Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(2): 3836–3847.
DOI: 10.3934/math.2025178
Received: 01 January 2025
Revised: 12 February 2025
Accepted: 19 February 2025
Published: 26 February 2025

Research article

High relative accuracy for a Newton form of bivariate interpolation problems

Yasmina Khiar[1], **Esmeralda Mainar**[1], **Eduardo Royo-Amondarain**[2,*] and **Beatriz Rubio**[1]

[1] Departamento de Matemática Aplicada / IUMA, Universidad de Zaragoza, Spain
[2] Departamento de Matemática Aplicada / CAPA, Universidad de Zaragoza, Spain

* **Correspondence:** Email: eduroyo@unizar.es.

Abstract: The problem of bivariate polynomial interpolation using Newton-type bases is examined, leading to the application of a generalized Kronecker matrix product. Algorithms for computing the coefficients of the interpolating polynomial are presented, along with conditions that ensure relative errors of the order of machine precision. A generalization of the classical recursion formula of divided differences in two dimensions is proposed for grids that generalize the standard rectangular layout. Numerical experiments demonstrate the high accuracy achieved by the proposed approach.

Keywords: high relative accuracy; bidiagonal decompositions; totally positive matrices; bivariate interpolation
Mathematics Subject Classification: 15A06, 15A18, 15A23, 65F05, 65G50

1. Introduction

A classical topic in numerical analysis is the univariate Lagrange interpolation problem, which has been extensively studied in the literature. Other univariate interpolation problems, such as Hermite and Hermite–Birkhoff interpolation, have also received significant attention. However, when extending this to the multivariate case, interpolation problems increase notably their complexity—e.g., the uniqueness of the solution when the number of coefficients to be determined equals the initial data cannot be guaranteed, in contrast to the univariate case. For a detailed survey of this theory, we refer the reader to [12].

The interest in bivariate interpolation extends beyond purely mathematical fields and finds applications in diverse areas such as geostatistics and image processing [8, 9]. Among the different approaches used to address these problems, a natural one—one of the first to be developed—is interpolation by the tensor product. In this case, the interpolation problem can be reformulated as a linear system whose coefficient matrix is a generalized Kronecker product related to collocation matrices of the univariate bases involved. Furthermore, the linear system for the bivariate problem can be simplified by breaking it down into several smaller linear systems associated with univariate interpolation problems. These smaller systems are defined using subsets of the original nodes and their corresponding values [21]. It is worth noting that this approach can be made in grids that generalize the usual rectangular layout by allowing nodes of the form (x_i, y_{ij}) for $i = 0, \dots, n$, $j = 1, \dots, m$. Particular examples of these type of lattices which are of interest include schemes composed of interlacing pairs of rectangular grids, such as those of Morrow and Patterson [22] or, notably, Padua [2] and Padua-like points [1], which have been shown to posses remarkable properties when used for approximation and cubature; a discussion of a wider class of nodes can be found in [10]. In [4], the bivariate interpolation problem considering a Lagrange basis with Padua and scattered points is addressed. Related to a least squares approximation problem, a recent contribution in the field explores the benefits of considering subsets of rectangular lattices [5], favoring those that are close to Chebyshev–Lobatto nodes.

However, the tensor interpolation procedure described above has a caveat: If any of the univariate problems is ill-conditioned, the bivariate problem will also exhibit a poor condition number. Unfortunately, since the univariate systems are defined by collocation matrices—notoriously poorly conditioned for many bases—this is a common scenario. A desirable approach to addressing these problems is to find the vector $\boldsymbol{c}$ with the coefficients of the interpolant with respect to the considered basis of the interpolating space using algorithms that compute an approximation $\tilde{\boldsymbol{c}}$ to a high relative accuracy (HRA). This means ensuring that

$$\frac{\|\boldsymbol{c} - \tilde{\boldsymbol{c}}\|}{\|\boldsymbol{c}\|} \leq Ku, \tag{1.1}$$

where $K > 0$ is a constant and u is the given floating-point arithmetic unit roundoff. Sufficient conditions to guarantee HRA are that the algorithm performs only products, quotients, sums of numbers of the same sign, subtractions of numbers of opposite sign, and the initial data. In other words, the subtraction of numbers of the same sign that are not the initial data is excluded—this is known as the NIC (no inaccurate cancellation) condition [6].

This class of algorithms has been explored over the past decades in the context of totally positive (TP) matrices, which are characterized by all their minors being non-negative (cf. [3, 17–19]). These studies achieve HRA by obtaining a bidiagonal factorization of the considered TP matrices. Combined with the algorithms provided in the TNTool package [15], this approach enables the solution of significant algebraic problems to HRA. In particular, these include the resolution of linear systems $A\boldsymbol{x} = \boldsymbol{b}$, where A is a TP matrix and $\boldsymbol{b}$ is a vector whose components have an alternating sign pattern. This can be done to achieve HRA, provided that the bidiagonal decomposition of the TP matrix is obtained to HRA and used as an input with the TNSolve subroutine of the abovementioned package. This is of relevance in the resolution of univariate interpolation problems using polynomial bases, such as monomial [7], Bernstein [19], or Newton [14]. Furthermore, it is possible to extend this approach to the bivariate interpolation problem through the generalized Kronecker product [21]. In fact, in [20], the tensor product Bernstein basis case is addressed, whereas the more general case of rank-descending matrices is analyzed in [23] under different perspectives—in particular, the specific conditions under which the bivariate problem formulated in the monomial tensor product basis can be solved to HRA are derived.

In this work, the bivariate interpolation problem in the tensor product's Newton basis is addressed, providing an alternative perspective to [23], and deriving the specific conditions under which the coefficients of the polynomial interpolant can be computed to HRA. To do so, it is shown how the two-dimensional problem can be reduced to a series of univariate problems that can be solved with two different approaches: the recursive formula of divided differences or the bidiagonal decomposition strategy discussed above. In the former case, the equivalence to the two-dimensional version of the divided differences is observed for nodes displayed in a rectangular layout, and a generalization for grids of the form (x_i, y_{ij}) for $i = 0, \ldots, n, j = 1, \ldots, m$ is proposed; in the latter, the procedure can be generalized to any basis that satisfies similar conditions. These results are gathered in Section 2. Additionally, Section 3 outlines sufficient conditions on the input data for the bivariate interpolation problem in the Newton-type basis, guaranteeing its resolution up to HRA using the proposed approaches. Finally, Section 4 includes numerical experiments that demonstrate and compare the accuracy achieved by each method.

2. The bivariate polynomial interpolation problem

We consider $(n + 1)(m + 1)$ distinct interpolation nodes $\{(x_i, y_{ij}) \mid i = 0, \ldots, n; j = 0, \ldots m\}$ and the interpolation data $f_{ij} = f(x_i, y_{ij}) \in \mathbb{R}$, $i = 0, \ldots, n$, $j = 0, \ldots m$, for some function f. The bivariate polynomial interpolation problem can be understood as the problem of finding a polynomial p of $P_{n,m}(x, y)$—the space of polynomials of a degree less than or equal to n in x and less than or equal to m in y—satisfying

$$p(x_i, y_{ij}) = f_{ij}, \quad i = 0, \ldots, n, \quad j = 0, \ldots m. \tag{2.1}$$

We shall consider a Newton-type basis of $P_{n,m}(x, y)$ given by

$$\{w_{ij}^{(n,m)}(x, y) \mid i = 0, \ldots, n; j = 0, \ldots m\}, \tag{2.2}$$

with

$$w_{ij}^{(n,m)}(x, y) = \prod_{k=0}^{i-1} \prod_{l=0}^{j-1} (x - x_k)(y - y_{kl}), \quad i = 0, \ldots, n, \quad j = 0, \ldots m,$$

and the convention that the empty product is equal to 1. In this case, we can write

$$p(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{m} d_{ij} w_{ij}^{(n,m)}(x, y). \tag{2.3}$$

Let us note that for the particular case of a rectangular grid (x_i, y_j) for $i = 0, \ldots, n, j = 0, \ldots, m$, the coefficients d_{ij} can be identified with the two-dimensional version of the divided differences, as in [13].

In order to determine the coefficients of the interpolant, it is possible to reformulate the above problem through the generalized Kronecker product (cf. [11]). In this way, computing the coefficients d_{ij} in (2.3) is equivalent to solving the linear system

$$L\boldsymbol{d} = \boldsymbol{b}, \tag{2.4}$$

where $\boldsymbol{d} = (\boldsymbol{d}_0, \ldots, \boldsymbol{d}_n)^T \in \mathbb{R}^{(n+1)(m+1)}$, $\boldsymbol{b} = (\boldsymbol{f}_0, \ldots, \boldsymbol{f}_n)^T \in \mathbb{R}^{(n+1)(m+1)}$, with $\boldsymbol{d}_i = (d_{i0}, \ldots, d_{im}), \boldsymbol{f}_i = (f_{i0}, \ldots, f_{im}) \in \mathbb{R}^{m+1}$, for $i = 0, \ldots, n$, and the matrix $L \in \mathbb{R}^{(n+1)(m+1)\times(n+1)(m+1)}$ is given by the generalized

Kronecker product

$$L = L_x \otimes L_i = \begin{pmatrix} l_{00}L_0 & l_{01}L_0 & \cdots & l_{0n}L_0 \\ l_{10}L_1 & l_{11}L_1 & \cdots & l_{1n}L_1 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n0}L_n & l_{n1} & L_n \cdots & l_{nn}L_n \end{pmatrix}, \tag{2.5}$$

where $L_x = (l_{ij}) \in \mathbb{R}^{(n+1)\times(n+1)}$ is the collocation matrix of the univariate Newton basis at the nodes $x_0, \dots, x_n$

$$L_x = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & 0 & \cdots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & \prod_{j=0}^{n-1}(x_n - x_j) \end{pmatrix}, \tag{2.6}$$

and $L_i \in \mathbb{R}^{(m+1)\times(m+1)}$ is the collocation matrix at the nodes $y_{i0}, \dots, y_{im}$

$$L_i = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & y_{i1} - y_{i0} & 0 & \cdots & 0 \\ 1 & y_{i2} - y_{i0} & (y_{i2} - y_{i0})(y_{i2} - y_{i1}) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & y_{im} - y_{i0} & (y_{im} - y_{i0})(y_{im} - y_{i1}) & \cdots & \prod_{j=0}^{m-1}(y_{im} - y_{ij}) \end{pmatrix}, \quad i = 0, \dots, n. \tag{2.7}$$

Observe that if the nodes satisfy $x_j \neq x_k$ and $y_{ij} \neq y_{ik}$ for $j \neq k$, the matrices L_x and L_i, $i = 0, \dots, n$, are nonsingular. In this case, $L_x \otimes L_i$ is also nonsingular and the solution of our bivariate interpolation problem is unique.

To solve the bivariate interpolation problem in the bivariate Newton-type basis (2.2), we consider the algorithm presented by Martínez in [21] for solving nonsingular linear systems whose coefficient matrices have a generalized Kronecker product structure. The steps are as follows.

Algorithm 2.1.

(1) Solve the $n+1$ linear systems $L_i z = \boldsymbol{f}_i^T$, whose solutions are denoted by $\boldsymbol{b}_i \in \mathbb{R}^{m+1}$, $i = 0, \dots, n$. Then, define the matrix $B := (\boldsymbol{b}_0, \boldsymbol{b}_1, \dots, \boldsymbol{b}_n) \in \mathbb{R}^{(m+1)\times(n+1)}$.

(2) Solve the $m+1$ linear systems $L_x z = \boldsymbol{r}_j^T$, where $\boldsymbol{r}_j \in \mathbb{R}^{n+1}$ is the j-th row of B, and whose solutions are denoted by $\boldsymbol{d}_j \in \mathbb{R}^{n+1}$, $j = 0, \dots, m$. Then, define the matrix $D := (\boldsymbol{d}_0, \dots, \boldsymbol{d}_m) \in \mathbb{R}^{(n+1)\times(m+1)}$.

(3) Finally, the solution of the linear system $L\boldsymbol{d} = \boldsymbol{b}$ in (2.4) is given by

$$\boldsymbol{d} = (d_{00}, \dots, d_{0m}, d_{10}, \dots, d_{1m}, \dots, d_{n0}, \dots, d_{nm}) \in \mathbb{R}^{(n+1)(m+1)}.$$

It is important to note that each system to be solved in Algorithm 1 corresponds to an univariate Lagrange interpolation problem whose solution is given by the coefficients of the interpolating polynomial expressed in terms of a Newton basis, i.e., the divided differences. Let us recall that, in the one-dimensional case, the divided differences at distinct points $x_i, \dots, x_{i+k}$ of a function f can be computed through the following recursion formula:

$$[x_i, \dots, x_{i+k}]f = \begin{cases} f(x_i), & \text{for } k = 0, \\ \dfrac{[x_{i+1}, \dots, x_{i+k}]f - [x_i, \dots, x_{i+k-1}]f}{x_{i+k} - x_i}, & \text{for } k \geq 1. \end{cases} \tag{2.8}$$

Additionally, in [14], the authors present an alternative procedure to compute the divided differences, based on the bidiagonal factorization of the collocation matrix involved using Neville elimination. It should be noted that, in terms of computational cost, both approaches achieve the resolution of the system in $O(n^2)$ operations.

As an alternative, the bivariate problem formulated in a rectangular mesh (x_i, y_j), $i = 0, \dots, n$, $j = 0, \dots m$, can be directly addressed defining the two-dimensional version of the divided differences at Cartesian nodes, which can be identified with the coefficients d_{ij} in (2.3). This is done by fixing one of the variables and applying recursion (2.8) in the remaining component (see, e.g., [13]). In this way, fixing the x variable, we first compute $[x_i; y_0, \dots, y_j]f$, $i = 0, \dots, n$, $j = 0, \dots, m$, by means of the following recursion:

$$[x_i; y_j, \dots, y_{j+l}]f = \begin{cases} f(x_i, y_j), & l = 0, \\ \dfrac{[x_i; y_{j+1}, \dots, y_{j+l}]f - [x_i; y_j, \dots, y_{j+l-1}]f}{y_j - y_0}, & l > 0. \end{cases} \tag{2.9}$$

Once this is done, $d_{ij} := [x_0, \dots, x_i; y_0, \dots, y_j]f$, $i = 0, \dots, n$, $j = 0, \dots, m$, can be determined by fixing the other variable and applying the same idea again. Notably, it is straightforward to verify that these two steps are identical to those in Algorithm 2.1.

A natural question that arises is whether the equivalence between Algorithm 2.1 and the recursive computation of divided differences from (2.9) can be extended from rectangular lattices to the more general case (x_i, y_{ij}) for $i = 0, \dots, n$ and $j = 1, \dots, m$. This includes interlacing rectangular grids (with equal lengths in at least one dimension), discussed by Floater in [10], with notable examples such as Padua points [1,2]. The answer comes easily, since the two-dimensional version of divided differences can be extended to the considered grids in a straightforward manner, defining $[x_0, \dots, x_i; \boldsymbol{y}_0, \dots, \boldsymbol{y}_j]f$ with $\boldsymbol{y}_j = (y_{0j}, \dots, y_{ij})^T$. Then recursion (2.9) can be generalized simply by selecting the i-th component

$$[x_i; \boldsymbol{y}_j, \dots, \boldsymbol{y}_{j+l}]f = \begin{cases} f(x_i, y_{ij}), & l = 0, \\ \dfrac{[x_i; y_{i,j+1}, \dots, y_{i,j+l}]f - [x_i; y_{ij}, \dots, y_{i,j+l-1}]f}{y_{i,j+l} - y_{ij}}, & l > 0. \end{cases} \tag{2.10}$$

Finally, the coefficients d_{ij} in (2.3) can be computed as

$$d_{ij} = [x_0, \dots, x_i; \boldsymbol{y}_0, \dots, \boldsymbol{y}_j]f = \frac{[x_1, \dots, x_i; \boldsymbol{y}_0, \dots, \boldsymbol{y}_j]f - [x_0, \dots, x_{i-1}; \boldsymbol{y}_0, \dots, \boldsymbol{y}_j]f}{x_i - x_0}. \tag{2.11}$$

Let us illustrate this process with a small example. Consider a grid (x_i, y_{ij}), $i = 0, 1, 2$, $j = 0, 1$. First, we compute the $x$table of the generalized divided differences

$[x_0; \boldsymbol{y}_0]f$	$[x_1; \boldsymbol{y}_0]f$	$[x_2; \boldsymbol{y}_0]f$
$[x_0; \boldsymbol{y}_0, \boldsymbol{y}_1]f$	$[x_1; \boldsymbol{y}_0, \boldsymbol{y}_1]f$	$[x_2; \boldsymbol{y}_0, \boldsymbol{y}_1]f$

where the entries of each column are computed by (2.10). Then, by (2.11), the following xytable is obtained:

$[x_0; \boldsymbol{y}_0]f$	$[x_0; \boldsymbol{y}_0, \boldsymbol{y}_1]f$
$[x_0, x_1; \boldsymbol{y}_0]f$	$[x_0, x_1; \boldsymbol{y}_0, \boldsymbol{y}_1]f$
$[x_0, x_1, x_2; \boldsymbol{y}_0]f$	$[x_0, x_1, x_2; \boldsymbol{y}_0, \boldsymbol{y}_1]f$

in which the entries are precisely d_{ij}, and the coefficients of the solution (2.3).

Finally, let us observe that the computation of the proposed generalization for the two-dimensional divided differences remains equivalent to the steps of Algorithm 2.1, as previously demonstrated for rectangular lattices. This equivalence is evident upon observing that the earlier $x$table and xytable correspond directly to matrices B and D in Algorithm 2.1.

3. Accurate computations in the bivariate interpolation problem

We now analyze the accuracy of Algorithm 2.1 under both approaches for solving the linear systems involved. To proceed, we first introduce the following definitions.

Definition 3.1. *The set of nodes* (x_i, y_{ij})*,* $i = 0, \dots, n,\ j = 0, \dots, m$ *is said to be ordered if* $x_0 \leq \cdots \leq x_n$ *or* $x_0 \geq \cdots \geq x_n$*, and* $y_{i0} \leq \cdots \leq y_{im}$ *or* $y_{i0} \geq \cdots \geq y_{im}$*, for* $i = 0, \dots, n$*. On the other hand, a set of values* f_{ij} *is said to have a chessboard sign pattern if* $sgn(f_{ij}) = \pm(-1)^{i+j}$*,* $i = 0, \dots, n,\ j = 0, \dots m$*.*

We use the following auxiliary result.

Lemma 3.1. *Let* $A\boldsymbol{x} = \boldsymbol{b}$ *be a linear system, where A is a TP matrix and the entries of* $\boldsymbol{b}$ *exhibit an alternating sign pattern. Then the entries of x also have an alternating sign pattern.*

Proof. The result readily follows from the fact that the entries of A^{-1} have a chessboard sign pattern (see Section 3.1 of [16]). □

Now, we can provide sufficient conditions on the input data of the given bivariate interpolation problem in the Newton-type basis to guarantee that it can be solved to HRA.

Theorem 3.2. *Let* $\{(x_i, y_{ij}) \mid i = 0, \dots, n; j = 0, \dots m\}$ *be a set of ordered nodes. Then the interpolation problem in the bivariate Newton-type basis, formulated as in* (2.1)*, can be solved to HRA as long as the interpolation data* f_{ij} *have a chessboard sign pattern.*

Proof. Using Algorithm 1, the bivariate polynomial interpolation problem, equivalent to the linear system $L\boldsymbol{d} = \boldsymbol{b}$ in (2.4), gets reduced to $n + m + 2$ linear systems: $L_i\boldsymbol{z} = \boldsymbol{f}_i^T$, $i = 0, \dots, n$, and $L_x\boldsymbol{z} = \boldsymbol{r}_j^T$, $j = 0, \dots, m$. We see that the chessboard sign pattern of f_{ij} guarantees the resolution to HRA of each linear system and thus the computation of the coefficient vector d.

Since L_i, $i = 0, \dots, n$, and L_x are collocation matrices of Newton bases, Theorem 1 and Corollary 1 from [14] establish that the corresponding linear systems can be solved to HRA using either the bidiagonal factorization with TNSolve or recurrence (2.8), provided that all vectors $\boldsymbol{f}_i$ and $\boldsymbol{r}_j$ exhibit alternating signs and the nodes are ordered either increasingly or decreasingly.

Requiring the initial data f_{ij} to follow a chessboard sign pattern is sufficient to ensure that the alternating sign condition of $\boldsymbol{f}_i$ is met. When $y_{i0} \leq \cdots \leq y_{im}$, $i = 0, \dots, n$, Lemma 3.1 ensures that the columns $\boldsymbol{b}_j$ of the solution matrix B inherit the sign pattern of $(f_{0i}, \dots, f_{ni})$. Consequently, the entries b_{ij} of B also have a chessboard sign pattern. Thus, the $m + 1$ linear systems in Step 2 can be solved to

HRA for $x_0, \dots, x_n$ arranged in increasing or decreasing order, as the vectors $\boldsymbol{r}_j$ (the rows of D) also exhibit alternating signs.

On the other hand, when $y_{i0} \geq \cdots \geq y_{im}$, $i = 0, \dots, n$, each column of B maintains a constant sign (see the proof of Corollary 1 in [14]), while consecutive columns have the opposite sign. As a result, the rows of B have an alternating sign pattern, ensuring that the coefficients d_{ij} can also be determined to HRA. □

Finally, note that since Lemma 3.1 relies solely on the total positivity of the matrix involved, Theorem 3.2 can be readily extended to bivariate problems formulated using tensor product bases,

$$\{p_{i,j}^{(n,m)}(x,y) \mid i = 0, \dots, n; j = 0, \dots m\} = \{u_i^{(n)}(x)v_j^{(m)}(y) \mid i = 0, \dots, n; j = 0, \dots m\}, \tag{3.1}$$

provided that the corresponding collocation matrices of the univariate polynomial bases $(u_0^{(n)}, \dots, u_n^{(n)})$ and $(v_0^{(m)}, \dots, v_m^{(m)})$ are TP and their bidiagonal decompositions can be computed to HRA.

Theorem 3.3. *Let $(u_0^{(n)}, \dots, u_n^{(n)})$ and $(v_0^{(m)}, \dots, v_m^{(m)})$ be bases of P_n and P_m, the spaces of polynomials of a degree not greater than n and m, respectively. Consider the collocation matrix L_x of $(u_0^{(n)}, \dots, u_n^{(n)})$ at the nodes $x_0, \dots, x_n$ and the collocation matrices L_j of $(v_0^{(m)}, \dots, v_m^{(m)})$ at the nodes $y_{j0}, \dots, y_{jm}$, $j = 0, \dots, n$. Then, the coefficients of the interpolating polynomial*

$$p(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{m} c_{ij} p_{i,j}^{(n,m)}(x,y), \tag{3.2}$$

such that $p(x_i, y_{ij}) = f_{ij}$, $i = 0, \dots, n$, $j = 0, \dots, m$, can be computed to HRA if the interpolation data f_{ij} have a chessboard sign pattern, the matrices L_x, L_i, $i = 0, \dots, n$ are TP, and their bidiagonal factorization can be obtained to HRA.

4. Numerical experiments

This section presents numerical experiments to demonstrate the accuracy of the proposed approaches. As previously mentioned, the bivariate interpolation problem in the Newton-type basis (2.2) can be reformulated as a linear system $\boldsymbol{L}\boldsymbol{d} = \boldsymbol{b}$ of $(n+1)(m+1)$ dimension. For the numerical experiments, we computed solutions for system dimensions ranging from 50 to 125000, using various layouts. Following Algorithm 2.1, we analyzed the performance of the proposed methods in handling ill-conditioned matrices and compared them with standard methods.

Recall that Algorithm 2.1 reduces the initial bivariate problem to solving $n + m + 2$ univariate problems, which can be approached in different ways. Three possible methods for solving these systems are presented below.

(1) **Bidiagonal approach**: This method uses the bidiagonal decomposition obtained to HRA as input for the `TNSolve` subroutine, part of the `TNTool` package provided by Koev [15].
(2) **Recursive divided differences**: Since each univariate problem is also an interpolation problem in the Newton basis, the recursive formula for divided differences (2.8) provides an alternative approach. The conditions for achieving HRA and the performance of these two methods are discussed in detail in [14].

(3) **General-purpose solvers**: A standard routine for solving linear systems, such as Matlab's \ operator, can also be applied.

These methods have been compared to evaluate their efficiency and suitability for the problem at hand. It is worth mentioning that for the first two alternatives, the cost of Algorithm 2.1 is $O(nm(n+m))$, since Steps 1 and 2 have $O(nm^2)$ and $O(n^2m)$ operations, respectively. This is to be compared with, e.g., $O(nm(n^2+m^2))$ for Gauss elimination, since solving a $n \times n$ linear system has a computational cost of $O(n^3)$. Additionally, to give a reference for the computer time spent in the experiments, the largest mesh took less than 10 seconds in a standard desktop system (Intel i5-12400F).

As the exact solution of the linear systems considered, we take the results provided by Wolfram Mathematica 13.3 using 100-digit arithmetic. Then, for the exact coefficient vector $\boldsymbol{d}$, the relative error has been calculated as $e := \|\boldsymbol{d} - \tilde{\boldsymbol{d}}\|_2 / \|\boldsymbol{d}\|_2$, where $\tilde{\boldsymbol{d}}$ is the approximation obtained using the proposed methods.

First, we address rectangular lattices (x_i, y_j), $i = 0, \ldots, n$, $j = 0, \ldots, m$, chosen to be equidistant in a unit-length square. Secondly, Padua points, as an example of a nonrectangular grid (x_i, y_{ij}), $i = 0, \ldots, n$, $j = 0, \ldots, m$ are considered. This points are given for even values of n by

$$x_i = \cos\left(\frac{i\pi}{n}\right), \quad y_{ij} = \begin{cases} \cos\left(\frac{(2j+1)\pi}{n+1}\right), & \text{even } i, \\ \cos\left(\frac{2j\pi}{n+1}\right), & \text{odd } i, \end{cases}$$

for $i = 0, \ldots, n$, $j = 0, \ldots, n/2$. Moreover, the interpolation data f_{ij}, $i = 0, \ldots, n$, $j = 0, \ldots, m$, were chosen to be random integers with a uniform distribution in $[0, 10^4]$, adding a chessboard sign pattern $(-1)^{i+j}$ to guarantee HRA when obtaining the coefficients using the divided difference recurrence (2.8) and with `TNSolve`.

Relative errors are shown in Table 1 for rectangular equidistant ordered nodes, whereas the results when considering Padua points are depicted in Table 2. It should be noted that the size of the considered systems is limited by the bound imposed by double precision floating numbers ($\approx 1.8e+308$). In other words, for the biggest grids analyzed, the maximum value of the coefficients is near this limit; for the \ command, this limit is reached earlier (– symbol in tables), due to its loss of accuracy. In both cases, the results clearly illustrate the HRA achieved with the divided difference recurrence (2.8) and the function `TNSolve` applied to the bidiagonal decompositions considered in this work, supporting the theoretical findings outlined in the previous section.

Table 1. Relative errors of the approximations to the solution of the linear system $\boldsymbol{Ld} = \boldsymbol{b}$ for equidistant ordered nodes in a rectangular mesh.

n	m	\ command	Divided difference	`TNSolve`
10	10	$4.4e-14$	$3.6e-16$	$3.3e-16$
20	10	$3.8e-12$	$4.5e-16$	$3.6e-16$
50	10	$3.4e-03$	$8.1e-16$	$2.1e-15$
100	10	$3.4e+12$	$8.9e-16$	$3.6e-15$
20	20	$2.6e-12$	$4.7e-16$	$1.5e-15$
50	20	$2.7e-03$	$6.4e-16$	$2.9e-15$
40	40	$2.1e-06$	$9.5e-16$	$2.2e-15$
50	50	$3.5e-03$	$1.4e-15$	$4.0e-15$
100	100	$1.1e+25$	$1.2e-15$	$6.3e-15$
200	100	$1.0e+81$	$1.8e-15$	$1.9e-14$
200	150	–	$2.2e-15$	$1.7e-14$
200	200	–	$3.7e-15$	$3.2e-14$
300	100	–	$2.4e-15$	$3.8e-14$

Table 2. Relative errors of the approximations to the solution of the linear system $\boldsymbol{Ld} = \boldsymbol{b}$ for Padua points.

n	m	\ command	Divided difference	`TNSolve`
10	5	$3.4e-14$	$1.4e-15$	$1.1e-15$
20	10	$2.2e-10$	$3.8e-15$	$2.9e-15$
30	15	$9.0e-07$	$2.7e-15$	$2.6e-15$
40	20	$1.6e-02$	$9.7e-15$	$1.0e-14$
50	25	$8.7e+01$	$1.4e-14$	$1.5e-14$
70	35	$1.3e+10$	$2.1e-14$	$1.9e-14$
100	50	$1.5e+25$	$3.8e-14$	$4.0e-14$
200	100	$2.6e+117$	$1.0e-13$	$9.3e-14$
300	150	-	$7.5e-14$	$7.8e-14$
400	200	-	$2.5e-13$	$2.5e-13$
500	250	-	$1.7e-13$	$1.4e-13$

5. Conclusions

This work examines the bivariate polynomial interpolation problem formulated in the Newton basis. The methods introduced in [14] for the univariate case are extended using the generalized Kronecker product framework from [21]. Notably, the classical two-dimensional divided differences formula for rectangular grids is shown to be equivalent to Algorithm 2.1 presented in [21], and a generalization to nonrectangular grids of the form (x_i, y_{ij}), with $i = 0, \ldots, n$ and $j = 0, \ldots, m$, is provided. Sufficient conditions for solving the problem with high relative accuracy are established, as demonstrated through numerical examples on both rectangular and nonrectangular meshes.

Author contributions

Yasmina Khiar: Conceptualization, methodology, investigation, writing—original draft, writing—review and editing. Esmeralda Mainar: Conceptualization, methodology, investigation, writing—original draft, writing—review and editing. Eduardo Royo-Amondarain: Conceptualization, methodology, investigation, writing—original draft, writing—review and editing. Beatriz Rubio: Conceptualization, methodology, investigation, writing—original draft, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors are thankful for the useful suggestions received by the anonymous referees, which have improved the paper. This work was partially supported by Spanish research grants PID2022-138569NB-I00 (MCI/AEI) and RED2022-134176-T (MCI/AEI), and by Gobierno de Aragón (E41_23R).

Conflict of interest

Esmeralda Mainar is the Guest Editor of special issue "Advances in Numerical Linear Algebra: Theory and Methods" for AIMS Mathematics. Esmeralda Mainar was not involved in the editorial review and the decision to publish this article.

The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. L. Bos, S. De Marchi, S. Waldron, On the Vandermonde determinant of Padua-like points, *Dolomites Research Notes Approx.*, **2** (2009), 1–14. https://doi.org/10.14658/pupj-drna-2009-1-1

2. M. Caliari, S. De Marchi, M. Vianello, Bivariate polynomial interpolation on the square at new nodal sets, *Appl. Math. Comput.*, **165** (2005), 261–274. https://doi.org/10.1016/j.amc.2004.07.001

3. J. Delgado, H. Orera, J. M. Peña, Accurate computations with Laguerre matrices, *Numer. Linear Algebra Appl.*, **26** (2019), e2217. https://doi.org/10.1002/nla.2217

4. F. Dell'Accio, F. Di Tommaso, N. Siar, On the numerical computation of bivariate Lagrange polynomials, *Appl. Math. Lett.*, **112** (2021), 106845. https://doi.org/10.1016/j.aml.2020.106845

5. F. Dell'Accio, F. Marcellán, F. Nudo, An extension of a mixed interpolation–regression method using zeros of orthogonal polynomials, *J. Comput. Appl. Math.*, **450** (2024), 116010. https://doi.org/10.1016/j.cam.2024.116010

6. J. Demmel, Accurate singular value decompositions of structured matrices, *SIAM J. Matrix Anal. Appl.*, **21** (2000), 562–580. https://doi.org/10.1137/S0895479897328716

7. J. Demmel, P. Koev, The accurate and efficient solution of a totally positive generalized Vandermonde linear system, *SIAM J. Matrix Anal. Appl.*, **27** (2005), 142–152. https://doi.org/10.1137/S0895479804440335

8. S. Erol, B. Erol, A comparative assessment of different interpolation algorithms for prediction of gnss/levelling geoid surface using scattered control data, *Measurement*, **173** (2021), 108623. https://doi.org/10.1016/j.measurement.2020.108623

9. M. Fan, X. Zuo, B. Zhou, Parallel computing method of commonly used interpolation algorithms for remote sensing images, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **17** (2024), 315–322. https://doi.org/10.1109/jstars.2023.3329018

10. M. S. Floater, Polynomial interpolation on interlacing rectangular grids, *J. Approx. Theory*, **222** (2017), 64–73. https://doi.org/10.1016/j.jat.2017.06.002.

11. M. Gasca, J. J. Martínez, On the solvability of bivariate Hermite-Birkhoff interpolation problems, *J. Comput. Appl. Math.*, **32** (1990), 77–82. https://doi.org/10.1016/0377-0427(90)90418-y

12. M. Gasca, T. Sauer, Polynomial interpolation in several variables, *Adv. Comput. Math.*, **12** (2000), 377–410. https://doi.org/10.1023/a:1018981505752

13. E. Isaacson, H. B. Keller, *Analysis of numerical methods*, Courier Corporation, 2012.

14. Y. Khiar, E. Mainar, E. Royo-Amondarain, B. Rubio, On the accurate computation of the Newton form of the Lagrange interpolant, *Numer. Algor.*, **98** (2025), 1553–1573. https://doi.org/10.1007/s11075-024-01843-7

15. P. Koev, TNTool software package for performing virtually all matrix computations with nonsingular totally nonnegative matrices to high relative accuracy. Available from: `https://sites.google.com/sjsu.edu/plamenkoev/home/software/tntool`.

16. P. Koev, Accurate computations with totally nonnegative matrices, *SIAM J. Matrix Anal. Appl.*, **29** (2007), 731–751. https://doi.org/10.1137/04061903x

17. E. Mainar, J. M. Peña, B. Rubio, Accurate and efficient computations with Wronskian matrices of Bernstein and related bases, *Numer. Linear Algebra Appl.*, **29** (2022), e2423. https://doi.org/10.1002/nla.2423

18. E. Mainar, J. M. Peña, B. Rubio, Accurate computations with matrices related to bases $\{t^i e^{\lambda t}\}$, *Adv. Comput. Math.*, **48** (2022), 38. https://doi.org/10.1007/s10444-022-09954-2

19. A. Marco, J. J. Martínez, Accurate computations with totally positive Bernstein-Vandermonde matrices, *Electron. J. Linear Algebra*, **26** (2013), 357–380. https://doi.org/10.13001/1081-3810.1658

20. A. Marco, J. J. Martínez, R. Viaña, Accurate polynomial interpolation by using the Bernstein basis, *Numer. Algor.*, **75** (2017), 655–674. https://doi.org/10.1007/s11075-016-0215-7

21. J. J. Martínez, A generalized Kronecker product and linear systems, *Int. J. Math. Educ. Sci. Technol.*, **30** (1999), 137–141. https://doi.org/10.1080/002073999288193

22. C. R. Morrow, T. N. L. Patterson, Construction of algebraic cubature rules using polynomial ideal theory, *SIAM J. Numer. Anal.*, **15** (1978), 953–976. https://doi.org/10.1137/0715062

23. Z. Yang, R. Huang, W. Zhu, J. Liu, Accurate solutions of structured generalized Kronecker product linear systems, *Numer. Algor.*, **87** (2021), 797–818. https://doi.org/10.1007/s11075-020-00988-5