PADOVA UNIVERSITY PRESS

Dolomites Research Notes on Approximation

Special Issue SA2022 – *Software for Approximation 2022*, Volume 15 · 2022 · Pages 81–91

DISC: an adaptive numerical Differentiator by local polynomial Interpolation on multivariate SCattered data*

Francesco Dell'Accio[a] · Filomena Di Tommaso[a] · Najoua Siar[b] · Marco Vianello[c]

Abstract

We present an adaptive pointwise numerical differentiator on multivariate scattered data, based on local polynomial interpolation in the Taylor basis at discrete Leja points. We also provide the corresponding Matlab code for bivariate differentiation and a demo with the computation of first and second partial derivatives of Franke's test function.

1 Differentiation via local interpolation on scattered data

In the recent paper [7] we have proposed a multivariate numerical differentiation method, based on local polynomial interpolation in the Taylor basis at discrete extremal sets [2] extracted from multivariate scattered data. Such an approach, which complements other techniques that adopt least-square approximation or different function spaces (cf. e.g. [1, 4, 10]), has shown a very good accuracy on smooth functions with uniform or quasi-uniform samples.

In order to clarify the theoretical foundations, we state the following result that summarizes the relevant error estimates. In what follows, we shall denote by $\mathbb{P}_d(\mathbb{R}^s)$ the space of s-variate polynomials with total degree not exceeding d, by $B_h(\overline{\mathbf{x}})$ the Euclidean ball of radius h centered at $\overline{\mathbf{x}}$, by $D^\alpha = \partial_{x_1}^{\alpha_1}\cdots\partial_{x_s}^{\alpha_s}$ the differentiation operator with multi-index $\alpha=(\alpha_1,\ldots,\alpha_s)$ of length $|\alpha|=\alpha_1+\cdots+\alpha_s$, and by $C^{d,1}(\Omega)$ the space of C^d functions with Lipschitz-continuous derivatives of order d on a convex body Ω, equipped with the seminorm $\|f\|_{C^{d,1}(\Omega)} = \sup\left\{\frac{|D^\alpha f(\mathbf{u})-D^\alpha f(\mathbf{v})|}{\|\mathbf{u}-\mathbf{v}\|_2} : \mathbf{u},\mathbf{v}\in\Omega,\ \mathbf{u}\neq\mathbf{v},\ |\alpha|=d\right\}$. Moreover, we consider the graded lexicographical ordering of multi-indices.

Theorem 1.1. *Let $\Omega\subset\mathbb{R}^s$ be a convex body, $\overline{\mathbf{x}}\in\Omega$, $f\in C^{d,1}(\Omega)$, and $p_d[y,X]\in\mathbb{P}_d(\mathbb{R}^s)$ the interpolating polynomial at a unisolvent subset $X=\{\mathbf{x}_1,\ldots,\mathbf{x}_{m_d}\}\subset\mathcal{N}_h=B_h(\overline{\mathbf{x}})\cap\Omega$, where $m_d=\binom{s+d}{s}=dim(\mathbb{P}_d(\mathbb{R}^s))$ and $y=\{y_i\}=f(X)=\{f(\mathbf{x}_i)\}$. Moreover, let $\tilde{y}=\{\tilde{y}_i\}$ be a perturbed sample of f at X, where $\|y-\tilde{y}\|_\infty\le\varepsilon$.*

Then the following pointwise differentiation error estimate holds

$$\left|D^\nu f(\overline{\mathbf{x}})-D^\nu p_d[\tilde{y},X](\overline{\mathbf{x}})\right| \le \lambda_{\nu,h,d}(\overline{\mathbf{x}})\left(\frac{s^d}{(d-1)!}\|f\|_{C^{d,1}(\mathcal{N}_h)}h^{d+1}+\varepsilon\right),\ |\nu|\le d\,, \tag{1}$$

$$\lambda_{\nu,h,d}(\overline{\mathbf{x}}) = \nu_1!\ldots\nu_s!\,h^{-|\nu|}\|\rho_{\nu,h,d}(\overline{\mathbf{x}})\|_1\,,$$

where $\rho_{\nu,h,d}(\overline{\mathbf{x}})$ denotes the row indexed by ν of the inverse Vandermonde matrix $(V_{d,h}(X))^{-1}$, with $V_{d,h}(X)=\left[\left(\frac{\mathbf{x}_i-\overline{\mathbf{x}}}{h}\right)^\alpha\right]$, $1\le i\le m_d$, $|\alpha|\le d$.

For the reader's convenience we sketch now the proof, that can be found in [7], to which we refer for all the necessary definitions and full details. For the sake of concision, we adopt the usual notation with multi-indices where factorials and powers are interpreted as componentwise products, e.g. $\alpha!=\alpha_1!\ldots\alpha_s!$ and $\mathbf{x}^\alpha=x_1^{\alpha_1}\ldots x_s^{\alpha_s}$.

Proof. First, we recall that the interpolating polynomial can be written as

$$p_d[y,X](\mathbf{x}) = \sum_{i=1}^{m_d} y_i\ell_i(\mathbf{x})$$

in the Lagrange cardinal basis $\{\ell_i(\mathbf{x})\}$, and that each $\ell_i(\mathbf{x})$ can be written in a scaled Taylor-like basis (originally proposed in [5]) as

$$\ell_i(\mathbf{x}) = \sum_{|\alpha|\le d} u_{\alpha i}\left(\frac{\mathbf{x}-\overline{\mathbf{x}}}{h}\right)^\alpha$$

*The preface of this special issue to which the article belongs is given in [3].

[a] Department of Mathematics and Computer Science, University of Calabria, Rende (CS), Italy
[b] Department of Mathematics, Ibn Tofail University, Kenitra, Morocco
[c] University of Padova, Italy

where $\{u_{\alpha i}\}$ is the i-th column of the inverse Vandermonde matrix in such a basis. Then,

$$D^\nu p_d[y,X](\mathbf{x}) = \sum_{i=1}^{m_d} y_i D^\nu \ell_i(\mathbf{x}) = \sum_{i=1}^{m_d} y_i \sum_{\alpha_j \geq \nu_j \, \forall j} u_{\alpha i} \frac{\alpha!}{(\alpha-\nu)!} h^{-|\alpha|} (\mathbf{x}-\overline{\mathbf{x}})^{\alpha-\nu},$$

so that for any vector η of approximate function values we get the estimate

$$\begin{aligned}
\left| D^\nu p_d[y,X](\mathbf{x}) - D^\nu p_d[\eta,X](\mathbf{x}) \right| &\leq \sum_{i=1}^{m_d} |y_i - \eta_i| \left| D^\nu \ell_i(\mathbf{x}) \right| \\
&= \sum_{i=1}^{m_d} |y_i - \eta_i| \left| \sum_{\alpha_j \geq \nu_j \, \forall j} u_{\alpha i} \frac{\alpha!}{(\alpha-\nu)!} h^{-|\alpha|} (\mathbf{x}-\overline{\mathbf{x}})^{\alpha-\nu} \right| \\
&\leq \|y-\eta\|_\infty \sum_{i=1}^{m_d} \left| \sum_{\alpha_j \geq \nu_j \, \forall j} u_{\alpha i} \frac{\alpha!}{(\alpha-\nu)!} h^{-|\alpha|} (\mathbf{x}-\overline{\mathbf{x}})^{\alpha-\nu} \right|
\end{aligned}$$

and in particular for $\mathbf{x} = \overline{\mathbf{x}}$ (where only the term of the interior sum with $\alpha = \nu$ does not vanish)

$$\left| D^\nu p_d[y,X](\overline{\mathbf{x}}) - D^\nu p_d[\eta,X](\overline{\mathbf{x}}) \right| \leq \|y-\eta\|_\infty \, \nu! h^{-|\nu|} \sum_{i=1}^{m_d} |u_{\nu i}| \,. \tag{2}$$

In order to estimate the pointwise differentiation error in (1), considering the Taylor polynomial of degree d for f centered at $\overline{\mathbf{x}}$, say $t_d(\mathbf{x})$, and setting $\tau = \{t_d(\mathbf{x}_i)\}$, we can write

$$\begin{aligned}
\left| D^\nu f(\overline{\mathbf{x}}) - D^\nu p_d[\tilde{y},X](\overline{\mathbf{x}}) \right| &\leq \left| D^\nu f(\overline{\mathbf{x}}) - D^\nu t_d(\overline{\mathbf{x}}) \right| + \left| D^\nu t_d(\overline{\mathbf{x}}) - D^\nu p_d[\tau,X](\overline{\mathbf{x}}) \right| \\
&+ \left| D^\nu p_d[\tau,X](\overline{\mathbf{x}}) - D^\nu p_d[y,X](\overline{\mathbf{x}}) \right| + \left| D^\nu p_d[y,X](\overline{\mathbf{x}}) - D^\nu p_d[\tilde{y},X](\overline{\mathbf{x}}) \right| .
\end{aligned} \tag{3}$$

Now, the first summand in (3) vanishes by construction of the Taylor polynomial while the second vanishes since $t_d = p_d[\tau,X]$ (a polynomial in $\mathbb{P}_d$ coincides with its interpolator of the same degree by unisolvency of X). On the other hand, taking $\eta = \tilde{y}$ in (2) we get the following estimate of the fourth summand

$$\left| D^\nu p_d[y,X](\overline{\mathbf{x}}) - D^\nu p_d[\tilde{y},X](\overline{\mathbf{x}}) \right| \leq \varepsilon \, \nu! h^{-|\nu|} \sum_{i=1}^{m_d} |u_{\nu i}| \,.$$

Finally, taking $\eta = \tau$ in (2) we can estimate the third summand as

$$\left| D^\nu p_d[\tau,X](\overline{\mathbf{x}}) - D^\nu p_d[y,X](\overline{\mathbf{x}}) \right| \leq \max |t_d(\mathbf{x}_i) - f(\mathbf{x}_i)| \, \nu! h^{-|\nu|} \sum_{i=1}^{m_d} |u_{\nu i}| \,,$$

from which we conclude, by recalling the well-known estimate of the multivariate Taylor formula remainder

$$|t_d(\mathbf{x}) - f(\mathbf{x})| \leq \frac{s^d}{(d-1)!} \|f\|_{C^{d,1}(\mathcal{N}_h)} h^{d+1} \,,$$

valid for any $\mathbf{x}$ in a convex neighborhood of $\overline{\mathbf{x}}$ (cf. [8, 11]). □

It is also worth observing that the approximate derivative with multi-index ν can be conveniently computed as

$$D^\nu p_d[y,X](\overline{\mathbf{x}}) = \nu! h^{-|\nu|} c_\nu \approx D^\nu f(\overline{\mathbf{x}}) \tag{4}$$

where c_ν is the interpolation coefficient corresponding to $\left(\frac{\mathbf{x}-\overline{\mathbf{x}}}{h}\right)^\nu$, i.e. the component indexed by ν of the vector c that solves the linear system $V_{d,h}(X)c = y$.

In the next section, we propose an adaptive algorithm that implements pointwise numerical differentiation by local polynomial interpolation, along the lines given by the error analysis in Theorem 1.1. Key tools are the use of the scaled Taylor-like basis and the computation of discrete Leja points for local interpolation. In the last section we test the numerical differentiator in computing first and second partial derivatives of Franke's test function. The corresponding Matlab implementation is freely available to the scientific community.

2 An adaptive numerical differentiator

We describe now an adaptive pointwise differentiation algorithm, starting from a set of scattered sample sites $S \subset \Omega = [0,1]^s$ (by an affine change of variables, it works on a general box). Following the approach we have proposed in [7], given a local interpolation degree d and a ball $B_h(\overline{\mathbf{x}})$ such that $card(S \cap B_h(\overline{\mathbf{x}})) \geq m_d = \binom{s+d}{s}$, we choose in $S_h = S \cap B_h(\overline{\mathbf{x}})$ a set X of m_d *discrete Leja points* as interpolation set. These are found by a greedy maximization of the absolute value of the Vandermonde determinant, that can be obtained by simply computing a LU factorization with row pivoting of the rectangular Vandermonde matrix for degree d at S_h; cf. [2].

Clearly, this is possible whenever the local scattered set S_h is $\mathbb{P}_d$-determining, i.e. polynomials in $\mathbb{P}_d$ vanishing there vanish everywhere, or equivalently the Vandermonde matrix for degree d at S_h has full rank, that is S_h contains a unisolvent interpolation set. Though this property is not valid for an arbitrary scattered distribution, it is almost sure with a uniform distribution.

Consider indeed for simplicity a uniform random vector of dimension m_d in $\Omega = [0,1]^s$, that are m_d independent s-variate uniform random variables, or equivalently sm_d independent univariate uniform random variables. Notice that the Vandermonde determinant, as every determinant, is a polynomial of degree m_d in the matrix $m_d \times m_d$ entries, that here in turn are polynomials of degree not exceeding d in the $\{\mathbf{x}_i\}$, and thus the Vandermonde determinant is a polynomial of degree $k = dm_d$ in $t = sm_d$ variables (the interpolation points coordinates). Now, as it is well-known in full generality, the zero set of any polynomial of degree $k \geq 1$ in t variables has Lebesgue measure zero in $\mathbb{R}^t$ (cf. e.g. [9]), hence the probability that the Vandermonde determinant of m_d uniform random points in Ω vanishes is zero, i.e. m_d uniform random points in Ω are almost surely unisolvent for polynomial interpolation of total degree d.

We stress that the choice of discrete Leja points not only ensures unisolvency whenever the local scattered set S_h is $\mathbb{P}_d$-determining, but is also aimed at keeping small the entries of the inverse Vandermonde matrix (which are cofactors divided by the Vandermonde determinant), and thus also the relevant row 1-norm in estimate (1). Indeed, it is worth observing that the stability constant $\lambda_{\nu,h,d}(\overline{\mathbf{x}}) = \sum_{i=1}^{m_d} |D^\nu \ell_i(\overline{\mathbf{x}})|$ is nothing but the stability function of differentiation by interpolation on X, computed at $\overline{\mathbf{x}}$ (for $\nu = 0$ it is the Lebesgue function of X at $\overline{\mathbf{x}}$).

We observe that discrete Leja points are not the only computable point set trying to maximize the absolute value of the Vandermonde determinant, another relevant set are the approximate Fekete points, corresponding to a QR factorization with column pivoting of the transposed Vandermonde matrix; cf. [2]. Though approximate Fekete points could show a lower Lebesgue constant, discrete Leja points have an additional property that makes them appealing from the computational point of view, namely they form a *sequence* if the polynomial basis has the lexicographical order. This means that the first $\binom{k+s}{s}$ among $\binom{d+s}{s}$ discrete Leja points are just the discrete Leja points for interpolation of degree k in s variables, $1 \leq k \leq d$, therefore once computed such points for degree d we can interpolate at any lower degree; notice that this property, which will be used in the algorithm to construct an a posteriori error estimate, does not hold with the approximate Fekete points that have to be computed separately for each degree.

We can now sketch a pseudo-code of the differentiation algorithm. The corresponding Matlab code, written for functions of $s = 2$ variables, is available at [6].

DISC: adaptive Differentiator by local polynomial Interpolation on SCattered data

- INPUT: the scattered data $(S, f(S))$, the evaluation point $\overline{\mathbf{x}}$, the differentiation multi-index ν, the initial degree d_0 and the maximum degree $dmax$, the degree step σ and the error control step δ (constraint: $d_0 - \delta \geq |\nu|$), the maximum allowed radius $hmax < 1$
- INIZIALIZATION: $d := d_0$, $est_{min} := realmax$, $h := 0$
- REPEAT

  % selecting a suitable interpolation ball

1. find the minimum radius h_d such that $card(S_{h_d}) = card(S \cap B_{h_d}(\overline{\mathbf{x}})) \geq m_d = \binom{s+d}{s}$
2. $h := \max\{h, h_d\}$

   IF $h \leq hmax$ THEN

   REPEAT

   3. compute $V_{d,h}(S_h) = \left[\left(\frac{\xi - \overline{\mathbf{x}}}{h}\right)^\alpha\right] \in \mathbb{R}^{card(S_h) \times m_d}$, $\xi \in S_h$, $|\alpha| \leq d$
   4. $r := rank(V_{d,h}(S_h))$
   5. IF $r < m_d$ THEN $h := (1+h)/2$, $S_h := S \cap B_h(\overline{\mathbf{x}})$ ENDIF

   UNTIL $r \geq m_d \vee h > hmax$

   ENDIF

   % differentiating by Leja-like interpolation and estimating the error

   IF $h \leq hmax$ THEN

   6. extract m_d discrete Leja points $X_d = \{\mathbf{x}_1, \ldots, \mathbf{x}_{m_d}\}$ from S_h by the algorithm in [2]
   7. compute the Vandermonde matrices $V_{d,h}(X_d) = \left[\left(\frac{\mathbf{x}_i - \overline{\mathbf{x}}}{h}\right)^\alpha\right] \in \mathbb{R}^{m_d \times m_d}$, $1 \leq i \leq m_d$, $|\alpha| \leq d$ and $V_{d-\delta,h}(X_{d-\delta}) \in \mathbb{R}^{m_{d-\delta} \times m_{d-\delta}}$, $1 \leq i \leq m_{d-\delta}$, $|\alpha| \leq d - \delta$, where $X_{d-\delta} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{m_{d-\delta}}\}$

8. compute the interpolation coefficients for degree d and $d-\delta$ by solving the couple of linear systems $V_{d,h}(X_d)c = f(X_d)$ and $V_{d-\delta,h}(X_{d-\delta})\gamma = f(X_{d-\delta})$
9. compute the approximate derivatives $D^\nu_d := \nu! h^{-|\nu|} c_\nu$ and $D^\nu_{d-\delta} := \nu! h^{-|\nu|}\gamma_\nu$, and the error estimate $est := |D^\nu_d - D^\nu_{d-\delta}|$
10. IF $est < est_{min}$ THEN
$est_{min} := est$, $\tilde{D}^\nu f := D^\nu_{d-\delta}$
$d^* := d$, $h^* := h$
ENDIF
11. $d := d + \sigma$

ENDIF

UNTIL $d > dmax \vee h > hmax$

- OUTPUT: $d > d_0 \Rightarrow \tilde{D}^\nu f$, est_{min}: $|\tilde{D}^\nu f - D^\nu f(\overline{\mathbf{x}})| \approx est_{min}$

Some comments on the algorithm are in order. First, we observe that Theorem 1.1 gives qualitatively the effect of h, d and ν in the error analysis, but is hard to use in practice, since one should have at hand an estimate of the function seminorm, which in general is out of reach. We have then adopted an a posteriori error bound

$$est(\overline{\mathbf{x}}) = |D^\nu_d - D^\nu_{d-\delta}| \approx e(\overline{\mathbf{x}}) = |D^\nu_{d-\delta} - D^\nu f(\overline{\mathbf{x}})| \tag{5}$$

that, though essentially empirical, has shown a good mean accuracy in all the numerical tests (and can be tuned by the error control step δ). Notice that, being a sequence, the discrete Leja points can be computed once in **6.** and then used in **7.**$-$**8.** for different degrees.

Moreover, we stress that the algorithm is *adaptive*, in the sense that given the scattered data and the maximum allowed degree and radius, it increases the local interpolation degree by adapting the local radius until possible, updating the best error estimate and the corresponding derivative approximation during the computational process. That is, it tries to give the best possible approximation of the required partial derivative given the scattered data, the bounds $dmax$ and $hmax$, the degree increase step σ and control step δ.

3 Numerical tests

In the first experiment, we illustrate the performance of the differentiator DISC by applying it to the popular Franke's test function on the square $[0,1]^2$

$$\begin{aligned} f_1(x,y) \quad &= 0.75\exp\Big(-\frac{(9x-2)^2+(9y-2)^2}{4}\Big) + 0.50\exp\Big(-\frac{(9x-7)^2+(9y-3)^2}{4}\Big) \\ &+0.75\exp\Big(-\frac{(9x+1)^2}{49} - \frac{(9y+1)^2}{10}\Big) - 0.20\exp\Big(-(9x-4)^2-(9y-7)^2\Big), \end{aligned} \tag{6}$$

whose graph is plotted in Figure 1. To this purpose, in Figures 2-3 we plot the absolute error $e(\overline{\mathbf{x}})$ and its a posteriori estimate $est(\overline{\mathbf{x}})$ in (5) for 100 random points $\overline{\mathbf{x}} = (x,y)$ in $\Omega = [0,1]^2$, with differentiation indexes $\nu = (1,0),(1,1),(0,2)$, i.e. $\partial/\partial x$ (top), $\partial^2/\partial x \partial y$ (central), $\partial^2/\partial y^2$ (bottom), and three different sets of Halton and random sampling points. The tests have been carried out with parameters setting $d_0 = 5$, $\sigma = 3$, $\delta = 2$, $hmax = 0.8$ and $dmax = \lfloor(\sqrt{8N+1}-3)/2)\rfloor$ that is the maximum degree d such that $m_d = (d+1)(d+2)/2 \le N$, where N is the overall number of sampling points. The corresponding demo is available within the complete package at [6].

Observe that $est(\overline{\mathbf{x}})$ is close to the true error $e(\overline{\mathbf{x}})$ in most cases. In very few cases it substantially underestimates or overestimates the error, by one or seldom two orders of magnitude. The resulting mean estimate (dashed line) is a good approximation of the mean error (solid line) and, as expected, both decrease by increasing the number of sampling points.

On the other hand, it turns out that the errors over the mean tend to occur near the boundary (but not everywhere). This fact, already investigated in [7], can be ascribed to a boundary effect, since the neighborhoods $B_h(\overline{\mathbf{x}}) \cap \Omega$ have less sample points so that the maximum allowed interpolation degree is smaller and the quality of the extracted interpolation points can be worse, combined with the local behavior of the sampled function, in particular of the local seminorm $\|f\|_{C^{d,1}}$.

In order to show the perfomance of the method in the case of C^k functions, $k \in \mathbb{N}$, in the second experiment, we apply the differentiator DISC to the functions

$$f_j(x,y) = \|(x,y)-(0.5,0.5)\|_2^{2j-1}, \quad j = 2,3,4, \tag{7}$$

which are of class C^{2j-2}, respectively. To this purpose, in Figures 4-6 we plot the absolute error $e(\overline{\mathbf{x}})$ and its a posteriori estimate $est(\overline{\mathbf{x}})$ in (5) on the uniform grid of 121 points of $\Omega = [0,1]^2$, which includes the point $(0.5,0.5)$ which is singular for the derivatives of order $2j-1$. We use differentiation indexes $\nu = (1,0),(1,1),(0,2)$ and three different sets of Halton sampling points. The tests have been carried out with the same parameters setting of the first experiment.

The behaviour of the error and its estimate is similar to the one of the first experiment for points far from the point $(0.5,0.5)$. This is not surprising since in a suitable neighborhood of those points all functions are C^∞. As soon as we approach the singular point $(0.5,0.5)$, we can observe some loss of accuracy, which in any case is never unsatisfactory. In particular, the estimates relative to the point $(0.5,0.5)$ correspond to the points of abscissa 0.5 in Figures 4-6.

In order to show the response of the differentiation algorithm to measurement errors, in Figure 7 we consider the case of a perturbed sample at a set S of 1000 Halton points. To this purpose, we have taken perturbed values

$$\tilde{f}(S) = f(S) + \varepsilon\, U(-1,1)\,, \tag{8}$$

U being a uniform random array in $[-1,1]^{card(S)}$. We see that about two orders of magnitude with respect to ε are lost for the first derivative $\partial/\partial x$ (top), and up to three orders for the second derivatives $\partial^2/\partial x\partial y$ (central) and $\partial^2/\partial y^2$ (bottom). Such precision losses are partially explained by the size of the pointwise stability constant $\lambda_{\nu,h^*,d^*}(\overline{\mathbf{x}})$ in (1) for the exit values from Algorithm DISC of the radius and the degree corresponding to the minimal error estimate. Notice that h^* and d^* depend on $\overline{\mathbf{x}}$, on the sampling set S and on the function f. The quantity $\lambda_{\nu,h^*,d^*}(\overline{\mathbf{x}})$ is plotted in Figure 8, ordering again the control points by distance from the boundary (such an amplification factor, however, is clearly an overestimate of the actual effect).

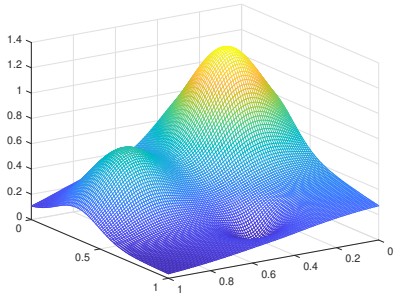

Figure 1: Franke's test function.

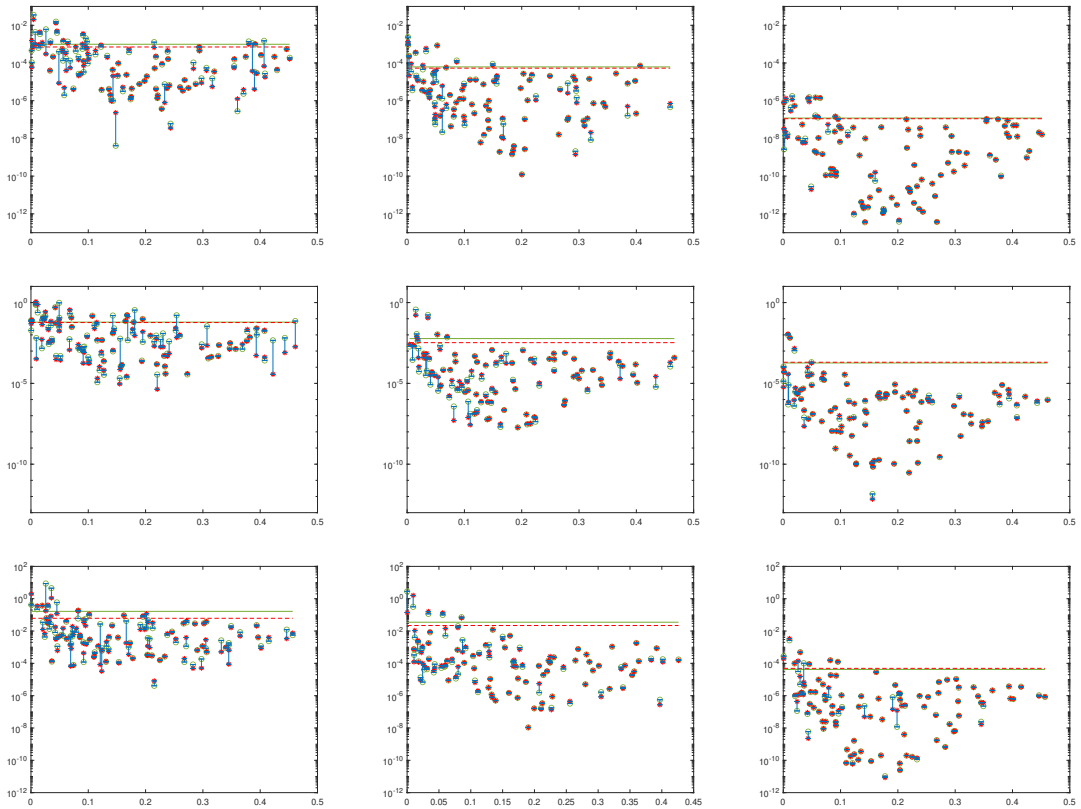

Figure 2: Pointwise differentiation errors (circles) and estimates (asterisks) with Franke's test function on 100 random points in $[0,1]^2$ ordered by distance from the boundary, for $\partial/\partial x$ (top), $\partial^2/\partial x\partial y$ (central) and $\partial^2/\partial y^2$ (bottom), with 500 (left), 1000 (center) and 2000 (right) Halton sampling points (solid line: mean error; dashed line: mean of the estimate (5)).

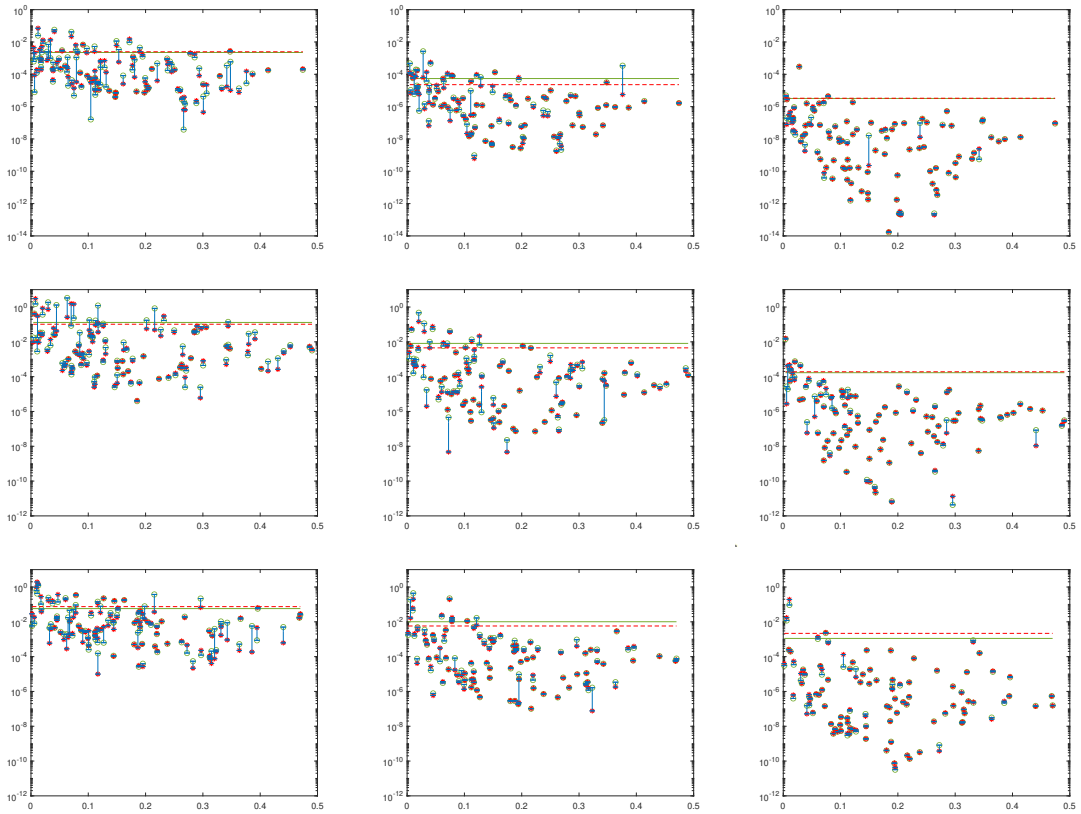

Figure 3: Pointwise differentiation errors (circles) and estimates (asterisks) with Franke's test function on 100 random points in $[0,1]^2$ ordered by distance from the boundary, for $\partial/\partial x$ (top), $\partial^2/\partial x \partial y$ (central) and $\partial^2/\partial y^2$ (bottom), with 500 (left), 1000 (center) and 2000 (right) random sampling points (solid line: mean error; dashed line: mean of the estimate (5)).

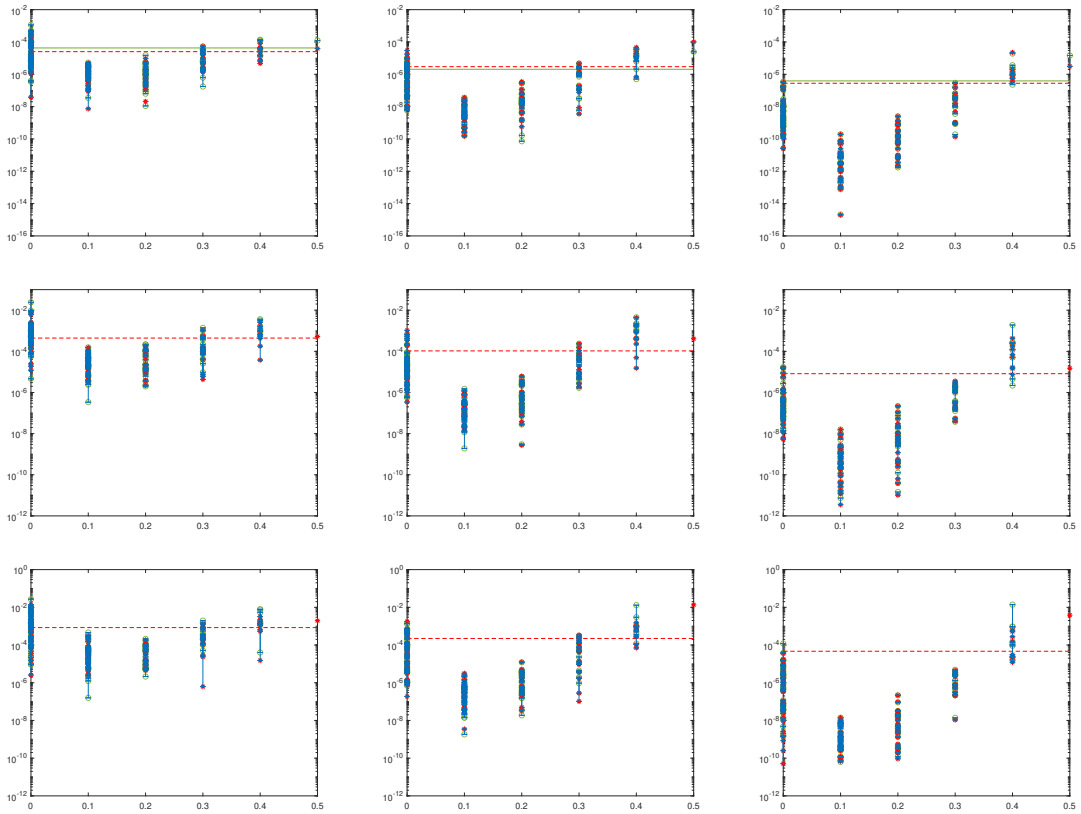

Figure 4: Pointwise differentiation errors (circles) and estimates (asterisks) with function $f_2(x,y)$ on the uniform grid of 121 points of $[0,1]^2$ ordered by distance from the boundary, for $\partial/\partial x$ (top), $\partial^2/\partial x \partial y$ (central) and $\partial^2/\partial y^2$ (bottom), with 500 (left), 1000 (center) and 2000 (right) Halton sampling points (solid line: mean error; dashed line: mean of the estimate (5)).

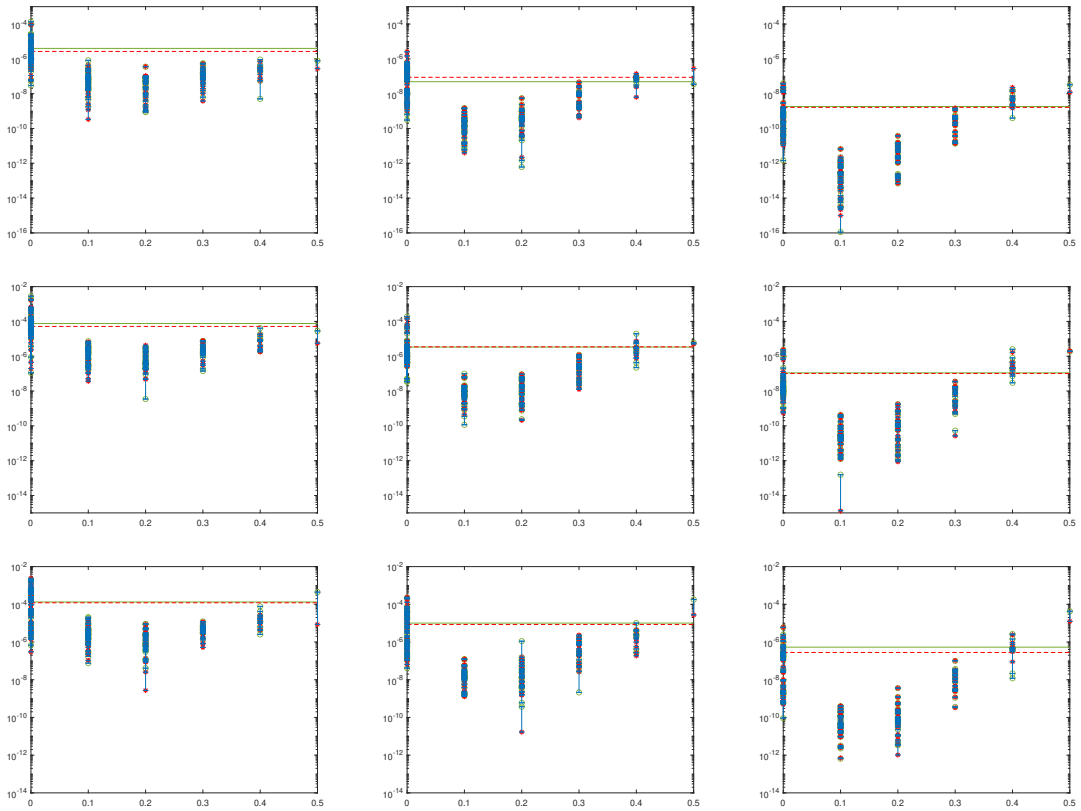

Figure 5: Pointwise differentiation errors (circles) and estimates (asterisks) with function $f_3(x,y)$ on the uniform grid of 121 points of $[0,1]^2$ ordered by distance from the boundary, for $\partial/\partial x$ (top), $\partial^2/\partial x \partial y$ (central) and $\partial^2/\partial y^2$ (bottom), with 500 (left), 1000 (center) and 2000 (right) Halton sampling points (solid line: mean error; dashed line: mean of the estimate (5)).

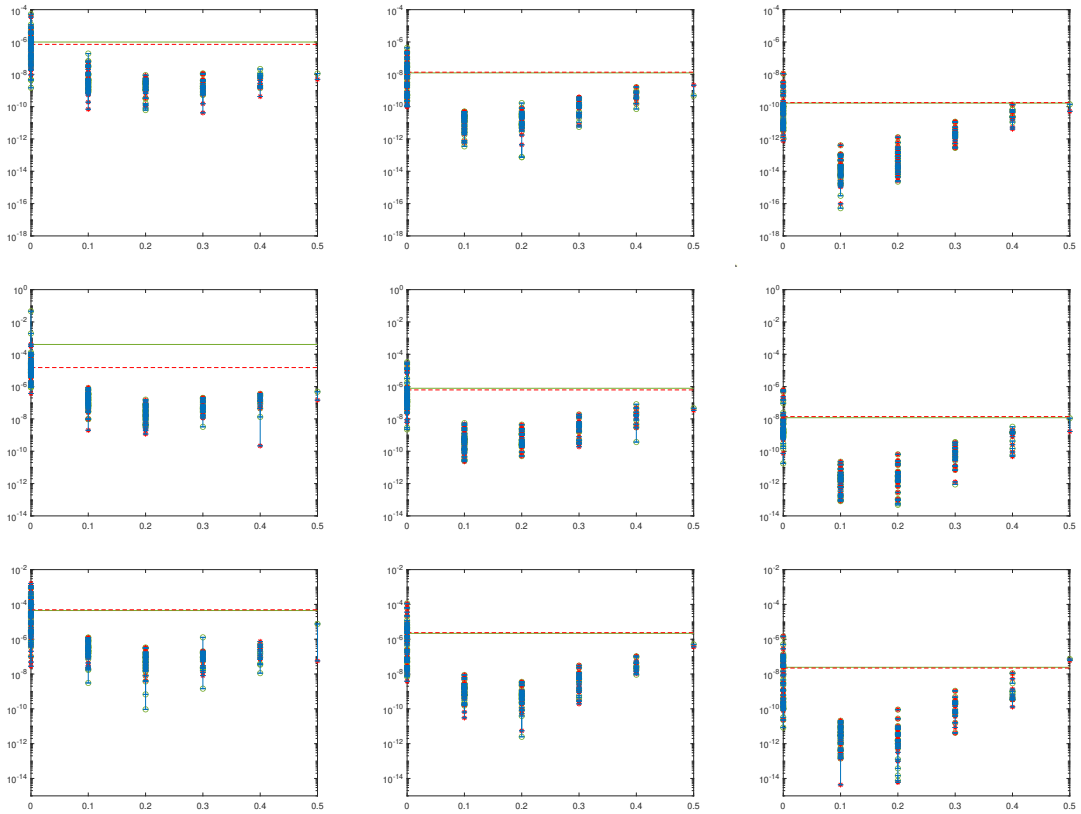

Figure 6: Pointwise differentiation errors (circles) and estimates (asterisks) with function $f_4(x,y)$ on the uniform grid of 121 points of $[0,1]^2$ ordered by distance from the boundary, for $\partial/\partial x$ (top), $\partial^2/\partial x \partial y$ (central) and $\partial^2/\partial y^2$ (bottom), with 500 (left), 1000 (center) and 2000 (right) Halton sampling points (solid line: mean error; dashed line: mean of the estimate (5)).

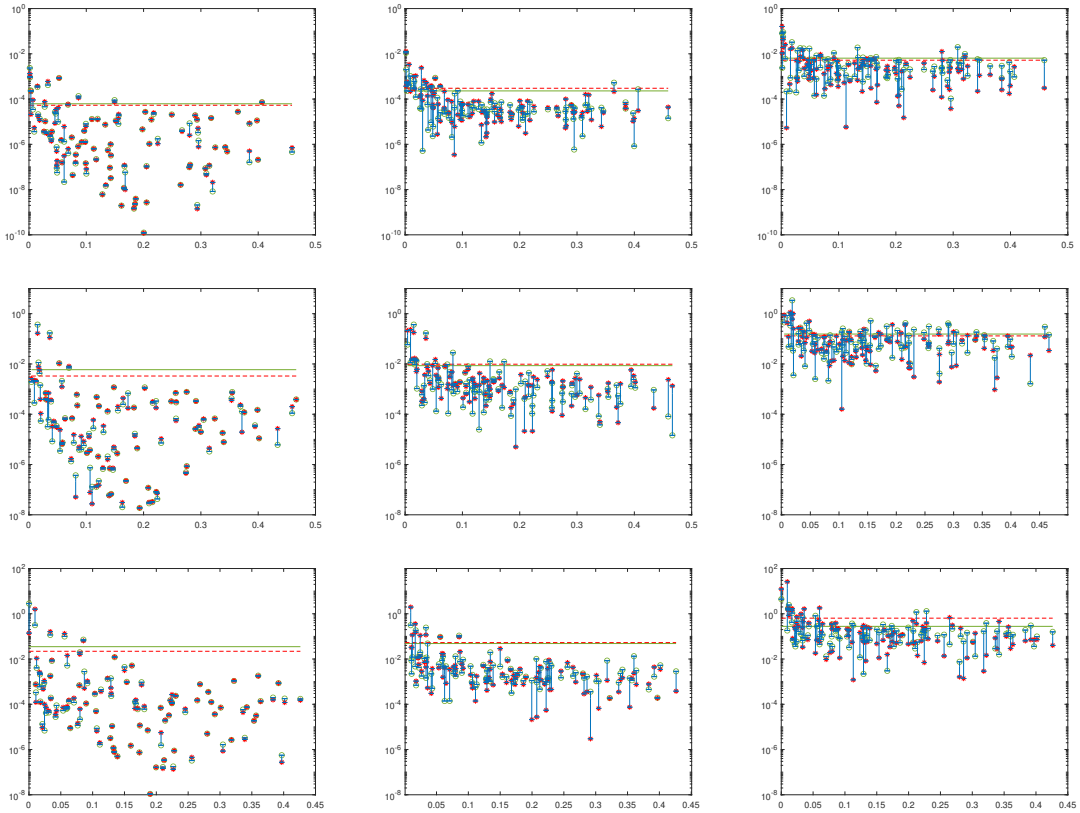

Figure 7: Pointwise differentiation errors (circles) and estimates (asterisks) with Franke's test function on 100 random points in $[0,1]^2$ ordered by distance from the boundary, for $\partial/\partial x$ (top), $\partial^2/\partial x \partial y$ (central) and $\partial^2/\partial y^2$ (bottom), with 1000 Halton sampling points and perturbed function values as in (8), where $\varepsilon = 0$ (left), $\varepsilon = 10^{-6}$ (center) and $\varepsilon = 10^{-4}$ (right) (solid line: mean error; dashed line: mean of the estimate (5)).

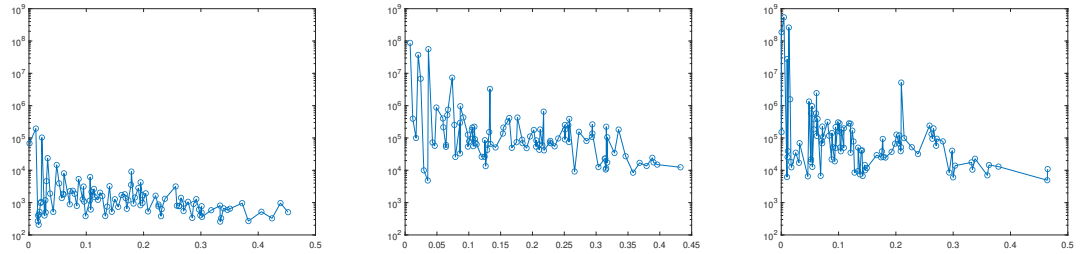

Figure 8: Pointwise values of the stability constant $\lambda_{\gamma,h^*,d^*}(\overline{\mathbf{x}})$ in (1) for the exit values of h^* and d^* from Algorithm DISC on 100 random points in $[0,1]^2$ ordered by distance from the boundary, using a sample of 1000 Halton points for $\partial/\partial x$ (left), $\partial^2/\partial x \partial y$ (central) and $\partial^2/\partial y^2$ (right).

Acknowledgments

This research has been partially supported by the GNCS-INdAM 2020 Projects "Interpolation and smoothing: theoretical, computational and applied aspects with emphasis on image processing and data analysis" and "Multivariate approximation and functional equations for numerical modelling", and has been accomplished within RITA "Research ITalian network on Approximation" and the UMI Group TAA "Approximation Theory and Applications". The third author was supported by the National Center for Scientific and Technical Research (CNRST-Morocco) as part of the Research Excellence Awards Program (No. 103UIT2019). The fourth author was partially supported by the DOR funds and the biennial project BIRD 192932 of the University of Padova.

References

[1] J.A. Belward, I.W. Turner, and M. Ilić. On derivative estimation and the solution of least squares problems. *Journal of Computational and Applied Mathematics*, 222(2):511–523, 2008.

[2] L. Bos, S. De Marchi, A. Sommariva, and M. Vianello. Computing multivariate Fekete and Leja points by numerical linear algebra. *SIAM Journal on Numerical Analysis*, 48(5):1984–1999, 2010.

[3] R. Cavoretto, A. De Rossi. Software for Approximation 2022 (SA2022). *Dolomites Res. Notes Approx.*, Special Issue SA2022, 15:i–ii, 2022.

[4] O. Davydov and R. Schaback. Minimal numerical differentiation formulas. *Numerische Mathematik*, 140(3):555–592, 2018.

[5] F. Dell'Accio, F. Di Tommaso, and N. Siar. On the numerical computation of bivariate Lagrange polynomials. *Applied Mathematics Letters*, 112: 106845, 2020.

[6] F. Dell'Accio, F. Di Tommaso, N. Siar, and M. Vianello. DISC: a Matlab adaptive Differentiator by local polynomial Interpolation on bivariate SCattered data. *http://lan.unical.it/software.php*, 2022.

[7] F. Dell'Accio, F. Di Tommaso, N. Siar, and M. Vianello. Numerical differentiation on scattered data through multivariate polynomial interpolation. *BIT Numer. Math.*, 62: 773–801, 2022.

[8] R. Farwig. Rate of convergence of Shepard's global interpolation formula. *Mathematics of Computation*, 46(174):577–590, 1986.

[9] H. Federer. *Geometric measure theory*. Springer, 1969.

[10] N. Mai-Duy and T. Tran-Cong. Approximation of function and its derivatives using radial basis function networks. *Applied Mathematical Modelling*, 27(3):197–220, 2003.

[11] S. Waldron. Multipoint Taylor formulæ. *Numerische Mathematik*, 80(3):461–494, 1998.