Journal of Scientific Computing (2023) 97:37
https://doi.org/10.1007/s10915-023-02348-6

Discontinuity Detection by Null Rules for Adaptive Surface Reconstruction

Cesare Bracco[1] · Francesco Calabrò[2] · Carlotta Giannelli[1]

Received: 9 October 2022 / Revised: 21 July 2023 / Accepted: 3 September 2023

Abstract

We present a discontinuity detection method based on the so-called null rules, computed as a vector in the null space of certain collocation matrices. These rules are used as weights in a linear combination of function evaluations to indicate the local behavior of the function itself. By analyzing the asymptotic properties of the rules, we introduce two indicators (one for discontinuities of the function and one for discontinuities of its gradient) by locally computing just one rule. This leads to an efficient and reliable scheme, which allows us to effectively detect and classify points close to discontinuities. We then show how this information can be suitably combined with adaptive approximation methods based on hierarchical spline spaces in the reconstruction process of surfaces with discontinuities. The considered adaptive methods exploit the ability of the hierarchical spaces to be locally refined, and fault detection is a natural way to guide the refinement with low computational cost. A selection of test cases is presented to show the effectiveness of our approach.

Keywords Fault detection · Gradient fault detection · Discontinuity detection · Null rules · Scattered data · Adaptive surface reconstruction

Mathematics Subject Classification 65N35

✉ Cesare Bracco
cesare.bracco@unifi.it

Francesco Calabrò
francesco.calabro@unina.it

Carlotta Giannelli
carlotta.giannelli@unifi.it

[1] Dipartimento di Matematica e Informatica "Ulisse Dini", Università degli Studi di Firenze, Florence, Italy

[2] Dipartimento di Matematica e Applicazioni "Renato Caccioppoli", Università degli Studi di Napoli "Federico II", Naples, Italy

Published online: 25 September 2023

1 Introduction

The detection of discontinuity curves of bivariate functions (and of their gradients) is an important problem that arises in many contexts, like surface reconstruction from scattered data, edge detection in image and geometric processing, geological lineaments extraction, or general sudden changes in data, see for example [1, 5, 8, 16, 24] and references therein. We call *ordinary faults* the discontinuity curves of the function and *gradient fault* the ones of its gradient. Many different techniques have been proposed to address the problem of fault detection, according to the considered application and to the type of discontinuity which needs to be found.

In image processing, edge detection essentially corresponds to finding only the discontinuity curves of the function describing the color of the pixels composing the image (often in greyscale). As a consequence, in this application the data are gridded: they are the pixel locations and the corresponding color values. The goal can be achieved by applying linear filters to the image and then looking for the local maxima of the result, as described e.g. in [13] and more recently in [26]. Another popular approach relies on the definition of an indicator which can be evaluated at each data point to determine whether or not the considered point is close to a fault. The indicator can be based for example on an estimate of the derivative [2] or on the principle of gravitation law [23].

The construction of fault indicators has been widely employed in recent years, being easily applicable to scattered data. In [16, 24] indicators based on coefficients obtained by radial basis function interpolation were successfully used to find ordinary faults. The possibility of defining different indicators enables the identification of both ordinary and gradient faults, like in [5, 6, 8], where the indicators are based on estimates of the gradient and of the Laplacian. At each point, an approximation of the gradient and an approximation of the Laplacian are computed in order to assess if the point is close to an ordinary or a gradient fault. In [5, 6] the method essentially works on gridded data, as in the case of scattered data the function is simply approximated at points on a grid. In [8] both ordinary and gradient fault indicators can be directly applied to scattered data, as they are based on numerical differentiation formulas which do not need any assumptions on the data configuration.

In this paper, we explore for the first time the use of null rules in the framework of fault detection from scattered data with application to (adaptive) surface reconstruction. Null rules are used in numerical quadrature since the paper by Lyness [22], where the construction of nested quadrature rules was addressed. The same rules were then considered for the error detection in adaptive quadrature, see [4, 15, 19]. In the recent paper [12], interested readers can find results concerning the use of null rules to find low regularity intervals of univariate functions and a complete review of the literature. In this paper, we analyze the asymptotic behaviour of null rules and we use the results to construct two indicators for ordinary and gradient faults, whose evaluation requires the computation of only one null rule at each point. We then apply the result to design effective adaptive methods for surface reconstruction from scattered data. While this problem can be addressed with a variety of techniques, see e.g., [3] and references therein, we here focus on using adaptive spline spaces [14, 17, 21] to perform local refinement and obtain high accuracy with only a fraction of the degrees of freedom needed with traditional global refinement schemes. We show that the result of the fault detection algorithm can effectively drive the refinement, as the low regularity areas naturally identify regions where additional degrees of freedom are needed to properly increase the accuracy of the reconstruction. In particular, we consider truncated B-hierarchical splines, an effective choice for surface reconstruction methods [9, 10, 18]. The key advantages of the proposed approach are (1) the possibility to deal with scattered data configurations without additional postprocessing, (2) the definition of two indicators based on the application of a single null rule, and (3) an efficient adaptive spline surface reconstruction scheme driven by fault detection.

The paper is organized as follows. In Sect. 2 we introduce the fault detection problem, as well as basic concepts and properties related to null rules. Our main results for the analysis of the proposed indicators are presented in Sect. 3. The indicators for the detection of the numerical fault lines and the corresponding fault detection algorithm are introduced in Sect. 4, while Sect. 5 discusses how to properly exploit the result of the proposed algorithm for the design of effective surface reconstruction schemes. Section 6 presents a selection of numerical tests on synthetic and geological data sets. Finally, Sect. 7 concludes the paper.

2 Notations and Preliminaries

Let $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ be our target function. Our aim is to detect the curves, usually indicated as *fault curves*, where the function f or its gradient are not continuous.

We assume that $f \in C^2(\Omega_R)$, with $\Omega_R := \Omega \setminus (\mathcal{F}_O \cup \mathcal{F}_G)$, $\mathcal{F}_O = \bigcup_{i=1}^{M_O} \mathcal{C}_i$, $\mathcal{F}_G = \bigcup_{i=1}^{M_G} \mathcal{G}_i$, where $\mathcal{C}_i$, $i = 1, ..., M_O$, and $\mathcal{G}_i$, $i = 1, ..., M_G$, are planar curves in Ω. Without loss of generality we can assume that all the curves are regular: a curve including non regular points can be always split into regular curves. For the same reason, we can also assume that neither $\mathring{\mathcal{C}}_i$ or $\mathring{\mathcal{G}}_i$ includes intersection points. Each *ordinary fault* $\mathcal{C}_i$ is characterized by the condition

$$\lim_{\theta \to 0^+} f(\mathbf{z} + \theta \mathbf{d}_{\mathbf{z}}^L) =: f_{\mathbf{z}}^L \neq f_{\mathbf{z}}^R := \lim_{\theta \to 0^+} f(\mathbf{z} + \theta \mathbf{d}_{\mathbf{z}}^R), \quad \mathbf{z} \in \mathring{\mathcal{C}}_i, \tag{1}$$

where, denoting by $\mathbf{t}_{\mathbf{z}}$ the unit tangent to $\mathcal{C}_i$ in $\mathbf{z}$, $\mathbf{d}_{\mathbf{z}}^L$ and $\mathbf{d}_{\mathbf{z}}^R$ are arbitrary vectors such that $\mathbf{t}_{\mathbf{z}} \times \mathbf{d}_{\mathbf{z}}^L < 0$ and $\mathbf{t}_{\mathbf{z}} \times \mathbf{d}_{\mathbf{z}}^R > 0$. Analogously, each *gradient fault* $\mathcal{G}_i$ is characterized by the condition

$$\lim_{\theta \to 0^+} \nabla f(\mathbf{z} + \theta \mathbf{d}_{\mathbf{z}}^L) =: \mathbf{v}_{\mathbf{z}}^L \neq \mathbf{v}_{\mathbf{z}}^R := \lim_{\theta \to 0^+} \nabla f(\mathbf{z} + \theta \mathbf{d}_{\mathbf{z}}^R), \qquad \forall \mathbf{z} \in \mathring{\mathcal{G}}_i, \tag{2}$$

where the vector $\mathbf{v}_{\mathbf{z}}^L - \mathbf{v}_{\mathbf{z}}^R$ is parallel to the unit normal to $\mathcal{G}_i$ at $\mathbf{z}$. By starting from a set of (scattered) points $\boldsymbol{X} = \{\mathbf{z}_l = (x_l, y_l) : l = 1, \ldots, N\} \subset \mathbb{R}^2$ and the corresponding set of function values $F = \{f(\mathbf{z}_l) : l = 1, ..., N\}$, we want to determine a suitable subset of points in $\boldsymbol{X}$ sufficiently close to the fault lines. From the result of the detection algorithm, one can then reconstruct the discontinuity curves.

In order to detect points close to the fault lines, we propose the use of the so-called *null rules*, by focusing on the bidimensional case. We refer to [12] for additional details on their construction and related use in the unidimensional case.

Definition 1 Given $\boldsymbol{Z} = \{\mathbf{z}_l = (x_l, y_l) \in \mathbb{R}^2, l = 1, ..., n\} \subset \Omega$ and the class of functions $\Phi = \{\phi_i(\mathbf{z}), i = 1, \ldots, q\}$, we define the null rule $\mathcal{N}(\cdot, \boldsymbol{Z}, \Phi)$ on the set $\boldsymbol{Z}$, exact on the class of functions Φ, as

$$\mathcal{N}(f, \boldsymbol{Z}, \Phi) := \sum_{l=1}^{n} w_l f(\mathbf{z}_l) \quad \forall f : \Omega \to \mathbb{R} \quad \text{such that}$$

$$\sum_{l=1}^{n} w_l = 0\,, \qquad \exists h : \; w_h \neq 0\,,$$

$$\mathcal{N}(f, \mathbf{Z}, \Phi) = 0, \forall f \in \Phi.$$

Note that, due to the linearity of the definitions and of the operators, if a rule is exact on a family of functions $\{\phi_i\}$, then it is exact on the full space spanned by the same functions. The existence of null rules is related to the interpolation properties of the functions for which exactness is required, and to the location of points. If we denote by $\Phi \in \mathbf{R}^{q\times n}$ the collocation matrix $[\phi_i(\mathbf{z}_l)]_{i,l}$, then we have

$$\mathcal{N}(\phi_i, \mathbf{Z}, \Phi) = 0 \quad \forall\, i, \qquad \mathbf{w} \neq \mathbf{0} \;\Leftrightarrow\; \exists\, \mathbf{w} \neq \mathbf{0} :\; \Phi\mathbf{w} = \mathbf{0}. \tag{3}$$

In other words, the null rule on the set $\mathbf{Z}$ exact on the class of functions Φ exists if and only if there is a non-trivial vector in the null space of the matrix Φ. While any multiple of a non-trivial solution of the homogeneous linear system (3) is still a non-trivial solution, we consider solutions that have unitary L^1 norm, i.e. $\sum_l |w_l| = 1$.

Usually, exactness is required with respect to polynomials of total degree t, whose basis we denote by $\Pi_t^2 := \{x_l^{\nu_x} y_l^{\nu_y} \,:\, \nu_x, \nu_y \in \mathbb{N}_0 \; s.t. \; \nu_x + \nu_y \leq t\}$. We call *degree of precision of the null rule* the maximum integer t such that the rule is exact for all polynomials of total degree less than or equal to t, that is, such that it satisfies

$$\sum_{l=1}^{n} w_l x_l^{\nu_x} y_l^{\nu_y} = 0 \;\; \forall\, \nu_x, \nu_y \in \mathbb{N}_0 \; s.t. \; \nu_x + \nu_y \leq t. \tag{4}$$

Since $N_P(t) := |\Pi_t^2| = (t+1)(t+2)/2$, we need at most $N_P(t) + 1$ points to have non trivial solutions to the Eq. (4). For example, a null rule with a degree of precision $t = 1$ is then constructed on a set of at most 4 points, while one rule with a degree of precision $t = 2$ requires at most 7 points. In order to guarantee that there exists a unique null rule (up to multiplicative constant factor) we require that the set of points $\mathbf{Z}$ is such that the rank of the collocation matrix Φ is $N_P(t)$, so that the dimension of the null space is exactly 1. In other words, when constructing a null rule with degree of precision t, we assume that $\mathbf{Z}$ has cardinality $N_P(t) + 1$ and has at least a subset which is a unisolvent point set for Π_t^2.

The construction of the null rule depends only on the geometric location of the points, but it is independent of the position and distances between points in the two principal directions, as formalized in the following proposition.

Proposition 2 *Let $\mathbf{Z}$ have cardinality $N_P(t) + 1$ and let it have a subset which is a unisolvent point set for Π_t^2. Let $\mathcal{N}(\cdot, \mathbf{Z}, \Pi_t^2)$ be a null rule with degree of precision t on a set of nodes $\mathbf{Z} = \{\mathbf{z}_l = (x_l, y_l), l = 1, ..., n\}$ with weights $\{w_l\}_{l=1...n}$. Let $\mathbf{d} = (d_x, d_y) \in \mathbb{R}^2$ be an arbitrary vector. Then:*

- *The null rule on the (shifted) set of nodes $\tilde{\mathbf{Z}} = \{\mathbf{z}_l = (x_l + d_x, y_l + d_y), l = 1, ..., n\}$ with a degree of precision t has the same weights;*
- *The null rule on the (shrinked) set of nodes $\tilde{\mathbf{Z}} = \{\mathbf{z}_l = (d_x x_l, d_y y_l), l = 1, ..., n\}$ with a degree of precision t has the same weights.*

Proof The first statement follows from the fact that exactness on monomials implies exactness on general polynomials:

$$\sum_{l\in\mathcal{I}}^{n} w_l x_l^{\nu_1} y_l^{\nu_2} = 0 \,\forall \nu_1, \nu_2 \in \mathbb{N}_0 \; s.t. \; \nu_1 + \nu_2 \leq t \quad \Rightarrow$$

$$\sum_{l\in\mathcal{I}}^{n} w_l (x_l + d_1)^{\nu_1} (y_l + d_2)^{\nu_2} = 0 \,\forall \nu_1, \nu_2 \in \mathbb{N}_0 \; s.t. \; \nu_1 + \nu_2 \leq t.$$

The second statement follows from the properties of homogeneous operator:

$$\sum_{l\in\mathcal{I}}^{n} w_l (d_1 x_l)^{\nu_1} (d_2 y_l)^{\nu_2} = \sum_{l\in\mathcal{I}}^{n} w_l d_1^{\nu_1} x_l^{\nu_1} d_2^{\nu_2} y_l^{\nu_2} = d_1^{\nu_1} d_2^{\nu_2} \sum_{l\in\mathcal{I}}^{n} w_l x_l^{\nu_1} y_l^{\nu_2} = 0.$$

□

3 Main Results: Asymptotic Behaviour of Null Rules

In this section, we prove our main results on the behaviour of a null rule applied to a point near a fault. In what follows, when the null rule $\mathcal{N}(\cdot, \boldsymbol{Z}, \Pi_t^2)$ has a degree of precision $t \geq 1$, we assume that the set of points $\boldsymbol{Z}$ has cardinality $N_P(t)+1$ and a subset which is a unisolvent point set for Π_t^2.

Theorem 3 *Let $\mathbf{z}_k = (x_k, y_k) \in \mathring{\mathcal{C}} \cap \boldsymbol{X}$, where $\mathcal{C}$ is an ordinary fault of the function $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$. Let us consider null rules $\mathcal{N}(\cdot, \mathbf{Z}_\delta, \Pi_t^2)$, with $\mathbf{Z}_\delta := \{\bar{\mathbf{z}}_l^\delta,\ l = 1, .., n\}$ and weights $\{w_l^\delta\}_{l\in\mathcal{I}}$, such that:*

(i) $\mathbf{z}_k \in \mathbf{Z}_\delta$;
(ii) $\mathbf{Z}_\delta \subset B_\delta(\mathbf{z}_k) = \{\mathbf{z} \in \mathbb{R}^2 : \|\mathbf{z} - \mathbf{z}_k\| < \delta\}$;
(iii) *For any $\delta_1 \neq \delta_2$, $\mathbf{Z}_{\delta_1}$ and $\mathbf{Z}_{\delta_2}$ differ only for the transformations (shifts and shrinkings) of Proposition 2;*
(iv) *Being $\mathbf{t}_{\mathbf{z}_k}$ the unit vector tangent to $\mathcal{C}$ at $\mathbf{z}_k$, and denoting $\mathcal{I}_\delta^L := \{1 \leq l \leq n : (\bar{\mathbf{z}}_l^\delta - \mathbf{z}_k) \times \mathbf{t}_{\mathbf{z}_k} < 0\}$ and $\mathcal{I}_\delta^R := \{1 \leq l \leq n : (\bar{\mathbf{z}}_l^\delta - \mathbf{z}_k) \times \mathbf{t}_{\mathbf{z}_k} > 0\}$, then $\mathcal{I}_\delta^L, \mathcal{I}_\delta^R \neq \emptyset$;*
(v) $\sum_{j\in\mathcal{I}_\delta^L} w_j^\delta \neq 0$ *and* $\sum_{j\in\mathcal{I}_\delta^R} w_j^\delta \neq 0$.

We have

$$\lim_{\delta\to 0} |\mathcal{N}(f, \mathbf{Z}_\delta, \Pi_t^2)| = C > 0, \ \textit{where } C < |f_{\mathbf{z}_k}^R - f_{\mathbf{z}_k}^L|.$$

Proof First, we note that by assumption (iii) and Proposition 2 the weights of the considered null rules do not depend on δ, and therefore we will denote them just by w_l. Moreover, by definition of $\mathbf{Z}_\delta$, $\bar{\mathbf{z}}_k = \mathbf{z}_k \in \mathbf{Z}_\delta$. Assuming that $f(\mathbf{z}_k) = f_{\mathbf{z}_k}^L$ (the proof is analogous assuming $f(\mathbf{z}_k) = f_{\mathbf{z}_k}^R$), since $\sum_{l\in\mathcal{I}_\delta} w_l = 0$, we have

$$\mathcal{N}(f, \mathbf{Z}_\delta, \Pi_t^2) = \sum_{l=1}^{n} w_l f(\bar{\mathbf{z}}_l) = \sum_{l=1}^{n} w_l f(\bar{\mathbf{z}}_l) - \sum_{l=1}^{n} w_l f_{\mathbf{z}_k}^L = \sum_{l=1}^{n} w_l (f(\bar{\mathbf{z}}_l) - f_{\mathbf{z}_k}^L).$$

Then, dividing the indices into two parts:

$$\sum_{l=1}^{n} w_l (f(\bar{\mathbf{z}}_l) - f_{\mathbf{z}_k}^L) = \sum_{l\in\mathcal{I}_\delta^L\cup\{k\}} w_l (f(\bar{\mathbf{z}}_l) - f_{\mathbf{z}_k}^L) + \sum_{l\in\mathcal{I}_\delta^R} w_l (f(\bar{\mathbf{z}}_l) - f_{\mathbf{z}_k}^L).$$

Note that, by definition of $\mathbf{Z}_\delta$, $\lim_{\delta\to 0} \max_{l\in\mathcal{I}_\delta} \operatorname{dist}(\bar{\mathbf{z}}_l, \mathbf{z}_k) = 0$. Since $\mathbf{z}_k$ belongs to (the interior of) an ordinary fault $\mathcal{C}$ and assumption (iii) holds, by definition of ordinary fault (1) we obtain the following limit behaviour:

$$\lim_{\delta\to 0} \sum_{l\in\mathcal{I}_\delta^L\cup\{k\}} w_l (f(\bar{\mathbf{z}}_l) - f_{\mathbf{z}_k}^L) = 0,$$

$$\lim_{\delta\to 0}\sum_{l\in\mathcal{I}_\delta^R} w_l(f(\bar{\mathbf{z}}_l)-f^L_{\mathbf{z}_k}) = \sum_{l\in\mathcal{I}_\delta^R} w_l(f^R_{\mathbf{z}_k}-f^L_{\mathbf{z}_k}) = (f^R_{\mathbf{z}_k}-f^L_{\mathbf{z}_k})\sum_{l\in\mathcal{I}_\delta^R} w_l.$$

By assumption (ii), we can conclude that the second limit is not null. The upper bound for the constant C can be derived as follows

$$\begin{aligned} C &= \left|(f^R_{\mathbf{z}_k}-f^L_{\mathbf{z}_k})\sum_{l\in\mathcal{I}_\delta^R} w_l\right| \leq \left|f^R_{\mathbf{z}_k}-f^L_{\mathbf{z}_k}\right|\left|\sum_{l\in\mathcal{I}_\delta^R} w_l\right| \leq \left|f^R_{\mathbf{z}_k}-f^L_{\mathbf{z}_k}\right|\sum_{l\in\mathcal{I}_\delta^R}|w_l| \\ &< \left|f^R_{\mathbf{z}_k}-f^L_{\mathbf{z}_k}\right|\sum_{l\in\mathcal{I}_\delta}|w_l| \leq \left|f^R_{\mathbf{z}_k}-f^L_{\mathbf{z}_k}\right|, \end{aligned}$$

where in the last step we have applied hypothesis (iv) and the L^1 normalization property of null rules. □

The previous theorem holds if the point is exactly on the fault line, but the same results hold with slightly different assumptions for sequences of points moving toward the fault curve. The proof of such a result is analogous to the one of Theorem 3.

Corollary 4 *Let $\mathcal{C}$ be an ordinary fault of the function $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$. Let us consider null rules $\mathcal{N}(\cdot, \mathbf{Z}_\delta, \Pi^2_t)$, where $\mathbf{Z}_\delta := \{\bar{\mathbf{z}}^\delta_l,\ l = 1, .., n\}$ and weights $\{w^\delta_l\}_{l\in\mathcal{I}}$, such that:*

(i) $\mathbf{Z}_\delta \subset B_\delta(\mathbf{z}^\delta)$, $\mathbf{z}^\delta \in \mathbb{R}^2$, with $\lim_{\delta\to 0}\mathbf{z}^\delta = \bar{\mathbf{z}} \in \mathring{\mathcal{C}}$;

(ii) For any $\delta_1 \neq \delta_2$, $\mathbf{Z}_{\delta_1}$ and $\mathbf{Z}_{\delta_2}$ differ only for the transformations (shifts and shrinkings) of Proposition 2;

(iii) Being $\mathbf{t}_{\bar{\mathbf{z}}}$ the unit vector tangent to $\mathcal{C}$ at $\bar{\mathbf{z}}$, and denoting $\mathcal{I}^L_\delta := \{1 \leq l \leq n : (\bar{\mathbf{z}}^\delta_l - \mathbf{z}_k) \times \mathbf{t}_{\mathbf{z}_k} < 0\}$ and $\mathcal{I}^R_\delta := \{1 \leq l \leq n : (\bar{\mathbf{z}}^\delta_l - \mathbf{z}_k) \times \mathbf{t}_{\mathbf{z}_k} > 0\}$, then $\mathcal{I}^L_\delta, \mathcal{I}^R_\delta \neq \emptyset$;

(iv) $\sum_{j\in\mathcal{I}^L_\delta} w^\delta_j \neq 0$ and $\sum_{j\in\mathcal{I}^R_\delta} w^\delta_j \neq 0$.

Then:

$$\lim_{\delta\to 0}|\mathcal{N}(\cdot, \mathbf{Z}_\delta, \Pi^2_t)| = C > 0, \ \textit{where } C < |f^R_{\bar{\mathbf{z}}} - f^L_{\bar{\mathbf{z}}}|.$$

The above results cover the case of points where we expect ordinary faults, as we will discuss in the next section. In general, a null rule has an asymptotic behaviour that depends on the regularity of the function. We consider building null rules centered on the target point. When f is known to be regular we then have the following:

Theorem 5 *Let f be r times differentiable at $\bar{\mathbf{z}} = (\bar{x}, \bar{y})$, and let $\mathcal{N}(\cdot, \mathbf{Z}_\delta, \Pi^2_t)$ be null rules with $\bar{\mathbf{z}} \in \mathbf{Z}_\delta$ and such that $\mathbf{Z}_\delta \subset B_\delta(\bar{\mathbf{z}})$. Then*

$$\mathcal{N}(f, \mathbf{Z}_\delta, \Pi^2_t) = o\left(\delta^{min\{r,t\}}\right)$$

Proof Let $s = \min\{r, t\}$, and let

$$f(\mathbf{z}) = f(\bar{\mathbf{z}}) + \frac{\partial f(\bar{\mathbf{z}})}{\partial x}(x-\bar{x}) + \frac{\partial f(\bar{\mathbf{z}})}{\partial y}(y-\bar{y}) + ... + R(\mathbf{z})$$

be the Taylor expansion of degree s of $f(\mathbf{z})$ centered at $\bar{\mathbf{z}}$, where $R(\mathbf{z})$ is the remainder term. Since the rule has degree of precision t, we have $\mathcal{N}(f, \mathbf{Z}_\delta) = \mathcal{N}(R, \mathbf{Z}_\delta)$. Then:

$$\mathcal{N}(R, \mathbf{Z}_\delta, \Pi^2_t) = \sum_{l=1}^{n} w_l R(\mathbf{z}_l) \leq \|\mathbf{w}\| \cdot \|\mathbf{R}\| \tag{5}$$

where $\mathbf{R} = [R(\mathbf{z}_l)]_{l=1,...,n}$. Finally, because points $\mathbf{z}_l \in B_\delta(\bar{\mathbf{z}})$, we have $R(\mathbf{z}_l) \leq k \cdot \delta^s$ and this concludes our proof. □

4 Discontinuity Detection by Null Rules

Our goal is to be able to detect which points are close to ordinary and gradient faults starting from the given set of scattered points $\mathbf{X}$ and corresponding function values F. In other words, with the given data we cannot consider a sequence of null rules like in Theorem 3, Corollary 4, and Theorem 5. However, since these results describe the asymptotic behaviour of the null rules as the distances between the points decrease, they also provide accurate information when the density of the given points is sufficiently high. In this section, we use these results in order to define a reliable and efficient estimator to be used in the case of general scattered data configurations.

We introduce two indicators to detect ordinary and gradient fault, respectively. Starting from Theorem 5, we know that a null rule will be rapidly vanishing when applied to points that belong to regions where the function is regular. On the other side, as a consequence of Theorems 3 and 4, it will not vanish if the point is near an ordinary fault. Since the computation of high order null rules can be expensive, for any point $\mathbf{z}_k \in \boldsymbol{X}$ we chose to construct a null rule with a degree of precision $t = 2$ on $\boldsymbol{Z}_k$, where $\boldsymbol{Z}_k$ is the set of the $n = 7$ nearest neighbours of $\mathbf{z}_k$ ($\mathbf{z}_k$ included) such that it cointains a subset of unisolvent points for Π_2^2 (or, equivalently, the corresponding collocation matrix Φ in (3) has rank 6). The 7 points satisfying this requirement coincide very often with the 7 points closest to $\mathbf{z}_k$, but in some cases, we need to select a slightly different set (see lines 6–20 of Algorithm 1). This set is obtained by first adding to $\boldsymbol{Z}_k$ just enough neighbours so that $\text{rank}(\Phi) = 6$ (lines 6–8 of Algorithm 1), for then removing some of these points, starting from the farthest neighbours, until a matrix $\Phi \in \mathbb{R}^{6\times 7}$ with $\text{rank}(\Phi) = 6$ is suitably identified (lines 6–8 of Algorithm 1). Therefore, with this choice of $\boldsymbol{Z}_k$ the null rule with a degree of precision 2 is unique. We simplify the notation by using $\mathcal{N}(f, \boldsymbol{Z}_k)$ instead of $\mathcal{N}(f, \boldsymbol{Z}_k, \Pi_t^2)$. We define the ordinary fault indicator as

$$I_O(\mathbf{z}_k, \boldsymbol{X}) := \frac{\mathcal{N}(f, \boldsymbol{Z}_k)}{\eta}, \tag{6}$$

with $\eta = \max_{\mathbf{z}\in\boldsymbol{Z}_k} \|\mathbf{z} - \mathbf{z}_k\|$. In view of Theorem 5, we expect that this indicator assumes small values if f is C^0 at $\mathbf{z}_k$ and large values if f is C^{-1} at $\mathbf{z}_k$. Therefore, it is reasonable to classify $\mathbf{z}_k$ as close to an ordinary fault if and only if $I_O(\mathbf{z}_k, \boldsymbol{X}) > \alpha_O$ for a certain threshold α_O. Particular attention should be devoted to fixing this threshold. Note that if f has a jump at $\mathbf{z}_k$, according to Theorems 3–4, asymptotically the value of $\mathcal{N}(f, \boldsymbol{Z}_k)$ depends on $|f_{\bar{\mathbf{z}}}^R - f_{\bar{\mathbf{z}}}^L|$, that is, on the height of the jump. Because of this, using a large threshold α_O could make it difficult to catch discontinuities with small jumps. On the other hand, setting a small value for α_O could cause the classification of non-fault points as fault ones, maybe because of a rapid variation of the functions. Therefore, we use a non-constant α_O, depending on an estimate of the jump:

$$\alpha_O = \sigma\left(\left(\max_{\mathbf{z}\in\boldsymbol{Z}_k} f(\mathbf{z}) - \min_{\mathbf{z}\in\boldsymbol{Z}_k} f(\mathbf{z})\right)/\eta\right), \tag{7}$$

where σ is the sigmoidal function

$$\sigma(s) := \frac{1}{1 + e^{\sigma_1 s + \sigma_2}}, \qquad \sigma_1, \sigma_2 \in \mathbb{R}.$$

The function σ is a bounded and differentiable function with non-negative derivative and one inflection point chosen to take a value close to 1 or 0.1 when its argument is 10 or 0, respectively.

In view of Theorem 5, the third power of the null rule of degree 2 is expected to be $o((\eta^2)^3)$ in the regular cases and $o(\eta^0)$ or less if f is C^0 or C^{-1} at z_k. As an indicator of the gradient fault, we then evaluate the ratio between this value and the fifth power of η:

$$I_G(\mathbf{z}_k, \boldsymbol{X}) := \frac{[\mathcal{N}(f, \mathbf{Z}_k)]^3}{\eta^5}. \tag{8}$$

What we expect is that the indicator $I_G(\mathbf{z}_k, \boldsymbol{X})$, if f is no more than C^0 at $\mathbf{z}_k$, tends to take large values, while tends to have small values if f is C^2 at $\mathbf{z}_k$. Also in this case we need to set a threshold, which we denote by α_G and will be a free parameter of the detection algorithm. Finally, in order to not classify ordinary faults as gradient ones, we classify $\mathbf{z}_k$ as close to a gradient fault if and only if $I_G(z_k, \boldsymbol{X}) > \alpha_G$ and $I_O(z_k, \boldsymbol{X}) \leq \alpha_O$.

The fault detection method resulting from the combination of the construction of the null rule, the computation of the two indicators, and their application to the data is formally described in Algorithm 1. The parameter $active_G$ in Algorithm 1 is a boolean variable determining whether or not the algorithm must look for gradient fault points.

Algorithm 1: FAULT_DET

```
Input   : $\mathbf{X}$, $F$, $\alpha_G$, $active_G$
Set $X_O = \emptyset$ and $X_G = \emptyset$;
for $k = 1, \dots, N$ do
    Sort the points in $\boldsymbol{X}$ with respect to their distance from $\mathbf{z}_k$;
    Initialize $\mathbf{Z}_k$ with the 7 nearest neighbors of $\mathbf{z}_k$ in $\boldsymbol{X}$;
    Set $\Phi$ as the collocation matrix of the linear system (4) with $t = 2$;
    while rank$(\Phi) < 6$ do
        Add to $\mathbf{Z}_k$ the successive neighbor of $\mathbf{z}_k$ and update $\Phi$ accordingly;
    end
    Set $\tilde{\mathbf{Z}}_k = \mathbf{Z}_k$, $\tilde{\Phi} = \Phi$, and $\hat{Z}_k = |\mathbf{Z}_k|$;
    for $j = 1, \dots, \hat{Z}_k - 7$ do
        Set $i = |\mathbf{Z}_k|$;
        Update $\tilde{\mathbf{Z}}_k$ by removing the $i$-th point from $\mathbf{Z}_k$;
        Update $\tilde{\Phi}$ by removing the $i$-th column from $\Phi$;
        while rank$(\tilde{\Phi}) < 6$ do
            Decrease $i$ by 1;
            Update $\tilde{\mathbf{Z}}_k$ by removing the $i$-th point from $\mathbf{Z}_k$;
            Update $\tilde{\Phi}$ by removing the $i$-th column from $\Phi$;
        end
        Set $\mathbf{Z}_k = \tilde{\mathbf{Z}}_k$ and $\Phi = \tilde{\Phi}$;
    end
    Find the null rule $\mathcal{N}(\cdot, \mathbf{Z}_k)$ as a non-null normalized vector of $ker(\Phi)$;
    Compute the indicators $I_O(\mathbf{z}_k, \boldsymbol{X})$ and $I_G(\mathbf{z}_k, \boldsymbol{X})$ from equations (6)-(8);
    Compute the threshold $\alpha_O$ from equation (7) ;
    if $I_O(\mathbf{z}_k, \boldsymbol{X}) > \alpha_O$ then
        Add $\mathbf{z}_k$ to the set of ordinary fault points $X_O$;
    else
        if $I_G(\mathbf{z}_k, \boldsymbol{X}) > \alpha_G$ & $active_G$ =TRUE then
            Add $\mathbf{z}_k$ to the set of gradient fault points $X_G$;
        end
    end
end
Output: Sets of ordinary ($X_O$) and gradient ($X_G$) fault points
```

Remark 6 Being based on null rules, the indicators (6–8) can be directly applied to a set $\mathbf{X}$ of scattered data. Moreover, for each point $\mathbf{z}_k$ the two indicators $I_O(\mathbf{z}_k, X)$ and $I_G(\mathbf{z}_k, X)$ are computed using the same null rule $\mathcal{N}(f, \mathbf{Z}_k)$, which means that in our approach the cost of finding both ordinary and gradient fault points is very similar to the cost of finding only ordinary fault points.

5 Adaptive Surface Reconstruction

Detecting the points close to faults provides essential information for the accurate reconstruction of a target surface and its discontinuities. In particular, in the context of adaptive approximation methods, it is natural to use the location of the points close to discontinuities to drive the refinement. Here, as a proof of concept, we focus on adaptive global least squares approximations based on hierarchical spline spaces, but the basic idea can be easily applied to any adaptive approximation method based on spaces with local refinement capabilities.

Let us consider the space of hierarchical splines, see, e.g., [17], which have already been employed to define adaptive approximation methods in different contexts, see, e.g., [9, 10, 18]. Let $V^0 \subset \ldots \subset V^{L-1}$ be a sequence of L spaces of tensor-product splines of degree $\mathbf{p} := (p_1, p_2)$ defined on a certain domain Ω and constructed on the meshes $\mathcal{G}^\ell$, $\ell = 0, ..., L-1$. Each mesh $\mathcal{G}^\ell$ is usually obtained from the one of the previous level by performing uniform dyadic refinement. Let $\Omega^0 \supseteq \ldots \supseteq \Omega^L$ be a sequence of subdomains (each of them being a union of cells of the tensor-product grid of the corresponding level), with $\Omega^0 := \Omega$ and $\Omega^L := \emptyset$. The hierarchical mesh $\mathcal{G}_\mathcal{H}$ is the set of active cells at different levels.

We indicate by $\mathcal{B}_\mathbf{p}^\ell := \left\{B_J^\ell\right\}_{J\in\Gamma_\mathbf{p}^\ell}$ the B-spline basis of V^ℓ. For any $s \in V^\ell$, $\ell = 0, \ldots, L-2$, let

$$s = \sum_{J\in\Gamma_\mathbf{p}^{\ell+1}} c_J^{\ell+1} B_J^{\ell+1}$$

be its representation in terms of B-splines of the finer space $V^{\ell+1}$. The truncation of s with respect to level $\ell+1$ and its (cumulative) truncation with respect to all levels are defined as

$$\text{trunc}^{\ell+1}(s) := \sum_{J\in\Gamma_\mathbf{p}^{\ell+1}:\,\text{supp}(B_J^{\ell+1})\nsubseteq\Omega^{\ell+1}} c_J^{\ell+1} B_J^{\ell+1},$$

and

$$\text{Trunc}^{\ell+1}(s) := \text{trunc}^{L-1}(\text{trunc}^{L-2}(\ldots(\text{trunc}^{\ell+1}(s))\ldots)),$$

respectively. For the finest level, we set $\text{Trunc}^L(s) := s$ for $s \in V^{L-1}$. The truncated hierarchical B-spline (THB-spline) basis $\mathcal{T}_\mathbf{p}(\mathcal{G}_\mathcal{H})$ of the hierarchical space $S_\mathcal{H}$ is defined as

$$\mathcal{T}_\mathbf{p}(\mathcal{G}_\mathcal{H}) := \left\{T_J^\ell := \text{Trunc}^{\ell+1}(B_J^\ell) : J \in A_\mathbf{p}^\ell, \ell = 0, \ldots, L-1\right\}$$

where

$$A_\mathbf{p}^\ell := \left\{J \in \Gamma_\mathbf{p}^\ell : \text{supp}(B_J^\ell) \subseteq \Omega^\ell \wedge \text{supp}(B_J^\ell) \nsubseteq \Omega^{\ell+1}\right\}$$

is the set of *active* multi-indices, and $\text{supp}(B_J^\ell)$ denotes the intersection of the support of B_J^ℓ with Ω^0. The B-spline B_J^ℓ is called the *mother* B-spline of the truncated basis function T_J^ℓ.

In other words, in different areas of the domain the space is spanned by splines defined on finer or coarser meshes. The choice of the subdomains determines where the space is coarser or finer.

Given the set of scattered data points $\mathbf{X} = \{\mathbf{z}_k,\ k = 1, ..., N\}$ and the corresponding set of function values $F = \{f(\mathbf{z}_k)\ :\ k = 1, ..., N\}$, we can define a classic regularized least squares approximation

$$\mathbf{s}(\mathbf{z}) := \sum_{T \in \mathcal{T}_{\mathbf{p}}(\mathcal{G}_{\mathcal{H}})} c_T T(\mathbf{z}) \tag{9}$$

minimizing the objective function

$$\sum_{k=1}^{N} (\mathbf{s}(\mathbf{z}_k) - f(\mathbf{z}_k))^2 + \mu E(\mathbf{s}), \tag{10}$$

where $\mu > 0$ is a smoothing coefficient and $E(\mathbf{s})$ the thin-plate energy

$$E(\mathbf{s}_J) := \int_{\Omega} \left\| \frac{\partial^2 \mathbf{s}}{\partial x^2} \right\|_2^2 + 2 \left\| \frac{\partial^2 \mathbf{s}}{\partial x \partial y} \right\|_2^2 + \left\| \frac{\partial^2 \mathbf{s}}{\partial y^2} \right\|_2^2 \mathrm{d}x\mathrm{d}y.$$

The key point in order to effectively use the approximant (9) is costructing a suitable hierarchical space $\mathcal{T}_{\mathbf{p}}(\mathcal{G}_{\mathcal{H}})$ over the adaptively refined mesh $\mathcal{G}_{\mathcal{H}}$. This is where we can suitably exploit the result of the fault detection algorithm.

The basic idea relies upon refining the mesh in the areas containing the fault points. However, it is a good idea to first apply a post-processing step to the detected fault points. The fault point cloud has a certain thickness, while the fault itself is, in principle, simply a curve. Consequently, the location of the fault can be better described if we apply narrowing techniques to the cloud of fault points. In particular, we here employ the method successfully applied in [8, Section 4], based on the computation of local least squares approximations, see also [20] and [25] for the details.

By starting from a tensor-product mesh (and the corresponding spline space), we can then set up an iterative scheme where at each refinement step one hierarchical level is added by refining the union of the supports of the mothers of the basis functions containing at least one fault point (and update the spline space accordingly). Moreover, in order to have a limited number of basis functions active on each cell, we apply an algorithm that guarantees the *admissibility* of the mesh, see, e.g., [11]. The refinement stops once a maximum number of levels L is reached, and the hierarchical spline approximation $\mathbf{s}$ in (9) is then computed in the resulting adaptively refined spline space.

It should be noted that since gradient faults have higher regularity than ordinary faults, it is reasonable to assume that gradient fault areas require less refinement than ordinary fault ones. Therefore, we allow to set a different maximum number of refinement levels for areas with ordinary and gradient fault points, denoted respectively by L_O and L_G.

The adaptive surface reconstruction method resulting from the combination of the refinement based on the fault detection algorithm here proposed and the hierarchical spline approximation given by (9) is detailed in Algorithm 2 and denoted `FDSR` (Fault Detection Surface Refinement).

Algorithm 2: FDSR

```
   Input  : $\mathbf{X}, F, \alpha_G, active_G, \mathcal{G}^0, V^0, L_O, L_G, \mu$
1  $[\mathbf{X}_O, \mathbf{X}_G]$ = FAULT_DET($\mathbf{X}, F, n, \alpha_G, active_G$);
2  Update the sets $\mathbf{X}_O$ and $\mathbf{X}_G$ by applying to them the narrowing algorithm ;
3  Initialize the hierarchical mesh and the corresponding spline space: $\mathcal{G}_{\mathcal{H}} = \mathcal{G}^0$ and $S_{\mathcal{H}} = V^0$;
4  for $i = 1, \ldots, L_O - 1$ do
5      if $i \leq L_G - 1$ then
6          Set $\Omega_i = \bigcup_{T:\mathrm{supp}(T)\cap(\mathbf{X}_O\cup\mathbf{X}_G)\neq\emptyset} \bigcup_{Q\in\mathrm{supp}(T)} Q$;
7      else
8          Set $\Omega_i = \bigcup_{T:\mathrm{supp}(T)\cap\mathbf{X}_G\neq\emptyset} \bigcup_{Q\in\mathrm{supp}(T)} Q$;
9      end
10     Update $\Omega_1$, ..., $\Omega_i$ by applying the admissibility algorithm;
11     Update the hierarchical space $S_{\mathcal{H}}$ accordingly;
12 end
13 Compute the approximation $s$ (9) in the space $S_{\mathcal{H}}$ with smoothing coefficient $\mu$;
   Output: Regularized least squares approximation $s$
```

Remark 7 Note that the adaptive surface refinement based on fault detection proposed in Algorithm 2 enables the construction of the hierarchical mesh by simply taking into account the location of the detected fault points. The THB-spline approximation in (9) can then be computed only once, after the hierarchical spline space has been finalized, see line 13 of Algorithm 2. In this way, our adaptive method is computationally less expensive than standard adaptive methods based on error-driven refinements. In that case, not only the approximation in (9) but also the corresponding errors at the points $\mathbf{z} \in \mathbf{X}$ need to be computed at each refinement step, as the refinement is driven by the location of points where the error exceeds a given tolerance ϵ—see lines 8–9 inside the iterative loop of Algorithm 3 presented in the next section.

6 Numerical Tests

In this section, we first test the fault detection algorithm on several datasets, including analytical benchmarks and a geological point cloud. Then, we compare the adaptive algorithm for the (adaptive) reconstruction of surfaces with discontinuities with the classical approach where adaptivity is driven by the error evaluation.

If not indicated otherwise, in all tests the parameters in the sigmoidal functions are $\sigma_1 = -0.40$, $\sigma_2 = 1.96$, and the threshold α_G for the detection of gradient faults is set to $1/10$. In all tests, except for the one of Sect. 6.2, when applying Algorithm 1, the collocation matrices Φ used to compute the null rules obtained with the initially selected 7 neighboring points always have rank 6. For the example of Sect. 6.2, we will provide an analysis of the initial rank of the collocation matrices.

6.1 Fault Detection on Benchmark Data

We first test the detection algorithm on some benchmark examples. The first two test functions (already considered in [7, 8]) are

$$f_1(x,y) = \begin{cases} x - 0.4 - 0.1\sin(2\pi y) - 0.2 & \text{if } x > 0.7 + 0.1\sin(2\pi y) \\ x - 0.4 - 0.1\sin(2\pi y) & \text{if } x \le 0.7 + 0.1\sin(2\pi y) \\ & \quad \text{and } x^2 + y^2 > 0.16, \\ \sqrt{4 - x^2 - y^2} - \sqrt{4 - 0.16} + 0.1 & \text{if } x^2 + y^2 \le 0.16, \end{cases}$$

$$f_2(x,y) = \left| x^2 + y^2 - \frac{1}{2} \right| - x + y^2 + 0.3(-x + 0.4 + 0.1\sin(2\pi y))_+,$$

where

$$(t)_+ := \begin{cases} t & \text{if} \quad t \ge 0, \\ 0 & \text{if} \quad t < 0, \end{cases}$$

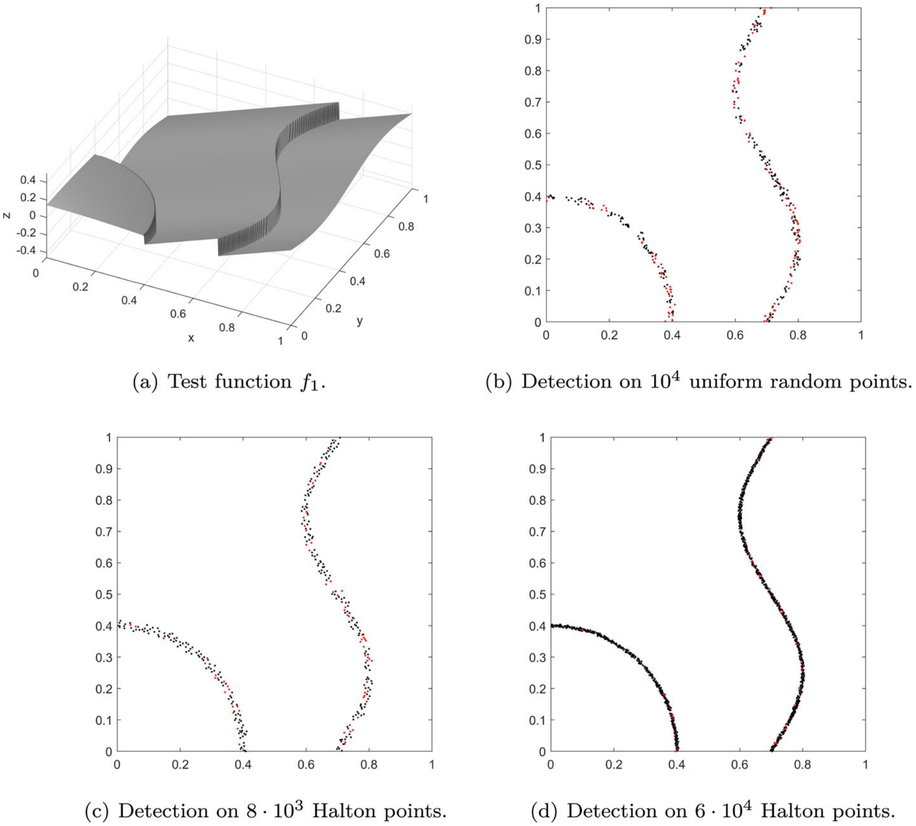
and the function describing the well-known Shepp-Logan phantom surface. In order to assess the behavior of the indicators on unorganized point clouds of different nature and density, we consider the following three scattered data sets $\mathbf{X}$ in $[0,1]^2$: $N = 10^4$ random points generated according to the uniform distribution, $N = 8 \cdot 10^3$ Halton points in $[0,1]^2$, and $N = 6 \cdot 10^4$ Halton points in $[0,1]^2$. Halton points are quasi-random deterministic points that are often used to test numerical methods dealing with scattered data, and they can be easily generated in Matlab via the $haltonset$ routine. In all the cases, the detection algorithm is applied with the parameter $active_G = TRUE$, that is, we look for both ordinary and gradient faults. Figures 1, 2 and 3 show the surfaces corresponding to the test functions and the results obtained by applying our fault detection algorithm. In the figures, the points are black and red according to their classification as ordinary and gradient fault points, respectively.

For all test functions and sets of points the detection is able to detect the correct shape of the discontinuities. As with other methods based on the analysis of the asymptotic behaviour of indicators, the number (and, as a consequence, the density) of the points has some influence on the results, see, e.g., Example 6.1 in [7]. As the density of the points increases, the strip of detected points becomes narrower and the classification of the fault type is more accurate, see Figs. 1d, 2d and 3d. From this point of view, the Shepp-Logan Phantom example is the most challenging of the presented ones, since both modest and high jumps are present in the surface. However, also in this case, as shown in Fig. 3b–d, the method correctly provides a more and more accurate detection and classification of the discontinuities as the density of points increases.

6.2 Fault Detection on Geological Data

We also test our detection method in a case of interest previously addressed in [8]. We consider the geological dataset composed of 149511 scattered points with their associated heights, corresponding to the surface shown in Fig. 4 (left). In this case, since the data are affected by a lot of small bumps, we do not expect to be able to detect gradient faults, so we apply the fault detection algorithm with parameter $active_G = FALSE$. This is not restrictive for this kind of application, since we are interested in detecting fractures in the terrain, which can be mathematically identified as ordinary faults. In this case the parameters of the sigmoidal function are $\sigma_1 = -0.65$ and $\sigma_1 = 1.30$. The plot on the right of Fig. 4 clearly shows that the algorithm is able to properly catch all the main ordinary fault curves.

In this case, when applying Algorithm 1, for some points the collocation matrix Φ used to obtain the null rules initially has rank less than 6, and then some of the 7 neighbors were replaced by near points (according to lines 9–20 of Algorithm 1). More precisely, for 98.52%



(a) Test function f_1.

(b) Detection on 10^4 uniform random points.

(c) Detection on $8 \cdot 10^3$ Halton points.

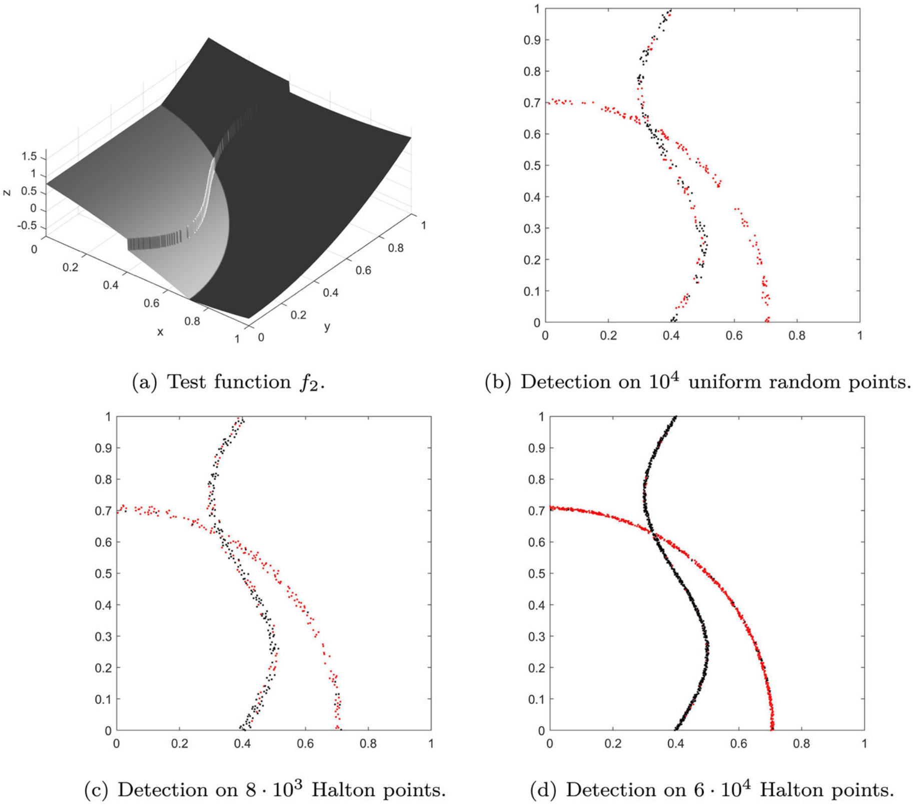
(d) Detection on $6 \cdot 10^4$ Halton points.

Fig. 1 Application of the detection algorithm on data sampled from the test function f_1 (top left). The results obtained by considering 10^4 uniformly distributed random points (top right), as well as $8 \cdot 10^3$ (bottom left) and $6 \cdot 10^4$ (bottom right) Halton points are shown

of points the rank of Φ was already 6 considering the initial 7 neighbors, for 1.47% of points the rank was 5, and for the remaining points it was 4 or less.

6.3 Adaptive Reconstruction of Surfaces with Discontinuities

In this subsection, we test the application of fault detection in adaptive surface reconstruction, as previously described in Sect. 5. In order to show the advantages of this approach, we compare the `FDSR` method introduced in Algorithm 2 with the `ESR` (Error-based Surface Reconstruction) method detailed in Algorithm 3. The difference with respect to `FDSR` relies upon the nature of the refinement scheme, which is error-based, as commonly done in several adaptive approximation methods based on hierarchical splines, see, e.g., [9, 10, 18]. As anticipated in Remark 7, in `ESR` both the approximation (9) in the current hierarchical space $S_{\mathcal{H}}$ and the errors at the points in $\mathbf{X}$ are evaluated at each iteration of the adaptive loop. The hierarchical mesh is then updated by refining the supports of the basis functions containing at least one point where a given tolerance ϵ is exceeded.



(a) Test function f_2.

(b) Detection on 10^4 uniform random points.

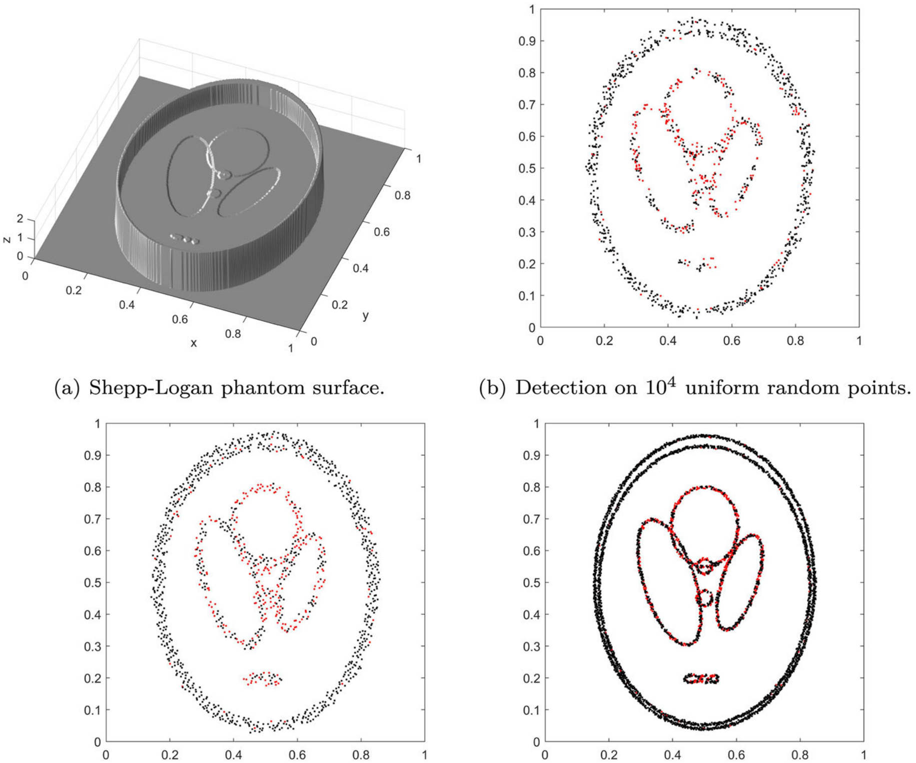
(c) Detection on $8 \cdot 10^3$ Halton points.

(d) Detection on $6 \cdot 10^4$ Halton points.

Fig. 2 Application of the detection algorithm on data sampled from the test function f_2 (top left). The results obtained by considering 10^4 uniformly distributed random points (top right), as well as $8 \cdot 10^3$ (bottom left) and $6 \cdot 10^4$ (bottom right) Halton points are shown

Algorithm 3: ESR

```
Input  : $\mathbf{X}$, $F$, $\mathcal{G}^0$, $V^0$, $L$, $\mu$, $\epsilon$
Set $\mathcal{G}_{\mathcal{H}} = \mathcal{G}^0$ and $S_{\mathcal{H}} = V^0$;
Compute the approximation $\mathbf{s}$ (9) in the space $S_{\mathcal{H}}$ with the smoothing coefficient $\mu$;
Evaluate $e_k := |f(\mathbf{z}_k) - s(\mathbf{z}_k)|$ for $k = 1, ..., N$ and set $D = \{\mathbf{z}_k \in \mathcal{X} : e_k > \epsilon\}$;
while $S_{\mathcal{H}}$ has less than L levels and $D \neq \emptyset$ do
    Set $\Omega_i = \bigcup_{T : \mathrm{supp}(T) \cap D \neq \emptyset} \bigcup_{Q \in \mathrm{supp}(T)} Q$;
    Update $\Omega_1, ..., \Omega_i$ by applying the admissibility algorithm;
    Update the hierarchical space $S_{\mathcal{H}}$ accordingly;
    Compute the approximation $s$ (9) in the space $S_{\mathcal{H}}$ with smoothing coefficient $\mu$;
    Evaluate $e_k := |f(\mathbf{z}_k) - s(\mathbf{z}_k)|$, for $k = 1, ..., N$, and $D = \{\mathbf{z}_k \in \mathcal{X} : e_k > \epsilon\}$;
end
Output: Regularized least squares approximation $s$
```



(a) Shepp-Logan phantom surface.

(b) Detection on 10^4 uniform random points.

(c) Detection on $8 \cdot 10^3$ Halton points.

(d) Detection on $6 \cdot 10^4$ Halton points.

Fig. 3 Application of the detection algorithm on data sampled from the Shepp-Logan phantom surface (top left). The results obtained by considering 10^4 uniformly distributed random points (top right), as well as $8 \cdot 10^3$ (bottom left) and $6 \cdot 10^4$ (bottom right) Halton points are shown

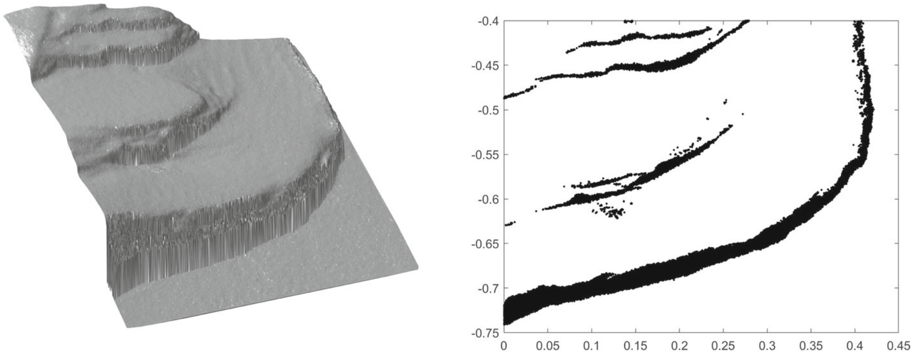
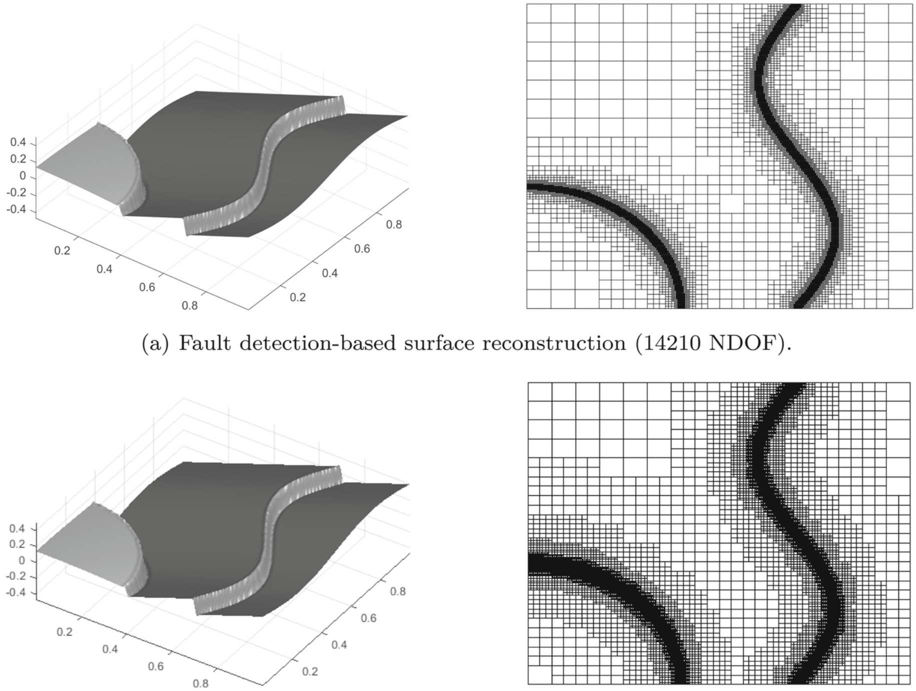


Fig. 4 A geological dataset visualized as a piecewise linear surface obtained from a triangulation of the point cloud (left) and the results of the detection algorithm (right)

Table 1 Comparison between approximations obtained by using fault detection-based and error-based refinements for test surface f_1, geological dat,a and test surface f_2

		FDSR		ESR	
		RMSE	NDOF	RMSE	NDOF
test surf. f_1	$(L_O = L_G = 6)$	$1.004e - 02$	14210	$1.004e - 02$	21306
geo. data-set	$(L_O = L_G = 6)$	$1.193e - 02$	29468	$1.187e - 02$	35201
test surf. f_2	$(L_O = L_G = 6)$ $(L_O = 6, L_G = 4)$	$9.400e - 03$ $9.379e - 03$	16588 12138	$9.383e - 03$	16145

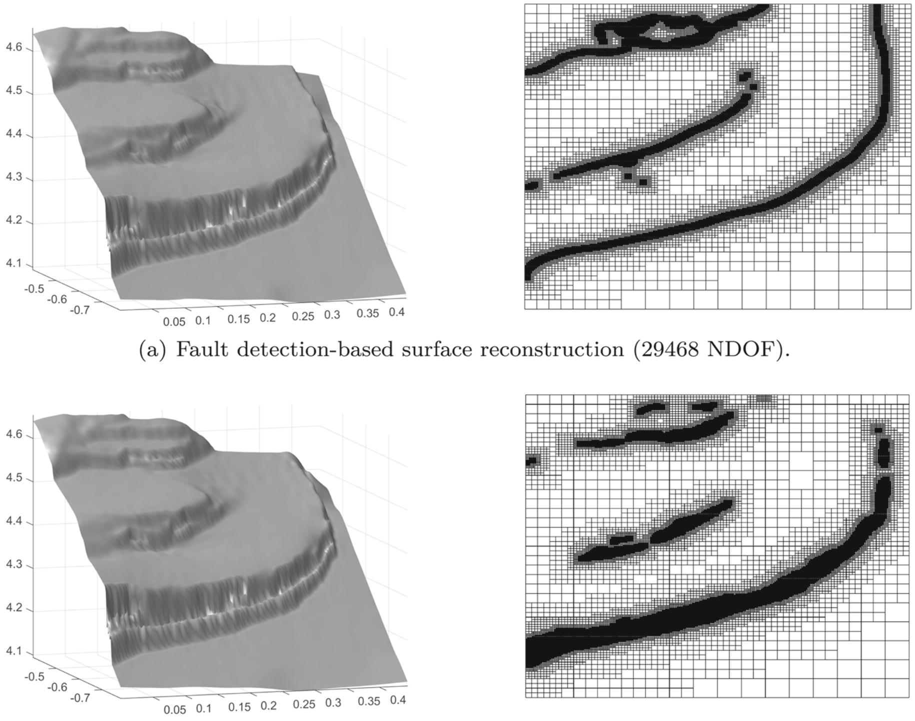


(a) Fault detection-based surface reconstruction (14210 NDOF).

(b) Error-based surface reconstruction (21306 NDOF).

Fig. 5 The surfaces and hierarchical meshes obtained applying to test surface f_1 the regularized least-squares approximation with fault detection-based (**a**) and error-based (**b**) refinement

For all the examples we start from a 16 × 16 tensor-product mesh $\mathcal{G}^0$ and the space of biquadratic splines V^0 defined on it. First, we try to reconstruct two surfaces with only ordinary faults, by considering the dataset coming from sampling the test function f_1 and the geological dataset. We compare the results obtained by applying the algorithms FDSR($\mathbf{X}$, F, 0.1, $FALSE$, $\mathcal{G}^0$, V^0, 6, 6, 10^{-5}) and ESR($\mathbf{X}$, F, $\mathcal{G}^0$, V^0, L, 10^{-5}, 10^{-2}). We test the methods also on a third example where both ordinary and gradient faults are present, namely, the dataset coming from sampling the test function f_2. In this case, we then apply FDSR with $active_G = TRUE$. Moreover, since we have both types of fault, for this third example we add to the comparison also the result obtained by choosing a different maximum number of levels for the areas with ordinary and gradient faults: $L_O = 6, L_G = 4$.



(a) Fault detection-based surface reconstruction (29468 NDOF).

(b) Error-based surface reconstruction (35201 NDOF).

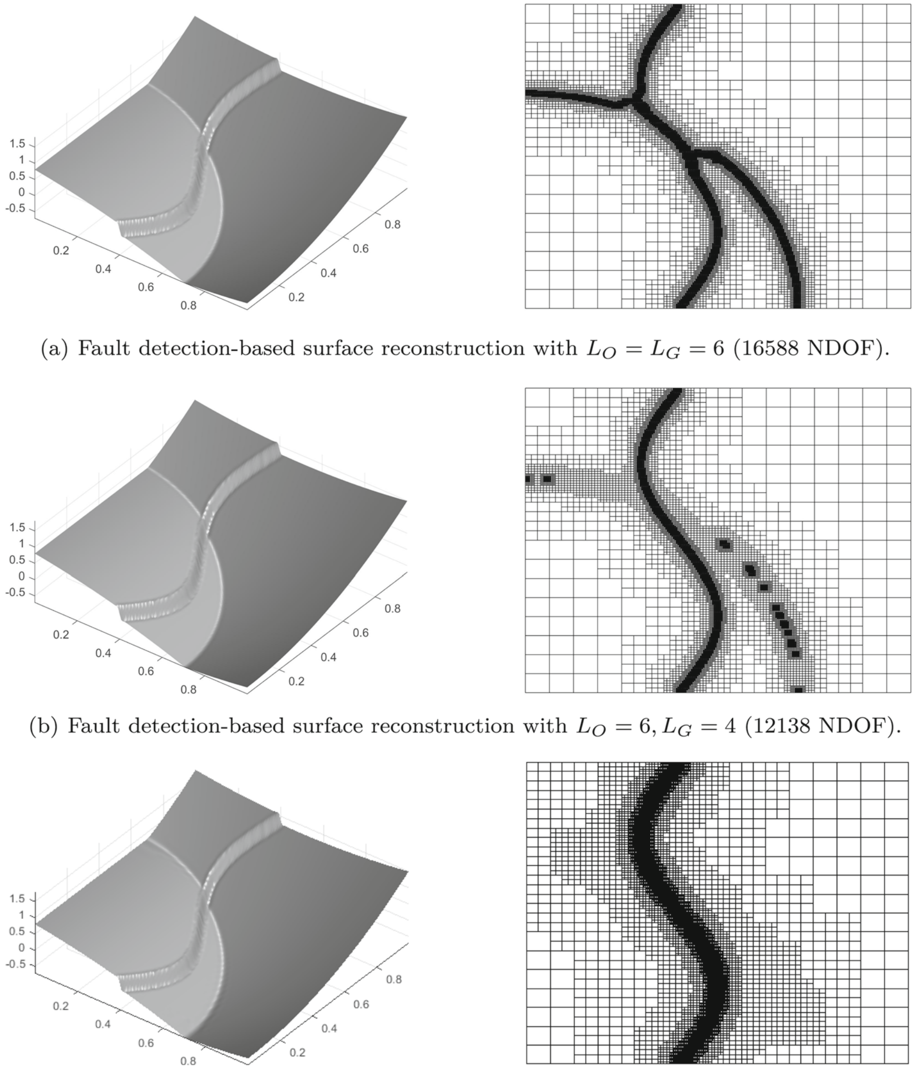
Fig. 6 The surfaces and hierarchical meshes obtained applying to the geological data the regularized least-squares approximation with fault detection-based (**a**) and error-based (**b**) refinement

Table 1 reports, for each test, the Root Mean Square Error (RMSE) obtained and the Number of Degrees of Freedom (NDOF) obtained. In Figs. 5, 6 and 7 we give a visual comparison between the application of the two methods in the case of the surface reconstruction for the three tests.

From Table 1, it is clear that in all examples the fault detection-based and the error-based refinements achieve analogous error values. However, the fault detection-based refinement FDSR reaches this accuracy with significantly fewer degrees of freedom than ESR, the refinement driven by error computation. Note that in the third example on function f_2, this advantage is obtained by exploiting the ability to refine less in areas close to the gradient fault. In fact, being the gradient fault a discontinuity of higher order, less refinement is sufficiently close to this fault with respect to the refined regions that would be needed in the neighborhood of an ordinary fault. This is achieved by setting L_G to 4, which leads to a strongly reduced number of degrees of freedom (11858) with respect to ESR (16145).

In order to complete the analysis of our approach, we also compare the refinement driven by the fault detection based on null rules with the fault detection based on minimal numerical differentiation presented in [8], which were also designed to work on scattered data. We consider the test function (see Example 6.1 in [8])

$$f_4(x, y) = \begin{cases} |x - 0.4 - 0.1\sin(2\pi y)| & \text{if } x \leq 0.7 + 0.2\sin(2\pi y), \\ |x - 0.4 - 0.1\sin(2\pi y) - 0.2| & \text{otherwise.} \end{cases}$$



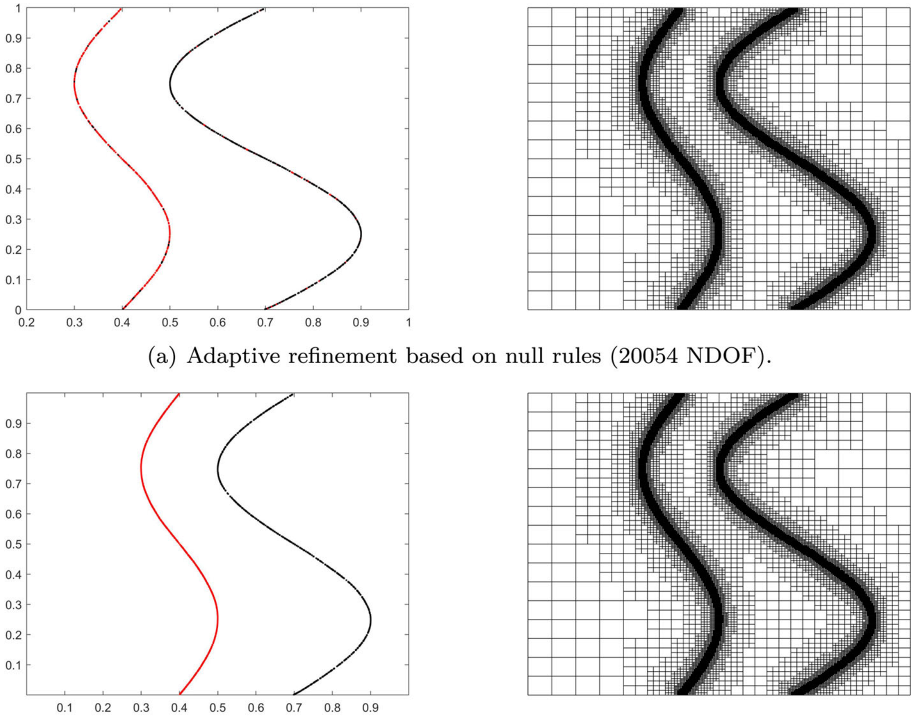
(a) Fault detection-based surface reconstruction with $L_O = L_G = 6$ (16588 NDOF).

(b) Fault detection-based surface reconstruction with $L_O = 6, L_G = 4$ (12138 NDOF).

(c) Error-based surface reconstruction (16145 NDOF).

Fig. 7 The surfaces and hierarchical meshes obtained applying to test surface f_2 the regularized least-squares approximation with fault detection-based refinement and $L_O = L_G = 6$ (**a**), fault detection-based refinement and $L_O = 6, L_G = 4$ (**b**), and error-based refinement (**c**)

For this example we start again from the space of biquadratic splines V^0 defined on a 16×16 tensor-product mesh $\mathcal{G}^0$, and the data are the set $\mathbf{X}$ of $N = 6 \cdot 10^4$ Halton points in $[0, 1]^2$ and the set $\mathbf{F}$ of corresponding values of f_4. A visual comparison of the results is presented in Fig. 8. In the left column of the figure, the sets of detected points obtained with the two approaches after the narrowing post-processing are shown. On the right column of the figure, instead, the corresponding hierarchical meshes obtained during the adaptive surface reconstruction scheme which suitably exploits these points to drive the refinement



(a) Adaptive refinement based on null rules (20054 NDOF).

(b) Adaptive refinement based on numerical differentiation formulas (20183 NDOF).

Fig. 8 Adaptive refinement driven by fault detection based on null rules (top) and numerical differentiation formulas (bottom). The detected ordinary and gradient fault points after the narrowing process are shown on the left column, while the corresponding hierarchical meshes are shown on the right

are presented. The two approaches produce very similar results: the hierarchical meshes are almost the same and lead to very close numbers of degrees of freedom (20054 vs. 20183). Note that the indicators defined in [8] require, for each point, the computation of two rules: one to approximate the Laplacian and one for the gradient of the function. This corresponds to solving 3 local linear systems for each point. Our approach based on null rules requires solving only one local linear system for each point, which results in a clear computational advantage.

7 Conclusions

We analyzed the behaviour of null rules close to faults of bivariate functions as well as in areas where the functions are regular. Based on the results reported in Theorems 4–5 that link the asymptotic behaviour of the application of the null rules to the local regularity of the function, we construct two indicators. The values of the indicators are used to determine if a point is close to an ordinary or gradient fault (or it is regular). Since null rules do not require any structured data configuration, these indicators can be directly applied to scattered data. Moreover, they are very cheap: only one null rule (for each point) must be computed in order to evaluate both indicators, and the null rule is obtained by solving a local, small linear system of dimension 6 × 7. The numerical experiments show that the indicators are able to properly detect and classify the points close to faults. This information can then be suitably exploited to drive the refinement in adaptive reconstruction methods based on THB-splines. The results show that this kind of refinement, compared to the traditional error-driven adaptive loop, gives an equally accurate approximation with a reduced number of degrees of freedom.

Acknowledgements Cesare Bracco, Francesco Calabrò and Carlotta Giannelli are members of INdAM-GNCS and are partially supported by INdAM-GNCS Project (CUP E53C22001930001). Cesare Bracco and Carlotta Giannelli are also partially supported by the National Recovery and Resilience Plan, Mission 4 Component 2 - Investment 1.4 - NATIONAL CENTER FOR HPC, BIG DATA AND QUANTUM COMPUTING - funded by the European Union - NextGenerationEU - (CUP B83C22002830001).

Funding This work was partially supported by the Istituto Nazionale di Alta Matematica - Gruppo Nazionale per il Calcolo Scientifico (INdAM-GNCS), Italy. Open Access funding enabled and organized by Italy Transformative Agreement.

Data Availability The paper contains all the information to simply generate the datasets supporting the conclusions of this article, except for the geological dataset, courtesy of Giacomo Corti (Istituto di Geoscienze e Georisorse CNR, Florence - Italy).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ahmadi, H., Pekkan, E.: Fault-based geological lineaments extraction using remote sensing and gis-a review. Geosciences. **11**, 1–31 (2021)
2. Azarafza, M., Ghazifard, A., Akgün, H., Asghari-Kaljahi, E.: Development of a 2d and 3d computational algorithm for discontinuity structural geometry identification by artificial intelligence based on image processing techniques. Bull. Eng. Geol. Environ. **78**, 3371–3383 (2019)
3. Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C.T.: A survey of surface reconstruction from point clouds. Comput. Graph. Forum **36**(1), 301–329 (2017)
4. Berntsen, J., Espelid, T.O.: Error estimation in automatic quadrature routines. ACM Trans. Mathe. Softw. **17**(2), 233–253 (1991)
5. Bozzini, M., Linarduzzi, L., Rossini, M.: Non-regular surface approximation. In: Floater, M., et al. (eds.) Mathematical Methods for Curves and Surfaces: 8th International Conference. MMCS 2012, Oslo, Norway, pp. 68–87. Lecture Notes in Computer Science, Springer (2014)
6. Bozzini, M., Rossini, M.: The detection and recovery of discontinuity curves from scattered data. J. Comput. Appl. Math. **240**, 148–162 (2013)
7. Bracco, C., Davydov, O., Giannelli, C., Sestini, A.: An application of numerical differentiation formulas to discontinuity curve detection from irregularly sampled data. Rend. Semin. Mat. Univ. Politec. Torino **76**(2), 69–76 (2018)
8. Bracco, C., Davydov, O., Giannelli, C., Sestini, A.: Fault and gradient fault detection and reconstruction form scattered data. CAGD **75**(2), 101786 (2019)
9. Bracco, C., Giannelli, C., Großmann, D., Sestini, A.: Adaptive fitting with thb-splines: Error analysis and industrial applications. CAGD **62**, 239–252 (2018)
10. Bracco, C., Giannelli, C., Sestini, A.: Adaptive scattered data fitting by extension of local approximations to hierarchical splines. CAGD **52–53**, 90–105 (2017)
11. Bracco, C., Giannelli, C., Vázquez, R.: Refinement algorithms for adaptive isogeometric methods with hierarchical splines. Axioms **7**, 1–25 (2018)
12. Calabrò, F., Bravo, D., Carissimo, C., Di Fazio, E., Di Pasquale, A., Eldray, A.A.M.O., Fabrizi, C., Gerges, J.G.S., Palazzo, S., Wassef, J.F.F.T.: Null rules for the detection of lower regularity of functions. J. Comput. Appl. Math. **361**, 547–553 (2019)
13. Canny, J.: A a computational approach to edge detection. IEEE T. Pattern Anal. **6**, 679–698 (1986)

14. Dokken, T., Lyche, T., Pettersen, K.F.: Polynomial splines over locally refined box-partitions. CAGD **30**, 331–356 (2013)
15. Espelid, T.O.: Doubly adaptive quadrature routines based on Newton-Cotes rules. BIT **43**, 319–337 (2003)
16. Gałkowski, T., Krzyzak, A., Filutowicz, Z.: A new approach to detection of changes in multidimensional patterns. J. Artif. Intell. Soft Comput. Res. **10**, 125–136 (2020)
17. Giannelli, C., Jüttler, B., Speleers, H.: Thb-splines: the truncated basis for hierarchical splines. CAGD **29**, 485–498 (2012)
18. Kiss, G., Giannelli, C., Zore, U., Jüttler, B., Großmann, D., Barner, J.: Adaptive cad model (re-)construction with thb-splines. Graph. Models **76**, 273–288 (2014)
19. Laurie, D.P.: Null rules and orthogonal expansions. In R.V. Zahar, editor, Proceedings of the International Conference on Special Functions, Approximation, Numerical Quadrature and Orthogonal Polynomials, pp. 359–370, Basel, (1994). Birkhäuser
20. Lee, I.K.: Curve reconstruction from unorganized points. CAGD **17**, 161–177 (2000)
21. Li, X., Zheng, J., Sederberg, T.W., Hughes, T.J.R., Scott, M.A.: On linear independence of t-spline blending functions. CAGD **29**, 63–76 (2012)
22. Lyness, J.N.: Symmetric integration rules for hypercubes. iii. construction of integration rules using null rules. Math. Comput. **19**(92), 625–637 (1965)
23. Rezaei, K., Agahi, H.: An automatic thresholding approach to gravitation based edge detection in grey scale images. J. Inf. Syst. Telecommun. **36**, 285–296 (2021)
24. Romani, L., Rossini, M., Schenone, D.: Edge detection methods based on rbf interpolation. J. Comput. Appl. Math. **349**, 532–547 (2019)
25. Sober, B. and Levin, D.: Manifold approximation by moving least-squares projection (MMLS). Technical report available on (arXiv:1606.07104v4), (2018)
26. Xu, J., Wen, X., Zhang, H., Luo, D., Li, J., Xu, L., Yu, M.: Automatic extraction of lineaments based on wavelet edge detection and aided tracking by hillshade. Adv. Space Res. **65**, 506–517 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.