



# On the accurate computation of the Newton form of the Lagrange interpolant

Y. Khia<sup>1</sup> · E. Mainar<sup>1</sup> · E. Royo-Amondarain<sup>2</sup> · B. Rubio<sup>1</sup>

Received: 2 October 2023 / Accepted: 24 April 2024

© The Author(s) 2024

## Abstract

In recent years many efforts have been devoted to finding bidiagonal factorizations of nonsingular totally positive matrices, since their accurate computation allows to numerically solve several important algebraic problems with great precision, even for large ill-conditioned matrices. In this framework, the present work provides the factorization of the collocation matrices of Newton bases—of relevance when considering the Lagrange interpolation problem—together with an algorithm that allows to numerically compute it to high relative accuracy. This further allows to determine the coefficients of the interpolating polynomial and to compute the singular values and the inverse of the collocation matrix. Conditions that guarantee high relative accuracy for these methods and, in the former case, for the classical recursion formula of divided differences, are determined. Numerical errors due to imprecise computer arithmetic or perturbed input data in the computation of the factorization are analyzed. Finally, numerical experiments illustrate the accuracy and effectiveness of the proposed methods with several algebraic problems, in stark contrast with traditional approaches.

**Keywords** High relative accuracy · Bidiagonal decompositions · Totally positive matrices · Lagrange interpolation · Newton basis · Divided differences

**Mathematics Subject Classification (2010)** 65F15 · 65F05 · 15A23 · 15A18 · 15A06

---

✉ E. Royo-Amondarain  
eduroyo@unizar.es

Y. Khia  
ykhia@unizar.es

E. Mainar  
esmemain@unizar.es

B. Rubio  
brubio@unizar.es

<sup>1</sup> Departamento de Matemática Aplicada/IUMA, Universidad de Zaragoza, Zaragoza, Spain

<sup>2</sup> Departamento de Matemáticas/CAPA, Universidad de Zaragoza, Zaragoza, Spain

## 1 Introduction

A classical approach to the Lagrange interpolation problem is the Newton form, in which the polynomial interpolant is written in terms of the Newton basis. Its coefficients, called divided differences, can be determined both by means of a recursive formula or by solving a linear system that involves the collocation matrix of the basis. The recursive computation of divided differences requires many subtractions which can lead to cancellation: an error-inducing phenomenon that takes place when two nearly equal numbers are subtracted and that usually elevates the effect of earlier errors in the computations. In fact, this is also the case of the notoriously ill-conditioned collocation matrices that are associated with the Lagrange interpolation problem—these errors can heavily escalate when the number of nodes increases, eventually turning unfeasible the attainment of a solution in any algebraic problem involving these matrices. It is worth noting that sometimes the errors in computing the divided differences can be ameliorated by considering different node orderings [9, 31], although this strategy may not be sufficient to obtain the required level of precision in a given high-order interpolation problem.

However, in some scenarios it is possible to keep numerical errors under control. In a given floating-point arithmetic, a real value is said to be determined to high relative accuracy (HRA) whenever the relative error of the computed value is bounded by the product of the unit round-off and a positive constant, independent of the arithmetic precision. HRA implies great accuracy in the computations since the relative errors have the same order as the machine precision and the accuracy is not affected by the dimension or the conditioning of the problem to be solved. A sufficient condition to assure that an algorithm can be computed to HRA is the non inaccurate cancellation condition. Sometimes denoted as NIC condition, it is satisfied if the algorithm does not require inaccurate subtractions and only evaluates products, quotients, sums of numbers of the same sign, subtractions of numbers of opposite sign or subtraction of initial data (cf. [6, 14]). In this sense, in this paper we provide the precise conditions under which the recursive computation of divided differences can be performed to HRA—these include strictly ordered nodes and a particular sign structure of the values of the interpolated function at these nodes.

In the research of algorithms that preserve HRA, a major step forward was given in the seminal work of Gasca and Peña for nonsingular totally positive matrices, since these can be written as a product of bidiagonal matrices [10–12]. This factorization can be seen as a representation of the matrices exploiting their total positivity property to achieve accurate numerical linear algebra since, if it is provided to HRA, it allows solving several algebraic problems involving the collocation matrix of a given basis to HRA. In the last years, the search for bidiagonal decompositions of different totally positive bases has been a very active field of research [4–7, 18–20, 28].

The present work can be framed as a contribution to the above picture for the particular case of the well-known Newton basis and its collocation matrices. The conditions guaranteeing their total positivity are derived and a fast algorithm to obtain their bidiagonal factorization to HRA is provided. Under these conditions, the obtained bidiagonal decomposition can be applied to compute to HRA the singular values, the inverse, and also the solution of some linear systems of equations. As will be illustrated,



This representation will allow us to define algorithms adapted to the totally positive structure, providing accurate computations with  $A$ .

If the bidiagonal factorization (1) of a nonsingular and totally positive matrix  $A$  can be computed to high relative accuracy, the computation of its eigenvalues and singular values, the computation of  $A^{-1}$  and even the resolution of systems of linear equations  $Ax = b$ , for vectors  $b$  with alternating signs, can be also computed to high relative accuracy using the algorithms provided in [15].

Let  $(u_0, \dots, u_n)$  be a basis of a space  $U$  of functions defined on  $I \subseteq \mathbb{R}$ . Given a sequence of parameters  $t_1 < \dots < t_{n+1}$  on  $I$ , the corresponding collocation matrix is defined by

$$M(t_1, \dots, t_{n+1}) := (u_{j-1}(t_i))_{1 \leq i, j \leq n+1}. \tag{4}$$

Let  $\mathbf{P}^n(I)$  be the  $(n + 1)$ -dimensional space formed by all polynomials of degree not greater than  $n$ , in a variable defined on  $I \subseteq \mathbb{R}$ , that is,

$$\mathbf{P}^n(I) := \text{span}\{1, t, \dots, t^n\}, \quad t \in I.$$

Given nodes  $t_1, \dots, t_{n+1}$  on  $I$  and a function  $f : I \rightarrow \mathbb{R}$ , we are going to address the Lagrange interpolation problem for finding  $p_n \in \mathbf{P}^n(\mathbb{R})$  such that

$$p_n(t_i) = f(t_i), \quad i = 1, \dots, n + 1.$$

When considering the monomial basis  $(m_0, \dots, m_n)$ , with  $m_i(t) = t^i$ , for  $i = 0, \dots, n$ , the interpolant can be written as follows,

$$p_n(t) = \sum_{i=0}^n c_{i+1} m_i(t), \tag{5}$$

and the coefficients  $c_i, i = 1, \dots, n + 1$ , form the solution vector  $c = (c_1, \dots, c_{n+1})^T$  of the linear system

$$Vc = f,$$

where  $f := (f(t_1), \dots, f(t_{n+1}))^T$  and  $V \in \mathbb{R}^{(n+1) \times (n+1)}$  is the collocation matrix of the monomial basis at the nodes  $t_i, i = 1, \dots, n + 1$ , that is,

$$V := (t_i^{j-1})_{1 \leq i, j \leq n+1}. \tag{6}$$

Let us observe that  $V$  is the Vandermonde matrix at the considered nodes and recall that Vandermonde matrices have relevant applications in Lagrange interpolation and numerical quadrature (see for example [8] and [29]). In fact, for any increasing sequence of positive values,  $0 < t_1 < \dots < t_{n+1}$ , the corresponding Vandermonde matrix  $V$  in (6) is known to be strictly totally positive (see Section 3 of [14]) and so, we can write

$$c = V^{-1} f. \tag{7}$$

In [14] or Theorem 3 of [17] it is shown that the Vandermonde matrix  $V$  admits a bidiagonal factorization of the form (1):

$$V = F_n F_{n-1} \cdots F_1 D G_1 G_2 \cdots G_n, \tag{8}$$

where  $F_i \in \mathbb{R}^{(n+1) \times (n+1)}$  and  $G_i \in \mathbb{R}^{(n+1) \times (n+1)}$ ,  $i = 1, \dots, n$ , are the lower and upper triangular bidiagonal matrices described by (2) with

$$m_{i,j} = \prod_{k=1}^{j-1} \frac{t_i - t_{i-k}}{t_{i-1} - t_{i-k-1}}, \quad \tilde{m}_{i,j} = t_j, \quad 1 \leq j < i \leq n + 1, \tag{9}$$

and  $D$  is the diagonal matrix whose entries are

$$d_{i,i} = \prod_{k=1}^{i-1} (t_i - t_k), \quad i = 1, \dots, n + 1. \tag{10}$$

Using Koev’s notation, this factorization of  $V$  can be represented through the matrix  $BD(V) \in \mathbb{R}^{(n+1) \times (n+1)}$  with

$$BD(V)_{i,j} := \begin{cases} t_i, & \text{if } i < j, \\ \prod_{k=1}^{i-1} (t_i - t_k), & \text{if } i = j, \\ \prod_{k=1}^{j-1} \frac{t_i - t_{i-k}}{t_{i-1} - t_{i-k-1}}, & \text{if } i > j. \end{cases} \tag{11}$$

Moreover, it can be easily checked that the computation of  $BD(V)$  does not require inaccurate cancellations and can be performed to high relative accuracy.

When considering interpolation nodes  $t_1, \dots, t_{n+1}$  such that  $t_i \neq t_j$  for  $i \neq j$ , and the corresponding Lagrange polynomial basis  $(\ell_0, \dots, \ell_n)$ , with

$$\ell_i(t) := \prod_{j \neq i} \frac{t - t_{j+1}}{t_{i+1} - t_{j+1}}, \quad i = 0, \dots, n, \tag{12}$$

the Lagrange formula of the polynomial interpolant  $p_n$  is

$$p_n(t) = \sum_{i=0}^n f(t_{i+1}) \ell_i(t). \tag{13}$$

Taking into account (13), (5) and (7), we have

$$p_n(t) = (\ell_0(t), \dots, \ell_n(t)) f = (m_0(t), \dots, m_n(t)) V^{-1} f,$$

with  $f = (f(t_1), \dots, f(t_{n+1}))^T$ , and deduce that

$$(m_0, \dots, m_n) = (\ell_0, \dots, \ell_n) V. \tag{14}$$

Let us note that identity (14) means that the Vandermonde matrix  $V \in \mathbb{R}^{(n+1) \times (n+1)}$  is the change of basis matrix between the  $(n + 1)$ -dimensional monomial and Lagrange basis of the polynomial space  $\mathbf{P}^n(\mathbb{R})$  corresponding to the considered interpolation nodes.

The Lagrange’s interpolation formula (13) is usually considered for small numbers of interpolation nodes, due to certain shortcomings claimed such as:

- The evaluation of the interpolant  $p_n(t)$  requires  $O(n^2)$  flops.
- A new computation from scratch is needed when adding a new interpolation data pair.
- The computations are numerically unstable.

Nevertheless, as explained in [2], the Lagrange formula can be improved by using the first form of the barycentric interpolation formula, as called in [30], given by

$$p_n(t) = \sum_{i=0}^n f(t_{i+1})\ell_i(t) = \ell(t) \sum_{i=0}^n f(t_{i+1}) \frac{\omega_{i+1}}{t - t_{i+1}}, \tag{15}$$

where

$$\ell(t) := \prod_{i=0}^n (t - t_{i+1}), \quad \omega_i := 1 / \prod_{k \neq i} (t_i - t_k) = 1 / \ell'(t_i), \quad i = 1, \dots, n + 1.$$

Furthermore, since  $\sum_{i=0}^n \ell_i(t) = 1$ , for all  $t$ , the second form of the barycentric interpolation formula (see [30]) can be derived for the Lagrange interpolant

$$p_n(t) = \frac{\sum_{i=0}^n f(t_{i+1}) \frac{\omega_{i+1}}{t - t_{i+1}}}{\sum_{i=0}^n \frac{\omega_{i+1}}{t - t_{i+1}}}. \tag{16}$$

Let us observe that any common factor in the values  $\omega_j$ ,  $i = 1, \dots, n + 1$ , can be cancelled without changing the value of the interpolant  $p_n(t)$ . This property is used in [2] to derive interesting properties of the barycentric formula (16) for the interpolant.

### 3 Accurate computation of divided differences

Instead of Lagrange’s formulae (15) and (16), one can use the Newton form of the interpolant, which is obtained when the interpolant is written in terms of the Newton basis  $(w_0, \dots, w_n)$  determined by the interpolation nodes  $t_1, \dots, t_{n+1}$ ,

$$w_0(t) := 1, \quad w_i(t) := \prod_{k=1}^i t - t_k, \quad i = 1, \dots, n, \tag{17}$$

as follows,

$$p_n(t) = \sum_{i=0}^n [t_1, \dots, t_{i+1}]f w_i(t), \tag{18}$$

where  $[t_1, \dots, t_i]f$  denotes the divided difference of the interpolated function  $f$  at the nodes  $t_1, \dots, t_i$ . If  $f$  is  $n$ -times continuously differentiable on  $[t_1, t_{n+1}]$ , the divided differences  $[t_1, \dots, t_i]f, i = 1, \dots, n + 1$ , can be obtained using the following recursion

$$[t_i, \dots, t_{i+k}]f = \begin{cases} \frac{[t_{i+1}, \dots, t_{i+k}]f - [t_i, \dots, t_{i+k-1}]f}{t_{i+k} - t_i}, & \text{if } t_{i+k} \neq t_i, \\ \frac{f^{(k)}(t_i)}{k!}, & \text{if } t_{i+k} = t_i. \end{cases} \tag{19}$$

Note that the divided differences only depend on the interpolation nodes and, once computed, the interpolant (18) can be evaluated in  $O(n)$  flops per evaluation.

The following result shows under what conditions the divided differences can be computed to high relative accuracy using the recursion (19).

**Theorem 1** *Let  $t_1, \dots, t_{n+1}$  be strictly ordered nodes and  $f$  a function such that the entries of the vector  $(f(t_1), \dots, f(t_{n+1}))$  can be computed to high relative accuracy and have alternating signs. Then, the divided differences of order  $k = 0, \dots, n$*

$$[t_i, \dots, t_{i+k}]f, \quad i = 1, \dots, n + 1 - k,$$

*have alternating signs and, using recurrence (19), can be computed to high relative accuracy.*

**Proof** Let us prove the result by induction on the order  $k$  of the computed divided differences. First, for  $k = 0$ , we have simply

$$[t_i]f = f(t_i), \quad i = 1, \dots, n + 1, \tag{20}$$

and so, the zero-th order divided differences are trivially known to high relative accuracy and have the desired alternating sign pattern. Now, let us assume that  $[t_i, \dots, t_{i+k}]f$  is known to high relative accuracy for  $i = 1, \dots, n + 1 - k$  and that its sign alternates with  $i$ . Using (19), the  $(k + 1)$  order divided difference is computed as follows:

$$[t_i, \dots, t_{i+k+1}]f = \frac{[t_{i+1}, \dots, t_{i+k+1}]f - [t_i, \dots, t_{i+k}]f}{t_{i+k+1} - t_i}, \quad i = 1, \dots, n + 1 - k. \tag{21}$$

By the inductive hypothesis, the two divided differences in the right hand side of (21) have opposed signs and are known to high relative accuracy. Consequently, no inaccurate cancellations occur when applying recurrence (19)—the subtractions in the denominator being only of initial data—and we can derive that  $[t_i, \dots, t_{i+k+1}]f$  is also computed to high relative accuracy.

Now, regarding the alternating sign pattern, we can compare (21) with the next divided difference  $[t_{i+1}, \dots, t_{i+k+2}]f$ , which can be written as

$$[t_{i+1}, \dots, t_{i+k+2}]f = -\frac{[t_{i+1}, \dots, t_{i+k+1}]f - [t_{i+2}, \dots, t_{i+k+2}]f}{t_{i+k+2} - t_{i+1}}, \quad i = 0, \dots, n - k,$$

Since, by the inductive hypothesis, the  $k$ -th order divided differences  $[t_i, \dots, t_{i+k}]f$  and  $[t_{i+2}, \dots, t_{i+k+2}]f$  have the same sign, it readily follows that  $[t_i, \dots, t_{i+k+1}]f$  and  $[t_{i+1}, \dots, t_{i+k+2}]f$  have opposite signs for  $i = 1, \dots, n - k$ , provided that the nodes  $t_i$  are strictly ordered.  $\square$

Section 6 will illustrate the accuracy obtained using the recursive computation of divided differences and compare the results with those obtained through an alternative method proposed in the next section.

### 4 Total positivity and bidiagonal factorization of collocation matrices of Newton bases

Let us observe that, since the polynomial  $m_i(t) = t^i, i = 0, \dots, n$ , coincides with its interpolant at  $t_1, \dots, t_{n+1}$ , taking into account the Newton formula (18) for the monomials  $m_i, i = 0, \dots, n$ , we deduce that

$$(m_0, \dots, m_n) = (w_0, \dots, w_n)U, \tag{22}$$

where the change of basis matrix  $U = (u_{i,j})_{1 \leq i, j \leq n+1}$  is upper triangular and satisfies  $u_{i,j} = [t_1, \dots, t_i]m_{j-1}$ , that is,

$$U = \begin{pmatrix} 1 & [t_1]m_1 & [t_1]m_2 & \cdots & [t_1]m_n \\ 0 & 1 & [t_1, t_2]m_2 & \cdots & [t_1, t_2]m_n \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & [t_1, \dots, t_n]m_n \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}. \tag{23}$$

On the other hand, the collocation matrix of the Newton basis  $(w_0, \dots, w_n)$  (17) at the interpolation nodes  $t_1, \dots, t_{n+1}$  is a lower triangular matrix  $L = (l_{i,j})_{1 \leq i, j \leq n+1}$  whose entries are

$$l_{i,j} = w_{j-1}(t_i) = \prod_{k=1}^{j-1} (t_i - t_k), \tag{24}$$

that is,

$$L = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & t_2 - t_1 & 0 & \cdots & 0 \\ 1 & t_3 - t_1 & (t_3 - t_1)(t_3 - t_2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & t_{n+1} - t_1 & (t_{n+1} - t_1)(t_{n+1} - t_2) \cdots (t_{n+1} - t_{n-1}) & \cdots & (t_{n+1} - t_n) \end{pmatrix}.$$

Taking into account (14) and (22), we obtain the following Crout factorization of Vandermonde matrices at nodes  $t_i, i = 1, \dots, n + 1$ , with  $t_i \neq t_j$  for  $i \neq j$ ,

$$V = M \begin{bmatrix} \ell_0, \dots, \ell_n \\ t_1, \dots, t_{n+1} \end{bmatrix} V = M \begin{bmatrix} m_0, \dots, m_n \\ t_1, \dots, t_{n+1} \end{bmatrix} = M \begin{bmatrix} w_0, \dots, w_n \\ t_1, \dots, t_{n+1} \end{bmatrix} U = LU, \tag{25}$$

where  $L$  is the lower triangular collocation matrix of the Newton basis  $(w_0, \dots, w_n)$  and  $U$  is the upper triangular change of basis matrix satisfying (22).

The following result deduces the bidiagonal factorization of the collocation matrix  $L$  of the Newton basis.

**Theorem 2** *Given interpolation nodes  $t_1, \dots, t_{n+1}$ , with  $t_i \neq t_j$  for  $i \neq j$ , let  $L \in \mathbb{R}^{(n+1) \times (n+1)}$  be the collocation matrix described by (24) of the Newton basis (17). Then,*

$$L = F_n \cdots F_1 D, \tag{26}$$

where  $F_i \in \mathbb{R}^{(n+1) \times (n+1)}, i = 1, \dots, n$ , are lower triangular bidiagonal matrices whose structure is described by (2) and their off-diagonal entries are

$$m_{i,j} = \prod_{k=1}^{j-1} \frac{t_i - t_{i-k}}{t_{i-1} - t_{i-k-1}}, \quad 1 \leq j < i \leq n + 1, \tag{27}$$

and  $D \in \mathbb{R}^{(n+1) \times (n+1)}$  is the diagonal matrix  $D = \text{diag}(d_{1,1}, \dots, d_{n+1,n+1})$  with

$$d_{i,i} = \prod_{k=1}^{i-1} (t_i - t_k), \quad 1 \leq i \leq n + 1. \tag{28}$$

**Proof** From identities (8), (11) and (25), we have

$$V = M \begin{bmatrix} m_0, \dots, m_n \\ t_1, \dots, t_{n+1} \end{bmatrix} = F_n F_{n-1} \cdots F_1 D G_1 G_2 \cdots G_n,$$

where  $F_i \in \mathbb{R}^{(n+1) \times (n+1)}$  and  $G_i \in \mathbb{R}^{(n+1) \times (n+1)}, i = 1, \dots, n$ , are the lower and upper triangular bidiagonal matrices described by (2) with

$$m_{i,j} = \prod_{k=1}^{j-1} \frac{t_i - t_{i-k}}{t_{i-1} - t_{i-k-1}}, \quad \tilde{m}_{i,j} = t_j, \quad 1 \leq j < i \leq n + 1,$$

and  $D$  is the diagonal matrix whose entries are  $d_{i,i} = \prod_{k=1}^i (t_i - t_k)$  for  $i = 1, \dots, n+1$ .

From Theorem 3 of [21], the upper triangular matrix  $U$  in (23) satisfies

$$U = G_1 \cdots G_n, \tag{29}$$

where  $G_i \in \mathbb{R}^{(n+1) \times (n+1)}$ ,  $i = 1, \dots, n$ , are upper triangular bidiagonal matrices whose structure is described by (2) and their off-diagonal entries are

$$\tilde{m}_{i,j} = t_j, \quad 1 \leq j < i \leq n+1.$$

On the other hand, by (25), we also have,

$$V = M \begin{bmatrix} w_0, \dots, w_n \\ t_1, \dots, t_{n+1} \end{bmatrix} U = M \begin{bmatrix} w_0, \dots, w_n \\ t_1, \dots, t_{n+1} \end{bmatrix} G_1 G_2 \cdots G_n,$$

and conclude

$$L = M \begin{bmatrix} w_0, \dots, w_n \\ t_1, \dots, t_{n+1} \end{bmatrix} = F_n F_{n-1} \cdots F_1 D.$$

□

Taking into account Theorem 2, the bidiagonal factorization of the matrix  $L$  can be stored by the matrix  $BD(L)$  with

$$BD(L)_{i,j} := \begin{cases} 0, & \text{if } i < j, \\ \prod_{k=1}^{i-1} (t_i - t_k), & \text{if } i = j, \\ \prod_{k=1}^{j-1} \frac{t_i - t_{i-k}}{t_{i-1} - t_{i-k-1}}, & \text{if } i > j. \end{cases} \tag{30}$$

The analysis of the sign of the entries in (30) will allow us to characterize the total positivity property of the collocation matrix of Newton bases in terms of the ordering of the nodes. This fact is stated in the following result.

**Corollary 1** *Given interpolation nodes  $t_1, \dots, t_{n+1}$ , with  $t_i \neq t_j$  for  $i \neq j$ , let  $L \in \mathbb{R}^{(n+1) \times (n+1)}$  be the collocation matrix (24) of the Newton basis (17) and  $J$  the diagonal matrix  $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$ .*

- a) *The matrix  $L$  is totally positive if and only if  $t_1 < \dots < t_{n+1}$ . Moreover, in this case,  $L$  and the matrix  $BD(L)$  in (30) can be computed to HRA.*
- b) *The matrix  $L_J := LJ$  is totally positive if and only if  $t_1 > \dots > t_{n+1}$ . Moreover, in this case, the matrix  $BD(L_J)$  can be computed to HRA.*

*Furthermore, the singular values and the inverse matrix of  $L$ , as well as the solution of linear systems  $Ld = f$ , where the entries of  $f = (f_1, \dots, f_{n+1})^T$  have alternating signs, can be performed to HRA.*

**Proof** Let  $L = F_n \cdots F_1 D$  be the bidiagonal factorization provided by Theorem 2.

a) If  $t_1 < \cdots < t_{n+1}$ , the entries  $m_{i,j}$  in (27) and  $d_{i,i}$  in (28) are all positive and we conclude that the diagonal matrix  $D$  and the bidiagonal matrix factors  $F_i, i = 1, \dots, n$ , are totally positive. Taking into account that the product of totally positive matrices is a totally positive matrix (see Theorem 3.1 of [1]), we can guarantee that  $L$  is totally positive. Conversely, if  $L$  is totally positive then the entries  $m_{i,j}$  in (27) and  $d_{i,i}$  in (28) take all positive values. Moreover, since

$$t_2 - t_1 = d_{2,2} > 0, \quad t_i - t_{i-1} = m_{i,2}(t_{i-1} - t_{i-2}), \quad i = 3, \dots, n + 1,$$

we derive by induction that  $t_i - t_{i-1} > 0$  for  $i = 2, \dots, n + 1$ .

On the other hand, for increasing sequences of nodes, the subtractions in the computation of the entries  $m_{i,j}$  and  $p_{i,i}$  involve only initial data and so, will not lead to subtractive cancellations. So, the computation to high relative accuracy of the above mentioned algebraic problems can be performed to high relative accuracy using the matrix representation (30) and the Matlab commands in Koev’s web page (see Section 3 of [6]).

b) Now, using (26) and defining  $\tilde{D} := DJ$ , we can write

$$L_J = F_n \cdots F_1 \tilde{D}, \tag{31}$$

where  $F_i \in \mathbb{R}^{(n+1) \times (n+1)}, i = 1, \dots, n$ , are the lower triangular bidiagonal matrices described in (2), whose off-diagonal entries are given in (27), and  $\tilde{D} \in \mathbb{R}^{(n+1) \times (n+1)}$  is the diagonal matrix  $\tilde{D} = \text{diag}(\tilde{d}_{1,1}, \dots, \tilde{d}_{n+1})$  with

$$\tilde{d}_{i,i} = (-1)^{i-1} \prod_{k=1}^{i-1} (t_i - t_k), \quad 1 \leq i \leq n + 1. \tag{32}$$

It is worth noting that, according to (31), the bidiagonal decomposition of  $L_J$  is given by

$$BD(L_J)_{i,j} := \begin{cases} 0, & \text{if } i < j, \\ (-1)^{i-1} \prod_{k=1}^{i-1} (t_i - t_k), & \text{if } i = j, \\ \prod_{k=1}^{j-1} \frac{t_i - t_{i-k}}{t_{i-1} - t_{i-k-1}}, & \text{if } i > j. \end{cases} \tag{33}$$

If  $t_1 < \cdots < t_{n+1}$  then  $m_{i,j} > 0, \tilde{d}_{i,i} > 0$  and, using the above reasoning, we conclude that  $L_J$  is totally positive. Conversely, if  $L_J$  is totally positive and so,  $m_{i,j} > 0, \tilde{d}_{i,i} > 0$ , taking into account that

$$t_2 - t_1 = -\tilde{d}_{2,2} < 0, \quad t_i - t_{i-1} = m_{i,2}(t_{i-1} - t_{i-2}), \quad i = 3, \dots, n + 1,$$

we derive by induction that  $t_i - t_{i-1} < 0$  for  $i = 2, \dots, n + 1$ .

For decreasing nodes, the computation to high relative accuracy of  $L_J$ , its singular values, the inverse matrix  $L_J^{-1}$  and the resolution of  $L_J c = f$ , where  $f = (f_1, \dots, f_{n+1})^T$  has alternating signs can be deduced in a similar way to the increasing case. Finally, since  $J$  is a unitary matrix, the singular values of  $L_J$  coincide with

those of  $L$ . Similarly, taking into account that

$$L^{-1} = JL_J^{-1},$$

we can compute  $L^{-1}$  accurately. Finally, if we have a linear system of equations  $Ld = f$ , where the elements of  $f = (f_1, \dots, f_{n+1})^T$  have alternating signs, we can solve to high relative accuracy the system  $L_Jc = f$  and then obtain  $d = Jc$ .  $\square$

Let us observe that the factorization (26) of the collocation matrix of the Newton basis corresponding to the nodes  $t_1, \dots, t_{n+1}$  can be used to solve Lagrange interpolation problems. The Newton form of the Lagrange interpolant can be written as follows

$$p_n(t) = \sum_{i=0}^n d_{i+1} w_i(t), \tag{34}$$

with  $d_i := [t_1, \dots, t_i]f, i = 1, \dots, n + 1$ . The computation of the divided differences, which are usually obtained through the recursion in (19), can be alternatively obtained by solving the linear system

$$Ld = f,$$

with  $d := (d_1, \dots, d_{n+1})^T$  and  $f := (f_1, \dots, f_{n+1})^T$ , using the Matlab function `TNSOLVE` in Koev's web page [16] and taking the matrix form (3) of the bidiagonal decomposition of  $L$  as input argument. Taking into account Corollary 1,  $BD(L)$  (for increasing nodes) and  $BD(L_J)$  (for decreasing nodes) can be computed to high relative accuracy. Then, if the elements of the vector  $f$  are given to high relative accuracy and have alternating signs, the computation of the vector  $d$  can also be achieved to high relative accuracy.

Finally, the numerical experimentation illustrated in Section 6 will compare the vectors of divided differences obtained using the provided bidiagonal factorization, the recurrence (19) and, finally, the Matlab command `\` for the resolution of linear systems.

### 5 Error analysis and perturbation theory

Let us consider the collocation matrix  $L \in \mathbb{R}^{(n+1) \times (n+1)}$  of the Newton basis (17) at the nodes  $t_1 < \dots < t_{n+1}$  (see (24)). Now, we present a procedure for the efficient computation of  $BD(L)$ , that is, the matrix representation of the bidiagonal factorization (1) of  $L$ .

Taking into account (30), Algorithms 1 and 2 compute  $m_{i,j} := BD(L)_{i,j}, j < i$ , and  $p_i := BD(L)_{i,i}$ , respectively.

In the sequel, we analyze the stability of Algorithms 1 and 2 under the influence of imprecise computer arithmetic or perturbed input data. For this purpose, let us first introduce some standard notations in error analysis.

For a given floating-point arithmetic and a real value  $a \in \mathbb{R}$ , the computed element is usually denoted by either  $fl(a)$  or by  $\hat{a}$ . In order to study the effect of rounding

---

**Algorithm 1** Computation of  $m_{i,j}$ .

---

```

for  $i := 2$  to  $n+1$ 
   $m_{i,1} := 1$ 
  for  $j := 2$  to  $i-1$ 
     $M := \frac{t_i - t_{i-j} + 1}{t_{i-1} - t_{i-j}}$ 
     $m_{i,j} := m_{i,j-1} \cdot M$ 
  end j
end i

```

---



---

**Algorithm 2** Computation of  $p_{i,i}$ .

---

```

 $p_1 := 1$ 
for  $i := 2$  to  $n+1$ 
   $p_i := 1$ 
  for  $k := 1$  to  $i-1$ 
     $p_i := p_i \cdot (t_i - t_{i-k})$ 
  end k
end i

```

---

errors, we shall use the well-known models

$$\text{fl}(a \text{ op } b) = (a \text{ op } b)(1 + \delta)^{\pm 1}, \quad |\delta| \leq u, \tag{35}$$

where  $u$  denotes the unit roundoff and op any of the elementary operations  $+$ ,  $-$ ,  $\times$ ,  $/$  (see [13], p. 40 for more details).

Following [13], when performing an error analysis, one usually deals with quantities  $\theta_k$  such that

$$|\theta_k| \leq \gamma_k, \quad \gamma_k := \frac{ku}{1 - ku}, \tag{36}$$

for a given  $k \in \mathbb{N}$  with  $ku < 1$ . Taking into account, Lemmas 3.3 and 3.4 of [13], the following properties of the values (36) hold:

- a)  $(1 + \theta_k)(1 + \theta_j) = 1 + \theta_{k+j}$ ,
- b)  $\gamma_k + \gamma_j + \gamma_k \gamma_j \leq \gamma_{k+j}$ ,
- c)  $\gamma_k + u \leq \gamma_{k+1}$ ,
- d) if  $\rho_i = \pm 1$ ,  $|\delta_i| \leq u$ ,  $i = 1, \dots, k$ , then

$$\prod_{i=1}^k (1 + \delta_i)^{\rho_i} = 1 + \theta_k.$$

For example, statement a) above means that for any given two values  $\theta_k$  and  $\theta_j$ , bounded by  $\gamma_k$  and  $\gamma_j$ , respectively, there exists a number  $\theta_{k+j}$ , bounded by  $\gamma_{k+j}$ , such that the above identity holds. Further use of the previous symbols must be intended in this respect.

The following result analyzes the numerical error due to imprecise computer arithmetic in Algorithms 1 and 2, showing that both compute the bidiagonal factorization (1) of  $L$  accurately in a floating point arithmetic.

**Theorem 3** For  $n > 1$ , let  $L \in \mathbb{R}^{(n+1) \times (n+1)}$  be the collocation matrix (24) of the Newton basis (17) at the nodes  $t_1 < \dots < t_{n+1}$ . Let  $BD(L) = (b_{i,j})_{1 \leq i,j \leq n+1}$  be the matrix form of the bidiagonal decomposition (1) of  $L$  and  $\text{fl}(BD(L)) = (\text{fl}(b_{i,j}))_{1 \leq i,j \leq n+1}$  be the matrix computed with Algorithms 1 and 2 in floating point arithmetic with machine precision  $u$ . Then

$$\left| \frac{b_{i,j} - \text{fl}(b_{i,j})}{b_{i,j}} \right| \leq \gamma_{4n-5}, \quad 1 \leq i, j \leq n + 1. \tag{37}$$

**Proof** For  $i > j$ ,  $b_{i,j}$  can be computed using Algorithm 1. Accumulating relative errors as proposed in [13], we can easily derive

$$\left| \frac{b_{i,j} - \text{fl}(b_{i,j})}{b_{i,j}} \right| \leq \gamma_{4j-5} \leq \gamma_{4i-9} \leq \gamma_{4n-5}, \quad 1 \leq j < i \leq n + 1. \tag{38}$$

Analogously, for  $i = j$ ,  $b_{i,i}$  can be computed using Algorithm 2 and we have

$$\left| \frac{b_{i,i} - \text{fl}(b_{i,i})}{b_{i,i}} \right| \leq \gamma_{2i-3} \leq \gamma_{2n-1}, \quad 1 \leq i \leq n + 1. \tag{39}$$

Finally, since  $2n - 1 \leq 4n - 5$  for  $n > 1$ , the result follows. □

Now, we analyze the effect on the bidiagonal factorization of the collocation matrices of Newton bases due to small relative perturbations in the interpolation nodes  $t_i$ ,  $i = 1, \dots, n + 1$ . Let us suppose that the perturbed nodes are

$$t'_i = t_i(1 + \delta_i), \quad i = 1, \dots, n + 1.$$

We define the following values that will allow us to obtain an appropriate structured condition number, in a similar way to other analyses performed in [7, 14, 22, 23, 25, 27]:

$$\text{rel\_gap}_t := \min_{i \neq j} \frac{|t_i - t_j|}{|t_i| + |t_j|}, \quad \theta := \max_i \frac{|t_i - t'_i|}{|t_i|} = \max_i \delta_i, \quad \kappa := \frac{1}{\text{rel\_gap}_t}, \tag{40}$$

where  $\text{rel\_gap}_t \gg \theta$ .

**Theorem 4** Let  $L$  and  $L'$  be the collocation matrices (24) of the Newton basis (17) at the nodes  $t_1 < \dots < t_{n+1}$  and  $t'_1 < \dots < t'_{n+1}$ , respectively, with  $t'_i = t_i(1 + \delta_i)$ , for  $i = 1, \dots, n + 1$ , and  $|\delta_i| \leq \theta_i$ . Let  $BD(L) = (b_{i,j})_{1 \leq i,j \leq n+1}$  and  $BD(L') = (b'_{i,j})_{1 \leq i,j \leq n+1}$  be the matrix form of the bidiagonal factorization of  $L$  and  $L'$ , respectively. Then

$$\left| \frac{b_{i,j} - b'_{i,j}}{b_{i,j}} \right| \leq \frac{(2n - 2)\kappa\theta}{1 - (2n - 2)\kappa\theta}, \quad 1 \leq i, j \leq n + 1.. \tag{41}$$

**Proof** First, let us observe that

$$t'_i - t'_j = (t_i - t_j)(1 + \delta_{i,j}), \quad |\delta_{i,j}| \leq \frac{\theta}{rel\_gap_t}.$$

Accumulating the perturbations in the style of Higham (see Chapter 3 of [13]), we derive

$$m'_{i,j} = m_{i,j}(1 + \bar{\delta}), \quad |\bar{\delta}| \leq \frac{2(n-1)\kappa\theta}{1 - (2n-2)\kappa\theta}. \tag{42}$$

Analogously, for the diagonal entries  $p_i$  we have

$$p'_i = p_i(1 + \bar{\delta}), \quad |\bar{\delta}| \leq \frac{(n-1)\kappa\theta}{1 - (n-1)\kappa\theta}. \tag{43}$$

Finally, using (42) and (43), the result follows. □

Formula (41) can also be obtained using Theorem 7.3 of [25] where it is shown that small relative perturbations in the nodes of a Cauchy-Vandermonde matrix produce only small relative perturbations in its bidiagonal factorization. Let us note that the entries  $m_{i,j}$  and  $p_{i,i}$  of the bidiagonal decomposition of collocation matrices of Newton bases at distinct nodes coincide with those of Cauchy-Vandermonde matrices with  $l = 0$ . Finally, let us note that quantity  $(2n - 2)\kappa\theta$  can be seen as an appropriated structured condition number for the mentioned collocation matrices of the Newton bases (17).

## 6 Numerical experiments

In order to give a numerical support to the theoretical methods discussed in the previous sections, we provide a series of numerical experiments. In all cases, we have considered ill-conditioned nonsingular collocation matrices  $L$  of  $(n + 1)$ -dimensional Newton bases (17) with equidistant increasing or decreasing nodes for  $n = 15, 25, 50, 100$  in a unit-length interval—with the exception of the experiments concerning the Runge function, where the chosen interval is  $[-2, 2]$ .

Let us recall that once the bidiagonal decomposition of a TP matrix  $A$  is computed to high relative accuracy, its matrix representation  $BD(A)$  can be used as an input argument of the functions of the `TNTool` package, made available by Koev in [16], to perform to high relative accuracy the solution of different algebraic problems. In particular, `TNSolve` is used to resolve linear systems  $Ax = b$  for vectors  $b$  with alternating signs, `TNSingularValues` for the computation of the singular values of  $A$  and `TNInverseExpand` (see [24]) to obtain the inverse  $A^{-1}$ . The computational cost of `TNSingularValues` is  $O(n^3)$ , being  $O(n^2)$  for the other methods.

In order to check the accuracy of our bidiagonal decomposition approach,  $BD(L)$  and  $BD(L_J)$  (see (30) and (33)) have been considered with the mentioned routines to solve each problem. The obtained approximations have been compared with those

computed by other commonly used procedures, such as standard Matlab routines or the divided difference recurrence (19), in the case of the interpolant coefficients.

In this context, the values provided by Wolfram Mathematica 13.1 with 100-digit arithmetic have been taken as the exact solution of the considered algebraic problems. Then, for each computed scalar  $\sigma$ , vector  $y$ , and matrix  $A$ , the corresponding relative errors have been calculated by  $e := |(\sigma - \tilde{\sigma})/\sigma|$ ,  $e := \|y - \tilde{y}\|_2/\|y\|_2$ , and  $e := \|A - \tilde{A}\|_2/\|A\|_2$ , respectively, where  $\tilde{\sigma}$ ,  $\tilde{y}$ , and  $\tilde{A}$  denote the approximations obtained with the considered methods.

**Computation of coefficients of the Newton form of the Lagrange interpolant to HRA**

In this numerical experiment, the coefficients  $d_i := [t_1, \dots, t_i]f, i = 1, \dots, n + 1$ , of the Newton form of the Lagrange interpolant (34) were computed using three different methods. First, the divided difference recurrence (19) determined  $d_i$  directly. Later, the standard \ Matlab command and the bidiagonal decomposition of Section 4 were also used. Let us note that both of them obtain  $d_i$  as the solution of the linear system  $Ld = f$ , where  $L$  is the collocation matrix of the considered Newton basis with  $d := (d_1, \dots, d_{n+1})^T, f := (f_1, \dots, f_{n+1})^T$ . Let us recall that the last method requires the bidiagonal decomposition  $BD(L)$  (for increasing nodes) and  $BD(L_J)$  (for decreasing nodes) as an input argument for the Octave/Matlab function `TNSolve`, available in [16]. Note that for nodes in decreasing order, the system  $L_Jc = f$ , with  $L_J := LJ$  and  $J := \text{diag}((-1)^{i-1})_{1 \leq i \leq n+1}$ , is solved and the obtained solution  $c$  allows to recover  $d$  as  $d = Jc$  (see the proof of Corollary 1).

In all cases,  $f_i, i = 1, \dots, n + 1$ , were chosen to be random integers with uniform distribution in  $[0, 10^3]$ , adding alternating signs to guarantee high relative accuracy when obtaining the coefficients with the divided difference recurrence (19) and with `TNSolve`.

Relative errors are shown in Table 1. The results clearly illustrate the high relative accuracy achieved with the divided difference recurrence (19) and the function `TNSolve` applied to the bidiagonal decompositions proposed in this work, supporting the theoretical results of the previous sections.

Let us note that the extension to the bivariate interpolation with the Lagrange-type data using Newton bases and leading to the use of a generalized Kronecker product of their collocation matrices will be addressed by the authors by applying the results in this paper. In this sense, the bivariate Lagrange interpolation problem with Bernstein bases and a generalized Kronecker product is analyzed in [26].

**Computation of the coefficients of the Newton form for a function of constant sign**

As explained in the previous case, when determining the interpolating polynomial coefficients of the Newton form, to guarantee high relative accuracy the vector  $(f_1, \dots, f_{n+1})$  is required to have alternating signs. However, other sign structures can be addressed by the methods discussed in this work, although HRA cannot be assured in these cases. To illustrate the behavior of both the divided differences recurrence and the bidiagonal decomposition approach in such scenario, numerical experiments for the classical Runge function  $1/(1 + 25x^2)$  with equidistant nodes in the  $[-2, 2]$

**Table 1** Relative errors of the approximations to the solution of the linear system  $Ld = f$ , for equidistant nodes in a unit-length interval in increasing and decreasing order

$n + 1$	$t_1 < \dots < t_{n+1}$		$t_1 > \dots > t_{n+1}$	
	Div. diff.	$L \setminus f$	Div. diff.	$L \setminus f$
15	$1.6e - 16$	$1.3e - 13$	$8.7e - 17$	$3.0e - 17$
25	$8.7e - 16$	$6.2e - 11$	$6.2e - 16$	$2.2e - 10$
50	$1.1e - 15$	$1.0e - 3$	$2.9e - 15$	$2.8e - 4$
100	$4.7e - 15$	$2.6e + 12$	$5.2e - 15$	$5.8e + 12$
		TNSolve( $BD(L), f$ )		$J \cdot \text{TNSolve}(BD(L), f)$
		$2.4e - 17$		$3.0e - 17$
		$8.0e - 16$		$1.9e - 16$
		$2.1e - 15$		$3.8e - 15$
		$5.8e - 15$		$6.1e - 15$

**Table 2** Relative errors of the approximations to the solution of the linear system  $Ld = f$ , for the classical Runge function  $(1 + 25x^2)^{-1}$ , with equidistant nodes in increasing order in  $[-2, 2]$

$n + 1$	Div. diff.	$L \setminus f$	$TNSolve(BD(L), f)$
15	$1.5e - 16$	$9.2e - 16$	$2.5e - 16$
25	$7.0e - 16$	$5.6e - 14$	$7.2e - 16$
50	$8.1e - 14$	$2.4e - 09$	$7.5e - 14$
100	$3.2e - 08$	$7.5e + 01$	$3.3e - 08$

interval were performed. Relative errors of the computed coefficients of the Newton form are gathered for several  $n$  in Table 2, comparing the performance of the divided difference recurrence (19), the function `TNSolve` applied to the bidiagonal factorization proposed in Section 4 and the standard `\ Matlab` command. As can be seen, good accuracy is achieved for small enough values of  $n$ , while relative errors for increasing  $n$  behave considerably worse with the standard `\ Matlab` command than with both the divided differences recurrence and the bidiagonal decomposition approaches, which achieve similar precision.

**Computation of singular values of  $L$  to HRA** In this third numerical experiment, the precision achieved by two methods in the computation of the lowest singular value of  $L$  is analyzed. It should be noted that ill-conditioned matrices have very small singular values, and that small relative variations in the entries of a TP matrix can produce huge variations on them. As a consequence, a good relative accuracy in the determination of the singular values of a TP matrix is only achieved by standard methods in the case of the largest ones (see [14]).

On the one hand, we computed the lowest singular value of  $L$  by means of the routine `TNSingularValues`, using  $BD(L)$  for increasing and  $BD(L_J)$  for decreasing order. Notice that, for nodes in decreasing order, the singular values of  $L_J$  coincide with those of  $L$  since  $J$  is unitary. On the other hand, the lowest singular value of  $L$  was computed with the Matlab command `svd`.

The results are gathered in Table 3. As expected, when the bidiagonal decomposition approach is used, the singular values are computed to high relative accuracy for every dimension  $n$ , despite the ill-conditioning of the matrices and in contrast with the standard Matlab procedure.

**Table 3** Relative errors when computing the lowest singular value of  $L$  for equidistant nodes in an interval of unit-length in increasing and decreasing order

$n + 1$	$t_1 < \dots < t_{n+1}$		$t_1 > \dots > t_{n+1}$	
	<code>svd(L)</code>	<code>TNSingV(BD(L))</code>	<code>svd(L)</code>	<code>TNSingV(BD(L_J))</code>
15	$7.3e - 12$	$6.6e - 16$	$5.9e - 12$	$5.7e - 16$
25	$1.0e - 7$	$5.2e - 16$	$1.7e - 7$	$4.3e - 15$
50	$1.7e + 1$	$5.1e - 16$	$1.9e + 1$	$8.3e - 15$
100	$1.4e + 12$	$1.4e - 15$	$5.1e + 13$	$2.6e - 16$

**Table 4** Relative errors when computing the inverse of collocation matrices of the Newton basis with equidistant nodes in a unit-length interval in increasing and decreasing order

$n + 1$	$t_1 < \dots < t_{n+1}$		$t_1 > \dots > t_{n+1}$	
	$\text{inv}(L)$	$\text{TNInvEx}(BD(L))$	$\text{inv}(L)$	$J \cdot \text{TNInvEx}(BD(L_J))$
15	$2.5e - 13$	$1.3e - 15$	$1.5e - 13$	$5.7e - 16$
25	$8.8e - 11$	$4.8e - 15$	$1.4e - 10$	$8.8e - 16$
50	$2.7e - 3$	$8.7e - 15$	$2.9e - 3$	$2.5e - 15$
100	$1.7e + 12$	$6.8e - 15$	$2.8e + 12$	$5.1e - 16$

**Computation of inverse of  $L$  to HRA** Finally, to show another application of the bidiagonal decompositions presented in this work, the inverse of the considered collocation matrices were computed. We used two different procedures: the Matlab `inv` command and the function `TNInverseExpand` with our bidiagonal decompositions  $BD(L)$  for increasing and  $BD(L_J)$  for decreasing order. To compute the inverse for the decreasing order case to HRA, note that first the inverse of  $L_J$  is obtained and then  $L^{-1}$  is recovered with  $L^{-1} = JL_J^{-1}$ .

It should be pointed out that formula (13) in [3] allows to compute  $L^{-1}$  to high relative accuracy for any order of the nodes. Displayed in Table 4, the results of these numerical experiments show that our bidiagonal decomposition approach preserves high relative accuracy for any of the values of  $n$  tested, in stark contrast with the standard Matlab routine.

**Acknowledgements** We thank the editors and the anonymous referees for their helpful comments and suggestions, which have improved this paper.

**Author Contributions** The authors contributed equally to this work.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was partially supported through the Spanish research grants PID2022-138569NB-I00 and RED2022-134176-T (MCI/AEI) and by Gobierno de Aragón (E41\_23R, S60\_23R).

**Availability of supporting data** The authors confirm that the data supporting the findings of this study are available within the manuscript. The Matlab and Mathematica codes to run the numerical experiments are available upon request.

## Declarations

**Competing interests** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Ando, T.: Totally positive matrices. *Linear Algebra Appl.* **90**, 165–219 (1987)
2. Berut, J.P., Trefethen, L.N.: Barycentric Lagrange interpolation. *SIAM Rev.* **46**, 501–517 (2004)
3. Carnicer, J.M., Khier, Y., Peña, J.M.: Factorization of Vandermonde matrices and the Newton formula. In: Ahusborde, É., Amrouche, C., et al. (eds.) Thirteenth International Conference Zaragoza-Pau on Mathematics and its Applications. *Monografías Matemáticas "García de Galdeano"*, vol. 40, pp. 53–60. Universidad de Zaragoza, Zaragoza (2015)
4. Delgado, J., Orera, H., Peña, J.M.: Accurate computations with Laguerre matrices. *Numer. Linear Algebra Appl.* **26**(10), e2217 (2019)
5. Delgado, J., Orera, H., Peña, J.M.: Accurate algorithms for Bessel matrices. *J. Sci. Comput.* **80**, 1264–1278 (2019)
6. Demmel, J., Koev, P.: The accurate and efficient solution of a totally positive generalized Vandermonde linear system. *SIAM J. Matrix Anal. Appl.* **27**, 42–52 (2005)
7. Demmel, J., Koev, P.: Accurate SVDs of polynomial Vandermonde matrices involving orthonormal polynomials. *Linear Algebra Appl.* **27**, 382–396 (2006)
8. Finck, T., Heinig, G., Rost, K.: An inversion formula and fast algorithms for Cauchy-Vandermonde matrices. *Linear Algebra Appl.* **183**, 179–191 (1993)
9. Fischer, B., Reichel, L.: Newton interpolation in Fejér and Chebyshev points. *Math. Comp.* **53**, 265–278 (1989)
10. Gasca, M., Peña, J.M.: Total positivity and Neville elimination. *Linear Algebra Appl.* **165**, 25–44 (1992)
11. Gasca, M., Peña, J.M.: A matricial description of Neville elimination with applications to total positivity. *Linear Algebra Appl.* **202**, 33–53 (1994)
12. Gasca, M., Peña, J.M.: On factorizations of totally positive matrices. In: Gasca, M., Micchelli, C.A. (eds.) *Total Positivity and Its Applications*, pp 109–130. Kluwer Academic Publishers, Dordrecht, The Netherlands (1996)
13. Higham, N.J.: *Accuracy and stability of numerical algorithms*, 2nd edn. SIAM, Philadelphia (2002)
14. Koev, P.: Accurate eigenvalues and SVDs of totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **27**, 1–23 (2005)
15. Koev, P.: Accurate computations with totally nonnegative matrices. *SIAM J. Matrix Anal. Appl.* **29**, 731–751 (2007)
16. Koev, P.: Software for performing virtually all matrix computations with nonsingular totally nonnegative matrices to high relative accuracy. <http://math.mit.edu/~plamen/software/TNTool.html>
17. Mainar, E., Peña, J.M.: Accurate computations with collocation matrices of a general class of bases. *Numer. Linear Algebra Appl.* **25**, e2184 (2018)
18. Mainar, E., Peña, J.M., Rubio, B.: Accurate computations with collocation and Wronskian matrices of Jacobi polynomials. *J. Sci. Comput.* **87**, 77 (2021)
19. Mainar, E., Peña, J.M., Rubio, B.: Accurate computations with matrices related to bases  $\{t^i e^{\lambda t}\}$ . *Adv. Comput. Math.* **48**, 38 (2022)
20. Mainar, E., Peña, J.M., Rubio, B.: Accurate computations with Gram and Wronskian matrices of geometric and Poisson bases. *Rev. Real Acad. Cienc. Exactas Fis. Nat. Ser. A-Mat* **116**, 126 (2022)
21. Mainar, E., Peña, J.M., Rubio, B.: High relative accuracy through Newton bases. *Numer. Algorithms* (2023). <https://doi.org/10.1007/s11075-023-01588-9>
22. Marco, A., Martínez, J.J.: Accurate computations with totally positive Bernstein-Vandermonde matrices. *Electron. J. Linear Algebra* **26**, 357–380 (2013)
23. Marco, A., Martínez, J.J.: Bidiagonal decomposition of rectangular totally positive Said-Ball-Vandermonde matrices: error analysis, perturbation theory and applications. *Linear Algebra Appl.* **495**, 90–107 (2016)
24. Marco, A., Martínez, J.J.: Accurate computation of the Moore-Penrose inverse of strictly totally positive matrices. *J. Comput. Appl. Math.* **350**, 299–308 (2019)
25. Marco, A., Martínez, J.J., Peña, J.M.: Accurate bidiagonal decomposition of totally positive Cauchy-Vandermonde matrices and applications. *Linear Algebra Appl.* **517**, 63–84 (2017)
26. Marco, A., Martínez, J.J., Viaña, R.: Accurate polynomial interpolation by using Bernstein basis. *Numer. Algor.* **75**, 655–674 (2017)
27. Marco, A., Martínez, J.J., Viaña, R.: Accurate computations with collocation matrices of the Lupaş-type (p, q)-analogue of the Bernstein basis. *Linear Algebra Appl.* **651**, 312–331 (2022)

28. Marco, A., Martínez, J.J., Viaña, R.: Accurate bidiagonal decomposition of Lagrange-Vandermonde matrices and applications. *Numer. Linear Algebra Appl.* **31**(1), e2527 (2024)
29. Oruç, H., Phillips, G.M.: Explicit factorization of the Vandermonde matrix. *Linear Algebra Appl.* **315**, 113–123 (2000)
30. Rutishauser, H.: *Vorlesungen über numerische Mathematik*, vol. 1, Birkhäuser, Basel, Stuttgart, 1976; English translation. In W. Gautschi (ed.), *Lectures on Numerical Mathematics*, Birkhäuser, Boston, (1990)
31. Tal-Ezer, H.: High degree polynomial interpolation in Newton form. *SIAM J. Sci. Statist. Comput.* **12**, 648–667 (1991)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.