NFN - Nationales Forschungsnetzwerk

Geometry + Simulation

http://www.gs.jku.at

On Transfinite Interpolation by Low-Rank Functions

Nira Dyn and Bert Jüttler and Dominik Mokriš

G+S Report No. 63

December 2017

Der Wissenschaftsfonds.

JOHANNES KEPLER UNIVERSITY LINZ

On Transfinite Interpolation by Low-Rank Functions

Nira Dyn[a], Bert Jüttler[b,*], Dominik Mokriš[b]

[a] *School of Mathematical Sciences, Tel Aviv University, Israel*
[b] *Institute of Applied Geometry, Johannes Kepler University, Linz, Austria*

Abstract

Given a bivariate function and a finite rectangular grid, we perform transfinite interpolation at all the points on the grid lines. We present a method that generates an interpolating function of low rank and show that the interpolating function is equivalent to the output of Schneider's CA2D algorithm [27]. Based on the tensor-product version of bivariate divided differences, we derive a new error bound that confirms the optimal approximation order of rank-n functions.

Keywords: transfinite interpolation, approximation order, low-rank function
2010 MSC: 41A05 65D05

1. Introduction

Transfinite interpolation addresses the task of constructing a function matching given data at a non-denumerable (transfinite) number of points. It was introduced by Gordon and Hall [9] based on their earlier work on blending functions [7, 8], although the particular case of Coons interpolation was proposed before [2]. Applications include mesh generation [9, 11] and construction of finite elements accurately capturing the boundary [10]. We refer to Sabin's survey [26] for an overview.

The idea of transfinite interpolation has been an active research topic ever since. It was extended to domains that are not of tensor-product type [24, 29]. Kuzmenko and Skorokhodov [19] recently studied transfinite interpolation of functions with bounded Laplacian. The Hermite-Lagrange transfinite interpolation by trigonometric blending functions was also investigated [3]. Transfinite mean value interpolation was proposed by Dyken and Floater [4], while Rvachev et al. [25] used R-functions to construct the interpolating function when the data are given implicitly. Also, the question of when the data can be interpolated by a smooth function was extensively investigated in the case of splines [22]. For interpolation by parametric Bézier surfaces, conditions on the input curves to become geodesics of the resulting surface have been studied [5].

*Corresponding author
Email address: `bert.juettler@jku.at` (Bert Jüttler)

The task of lofting (skinning) can also be seen as a particular case of transfinite interpolation. See Woodward [30] for the case of B-splines and Hohmeyer and Barsky [13] or Piegl and Tiller [23] for the case of NURBS.

Low-rank functions — that is, sums of a low number of separable functions — appear in numerical tensor calculus when using sparse tensor formats for representing multivariate functions [12, 17]. Interpolation by low-rank functions was studied by Schneider [27], and an efficient algorithm for low-rank approximation with bivariate tensor-product splines has been proposed [6]. In the context of isogeometric analysis [15], low-rank approximation has been successfully applied to adress the efficiency problem of matrix assembly [21]. This has motivated us to explore transfinite interpolation by low-rank functions, so far in the bivariate case.

In our recent paper [16], we proposed two constructions of a parametric low-rank spline surfaces from given boundary data. In particular, we focused on achieving geometric invariance and on the validity of the permanence principle. Furthermore, the two constructions were compared with other methods, in particular with Coons interpolation.

The current paper generalizes one of the proposed constructions (CR2I – coordinate-wise interpolation by rank 2 functions) to the case when the given data consist of a tensor-product grid of prescribed values (possibly more than 2×2 curves, which are no longer required to be splines), see Fig. 1. We analyze the properties of the algorithm and investigate the rank of the resulting function. In addition we observe that the proposed generalization is equivalent to the output of Schneider's CA2D (cross interpolation of bivariate functions) algorithm [27]. Furthermore, we use bivariate divided differences to derive a new error bound that confirms that interpolation by rank-n functions provides the same approximation order as n-fold transfinite interpolation with blending functions [9].

The rest of the paper is organized as follows. Section 2 recalls the generalization of divided differences to the bivariate tensor-product setting. Section 3 presents the new interpolation method, which generalizes Algorithm CR2I [16]. This is used in Section 4 to approximate bivariate functions. Finally, the L^∞-error of the approximation is analyzed in Section 5, and the paper concludes with two examples in Section 6.

2. Divided differences

Divided differences are a useful notion in the context of numerical analysis [28]. We recall the non-recursive version of their definition in the univariate case and the generalization to a bivariate tensor-product grid [18]. The latter will be used for deriving the error estimates.

Definition. The m-th *divided difference* of a univariate function φ with respect to real values $s_0, \ldots, s_m$ is defined by

$$\varphi[s_0, \ldots, s_m] = \sum_{i=0}^{m} \frac{\varphi(s_i)}{\prod\limits_{\substack{p=0,\ldots,m \\ p\neq i}} (s_i - s_p)}$$

whenever $s_0, \ldots, s_m$ are mutually different, and by the corresponding limit otherwise.

Given a system of functions $\varphi_0, \ldots, \varphi_m$, we recall that their discrete Wronskian with respect to the nodes $s_0, \ldots, s_m$ is the matrix with elements

$$\varphi_i[s_0, \ldots, s_j], \quad i, j = 0, \ldots, m \ ,$$

see Lascoux [20]. Its determinant satisfies the following identity:

$$\det\left(\varphi_i(s_j)\right)_{i,j=0,\ldots,m} = \Big(\prod_{0\leq k<\ell\leq m} (s_\ell - s_k)\Big) \det\left(\varphi_i[s_0, \ldots, s_j]\right)_{i,j=0,\ldots,m} \ . \tag{1}$$

For distinct nodes, the proof follows directly from the above Definition, see also [20, Proposition 9.3.1]. The identity is trivially satisfied whenever at least two nodes are equal.

Now we extend this definition to bivariate functions, cf. Kunz [18], Section 11.17.

Definition. Let φ be a bivariate function. The (m, n)-th *divided difference* of φ with respect to real values $s_0, \ldots, s_m$ and $t_0, \ldots, t_n$ is defined by

$$\varphi[s_0, \ldots, s_m][t_0, \ldots, t_n] = \sum_{i=0}^{m} \sum_{j=0}^{n} \frac{\varphi(s_i, t_j)}{\prod\limits_{\substack{k=0,\ldots,m \\ k\neq i}} (s_i - s_k) \prod\limits_{\substack{\ell=0,\ldots,n \\ \ell\neq j}} (t_j - t_\ell)} \tag{2}$$

whenever the denominator on the right-hand side is non-zero, and by the corresponding limit whenever some of $s_0, \ldots, s_m$ or $t_0, \ldots, t_n$ coincide.

Applying Eq. (1) repeatedly gives the following result:

Lemma 1. *For a bivariate function φ and real values $s_0, \ldots, s_m$ and $t_0, \ldots, t_m$ we have the identity*

$$\begin{aligned} &\det\big(\varphi(s_i, t_j)\big)_{i,j=0,\ldots,m} \\ =&\Big(\prod_{0\leq k<\ell\leq m} (s_\ell - s_k)(t_\ell - t_k)\Big) \det\big(\varphi[s_0, \ldots, s_i][t_0, \ldots, t_j]\big)_{i,j=0,\ldots,m} \ . \end{aligned}$$

Proof. Using Eq. (1) confirms

$$\det\big(\varphi(s_i, t_j)\big)_{i,j=0,\ldots,m} = \Big(\prod_{0\leq k<\ell\leq m} (t_\ell - t_k)\Big) \det\big(\varphi[s_i][t_0, \ldots, t_j]\big)_{i,j=0,\ldots,m} \ .$$

Using the same equation once more, $\det\left(\varphi[s_j][t_0,\ldots,t_i]\right)_{i,j=0,\ldots,m}$ equals

$$\Big(\prod_{0\le k<\ell\le m}(s_\ell - s_k)\Big)\det\left(\varphi[s_0,\ldots,s_j][t_0,\ldots,t_i]\right)_{i,j=0,\ldots,m} \quad . \qquad \square$$

In the following, we use the abbreviation

$$\varphi^{(k,\ell)}(x,y) = \frac{\partial^k}{\partial x^k}\frac{\partial^\ell}{\partial y^\ell}\varphi(x,y) \quad .$$

Furthermore, for an open set Ω we will use the symbol $\mathcal{C}^{m,n}(\overline{\Omega})$ to denote the class of functions φ such that for each $k = 0,\ldots,m$ and $\ell = 0,\ldots,n$ the derivative $\varphi^{(k,\ell)}$ is continuous in Ω and it can be continuously extended to $\overline{\Omega}$.

The following result is a bivariate analogue of the mean value theorem for divided differences [18, Section 5.7].

Lemma 2 (see Kunz [18], Section 11.17)**.** *Consider real values $s_0,\ldots,s_k$ and $t_0,\ldots,t_\ell$ and let $\varphi \in \mathcal{C}^{k,\ell}(\Delta)$, where*

$$\Delta = \left[\min_{i=0,\ldots,k} s_i,\ \max_{i=0,\ldots,k} s_i\right] \times \left[\min_{i=0,\ldots,\ell} t_i,\ \max_{i=0,\ldots,\ell} t_i\right] \quad .$$

Then there exists $(\widehat{s},\widehat{t}) \in \Delta$ such that

$$\varphi[s_0,\ldots,s_k][t_0,\ldots,t_\ell] = \frac{\varphi^{(k,\ell)}(\widehat{s},\widehat{t})}{k!\ \ell!} \quad .$$

3. Low rank interpolation

In this section we show how to construct a bivariate function interpolating a tensor-product grid of univariate functions.

From now on we assume that real values $x_1 < \cdots < x_n$ and $y_1 < \cdots < y_n$ are given, and we call them *nodes*. We use the notation

$$\Omega_x = [x_1, x_n] \ , \quad \Omega_y = [y_1, y_n] \quad \text{and} \quad \Omega = \Omega_x \times \Omega_y \ .$$

Furthermore, we use x_0 and y_0 to denote the variables in order to simplify the notation.

Consider a positive integer $\ell \le n$. Assume that we are given 2ℓ continuous functions

$$\sigma_i : \Omega_y \to \mathbb{R} \quad \text{and} \quad \tau_j : \Omega_x \to \mathbb{R}, \quad i,j = 1,\ldots,\ell \ ,$$

which satisfy the compatibility conditions

$$\sigma_i(y_j) = \tau_j(x_i) = c_{i,j} \ ,$$

and the additional constraint

$$\det\begin{pmatrix} c_{1,1} & \cdots & c_{1,\ell} \\ \vdots & \ddots & \vdots \\ c_{\ell,1} & \cdots & c_{\ell,\ell}\end{pmatrix} \neq 0 \ . \tag{3}$$

Lemma 3. *The function ψ defined by*

$$\psi(x_0, y_0) = -\frac{\det\begin{pmatrix} 0 & \tau_1(x_0) & \cdots & \tau_\ell(x_0) \\ \sigma_1(y_0) & c_{1,1} & \cdots & c_{1,\ell} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_\ell(y_0) & c_{\ell,1} & \cdots & c_{\ell,\ell} \end{pmatrix}}{\det\begin{pmatrix} c_{1,1} & \cdots & c_{1,\ell} \\ \vdots & \ddots & \vdots \\ c_{\ell,1} & \cdots & c_{\ell,\ell} \end{pmatrix}} \tag{4}$$

for all arguments $(x_0, y_0) \in \Omega$ interpolates the given functions, i.e.,

$$\forall i = 1, \ldots, \ell,\ \forall (x_0, y_0) \in \Omega :\ \psi(x_0, y_i) = \tau_i(x_0) \ \textit{and}\ \psi(x_i, y_0) = \sigma_i(y_0)\ .$$

Proof. Writing $\sigma_i(y_j)$ instead of $c_{i,j}$ and expanding the numerator determinant in (4) with respect to the first row gives

$$\psi(x_0, y_0) = \sum_{k=1}^{\ell} (-1)^k \frac{\det\begin{pmatrix} \sigma_1(y_0) & \cdots & \sigma_1(y_{k-1}) & \sigma_1(y_{k+1}) & \cdots & \sigma_1(y_\ell) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \sigma_\ell(y_0) & \cdots & \sigma_\ell(y_{k-1}) & \sigma_\ell(y_{k+1}) & \cdots & \sigma_\ell(y_\ell) \end{pmatrix}}{\det\begin{pmatrix} \sigma_1(y_1) & \cdots & \sigma_1(y_\ell) \\ \vdots & \ddots & \vdots \\ \sigma_\ell(y_1) & \cdots & \sigma_\ell(y_\ell) \end{pmatrix}} \tau_k(x_0)\ . \tag{5}$$

We evaluate $\psi(x_0, y_i)$ for any $i = 1, \ldots, n$ by setting $y_0 = y_i$. On the one hand, the determinants in the numerator are equal to zero whenever the summation index satisfies $k \neq i$, because they contain two identical columns. On the other hand, the determinant in the numerator for $k = i$ equals $(-1)^k$ times the determinant in the denominator and thus $\psi(x_0, y_i) = \tau_i(x_0)$.

To prove that $\psi(x_i, y_0) = \sigma_i(y_0)$ we proceed analogously, expanding the numerator determinant in (4) with respect to the first column. $\square$

Note that the lemma defines a procedure to construct a bivariate function ψ that interpolates the 2ℓ univariate functions σ_i, τ_i, $i = 1, \ldots, \ell$. See Fig. 1 for an example of input and output of this procedure.

Definition. Let $\Omega \subseteq \mathbb{R}^2$ be given. The *rank* of a function $\psi : \Omega \to \mathbb{R}$ is the minimal number r such that there exists a representation of the form

$$\forall (x_0, y_0) \in \Omega : \quad \psi(x_0, y_0) = \sum_{s=1}^{r} \gamma_s(x_0)\, \eta_s(y_0) \tag{6}$$

for some functions $\gamma_s : \Omega_x \to \mathbb{R}$ and $\eta_s : \Omega_y \to \mathbb{R}$, $s = 1, \ldots, r$.

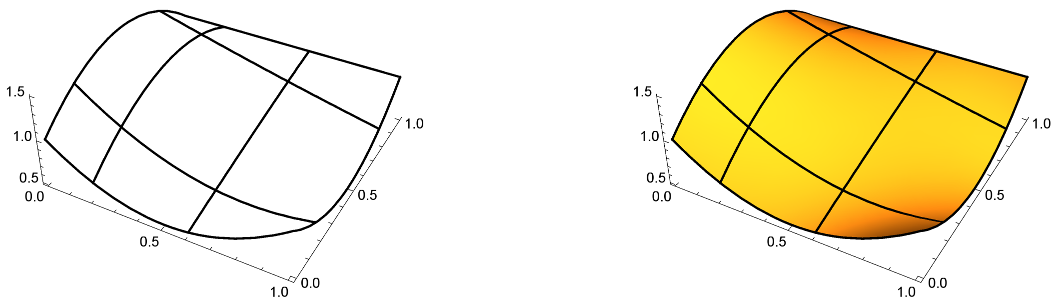

Figure 1: Left: given univariate functions; right: interpolating bivariate function.

The following lemma establishes a relation between the rank of a function and the rank of a matrix.

Lemma 4. *The rank of any matrix* $(\psi(x_i, y_j))_{i=1,\dots,k;j=1,\dots,\ell}$ *obtained by sampling values of a rank-*r *function* ψ *does not exceed* r*.*

Proof. We use the representation (6) to rewrite the matrix as a product of two matrices with dimensions (k, r) and (r, ℓ),

$$(\psi(x_i, y_j))_{i=1,\dots,k;j=1,\dots,\ell} = (\gamma_s(x_i))_{i=1,\dots,k;s=1,\dots,r} \; (\eta_s(y_j))_{s=1,\dots,r;j=1,\dots,\ell}$$

and note that the rank of the product does not exceed the rank of each factor. $\square$

Clearly, the rank of a function need not exist. However, the rank of the functions obtained by using Lemma 3 is finite:

Lemma 5. *The function* ψ *constructed in Lemma 3 has rank* ℓ *on* Ω*.*

Proof. On the one hand, the determinants in the numerators in (5) are functions of y_0. Hence (5) is an expansion of the form (6) with $r = \ell$. On the other hand, we have that $c_{i,j} = \psi(x_i, y_j)$ for $i, j = 1, \dots, \ell$. Due to the constraint (3), the matrix $(\psi(x_i, y_j))_{i,j=1,\dots,\ell}$ has full rank, thus the rank of ψ cannot be less than ℓ. $\square$

4. Function approximation through interpolation

Now we specialize the procedure of the previous section to the case when the univariate functions σ_i and τ_i are slices of a bivariate function. This will enable us to study the approximation power of the construction.

In view of Lemma 5 we introduce the following notions for any $\ell = 1, \dots, n$:

- A function φ is said to be *ℓ-admissible* (with respect to the nodes $x_1, \ldots, x_\ell$ and $y_1, \ldots, y_\ell$), if it satisfies the condition

$$\det \begin{pmatrix} \varphi(x_1, y_1) & \cdots & \varphi(x_1, y_\ell) \\ \vdots & \ddots & \vdots \\ \varphi(x_\ell, y_1) & \cdots & \varphi(x_\ell, y_\ell) \end{pmatrix} \neq 0 \ . \tag{7}$$

- For an ℓ-admissible function φ, we define the *rank-ℓ approximation operator* (again with respect to the nodes $x_1, \ldots, x_\ell$ and $y_1, \ldots, y_\ell$) by

$$(\mathbf{R}_\ell \varphi)(x_0, y_0) = -\frac{\det \begin{pmatrix} 0 & \varphi(x_0, y_1) & \cdots & \varphi(x_0, y_\ell) \\ \varphi(x_1, y_0) & \varphi(x_1, y_1) & \cdots & \varphi(x_1, y_\ell) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(x_\ell, y_0) & \varphi(x_\ell, y_1) & \cdots & \varphi(x_\ell, y_\ell) \end{pmatrix}}{\det \begin{pmatrix} \varphi(x_1, y_1) & \cdots & \varphi(x_1, y_\ell) \\ \vdots & \ddots & \vdots \\ \varphi(x_\ell, y_1) & \cdots & \varphi(x_\ell, y_\ell) \end{pmatrix}} \ .$$

In particular, the operator $\mathbf{R}_\ell$ transforms any ℓ-admissible function φ into the function $\psi = \mathbf{R}_\ell \varphi$ constructed according to Lemma 3, where

$$\sigma_i(y_0) = \varphi(x_i, y_0), \quad \tau_j(x_0) = \varphi(x_0, y_j), \quad c_{i,j} = \varphi(x_i, y_j), \quad i, j = 1, \ldots, \ell \ .$$

Note that ψ is also ℓ-admissible (with respect to $x_1, \ldots, x_\ell$ and $y_1, \ldots, y_\ell$).

In addition we note the following observation, which imposes restrictions on the input of the operator $\mathbf{R}_\ell$:

Lemma 6. *Functions φ of rank $r < \ell$ are not ℓ-admissible on Ω.*

This is obvious since by Lemma 4 the rank of any matrix obtained by sampling values of a rank-r function on a tensor-product grid cannot exceed r.

Next we show that the non-linear operator $\mathbf{R}_\ell$ is a *projector* onto the set of rank ℓ functions, as made precise in the following lemma:

Lemma 7. *If φ is an ℓ-admissible function with respect to the nodes $x_1, \ldots, x_\ell$ and $y_1, \ldots, y_\ell$, then*

$$\varphi(x_0, y_0) - (\mathbf{R}_\ell \varphi)(x_0, y_0) = \frac{\det \left(\varphi(x_i, y_j)\right)_{i,j=0,\ldots,\ell}}{\det \left(\varphi(x_i, y_j)\right)_{i,j=1,\ldots,\ell}} \ . \tag{8}$$

In particular, $\varphi = \mathbf{R}_\ell \varphi$ if φ has rank ℓ.

Proof. Using the multilinearity of determinants with respect to the row vectors, we rewrite the numerator of the right-hand side in (8) as

$$\det\begin{pmatrix}\varphi(x_0,y_0) & 0 & \cdots & 0\\ \varphi(x_1,y_0) & \varphi(x_1,y_1) & \cdots & \varphi(x_1,y_\ell)\\ \vdots & \vdots & \ddots & \vdots\\ \varphi(x_\ell,y_0) & \varphi(x_\ell,y_1) & \cdots & \varphi(x_\ell,y_\ell)\end{pmatrix} + \det\begin{pmatrix}0 & \varphi(x_0,y_1) & \cdots & \varphi(x_0,y_\ell)\\ \varphi(x_1,y_0) & \varphi(x_1,y_1) & \cdots & \varphi(x_1,y_\ell)\\ \vdots & \vdots & \ddots & \vdots\\ \varphi(x_\ell,y_0) & \varphi(x_\ell,y_1) & \cdots & \varphi(x_\ell,y_\ell)\end{pmatrix} ,$$

from where (8) follows directly due to the ℓ-admissibility of φ and the definition of $\mathbf{R}_\ell\varphi$. Finally we note that in view of Lemma 4 the numerator of the right-hand side in (8) vanishes if φ has rank ℓ, since it is the determinant of a matrix obtained by sampling values on a tensor-product grid with $\ell+1$ rows and columns. □

Next we establish the relation between $\mathbf{R}_\ell\varphi$ and the CA2D algorithm ("two-dimensional cross approximation") of Schneider [27]. For that we introduce the cross approximation operator $\mathbf{I}_\ell$ for $\ell = 1,\ldots,n$ that transforms any function γ satisfying $\gamma(x_\ell,y_\ell)\neq 0$ into the rank-1 function

$$(\mathbf{I}_\ell\gamma)(x_0,y_0) = \frac{\gamma(x_\ell,y_0)\,\gamma(x_0,y_\ell)}{\gamma(x_\ell,y_\ell)} , \tag{9}$$

which interpolates γ on $(\{x_\ell\}\times\mathbb{R})\cup(\mathbb{R}\times\{y_\ell\})$. Note that $\mathbf{I}_1 = \mathbf{R}_1$. In case $\gamma(x_\ell,y_\ell) = 0$, $\mathbf{I}_\ell\gamma = 0$.

Theorem 8. *Let φ be ℓ-admissible with respect to the nodes $x_1,\ldots,x_\ell$ and $y_1,\ldots,y_\ell$ for $\ell = 1,\ldots,n$. The sequence of rank-ℓ approximations of φ satisfies the recurrence*

$$\mathbf{R}_\ell\varphi = \mathbf{R}_{\ell-1}\varphi + \mathbf{I}_\ell(\varphi - \mathbf{R}_{\ell-1}\varphi) \tag{10}$$

for $\ell = 2,\ldots,n$

Proof. First, we prove that the functions on both sides in (10) take the same values on the grid lines defined by $x_1,\ldots,x_\ell$ and $y_1,\ldots,y_\ell$. On the one hand, the definitions of the operators imply

$$(\mathbf{R}_\ell\varphi)(x_i,y_0) = (\mathbf{R}_{\ell-1}\varphi)(x_i,y_0) + \frac{(\varphi-\mathbf{R}_{\ell-1}\varphi)(x_\ell,y_0)\,(\varphi-\mathbf{R}_{\ell-1}\varphi)(x_i,y_\ell)}{(\varphi-\mathbf{R}_{\ell-1}\varphi)(x_\ell,y_\ell)}$$

if $1\le i<\ell$, since $\mathbf{R}_{\ell-1}\varphi$ interpolates φ on the grid lines defined by $x_1,\ldots,x_{\ell-1}$. On the other hand, this equation is clearly true if $i=\ell$. Analogously, we see that

$$(\mathbf{R}_\ell\varphi)(x_0,y_j) = (\mathbf{R}_{\ell-1}\varphi)(x_0,y_j) + \frac{(\varphi-\mathbf{R}_{\ell-1}\varphi)(x_\ell,y_j)\,(\varphi-\mathbf{R}_{\ell-1}\varphi)(x_0,y_\ell)}{(\varphi-\mathbf{R}_{\ell-1}\varphi)(x_\ell,y_\ell)}$$

if $1\le j\le\ell$.

Second, we use this fact to establish the identity

$$\mathbf{R}_\ell(\mathbf{R}_\ell\varphi) = \mathbf{R}_\ell(\mathbf{R}_{\ell-1}\varphi + \mathbf{I}_\ell(\varphi-\mathbf{R}_{\ell-1}\varphi)) , \tag{11}$$

which follows from the fact that the rank-ℓ interpolation of a function depends solely on the values on the associated grid lines.

Finally, we complete the proof by noting that (11) implies the recurrence (10), as $\mathbf{R}_\ell$ reproduces rank-ℓ functions according to Lemma 7. $\square$

This proves that $\mathbf{R}_\ell\varphi$ is the same as the function obtained by Schneider's CA2D algorithm, which constructs a function by iteratively using the recurrence (10). Moreover, Theorem 8 also confirms the expression for the error derived in Lemma 7, since it is equivalent to Schneider's Remark 3.3 [27].

5. Error estimates

Now we analyze the L^∞–error of the approximation introduced in the previous section. Throughout this section we assume that $\Omega = [0,h]^2$ and that the nodes satisfy $x_1 = y_1 = 0$, $x_n = y_n = h$.

It follows directly from Lemmas 1 and 2 that

Lemma 9. *For $\varphi \in \mathcal{C}^{n,n}(\Omega)$ there exist $(\widehat{x}_{ij}, \widehat{y}_{ij}) \in \Omega$, $i,j = 0,\ldots,n-1$, such that*

$$\det\Big(\varphi(x_i,y_j)\Big)_{i,j=1,\ldots,n} = \frac{\prod\limits_{1\le k<\ell\le n}(x_\ell - x_k)(y_\ell - y_k)}{\big(1!\cdots(n-1)!\big)^2}\det\Big(\varphi^{(i,j)}(\widehat{x}_{ij},\widehat{y}_{ij})\Big)_{i,j=0,\ldots,n-1} .$$

The smoothness of φ then implies the following result:

Corollary 10. *If φ satisfies the assumption of the previous lemma and if*

$$\det\Big(\varphi^{(i,j)}(0,0)\Big)_{i,j=1,\ldots,n} \neq 0 \ , \tag{12}$$

then there exists $h^* > 0$ such that φ is n-admissible for any two monotonic sequences $0 = x_1 < \cdots < x_n = h$ and $0 = y_1 < \cdots < y_n = h$ with $h < h^*$.

Now we can state the first error bound:

Lemma 11. *For $\varphi : \Omega \to \mathbb{R}$ satisfying $\det\left(\varphi[x_1,\ldots,x_i][y_1,\ldots,y_j]\right)_{i,j=1,\ldots,n} \neq 0$,*

$$\|\varphi - \mathbf{R}_n\varphi\|_{L^\infty(\Omega)} \le h^{2n} \sup_{(x_0,y_0)\in\Omega}\left|\frac{\det\left(\varphi[x_0,\ldots,x_i][y_0,\ldots,y_j]\right)_{i,j=0,\ldots,n}}{\det\left(\varphi[x_1,\ldots,x_i][y_1,\ldots,y_j]\right)_{i,j=1,\ldots,n}}\right| . \tag{13}$$

Proof. Using Lemma 7, we get

$$\|\varphi - \mathbf{R}_n\varphi\|_{L^\infty(\Omega)} = \sup_{(x_0,y_0)\in\Omega}\left|\frac{\det\left(\varphi(x_i,y_j)\right)_{i,j=0,\ldots,n}}{\det\left(\varphi(x_i,y_j)\right)_{i,j=1,\ldots,n}}\right| .$$

Applying Lemma 1, the right-hand side above becomes

$$\sup_{(x_0,y_0)\in\Omega}\left|\frac{\prod_{0\le k<\ell\le n}(x_\ell-x_k)(y_\ell-y_k)}{\prod_{1\le k<\ell\le n}(x_\ell-x_k)(y_\ell-y_k)}\right|\ \left|\frac{\det\left(\varphi[x_0,\dots,x_i][y_0,\dots,y_j]\right)_{i,j=0,\dots,n}}{\det\left(\varphi[x_1,\dots,x_i][y_1,\dots,y_j]\right)_{i,j=1,\dots,n}}\right| \ .$$

Noting that

$$\left|\frac{\prod_{0\le k<\ell\le n}(x_\ell-x_k)(y_\ell-y_k)}{\prod_{1\le k<\ell\le n}(x_\ell-x_k)(y_\ell-y_k)}\right| = \left|\prod_{\ell=1}^{n}(x_\ell-x_0)(y_\ell-y_0)\right| \le h^{2n},$$

we conclude the claim of the lemma. $\square$

Now we state the main result of the paper:

Theorem 12. *Let $\varphi\in\mathcal{C}^{n,n}(\Omega)$ be an n-admissible function and $\overline{h}$ a real number such that*

$$c_\varphi = \frac{\displaystyle\sup_{\substack{\widehat{u}_{ij},\widehat{v}_{ij}\in(0,\overline{h})\\ i,j=0,\dots,n}}\left|\det\left(\varphi^{(i,j)}(\widehat{u}_{ij},\widehat{v}_{ij})\right)_{i,j=0,\dots,n}\right|}{\displaystyle\inf_{\substack{\check{u}_{ij},\check{v}_{ij}\in(0,\overline{h})\\ i,j=1,\dots,n}}\left|\det\left(\varphi^{(i,j)}(\check{u}_{ij},\check{v}_{ij})\right)_{i,j=1,\dots,n}\right|} < \infty \ .$$

Then

$$\forall h\in(0,\overline{h}): \quad \|\varphi-\mathbf{R}_n\varphi\|_{\infty,\Omega}\le c_\varphi\frac{h^{2n}}{(n!)^2} \ . \tag{14}$$

Proof. Consider a particular point $(x_0,y_0)\in\Omega$. According to Lemma 2 there exist $(\widehat{x}_{ij},\widehat{y}_{ij})\in\Omega$, $i,j=0,\dots,n$ and $(\check{x}_{ij},\check{y}_{ij})\in\Omega$, $i,j=1,\dots,n$, such that

$$\left|\frac{\det\left(\varphi[x_0,\dots,x_i][y_0,\dots,y_j]\right)_{i,j=0,\dots,n}}{\det\left(\varphi[x_1,\dots,x_i][y_1,\dots,y_j]\right)_{i,j=1,\dots,n}}\right|$$

is equal to

$$\left|\left(\frac{1!\cdots(n-1)!}{1!\cdots n!}\right)^2\frac{\det\left(\varphi^{(i,j)}(\widehat{x}_{ij},\widehat{y}_{ij})\right)_{i,j=0,\dots,n}}{\det\left(\varphi^{(i,j)}(\check{x}_{ij},\check{y}_{ij})\right)_{i,j=1,\dots,n}}\right| \ .$$

The definition of the constant c_φ implies

$$\left|\frac{\det\left(\varphi^{(i,j)}(\widehat{x}_{ij},\widehat{y}_{ij})\right)_{i,j=0,\dots,n}}{\det\left(\varphi^{(i,j)}(\check{x}_{ij},\check{y}_{ij})\right)_{i,j=1,\dots,n}}\right|\le c_\varphi \ .$$

hence for all $(x_0,y_0)\in\Omega$

$$\left|\frac{\det\left(\varphi[x_0,\dots,x_i][y_0,\dots,y_j]\right)_{i,j=0,\dots,n}}{\det\left(\varphi[x_1,\dots,x_i][y_1,\dots,y_j]\right)_{i,j=1,\dots,n}}\right|\le\frac{1}{(n!)^2}c_\varphi \ ,$$

and (14) follows, in view of Lemma 11. $\square$

Note that if

$$\det\left(\varphi^{(i,j)}(0,0)\right)_{i,j=1,\dots,n} \neq 0 \ , \tag{15}$$

there is an $\overline{h}$ such that $c_\varphi < \infty$. Additionally, this assumption also ensures the n–admissibility for a certain neighborhood of the origin according to Corollary 10. Thus the assumptions of Theorem 12 hold under (15).

We briefly analyze the relation to existing results:

- It has been proved [9, Theorem 2] that transfinite interpolation with blending functions also gives an approximation error of order h^{2n}.

- Schneider [27, Proposition 2.3] derives an error estimate for $h = 1$, i.e., for functions defined on $\Omega = [0,1]^2$, which is valid for a particular choice of the nodes ('partial pivoting'). We apply his result to the function $\widetilde{\varphi}(\widetilde{x}_0, \widetilde{y}_0) = \varphi(h\widetilde{x}_0, h\widetilde{y}_0)$ and to the low-rank interpolation operator $\widetilde{\mathbf{R}}_n$ with respect to the nodes $\widetilde{x}_i = x_i/h$ and $\widetilde{y}_i = y_i/h$ and obtain the inequality

$$\begin{aligned}|(\varphi - \mathbf{R}_n\varphi)(x_0,y_0)| &= |(\widetilde{\varphi} - \widetilde{\mathbf{R}}_n\widetilde{\varphi})(\frac{x_0}{h},\frac{y_0}{h})| \leq \frac{2^n}{n!}\prod_{i=1}^{n}|\frac{x_0}{h} - \widetilde{x}_i| \sup_{\widetilde{u}\in[0,1]}|\widetilde{\varphi}^{(n,0)}(\widetilde{u},\frac{y_0}{h})| \\ &= \frac{2^n}{h^n n!}\prod_{i=1}^{n}|x_0 - x_i| \sup_{u\in[0,h]} h^n|\varphi^{(n,0)}(u,y_0)| \leq \frac{2^n}{n!}h^n \sup_{u,v\in[0,h]}|\varphi^{(n,0)}(u,v)|\end{aligned}$$

  for all $x_0, y_0 \in [0,h]$. Consequently, Schneider's result implies approximation order h^n, which is, however, not optimal. This may be caused by the asymmetry with respect to the order of the two variables.

6. Two examples

First we apply the interpolation scheme to a polynomial φ of bidegree $(7,7)$ with randomly chosen coefficients. Figure 2 compares the L^∞–error of three methods: rank-2 interpolation $\mathbf{R}_2$, rank 3-interpolation $\mathbf{R}_3$ with $x_2 = y_2 = \frac{h}{2}$, and Coons interpolation $\mathbf{C}$ [2, 14]. Clearly, both $\mathbf{R}_2$ and $\mathbf{C}$ yield an error of the order h^4, whereas $\mathbf{R}_3$ gives a higher order h^6 of accuracy, as expected.

Coons interpolation is known to converge with the fourth order. This can be verified by noticing that it is the particular case of the transfinite interpolation with blending functions [9] with $n = 2$, see the remark at the end of the previous section.

In the second example, we study the L^∞–error which is present when approximating the function

$$\varphi = 1 + s + s^2 + s^3 + (s + s^2 + s^3)t + s^3t^2 + (s^2 + s^3)t^3 \ .$$

We consider the same three methods as in the previous example. The results are visualized in Figure 3. Note that since

$$\det\left(\varphi^{(i,j)}(0,0)\right)_{i,j=0,1,2} = 0 \ ,$$

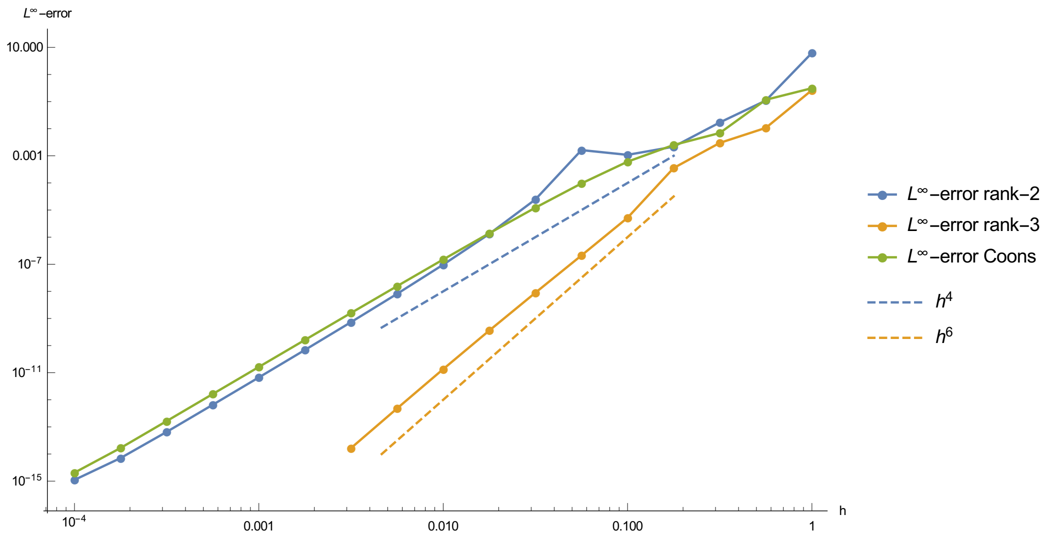

Figure 2: First example: Approximation errors for a randomly chosen degree 7 polynomial when using $\mathbf{R}_2$, $\mathbf{R}_3$ and $\mathbf{C}$.

the assumption of Theorem 12 is *not* satisfied. Consequently, both $\mathbf{R}_2$ and $\mathbf{R}_3$ do not exhibit the same order of convergence as in the previous example. In both cases, the order seems to be equal to h^5.

7. Closure

We studied methods for transfinite interpolation by bivariate functions of low rank and investigated the approximation power of these schemes. Future work might address the generalization to the multivariate case (see [1] for cross interpolation of multivariate functions), although the underlying tensor rank is much harder to characterize than matrix rank.

Acknowledgments

This research has been supported by the Austrian Science Fund (FWF, NFN S117 "Geometry + Simulation").

References

[1] Bebendorf, M. [2011]. Adaptive cross approximation of multivariate functions, *Constr. Approx.* **34**: 149–179.

[2] Coons, S. A. [1964]. Surfaces for computer aided design, *Technical Report, MIT* .

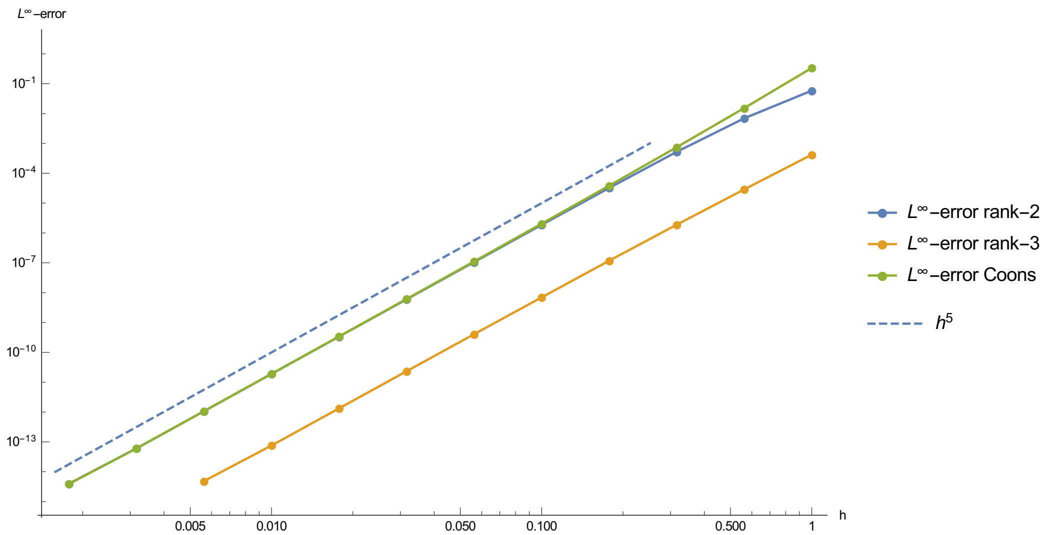

Figure 3: Second example: Approximation errors for a function φ not satisfying the assumptions of Theorem 12. Note that the blue and the green curves are virtually identical for small values of h.

[3] Dryanov, D. and Petrov, P. [2012]. Interpolation and L_1-approximation by trigonometric polynomials and blending functions, *J. Approx. Theory* **164**(8): 1049–1064.

[4] Dyken, C. and Floater, M. S. [2009]. Transfinite mean value interpolation, *Comput. Aided Geom. Design* **26**(1): 117–134.

[5] Farouki, R. T., Szafran, N. and Biard, L. [2009]. Existence conditions for coons patches interpolating geodesic boundary curves, *Comput. Aided Geom. Design* **26**(5): 599–614.

[6] Georgieva, I. and Hofreither, C. [2017]. An algorithm for low-rank approximation of bivariate functions using splines, *J. Comput. Appl. Math.* **310**: 80–91.

[7] Gordon, W. J. [1969]. Distributive lattices and the approximation of multivariate functions, *in* I. J. Schoenberg (ed.), *Approximation with special emphasis on spline functions*, Academic Press, New-York.

[8] Gordon, W. J. [1971]. Blending-function methods of bivariate and multivariate interpolation and approximation, *SIAM J. Numer. Anal.* **8**(1): 158–177.

[9] Gordon, W. J. and Hall, C. A. [1973a]. Construction of curvilinear co-ordinate systems and applications to mesh generation, *Int. J. Numer. Meth. Eng.* **7**(4): 461–477.

[10] Gordon, W. J. and Hall, C. A. [1973b]. Transfinite element methods: Blending-function interpolation over arbitrary curved element domains, *Numer. Math.* **21**(2): 109–129.

[11] Gordon, W. J. and Thiel, L. C. [1982]. Transfinite mappings and their application to grid generation, *Appl. Math. Comput.* **10**: 171–233.

[12] Hackbusch, W. [2012]. *Tensor Spaces and Numerical Tensor Calculus*, Springer.

[13] Hohmeyer, M. E. and Barsky, B. A. [1991]. Skinning rational B-spline curves to construct an interpolatory surface, *CVGIP: Graphical Models and Image Processing* **53**(6): 511–521.

[14] Hoschek, J., Lasser, D. and Schumaker, L. L. [1993]. *Fundamentals of computer aided geometric design*, AK Peters, Ltd.

[15] Hughes, T. J. R., Cottrell, J. A. and Bazilevs, Y. [2005]. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Comput. Methods Appl. Mech. Engrg.* **194**: 4135–4195.

[16] Jüttler, B. and Mokriš, D. [2017]. Low rank interpolation of boundary spline curves, *Comput. Aided Geom. Design* **55**: 48–68.

[17] Khoromskij, B. N. and Schwab, C. [2011]. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs, *SIAM J. Sci. Comput.* **33**(1): 364–385.

[18] Kunz, K. [1957]. *Numerical Analysis*, McGraw-Hill.

[19] Kuzmenko, D. and Skorokhodov, D. [2016]. Optimization of transfinite interpolation of functions with bounded Laplacian by harmonic splines on box partitions, *J. Approx. Theory* **209**: 44–57.

[20] Lascoux, A. [2003]. *Symmetric functions and combinatorial operators on polynomials*, Vol. 99, American Mathematical Society.

[21] Mantzaflaris, A., Jüttler, B., Khoromskij, B. and Langer, U. [2014]. Matrix generation in isogeometric analysis by low rank tensor approximation, *in* J.-D. Boissonnat, A. Cohen, O. Gibaru, C. Gout, T. Lyche, M.-L. Mazure and L. L. Schumaker (eds), *Curves and Surfaces*, Lecture Notes in Computer Science, pp. 321–340.

[22] Peters, J. [1991]. Smooth interpolation of a mesh of curves, *Constr. Approx.* **7**(1): 221–246.

[23] Piegl, L. and Tiller, W. [2012]. *The NURBS book*, Springer.

[24] Randrianarivony, M. [2011]. On transfinite interpolations with respect to convex domains, *Comput. Aided Geom. Design* **28**(2): 135–149.

[25] Rvachev, V. L., Sheiko, T. I., Shapiro, V. and Tsukanov, I. [2001]. Transfinite interpolation over implicitly defined sets, *Comput. Aided Geom. Design* **18**(3): 195–220.

[26] Sabin, M. [1994]. Transfinite surface interpolation, *Proceedings of the 6th IMA Conference on the Mathematics of Surfaces*, Clarendon Press, pp. 517–534.

[27] Schneider, J. [2010]. Error estimates for two-dimensional cross approximation, *J. Approx. Theory* **162**: 1685–1700.

[28] Stoer, J. and Bulirsch, R. [2013]. *Introduction to Numerical Analysis*, Vol. 12, Springer.

[29] Várady, T., Rockwood, A. and Salvi, P. [2011]. Transfinite surface interpolation over irregular n-sided domains, *Comput. Aided Design* **43**(11): 1330–1340.

[30] Woodward, C. D. [1987]. Cross-sectional design of B-spline surfaces, *Comput. & Graphics* **11**(2): 193–201.