PADOVA UNIVERSITY PRESS

Dolomites Research Notes on Approximation

Volume 8 · 2015 · Pages 1–11

A few remarks on "On certain Vandermonde determinants whose variables separate"

André Pierro de Camargo[a] · Stefano De Marchi[b]

Abstract

In the recent paper "On certain Vandermonde determinants whose variables separate" [*Linear Algebra and its Applications* 449 (2014) pp. 17–27], there was established a factorized formula for some bivariate Vandermonde determinants (associated to almost square grids) whose basis functions are formed by Hadamard products of some univariate polynomials. That formula was crucial for proving a conjecture on the Vandermonde determinant associated to Padua-like points. In this note we show that the same formula holds when those polynomials are replaced by arbitrary functions and we extend this formula to general rectangular grids. We also show that the Vandermonde determinants associated to Padua-like points are nonvanishing.

1 Introduction

The Padua points [4] are the first known unisolvent set of nodes for explicit full polynomial interpolation in $[-1,1]^2$ whose Lebesgue constant has the minimal order of growth $O(\log^2(n))$ [2]. In 2009, Bos, De Marchi and Waldron [3] conjectured a factorized formula for the Vandermonde determinant associated to the Padua points (and to other sets of points sharing the same structure, named the Padua-like points) and this conjecture was recently proved by De Marchi and Usevich [8]. The central result of [8] states that some bivariate Vandermonde determinants of size $m \times m$ or $m \times (m+1)$, $m \geq 1$, whose basis functions are formed by Hadamard products of some univariate polynomials split into products of univariate Vandermonde determinants, when evaluated at rectangular grids of size $m \times m$ or $m \times (m+1)$, respectively. One of the goals of this note is to generalize that result to more general basis functions and rectangular grids of arbitrary size.

The unisolvence property of the Padua points was established in [2] by exhibiting explicitly the Lagrange polynomials associated to corresponding interpolation problem. This is equivalent to showing that the Vandermonde determinant associated to the Padua points is nonvanishing, but [8] do not explain why the factors arising in the factorized formula for such Vandermonde determinant must be nonvanishing. In fact, the overall structure of these Vandermondians is still not completely understood. For instance, [8] mentions that some of those factors can be further factorized, but the resulting prime factors are not so simple to handle.

The other goal of this note is to establish the unisolvence property for all sets of Padua-like points by showing that the corresponding Vandermonde determinant is indeed nonvanishing.

1.1 Notation

Given a set of multivariate real functions $\mathcal{F} = \{f_1(z), f_2(z), \ldots, f_k(z)\}$, with $z \in \mathbb{R}^k$, and a set of points $\mathcal{A} = \{a_1, a_2, \ldots, a_k\} \subset \mathbb{R}^k$ we denote by $VDM(\mathcal{A}, \mathcal{F})$ the Vandermonde matrix $\left(f_j(a_i)\right)_{i=1,2\ldots,k}^{j=1,2\ldots,k}$ and by $vdm(\mathcal{A}, \mathcal{F}) = \det VDM(\mathcal{A}, \mathcal{F})$, its determinant. Notice that $VDM(\mathcal{A}, \mathcal{F})$ depends on the order of the elements of $\mathcal{F}$ and $\mathcal{A}$. However, since we are mainly concerned with the absolute value of $vdm(\mathcal{A}, \mathcal{F})$, then the order does not matter. Thus, if $\mathcal{F} \in \mathbb{R}^{m\times n}(z)$ is given as a function matrix and $\mathcal{A} \in \mathbb{R}^{m*n}$, for example, we will simply write $vdm(\mathcal{A}, \mathcal{F})$.

For a function matrix $\mathcal{F} \in \mathbb{R}^{m\times n}(z)$, we shall denote by $\mathcal{F}_{i,:} \in \mathbb{R}^n(z)$, $\mathcal{F}_{:,j} \in \mathbb{R}^m(z)$ the i-th row and the j-th columm of $\mathcal{F}$, respectively, and by $\mathcal{F}_{:,j:k} \in \mathbb{R}^{m\times(k-j+1)}$ the submatrix of $\mathcal{F}$ matrix formed from the j-th to the k-th columns of $\mathcal{F}$. Moreover, given function matrices $P, Q \in \mathbb{R}^{m\times n}(z)$, then $P \circ Q$ will denote their Hadamard (element-wise) product. Finally $diag(V) \in \mathbb{R}^{n\times n}$ will denote the diagonal matrix defined by $V \in \mathbb{R}^n$ and I_n will denote the identity matrix of order n.

2 On certain multivariate Vandermonde determinants whose variables separate

Our generalization of the central statement of [8] can be stated as follows

[a] Instituto de Matemática e Estatística, Universidade de São Paulo, Cidade Universitária, Rua do Matão 1010, São Paulo SP, Brazil. CEP 05508-090. Email: andrecamargo.math@gmail.com, supported by grant 14225012012-0 from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)

[b] University of Padova, Department of Mathematics, Via Trieste, 63, I-35121 PADOVA, Italy. Email: demarchi@math.unipd.it

Theorem 2.1. *Let $m,n\geq 1$ be positive integers, $k=\min\{m,n\}$, and $P\in\mathbb{R}^{m\times n}(x), Q\in\mathbb{R}^{m\times n}(y)$ be two real valued function matrices of the form*

$$Q(y)=\begin{bmatrix} q_0(y) & * & \dots & \dots & * & \dots & * \\ q_0(y) & q_1(y) & * & \dots & * & \dots & * \\ \vdots & \vdots & \ddots & \ddots & \vdots & & \vdots \\ q_0(y) & q_1(y) & \dots & q_{k-1}(y) & * & \dots & * \end{bmatrix}, P(x)=\begin{bmatrix} * & p_0(x) & p_0(x) & \dots & p_0(x) & \dots & p_0(x) \\ * & * & p_1(x) & \dots & p_1(x) & \dots & p_1(x) \\ \vdots & \vdots & \ddots & \ddots & \vdots & & \vdots \\ * & * & \dots & * & p_{k-1}(x) & \dots & p_{k-1}(x) \end{bmatrix}, \tag{1}$$

for $m<n$, or

$$Q(y)=\begin{bmatrix} q_0(y) & * & \dots & * \\ q_0(y) & q_1(y) & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ q_0(y) & q_1(y) & \dots & q_{k-1}(y) \\ \vdots & \vdots & & \vdots \\ q_0(y) & q_1(y) & \dots & q_{k-1}(y) \end{bmatrix}, \quad P(x)=\begin{bmatrix} * & p_0(x) & p_0(x) & \dots & p_0(x) \\ * & * & p_1(x) & \dots & p_1(x) \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ * & * & \dots & * & p_{k-2}(x) \\ \vdots & \vdots & & \vdots & \vdots \\ * & * & \dots & * & * \end{bmatrix}, \tag{2}$$

for $m\geq n$, where $q_0,q_1,\dots,q_{k-1},p_0,p_1,\dots,p_{k-1}$ are arbitrary functions and $*$ denote also arbitrary (not equal to each other) functions. Then for $X=(x_1,\dots,x_m)\in\mathbb{R}^m$ and $Y=(y_1,\dots,y_n)\in\mathbb{R}^n$, the following equality holds

$$vdm(X\times Y, P\circ Q)=\pm\left(\prod_{j=1}^{n} vdm(X,P_{:,j})\right)\left(\prod_{i=1}^{m} vdm(Y,Q_{i,:})\right). \tag{3}$$

In [8], m and n were assumed to be almost equal, that is $n=m$ or $n=m+1$. Moreover, $q_0,q_1,\dots,q_{k-1}$ and $p_0,p_1,\dots,p_{k-1}$ were assumed to be monic polynomials satisfying

$$deg(p_j) \;=\; deg(q_j) \;=\; j,\; j=1,2,\dots,m-1 \tag{4}$$

and, in particular,

$$q_0(y)\equiv 1 \quad\text{and}\quad p_0(x)\equiv 1. \tag{5}$$

Following Section 3.2 of [8], it turns out that Theorem 2.1 generalizes the factorization property (valid for arbitrary m and n) for rank-one matrices

$$Q(y)=\begin{bmatrix} q_0(y) & q_1(y) & \dots & q_n(y) \\ q_0(y) & q_1(y) & \dots & q_n(y) \\ \vdots & \vdots & & \vdots \\ q_0(y) & q_1(y) & \dots & q_n(y) \end{bmatrix} \quad\text{and}\quad P(x)=\begin{bmatrix} p_0(x) & p_0(x) & \dots & p_0(x) \\ p_1(x) & p_1(x) & \dots & p_1(x) \\ \vdots & \vdots & & \vdots \\ p_m(x) & p_m(x) & \dots & p_m(x) \end{bmatrix}, \tag{6}$$

for which (3) is readily obtained by the properties of the Kronecker product. Notice that, by definition of P and Q, when $|m-n|\geq 2$, then the right-hand side of (3) will have at least $|m-n|$ identical factors. However, our numerical experiments suggested that the presented generalization (Theorem 2.1) of the factorization property is maximal in the sense that its hypothesis can not be further relaxed.

2.1 Proof of Theorem 2.1

Our proof is by induction on $k=\min\{m,n\}$ and follows the same lines of the argument in [8]. It is easy to see that (3) holds for $k=1$. For instance, if $m=1$, then $X=(x_1)$, $Y=(y_1,y_2,\dots,y_n)$, $P(x)=\begin{bmatrix} u_1(x) & u_2(x) & \dots & u_n(x)\end{bmatrix}$ and $Q(y)=\begin{bmatrix} v_1(y) & v_2(y) & \dots & v_n(y)\end{bmatrix}$ are row vectors and $vdm(X\times Y,P\circ Q)=$

$$\pm\begin{vmatrix} u_1(x_1)v_1(y_1) & u_2(x_1)v_2(y_1) & \dots & u_n(x_1)v_n(y_1) \\ u_1(x_1)v_1(y_2) & u_2(x_1)v_2(y_2) & \dots & u_n(x_1)v_n(y_2) \\ \vdots & \vdots & & \vdots \\ u_1(x_1)v_1(y_n) & u_2(x_1)v_2(y_n) & \dots & u_n(x_1)v_n(y_n) \end{vmatrix} = \pm\left(\prod_{j=1}^{n} u_j(x_1)\right)\begin{vmatrix} v_1(y_1) & v_2(y_1) & \dots & v_n(y_1) \\ v_1(y_2) & v_2(y_2) & \dots & v_n(y_2) \\ \vdots & \vdots & & \vdots \\ v_1(y_n) & v_2(y_n) & \dots & v_n(y_n) \end{vmatrix}.$$

Assuming that (3) holds for $k-1 \geq 1$, we will prove it for k.

Put $N_j = vdm(X, P_{:,j})$. Following [8], we have $vdm(X \times Y, P \circ Q)$

$$
\begin{aligned}
&= \pm \det \begin{bmatrix} N_1\ diag(Q_{:,1}(y_1)) & N_2\ diag(Q_{:,2}(y_1)) & \dots & N_n\ diag(Q_{:,n}(y_1)) \\ N_1\ diag(Q_{:,1}(y_2)) & N_2\ diag(Q_{:,2}(y_2)) & \dots & N_n\ diag(Q_{:,n}(y_2)) \\ \vdots & \vdots & & \vdots \\ N_1\ diag(Q_{:,1}(y_n)) & N_2\ diag(Q_{:,2}(y_n)) & \dots & N_n\ diag(Q_{:,n}(y_n)) \end{bmatrix} \\
&= \pm \det \begin{bmatrix} q_0(y_1)N_1 & N_2\ diag(Q_{:,2}(y_1)) & \dots & N_n\ diag(Q_{:,n}(y_1)) \\ q_0(y_2)N_1 & N_2\ diag(Q_{:,2}(y_2)) & \dots & N_n\ diag(Q_{:,n}(y_2)) \\ \vdots & \vdots & & \vdots \\ q_0(y_n)N_1 & N_2\ diag(Q_{:,2}(y_n)) & \dots & N_n\ diag(Q_{:,n}(y_n)) \end{bmatrix} \\
&= \pm \det(N_1) \det \begin{bmatrix} q_0(y_1)I_m & N_2\ diag(Q_{:,2}(y_1)) & \dots & N_n\ diag(Q_{:,n}(y_1)) \\ q_0(y_2)I_m & N_2\ diag(Q_{:,2}(y_2)) & \dots & N_n\ diag(Q_{:,n}(y_2)) \\ \vdots & \vdots & & \vdots \\ q_0(y_n)I_m & N_2\ diag(Q_{:,2}(y_n)) & \dots & N_n\ diag(Q_{:,n}(y_n)) \end{bmatrix}.
\end{aligned}
$$

Let us assume, initially, that

$$q_0(y_1) \neq 0. \tag{7}$$

Then, by an obvious block elimination process, we obtain

$$vdm(X \times Y, P \circ Q) = \pm \det(N_1) q_0(y_1)^m\, M, \tag{8}$$

with

$$
\begin{aligned}
M &= \det \begin{bmatrix} N_2\ diag\left(Q_{:,2}(y_2) - \frac{q_0(y_2)}{q_0(y_1)} Q_{:,2}(y_1)\right) & \dots & N_n\ diag\left(Q_{:,n}(y_2) - \frac{q_0(y_2)}{q_0(y_1)} Q_{:,n}(y_1)\right) \\ \vdots & & \vdots \\ N_2\ diag\left(Q_{:,2}(y_n) - \frac{q_0(y_n)}{q_0(y_1)} Q_{:,2}(y_1)\right) & \dots & N_n\ diag\left(Q_{:,n}(y_n) - \frac{q_0(y_n)}{q_0(y_1)} Q_{:,n}(y_1)\right) \end{bmatrix} \\
&= \det \begin{bmatrix} N_2\ diag\left(\tilde{Q}_{:,1}(y_2)\right)\right) & \dots & N_n\ diag\left(\tilde{Q}_{:,n-1}(y_2)\right)\right) \\ \vdots & & \vdots \\ N_2\ diag\left(\tilde{Q}_{:,1}(y_n)\right)\right) & \dots & N_n\ diag\left(\tilde{Q}_{:,n-1}(y_n)\right)\right) \end{bmatrix} \\
&= vdm(X \times \tilde{Y}, P_{:,2:n} \circ \tilde{Q}),
\end{aligned}
$$

where $\tilde{Y} = (y_2, \dots, y_n)$ and

$$\tilde{Q}(y) := Q_{:,2:n}(y) - \frac{q_0(y)}{q_0(y_1)} Q_{:,2:n}(y_1). \tag{9}$$

- First case: $k = n$. In this case, we can apply the induction step to the tranpose matrices $\left(\tilde{Q}\right)^t$ and $\left(P_{:,2:n}\right)^t$ to obtain

$$vdm(X \times Y, P \circ Q) = \pm\ \det(N_1) q_0(y_1)^m \left(\prod_{j=2}^{n} vdm(X, P_{:,j})\right)\left(\prod_{i=1}^{m} vdm(\tilde{Y}, \tilde{Q}_{i,:})\right). \tag{10}$$

Then, notice that, for $Q_{i,:}(y) = \begin{bmatrix} q_0(y) & a_1(y) & \dots & a_{n-1}(y) \end{bmatrix}$, we have $vdm(Y, Q_{i,:}) =$

$$
\det \begin{bmatrix} q_0(y_1) & a_1(y_1) & \dots & a_{n-1}(y_1) \\ q_0(y_2) & a_1(y_2) & \dots & a_{n-1}(y_2) \\ \vdots & \vdots & & \vdots \\ q_0(y_n) & a_1(y_n) & \dots & a_{n-1}(y_n) \end{bmatrix} =
$$

$$
\begin{aligned}
& q_0(y_1) \det \begin{bmatrix} a_1(y_2) - \frac{q_0(y_2)}{q_0(y_1)} a_1(y_1) & \dots & a_{n-1}(y_2) - \frac{q_0(y_2)}{q_0(y_1)} a_{n-1}(y_1) \\ \vdots & & \vdots \\ a_1(y_n) - \frac{q_0(y_n)}{q_0(y_1)} a_1(y_1) & \dots & a_{n-1}(y_n) - \frac{q_0(y_n)}{q_0(y_1)} a_{n-1}(y_1) \end{bmatrix} \\
&= q_0(y_1) vdm(\tilde{Y}, \tilde{Q}_{i,:})
\end{aligned} \tag{11}
$$

and this completes the proof for case (7).

To prove the general case, notice that both sides of (3) vanish when $q_0(y_1)=q_0(y_2)=\ldots=q_0(y_n)=0$. On the other hand, if at least one of these values is different from zero, we fall into case (7) through a suitable permutation of the variables $y_1, y_2, \ldots, y_n$.

- Second case: $k=m$. In this case, the tranpose matrices $\left(\tilde{Q}\right)^t$ and $\left(P_{:,2:n}\right)^t$ have $k=m$ columns, (10) holds by the first case and we can proceed as before.

2.2 Numerical experiments

In order to check the validity of formula (3), we performed two numerical experiments in R (software for Statistical computing). The code is annexed in Appendix A.

2.2.1 First experiment

In the first experiment, we fixed $m=3$, $n=7$,

$$\begin{cases} q_0(y) &= \sin(y) \\ q_1(y) &= \pi y-4 \\ q_2(y) &= -5y^4+y^2-3 \end{cases} \qquad \begin{cases} p_0(x) &= \cos\left(x^2\right) \\ p_1(x) &= x^2-7x-3 \\ p_2(x) &= x-\sqrt{2}\log(x+2) \end{cases}$$

$$Q(y)=\begin{bmatrix} q_0(y) & y^3-\pi & \cos\left(y^2+7\right) & y^5-2y+19 & -3y^5+4y^4-1 & \log(y) & y \\ q_0(y) & q_1(y) & y^2-2 & \tan(y) & -y^7+y^3-5 & \cos(\sqrt{2}y) & y-1.2 \\ q_0(y) & q_1(y) & q_2(y) & -\sqrt{2}y+1 & 4y^5 & \sin(2y) & \cos(-3y) \end{bmatrix}$$

and

$$P(x)=\begin{bmatrix} x^4-7 & p_0(x) & p_0(x) & p_0(x) & p_0(x) & p_0(x) & p_0(x) \\ x\log(x+2) & x+1 & p_1(x) & p_1(x) & p_1(x) & p_1(x) & p_1(x) \\ x^5-1 & x^2+2 & x^3-1.21 & p_2(x) & p_2(x) & p_2(x) & p_2(x) \end{bmatrix}.$$

Then we sampled (randomly) 10^4 pairs $(X,Y)\in[1,5]^3\times[1,5]^7$ and computed the left-hand ($lh_{(3)}$)and the right-hand ($rh_{(3)}$) sides of (3).

The frequencies of the computed relative differences $\log_{10}\left|\frac{lh_{(3)}}{rh_{(3)}}-1\right|$ are shown in Figure 1. Recalling that this experiment computes determinants of order even $m\times n=21$ and that Vandermonde type matrices are ill conditioned in general, a mean order of $1.e-10$ for the relative difference due to rounding in double precision (i.e. IEEE 754 standard) seems reasonable.

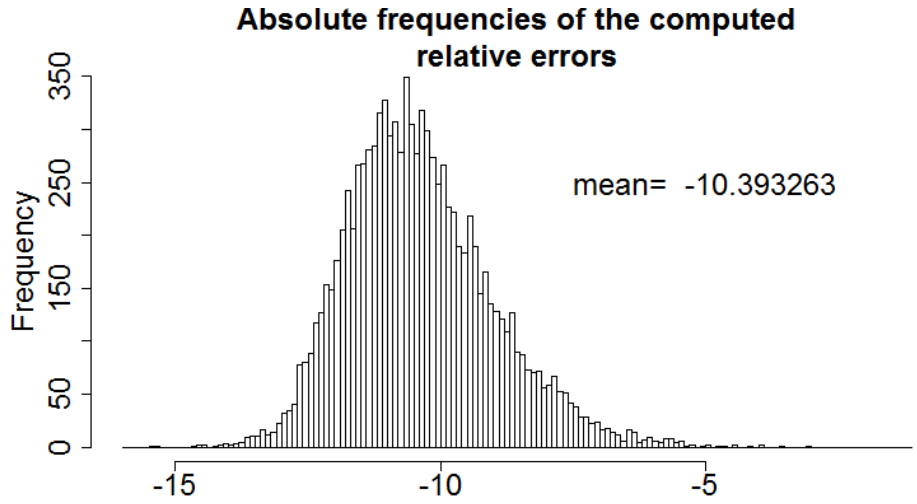

Figure 1: Distribution of the computed relative differences $\log_{10}\left|\frac{lh_{(3)}}{rh_{(3)}}-1\right|$ of 10^4 sampled pairs (X,Y) in $[1,5]^4\times[1,5]^5$.

2.3 Second experiment

In the second experiment, we fixed $m=6$, $n=4$, $X=\left(-1,-0.3,\frac{1}{e},\sqrt{2},\pi\right)$ and $Y=(-1,\cos(2),1,\log(10))$. Then we sampled (randomly) 10^4 pairs $(\alpha,\beta)\in[1,5]^6\times[1,5]^5$ and, for each sampled pair (α,β), we set q_0,q_1,q_2,q_3, p_0,p_1,p_2 and Q and P as follows

$$\begin{cases} q_0(y) &= |y|+\beta_1 \\ q_1(y) &= y^2-\beta_2 \\ q_2(y) &= -5y^4+\beta_3 y^2-\beta_4 \\ q_3(y) &= \beta_5 y^6-\beta_2 y+\beta_3 \end{cases} \quad \begin{cases} p_0(x) &= \begin{cases} x, & ,\text{if } x<\alpha_1 \\ \log(x+4) & ,\text{if } x\geq\alpha_1 \end{cases} \\ p_1(x) &= \alpha_2 x^2+\alpha_3 x-\alpha_4 \\ p_2(x) &= \alpha_5\sin(x+\alpha_6), \end{cases}$$

$$Q(y)=\begin{bmatrix} q_0(y) & y^3-2y+2 & \cos\big(y\tan(y)\big) & y^3-\sqrt{7}y^2) \\ q_0(y) & q_1(y) & \sin(y+2) & 3y^3-7 \\ q_0(y) & q_1(y) & q_2(y) & \big|y-3\cos(y)\big| \\ q_0(y) & q_1(y) & q_2(y) & q_3(y) \\ q_0(y) & q_1(y) & q_2(y) & q_3(y) \\ q_0(y) & q_1(y) & q_2(y) & q_3(y) \end{bmatrix},$$

$$P(x)=\begin{bmatrix} x+\cos(x) & p_0(x) & p_0(x) & p_0(x) \\ \alpha_1 x^3-\alpha_2 & \alpha_3 x+1 & p_1(x) & p_1(x) \\ x^5-1 & x^2+2 & x^3-1.21 & p_2(x) \\ \beta_1 x^3-2x^2+2 & x^6-2x^3+\beta_2 & \sin(x) & -\cos(x) \\ \beta_4|x-1| & \beta_1 x & \beta_2 x^2 & \beta_3\log(x+3) \\ \cos(x) & \beta_5\sin(x) & \pi x & -\tan(x) \end{bmatrix}$$

and we computed the left-hand ($lh_{(3)}$)and the right-hand ($rh_{(3)}$) sides of (3).

The frequencies of the computed relative errors $\log_{10}\left|\frac{lh_{(3)}}{rh_{(3)}}-1\right|$ are shown in Figure 2. Again, a mean order of $1.e-12$ for the relative difference due to rounding seems reasonable.

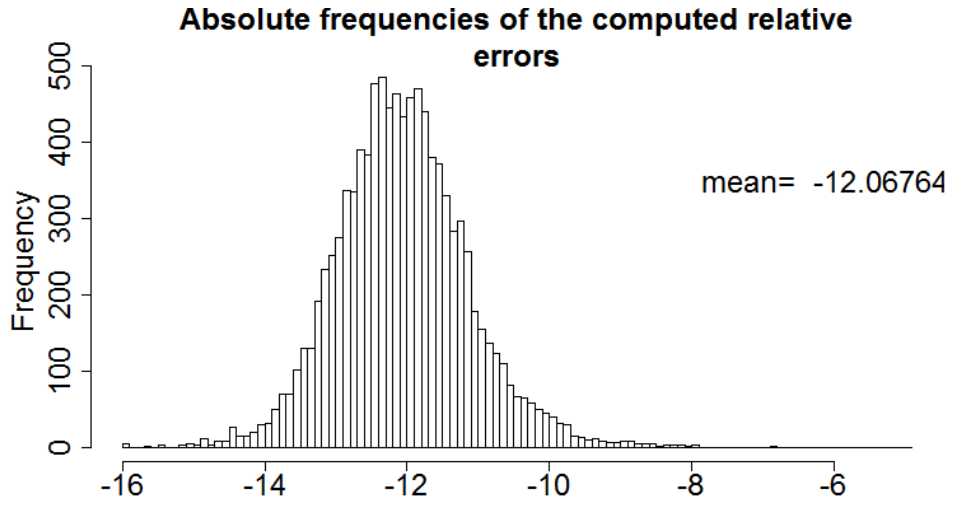

Figure 2: Distribution of the computed relative differences $\log_{10}\left|\frac{lh_{(3)}}{rh_{(3)}}-1\right|$ of 10^4 sampled pairs (α,β) in $[1,5]^6\times[1,5]^5$.

3 On the Vandermonde determinant associated to Padua-like points

Let us consider the univariate Chebyshev-Lobatto grid of $n+1$ points

$$C_{n+1}=\left\{z_j^n=\cos\left(\frac{(j-1)\pi}{n}\right), j=1,2,\ldots,n+1\right\} \tag{12}$$

and its disjoint decomposition

$$\begin{aligned} C_{n+1}^o &= \left\{z_j^n, j=1,2,\ldots,n+1, j \text{ odd}\right\} \\ C_{n+1}^e &= \left\{z_j^n, j=1,2,\ldots,n+1, j \text{ even}\right\}. \end{aligned} \tag{13}$$

The Padua points [2] $\mathcal{P}_n$ for interpolation in $[-1,1]^2$ is defined as follows

$$\mathcal{P}_n=(C_{n+1}^o\times C_{n+2}^o)\cup(C_{n+1}^e\times C_{n+2}^e)\ \subset C_{n+1}\times C_{n+2}. \tag{14}$$

This set has cardinality equal to the dimension $N = \binom{n+2}{2}$ of the space $\Pi_n(\mathbb{R}^2)$ of bivariate polynomials of total degree less than or equal to n. A fundamental property of the Padua points is that they are self intersections and boundary contacts of the following *Lissajous curve*

$$\gamma(t) = (\cos((n+1)t), \cos(nt)), \ t \in [0, \pi]. \tag{15}$$

Erb [9] recently considered more general *Lissajous curves* for generating nodes for interpolation in $[-1,1]^2$, namely

$$\gamma_p(t) = (\cos((n+p)t), \cos(nt)), \ t \in [0, \pi], \tag{16}$$

where n and $n+p$ are coprime. It was show in [9] that, for each value of p, the self intersections of the corresponding curve can be arranged in two rectangular grids. For $p = 1$, nodes characterized by this property were previously referred to as Padua-like points [8, 3]. Formally, the generic Padua-like points are formed by the union of two disjoint grids,

$$\mathcal{A}_n := \mathcal{A}_n^o \cup \mathcal{A}_n^e \tag{17}$$

where

$$\begin{aligned} A_n^o &= \left\{(x_{2i+1}, y_{2j+1}), 0 \le i \le \tfrac{n}{2}, \ 0 \le j \le \tfrac{n}{2}\right\} \\ A_n^e &= \left\{(x_{2i}, y_{2j}), 0 \le i \le \tfrac{n}{2}, \ 0 \le j \le \tfrac{n}{2}+1\right\} \end{aligned} \tag{18}$$

and $\{x_i\}_{i=1}^n, \{y_i\}_{i=1}^n$ are two sets containing n distinct elements each (the grids in (18) are displayed for n even. The case n odd is similar). In this section we show that the Vandermonde determinant associated to the generic Padua-like points (17) with respect to the basis of monomials

$$\mathcal{B}_n = \{x^\alpha y^\beta | \alpha + \beta \le n, \alpha \ge 0, \beta \ge 0\} \tag{19}$$

is nonvanishing. For the sake of simplicity, we will consider only n even. We will also assume that

$$x_i < x_j \text{ and } y_i < y_j, \quad \text{for } i < j. \tag{20}$$

Let

$$\mathcal{T}_n = \{x^\alpha y^\beta | 0 \le \alpha, \beta \le n\} \ \text{ and } \ \mathcal{T}_n^e = \left(a(x)\mathcal{B}_{\frac{n}{2}-1}\right) \cup \left(b(y)\mathcal{B}_{\frac{n}{2}-1}\right), \tag{21}$$

where $a(x)$ and $b(y)$ are the annihilating polynomials of the points in $\mathcal{A}_n^o$, that is

$$a(x) = \prod_{i=0}^{\frac{n}{2}} (x - x_{2i+1}), \quad b(y) = \prod_{j=0}^{\frac{n}{2}} (y - y_{2j+1}). \tag{22}$$

It is shown in [8] that the Vandermonde determinant $vdm(\mathcal{A}_n, \mathcal{B}_n)$ associated to the Padua-like points (17) with respect to the monomial basis (19) can be written as

$$vdm(\mathcal{A}_n, \mathcal{B}_n) \ = \ \pm vdm(\mathcal{A}_n^o, \mathcal{T}_{\frac{n}{2}}) \times vdm(\mathcal{A}_n^e, \mathcal{T}_{\frac{n}{2}}^e). \tag{23}$$

The first factor in the right-hand side of (23) is equal [3] to

$$\pm \left(\prod_{0 \le i < j \le n/2} (x_{2j+1} - x_{2i+1}) \right)^{n/2+1} \left(\prod_{0 \le i < j \le n/2} (y_{2j+1} - y_{2i+1}) \right)^{n/2+1} \tag{24}$$

and, therefore, is nonvanishing. Moreover, the version of Theorem 2.1 in [8] is used to show that the second factor in the right-hand side of (23) factors as

$$vdm(\mathcal{A}_n^e, \mathcal{T}_{\frac{n}{2}}^e) = \pm \left(\prod_{j=1}^{\frac{n}{2}+1} vdm(\mathcal{X}, P_{:,j}) \right) \left(\prod_{i=1}^{\frac{n}{2}} vdm(\mathcal{X}, Q_{i,:}) \right), \tag{25}$$

where

$$\mathcal{X} = \left\{x_{2i} | 1 \le i \le \frac{n}{2}\right\}, \ \ \mathcal{Y} = \left\{y_{2i} | 1 \le i \le \frac{n}{2}+1\right\}, \tag{26}$$

$$Q(y) = \begin{bmatrix} 1 & b(y) & y\,b(y) & \dots & y^{\frac{n}{2}-1} b(y) \\ 1 & y & b(y) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & y\,b(y) \\ 1 & y & \dots & y^{\frac{n}{2}-1} & b(y) \end{bmatrix} \tag{27}$$

and

$$P(x)=\begin{bmatrix} a(x) & 1 & 1 & \dots & 1 \\ xa(x) & a(x) & x & \dots & x \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ x^{\frac{n}{2}-2}a(x) & \dots & xa(x) & a(x) & x^{\frac{n}{2}-2} \end{bmatrix}. \tag{28}$$

It is also mentioned in [8] that the factors in (25) can be further factorized. Notice, for example, that, for every $k \le \frac{n}{2}$,

$$vdm(\mathcal{X},P_{:,j})=\det\begin{bmatrix} 1 & x_2 & \dots & x_2^{j-2} & a(x_2) & x_2a(x_2) & \dots & x_2^{\frac{n}{2}-j}a(x_2) \\ 1 & x_4 & \dots & x_4^{j-2} & a(x_4) & x_4a(x_4) & \dots & x_2^{\frac{n}{2}-j}a(x_4) \\ 1 & x_6 & \dots & x_6^{j-2} & a(x_6) & x_6a(x_6) & \dots & x_6^{\frac{n}{2}-j}a(x_6) \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^{j-2} & a(x_n) & x_na(x_n) & \dots & x_n^{\frac{n}{2}-j}a(x_n) \end{bmatrix} \tag{29}$$

is a polynomial in x_{2k} which vanishes at $\mathcal{X}/\{x_{2k}\}$. Therefore, it follows that $vdm(\mathcal{X},P_{:,j})$ must be divisible by $\prod_{r<s}(x_{2s}-x_{2r})$. The polynomial

$$\frac{vdm(\mathcal{X},P_{:,j})}{\prod_{r<s}(x_{2s}-x_{2r})} \tag{30}$$

can be regarded as a generalization of some Schur functions (where $a(x)=x^k$, for some $k>j-2$ [11]). Nevertheless, it is not easy in general to obtain further information on such polynomials [5, 6, 7], including characterizing all sets of nodes for which (30) is nonvanishing. Fortunately, in our case, we can directly infer that all factors in (25) are different from zero.

Theorem 3.1. *Let $m=\frac{n}{2}$. For each $j \in \{1,2,\dots m+1\}$, $i \in \{1,2\dots,m\}$ and $\mathcal{X},\mathcal{Y},P$ and Q defined in (26–28), we have $vdm(\mathcal{X},P_{:,j})\neq 0$ and $vdm(\mathcal{Y},Q_{i,:})\neq 0$.*

Proof. Let us prove the statement for P. The cases $j=1$ and $j=m+1$ are straightforward. For $j=m+1$,

$$vdm(\mathcal{X},P_{:,j})=\det\begin{bmatrix} 1 & x_2 & \dots & x_2^{m-1} \\ 1 & x_4 & \dots & x_4^{m-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^{m-1} \end{bmatrix} \tag{31}$$

is an ordinary Vandermonde determinant and it factors as $\prod_{r<s}(x_{2s}-x_{2r})$, as it is well known [10]. And, for $j=1$, one has $vdm(\mathcal{X},P_{:,j})=\left(\prod_{i=1}^{m}a(x_{2i})\right)vdm(\mathcal{X},P_{:,m+1})$.

For $2\le j\le m$ fixed, let $v=\begin{bmatrix} v_1 & v_2 & \dots & v_m \end{bmatrix}^t \in \mathbb{R}^m$ such that

$$VDM(\mathcal{X},P_{:,j})\times v=0, \tag{32}$$

where $VDM(\mathcal{X},P_{:,j})$ denotes the Vandermonde matrix on the right-hand side of (29). If $v\neq 0$, it follows by (29) that $x_2,x_4,\dots,x_{2m}$ are roots of the the nonzero polynomial

$$g(x):=\underbrace{\left(\sum_{k=0}^{j-2}v_kx^k\right)}_{r(x)}+a(x)\underbrace{\left(\sum_{k=j-1}^{m}v_kx^{k-j+1}\right)}_{s(x)}, \tag{33}$$

that is $g(x)$ must be divisible by $c(x):=\prod_{\ell=1}^{m}(x-x_{2\ell})$ (notice that, in this case, both $r(x)$ and $s(x)$ must be different from zero). Therefore, we must have

$$r(x)+a(x)s(x)=u(x)c(x), \tag{34}$$

where $deg(u(x)) \le deg\left(x^{m-j}a(x)\right)-deg(c(x))=(m-j)+(m+1)-m=m+1-j$.

Now consider the polynomial $h(x)=r(x)u(x)$. We have

$$h(x)+a(x)s(x)u(x)=u(x)^2c(x) \tag{35}$$

and

$$deg(h(x)) \leq j-2+m+1-j = m-1. \tag{36}$$

We shall proceed by showing that $h(x)$ must have at least m real roots, which will contradict (36). As a consequence, no $v \neq 0$ can be a solution of (32) and our proof will be complete.

Let $H = \{\xi_1, \xi_2, \ldots, \xi_\ell\}$ be the list of real roots of $h(x)$, counted with multiplicity, that is, $\xi_i = \xi_j$ may occur for $i \neq j$, but we will treat them as distinct objects. Let $\varphi : \{1,2,\ldots,m\} \to H$ be the function constructed inductively as follows: Assume that φ is already defined in $\{0,1,\ldots,i-1\} \cap \{1,2,\ldots,m\}$ and we shall define it for $i \geq 1$.

- If $u(x_{2i+1}) = 0$, then we set $\varphi(i) := \xi_j$ for some element ξ_j in H which corresponds to x_{2i+1}.
- If $u(x_{2i+1}) \neq 0$, there are two possibilities
  - If $u(x_{2i-1}) = 0$, then, by (22) and (34), we have $r(x_{2i-1}) = 0$. Therefore, x_{2i-1} is a multiple root of $h(x)$. In particular, besides of $\varphi(i-1)$ (when it exists), there is at least one other element ξ_k in H which corresponds to x_{2i-1}. Then we set $\varphi(i) := \xi_k$.
  - If $u(x_{2i-1}) \neq 0$, then (35) shows that $h(x_{2i-1})h(x_{2i+1}) = u(x_{2i-1})^4 c(x_{2i-1})c(x_{2i+1})$ and this quantity is negative, because, by (20), $c(x)$ has only one root (simple) in $[x_{2i-1}, x_{2i+1}]$. This shows that $h(x)$ must have at least one real root ξ_j in $]x_{2i-1}, x_{2i+1}[$ and we set $\varphi(i) := \xi_j$.

It is easy to see that the φ defined above is injective (notice, for example, that $\varphi(i) \in [x_{2i-1}, x_{2i+1}]\ \forall\ i$). Hence, we must have $\#H = \ell \geq m$ and we are done.

□

References

[1] Bos, L., De Marchi, S., and M. Vianello, Trivariate polynomial aproximation on Lissajous curves. *arXiv:1502.04114 [math.NA]* (2015)

[2] Bos, L., Caliari, M., De Marchi, S., Vianello, M., and Y. Xu, Bivariate Lagrange interpolation at the Padua points: the generating curve approach. *Journal of Aproximation Theory* 143 (1) (2006) pp. 15–25.

[3] Bos, L., De Marchi, S. and S. Waldron, On the Vandermonde Determinant of Padua-like Points. *Dolomites Research Notes on Approximation* 2 (2009) pp. 1–15.

[4] Caliari, M., De Marchi, S., and M. Vianello, Bivariate Lagrange interpolation on the square at new nodal sets. *Applied Mathematics and Computation* 165 (2005) pp. 261–274.

[5] De Marchi, S., On computing the factors of generalized Vandermonde determinants. *Recent Advances in Applied and Theoretical Mathematics* World Scientific and Engineering Society (2000) pp. 140–2144.

[6] De Marchi, S., Polynomials arising in factoring generalized Vandermonde determinants: an algorithm for computing their coefficients. *Mathematical and Computer Modelling* 34 (3-4) (2001) pp. 271–281.

[7] De Marchi, S., Polynomials arising in factoring generalized Vandermonde determinants II: A condition for monicity. *Applied Mathematics Letters* 15 (5) (2002) pp. 627–632.

[8] De Marchi, S. and K. Usevich, On certain multivariate Vandermonde determinants whose variables separate. *Linear Algebra and its Applications* 449 (2014) pp. 17–27.

[9] Erb, W., Bivariate Lagrange interpolation at the node points of Lissajous curves - the degenerate case. *arXiv:1503.00895* (2015).

[10] Krattenthaler, C., *Advanced determinantal calculus*. Séminaire Lotharingien Combin. (electronic) **42** (1999).

[11] Macdonald, I., *Symmetric functions and Hall polynomials*. Oxford university prees. New york (2008).

A R code

```
# R script for testing the factorization of the multivariate Vandermonde determinant.
# In the following, q_0, q_1, q_2, q_3, p_0,p_1, p_2, p_3 are funcions of the variables (x
# for p and y for q) Q and P are function matrices of x and y which depend also on q_0, q_1,
# q_2, q_3 and p_0,p_1, p_2, p_3, respectively.

# General settings.
##############################################################################################

VDMlin = function(Y,Q,q_0,q_1,q_2,q_3,i) # This funcion computes the Vandermonde determinant
{  res = NULL                            # taken from the i-th row of Q and the vector Y.
   for(j in 1:length(Y))
   {
    res = rbind(res,Q(Y[j],q_0,q_1,q_2,q_3)[i,]) # Adds the row corresponding to the j-th
   }                                             # component of Y to the Vandermonde matrix.
  return(abs(det(res)))    # Returns the absolute value of the vandermonde determinant.
}

##############################################################################################

VDMcol = function(X,P,p_0,p_1,p_2,p_3,i) # This funcion computes the Vandermonde determinant
```

```
{ res = NULL                              # taken from the i-th columm of P and the vector X.
  for(j in 1:length(X))
  {
    res = cbind(res,P(X[j],p_0,p_1,p_2,p_3)[,i])
  }
  return(abs(det(res)))
}

#############################################################################################

PQlin = function(x,y,P,Q, p_0,p_1,p_2,p_3, q_0,q_1,q_2,q_3) # This funcion computes the
{ res = P(x,p_0,p_1,p_2,p_3)*Q(y,q_0,q_1,q_2,q_3)             # element-wise product of P and
  dim(res) = NULL                # Q evaluated at x, y and return
  return(res)                     # it as a vector.
}

#############################################################################################

VDMmultVar = function(X,Y,P,Q,p_0,p_1,p_2,p_3, q_0,q_1,q_2,q_3)
{ res = NULL                               # This function computes the absolue value of the
  for(i in 1:length(X))                   # multivariate vandermonde determinant (the lhs of equation (3))
  {                                       # corresponding to vectors X and Y.
    for(j in 1:length(Y))
    {
      res = rbind(res,PQlin(X[i],Y[j],P,Q, p_0,p_1,p_2,p_3, q_0,q_1,q_2,q_3))  # Adds the
    }                                          # row corresponding  to the pointwise product
  }                                            # of P and Q evaluated  at X[i], Y[j].
  return(abs(det(res)))
}

#############################################################################################

VDMfac = function(X,Y,P,Q,p_0,p_1,p_2,p_3, q_0,q_1,q_2,q_3) # This function evaluates the
{ res = 1                                     # right-hand side of equation (3), i. e. the
  for(j in 1:length(Y))                    # products of the vandermondians taken from
  {                                             # the rows of Q and columns of Q and returns
    res = res*VDMcol(X,P,p_0,p_1,p_2,p_3,j)  # its absolute value.
  }

  for(j in 1:length(X))
  {
    res = res*VDMlin(Y,Q,q_0,q_1,q_2,q_3,j)

  }
  return(abs(res))

}

# Our thesis is that VDMfac must be identical to VDMmultVar in exact arithmetic.

#############################################################################################
# Especific settings
#############################################################################################

# First experiment (m = 3, n = 7)

# Defines p_0, p_1, p_2,p_3 and q_0, q_1, q_2,q_3.

q_0 = function(y){return(sin(y))}
q_1 = function(y){return(pi*y-4)}
q_2 = function(y){return(-5*y^4+y^2-3)}

p_0 = function(x){return(cos(x^2))}
p_1 = function(x){return(x^2-7*x-3)}
p_2 = function(x){return(x- sqrt(2)*log(x+2))}

# Defines the matrix Q as a function of q_0,q_1, q_2, q_3.

Q = function(y,q_0,q_1,q_2,q_3) # Evaluates the matrix function Q at the point y. The
{                              # functions q_0, q_1,q_2 and q_3 are passed as parameters.
  res = array(0,c(3,7))
```

```
  res[1,] = c(q_0(y), y^3-pi, cos(y^2+7), y^5-2*y+19,-3*y^5+4*y^4-1, log(y), y)  # Defines the first row of Q.
  res[2,] = c(q_0(y), q_1(y), y^2-2, tan(y),-y^7+y^3-5, cos(sqrt(2)*y), y-1.2)        # Defines the second row of Q.
  res[3,] = c(q_0(y), q_1(y), q_2(y), -sqrt(2)*y+1,4*y^5, sin(2*y), cos(-3*y))        # Defines the third row of Q.
  return(res)
}

P = function(x,p_0,p_1,p_2,p_3) # Evaluates the matrix function P at the point x. The functions
{ res = array(0,c(3,7))         # p_0, p_1,p_2 and p_3 are passed as parameters.
  res[1,] = c(x^4-7, p_0(x),p_0(x),p_0(x),p_0(x),p_0(x), p_0(x))    # Defines the first row of P.
  res[2,] = c(x*log(x+2), x+1,p_1(x),p_1(x),p_1(x), p_1(x), p_1(x))  # Defines the second row of P.
  res[3,] = c(x^5-1, x^2+2, x^3-1.21,p_2(x),p_2(x), p_2(x), p_2(x))  # Defines the third row of P.
  return(res)
}

M = 10000 # Number of samples.

relErrors = array(0,c(M))
for(i in 1:M)                # Run test.
{
    X = runif(3,min = 1, max = 5) # Samples X randomly.
    dim(X) = c(1,length(X))
    Y = runif(7,min = 1, max = 5) # Samples Y randomly.
    dim(Y) = c(1,length(Y))

    # Computes the left-hand side of equation (3).
    a =  VDMmultVar(X,Y,P,Q,p_0,p_1,p_2,p_3, q_0,q_1,q_2,q_3)

    # Computes the right-hand side of equation (3).
    b = VDMfac(X,Y,P,Q,p_0,p_1,p_2,p_3, q_0,q_1,q_2,q_3)

    relErrors[i] = abs(b/a - 1)   # Stores the computed relative error.
}

data = log(relErrors)/log(10)# plots the histogram of absolute value of the relative errors
data[which(relErrors == 0)] = -16
png("test_1.png", width = 800, height = 400)
par(mar = c(3,6,3,3))
hist(data,breaks = seq(-16,max(data)+2,0.1), main = paste("Absolute frequencies of the computed
relative errors"), cex.main = 2, xlab = NULL, cex.lab = 2,cex.axis = 2,   )
text(-5,250, paste("mean= ",round(mean(data),6)), cex= 2)
dev.off()

################################################################################

# Second experiment (m = 6, n = 4)

M = 10000 # Number of samples

 # fix X and Y
 X = c(-1,-0.3, 0, exp(-1), sqrt(2), pi)
 Y = c(-1,cos(2),1,log(10))

 relErrors = array(0,c(M)) # Run test

for(i in 1:M)
{
  alpha =  runif(6,min = -1, max = 1)  # Samples alpha and beta randomly.
  beta =  runif(5,min = -1, max = 1)

 q_0 = function(y){ return(abs(y) + beta[1])}
q_1 = function(y){return(y^2-beta[2])}
q_2 = function(y){return(-5*y^4+beta[3]*y^2-beta[4])}
q_3 = function(y){return(beta[5]*y^6-beta[2]*y+beta[3])}

p_0 = function(x) # Define a discontinuos function
{
  if(x < alpha[1]){return(x)}
  return(log(x+4))
 }
p_1 = function(x){return(alpha[2]*x^2+alpha[3]*x-alpha[4])}
p_2 = function(x){return(alpha[5]*sin(x+alpha[6]))}
```

```
p_3 = function(x){return(0)}   # p_3 is required as argument of Q, but is
# actually not used in this experiment.

Q = function(y,q_0,q_1,q_2,q_3)
{
  res = array(0,c(6,4))
  res[1,] = c(q_0(y), y^3-2*y+2, cos(y*tan(y)), y^3-sqrt(7)*y^2)
  res[2,] = c(q_0(y), q_1(y), sin(y+2), 3*y^3-7)
  res[3,] = c(q_0(y), q_1(y), q_2(y), abs(y-3*cos(y)))
  res[4,] = c(q_0(y), q_1(y), q_2(y), q_3(y))
  res[5,] = c(q_0(y), q_1(y), q_2(y), q_3(y))
res[6,] = c(q_0(y), q_1(y), q_2(y), q_3(y))

  return(res)
}

##########################################################################

# defines the matrix P as a function of  p_0,p_1, p_2, p_3

P = function(x,p_0,p_1,p_2,p_3)
{
  res = array(0,c(6,4))
  res[1,] = c(x+cos(x), p_0(x),p_0(x),p_0(x))
  res[2,] = c(alpha[1]*x^3-alpha[2], alpha[3]*x+1,p_1(x),p_1(x))
  res[3,] = c(x^5-1, x^2+2, x^3-1.21,p_2(x))
  res[4,] = c(beta[1]*x^3-2*x^2+2, x^6-2*x^3+beta[2], sin(x),-cos(x))
res[5,] = c(beta[4]*abs(x-1), beta[1]*x, beta[2]*x^2, beta[3]*log(x+3))
res[6,] = c(cos(x), beta[5]*sin(x), pi*x,-tan(x))
  return(res)
}

 # Computes the left-hand side of equation (3).
 a =  VDMmultVar(X,Y,P,Q,p_0,p_1,p_2,p_3, q_0,q_1,q_2,q_3)
 # Computes the right-hand side of equation (3).
 b = VDMfac(X,Y,P,Q,p_0,p_1,p_2,p_3, q_0,q_1,q_2,q_3)

 relErrors[i] = abs(b/a - 1)  # Stores the computed relative error.
}

data = log(relErrors)/log(10) # plots the histogram of absolute value of the relative errors
data[which(relErrors == 0)] = -16
png("test_2.png", width = 800, height = 400)
par(mar = c(3,6,3,3))
hist(data,breaks = seq(-16,max(data)+2,0.1), main = paste("Absolute frequencies of the computed relative
errors"), cex.main = 2, xlab = NULL, cex.lab = 2,cex.axis = 2,   )
text(-6,350, paste("mean= ",round(mean(data),6)), cex= 2)
dev.off()
```