
Article:
Winkler, J.R. and Yang, N. (2014) A structure-preserving matrix method for the deconvolution of two Bernstein basis polynomials. Computer Aided Geometric Design, 31 (6). 317 - 328. ISSN 0167-8396

https://doi.org/10.1016/j.cagd.2014.02.009

A structure-preserving matrix method for the deconvolution of two Bernstein basis polynomials

Joab R. Winkler, [a] Ning Yang [a]

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

j.winkler@dcs.shef.ac.uk, acp08ny@sheffield.ac.uk

Abstract

This paper describes the application of a structure-preserving matrix method to the deconvolution of two Bernstein basis polynomials. Specifically, the deconvolution $\hat{h}/\hat{f}$ yields a polynomial $\hat{g}$ provided the exact polynomial $\hat{f}$ is a divisor of the exact polynomial $\hat{h}$ and all computations are performed symbolically. In practical situations, however, inexact forms, h and f of, respectively, $\hat{h}$ and $\hat{f}$ are specified, in which case $g = h/f$ is a rational function and not a polynomial. The simplest method to calculate the coefficients of g is the least squares minimisation of an over-determined system of linear equations in which the coefficient matrix is Tœplitz, but the solution is a polynomial approximation of a rational function. It is shown in this paper that an improved result for g is obtained when the Tœpliz structure of the coefficient matrix is preserved, that is, a structure-preserving matrix method is used. In particular, this method guarantees that a polynomial solution to the deconvolution h/f is obtained, and thus an essential property of the theoretically exact solution is retained in the computed solution. Computational examples that show the improvement in the solution obtained from the structure-preserving matrix method with respect to the least squares solution are presented.

Key words: Polynomial deconvolution; Bernstein basis polynomials; structure-preserving matrix methods

1 Introduction

The Bernstein polynomial basis is used in computer aided geometric design because of its elegant geometric properties and superior numerical properties with respect to those of the power basis [2]. Algorithms for the addition, multiplication, division and subdivision of Bernstein basis polynomials have been

developed [3,4], and this paper considers the deconvolution of two Bernstein basis polynomials h and f, that is, the computation of the polynomial $g = h/f$. If $h = \hat{h}$ and $f = \hat{f}$ where $\hat{h}$ and $\hat{f}$ are the exact forms of h and f respectively, $\hat{f}$ is an exact divisor of $\hat{h}$, and all computations are performed symbolically, then $\hat{g} = \hat{h}/\hat{f}$ is a polynomial. In practical problems, however, the polynomials h and f are inexact, in which case $g = h/f$ is a rational function and not a polynomial. It is shown in this paper that in this circumstance, a polynomial is returned when a structure-preserving matrix method is used to compute g. This solution must be compared with the simplest solution, that is, the least squares (LS) solution of an over-determined set of linear equations, in which case a polynomial approximation of a rational function is returned.

The approximation of a rational function by a polynomial may be adequate in some applications, but other applications may require that the algorithm for polynomial deconvolution return a polynomial, and not an approximation of a polynomial. The requirement that a polynomial be returned arises in the computation of multiple roots of a polynomial because the algorithm includes a series of deconvolutions in which the denominator polynomial is an exact divisor of the numerator polynomial [8]. The discussion above shows that if the polynomials are subject to error and/or the computations are performed in finite precision arithmetic, it cannot be guaranteed that the result of each deconvolution is a polynomial. In this circumstance, the algorithm returns incorrect results or fails, and it is shown in this paper that a structure-preserving matrix method returns polynomial solutions to these deconvolutions, which is, as noted above, essential for the computation of multiple roots of a polynomial.

The deconvolution of two Bernstein basis polynomials is considered in Section 2. It is shown that the problem can be cast as the computation of the vector $\hat{\mathbf{g}}$, which stores the coefficients of $\hat{g}$, from the equation,

$$D^{-1}T(\hat{f})\hat{\mathbf{g}} = \hat{\mathbf{h}}, \tag{1}$$

where D^{-1} is a diagonal matrix of combinatorial factors, $T(\hat{f})$ is a Tœplitz matrix whose non-zero entries are the coefficients of $\hat{f}$, and $\hat{\mathbf{h}}$ stores the coefficients of $\hat{h}$. The Tœplitz structure of $T(\hat{f})$ is retained when inexact polynomials h and f are specified, but $\mathbf{h}$, the vector that contains the coefficients of h, does not lie in the column space of $D^{-1}T(f)$ in this circumstance, and thus (1) is replaced by

$$D^{-1}T(f)\mathbf{g} \approx \mathbf{h}. \tag{2}$$

The approximation in (2) implies that a computed solution $\mathbf{w}$ has a non-zero error, $\|D^{-1}T(f)\mathbf{w} - \mathbf{h}\| > 0$, from which it follows that the entries of $\mathbf{w}$ are the coefficients of a polynomial that is an approximation of a rational function.

It is shown in Section 3 that if the Tœplitz structure of $T(f)$ is retained in the computations, that is, a structure-preserving matrix method is used, an exact polynomial, and not a polynomial approximation of a rational function, is returned when an approximate solution of (2) is computed.

The application of a structure-preserving matrix method to the deconvolution of two Bernstein basis polynomials is considered in Section 4. This method requires the iterative solution of a non-linear equation, and the condition for its convergence is discussed in Section 5. Section 6 contains two examples that compare the solutions obtained from the method of LS with the solutions obtained from a structure-preserving matrix method. The solutions are discussed in Section 7, and Section 8 contains a summary of the paper.

2 The deconvolution of two Bernstein polynomials

This section considers the deconvolution of two Bernstein polynomials and it is shown it can be considered in the matrix form (1).

Let $\hat{f}(y), \hat{g}(y)$ and $\hat{h}(y)$ be Bernstein polynomials of degrees m, n and $m+n$ respectively,

$$\hat{f}(y) = \sum_{i=0}^{m} \hat{a}_i \binom{m}{i} (1-y)^{m-i} y^i, \tag{3}$$

$$\hat{g}(y) = \sum_{i=0}^{n} \hat{b}_i \binom{n}{i} (1-y)^{n-i} y^i, \tag{4}$$

$$\hat{h}(y) = \sum_{i=0}^{m+n} \hat{c}_i \binom{m+n}{i} (1-y)^{m+n-i} y^i. \tag{5}$$

It follows from (3) and (4) that the coefficients $\hat{c}_i$ in (5) are given by

$$\hat{c}_k = \sum_{i=\max(0,k-n)}^{\min(m,k)} \frac{\hat{a}_i \binom{m}{i} \hat{b}_{k-i} \binom{n}{k-i}}{\binom{m+n}{k}}, \qquad k = 0, \ldots, m+n,$$

which can be written in matrix form as

$$\left(D^{-1} T(\hat{f}) \right) \hat{\mathbf{b}} = \hat{\mathbf{c}}, \tag{6}$$

where

$$D^{-1} = \text{diag}\left[\begin{array}{cccc} \frac{1}{\binom{m+n}{0}} & \frac{1}{\binom{m+n}{1}} & \cdots & \frac{1}{\binom{m+n}{m+n-1}} \quad \frac{1}{\binom{m+n}{m+n}} \end{array}\right] \in \mathbb{R}^{(m+n+1)\times(m+n+1)},$$

$T = T(\hat{f}) \in \mathbb{R}^{(m+n+1)\times(n+1)}$, $\hat{\mathbf{b}} \in \mathbb{R}^{n+1}$, $\hat{\mathbf{c}} \in \mathbb{R}^{m+n+1}$ and

$$T = \begin{bmatrix} \hat{a}_0\binom{m}{0} & & & \\ \hat{a}_1\binom{m}{1} & \hat{a}_0\binom{m}{0} & & \\ \vdots & \hat{a}_1\binom{m}{1} & \ddots & \\ \vdots & \vdots & \ddots & \hat{a}_0\binom{m}{0} \\ \hat{a}_m\binom{m}{m} & \vdots & \ddots & \hat{a}_1\binom{m}{1} \\ & \hat{a}_m\binom{m}{m} & \ddots & \vdots \\ & & \ddots & \vdots \\ & & & \hat{a}_m\binom{m}{m} \end{bmatrix}, \quad \hat{\mathbf{b}} = \begin{bmatrix} \hat{b}_0\binom{n}{0} \\ \hat{b}_1\binom{n}{1} \\ \vdots \\ \vdots \\ \hat{b}_n\binom{n}{n} \end{bmatrix}, \quad \hat{\mathbf{c}} = \begin{bmatrix} \hat{c}_0 \\ \hat{c}_1 \\ \vdots \\ \vdots \\ \hat{c}_{m+n} \end{bmatrix}.$$

Previous work [11,12] has shown it is numerically advantageous to express $\hat{\mathbf{b}}$ as the product of a diagonal matrix $Q \in \mathbb{R}^{(n+1)\times(n+1)}$ and a vector $\hat{\mathbf{p}} \in \mathbb{R}^{n+1}$ of the coefficients $\hat{b}_i$,

$$\hat{\mathbf{b}} = Q\hat{\mathbf{p}}, \qquad Q = \text{diag}\left[\begin{array}{cccc} \binom{n}{0} & \binom{n}{1} & \cdots & \binom{n}{n} \end{array}\right], \qquad \hat{\mathbf{p}} = \left[\begin{array}{cccc} \hat{b}_0 & \hat{b}_1 & \cdots & \hat{b}_n \end{array}\right]^T,$$

and thus (6) can be written as

$$\Big(D^{-1}T(\hat{f})Q\Big)\hat{\mathbf{p}} = \hat{\mathbf{c}}, \tag{7}$$

where the coefficient matrix is of order $(m+n+1)\times(n+1)$. It is better to calculate the coefficients of $\hat{g}(y)$ from (7) than from (6) because numerous experiments showed that

$$\kappa\Big(D^{-1}T(\hat{f})Q\Big) < \kappa\Big(D^{-1}T(\hat{f})\Big),$$

where $\kappa(X)$ denotes the condition number of X. It therefore follows that (7) is more stable than (6), and thus improved solutions are expected from (7).

The magnitudes of the entries in the coefficient matrix and right hand side vector of (7) may differ by several orders of magnitude, and it is therefore advantageous to normalise them. This issue is discussed in [10,12], and it is shown that the normalisation of the entries in the coefficient matrix, and the

normalisation of the entries in the right hand side vector, by their geometric means yield significantly improved results. The geometric mean of the terms that contain the coefficients of $\hat{f}$ in $D^{-1}T(\hat{f})Q$ is [12]

$$\lambda = \frac{\left(\prod_{i=0}^{m}\left|\hat{a}_i\binom{m}{i}\right|\right)^{\frac{1}{m+1}}\left(\prod_{k=0}^{n}\binom{n}{k}\right)^{\frac{1}{n+1}}}{\left(\prod_{i=0}^{m}P_i\right)^{\frac{1}{(n+1)(m+1)}}}, \tag{8}$$

where

$$P_i = \prod_{j=i}^{i+n}\binom{m+n}{j}, \qquad i = 0, \ldots, m,$$

and thus the normalised form of $\hat{f}(y)$ is

$$\bar{f}(y) = \sum_{i=0}^{m}\bar{a}_i\binom{m}{i}(1-y)^{m-i}y^i, \qquad \bar{a}_i = \frac{\hat{a}_i}{\lambda}. \tag{9}$$

The normalised form of $\hat{h}(y)$ is

$$\bar{h}(y) = \sum_{i=0}^{m+n}\bar{c}_i\binom{m+n}{i}(1-y)^{m+n-i}y^i, \qquad \bar{c}_i = \frac{\hat{c}_i}{\mu}, \tag{10}$$

where the geometric mean μ of the coefficients $\hat{c}_i$ is

$$\mu = \left(\prod_{i=0}^{m+n}|\hat{c}_i|\right)^{\frac{1}{m+n+1}}. \tag{11}$$

It therefore follows from (9) and (10) that (7) becomes

$$\left(D^{-1}T(\bar{f})Q\right)\bar{\mathbf{p}} = \bar{\mathbf{c}}, \tag{12}$$

where $\bar{\mathbf{c}} \in \mathbb{R}^{m+n+1}$ and $\bar{\mathbf{p}} \in \mathbb{R}^{n+1}$ are, respectively,

$$\bar{\mathbf{c}} = \begin{bmatrix}\bar{c}_0 & \bar{c}_1 & \cdots & \bar{c}_{m+n}\end{bmatrix}^T \qquad \text{and} \qquad \bar{\mathbf{p}} = \begin{bmatrix}\bar{b}_0 & \bar{b}_1 & \cdots & \bar{b}_n\end{bmatrix}^T,$$

and it is required to compute the coefficients $\bar{b}_i$ of the polynomial $\bar{g}(y)$,

$$\bar{g}(y) = \sum_{i=0}^{n} \bar{b}_i \binom{n}{i} (1-y)^{n-i} y^i.$$

The normalisation of the coefficients of $\hat{f}(y)$ retains the Tœplitz structure of T, but the inclusion of the diagonal matrices D^{-1} and Q implies that the coefficient matrix in (12) is not Tœplitz, but it is still structured. It is appropriate to consider, therefore, a structure-preserving matrix method for the solution of (12), and this issue is addressed in the next section.

3 A structure-preserving matrix method

This section considers the application of a structure-preserving matrix method to the solution of (12), and the difference between the solution obtained using this method, and the solution obtained by solving a LS problem, is described.

If exact polynomials are considered and all computations are performed in infinite precision arithmetic, the coefficients of $\bar{g}$ can be computed from the LS solution of (12),

$$\bar{\mathbf{p}} = \left(D^{-1} T(\bar{f}) Q \right)^{\dagger} \bar{\mathbf{c}}, \qquad A^{\dagger} = (A^T A)^{-1} A^T,$$

and the error is zero. The pseudo-inverse $A^{\dagger}$ of a matrix A should be computed from the singular value decomposition (SVD) of A and not from the SVD USV^T of $A^{\dagger}$ because this can lead to large errors, such that $USV^T \neq A^{\dagger}$.

If the inexact polynomials $f(y)$ and $h(y)$ are considered, then (12) is replaced by an approximate equation whose LS solution defines the coefficients of a polynomial that is an approximation of a rational function. This is the simplest method of computing the coefficients of $\bar{g}(y)$, but it fails to consider the structure of the coefficient matrix. Since $f(y), g(y)$ and $h(y)$ are inexact forms of $\hat{f}(y), \hat{g}(y)$ and $\hat{h}(y)$ respectively, they are given by

$$f(y) = \hat{f}(y) + \delta \hat{f}(y) = \sum_{i=0}^{m} a_i \binom{m}{i} (1-y)^{m-i} y^i, \tag{13}$$

$$g(y) = \hat{g}(y) + \delta \hat{g}(y) = \sum_{i=0}^{n} b_i \binom{n}{i} (1-y)^{n-i} y^i,$$

$$h(y) = \hat{h}(y) + \delta \hat{h}(y) = \sum_{j=0}^{m+n} c_j \binom{m+n}{j} (1-y)^{m+n-j} y^j, \tag{14}$$

where

$$a_i = \hat{a}_i + \delta\hat{a}_i \qquad \text{and} \qquad c_j = \hat{c}_j + \delta\hat{c}_j, \tag{15}$$

and normalisation of the coefficients a_i and c_j by, respectively, the geometric means λ and μ, which are defined in (8) and (11), is implicitly included. This normalisation by the geometric means of the coefficients of $f(y)$ and $g(y)$ follows from the discussion in Section 2.

It follows from (12) that it is necessary to compute the LS solution of

$$\left(D^{-1}T(f)Q\right)\mathbf{p} \approx \mathbf{c}, \tag{16}$$

where

$$\mathbf{p} = \begin{bmatrix} b_0 \; b_1 \; \cdots \; b_n \end{bmatrix}^T \qquad \text{and} \qquad \mathbf{c} = \begin{bmatrix} c_0 \; c_1 \; \cdots \; c_{m+n} \end{bmatrix}^T.$$

The vector $\mathbf{c}$ does not lie in the column space of $D^{-1}T(f)Q$, but the approximation (16) can be transformed to an equation by perturbing $T(f)$ by a Tœplitz matrix $B(z)$ that has the same structure as $T(f)$, and perturbing $\mathbf{c}$ by a vector $\mathbf{t}$. A structure-preserving matrix method yields, therefore, a polynomial solution to the deconvolution problem because (16) is replaced by

$$\left(D^{-1}\left(T(f) + B(z)\right)Q\right)\mathbf{p} = \mathbf{c} + \mathbf{t}, \tag{17}$$

where the entries of $B(z)$ and $\mathbf{t}$ are the coefficients of the polynomials $s(y)$ and $e(y)$ respectively,

$$s(y) = \sum_{i=0}^{m} z_i \binom{m}{i} (1-y)^{m-i} y^i, \tag{18}$$

and

$$e(y) = \sum_{i=0}^{m+n} t_i \binom{m+n}{i} (1-y)^{m+n-i} y^i, \tag{19}$$

and the matrix $B(z)$ and vector $\mathbf{t}$ are given by, respectively,

$$B(z) = \begin{bmatrix} z_0\binom{m}{0} & & & \\ z_1\binom{m}{1} & z_0\binom{m}{0} & & \\ \vdots & z_1\binom{m}{1} & \ddots & \\ \vdots & \vdots & \ddots & z_0\binom{m}{0} \\ z_m\binom{m}{m} & \vdots & \ddots & z_1\binom{m}{1} \\ & z_m\binom{m}{m} & \ddots & \vdots \\ & & \ddots & \vdots \\ & & & z_m\binom{m}{m} \end{bmatrix}, \qquad \mathbf{t} = \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ \vdots \\ t_{m+n} \end{bmatrix}.$$

If the matrix $B(z)$ and vector $\mathbf{t}$ are chosen such that $\mathbf{c} + \mathbf{t}$ lies in the column space of $D^{-1}(T(f) + B(z))Q$, then (17) has an exact solution, and thus the solution vector $\mathbf{p}$ contains the coefficients of the polynomial formed from the deconvolution,

$$\frac{h(y) + e(y)}{f(y) + s(y)} = \frac{\sum_{i=0}^{m+n}(c_i + t_i)\binom{m+n}{i}(1-y)^{m+n-i}y^i}{\sum_{i=0}^{m}(a_i + z_i)\binom{m}{i}(1-y)^{m-i}y^i}. \tag{20}$$

An infinite number of pairs of polynomials $(s(y), e(y))$ satisfy (17), but it is desired to compute the coefficients z_i and t_i of minimum magnitude, that is, the solution of (17) that is nearest the solution defined by the given inexact data is sought. This is expressed mathematically as an LS minimisation with an equality constraint, the LSE problem,

$$\min \|\mathbf{z}\|^2 + \|\mathbf{t}\|^2 \quad \text{such that} \quad \Big(D^{-1}\left(T(f) + B(z)\right)Q\Big)\mathbf{p} = \mathbf{c} + \mathbf{t}, \tag{21}$$

where $\|\cdot\| = \|\cdot\|_2$,

$$\mathbf{z} = \begin{bmatrix} z_0 \; z_1 \; \cdots \; z_m \end{bmatrix}^T \qquad \text{and} \qquad z = \Big\{ z_0, \, z_1, \, \cdots, \, z_m \Big\}. \tag{22}$$

It follows from (21) that a structure-preserving matrix method perturbs the given inexact polynomials the minimum amount, such that the perturbed polynomials satisfy the exact divisor condition. This is desirable because the computed solution satisfies a property of the theoretical solution, which is different from the LS solution, which does not satisfy this condition.

4 The method of STLN for the deconvolution of two polynomials

This section considers the application of the method of structured total least norm (STLN) [6] to the deconvolution h/f, such that the result is a polynomial and not a rational function. This method is therefore used to obtain the solution of (21).

The residual associated with an approximate solution $(\mathbf{z}, \mathbf{p}, \mathbf{t})$ of (17) is

$$\mathbf{r}(\mathbf{z}, \mathbf{p}, \mathbf{t}) = (\mathbf{c} + \mathbf{t}) - \left(D^{-1}\left(T(f) + B(z)\right) Q\right) \mathbf{p}, \tag{23}$$

and thus

$$\begin{aligned}\tilde{\mathbf{r}} &:= \mathbf{r}(\mathbf{z} + \delta\mathbf{z}, \mathbf{p} + \delta\mathbf{p}, \mathbf{t} + \delta\mathbf{t}) \\ &= (\mathbf{c} + (\mathbf{t} + \delta\mathbf{t})) - \left(D^{-1}\left(T(f) + B(z + \delta z)\right) Q\right)(\mathbf{p} + \delta\mathbf{p}).\end{aligned}$$

It therefore follows that, to first order,

$$\tilde{\mathbf{r}} = \mathbf{r}(\mathbf{z}, \mathbf{p}, \mathbf{t}) + \delta\mathbf{t} - \left(D^{-1}\left(T + B\right) Q\right)\delta\mathbf{p} - \left(D^{-1}\left(\sum_{i=0}^{m} \frac{\partial B}{\partial z_i}\delta z_i\right) Q\right) \mathbf{p}, \tag{24}$$

where the last term on the right hand side represents the polynomial multiplication $\delta s(y)g(y)$, and $s(y)$ is defined in (18). The simplification of this term requires that the polynomial multiplication

$$g(y)s(y) = \left(\sum_{i=0}^{n} b_i \binom{n}{i} (1-y)^{n-i} y^i\right) \left(\sum_{i=0}^{m} z_i \binom{m}{i} (1-y)^{m-i} y^i\right),$$

which can also be expressed as

$$s(y)g(y) = \left(\sum_{i=0}^{m} z_i \binom{m}{i} (1-y)^{m-i} y^i\right) \left(\sum_{i=0}^{n} b_i \binom{n}{i} (1-y)^{n-i} y^i\right),$$

be considered. These polynomial multiplications can be expressed in matrix forms as, respectively,

$$\left(D^{-1}Y(p)R\right)\mathbf{z} \qquad \text{and} \qquad \left(D^{-1}B(z)Q\right)\mathbf{p}, \tag{25}$$

where z and $\mathbf{z}$ are defined in (22), $Y(p) \in \mathbb{R}^{(m+n+1)\times(m+1)}$ is a Tœplitz matrix,

$$p = \left\{ b_0,\, b_1,\, \cdots,\, b_n \right\} \qquad \text{and} \qquad R = \operatorname{diag}\left[\binom{m}{0} \; \binom{m}{1} \; \cdots \; \binom{m}{m} \right].$$

It therefore follows from (25) that

$$(YR)\,\mathbf{z} = (BQ)\,\mathbf{p},$$

and the differentiation of both sides of this equation with respect to $\mathbf{z}$ yields

$$(YR)\delta\mathbf{z} = \left(\sum_{i=0}^{m} \frac{\partial B}{\partial z_i} \delta z_i \right) Q\mathbf{p},$$

and thus (24) simplifies to

$$\tilde{\mathbf{r}} = \mathbf{r}(\mathbf{z}, \mathbf{p}, \mathbf{t}) + \delta\mathbf{t} - \Big(D^{-1} \left(T(f) + B(z) \right) Q \Big) \delta\mathbf{p} - (D^{-1} YR)\delta\mathbf{z}. \tag{26}$$

The jth iteration in the Newton-Raphson method for the calculation of $\mathbf{z}, \mathbf{p}$ and $\mathbf{t}$ is obtained from (26),

$$\left[H_z \; H_p \; H_t \right]^{(j)} \begin{bmatrix} \delta\mathbf{z} \\ \delta\mathbf{p} \\ \delta\mathbf{t} \end{bmatrix}^{(j)} = \mathbf{r}^{(j)}, \tag{27}$$

where $\mathbf{r}^{(j)} = \mathbf{r}^{(j)}(\mathbf{z}, \mathbf{p}, \mathbf{t})$,

$$\begin{aligned} H_z &= D^{-1} YR \in \mathbb{R}^{(m+n+1)\times(m+1)}, \\ H_p &= D^{-1} \left(T + B \right) Q \in \mathbb{R}^{(m+n+1)\times(n+1)}, \\ H_t &= -I \in \mathbb{R}^{(m+n+1)\times(m+n+1)}, \end{aligned}$$

and the values of $\mathbf{z}, \mathbf{p}$ and $\mathbf{t}$ at the jth iteration are

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{p} \\ \mathbf{t} \end{bmatrix}^{(j)} = \begin{bmatrix} \mathbf{z} \\ \mathbf{p} \\ \mathbf{t} \end{bmatrix}^{(j-1)} + \begin{bmatrix} \delta\mathbf{z} \\ \delta\mathbf{p} \\ \delta\mathbf{t} \end{bmatrix}^{(j)}, \qquad \begin{bmatrix} \mathbf{z} \\ \mathbf{p} \\ \mathbf{t} \end{bmatrix}^{(0)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{p}_0 \\ \mathbf{0} \end{bmatrix}.$$

The initial values of $\mathbf{z}$ and $\mathbf{t}$ are $\mathbf{z}^{(0)} = \mathbf{0}$ and $\mathbf{t}^{(0)} = \mathbf{0}$ because the given data is the inexact data, and the initial value $\mathbf{p}_0$ of $\mathbf{p}$ is calculated from (23),

$$\mathbf{p}_0 = \arg\min_{\mathbf{w}} \left\| \left(D^{-1} T(f) Q \right) \mathbf{w} - \mathbf{c} \right\|. \tag{28}$$

Equation (27) is of the form

$$C^{(j)} \delta\mathbf{y}^{(j)} = \mathbf{q}^{(j)}, \qquad \mathbf{q}^{(j)} = \mathbf{r}^{(j)} \in \mathbb{R}^{m+n+1}, \tag{29}$$

where $C^{(j)} \in \mathbb{R}^{(m+n+1)\times(2m+2n+3)}$, $\delta\mathbf{y}^{(j)} \in \mathbb{R}^{2m+2n+3}$, and

$$C^{(j)} = \begin{bmatrix} H_z & H_p & H_t \end{bmatrix}^{(j)}, \quad \delta\mathbf{y}^{(j)} = \begin{bmatrix} \delta\mathbf{z} \\ \delta\mathbf{p} \\ \delta\mathbf{t} \end{bmatrix}^{(j)}, \quad \mathbf{y}^{(j)} = \mathbf{y}^{(j-1)} + \delta\mathbf{y}^{(j)}. \tag{30}$$

It follows from (21) that, of all the solutions that satisfy the constraint equation, the solution that is nearest the LS solution defined by the given inexact data is required. The function to be minimised is therefore

$$\left\| \begin{bmatrix} \mathbf{z}^{(j)} - \mathbf{z}^{(0)} \\ \mathbf{p}^{(j)} - \mathbf{p}_0 \\ \mathbf{t}^{(j)} - \mathbf{t}^{(0)} \end{bmatrix} \right\| = \left\| \begin{bmatrix} \mathbf{z}^{(j-1)} + \delta\mathbf{z}^{(j)} \\ \mathbf{p}^{(j-1)} + \delta\mathbf{p}^{(j)} - \mathbf{p}_0 \\ \mathbf{t}^{(j-1)} + \delta\mathbf{t}^{(j)} \end{bmatrix} \right\| := \left\| \delta\mathbf{y}^{(j)} - \mathbf{w}^{(j-1)} \right\|, \tag{31}$$

where $\delta\mathbf{y}^{(j)}$ is defined in (30), $\delta\mathbf{y}^{(0)} = \mathbf{w}^{(0)} = \mathbf{0}$ and

$$\mathbf{w}^{(j-1)} = -\left(\mathbf{y}^{(j-1)} - \mathbf{y}^{(0)} \right) = - \begin{bmatrix} \mathbf{z}^{(j-1)} \\ \mathbf{p}^{(j-1)} - \mathbf{p}_0 \\ \mathbf{t}^{(j-1)} \end{bmatrix} \in \mathbb{R}^{2m+2n+3}. \tag{32}$$

The minimisation of (31) subject to (29) yields the LSE problem,

$$\min_{\delta\mathbf{y}^{(j)}} \left\| \delta\mathbf{y}^{(j)} - \mathbf{w}^{(j-1)} \right\| \qquad \text{subject to} \qquad C^{(j)} \delta\mathbf{y}^{(j)} = \mathbf{q}^{(j)}, \tag{33}$$

which can be solved, at each iteration, by the QR decomposition [5], where $C^{(j)}, \mathbf{q}^{(j)}$ and $\mathbf{w}^{(j-1)}$ are updated between successive iterations.

Algorithm 1 shows the application of the method of STLN to the deconvolution of two Bernstein polynomials.

Algorithm 1: Deconvolution of two Bernstein polynomials

Input Inexact polynomials f and h, which are of degrees m and $m+n$ respectively.

Output The polynomial $g = h/f$.

Begin

(1) Preprocess f and h as shown in Section 2.
(2) % Initialise the data
  (a) Set $\mathbf{z} = \mathbf{z}^{(0)} = \mathbf{0}$, which yields $B = B^{(0)} = 0$, and $\mathbf{t} = \mathbf{t}^{(0)} = \mathbf{0}$.
  (b) Calculate $T, Y^{(0)}$ and the initial value $\mathbf{p}_0$ of $\mathbf{p}$, which is defined in (28). Calculate $\mathbf{r}^{(0)}$, the initial value of the residual,

$$\mathbf{q}^{(0)} = \mathbf{r}^{(0)} = \mathbf{r}\left(\mathbf{z}^{(0)} = \mathbf{0}, \mathbf{p}_0, \mathbf{t}^{(0)} = \mathbf{0}\right) = \mathbf{c} - \left(D^{-1}T(f)Q\right)\mathbf{p}_0.$$

  (c) Define $C^{(0)}$ and set $\mathbf{w}^{(0)} = \mathbf{0}$.
(3) % Use the QR decomposition to solve the LSE problem at each iteration.

Set $j = 0$. % Initialise the iteration counter
repeat

(a) Set $j = j + 1$. % Increment the iteration counter
(b) Compute the QR decomposition of $C^{(j-1)^T}$,

$$C^{(j-1)^T} = (QR)^{(j-1)} = \left(Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix}\right)^{(j-1)}.$$

(c) Set $\mathbf{w}_1^{(j-1)} = \left(R_1^{-T}\mathbf{q}\right)^{(j-1)}$.
(d) Partition $Q^{(j-1)}$ as

$$Q^{(j-1)} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^{(j-1)},$$

where $Q_1^{(j-1)} \in \mathbb{R}^{(2m+2n+3)\times(m+n+1)}$ and $Q_2^{(j-1)} \in \mathbb{R}^{(2m+2n+3)\times(m+n+2)}$.
(e) Compute $\mathbf{z}_1^{(j-1)} = \left(Q_2^T\mathbf{w}\right)^{(j-1)}$ where $\mathbf{w}^{(j-1)}$ is defined in (32).
(f) Compute the solution

$$\delta\mathbf{y}^{(j)} = \left(Q \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{z}_1 \end{bmatrix}\right)^{(j-1)}.$$

(g) Set $\mathbf{z}^{(j)} = \mathbf{z}^{(j-1)} + \delta\mathbf{z}^{(j)}$, $\mathbf{p}^{(j)} = \mathbf{p}^{(j-1)} + \delta\mathbf{p}^{(j)}$ and $\mathbf{t}^{(j)} = \mathbf{t}^{(j-1)} + \delta\mathbf{t}^{(j)}$, and thus calculate $\mathbf{y}^{(j)}$ and $\mathbf{w}^{(j)}$.
(h) Update $B^{(j)}$ and $Y^{(j)}$, and therefore $C^{(j)}$, from $\mathbf{z}^{(j)}$ and $\mathbf{p}^{(j)}$. Compute the residual $\mathbf{q}^{(j)} = \mathbf{r}^{(j)}\left(\mathbf{z}^{(j)}, \mathbf{p}^{(j)}, \mathbf{t}^{(j)}\right)$ from (23),

$$\mathbf{r}^{(j)}\left(\mathbf{z}^{(j)}, \mathbf{p}^{(j)}, \mathbf{t}^{(j)}\right) = \left(\mathbf{c} + \mathbf{t}^{(j)}\right) - \left(D^{-1}\left(T + B^{(j)}\right)Q\right)\mathbf{p}^{(j)}.$$

until $\frac{\left\|\mathbf{r}^{(j)}\right\|}{\left\|\mathbf{c}+\mathbf{t}^{(j)})\right\|} \leq 10^{-12}$

End

Optimality conditions for the method of STLN, its formulation for the 1- and ∞-norms, and its relation to the Newton-Raphson iteration, are considered in [6]. The method can be extended to non-linear structures in the coefficient matrix and/or right hand side vector, which yields the method of structured non-linear total least norm [7]. This method has been used for the computation of a structured low rank approximation of the Sylvester resultant matrix [9].

5 Convergence analysis

This section considers the convergence of Algorithm 1 for the iterative solution of the LSE problem, which is defined in (21). It follows from steps (3c)-(3f) of this algorithm, and (32), that the jth iteration of this LSE problem yields

$$\begin{aligned}
\delta\mathbf{y}^{(j)} &= \left(Q\begin{bmatrix} R_1^{-T} \\ 0 \end{bmatrix}\mathbf{q} + Q\begin{bmatrix} 0 \\ \mathbf{z}_1 \end{bmatrix}\right)^{(j-1)} \\
&= \left(Q\begin{bmatrix} R_1^{-T} \\ 0 \end{bmatrix}\mathbf{q} + \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}\begin{bmatrix} 0 \\ Q_2^T \end{bmatrix}\mathbf{w}\right)^{(j-1)} \\
&= \left(Q_1 R_1^{-T}\mathbf{q}\right)^{(j-1)} + \left(Q_2 Q_2^T\right)^{(j-1)}\left(-\mathbf{y}^{(j-1)} + \mathbf{y}^{(0)}\right).
\end{aligned}$$

It follows from (30) that Algorithm 1 converges if

$$\begin{aligned}
\lim_{j\to\infty}\left\|\mathbf{y}^{(j)} - \mathbf{y}^{(j-1)}\right\| &= \lim_{j\to\infty}\left\|\delta\mathbf{y}^{(j)}\right\| \\
&= \lim_{j\to\infty}\left\|\left(Q_2 Q_2^T\right)^{(j-1)}\left(\mathbf{y}^{(0)} - \mathbf{y}^{(j-1)}\right) + \left(Q_1 R_1^{-T}\mathbf{q}\right)^{(j-1)}\right\| \\
&= 0,
\end{aligned}$$

and thus the convergence of (33) is dependent on the matrices and vectors at each iteration, and its *a priori* determination is therefore difficult. Computational experiments showed, however, that convergence is achieved in fewer than 5 iterations, even for high degree polynomials that are corrupted by noise and have multiple roots.

6 Examples

This section contains two examples that illustrate the application of the method of STLN to the computation of $g = h/f$, and the solution of each example is compared with the LS solution of (16). Also, it was stated in Section 2 that it is numerically advantageous to include the diagonal matrix Q of combinatorial factors in the coefficient matrix, and this is confirmed numerically by including the solutions obtained when (16) is written as

$$\Big(D^{-1}T(f)\Big)\mathbf{b} \approx \mathbf{c}, \qquad \mathbf{b} = Q\mathbf{p}. \tag{34}$$

Three error measures and two condition numbers were computed for each example:

The error r_1 between the theoretically exact solution, that is, the coefficients of $\hat{g}(y)$, and the solution of the LSE problem (33). If this solution defines the polynomial $g_1(y)$, then the forward error in the coefficients of $g_1(y)$ is

$$r_1 = \frac{\|\mathbf{g}_1 - \hat{\mathbf{g}}\|}{\|\hat{\mathbf{g}}\|}. \tag{35}$$

The error r_2 between the theoretically exact solution and the LS solution of (16). If this solution defines the polynomial $g_2(y)$, then the forward error in the coefficients of $g_2(y)$ is

$$r_2 = \frac{\|\mathbf{g}_2 - \hat{\mathbf{g}}\|}{\|\hat{\mathbf{g}}\|}. \tag{36}$$

The error r_3 in the satisfaction of the constraint equation in the LSE problem. The residual of the computed solution of the constraint is, from (23),

$$r_3 = \frac{\|\mathbf{r}(\mathbf{z}, \mathbf{p}, \mathbf{t})\|}{\|\mathbf{c} + \mathbf{t}\|}. \tag{37}$$

Condition number 1: The condition number $\kappa\left(C^{(j)}\right)$ of the coefficient matrix of the constraint equation in the LSE problem (33) at termination of the

iterative procedure, for Q included in the coefficient matrix, and Q included in the solution vector.

Condition number 2: The condition numbers $\kappa(D^{-1}T(f)Q)$ and $\kappa(D^{-1}T(f))$ of the LS problem, for Q included in the coefficient matrix and Q included in the solution vector, respectively.

Also, the number of iterations N required for the solution of the LSE problem was recorded.

Noise was added in the componentwise sense to the coefficients of $\hat{f}(y)$ and $\hat{h}(y)$, and it therefore follows from (13) and (14) that the coefficients of the polynomials $\delta\hat{f}(y)$ and $\delta\hat{h}(y)$ are, respectively,

$$\delta\hat{a}_i = \varepsilon r_i \hat{a}_i, \quad i = 0, \ldots, m; \qquad \delta\hat{c}_j = \varepsilon r_j \hat{c}_j, \quad j = 0, \ldots, m+n, \tag{38}$$

where r_i and r_j are uniformly distributed random variables in the interval $[-1, 1]$, and ε is the reciprocal of the upper bound of the componentwise signal-to-noise ratio. The normwise error models follow easily from these componentwise error models,

$$\|\delta\hat{\mathbf{a}}\| \leq \varepsilon \|\hat{\mathbf{a}}\| \qquad \text{and} \qquad \|\delta\hat{\mathbf{c}}\| \leq \varepsilon \|\hat{\mathbf{c}}\|, \tag{39}$$

and thus ε is also the upper bound of the reciprocal of the normwise signal-to-noise ratio.

Example 6.1 Noise with componentwise signal-to-noise ratio 10^8 was added to the coefficients of the Bernstein forms of the exact polynomials,

$$\hat{f}(y) = (y - 0.30)^5 (y - 0.70)^4 (y - 1.40)^5 (y - 7.00)^6,$$

and

$$\begin{aligned}\hat{h}(y) = {} & (y - 0.30)^8 (y - 0.70)^7 (y - 1.40)^8 (y - 7.00)^6 (y + 1.80)^3 \times \\ & (y + 0.90)^4,\end{aligned}$$

and these inexact polynomials were then normalised, thereby obtaining the polynomials $f(y)$ and $h(y)$, which are defined in (13) and (14). The results are shown in Table 1 and it is seen that $r_1 = r_2$, and thus the relative errors of the solutions in the LS and LSE problems are equal. The error r_3 is approximately equal to 10^{-16}, and it therefore follows from (37) and the discussion in Section 3 that the computed solution is a polynomial, and not a polynomial approximation of a rational function. This value of r_3 must be compared with the values of r_1 and r_2, which are about eight orders of magnitude larger, and

thus the difference between the solutions from the LS and LSE problems is clear.

Table 1 shows that the inclusion of Q in the coefficient matrix causes a reduction of about three orders of magnitude in $\kappa\left(C^{(j)}\right)$ with respect to its value when Q is included in the solution vector, as shown in (34), which is in accord with the results in [11,12]. The table also shows that, for Q included in the coefficient matrix, $\kappa\left(C^{(j)}\right)$ is about one order of magnitude larger than the condition number of $D^{-1}T(f)Q$, which is the coefficient matrix of (16). Also, one iteration is required for the convergence of the LSE problem when Q is included in the coefficient matrix, but four iterations are required for convergence when Q is included in the solution vector. □

	Q in coefficient matrix	Q in solution vector
r_1	3.20×10^{-8}	3.20×10^{-8}
r_2	3.20×10^{-8}	3.20×10^{-8}
r_3	1.41×10^{-16}	5.53×10^{-16}
$\kappa(C)$	$\kappa\left(C^{(1)}\right) = 6.92 \times 10^4$	$\kappa\left(C^{(4)}\right) = 3.45 \times 10^7$
	$\kappa(D^{-1}T(f)Q) = 7.49 \times 10^3$	$\kappa(D^{-1}T(f)) = 2.66 \times 10^7$
N	1	4

Table 1
The results of Example 6.1. The first column shows the results when (16) is solved, and the second column shows the results when (34) is solved.

Example 6.2 The procedure described in Example 6.1 was implemented for the polynomials,

$$\hat{f}(y) = (y - 0.30)^6(y - 0.40)^4(y - 0.50)^4(y - 0.60)^5(y - 0.70)^4,$$

and

$$\begin{aligned}\hat{h}(y) = &(y - 0.30)^8(y - 0.40)^6(y - 0.50)^6(y - 0.60)^6(y - 0.70)^6 \times \\ &(y - 0.80)^3(y - 0.90)^4(y - 0.99)^4.\end{aligned}$$

The results are shown in Table 2 and they are similar to the results in Table 1 because better results are obtained when Q is included in the coefficient matrix. It is also seen that

$$\frac{\kappa\left(C^{(1)}\right)}{\kappa(D^{-1}T(f)Q)} = \frac{7.19 \times 10^2}{5.83 \times 10^3} \approx 0.1,$$

	Q in coefficient matrix	Q in solution vector
r_1	2.80×10^{-6}	2.82×10^{-6}
r_2	2.82×10^{-6}	2.82×10^{-6}
r_3	1.27×10^{-15}	7.55×10^{-15}
$\kappa(C)$	$\kappa\left(C^{(1)}\right) = 7.19 \times 10^2$	$\kappa\left(C^{(54)}\right) = 1.35 \times 10^6$
	$\kappa(D^{-1}T(f)Q) = 5.83 \times 10^3$	$\kappa(D^{-1}T(f)) = 4.94 \times 10^8$
N	1	54

Table 2
The results of Example 6.2. The first column shows the results when (16) is solved, and the second column shows the results when (34) is solved.

and thus the condition number of the coefficient matrix of the constraint equation in the LSE problem (33) is about one order of magnitude smaller than the condition number of the coefficient matrix of (16). Also, only one iteration is required for the solution of the LSE problem when Q is included in the coefficient matrix, but 54 iterations are required when Q is included in the solution vector. □

7 Discussion

Tables 1 and 2 show that $\kappa\left(C^{(j)}\right)$, the condition number of the coefficient matrix of the constraint equation in the LSE problem, is significantly smaller when Q is included in the coefficient matrix than when it is included in the solution vector. This result, which confirms the remarks in Section 2, shows that large combinatorial factors in matrices associated with computations on Bernstein basis polynomials must be considered carefully, such that adverse numerical effects are minimised. Also, fewer iterations are required for the iterative solution of the LSE problem when Q is included in the coefficient matrix. The discussion in this section is therefore restricted to this situation, and the results obtained when Q is included in the solution vector are not considered.

Tables 1 and 2 show that the errors r_1 and r_2, which are defined in (35) and (36) respectively, are equal for Example 6.1, and for Example 6.2. It therefore follows that the relative errors in the solutions of the LS and LSE problems are the same, but the advantage of the solution of the LSE problem can be seen from the error r_3 in the tables. Since it is approximately equal to 10^{-16} for both examples, it follows that the solution of the LSE problem defines a polynomial, and not a polynomial approximation of a rational function.

A computed solution is acceptable if its backward error is less than or equal to the error in the given data, and it must therefore be checked that the solutions satisfy this condition. This calculation requires the data in Table 3, which shows the norms of the vectors of the coefficients of $f(y), h(y), s(y)$ and $e(y)$, which are defined in (13), (14), (18) and (19) respectively.

Example 6.1

$f(y)$	$\|\mathbf{a}\| = 7.16 \times 10^{4}$
$s(y)$	$\|\mathbf{z}\| = 1.18 \times 10^{-9}$
$h(y)$	$\|\mathbf{c}\| = 3.39 \times 10^{3}$
$e(y)$	$\|\mathbf{t}\| = 2.16 \times 10^{-7}$

Example 6.2

$f(y)$	$\|\mathbf{a}\| = 2.04 \times 10^{3}$
$s(y)$	$\|\mathbf{z}\| = 1.31 \times 10^{-5}$
$h(y)$	$\|\mathbf{c}\| = 7.27 \times 10^{3}$
$e(y)$	$\|\mathbf{t}\| = 1.85 \times 10^{-5}$

Table 3
The norms of the vectors of the coefficients of $f(y), s(y), h(y)$ and $e(y)$ for Examples 6.1 and 6.2.

It follows from Table 3 that

$$\frac{\|\mathbf{z}\|}{\|\mathbf{a}\|} = 1.65 \times 10^{-14} \qquad \text{and} \qquad \frac{\|\mathbf{t}\|}{\|\mathbf{c}\|} = 6.37 \times 10^{-11}, \tag{40}$$

for Example 6.1, and

$$\frac{\|\mathbf{z}\|}{\|\mathbf{a}\|} = 6.42 \times 10^{-9} \qquad \text{and} \qquad \frac{\|\mathbf{t}\|}{\|\mathbf{c}\|} = 2.54 \times 10^{-9}, \tag{41}$$

for Example 6.2, and these normwise errors must be checked against the normwise bounds (39) in order to verify that the computed solutions are acceptable. In particular, the random variables r_i and r_j in (38) are uniformly distributed in the interval $[-1, 1]$, and thus the random variables $|r_i|$ and $|r_j|$ are uniformly distributed in the interval $[0, 1]$ with mean value $1/2$. It therefore follows from (38) that

$$\|\delta\hat{\mathbf{a}}\| \approx \frac{\varepsilon}{2}\|\hat{\mathbf{a}}\| \qquad \text{and} \qquad \|\delta\hat{\mathbf{c}}\| \approx \frac{\varepsilon}{2}\|\hat{\mathbf{c}}\|. \tag{42}$$

The normwise backward errors of the solutions also require that the structured perturbations z_i and unstructured perturbations t_i be considered. In particular, it follows from (13), (14), (15) and (20) that the corrected forms of the exact polynomials $\hat{f}(y)$ and $\hat{h}(y)$ are, respectively,

$$f(y) + s(y) = \sum_{i=0}^{m} (\hat{a}_i + \delta\hat{a}_i + z_i) \binom{m}{i} (1-y)^{m-i} y^i,$$

and

$$h(y) + e(y) = \sum_{i=0}^{m+n} (\hat{c}_i + \delta\hat{c}_i + t_i) \binom{m+n}{i} (1-y)^{m+n-i} y^i,$$

and thus the squares of the backward errors of the coefficients of $f(y)$ and $h(y)$ are, respectively,

$$\sum_{i=0}^{m} (\delta\hat{a}_i + z_i)^2 = \|\delta\hat{\mathbf{a}}\|^2 + \|\mathbf{z}\|^2 + 2\delta\hat{\mathbf{a}}^T\mathbf{z},$$

and

$$\sum_{i=0}^{m+n} (\delta\hat{c}_i + t_i)^2 = \|\delta\hat{\mathbf{c}}\|^2 + \|\mathbf{t}\|^2 + 2\delta\hat{\mathbf{c}}^T\mathbf{t}.$$

It follows from (38) that the average values of $\delta\hat{a}_i$ and $\delta\hat{c}_j$ are zero, and thus (42) shows that $\eta(f)$ and $\eta(h)$, the average values of the squares of the backward errors of the coefficients of $f(y)$ and $h(y)$ satisfy

$$\eta(f)^2 \approx \left(\frac{\varepsilon}{2}\right)^2 \|\hat{\mathbf{a}}\|^2 + \|\mathbf{z}\|^2 \qquad \text{and} \qquad \eta(h)^2 \approx \left(\frac{\varepsilon}{2}\right)^2 \|\hat{\mathbf{c}}\|^2 + \|\mathbf{t}\|^2.$$

Since $\varepsilon = 10^{-8}$ in Examples 6.1 and 6.2, the errors in the equations $\|\mathbf{a}\| = \|\hat{\mathbf{a}}\|$ and $\|\mathbf{c}\| = \|\hat{\mathbf{c}}\|$ are negligible, and thus $\eta(f)^2$ and $\eta(h)^2$ can be approximated by

$$\eta(f)^2 \approx \left(\frac{\varepsilon}{2}\right)^2 \|\mathbf{a}\|^2 + \|\mathbf{z}\|^2 \qquad \text{and} \qquad \eta(h)^2 \approx \left(\frac{\varepsilon}{2}\right)^2 \|\mathbf{c}\|^2 + \|\mathbf{t}\|^2.$$

The substitution of (40) and (41) into these approximations yields

$$\eta(f) \approx \frac{\varepsilon}{2}\|\mathbf{a}\| \approx \frac{\varepsilon}{2}\|\hat{\mathbf{a}}\| \qquad \text{and} \qquad \eta(h) \approx \frac{\varepsilon}{2}\|\mathbf{c}\| \approx \frac{\varepsilon}{2}\|\hat{\mathbf{c}}\|,$$

for Example 6.1, and

$$\eta(f) \approx \varepsilon\|\mathbf{a}\| \approx \varepsilon\|\hat{\mathbf{a}}\| \qquad \text{and} \qquad \eta(h) \approx \varepsilon\|\mathbf{c}\| \approx \varepsilon\|\hat{\mathbf{c}}\|,$$

for Example 6.2, and thus the average values of the normalised backward errors are bounded, approximately, by $\varepsilon/2$ and ε, and they are therefore acceptable.

The condition number $\kappa\left(C^{(j)}\right)$ is not the condition number of the jth iteration in the LSE problem because it does not consider the constraint equation. The inverse of the upper triangular matrix $R_1^{(j)}$ is computed in step (3c) of Algorithm 1, and since

$$\kappa\left(C^{(j)}\right) = \kappa\left(R^{(j)}\right) = \kappa\left(R_1^{(j)}\right),$$

it is desirable to minimise $\kappa\left(C^{(j)}\right)$. This minimisation is achieved by including Q in the coefficient matrix, rather than in the solution vector.

The method used in Algorithm 1 is analysed in [1], where it is shown that it is numerically stable. The sensitivity of the solution of the LSE problem to perturbations in $C^{(j)}, \mathbf{q}^{(j)}$ and $\mathbf{w}^{(j-1)}$ is also considered in [1], and it is shown that the forward error r_1 is of the form $r_1 \lesssim \varepsilon\kappa_{\text{LSE}}$, where κ_{LSE} is the condition number of the LSE problem. Computational results in [1] show that an approximation to this bound may overestimate the forward error achieved in examples, and that this is due to the use of worst case bounds, which cannot be improved.

8 Summary

This paper has considered the method of STLN for the deconvolution of two Bernstein basis polynomials. This method preserves the Tœplitz structure of the coefficient matrix of the equation that defines the deconvolution of two polynomials, and it therefore returns a polynomial. This solution was compared with the solution of a LS problem, and it was shown that this solution defines a polynomial approximation of a rational function, and not a polynomial. It was shown that the method of STLN yields the LSE problem that is solved iteratively by the QR decomposition.

Improved answers were obtained when the coefficient matrix of the equation that defines the deconvolution operation includes the diagonal matrix Q of combinatorial factors. This improvement manifests itself in the requirement for fewer iterations for the solution of the LSE problem, and a smaller condition number of the coefficient matrix of the constraint equation in the LSE problem.

References

[1] A. Cox and N. J. Higham. Accuracy and stability analysis of the null space method for solving the equality constrained least squares problem. *BIT*,

39(1):34–50, 1999.

[2] R. T. Farouki and T. N. T. Goodman. On the optimal stability of the Bernstein basis. *Mathematics of Computation*, 65(216):1553–1566, 1996.

[3] R. T. Farouki and V. T. Rajan. Algorithms for polynomials in Bernstein form. *Computer Aided Geometric Design*, 5:1–26, 1988.

[4] R. Goldman. *Pyramid Algorithms: A Dynamic Programming Approach to Curves and Surfaces for Geometric Modeling*. Morgan Kaufmann Publishers, Academic Press, San Diego USA, 2002.

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, USA, 1996.

[6] J. Ben Rosen, H. Park, and J. Glick. Total least norm formulation and solution for structured problems. *SIAM J. Mat. Anal. Appl.*, 17(1):110–128, 1996.

[7] 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.

[8] J. V. Uspensky. *Theory of Equations*. McGraw-Hill, New York, USA, 1948.

[9] 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.

[10] 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.

[11] J. R. Winkler and N. Yang. Methods for the computation of the degree of an approximate greatest common divisor of two inexact Bernstein basis polynomials, 2013. Submitted.

[12] 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.