-1, 1]$, and ε_i and ε_j are uniformly distributed random numbers in an interval $\mathcal{I}$ whose lower and upper bounds define the range of the reciprocal of the signal-to-noise ratio of the coefficients of $f(y)$ and $g(y)$. It follows from (58) that these coefficients are, respectively,

$$a_i = \hat{a}_i + \delta\hat{a}_i, \quad i = 0, \ldots, m, \qquad \text{and} \qquad b_j = \hat{b}_j + \delta\hat{b}_j, \quad j = 0, \ldots, n. \tag{59}$$

Example 4.1 The polynomials $\hat{f}(y)$ and $\hat{g}(y)$ are

$$\begin{aligned} \hat{f}(y) &= \sum_{i=0}^{19} \hat{a}_i \binom{19}{i} (1-y)^{19-i} y^i \\ &= (y-0.10)^4 (y-0.30)^2 (y-0.50)^2 (y-0.70)^3 (y-0.80)^2 \times \\ &\quad (y-2.50)^3 (y+3.40)^3, \end{aligned}$$

and

$$\begin{aligned} \hat{g}(y) &= \sum_{i=0}^{16} \hat{b}_i \binom{16}{i} (1-y)^{16-i} y^i \\ &= (y-0.10)^3 (y-0.80)^2 (y-0.85)^4 (y-0.90)^4 (y-1.10)^3, \end{aligned}$$

and thus $\hat{t} = \deg \operatorname{GCD}(\hat{f}, \hat{g}) = 5$. The coefficients of $\hat{f}(y)$ and $\hat{g}(y)$ were perturbed, thereby forming the inexact polynomials $f(y)$ and $g(y)$, where

$\mathcal{I} = [10^{-10}, 10^{-8}]$, as shown in (58) and (59), which yielded normwise relative errors in $f(y)$ and $g(y)$ of 2.85×10^{-9} and 2.55×10^{-9} respectively. The polynomials were preprocessed, as discussed in Section 3, and the method of SNTLN was used to compute a structured low rank approximation of $S(\bar{f}, \alpha_0 \bar{g})$, as described in Section 3.1.

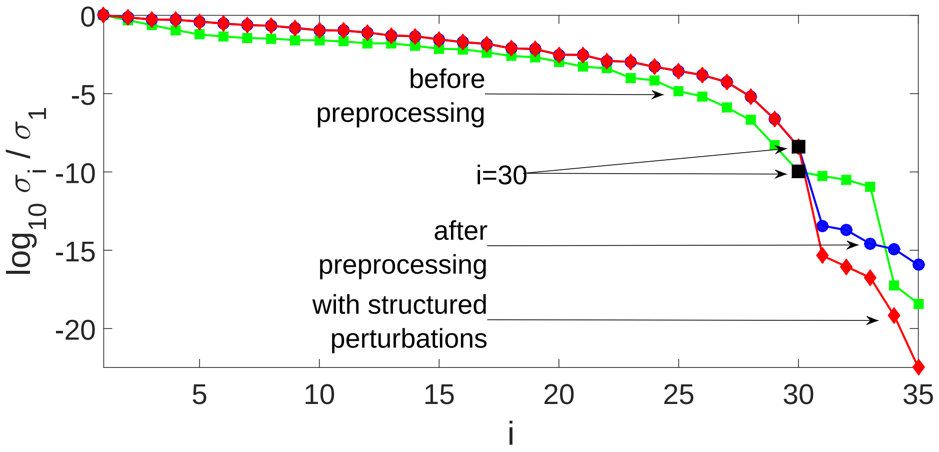

Fig. 1. The normalised singular values σ_i/σ_1 of the Sylvester matrix of (i) $f(y)$ and $g(y)$ 🟩 (before preprocessing), (ii) $\bar{f}(w)$ and $\alpha_0 \bar{g}(w)$ ● (after preprocessing), and (iii) $\tilde{f}(w)$ and $\beta^* \tilde{g}(w)$ ♦ (with structured perturbations), for Example 4.1. The point $i = 30$, which corresponds to the degree five of the GCD of $\hat{f}(y)$ and $\hat{g}(y)$, is marked.

Figure 1 shows the normalised singular values of the Sylvester matrix of (i) the given inexact polynomials $f(y)$ and $g(y)$, (ii) the polynomials $\bar{f}(w)$ and $\alpha_0 \bar{g}(w)$, which are defined in (15) and (16), and (iii) the polynomials $\tilde{f}(w)$ and $\beta^* \tilde{g}(w)$, which are defined in (40) and (41). It is seen that the numerical rank of $S(f, g)$ is 33, which is incorrect because this implies that the degree of an AGCD of $f(y)$ and $g(y)$ is two. The figure shows the importance of the preprocessing operations because the rank loss of $S(\bar{f}, \alpha_0 \bar{g})$ and $S(\tilde{f}, \beta^* \tilde{g})$ is 5, which is correct, and the inclusion of the structured perturbations improved the results slightly because the numerical rank is more clearly defined. The normwise relative errors in the computed coprime polynomials and AGCD in the Bernstein basis were

$$e\left(\tilde{u}\left(w = y/\theta^*\right)\right) = 1.25 \times 10^{-6}, \quad e\left(\tilde{v}\left(w = y/\theta^*\right)\right) = 4.86 \times 10^{-8}, \tag{60}$$

and

$$e\left(\tilde{d}\left(w = y/\theta^*\right)\right) = 8.07 \times 10^{-7}, \tag{61}$$

which are between one and three orders of magnitude larger than the relative errors in $f(y)$ and $g(y)$. Since $\theta^* = 1.6133$, it follows from (57) that $\kappa(T(f)) = 8.84 \times 10^3$ and $\kappa(T(g)) = 2.11 \times 10^3$.

The LSE problem requires the iterative solution of (39) and Figure 2 shows the variation of the error $r^{(j)}$, which is defined in (56). It is seen that convergence is achieved after a few iterations and that $r^{(j)} \approx 10^{-15}$ at convergence.

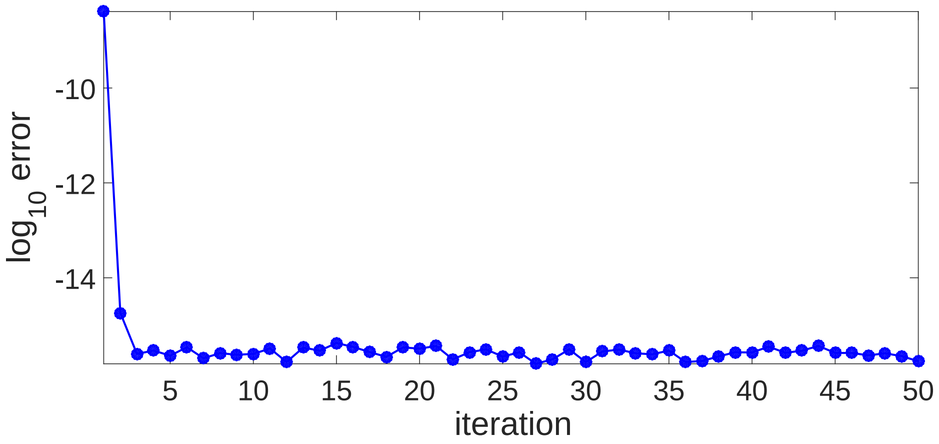

Fig. 2. The error $r^{(j)}$ of (39) against the iteration counter j for Example 4.1.

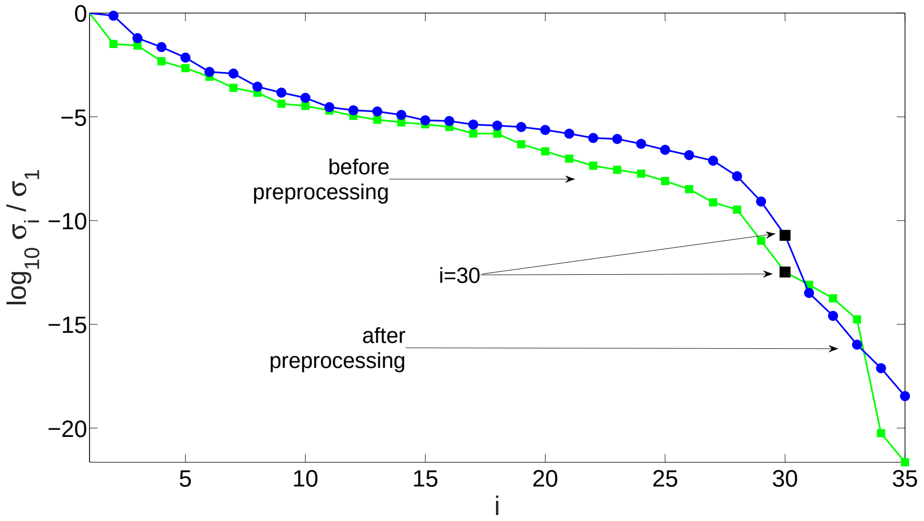

Fig. 3. The normalised singular values σ_i/σ_1 of the Sylvester matrix of (i) $f(y)$ and $g(y)$ 🟩 (before preprocessing), and (ii) $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ ● (after preprocessing), for Example 4.1. The point $i = 30$, which corresponds to the degree five of the GCD of $\hat{f}(y)$ and $\hat{g}(y)$, is marked.

The computations were repeated but the standard form $D_k^{-1}T_k(f,g), k = 1,\ldots,\min(m,n)$, of the Sylvester matrix and its subresultant matrices was used. Figure 3 shows the normalised singular values of the Sylvester matrix of the given inexact polynomials $f(y)$ and $g(y)$, and the Sylvester matrix of the polynomials $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ after preprocessing and before the application of the method of SNTLN. Unsatisfactory results were obtained because the numerical rank of these matrices is either not defined or incorrect, even though the methods described in [4] for the computation of t yielded $t = \hat{t} = 5$

and the errors in the coprime polynomials and AGCD were approximately equal to the errors (60) and (61). This result shows that even if the errors of two AGCDs are approximately equal, it does not follow that the structured low rank approximations of the Sylvester matrices will also be approximately equal. Also, it is clear that significantly improved results were obtained when the modified Sylvester matrix and its subresultant matrices $D_k^{-1}T_k(f,g)Q_k$, rather than the standard form $D_k^{-1}T_k(f,g)$ of these matrices, were used.

The coefficients of an AGCD of degree t were also computed by an approximate factorisation of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$, which is described in Section 3.2. Very similar results were obtained and the iterations in the LSE problem converged rapidly to a solution with a very small error. □

Example 4.2 The procedure described in Example 4.1 was repeated for the polynomials,

$$\begin{aligned}\hat{f}(y) &= \sum_{i=0}^{16} \hat{a}_i \binom{16}{i} (1-y)^{16-i} y^i \\ &= (y-0.23)^4(y-0.43)^3(y-0.57)^3(y-0.92)^3(y-1.70)^3,\end{aligned}$$

and

$$\begin{aligned}\hat{g}(y) &= \sum_{i=0}^{18} \hat{b}_i \binom{18}{i} (1-y)^{18-i} y^i \\ &= (y-0.23)^4(y-0.30)^2(y-0.77)^5(y-0.92)^2(y-1.20)^5,\end{aligned}$$

except that the interval $\mathcal{I}$ was equal to $[10^{-8}, 10^{-6}]$. The method of SNTLN was applied to the perturbed polynomials after they were preprocessed, and an approximate factorisation of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ was computed. The relative errors in $f(y)$ and $g(y)$ were 5.85×10^{-7} and 2.18×10^{-7} respectively.

Figure 4 shows the normalised singular values of the Sylvester matrices of the pairs of polynomials $(f(y), g(y))$, $(\bar{f}(w), \alpha_0\bar{g}(w))$ and $(\tilde{f}(w), \beta^*\tilde{g}(w))$. The figure is similar to Figure 1 because the best result is obtained when $f(y)$ and $g(y)$ are preprocessed and the method of SNTLN is used to compute an AGCD. In particular, it is seen that bad results are obtained when the method of SNTLN is not applied because the numerical rank of the Sylvester matrices $S(f,g)$ and $S(\bar{f}, \alpha_0\bar{g})$ is not defined. By contrast, the numerical rank of $S(\tilde{f}, \beta^*\tilde{g})$ is clearly defined and deg AGCD $(\tilde{f}, \beta^*\tilde{g}) =$ deg GCD $(\hat{f}, \hat{g}) = 6$.

The normwise relative errors in the computed coprime polynomials and AGCD in the Bernstein basis are

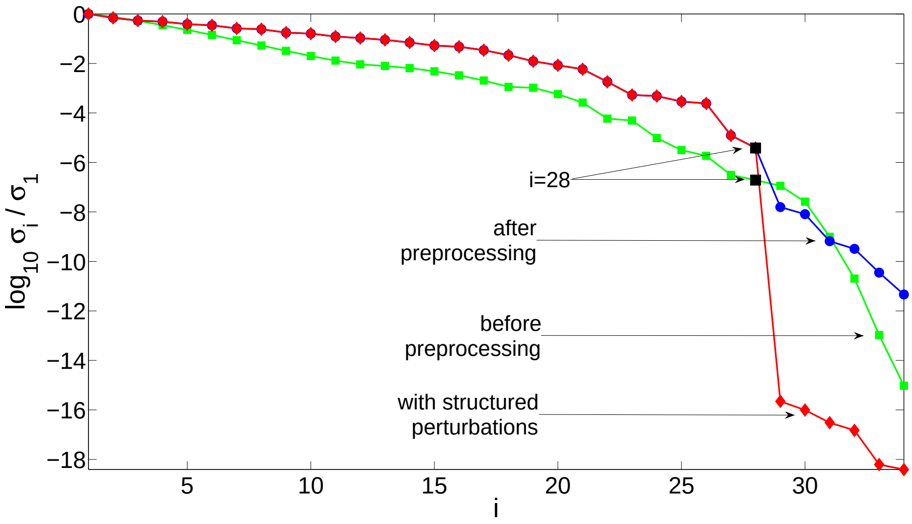

Fig. 4. The normalised singular values σ_i/σ_1 of the Sylvester matrix of (i) $f(y)$ and $g(y)$ 🟩 (before preprocessing), (ii) $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ 🔵 (after preprocessing), and (iii) $\tilde{f}(w)$ and $\beta^*\tilde{g}(w)$ ♦ (with structured perturbations), for Example 4.2. The point $i = 28$, which corresponds to the degree six of the GCD of $\hat{f}(y)$ and $\hat{g}(y)$, is marked.

$$e\left(\tilde{u}\left(w = y/\theta^*\right)\right) = 1.73 \times 10^{-4}, \qquad e\left(\tilde{v}\left(w = y/\theta^*\right)\right) = 1.36 \times 10^{-4},$$

and

$$e\left(\tilde{d}\left(w = y/\theta^*\right)\right) = 7.97 \times 10^{-4},$$

which are about three orders of magnitude larger than the relative errors in $f(y)$ and $g(y)$. Since $\theta^* = 1.7071$, this increase in errors may be due to the large condition numbers of $T(f)$ and $T(g)$, $\kappa(T(f)) = 5.20 \times 10^3$ and $\kappa(T(g)) = 1.52 \times 10^4$ respectively, as discussed in Example 4.1.

Figure 5 shows the convergence of the iterations for the solution of the LSE problem and it is seen that it is similar to Figure 2 because good convergence is achieved very rapidly.

An AGCD of $f(y)$ and $g(y)$ was also computed when the standard form $D_k^{-1}T_k(f,g)$ of the Sylvester matrix and its subresultant matrices of $f(y)$ and $g(y)$ was used, and the results are shown in Figure 6. This figure is similar to Figure 3 because bad results were obtained since the numerical rank of each of these matrices is not defined.

A structured low rank approximation of the Sylvester matrix of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ was then used to compute an AGCD of $f(y)$ and $g(y)$. The results were very similar to the results obtained from an approximate factorisation of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$, which is consistent with the results of Example 4.1. □

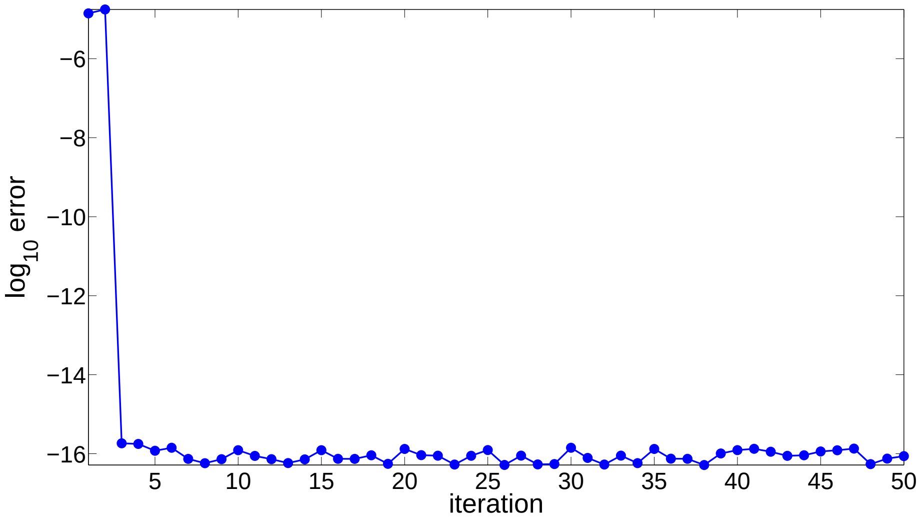

Fig. 5. The error $r^{(j)}$ of (39) against the iteration counter j for Example 4.2.

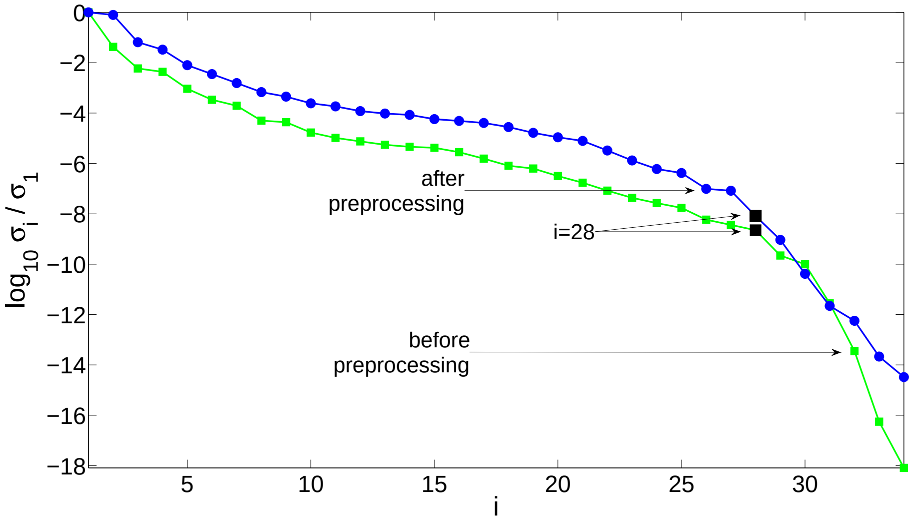

Fig. 6. The normalised singular values σ_i/σ_1 of the Sylvester matrix of (i) $f(y)$ and $g(y)$ 🟩 (before preprocessing), and (ii) $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$ ● (after preprocessing) for Example 4.2. The point $i = 28$, which corresponds to the degree six of the GCD of $\hat{f}(y)$ and $\hat{g}(y)$, is marked.

5 Summary

This paper has considered two applications of the method of SNTLN to the computation of the coefficients of an AGCD of two Bernstein polynomials. These applications are the calculation of a structured low rank approximation of the Sylvester matrix $S(\bar{f}, \alpha_0\bar{g})$ of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$, and an approximate factorisation of $\bar{f}(w)$ and $\alpha_0\bar{g}(w)$. It was shown that $f(y)$ and $g(y)$ must be processed by three operations before an AGCD is computed, and that the effect of one of these operations is a change in the basis, from the Bernstein basis to the modified Bernstein basis. The method of SNTLN yields a non-linear equation that is solved iteratively, where each iteration requires the solution of an LSE problem. It was shown that the standard form $D_t^{-1}T_t(\bar{f},\alpha_0\bar{g})$ of the tth subresultant matrix does not yield good results for the coefficients of an AGCD, of degree t, of $f(y)$ and $g(y)$, and that better results are obtained when the modified subresultant matrix $D_t^{-1}T_t(\bar{f},\alpha_0\bar{g})Q_t$ is used. Many examples showed that the AGCD computed from a structured low rank approximation of a Sylvester matrix and an approximate polynomial factorisation of $f(y)$ and $g(y)$ are very similar, and the results in this paper are therefore typical.

References

[1] S. Barnett. *Polynomials and Linear Control Systems.* Marcel Dekker, New York, USA, 1983.

[2] D. A. Bini and P. Boito. A fast algorithm for approximate polynomial GCD based on structured matrix computations. In D. A. Bini, V. Mehrmann, V. Olshevsky, E. Tyrtshnikov, and M. Van Barel, editors, *Numerical Methods for Structured Matrices and Applications: The Georg Heinig Memorial Volume*, pages 155–173. Birkhäuser, 2010.

[3] D. A. Bini and L. Gemignani. Bernstein-Bezoutian matrices. *Theoretical Computer Science*, 315:319–333, 2004.

[4] M. Bourne, J. R. Winkler, and S. Yi. The computation of the degree of an approximate greatest common divisor of two Bernstein polynomials, 2015. Submitted.

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

[6] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, Cambridge, MA, 1993.

[7] G. H. Golub and C. F. Van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, USA, 2013.

[8] E. Kaltofen, Z. Yang, and L. Zhi. Structured low rank approximation of a Sylvester matrix. In D. Wang and L. Zhi, editors, *Trends in Mathematics*, pages 69–83. Birkhäuser Verlag, Basel, Switzerland, 2006.

[9] Y. S. Lai and F. L. Chen. Implicitizing rational surfaces using moving quadrics constructed from moving planes. *Journal of Symbolic Computation*, 77:127–161, 2016.

[10] Y. S. Lai, W. P. Du, and R. H. Wang. The Viro method for construction of Bernstein-Bézier algebraic hypersurfaces. *Science China Mathematics*, 55(6):1269–1279, 2012.

[11] Y. S. Lai, W. P. Du, and R. H. Wang. The Viro method for construction of piecewise algebraic hypersurfaces. *Abstract Applied Analysis*, 2013. doi:10.1155/2013/690341.

[12] J. Ben Rosen, H. Park, and J. Glick. Structured total least norm for nonlinear problems. *SIAM J. Mat. Anal. Appl.*, 20(1):14–30, 1998.

[13] P. Stoica and T. Söderström. Common factor detection and estimation. *Automatica*, 33(5):985–989, 1997.

[14] Y. Tsai and R. T. Farouki. Algorithm 812: BPOLY: An object-oriented library for numerical algorithms for polynomials in Bernstein form. *ACM Trans. Mathematical Software*, 27(2):267–296, 2001.

[15] J. R. Winkler. A resultant matrix for scaled Bernstein polynomials. *Linear Algebra and Its Applications*, 319:179–191, 2000.

[16] J. R. Winkler. A companion matrix resultant for Bernstein polynomials. *Linear Algebra and Its Applications*, 362:153–175, 2003.

[17] J. R. Winkler. Structured matrix methods for the computation of multiple roots of a polynomial. *Journal of Computational and Applied Mathematics*, 272:449–467, 2014.

[18] J. R. Winkler and R. N. Goldman. The Sylvester resultant matrix for Bernstein polynomials. In T. Lyche, M. Mazure, and L. L. Schumaker, editors, *Curve and Surface Design: Saint-Malo 2002*, pages 407–416, Tennessee, USA, 2003. Nashboro Press.

[19] J. R. Winkler and M. Hasan. A non-linear structure preserving matrix method for the low rank approximation of the Sylvester resultant matrix. *Journal of Computational and Applied Mathematics*, 234:3226–3242, 2010.

[20] J. R. Winkler and M. Hasan. An improved non-linear method for the computation of a structured low rank approximation of the Sylvester resultant matrix. *Journal of Computational and Applied Mathematics*, 237(1):253–268, 2013.

[21] J. R. Winkler, M. Hasan, and X. Y. Lao. Two methods for the calculation of the degree of an approximate greatest common divsior of two inexact polynomials. *Calcolo*, 49:241–267, 2012.

[22] J. R. Winkler, X. Y. Lao, and M. Hasan. The computation of multiple roots of a polynomial. *Journal of Computational and Applied Mathematics*, 236:3478–3497, 2012.

[23] J. R. Winkler and N. Yang. Resultant matrices and the computation of the degree of an approximate greatest common divisor of two inexact Bernstein basis polynomials. *Computer Aided Geometric Design*, 30(4):410–429, 2013.

[24] Z. Zeng. The approximate GCD of inexact polynomials. Part 1: A univariate algorithm, 2004. Preprint.

[25] L. Zhi. Displacement structure in computing the approximate GCD of univariate polynomials. In W. Sit and Z. Li, editors, *Lecture Notes Series on Computing*, pages 288–298. World Scientific, 2003.

[26] L. Zhi and Z. Yang. Computing approximate GCD of univariate polynomials by structured total least norm. Technical report, Institute of Systems Science, AMSS, Academia Sinica, Beijing, China, 2004.