Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/100409

Transformations between Two-Variable Polynomial Bases with Applications

Dimitris Varsamis[1,*], *Nicholas Karampetakis*[2] *and Paris Mastorocostas*[1]

[1] Department of Informatics & Communications, Technological Educational Institute of Central Macedonia, 62124 Serres, Greece
[2] Department of Mathematics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

Received: 22 Mar. 2016, Revised: 8 May 2016, Accepted: 9 May 2016
Published online: 1 Jul. 2016

Abstract: It is well known, that any interpolating polynomial $p(x,y)$ on the vector space $P_{n,m}$ of two-variable polynomials with degree less than n in terms of x and less than m in terms of y, has various representations that depends on the basis of $P_{n,m}$ that we select i.e. monomial, Newton and Lagrange basis e.t.c.. The aim of this short note is twofold : a) to present transformations between the coordinates of the polynomial $p(x,y)$ in the aforementioned basis and b) to present transformations between these bases. Additionally, a computational numerical application that illustrate the usefulness of the transformations between two-variable polynomial bases is presented.

Keywords: Bivariate interpolation polynomial, Basis change, Transformations

1 Introduction

Interpolation is the problem of approximating a function f with another function p more usable, when its values at distinct points are known. When the function p is a polynomial we call the method *polynomial interpolation*. In case where the interpolating polynomial $p(x)$ belongs to the vector space P_n of single variable polynomials with degree less than n, $p(x)$ has various representations that depends on the basis of P_n that we select i.e. monomial, Newton and Lagrange basis e.t.c.. We can use coordinates relative to a basis to reveal the relationships between various forms of the interpolating polynomial. [1] shows how to change the form of the interpolating polynomial by transforming coordinates via a change of basis matrix. Moreover, [2] shows the transformations between the basis functions which map a specific representation to another. Additional work on this topic, from the numerical point of view, someone can find in [3], [4], [5]. In Sections 2 and 3, we are trying to extend the results of [1] and [2] to the case of two-variable interpolating polynomials with specific upper bounds in each variable. In Section 4, we make use of the transformations described in previous sections for the computation of the determinant of two-variable polynomial matrix. The proposed algorithm of the computation is based on the evaluation-interpolation technique. A numerical approach for the computation of these transforming matrices is also given.

2 Representations of the interpolating two-variable polynomial

Although the one-variable interpolation always has a solution for given *distinct* points, the multivariate interpolation problem through arbitrary given points may or may not have a solution when the number of unknown polynomial coefficients agree with the number of points. An interpolation problem is defined to be *poised* if it has a unique solution. Unlike the one-variable interpolation problem, the Hermite, Lagrange and Newton-form multivariate interpolation problem is not always poised. Let the set of interpolation points

$$S_{\Delta}^{(n,m)} = \left\{(x_i,y_j) \mid i=0,1,\dots,n,\ j=0,1,\dots,m\right\}$$

where $x_i \neq x_j$ and $y_i \neq y_j$ with function values on that points given by $f_{i,j} := f(x_i,y_j)$. Consider also the matrix

* Corresponding author e-mail: dvarsam@teicm.gr

$F \in \mathbb{R}^{(n+1)\times(m+1)}$ that is constructed from such values i.e.

$$F = \begin{bmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,m} \\ f_{1,0} & f_{1,1} & \cdots & f_{1,m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n,0} & f_{n,1} & \cdots & f_{n,m} \end{bmatrix} \tag{1}$$

It is well known [6], that for the specific selection of points $S_{\Delta}^{(n,m)}$ there exists a unique two-variable polynomial $p_{n,m}(x,y)$ on $P_{n,m}$ which interpolates these values i.e. $p_{n,m}(x_i,y_j) \equiv f(x_i,y_j) =: f_{i,j}$ and thus the interpolation problem is poised. This polynomial can be represented as a matrix product i.e.

$$p_{n,m}(x,y) = X^T \cdot A \cdot Y \tag{2}$$

where $X \in \mathbb{R}[x]^{(n+1)\times 1}$ (resp. $Y \in \mathbb{R}[y]^{(m+1)\times 1}$) are vectors that depends on the basis that we use (monomial, Lagrange, Newton) in terms of x (resp. in terms of y) and $A \in \mathbb{R}^{(n+1)\times(m+1)}$ is a two-dimensional matrix with elements the coefficients or otherwise the coordinates of the terms in the respective two-variable basis. (2) can be written as a Kronecker product i.e.

$$\begin{aligned} p_{n,m}(x,y) &= \left(Y^T \otimes X^T\right) \cdot vec(A) = (Y \otimes X)^T \cdot vec(A) = \\ &= vec(A)^T \cdot (Y \otimes X) = vec(A)^T \cdot g(x,y) \end{aligned} \tag{3}$$

where $(\otimes)$ is the Kronecker product and $vec(A)$ is the vectorization of a matrix, namely, is a linear transformation which converts the matrix into a column vector.

2.1 Monomial basis

The interpolating polynomial $p_{n,m}(x,y)$ in terms of the monomial basis is written as

$$p_{n,m}(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,j}x^i y^j = \mathbb{X}^T \cdot A \cdot \mathbb{Y} \tag{4}$$

where $\mathbb{X} = \begin{bmatrix} 1 \; x \cdots x^n \end{bmatrix}^T$, $\mathbb{Y} = \begin{bmatrix} 1 \; y \cdots y^m \end{bmatrix}^T$ and $A \in \mathbb{R}^{(n+1)\times(m+1)}$. By taking the relation $p_{n,m}(x_i,y_j) \equiv f(x_i,y_j)$ at all the interpolation points we can easily get the following relation

$$F = V_x \cdot A \cdot V_y^T \tag{5}$$

where

$$V_x = \begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & \cdots & x_{n-1}^n \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \; ; \; V_y = \begin{bmatrix} 1 & y_0 & \cdots & y_0^m \\ 1 & y_1 & \cdots & y_1^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & y_{m-1} & \cdots & y_{m-1}^m \\ 1 & y_m & \cdots & y_m^m \end{bmatrix}$$

with V_x (resp. V_y) the Vandermonde matrix with respect to x (resp. to y). It is easily seen that the matrix A is unique and it is easily computed in case where the Vandermonde matrices are nonsingular or otherwise the interpolation points x_i (resp. y_j) are different each other and the solution is given by

$$A = V_x^{-1} \cdot F \cdot V_y^{-T}$$

However, the computation of the inverse of a Vandermonde matrix is ill conditioned and standard numerically stable methods in general fail to accurately compute the entries of the inverse [7], [8], [9]. For this reason we may split (5) into the following system of Vandermonde equations i.e.

$$V_x \cdot A_1 = P \; ; \; A \cdot V_y^T = A_1$$

with unknowns A_1, A and solve it by using LU or QR decomposition. According to (3), the polynomial $p_{n,m}(x,y)$ is written as

$$\begin{aligned} p_{n,m}(x,y) &= \mathbb{X}^T \cdot A \cdot \mathbb{Y} \\ &= vec(A)^T \cdot (\mathbb{Y} \otimes \mathbb{X}) \\ &= vec(A)^T \cdot m(x,y) \end{aligned}$$

where

$$m(x,y) = (\mathbb{Y} \otimes \mathbb{X}) = \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^n \\ y \\ xy \\ x^2y \\ \vdots \\ x^ny \\ \vdots \\ y^m \\ xy^m \\ \vdots \\ x^ny^m \end{bmatrix}$$

is the two-variable monomial basis and

$$vec(A) = \begin{bmatrix} a_{0,0} \\ a_{1,0} \\ \vdots \\ a_{n,0} \\ a_{0,1} \\ a_{1,1} \\ \vdots \\ a_{n,1} \\ \vdots \\ a_{0,m} \\ a_{1,m} \\ \vdots \\ a_{n,m} \end{bmatrix}$$

Additionally, (5) can be rewritten as

$$vec(F) = (V_y \otimes V_x) \cdot vec(A) \tag{6}$$

Note that $vec(A)$ are the coordinates of $p_{n,m}(x,y)$ in terms of the monomial basis, whereas as we shall see below $vec(F)$ are the coordinates of $p_{n,m}(x,y)$ in terms of the Lagrange basis.

2.2 Lagrange basis

Similar results with the monomial basis are also applied to the Lagrange basis. The interpolating polynomial $p_{n,m}(x,y)$ in terms of the Lagrange basis is written as

$$p_{n,m}(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} f_{i,j} L_{i,n}(x) L_{m,j}(y) = \mathbb{X}_L^T \cdot F \cdot \mathbb{Y}_L \tag{7}$$

where

$$\mathbb{X}_L^T = \left[L_{0,n}(x) \; L_{1,n}(x) \; \cdots \; L_{n,n}(x) \right]$$

$$\mathbb{Y}_L = \left[L_{m,0}(y) \; L_{m,1}(y) \; \cdots \; L_{m,m}(y) \right]^T$$

with

$$L_{i,n}(x) = \prod_{\substack{k=0 \\ k\neq i}}^{n} \frac{(x-x_k)}{(x_i - x_k)} \text{ for } i = 0,1,..,n$$

$$L_{m,j}(y) = \prod_{\substack{k=0 \\ k\neq j}}^{m} \frac{(y-y_k)}{(y_j - y_k)} \text{ for } j = 0,1,...,m$$

and F defined in (1). For the Lagrange basis in two-variable polynomials see [6], [10], [11] and the references therein. According to (3), $p_{n,m}(x,y)$ can be written as

$$\begin{aligned} p_{n,m}(x,y) &= \mathbb{X}_L^T \cdot F \cdot \mathbb{Y}_L \\ &= vec(F)^T \cdot (\mathbb{Y}_L \otimes \mathbb{X}_L) \\ &= vec(F)^T \cdot \ell(x,y) \end{aligned}$$

in terms of the Lagrange basis

$$\ell(x,y) = \mathbb{Y}_L \otimes \mathbb{X}_L = \begin{bmatrix} L_{0,n}(x) \cdot L_{m,0}(y) \\ L_{1,n}(x) \cdot L_{m,0}(y) \\ \vdots \\ L_{n,n}(x) \cdot L_{m,0}(y) \\ L_{0,n}(x) \cdot L_{m,1}(y) \\ L_{1,n}(x) \cdot L_{m,1}(y) \\ \vdots \\ L_{n,n}(x) \cdot L_{m,m}(y) \end{bmatrix}$$

2.3 Newton basis

Another representation of $p_{n,m}(x,y)$ in terms of the Newton basis [6], [12] is the following

$$p_{n,m}(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} d_{i,j} \prod_{k=1}^{i}(x-x_{k-1}) \prod_{\ell=1}^{j}(y-y_{\ell-1}) = \mathbb{X}_N^T \cdot D \cdot \mathbb{Y}_N \tag{8}$$

where

$$\prod_{k=1}^{0}(x-x_{k-1}) \triangleq 1 \quad \text{and} \quad \prod_{\ell=1}^{0}(y-y_{\ell-1}) \triangleq 1$$

$$\mathbb{X}_N = \begin{bmatrix} 1 \\ x-x_0 \\ (x-x_0)(x-x_1) \\ \vdots \\ (x-x_0)(x-x_1)\cdots(x-x_{n-1}) \end{bmatrix}$$

$$\mathbb{Y}_N = \begin{bmatrix} 1 \\ y-y_0 \\ (y-y_0)(y-y_1) \\ \vdots \\ (y-y_0)(y-y_1)\cdots(y-y_{m-1}) \end{bmatrix}$$

and D is the coefficient matrix of Newton basis given by

$$D = \begin{bmatrix} d_{0,0} & d_{0,1} & \cdots & d_{0,m-1} & d_{0,m} \\ d_{1,0} & d_{1,1} & \cdots & d_{1,m-1} & d_{1,m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ d_{n-1,0} & d_{n-1,1} & \cdots & d_{n-1,m-1} & d_{n-1,m} \\ d_{n,0} & d_{n,1} & \cdots & d_{n,m-1} & d_{n,m} \end{bmatrix}$$

By taking the relation $p_{n,m}(x_i,y_j) \equiv f(x_i,y_j)$ at all the interpolation points we get

$$F = N_x^T \cdot D \cdot N_y \tag{9}$$

where

$$N_z = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & z_1 - z_0 & z_2 - z_0 & \cdots & z_n - z_0 \\ 0 & 0 & (z_2-z_0)(z_2-z_1) & \cdots & (z_n-z_0)(z_n-z_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \prod_{j=0}^{n-1}(z_n - z_j) \end{bmatrix}$$

and $z \in \{x,y\}$. The matrix D is unique since the matrices N_x and N_y are nonsingular ($x_i \neq x_j$ and $y_i \neq y_j$) and can be easily computed by

$$D = N_x^{-T} \cdot F \cdot N_y^{-1}$$

or similar to the monomial case by solving the system of equations $N_x^T \cdot D_1 = P$ and $D \cdot N_y = D_1$ with unknowns D_1

and D respectively. An alternative way to compute the coefficients of D is by means of the divided differences

$$d_{i,j}^{(k)} := \begin{cases} \dfrac{d_{i,j}^{(k-1)} - d_{i-1,j}^{(k-1)}}{x_i - x_{i-k}} & if \ (j < k \wedge i \geq k) \\ \dfrac{d_{i,j}^{(k-1)} - d_{i,j-1}^{(k-1)}}{y_j - y_{j-k}} & if \ (i < k \wedge j \geq k) \\ \dfrac{d_{i,j}^{(k-1)} + d_{i-1,j-1}^{(k-1)} - d_{i-1,j}^{(k-1)} - d_{i,j-1}^{(k-1)}}{(x_i - x_{i-k})(y_j - y_{j-k})} & if \ (i \geq k \wedge j \geq k) \\ d_{i,j}^{(k-1)} & if \ (i < k \wedge j < k) \end{cases}$$

which are defined in [12]. The polynomial $p_{n,m}(x,y)$ is written as

$$\begin{aligned} p_{n,m}(x,y) &= \mathbb{X}_N^T \cdot D \cdot \mathbb{Y}_N \\ &= vec(D)^T \cdot (\mathbb{Y}_N \otimes \mathbb{X}_N) \\ &= vec(D)^T \cdot n(x,y) \end{aligned}$$

in terms of the Newton basis

$$n(x,y) = \mathbb{Y}_N \otimes \mathbb{X}_N = \begin{bmatrix} 1 \\ x - x_0 \\ \vdots \\ \prod_{i=0}^{n-1}(x - x_i) \\ (y - y_0) \\ (y - y_0)(x - x_0) \\ \vdots \\ \prod_{i=0}^{n-1}(x - x_i) \prod_{j=0}^{m-1}(y - y_i) \end{bmatrix}$$

By using (9), we conclude that the coordinates $vec(D)$ of $p_{n,m}(x,y)$ in terms of the Newton basis are connected with the respective coordinates $vec(F)$ in terms of the Lagrange basis by

$$vec(F) = (N_y \otimes N_x)^T \cdot vec(D) \tag{10}$$

3 Change of basis in polynomial interpolation

As we show in the previous section, the interpolating polynomial $p_{n,m}(x,y)$ can be represented in one of the following ways

$$p_{n,m}(x,y) = \mathbb{X}^T \cdot A \cdot \mathbb{Y} = \mathbb{X}_L^T \cdot F \cdot \mathbb{Y}_L = \mathbb{X}_N^T \cdot D \cdot \mathbb{Y}_N \tag{11}$$

or equivalently

$$\begin{aligned} p_{n,m}(x,y) &= vec(A)^T \cdot m(x,y) \\ &= vec(F)^T \cdot \ell(x,y) \\ &= vec(D)^T \cdot n(x,y) \end{aligned} \tag{12}$$

From (12) we have

$$\begin{aligned} & vec(A)^T \cdot m(x,y) = vec(F)^T \cdot \ell(x,y) \overset{(6)}{\Longrightarrow} \\ & \Longrightarrow vec(A)^T \cdot m(x,y) = ((V_y \otimes V_x) \cdot vec(A))^T \cdot \ell(x,y) \\ & \Longrightarrow vec(A)^T \cdot m(x,y) = vec(A)^T \cdot (V_y \otimes V_x)^T \cdot \ell(x,y) \\ & \Longrightarrow m(x,y) = (V_y \otimes V_x)^T \cdot \ell(x,y) \\ & \Longrightarrow m(x,y) = \left(V_y^T \otimes V_x^T\right) \cdot \ell(x,y) = V_{xy}^T \cdot \ell(x,y) \end{aligned} \tag{13}$$

where V_{xy}^T is the transforming matrix between the coordinates of $p_{n,m}(x,y)$ in monomial and Lagrange base.

Similarly, from (12) we have

$$\begin{aligned} & vec(F)^T \cdot \ell(x,y) = vec(D)^T \cdot n(x,y) \overset{(10)}{\Longrightarrow} \\ & \Longrightarrow \left((N_y \otimes N_x)^T \cdot vec(D)\right)^T \cdot \ell(x,y) = vec(D)^T \cdot n(x,y) \\ & \Longrightarrow n(x,y) = (N_y \otimes N_x) \cdot \ell(x,y) = N_{xy} \cdot \ell(x,y) \end{aligned} \tag{14}$$

where N_{xy} is the transforming matrix between the coordinates of $p_{n,m}(x,y)$ in Newton and Lagrange base.

Since, the i-th element of the monomial base $m(x,y)$ has the same degree with the respective element in the Newton base $n(x,y)$ there exist a lower triangular matrix L such that

$$L \cdot n(x,y) = m(x,y)$$

From (13) and (14) we get

$$\left.\begin{aligned} m(x,y) = V_{xy}^T \cdot \ell(x,y) \\ n(x,y) = N_{xy} \cdot \ell(x,y) \end{aligned}\right\} \Longrightarrow$$

$$m(x,y) = V_{xy}^T \cdot N_{xy}^{-1} \cdot n(x,y) \equiv L \cdot n(x,y)$$

and therefore $L = V_{xy}^T \cdot N_{xy}^{-1}$ or equivalently

$$V_{xy}^T = L \cdot N_{xy} \tag{15}$$

Since, N_{xy} is upper triangular and L is lower triangular, (15) is a LU-decomposition of V_{xy}^T. Note also that

$$\begin{aligned} L &= V_{xy}^T \cdot N_{xy}^{-1} = (V_y \otimes V_x)^T \cdot (N_y \otimes N_x)^{-1} = \\ &= \left(V_y^T \otimes V_x^T\right) \cdot \left(N_y^{-1} \otimes N_x^{-1}\right) = \left(V_y^T \cdot N_y^{-1}\right) \otimes \left(V_x^T \cdot N_x^{-1}\right) \end{aligned} \tag{16}$$

According to [2]

$$L_y = V_y^T N_y^{-1} \text{ and } L_x = V_x^T N_x^{-1} \tag{17}$$

where

$$L_z := \begin{bmatrix} 1 & & & & \\ H_1(z_0) & 1 & & & \\ H_2(z_0) & H_1(z_0, z_1) & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ H_n(z_0) & H_{n-1}(z_0, z_1) & \cdots & H_1(x_0, \ldots, z_{n-1}) & 1 \end{bmatrix}$$

with $H_p(z_0, \ldots, z_k)$ be the sum of all homogeneous products of degree p of the variables $z_0, \ldots, z_k$ and $z \in \{x, y\}$. From (16) and (17) we conclude that

$$L = L_y \otimes L_x$$

Since the diagonal elements of L are equal to 1, (15) is the standard LU-decomposition of V_{xy}^T. The above results gives rise to the following Theorem.

Theorem 1. *Let $V_{xy}^T = L_{xy} \cdot N_{xy}$ be the standard LU-decomposition of the transposed Kronecker product of the matrices V_y, V_x i.e. $V_{xy} = V_y \otimes V_x$. Then, $L_{xy} = L_y \otimes L_x$ maps the Newton polynomials to the monomials and $N_{xy} = N_y \otimes N_x$ maps the Lagrange polynomials to the Newton polynomials.*

Theorem 1, extends the results presented in [2] for the one variable case. All the transformations described above are summarized in Table 1.

Table 1: Transformation matrices

Map	Basis transform	Coefficients transform
Lagrange to Monomial	$m(x,y) = V_{xy}^T \cdot \ell(x,y)$	$V_x \cdot A \cdot V_y^T = F$
Lagrange to Newton	$n(x,y) = N_{xy} \cdot \ell(x,y)$	$N_x^T \cdot D \cdot N_y = F$
Newton to Monomial	$m(x,y) = L_{xy} \cdot n(x,y)$	$L_x^T \cdot A \cdot L_y = D$

4 On the computation of determinant of a polynomial matrix

The computation of the determinant and the inverse of a two-variable polynomial matrix are some problems in linear algebra which are solved with symbolic operations. However, their main disadvantage, is their complexity. In order to overcome these difficulties we may use other techniques such as interpolation methods. Some remarkable examples, but not the only ones of the use of interpolation techniques in linear algebra problems, are the computation of the inverse by [13] and [14], the calculation of the determinant by [15] and [12].
In this section we present the computation of the determinant of a two variable polynomial matrix which by using the Lagrange interpolation instead to Newton interpolation which is presented in [12]. For the computation of the determinant we use the *evaluation–interpolation* technique. According this technique, first, we compute the determinants of the polynomial matrix for specific values and afterwards we interpolate these values to find the determinant which is a two-variable polynomial. The interpolation step can be done with the bivariate Lagrange interpolation method. An application of the proposed transformations is to replace the Lagrange interpolation method with the corresponding transformation. The transformation helps us to avoid the computational cost and the symbolic operations of the Lagrange method with numerical linear algebra operations.

The computation of the determinant is described in the following algorithm.

Algorithm 1 *Let the two-variable polynomial matrix $B(x,y) \in \mathbb{R}^{\ell\times\ell}$ where $B = [b_{i,j}(x,y)]$.*
Step 1: *Calculate the upper bound of the degree of the determinant in term of each variable with the following formulas*

$$n = \min\left\{ \sum_{i=1}^{\ell}\left(\max_{1\le j\le \ell}\left\{\deg_x[b_{i,j}(x,y)]\right\}\right), \sum_{j=1}^{\ell}\left(\max_{1\le i\le \ell}\left\{\deg_x[b_{i,j}(x,y)]\right\}\right)\right\}$$

and

$$m = \min\left\{ \sum_{i=1}^{\ell}\left(\max_{1\le j\le \ell}\left\{\deg_y[b_{i,j}(x,y)]\right\}\right), \sum_{j=1}^{\ell}\left(\max_{1\le i\le \ell}\left\{\deg_y[b_{i,j}(x,y)]\right\}\right)\right\}$$

Step 2: *Evaluate the determinants at the specific set of points*

$$\tilde{S}_{\Delta}^{(n,m)} = \left\{(x_i, y_j) \mid i = 0,1,\dots,n\ ,\ j = 0,1,\dots,m\right\}$$

which are on a rectangular basis.
In this step we create the matrix F which is contains the Lagrange coefficients.
Step 3: *Interpolate the values at the set $\tilde{S}_{\Delta}^{(n,m)}$. The interpolating polynomial is the determinant of $B(x,y)$.*
In this step we transform the matrix F to the matrix A where matrix A is the monomial coefficient matrix. Must be calculate the matrices V_x and V_y.

For the calculation of the matrices V_x and V_y we select the same sequential of values for the x_i and y_j, i.e. $x_i = i$ where $i = 1,2,\dots,n$ and $y_j = j$ where $j = 1,2,\dots,m$. With this approach we need less operations because the matrices V_x and V_y have same elements.
The computation of matrix A is given by

$$A = V_x^{-1} \cdot F \cdot V_y^{-T} \tag{18}$$

Thus, we do not interpolate with the Lagrange method but we make matrix manipulations for the computation of the determinant of $B(x,y)$.

Example 1. Let the polynomial matrix

$$B(x,y) = \begin{bmatrix} -1 & 0 & x \\ 5 & 1 & -1 \\ 2 & 3xy & 2 \end{bmatrix}$$

Step 1: We compute the upper bound n (resp. m) on the degree of x (resp. y) in $p(x,y) = \det B(x,y)$.

$$n = \min\left\{ \sum_{i=1}^{3}\left(\max_{1\le j\le 3}\left\{\deg_x[b_{i,j}(x,y)]\right\}\right), \sum_{j=1}^{3}\left(\max_{1\le i\le 3}\left\{\deg_x[b_{i,j}(x,y)]\right\}\right)\right\}$$

$$n = \min\{1+0+1, 0+1+1\} = 2$$

and

$$m = \min\left\{ \sum_{i=1}^{m} \left(\max_{1\le j\le m} \{\deg_y[b_{i,j}(x,y)]\} \right), \sum_{j=1}^{m} \left(\max_{1\le i\le m} \{\deg_y[b_{i,j}(x,y)]\} \right) \right\}$$

$$m = \min\{0+0+1, 0+1+0\} = 1$$

Step 2: We evaluate the determinants at the set

$$\tilde{S}_{\Delta}^{(2,1)} = \{(x_i = i, y_j = j) \mid i = 0,1,2,\ j = 0,1\}$$

and we create the matrix F which is given by

$$F = \begin{bmatrix} \det(B(x_0,y_0)) = -2 & \det(B(x_0,y_1)) = -2 \\ \det(B(x_1,y_0)) = -4 & \det(B(x_1,y_1)) = 8 \\ \det(B(x_2,y_0)) = -6 & \det(B(x_2,y_1)) = 48 \end{bmatrix}$$

Step 3:Since $n > m$, we compute the matrix V_x where

$$V_x = \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}$$

and the matrix V_y is a sub-matrix of the matrix V_x, that is,

$$V_y = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Then, the matrix A is given by

$$A = V_x^{-1} \cdot F \cdot V_y^{-T} = \begin{bmatrix} -2 & 0 \\ -2 & -3 \\ 0 & 15 \end{bmatrix}$$

Thus, the interpolating polynomial is

$$p(x,y) = -2 - 2x - 3xy + 15x^2y$$

which is the determinant of the polynomial matrix $B(x,y)$.

The matrices V_x^{-1} and V_y^{-T} defined in (18) can be calculate with an alternative way. As in the papers [16,3], we will use the wrapped convolution

$$a \star b = \begin{pmatrix} c_0 & c_1 & \dots & c_{n+m-1} \end{pmatrix}$$

where

$$a = \begin{pmatrix} a_0 & a_1 & \dots & a_{n-1} \end{pmatrix}, \quad b = \begin{pmatrix} b_0 & b_1 & \dots & b_{m-1} \end{pmatrix}$$

with coordinates equal to

$$c_i = \sum_{k=0}^{i} a_k b_{i-k} \quad i = 0,1,\dots,n+m-1$$

where $a_k = 0$ if $k < 0$ or $k \ge n$ and $b_k = 0$ if $k < 0$ or $k \ge m$.
Let the matrices

$$X_k = \begin{bmatrix} -x_k & 1 \end{bmatrix} \quad \text{where} \quad k = 0,1,\dots n$$

and

$$Y_k = \begin{bmatrix} -y_k & 1 \end{bmatrix} \quad \text{where} \quad k = 0,1,\dots m$$

which are the corresponding coefficient matrices of the terms $(x - x_k)$ and $(y - y_k)$ respectively.
We define the coefficient transformations matrices

$$T_{L_x} = \begin{bmatrix} \dfrac{X_1 \star X_2 \star X_3 \cdots X_{n-1} \star X_n}{\Pi_0^x} \\ \dfrac{X_0 \star X_2 \star X_3 \cdots X_{n-1} \star X_n}{\Pi_1^x} \\ \dfrac{X_0 \star X_1 \star X_3 \cdots X_{n-1} \star X_n}{\Pi_2^x} \\ \vdots \\ \dfrac{X_0 \star X_1 \star X_2 \cdots X_{n-2} \star X_n}{\Pi_{n-1}^x} \\ \dfrac{X_0 \star X_1 \star X_2 \cdots X_{n-2} \star X_{n-1}}{\Pi_n^x} \end{bmatrix}$$

where

$$\Pi_i^x = \prod_{\substack{k=0 \\ k\neq i}}^{n} (x_i - x_k)$$

and

$$T_{L_y} = \begin{bmatrix} \dfrac{Y_1 \star Y_2 \star Y_3 \cdots Y_{n-1} \star Y_n}{\Pi_0^y} \\ \dfrac{Y_0 \star Y_2 \star Y_3 \cdots Y_{n-1} \star Y_n}{\Pi_1^y} \\ \dfrac{Y_0 \star Y_1 \star Y_3 \cdots Y_{n-1} \star Y_n}{\Pi_2^y} \\ \vdots \\ \dfrac{Y_0 \star Y_1 \star Y_2 \cdots Y_{n-2} \star Y_n}{\Pi_{n-1}^y} \\ \dfrac{Y_0 \star Y_1 \star Y_2 \cdots Y_{n-2} \star Y_{n-1}}{\Pi_n^y} \end{bmatrix}$$

where

$$\Pi_j^y = \prod_{\substack{k=0 \\ k\neq j}}^{n} (y_j - y_k)$$

In case where $x_k = x_0 + k \cdot h_x$ and $y_k = y_0 + k \cdot h_y$ with $x_0 = y_0 = 0$ and $h_x = h_y = 1$ we have $x_k = y_k = k$. Thus, the above products are given by

$$\Pi_i^x = \prod_{\substack{k=0 \\ k\neq i}}^{n} (i-k) \quad \text{and} \quad \Pi_j^y = \prod_{\substack{k=0 \\ k\neq j}}^{n} (j-k)$$

or equivalently

$$\Pi_i^x = (-1)^{n-i} \cdot (n-i)! \cdot i! \quad \text{and} \quad \Pi_j^y = (-1)^{n-j} \cdot (n-j)! \cdot j!$$

For the matrices $\mathbb{X}_L$, T_{L_x} and $\mathbb{X}$ holds that

$$\mathbb{X}_L = T_{L_x} \cdot \mathbb{X}$$

Similarly, for the respectively matrices $\mathbb{Y}_L$, T_{L_y} and $\mathbb{Y}$ holds that

$$\mathbb{Y}_L = T_{L_y} \cdot \mathbb{Y}$$

Thus, the polynomial can be written

$$p(x,y) = \mathbb{X}_L^T \cdot F \cdot \mathbb{Y}_L = \mathbb{X}^T \cdot T_{L_x}^T \cdot F \cdot T_{L_y} \cdot \mathbb{Y}$$

and we have that

$$A = T_{L_x}^T \cdot F \cdot T_{L_y}$$

Therefore, we show that

$$T_{L_x}^T \equiv V_x^{-1} \quad \text{and} \quad T_{L_y} \equiv V_y^{-T}$$

*Example 2.*According to Example 1 we have the set

$$\tilde{S}_\Delta^{(2,1)} = \left\{(x_i = i, y_j = j) \mid i = 0,1,2, \ j = 0,1\right\}$$

and the matrix F which is given by

$$F = \begin{bmatrix} -2 & -2 \\ -4 & 8 \\ -6 & 48 \end{bmatrix}$$

We compute the elements of the matrices T_{L_x} and T_{L_y}

$$\begin{gathered} X_1 \star X_2 = [2,-3,1] \\ X_0 \star X_2 = [0,-2,1] \\ X_0 \star X_1 = [0,-1,1] \\ \Pi_0^x = (-1)^{n-i} \cdot (n-i)! \cdot i! = (-1)^{2-0} \cdot (2-0)! \cdot 0! = 2 \\ \Pi_1^x = (-1)^{n-i} \cdot (n-i)! \cdot i! = (-1)^1 \cdot 1! \cdot 1! = -1 \\ \Pi_2^x = (-1)^{n-i} \cdot (n-i)! \cdot i! = (-1)^0 \cdot 0! \cdot 2! = 2 \end{gathered}$$

and

$$\begin{gathered} Y_1 = [-1,1] \\ Y_0 = [0,1] \\ \Pi_0^y = (-1)^{n-j} \cdot (n-j)! \cdot j! = (-1)^{1-0} \cdot (1-0)! \cdot 0! = -1 \\ \Pi_1^y = (-1)^{n-j} \cdot (n-j)! \cdot j! = (-1)^0 \cdot 0! \cdot 1! = 1 \end{gathered}$$

Thus, we have

$$T_{L_x} = \begin{bmatrix} \frac{X_1 \star X_2}{\Pi_0^x} \\ \frac{X_0 \star X_2}{\Pi_1^x} \\ \frac{X_0 \star X_1}{\Pi_2^x} \end{bmatrix} = \begin{bmatrix} 1 & -\frac{3}{2} & \frac{1}{2} \\ 0 & 2 & -1 \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

and

$$T_{L_y} = \begin{bmatrix} \frac{Y_1}{\Pi_0^y} \\ \frac{Y_0}{\Pi_1^y} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

Then, the matrix A is given by

$$A = T_{L_x}^T \cdot F \cdot T_{L_y} = \begin{bmatrix} -2 & 0 \\ -2 & -3 \\ 0 & 15 \end{bmatrix}$$

Thus, the interpolating polynomial is

$$p(x,y) = -2 - 2x - 3xy + 15x^2y$$

which is the determinant of the polynomial matrix $B(x,y)$.

5 Conclusions

The first result that comes directly from this short note is that in case where we select interpolation points that belongs to $S_\Delta^{(n,m)}$ the interpolating polynomial problem is poised, since in that case the transforming matrices that we use become nonsingular and a unique solution of the coordinate vectors exists. The second result, is that any interpolating polynomial is easily expressed in the Lagrange basis, since in that case the only we need are the values of the function that we want to interpolate. Then, by using the transformations that we have presented in this work we can always express the interpolating polynomial in other bases like the monomial and the Newton base. Additionally, transformations between the monomial, Lagrange and Newton bases have been provided and the results in [1] and [2] have been extended to the bivariate polynomials. Finally, an illustrative example has been presented which uses the transformations from the Lagrange to the monomial basis. A numerical algorithm for the computation of the transforming matrices used in this application, is also given.

Acknowledgement

The authors wish to acknowledge financial support provided by the Research Committee of the Serres Institute of Education and Technology under grant SAT/IC/07022013-15/5.

References

[1] D.R. Hill. Interpolating polynomials and their coordinates relative to a basis. *The College Mathematics Journal*, 23(4):329–333, 1992.
[2] W. Gander. Change of basis in polynomial interpolation. *Numerical Linear Algebra with Applications*, 12(8):769–778, 2005.
[3] J. Kapusta and R. Smarzewski. Fast algorithms for multivariate interpolation and evaluation at special points. *Journal of Complexity*, 25(4):332–338, 2009.

[4] J. Kapusta. An efficient algorithm for multivariate maclaurin-newton transformation. 8(2):5–14, 2008.

[5] R. Smarzewski and J. Kapusta. Fast lagrange–newton transformations. *Journal of Complexity*, 23(3):336–345, 2007.

[6] G. M. Phillips. *Interpolation and approximation by polynomials*. Springer-Verlag, 2003.

[7] E.E. Tyrtyshnikov. How bad are hankel matrices? *Numerische Mathematik*, 67(2):261–269, 1994.

[8] W. Gautschi and G. Inglese. Lower bounds for the condition number of vandermonde matrices. *Numerische Mathematik*, 52(3):241–250, 1987.

[9] A. Eisinberg and G. Fedele. On the inversion of the vandermonde matrix. *Applied mathematics and computation*, 174(2):1384–1397, 2006.

[10] T. Sauer and Y. Xu. On multivariate lagrange interpolation. *Mathematics of Computation*, 64:1147–1170, 1995.

[11] M. Gasca and T. Sauer. Polynomial interpolation in several variables. *Advances in Computational Mathematics*, 12:377–410, 2000.

[12] Dimitris N. Varsamis and Nicholas P. Karampetakis. On the Newton bivariate polynomial interpolation with applications. *Multidimensional Systems and Signal Processing*, 25(1):179–209, 2014.

[13] A. Schuster and P. Hippe. Inversion of polynomial matrices by interpolation. *IEEE Transactions on Automatic Control*, 37:363–365, 1992.

[14] S. Vologiannidis and N. P. Karampetakis. Inverses of multivariable polynomial matrices by discrete fourier transforms. *Multidimensional Systems and Signal Processing*, 15:341–361, 2004.

[15] L. E. Paccagnella and G. L. Pierobon. Fft calculation of a determinantal polynomial. *IEEE Transactions on Automatic Control*, 29(3):401–402, 1976.

[16] A. Bostan and É. Schost. Polynomial evaluation and interpolation on special sets of points. *Journal of Complexity*, 21(4):420–446, 2005.

Dimitris N. Varsamis was born in Serres, Greece in 1976. He received the bachelor degree in Mathematics from the Department of Mathematics, University of Ioannina, Greece and the MSc in “Theoretical computer science, control and systems theory” from the Department of Mathematics, Aristotle University of Thessaloniki, Greece. He was awarded the Ph.D. at the Department of Mathematics (Aristotle University of Thessaloniki, Greece) in 2012, under the supervision of Associate Professor Nicholas Karampetakis. The subject of his doctoral thesis is “On the Development of Computational Methods for the solution of Control Theory problems”. From 2001 to 2009 he was a scientific associate with Department of Informatics Communications, Technological Educational Institute of Serres, Greece. Since 2009, he has been with the Technological Educational Institute of Central Macedonia (Serres, Greece), where he is currently an Assistant Professor in the Dept. of Informatics Engineering. Dr. Varsamis is the co-author of 25 articles in international journals and conference proceedings.His research interests include Scientific Programming, Computational methods, Control System Theory and Numerical Analysis.

Nicholas P. Karampetakis was born in Drama, Greece in 1967. He received the bachelor degree in Mathematics and Ph.D. in Mathematics from the Department of Mathematics, Aristotle University of Thessaloniki, Greece in 1989 and 1993 respectively. From November 1994 to September 1995 he was a research associate with the Department of Mathematical Sciences, Loughborough University of Technology, England. From September 1995 to March 1999 he was a Research Associate with the Department of Mathematics, Aristotle University of Thessaloniki, Greece. From August 2000, to July 2009 he was an Assistant Professor to the same Department, whereas from July 2009, he has been Associate Professor. During the above periods he received many fellowships from the Greek Government, British Council and Engineering and Physical Sciences Research Council of England, and contributes in many research projects sponsored by the Greek Government and the European Union. His present research interests lie mainly in algebraic methods for computer aided design of control systems (CACSD), numerical and symbolic algorithms for CACSD, polynomial matrix theory and issues related to the mathematical systems theory. Dr. Karampetakis is a frequent contributor to the field in the form of journal articles, conference presentations and reviews. He is Senior Member of IEEE and a vice-Chair of the IEEE Action Group on Symbolic Methods for CACSD. He is also an Associate Editor a) of the Journal of Multidimensional Systems and Signal Processing and b) of the International Journal of Applied Mathematics and Computer Science.

Paris Mastorocostas was born in Thessaloniki, Greece, in 1971. He received the Diploma and Ph.D. degrees in Electrical & Computer Engineering from Aristotle University of Thessaloniki, Greece. Presently he serves as Professor at the Dept. of Informatics & Communications, Technological Education Institute of

Serres, Greece. His research interests include fuzzy and neural systems with applications to identification and classification processes, data mining and scientific programming.