Adaptive detection and approximation of unknown surface discontinuities from scattered data

Adaptive detection and approximation of unknown surface discontinuities from scattered data

G. Allasia, R. Besenghi, and R. Cavoretto

Department of Mathematics,
University of Turin,
via Carlo Alberto 10, 10123 Torino, Italy
E-mail: giampietro.allasia, renata.besenghi, roberto.cavoretto@unito.it

Abstract

We consider a method for the detection and approximation of fault lines of a surface, which is known only on a finite number of scattered data. In particular, we present an adaptive approach to detect surface discontinuities, which allows us to give an (accurate) approximation of the detected faults. First, to locate all the nodes close to fault lines, we consider a procedure based on a local interpolation scheme involving a cardinal radial basis formula. Second, we find further sets of points generally closer to the faults than the fault points. Finally, after applying a nearest-neighbor searching procedure and a powerful refinement technique, we outline some different approximation methods. Numerical results highlight the efficiency of our approach.

Keywords: scattered data, surface discontinuities, adaptive detection, approximation methods, radial basis functions.

Mathematics Subject Classification 2000: 65D05, 65D10, 65D17.

1 Introduction

In this paper we propose an adaptive approach for the detection and approximation of surface discontinuities, which are known only on a finite number of scattered data. In geology a surface discontinuity is often named with the word “fault”, which means discontinuity in a layer caused by severe movements of the earth’s crust. In geological applications, such as oil finding, fault localization is particularly important, because it provides useful information on the occurrence of oil reservoirs [10]. It is well known that the interpretation of faults in seismic data is today a time-consuming manual task, and reducing time from exploration to production of an oil field has great economical benefits [11].

Besides geology, the problem of detecting surface discontinuities, often referred as fault lines or edges (i.e. jumps in a surface), is encountered frequently in a wide variety of scientific fields, such as image processing, medicine, geophysics, oceanography, tomography, cartography, etc.. Either the fault detecting and approximating problem, or the discontinuous surface approximating problem have been dealt with extensively in the literature and several authors have proposed different techniques and methodologies (see, for instance, [5, 16, 13, 8, 9, 15, 12] and references therein).

In the following, for simplicity, we will consider surface discontinuities of the type of geological vertical faults and will refer to these simply as faults.

In [2] we presented a method for the localization of unknown fault lines of a surface, moving from a large set of data points or nodes irregularly distributed in a plane region and the corresponding function values. In the present paper we improve the detection scheme proposed in [2], introducing an *adaptive detection process* (see Section 5 for more details). Moreover, we discuss different methods to approximate fault lines, considering polygonal lines, least squares, and best l_∞ approximations [3].

Note that in some real applications, e.g. in geology, the searching of information, i.e. discontinuity points, can be very expensive, since it is required to drill or explode mines in subsoil. Moreover, in general, one does not know the location of the fault lines on the surface or even whether or not the surface is faulted [4]. So, the adaptive procedure can reveal itself very useful, because this approach produces a considerable reduction of cost.

The paper is organized as follows. In Section 2 we explain the detection algorithm for the fault lines, relating to *cardinal radial basis interpolants* (CRBIs), whose properties are widely discussed in [1]. In Section 3 we briefly describe the different techniques of approximation. Section 4 contains some useful remarks. In Section 5 several numerical results show the effectiveness of our approach. Finally, in Section 6, we give an idea of possible developments.

2 Detection Scheme

In this section we present the basic ideas which allow us to pick out the data points on or close to the fault lines, named *fault points*. To characterize these points, first, we consider a procedure based on local data interpolation by CRBIs, where the difference between any known function value and the related value of the interpolant is computed and compared with a threshold parameter. Second, we introduce and define a new set of points, named *barycentres*, generally closer to the faults than the fault points. Finally, manipulating the barycentres, we find further sets of points subjected or not to any refinement. They supply more information on the faults.

Let $S_n = \{P_k,\ k = 1, \dots, n\}$ be a set of distinct and scattered data points in a domain $D \subset \mathbb{R}^2$, and $\{f(P_k),\ k = 1, \ldots, n\}$ a set of corresponding values of an unknown function $f : D \to \mathbb{R}$, which is discontinuous across a set $\Gamma = \{\Gamma_j,\ j = 1, \ldots, m\}$ of fault lines

$$\Gamma_j = \{\gamma_j(t) : t \in [0, 1]\} \subset D,$$

where γ_j are unknown parametric continuous curves. On $D \setminus \Gamma$ the function f is supposed to be smooth. The domain D is bounded, closed, simply connected, and contains the convex hull of S_n.

The detection algorithm is divided into five steps:

(I) A cell-based search method is applied to find the data point set $\mathcal{N}_{P_k}$ neighboring to each node P_k of the data set S_n [6]. It is a classical *nearest-neighbor searching procedure*, in which we make a subdivision of the domain D in cells and identify the points closest to a data point within each cell. Then, to determine the fault points, a CRBI is built on each set $\mathcal{N}_{P_k}$, ($k = 1, \ldots, n$), excluding P_k. As a significant example of CRBI we recall *Shepard's formula* (see [1])

$$F(x) = \sum_{i=1}^{n} f_i \ \frac{d(x, x_i)^{-p}}{\sum_{j=1}^{n} d(x, x_j)^{-p}}, \qquad F(x_i) = f_i, \quad i = 1, \ldots, n,$$

where $d(x, x_i)$ is the Euclidean distance in $\mathbb{R}^2$ and $p > 0$.

(II) We evaluate the absolute value of the difference $\sigma(P_k)$ between the function value $f(P_k)$ and the value of the interpolant at P_k, viz. $\sigma(P_k) = |f(P_k) - F(P_k)|$. Supposing the interpolant gives a good approximation to f in D, then $\sigma(P_k)$ gives a measure of the smoothness of the function around P_k. If we choose a suitable threshold value $\sigma_0 > 0$, we can compare the values $\sigma(P_k)$ and σ_0: $\sigma(P_k)$ is less than or equal to σ_0 when f is smooth in a neighborhood of P_k, otherwise it is greater than σ_0 when a steep variation of f at P_k is found, and accordingly P_k will be marked as a fault point. When the detection procedure of the nodes is concluded, we have characterized the *set of fault points* $\mathcal{F}(f; S_n) = \{P_k \in S_n : \sigma(P_k) > \sigma_0\}$ consisting of all the nodes either belonging to the faults or, at least, close to them.

(III) Given a fault point $P_k \in \mathcal{F}(f; S_n)$ and the corresponding nearest-neighbor set $\mathcal{N}_{P_k}$, we define $D_{P_k} = \{P \in D : d(P, P_k) \leq R_{P_k}\}$, where $R_{P_k} = \max\{d(P_i, P_k) : P_i \in \mathcal{N}_{P_k}\}$. Then we order the $N_{P_k} + 1$ points in $\tilde{\mathcal{N}}_{P_k} = \mathcal{N}_{P_k} \cup \{P_k\}$, being $N_{P_k} = \text{card}(\mathcal{N}_{P_k})$, so that $f(P_{k1}) \leq f(P_{k2}) \leq \cdots \leq f(P_{kN_{P_k}+1})$. The expected jump δ_k of f in the subdomain D_{P_k} is evaluated by

$$\delta_k = \max_{1 \leq l \leq N_{P_k}} \Delta f(P_{kl}),$$

where Δ is the forward difference operator. We set l_k the lowest value of the index $l \in \{1, \ldots, N_{P_k}\}$ for which δ_k is obtained.

The part $\Pi_k = D_{P_k} \cap \Gamma$ of a fault line separates the set $\Delta_k^L = \{Q \in \bar{\mathcal{N}}_{P_k} : f(Q) \leq f(P_{kl_k})\}$ of all nodes of $\bar{\mathcal{N}}_{P_k}$ with lower function values from the set $\Delta_k^H = \{Q \in \bar{\mathcal{N}}_{P_k} : f(Q) > f(P_{kl_k})\}$ of all nodes of $\bar{\mathcal{N}}_{P_k}$ with higher function values. If Δ_k^L or Δ_k^H is the empty set, then we enlarge $\mathcal{N}_{P_k}$ and repeat the process, so that $\bar{\mathcal{N}}_{P_k}$ contains points lying in the two parts of the subdomain separated by the fault line. Having determined Δ_k^L and Δ_k^H in this way, we calculate the barycentres A_k^L and A_k^H of Δ_k^L and Δ_k^H, respectively. Then we find $A_k = (A_k^L + A_k^H)/2$ and put it in $\mathcal{A}(f; S_n)$, the *set of the barycentres*.

(IV) We subdivide the domain D by a regular grid and point out the grid cells containing points of $\mathcal{A}(f; S_n)$, i.e. barycentres. Since each grid cell contains at least a barycentre, we calculate the barycentre of the points of $\mathcal{A}(f; S_n)$ in each cell, namely, the barycentre of the barycentres for each cell. In this way, we have a further set $\mathcal{B}(f; S_n)$ containing points that are generally closer to the faults than the barycentres. The points that belong to $\mathcal{B}(f; S_n)$ are ordered by applying the following nearest-neighbor searching procedure:

1. detection of any point of $\mathcal{B}(f; S_n)$ nearest to a side of the domain D and assumption of this one as the first probe point;

2. search of the nearest-neighbor point to the actual probe point and assumption of this one as the new probe point, excluding the points already considered;

3. stopping the process, when all points of $\mathcal{B}(f; S_n)$ are ordered.

It follows that the sorting process identifies the ordered set $\mathcal{C}(f; S_n) = \{C_i,\ i = 1, \ldots, m\}$, where m is the number of points of $\mathcal{B}(f; S_n)$.

(V) We consider the following *refinement technique*. Starting from the set $\mathcal{C}(f; S_n)$, we take C_{i-1}, C_i and C_{i+1}, $i = 2, \cdots, m-1$. They are the vertices of a triangle. Therefore, holding C_1 and C_m fixed, we find $m-2$ triangles and calculate the barycentre $C_j^{(1)}$, $2 \leq j \leq m-1$, for each triangle. This last stage can be repeated, so obtaining the barycentres $C_j^{(k)}$, for $j = 2, \ldots, m-1$, $k = 1, 2, \ldots$

In general, step (V) is required mainly when the used approximation scheme has not in itself the smoothing property, such as the polygonal line method described below.

3 Approximation Scheme

We sketch different approaches to approximate fault lines, namely the polygonal line method, the least squares method and the best l_∞ approximation method (see [3] for an exaustive presentation of the methods).

Polygonal line method. The polygonal line method we consider represents an improvement of the procedures developed by Gutzmer and Iske [13] and by Allasia, Besenghi and De Rossi [2].

The ordered points of $\mathcal{C}(f;S_n)$ are connected by straight line segments, obtaining a polygonal line which approximates the fault line.

Least squares method. The least squares method can be used to approximate a data set, $\{(x_i,y_i),\ \ i=1,\ldots,m\}$ by an algebraic polynomial

$$P_s(x)=c_0+c_1x+\cdots+c_{s-1}x^{s-1}+c_sx^s \tag{1}$$

of degree $s<m-1$. In our problem we choose the constants $c_0,c_1,\ldots,c_s$ to solve the *least squares problem*

$$\min_{c_0,\ldots,c_s}\sum_{i=1}^{m}\left[y_i-P_s(x_i)\right]^2,$$

where $(x_i,y_i)\in\mathcal{C}(f;S_n)$. Replacing the coefficients $c_0,c_1,\ldots,c_s$ in (1) with the values obtained by the least squares method, we get the polynomial approximating the fault line.

Best l_∞ approximation method. The best l_∞ approximation method can be considered as a tool for finding polynomial approximations to fault lines starting from the set $\mathcal{C}(f;S_n)$ or from some its refinement.

Given the abscissae x_k and the corresponding ordinates y_k, for $k=1,2,\ldots,m$, the polynomial in (1) is sought to solve the *minimax problem*

$$\min_{c_0,\ldots,c_s}\max_{k}\left|P_s(x_k)-y_k\right|. \tag{2}$$

Denoting by $M=\max_k|P_s(x_k)-y_k|>0$ the largest absolute value, we have

$$|P_s(x_k)-y_k|\le M,\quad k=1,2,\ldots,m, \tag{3}$$

and the inequalities (3) can be rewritten in the form

$$\left|\sum_{j=0}^{s}\left(\frac{c_j}{M}\right)x_k^j-\left(\frac{1}{M}\right)y_k\right|\le 1,\quad k=1,2,\ldots,m.$$

The linear programming problem (2) becomes

$$\text{maximize}\qquad Z=t_{s+2},$$

subject to the constraints

$$\begin{array}{rcr} t_1+x_kt_2+x_k^2t_3+\ldots+x_k^st_{s+1}-y_kt_{s+2} & \le & 1, \\ t_1+x_kt_2+x_k^2t_3+\ldots+x_k^st_{s+1}-y_kt_{s+2} & \ge & -1, \end{array}$$

for $k = 1, 2, \ldots, m$, and

$$t_{s+2} \geq 0,$$

with the unknowns

$$t_1 = \frac{c_0}{M}, \;\; t_2 = \frac{c_1}{M}, \;\; t_3 = \frac{c_2}{M}, \;\; \ldots, \;\; t_{s+1} = \frac{c_s}{M}, \;\; t_{s+2} = \frac{1}{M}.$$

Now, applying the *simplex method*, we obtain the coefficients c_0, $c_1, \ldots, c_s$ of (1) and identify an approximating polynomial to the fault line.

4 Some Remarks

Threshold value choice. In the detection algorithm a crucial point is the optimal choice of the threshold value σ_0. It is, in general, a difficult task, because the finite number of data is the only available information. Computing the largest deviation

$$S = \max\{|f_i - f_j| : i > j, \text{ for all } i, j = 1, 2, \ldots, n\},$$

we achieve a useful information on the variation of f and we can set $\sigma_0 = \epsilon S$, with $0 < \epsilon < 1$. Repeated tests pointed out that the procedure works well when $1/2^5 \leq \epsilon \leq 1/2^3$, and it ends successfully if the function f is smooth on $D \setminus \Gamma$. On the contrary, if the function f shows steep variations, the procedure must be repeatedly applied with increasing values of ϵ.

Smoothness. The performance of the approximation procedures depends, in particular, on the number of points and the form of the faults. Least squares and best l_∞ approximations give smooth curves, whereas polygonal lines have a simpler formulation. If a polygonal line appears too irregular and poorly accurate, we can yield another smoother (and, possibly, more accurate) polygonal line, using the refinement technique and connecting the barycentres $C_j^{(k)}$ by straight line segments. Iterating some times the refinement technique, we construct smooth, planar curves rather than polygonal curves. This is a meaningful advantage if compared with the Gutzmer-Iske method [13].

Complex surface discontinuities. The proposed approximation methods work well, when there is only one fault in D. Otherwise, if we deal with complex situations, such as several faults, intersections or bifurcations of faults, these methods cannot be directly applied, but a suitable subdivision of $\mathcal{C}(f; S_n)$ is required. Dealing with two or more surface discontinuities we need to split up the set $\mathcal{C}(f; S_n)$ in a number of subsets greater or equal to the number of fault lines before approximating the discontinuities. This procedure is suggested also when a fault line is not of open type. Acting in this way, very good results are obtained. Finally, when a fault line is parallel or nearly parallel to y-axis, it is convenient to make first a rotation of the coordinate axes and then to apply the least squares method or the best l_∞ approximation method.

Connection errors. When we have to deal with complex surface discontinuities, applying least squares and best l_∞ approximations require a further subdivision of data. This situation can create small errors at points in which two piecewise curves join, but the introduction of the refinement technique reduces considerably the occurrence of these errors of connection. Nevertheless, the decision of how subdividing the domain can hide several difficulties (see [9]). Obviously, in particularly difficult cases, the refinement technique can be used and iterated in both the least squares method and the best l_∞ approximation method.

5 Numerical Results

In this section we propose a few of several numerical and graphical results, obtained by computational procedures developed in *C/C++*, *Matlab* and *Maple* environments.

In various tests we consider n randomly scattered data points $P_i = (x_i, y_i)$ in the square $[0,1] \times [0,1] \in \mathbb{R}^2$ and the corresponding function values f_i, for $i = 1, \ldots, n$. The detection and approximation schemes are successfully tested against several functions with different kinds of faults, varying the dimension n of the sample generated by a uniform distribution and the threshold value σ_0. The results have been obtained by using Shepard's formula ($p = 2$), but other choices of CRBIs could work as well.

The cell-based search method and the CRBIs allow to partition the domain, to process the data in different stages, and to insert or remove nodes. These features are particularly important in surveying phenomena, such as geodetic or geophysical ones, whose data are distributed in regions with different characteristics.

We sketch the results yielded by our method considering five significant functions among those tested. Choosing the optimal value of ϵ, numerous tests pointed out that few thousand data points are sufficient to produce an accurate approximation.

Now, we briefly describe the adaptive approach to detect accurately the surface discontinuities. It consists of following steps:

`Step 1.` Starting from few hundreds of data points (here, we take $n = 250$) in the unit domain D, we apply the detection algorithm, locating roughly the position of possible fault lines. The process determines, first, the set of fault points $\mathcal{F}(f, S_n)$, then, the set of barycentres $\mathcal{A}(f, S_n)$ and, finally, the set of barycentres of barycentres $\mathcal{B}(f, S_n)$ and its ordered form $\mathcal{C}(f, S_n)$.

`Step 2.` We increase the number of nodes, generating a certain number of "local" scattered data on rectangular neighborhoods centred at C_i, $i =$

$1, \cdots, m$, the points of the set $\mathcal{C}(f, S_n)$ (in general, 15–20 nodes for each neighborhood are sufficient).

`Step 3.` From the enlarged set of nodes we obtain more fault points, discarding eventually false fault points previously considered, and accordingly we obtain more barycentres and barycentres of barycentres.

`Step 4.` Stop, if the faults are well individuated, else go back to Step 2.

In practice, a suitable choice of the (local) neighborhood size in step 2 is essential. It is advisable to change the rectangular neighborhood size, first considering greater dimension neighborhoods (here, side length and width are taken between 0.12 and 0.16) and then, in subsequent adaptive detection phases, reducing their sizes at about one half.

Note that the process allows us to evaluate only the new generated data points, exploiting the previous evaluations due to Shepard's formula. Obviously, if we have some information about the position of fault lines, we can locate from the beginning a reduced searching region.

It is remarkable that, also taking a starting sample with few hundreds of scattered data, the method of detection holds its efficiency. Nevertheless, reducing considerably the value of n, a loss of approximation accuracy is unavoidable. It depends essentially on the reduced information, that is, the number of data points. Hence, it is not convenient to take a too small sample dimension, namely n less than one hundred.

In the following, we present the graphics of the considered test functions, the corresponding fault line approximations and a picture of the adaptive detection process. An error analysis is showed as well. It gives an idea of the goodness of our approach.

Test function 1

The test function (see [7, 2])

$$f_1(x, y) = \begin{cases} y - x + 1, & \text{if } 0 \leq x < 0.5, \\ 0, & \text{if } 0.5 \leq x < 0.6, \\ 0.3, & \text{if } x \geq 0.6, \end{cases}$$

is a surface with discontinuities in $x = 0.5$ and $x = 0.6$, i.e.

$$\Gamma = \Gamma_1 \cup \Gamma_2 = \{(0.5, t) : 0 \leq t \leq 1\} \cup \{(0.6, t) : 0 \leq t \leq 1\}.$$

Figure 1 (left) shows that, outside the discontinuities, the function $f_1(x, y)$ is constant for $x > 0.5$, whereas changes enough quickly for $x \leq 0.5$, producing a variation of the jump size in Γ_1. Moreover, Figure 1 (right) provides the graphical representation of the least squares curves from the 75 points of the set $\mathcal{C}(f; S_n)$. In Figure 2 we show step by step the adaptive detection procedure, representing (bottom, from left to right) the subsequent enlarged sets of data points and (top, from left to right) the corresponding sets $\mathcal{C}(f;S_n)$. Precisely, in Figure 2 **(A)**, we report the picture of the points belonging to $\mathcal{C}(f;S_n)$ and obtained by applying the detection procedure from 250 scattered data (not represented). Figure 2 **(B)** and **(C)** refer to the sets $\mathcal{C}(f;S_n)$ of the points, obtained by increasing the number of nodes on or close to possible fault lines, as showed in Figure 2 **(X)** and **(Y)** respectively. Then, Figure 2 **(Z)** contains the enlarged set of data points from which we get the final set $\mathcal{C}(f;S_n)$ and the relative approximation lines (see Figure 1 (right)). These are obtained from the last set $\mathcal{C}(f;S_n)$, subdividing it into two subsets.

In the detection phase, to obtain an accurate approximation of the faults, we employ 3479 data points (this number is purely indicative and may vary considerably). In some cases, nevertheless, if a rough knowledge of faults (e.g. position, behaviour, etc.) is sufficient, the number of the considered points can be considerably reduced (say, at about a quarter). In doing so, we may reduce the number of steps in the adaptive detection process (see Figure 2 **(B)** or **(C)**), but obviously do not attain the accutacy obtained in this paper.

All these remarks hold also for the following test functions.

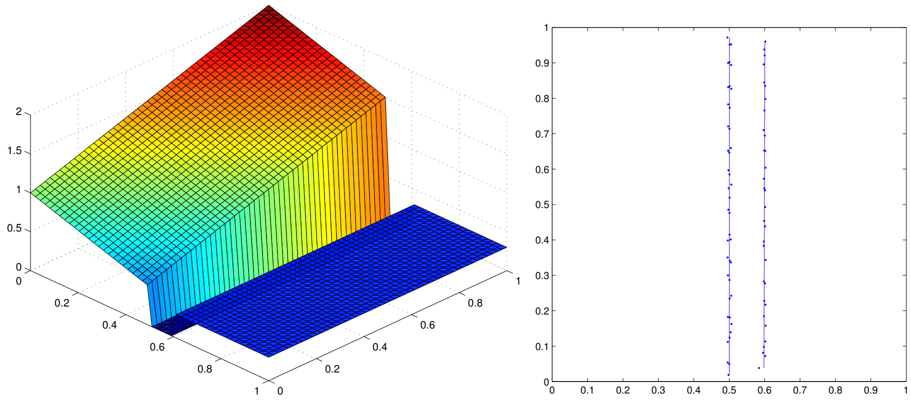

Figure 1: Test function f_1 (left) and least squares approximation curves (right).

<u>Test function 2</u>

The test function [14]

$$f_2(x,y) = \begin{cases} 0, & \text{if } \; x > 0.4, y > 0.4, y < x + 0.2, \\ (x-1)^2 + (y-1)^2, & \text{otherwise}, \end{cases}$$

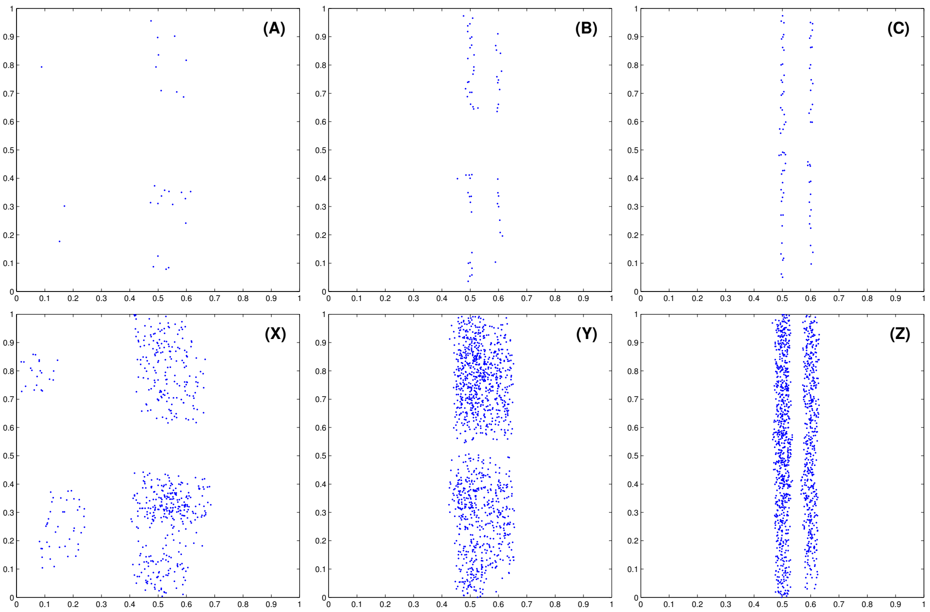

Figure 2: Output of the adaptive detection procedure for f_1.

is a surface with discontinuities across the set

$$\begin{aligned}\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 &= \{(t, 0.4) : 0.4 \le t \le 1\} \cup \{(0.4, t) : 0.4 \le t \le 0.6\} \\ &\cup \{(t, t + 0.2) : 0.4 \le t \le 0.8\},\end{aligned}$$

as showed in Figure 3 (left). Moreover, we can observe that the jump size of fault lines varies in Γ_1, Γ_2 and Γ_3.

Figure 3 (right) provides the graphical representation of the best l_∞ approximation curves from the 45 points belonging to the set $\mathcal{C}(f; S_n)$. Figure 4 presents, instead, a picture of the adaptive detection procedure. The fault line is obtained from the set $\mathcal{C}(f; S_n)$, subdividing the original set into three subsets.

The adaptive detection process ends successfully using 2099 scattered data.

Test function 3

The third discontinuous surface, already studied in [13, 2, 14],

$$f_3(x, y) = \begin{cases} 1 + 2\left\lfloor 3.5\sqrt{x^2 + y^2}\right\rfloor, & \text{if } (x - 0.5)^2 + (y - 0.5)^2 < 0.16, \\ 0, & \text{otherwise,} \end{cases}$$

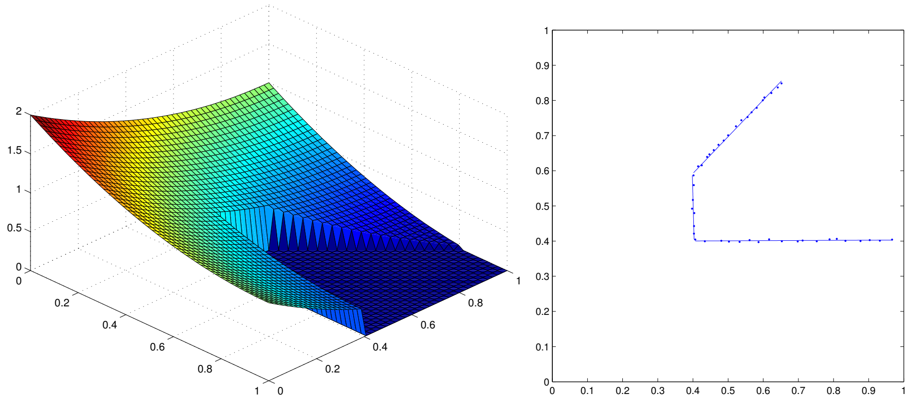

Figure 3: Test function f_2 (left) and best l_∞ approximation curves (right).

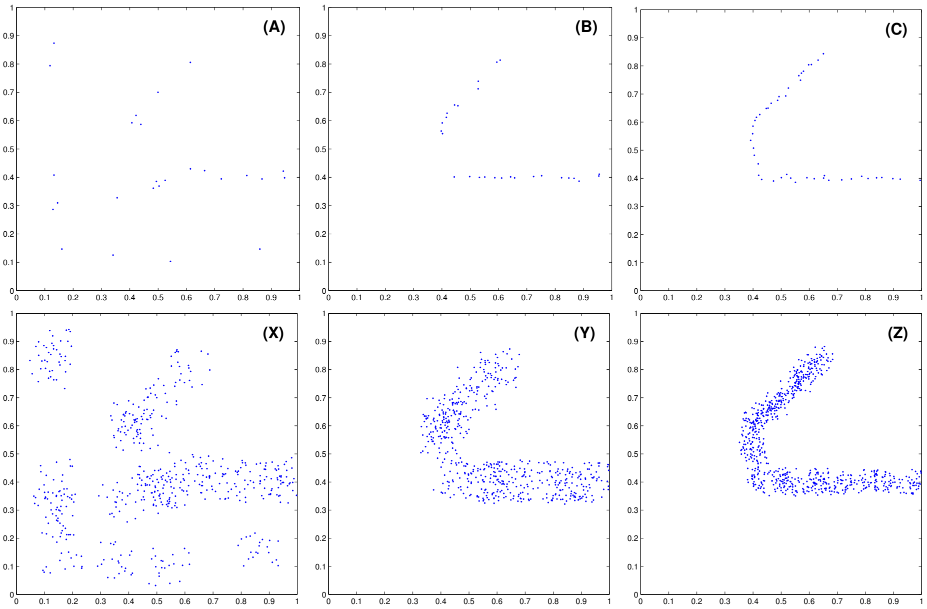

Figure 4: Output of the adaptive detection procedure for f_2.

has the discontinuity set given by

$$
\begin{aligned}
\Gamma &= \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 = \{(x,y) \in \mathbb{R}^2 : (x-0.5)^2 + (y-0.5)^2 = 0.16\} \\
&\cup \ \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 = 16/49, \ (x-0.5)^2 + (y-0.5)^2 \leq 0.16\} \\
&\cup \ \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 = 36/49, \ (x-0.5)^2 + (y-0.5)^2 \leq 0.16\}.
\end{aligned}
$$

The surface is always constant in $D \setminus \Gamma$.

In Figure 5 test function $f_3(x,y)$ (left) and graphic obtained by polygonal line method from the ordered 108 points of $\mathcal{C}(f;S_n)$ (right) are shown. Figure 6 presents the results of the adaptive detection procedure. Also in this case, we consider a subdivision of the set $\mathcal{C}(f;S_n)$. Finally, to obtain a greater smoothness, we apply the refinement technique three times.

In this test, to obtain the result of Figure 5 (right), 5185 nodes are considered in input.

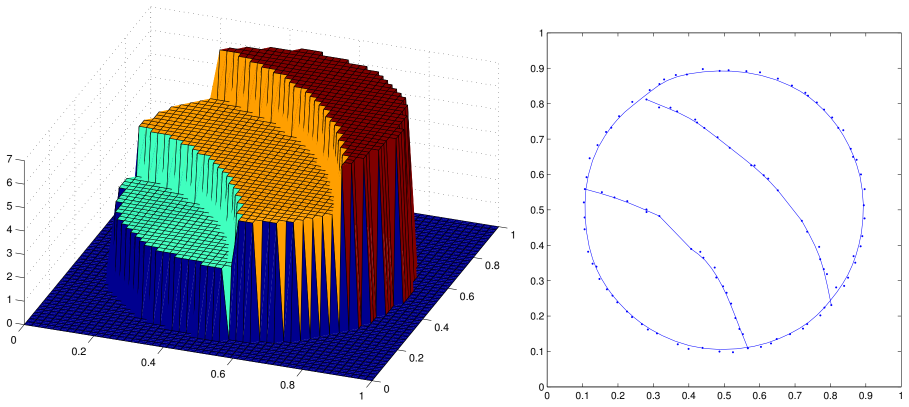

Figure 5: Test function f_3 (left) and polygonal line approximation curves (right).

Test function 4

The next test considers the function (see [16, 2, 14, 15]),

$$
f_4(x,y) = \begin{cases} (2-(x-0.8)^2)(2-(y-0.8)^2), & \text{if } \ y > l(x), \\ (1-(x-0.5)^2)(1-(y-0.2)^2), & \text{otherwise,} \end{cases}
$$

with $l(x) = 0.5 + 0.2\sin(5\pi x)/3$. The fault set is given by

$$
\Gamma = \{(t, l(t)) : 0 \leq t \leq 1\}
$$

and its jump size is almost constant on all the domain.

Figure 7 contains the test function f_4 (left) and the fault approximation obtained by least squares curve from the 49 points of the set $\mathcal{C}(f;S_n)$ (right),

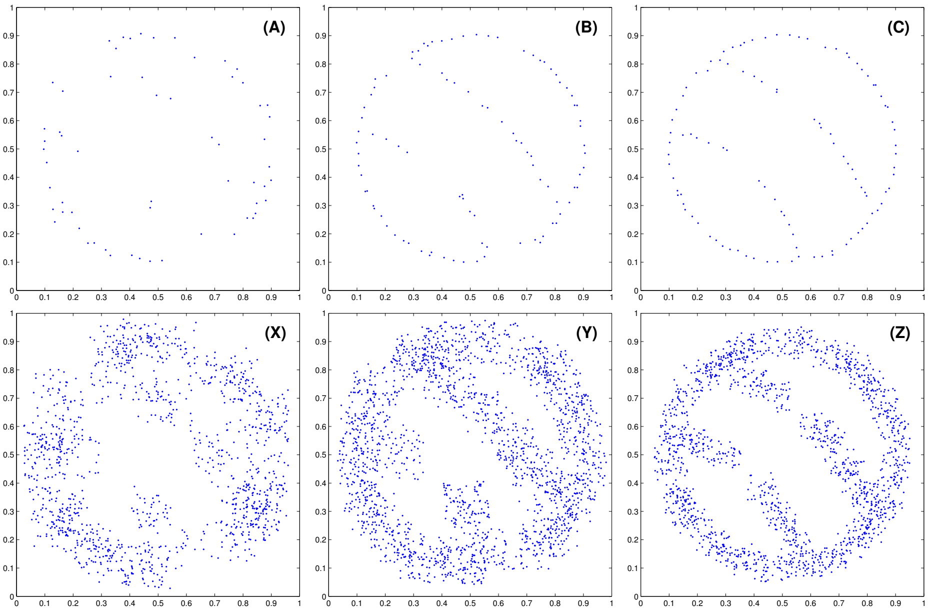

Figure 6: Output of the adaptive detection procedure for f_3.

while Figure 8 shows step by step the adaptive detection procedure. Splitting up $\mathcal{C}(f;S_n)$ in subsets do not produce considerable advantages.

To end the adaptive process, the number of the employed points is 2705.

<u>Test function 5</u>

Last, we consider the function (see [5])

$$f_5(x,y)=\begin{cases} g(x,y), & \text{if } \ y<\min(1/3x,0.25),\\ h(x,y), & \text{if } \ \min(1/3x,0.25)\le y<2x-0.5,\\ 6, & \text{otherwise,}\end{cases}$$

with

$$\begin{aligned} g(x,y) &= 1+1.4375\left(2-(x-1)^2\right)\exp(5\,(1-x)\,(4y-1)),\\ h(x,y) &= 1+\left(2-(x-1)^2\right)\left(2-(y-1)^2\right),\end{aligned}$$

and discontinuity set

$$\Gamma=\Gamma_1\cup\Gamma_2=\left\{\left(t,\frac{1}{3}t\right):0\le t\le 0.75\right\}\cup\left\{\left(t,2t-\frac{1}{2}\right):0.3\le t\le 0.75\right\}.$$

We observe that the jump size of fault line Γ_1 vanishes when x tends to 0.75.

In Figure 9 we set the test function and the best l_∞ approximation curves derived from the 46 points belonging to $\mathcal{C}(f;S_n)$. Figure 10 shows

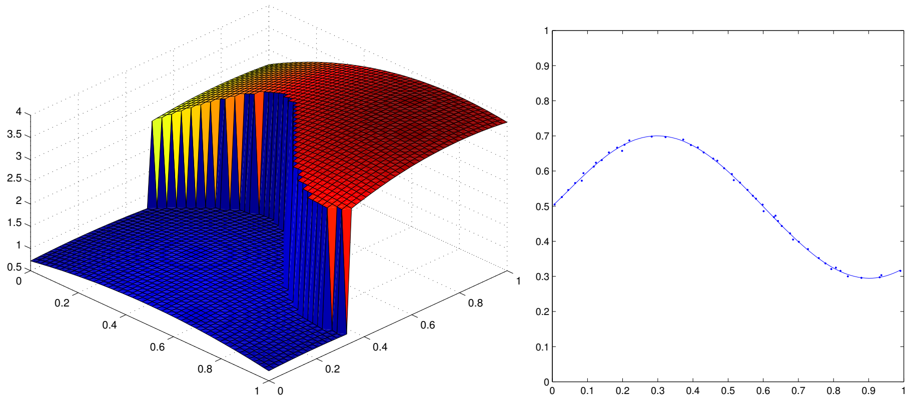

Figure 7: Test function f_4 (left) and least squares approximation curve (right).

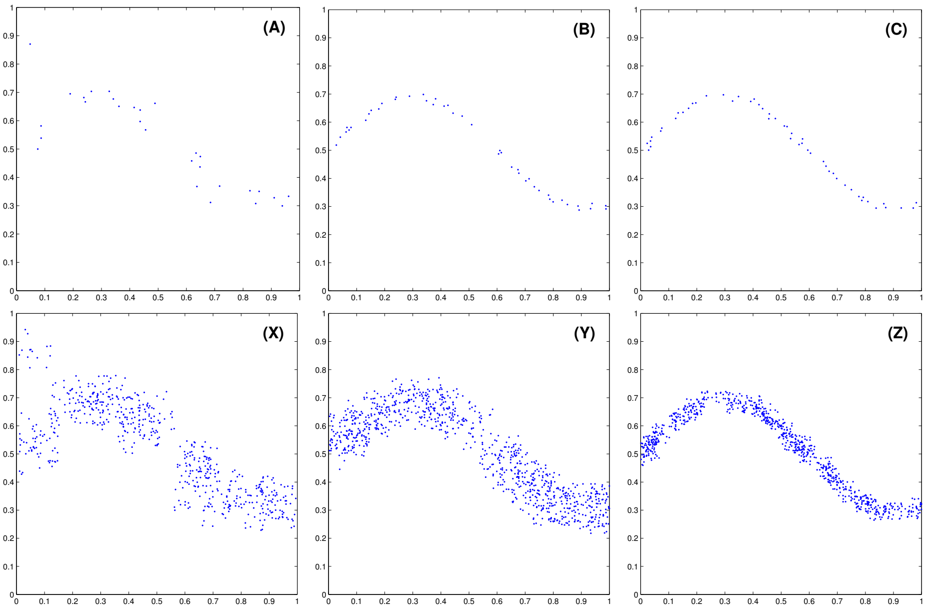

Figure 8: Output of the adaptive detection procedure for f_4.

the output of the adaptive detection process. All these results are obtained, subdividing the $\mathcal{C}(f;S_n)$ set into two subsets.

The entire detection procedure is completed using 2263 data points.

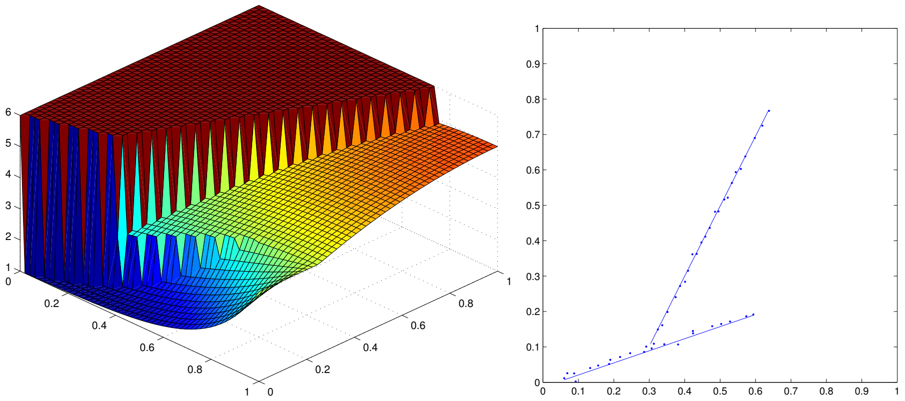

Figure 9: Test function f_5 (left) and best l_∞ approximation curves (right).

For each test function, we compute the maximum absolute error (MAE), the maximum relative error (MRE) and the root mean square error (RMSE), evaluating each fault line at 100 equispaced points in $0 \leq x \leq 1$. If a fault line is parallel or nearly parallel to y-axis, it is convenient first to rotate the coordinate axes and then to compute the errors. The results are reported in Table 1 and Table 2. For simplicity, we compare only least squares and best l_∞ approximation curves. In fact, also a polygonal line could be expressed analytically as a linear piecewise polynomial, but there would be too many pieces.

Test Function	MAE	MRE	RMSE
$f_1(x,y)$	$1.87486 \cdot 10^{-3}$	$3.12477 \cdot 10^{-3}$	$7.29411 \cdot 10^{-4}$
$f_2(x,y)$	$3.97532 \cdot 10^{-3}$	$9.93829 \cdot 10^{-3}$	$1.97662 \cdot 10^{-3}$
$f_3(x,y)$	$4.84865 \cdot 10^{-2}$	$4.56550 \cdot 10^{-1}$	$5.86301 \cdot 10^{-3}$
$f_4(x,y)$	$5.35214 \cdot 10^{-3}$	$1.77711 \cdot 10^{-2}$	$2.44892 \cdot 10^{-3}$
$f_5(x,y)$	$6.30713 \cdot 10^{-3}$	$3.16528 \cdot 10^{-1}$	$3.88514 \cdot 10^{-3}$

Table 1: Errors obtained by the least squares method.

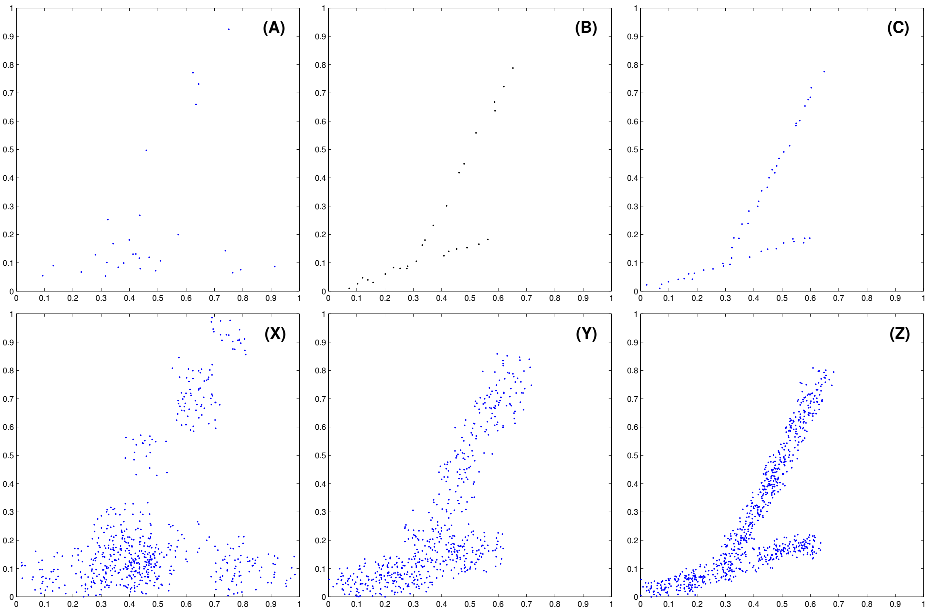

Figure 10: Output of the adaptive detection procedure for f_5.

Test Function	MAE	MRE	RMSE
$f_1(x,y)$	$7.39352 \cdot 10^{-3}$	$1.23225 \cdot 10^{-2}$	$3.02462 \cdot 10^{-3}$
$f_2(x,y)$	$7.41478 \cdot 10^{-3}$	$1.23230 \cdot 10^{-2}$	$2.57417 \cdot 10^{-3}$
$f_3(x,y)$	$5.75570 \cdot 10^{-2}$	$5.08761 \cdot 10^{-1}$	$7.57615 \cdot 10^{-3}$
$f_4(x,y)$	$1.67485 \cdot 10^{-2}$	$5.20277 \cdot 10^{-2}$	$5.71694 \cdot 10^{-3}$
$f_5(x,y)$	$1.29125 \cdot 10^{-2}$	$6.48021 \cdot 10^{-1}$	$7.74472 \cdot 10^{-3}$

Table 2: Errors obtained by the best l_∞ approximation method.

6 Final Remarks

In this paper, we have presented a method for the adaptive detection and approximation of fault lines in a discontinuous surface from scattered data. The approach we propose ensures good results, although the number of data points is meaningfully reduced. Indeed, in some applications, the search of information (data) is very expensive, so that a reduction of cost is a remarkable goal.

In our analysis, we suppose that the considered surface is smooth outside faults. If this does not happen, then the detection algorithm may identify false fault points. If the jump size varies rather fastly or vanishes, it follows that the optimal choice of a global threshold parameter σ_0 is an essential task. In future, a possible development could consist in splitting up the domain in a uniform grid and find a (local) threshold parameter for each grid cell. This idea should allow us to localize only the actual points of discontinuity, reducing the possibility of finding false fault points. However, it is yet an open problem and we are considering it.

Finally, we have tested the possibility to use a variable and adaptive grid in the cell-based search method, but the results have not showed meaningful advantages.

References

[1] Allasia, G., Cardinal basis interpolation on multivariate scattered data, *Nonlinear Analysis Forum* **6** (1) (2001), 1–13.

[2] Allasia, G., R. Besenghi, and A. De Rossi, A scattered data approximation scheme for the detection of fault lines, in T. Lyche, and L.L. Schumaker (eds.), *Mathematical methods for curves and surfaces: Oslo 2000*, Vanderbilt Univ. Press., Nashville TN, 2001, 25–34.

[3] Allasia, G., R. Besenghi, and R. Cavoretto, Accurate approximation of unknown fault lines from scattered data, submitted paper.

[4] Arcangéli, R., M.C. López de Silanes, and J.J. Torrens, *Multidimensional minimizing splines. Theory and applications.* Kluwer, Boston, MA, 2004.

[5] Arge, E., and M. Floater, Approximating scattered data with discontinuities, *Numer. Algorithms* **8** (1994), 149–166.

[6] Bentley, J.L., and J.H. Friedman, Data structures for range searching, *Comput. Surveys* **11** (1979), 397–409.

[7] Besenghi, R., and G. Allasia, Scattered data near-interpolation with application to discontinuous surfaces, in A. Cohen, C. Rabut, and L.L.

Schumaker (eds.), *Curves and surfaces fitting: Saint-Malo 1999*, Vanderbilt Univ. Press., Nashville TN, 2000, 75–84.

[8] Besenghi, R., M. Costanzo, and A. De Rossi, A parallel algorithm for modelling faulted surfaces from scattered data, *Int. J. Comput. Numer. Anal. Appl.* **3** (2003), 419–438.

[9] Crampton, A., and J.C. Mason, Detecting and approximating fault lines from randomly scattered data, *Numer. Algorithms* **39** (2005), 115–130.

[10] Fremming, N.P., Ø. Hjelle, and C. Tarrou, Surface modelling from scattered geological data, in M. Dæhlem, and A. Tveito (eds.), *Numerical methods and software tools in industrial mathematics*, Birkhäuser, Boston, 1997, 301–315.

[11] Gout, C., and C. Le Guyader, Segmentation of complex geophysical structures with well data, *Comput. Geosci.* **10** (2006), 361–372.

[12] Gout, C., C. Le Guyader, L. Romani, and A.G. Saint-Guirons, Approximation of surfaces with fault(s) and/or rapidly varying data, using a segmentation process, D^m-splines and the finite element method, *Numer. Algorithms* **48** (2008), 67–92.

[13] Gutzmer, T., and A. Iske, Detection of discontinuities in scattered data approximation, *Numer. Algorithms* **16** (1997), 155–170.

[14] López de Silanes, M.C., M.C. Parra, and J.J. Torrens, Vertical and oblique fault detection in explicit surfaces, *J. Comput. Appl. Math.* **140** (2002), 559–585.

[15] López de Silanes, M.C., M.C. Parra, and J.J. Torrens, On a new characterization of finite jump discontinuities and its application to vertical fault detection, *Math. Comput. Simul.* **77** (2008), 247–256.

[16] Parra, M.C., M.C. López de Silanes, and J.J. Torrens, Vertical fault detection from scattered data, *J. Comput. Appl. Math.* **73** (1996), 225–239.