Mesh-based and Meshless Design and Approximation of Scalar Functions

G. Patané, CNR-IMATI, Italy[1]

Abstract

In engineering, geographical applications, bio-informatics, and scientific visualisation, a variety of phenomena is described by data modelled as the values of a scalar function defined on a surface or a volume, and critical points (i.e., maxima, minima, saddles) usually represent a relevant information about the input data or an underlying phenomenon. Furthermore, the distribution of the critical points is crucial for geometry processing and shape analysis; e.g., for controlling the number of patches in quadrilateral remeshing and the number of nodes of Reeb graphs and Morse-Smale complexes. In this context, we address the design of a smooth function, whose maxima, minima, and saddles are selected by the user or imported from a template (e.g., Laplacian eigenfunctions, diffusion maps). In this way, we support the selection of the saddles of the resulting function and not only its extrema, which is one of the main limitations of previous work. Then, we discuss the meshless approximation of an input scalar function by preserving its persistent critical points and its local behaviour, as encoded by the spatial distribution and shape of the level-sets. Both problems are addressed by computing an implicit approximation with radial basis functions, which is independent of the discretisation of differential operators and of assumptions on the sampling of the input domain. This approximation allows us to introduce a meshless iso-contouring and classification of the critical points, which are characterised in terms of the differential properties of the meshless approximation and of the geometry of the input surface, as encoded by its first and second fundamental form. Furthermore, the computation is per-

[*] G. Patané, Consiglio Nazionale delle Ricerche, Istituto di Matematica Applicata e Tecnologie Informatiche, Via De Marini 6, 16149 Genova, Italy

Email address: patane@ge.imati.cnr.it (G. Patané, CNR-IMATI, Italy)

formed at an arbitrary resolution by locally refining the input surface and by applying differential calculus to the meshless approximation. As main applications, we consider the approximation and analysis of scalar functions on both 3D shapes and volumes in graphics, Geographic Information Systems, medicine, and bio-informatics.

Keywords: Scalar function design; meshless approximation; critical points; Laplacian matrix; visualisation; applications

1. Introduction

In several applications (e.g., engineering, geographical applications, bio-informatics, and scientific visualisation), scalar functions defined on a surface or a volume are used to model a relevant information about the input data or an underlying phenomenon. Controlling the distribution of the critical points (i.e., maxima, minima, saddles) of a scalar function during its design is also crucial for geometry processing and shape analysis. For instance, properly designing a scalar function with a prescribed set of critical points provides a flexible control over the number, shape, and size of the resulting quadrangular patches of remeshed surfaces [20, 30, 41], the number of nodes of the Reeb graph [44, 50] and of the Morse-Smale complex. However, harmonic functions allow the user to select only the number and position of its maxima and minima, with no control on the corresponding saddles. In a similar way, Laplacian eigenfunctions associated with small eigenvalues and diffusion maps at small scales are characterised by a generally low number of critical points, whose spatial location cannot be defined *a-priori*.

After a brief review of previous work (Sect. 2), our first goal (Sect. 3) is to design a smooth function whose maxima, minima, and saddles are selected by the user or imported from a template function. As main examples of template functions, we mention the Laplacian eigenfunctions [54] and the diffusion maps [12, 45], which are intrinsically defined by the input surface and their critical points represent relevant shape features, such as protrusions and high-curvature regions. The function with designed critical points is computed by applying either a mesh-based or a meshless approximation and by combining interpolating or least-squares constraints with the spectral properties of the Laplacian matrix. Then, the resulting problem reduces to the solution of a linear system. As novel contribution with respect to previous work, we allow the user to select the saddles of the designed function, and not only its extrema.

Our second goal (Sect. 4) is to compute a meshless approximation of an input scalar function by preserving its persistent critical points and its local behaviour (e.g., as encoded by the level-sets), instead of minimising only the approximation error as done by previous work. The proposed approximation preserves the topology of the *global structure* of the input scalar function by applying interpolating constraints on its values at the persistent critical points and at their 1-star vertices. It also approximates the *local details* according to the target accuracy, through least-squares constraints on the function values at a sub-sampling of the level-sets. The resulting function provides a good approximation accuracy without over-fitting the input data, and is robust against noise and local perturbations. As novelty with respect to previous work, we focus on the preservation of the spatial distribution and shape of the level-sets, which are useful to characterise the function behaviour. For instance, in medicine the shape of the level-sets is useful to discriminate between pathological and healthy cases, in Geographic Information Systems (GIS) it characterises the terrain morphology, and in Computer Graphics it encodes geometric properties of shape-driven functions, such as geodesic and diffusion distances.

The meshless techniques underlying the design and approximation of scalar functions with constrained critical points support the computation of the level-sets, the classification of the critical points, and the encoding of the approximations in a compact representation that saves input/output space (Sect. 5). All these computations are performed at an arbitrary resolution by locally refining the input surface and by applying differential calculus to the meshless approximation.

2. Previous work

We briefly summarise previous work on the classification and simplification of the critical points, and on the approximation of discrete data with implicit methods and radial basis functions.

Critical points. Given a $\mathcal{C}^1$ function $f : \mathcal{M} \rightarrow \mathbb{R}$ defined on a smooth 2-manifold surface $\mathcal{M}$, the *critical points* of f are defined as those points $\mathbf{p} \in \mathcal{M}$ such that $\nabla f(\mathbf{p}) = \mathbf{0}$ and they correspond to the maxima, minima, and saddles of f. For polyhedral surfaces, the method described in [7] classifies a vertex according to the values of f on its neighborhood. If $\mathcal{M}$ is a triangle mesh, then the vertex $\mathbf{p}$ is a *maximum* or a *minimum* if its function value is higher or lower than those ones on its 1-*star*, respectively. We briefly remind that the 1-star of a vertex $\mathbf{p}_i$ is defined as the set of vertices incident to $\mathbf{p}_i$; i.e., $\{\mathbf{p}_j : \ (\mathbf{p}_i, \mathbf{p}_j)$ edge$\}$. If two or more iso-curves related to the same iso-value share a vertex $\mathbf{p}$, then $\mathbf{p}$ is a *saddle*. Those points that do not fall in the previous classification are defined as *regular*. According to this classification, the *Euler formula* $\chi(\mathcal{M}) = m - s + M$ provides a link between the number of critical points (i.e., m minima, M maxima, s saddles) of the input scalar function and the Euler characteristic $\chi(\mathcal{M})$ of $\mathcal{M}$. For details on the classification of critical points for piecewise linear scalar functions on triangle meshes, we refer the reader to [11] and Sect. 3.1.

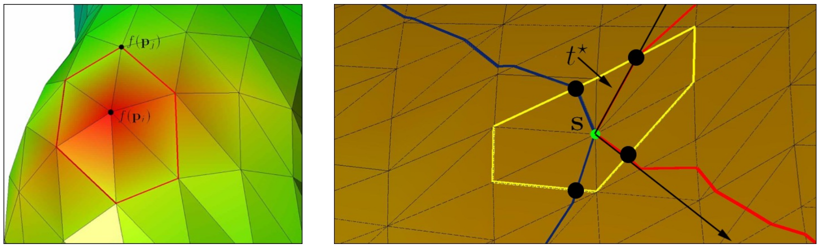
Topological simplification of critical points based on persistence. Given a scalar function $f : \mathcal{M} \rightarrow \mathbb{R}$ with a large number of critical points associated with a low variation of the corresponding f-values, previous work [11, 22] defines a topological hierarchy for f that is constructed by performing a progressive simplification of the Morse complex $\mathcal{F}$ of f through the cancellation of pairs of critical points. Then, the critical points are paired by visiting $\mathcal{M}$ with respect to the reordering of its vertices according to increasing values of f. The importance weight associated with the pair $(\mathbf{p}_i, \mathbf{p}_j)$ is measured as the *persistence* of $\mathbf{p}_i$, $\mathbf{p}_j$, that is, $|f(\mathbf{p}_i) - f(\mathbf{p}_j)|$. The local updates of the complex are performed by iteratively removing those pairs with the lowest persistence and reconnecting the neighbours of the removed nodes. Each node removal affects the number and configuration of the critical points of $\mathcal{F}$ without changing f. Therefore, the simplification provides a hierarchy for f where each Morse complex $\mathcal{F}^{(k)}$ is not associated with a corresponding scalar function $f^{(k)}$ on $\mathcal{M}$.

In [23], the input scalar function f is replaced with a new function $\tilde{f}$ that has the same points of persistence of f higher than a given threshold ε and the $\mathcal{L}_\infty$-error between f and $\tilde{f}$ is lower than ε. The ε-*simplification* of the structure of f and the construction of $\tilde{f}$ are based on an iterative process, which cancels minimum-saddle pairs by sweeping the vertices from bottom to top and lower the saddles that belong to a pair of persistence lower than ε.

To improve the approximation accuracy and the smoothness of the solution, topological simplification is combined with the minimisation of the $\mathcal{L}_\infty$ error between the input and the simplified functions [61, 59], isotropic Laplacian [20, 41, 56] or Gaussian [35] filters, and least-squares approximation constrained to preserve the persistent critical points [47, 61].

Meshless approximation with Moving-Least-Squares methods and radial basis functions. Given a point set $\mathcal{P}$, the *moving least-squares* (MLS) *surface* $\mathcal{M}$ underlying $\mathcal{P}$ is defined by a projection operator $\psi_\mathcal{P} : \mathbb{R}^3 \rightarrow \mathcal{M}$, which maps an arbitrary point $\mathbf{p}$ onto $\mathcal{M}$ [2, 3, 4, 34]. In particular, each point of $\mathcal{M}$ is a stationary point of $\psi_\mathcal{P}$; i.e., $\mathbf{p} \in \mathcal{M}$ implies $\psi_\mathcal{P}(\mathbf{p}) = \mathbf{p}$. Previous work on MLS surfaces assumes that the normal of the approximating tangent frame is the surface normal. Since this assumption is generally not satisfied, [1] proposes a different expression of the implicit function g that allows us to exactly compute the surface normals. More precisely, the surface $\mathcal{M}$ is defined as the zero-set $\{\mathbf{p} : g(\mathbf{p}) = 0\}$ of the implicit function $g(\mathbf{p}) := \langle \mathbf{p} - \mathbf{b}(\mathbf{p}), \mathbf{n}(\mathbf{p}) \rangle$, where $\mathbf{n}(\mathbf{p})$ is the oriented normal at $\mathbf{p}$, which is computed as previously discussed, and $\mathbf{b}(\mathbf{p})$ is the weighted average of points at $\mathbf{p}$, i.e. $\mathbf{b}(\mathbf{p}) := \frac{\sum_{i=1}^n \theta(\|\mathbf{p}-\mathbf{p}_i\|_2)\mathbf{p}_i}{\sum_{i=1}^n \theta(\|\mathbf{p}-\mathbf{p}_i\|_2)}$. Here, θ is a decreasing weighting function, e.g. $\theta(t) := \exp(-t^2/h^2)$, where h is a Gaussian scale parameter that defines the width of the kernel. A possible choice of h is $h := \sigma/\sqrt{3}$, where σ is the distance between $\mathbf{p}_i$ and the points of its k-nearest neighborhood; for more details on the choice of h, we refer the reader to [19].

In implicit modeling [10], a point set $\mathcal{P} := \{\mathbf{p}_i\}_{i=1}^n$ is approximated by the isosurface $\Sigma := \{\mathbf{p} \in \mathbb{R}^3 : \ g(\mathbf{p}) = 0\}$, where $g : \mathbb{R}^3 \rightarrow \mathbb{R}$ is an implicit function. In this context, implicit approximation [6, 21, 37, 53] computes $g(\mathbf{p}) := \sum_{i=1}^n a_i \phi_i(\mathbf{p})$ as a linear combination of the basis elements $\mathcal{B} := \{\phi(\|\mathbf{p} - \mathbf{p}_i\|_2)\}_{i=1}^n$, where ϕ is the kernel function. Depending on the properties of ϕ and of the corresponding approximation scheme, we distinguish globally- [15, 62] and compactly- [64, 39, 43] supported radial basis functions, and the partition of unity [42, 66]. Here, the *support* of a function



(a)

(b)

Figure 1: (a) 1-star of a vertex used for the design and classification of the critical points; maximum achieved by choosing the f-values such that $f(\mathbf{p}_j) > f(\mathbf{p}_i)$, $j \in \mathcal{N}(i)$. (b) Neighbour and level-set (blue, red curves) at a saddle $\mathbf{s}$ and points (black dot) that define the mixed link (c.f., Eq. (1)). Here, $t^\star$ is a triangle of the neighbour of $\mathbf{s}$ intersected by the corresponding level-set.

$g : \mathbb{R}^3 \rightarrow \mathbb{R}$ is defined as the set $\text{supp}(g) := \overline{\{\mathbf{p} \in \mathbb{R}^3 : g(\mathbf{p}) \neq 0\}}$. If $\text{supp}(g) := \mathbb{R}^3$, then g has *global support*. Main examples are the cubic ($\phi(\sigma, r) = \sigma r^3$), Gaussian ($\phi(\sigma, r) = \exp(-\sigma r^2)$), thin plate spline ($\phi(\sigma, r) = r^2 \log(\sigma r)$), inverse multi-quadratic ($\phi(\sigma, r) = (r^2 + \sigma^2)^{-1/2}$), and multi-quadratic ($\phi(\sigma, r) = (r^2 + \sigma^2)^{1/2}$) kernels.

3. Designing scalar function with constrained critical points

We aim at designing a smooth function whose critical points are selected by the user or imported from a template function, such as the Laplacian and the diffusion functions (Sect. 3.1). The function with designed critical points is computed by applying a mesh (Sect. 3.2) or meshless (Sect. 3.3) approximation and reduces to the solution of a linear system. Finally (Sect. 3.4), we discuss the main properties of the designed functions (e.g., number of critical points, selection of the type of constraints). In all the paper examples, the values of the input scalar function belong to the interval $[0, 1]$. In a similar way, the coordinates of the vertices of the input surface are normalised in such a way that the surface belongs to the unitary bounding box.

3.1. Selection of the critical points and values

Given a triangle mesh $\mathcal{M}$ with $\mathcal{P} := \{\mathbf{p}_i\}_{i=1}^n$ set of vertices, we want to design a piecewise linear function $f : \mathcal{M} \rightarrow \mathbb{R}$ by selecting the position and values of its critical points. To define a function f with a maximum at a vertex $\mathbf{p}_i$ (Fig. 1(a)), it is enough to define $f(\mathbf{p}_i)$ greater than the f-values at the points of its 1-star $\mathcal{N}(i) := \{j : \ (i,j) \text{ edge}\}$. A minimum is defined in a similar way. To design a saddle of f at $\mathbf{p}_i$ (Fig. 1(b)), let

$$Lk(i) := \{j_1, \ldots, j_k \in \mathcal{N}(i) : \ (j_s, j_{s+1})_{s=1}^{k-1} \text{ edges of } \mathcal{M}\}$$

be the *link* of i, and

$$\begin{aligned} Lk^{\pm}(i) := \{j_s \in Lk(i) : f(\mathbf{p}_{j_{s+1}}) > f(\mathbf{p}_i) > f(\mathbf{p}_{j_s}) \text{ or} \\ f(\mathbf{p}_{j_{s+1}}) < f(\mathbf{p}_i) < f(\mathbf{p}_{j_s})\}, \qquad j_{k+1} := j_k, \end{aligned} \tag{1}$$

the *mixed link* of i. Recalling that the cardinality of the mixed link of a saddle of multiplicity m is $2+2m$, a vertex $\mathbf{p}_i$ with a neighbouring vertices locates a saddle of multiplicity lower than or equal to $\lfloor (a-2)/2 \rfloor$, where $\lfloor \cdot \rfloor$ is the floor symbol. If this condition is not satisfied, then we either reduce the multiplicity of the saddle according to the previous relation, or we locally update the mesh connectivity at $\mathcal{N}(i)$ (e.g., by splitting each triangle incident to i into two sub-triangles). Indeed, the f-values at $\mathbf{p}_i$ and at its neighbouring points are chosen in such a way that the conditions in Eq. (1) are satisfied.

Let $\mathcal{C}$ be the set of critical points (i.e., maxima, minima, saddles) and let us consider the set $\tilde{\mathcal{I}} := \{j \in \mathcal{N}(i), \ i \in \mathcal{C}\}$ of the vertices of the 1-stars of the critical points. Then, let $\mathcal{I} := \mathcal{C} \cup \tilde{\mathcal{I}}$ be the set of designed critical points and of the corresponding 1-stars' vertices, without repetitions; we also assume that k is the cardinality of the set $\mathcal{I}$. To define the function $f : \mathcal{P} \to \mathbb{R}$ at all the points of $\mathcal{P}$ starting from the designed critical points and the values $\{f(\mathbf{p}_i)\}_{i \in \mathcal{I}}$, we propose a mesh-based (Sect. 3.2) and meshless (Sect. 3.3) computation. To this end, we reformulate the problem as the computation of a function that satisfies the conditions $f(\mathbf{p}_i) = f_i$, $i \in \mathcal{I}$, in an exact or least-squares way. By definition, the set $\tilde{\mathcal{C}}$ of critical points of f contains $\mathcal{C}$.

3.2. Mesh-based design of scalar functions with constrained critical points

We introduce a mesh-based design of the scalar function with constrained critical points by applying interpolating and least-squares constraints.

Design of scalar functions with constrained critical points. According to the previous discussion, the smooth scalar function $f : \mathcal{M} \to \mathbb{R}$, with a set $\mathcal{C}$ of critical points, is

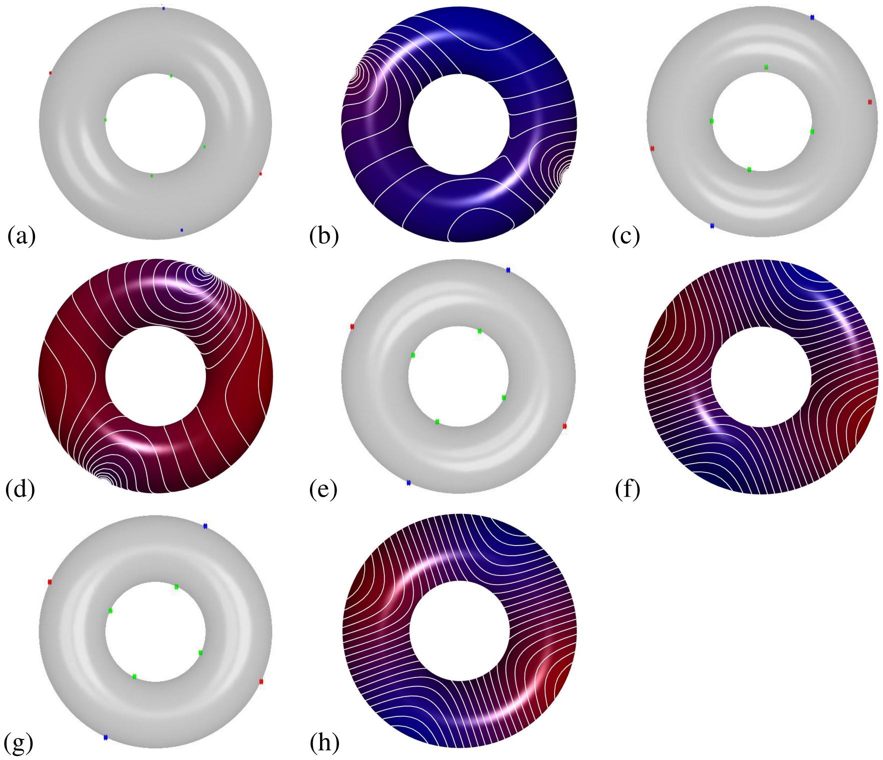
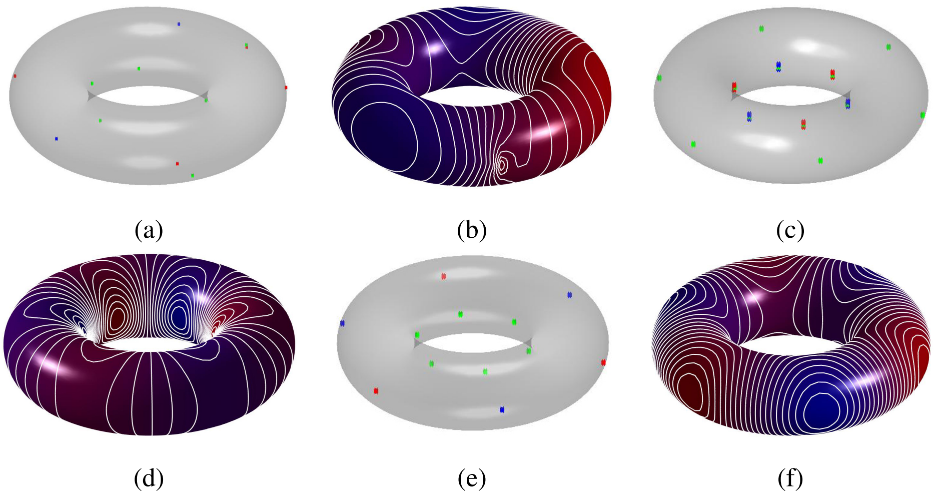


Figure 2: (a,c,e,g) Critical points and (b,d,f,h) of the designed scalar function. Each scalar function has been designed by applying the mesh-based approach with interpolating constraints at (a,b) 2 maxima, (c,d) 2 minima, (e,f) 2 maxima and 2 minima, (g,h) 4 saddles. Pictures in (a,c,e,g) show the constrained maxima and minima as well as the saddles of the designed scalar function. Maxima, minima, and saddles are represented as red, blue, and green dots, respectively.

defined as the solution of the constrained minimisation problem (Fig. 2)

$$\min_{\mathbf{f}\in\mathbb{R}^n}\{\|\mathbf{L}\mathbf{f}\|_2\}, \qquad f(\mathbf{p}_i) := f_i, \quad i \in \mathcal{I}, \tag{2}$$

where $\mathbf{L}$ is the Laplacian matrix with Voronoi-cotangent weights [18, 52]. We briefly recall that the Laplacian matrix discretises the Laplace-Beltrami operator and $\|\mathbf{L}\mathbf{f}\|_2$ represents the Dirichlet energy of f. Since this last term is null or close to zero for constant or smooth functions, its minimisation is aimed at controlling the oscillations of the solution to (2). To compute the solution to Eq. (2), we consider the complement $\mathcal{I}^C$


(a) (b) (c)

(d) (e) (f)

Figure 3: (a,c,e) Critical points and (b,d,f) level-sets of the designed scalar function achieved with least-squares constraints (mesh-based approach) at (a,b) 3 maxima, (c,d) 6 saddles (internal circle), (e,f) 3 maxima, 3 minima, and 6 saddles. Pictures in (a,c,e) show the constrained maxima and minima as well as the saddles of the designed scalar function. Maxima, minima, and saddles are represented as red, blue, and green dots, respectively.

of $\mathcal{I}$ and for $i \in \mathcal{I}^C$ we have that

$$\begin{aligned}(\mathbf{Lf})_i &= l_{ii} f(\mathbf{p}_i) - \sum_{j \in \mathcal{N}(i)} l_{ij} f(\mathbf{p}_j) \\ &= l_{ii} f(\mathbf{p}_i) - \sum_{j \in \mathcal{N}(i) \cap \mathcal{I}^C} l_{ij} f(\mathbf{p}_j) - \sum_{j \in \mathcal{N}(i) \cap \mathcal{I}} l_{ij} f_j.\end{aligned}$$

Indicating with $\mathbf{g} := (f(\mathbf{p}_i))_{i \in \mathcal{I}^C} \in \mathbb{R}^{n-k}$ the set of unknowns, the previous identities can be written in matrix form as $\tilde{\mathbf{L}}\mathbf{g} - \mathbf{b}$. Here, $\tilde{\mathbf{L}} \in \mathbb{R}^{(n-k)\times(n-k)}$ is the matrix achieved by cancelling the i^{th}-row and i^{th}-column of $\mathbf{L}$, $i \in \mathcal{I}$, and the entries of the constant term $\mathbf{b} \in \mathbb{R}^{n-k}$ are given by $\sum_{j \in \mathcal{N}(i) \cap \mathcal{I}} l_{ij} f_j$, $i \in \mathcal{I}^C$. Therefore, the constrained least-squares minimisation problem (2) is equivalent to $\min_{\mathbf{x} \in \mathbb{R}^{n-k}} \{\|\tilde{\mathbf{L}}\mathbf{x} - \mathbf{b}\|_2\}$, where the solution $\mathbf{x}$ provides the f-values at the points associated with $\mathcal{I}^C$. Since the rank of $\mathbf{L}$ is $n-1$, the rank of $\tilde{\mathbf{L}}$ is $n-k$, $k \geq 1$ and the vector $\mathbf{x}$ is the unique solution to the sparse linear system $\tilde{\mathbf{L}}\mathbf{x} = \mathbf{b}$. In Fig. 2(e,f), (g,h), the selection of critical points at symmetric locations makes the behaviour of the resulting scalar function symmetric.

Design of scalar functions with least-squares constraints on critical points. The function $f:\mathcal{M}\rightarrow\mathbb{R}$, which is the best compromise between the least-squares constraint $\sum_{i\in\mathcal{I}}|f(\mathbf{p}_i)-f_i|^2$ and the smoothness term $\|\mathbf{Lf}\|_2$, is defined as the solution to the problem (Fig. 3)

$$\min_{\mathbf{f}\in\mathbb{R}^n}\left\{\mathcal{F}(\mathbf{f})\right\}, \qquad \mathcal{F}(\mathbf{f}):=\varepsilon\sum_{i\in\mathcal{I}}|f(\mathbf{p}_i)-f_i|^2+\|\mathbf{Lf}\|_2^2. \tag{3}$$

Then, the derivative of $\mathcal{F}$ with respect to the unknown $f(\mathbf{p}_k)$ is

$$\begin{cases} \sum_{i,j=1}^n l_{ij}l_{ik}f(\mathbf{p}_j)+\varepsilon(f(\mathbf{p}_k)-f_k), & k\in\mathcal{I}, \\ \sum_{i,j=1}^n l_{ij}l_{ik}f(\mathbf{p}_j), & k\in\mathcal{I}^C, \end{cases}$$

which is re-written in matrix form as $(\mathbf{L}^\top\mathbf{L}+\varepsilon\Gamma)\mathbf{f}=\varepsilon\mathbf{b}$, with

$$\Gamma_{ij}:=\begin{cases} 1 & i=j\in\mathcal{I}, \\ 0 & \text{else}, \end{cases} \qquad b_i:=\begin{cases} f_i & i\in\mathcal{I}, \\ 0 & i\in\mathcal{I}^C, \end{cases} \qquad \mathbf{b}\in\mathbb{R}^n, \quad \Gamma\in\mathbb{R}^{n\times n}.$$

The coefficient matrix $\mathbf{L}^\top\mathbf{L}+\varepsilon\Gamma$, $\varepsilon>0$, is symmetric, sparse, and positive definite; indeed, our problem admits a unique solution. For the selection of ε, which represents the trade-off between approximation accuracy and smoothness of the solution, we apply the L-curve criterion [28]; an alternative is to consider statistical criteria [63]. For surfaces with a high genus or a high number of designed saddles, it is generally preferable to apply least-squares instead of interpolating constraints in order to reduce the number of additional critical points. For instance, least-squares constraints in Fig. 3 are associated with a regular distribution of smooth level-sets, even in case of a large number of selected critical points (Fig. 3(d)).

For the mesh-based design of a scalar function with constrained critical points, we can also consider the Laplacian matrix $\tilde{\mathbf{L}}:=\mathbf{B}^{-1}\mathbf{L}$ with linear FEM weights [54], where $\mathbf{L}$ is the Laplacian matrix with cotangent weights and $\mathbf{B}$ is the mass matrix, which encodes the variation of the triangles' area. In this case, we select the norm $\|\mathbf{f}\|_{\mathbf{B}}^2:=\mathbf{f}^\top\mathbf{Bf}$ induced by the symmetric, positive-definite mass matrix $\mathbf{B}$, instead of the $\mathcal{L}^2$-norm used in Eq. (2) and Eq. (3). The $\mathbf{B}$-norm generally improves the robustness of the resulting scalar function with respect to an irregular sampling of the input surface; in particular, the proposed formulation and its properties remain unchanged.

For the design of scalar functions with interpolating constraints, the Dirichlet energy $\|\mathbf{Lf}\|_{\mathbf{B}}$ can be re-written in terms of the $\mathcal{L}^2$-norm as $\|\mathbf{Lf}\|_{\mathbf{B}} = \|\mathbf{B}^{1/2}\mathbf{Lf}\|_2$, where $\mathbf{B}^{1/2}$ is the square-root of $\mathbf{B}$. Then, we can either compute $\mathbf{B}^{1/2}$ [27] (Ch. 4) or lump $\mathbf{B}$ to the diagonal matrix whose entries are the areas of the Voronoi regions of the mesh vertices. For the mesh-based design of scalar functions with least-squares constraints, the normal equation becomes $(\mathbf{L}^\top\mathbf{BL}+\varepsilon\mathbf{B}\Gamma\mathbf{B})\mathbf{f} = \varepsilon\mathbf{B}^\top\mathbf{b}$. For our experiments, we have selected the Voronoi-cotangent weights (c.f., Eq. (2)).

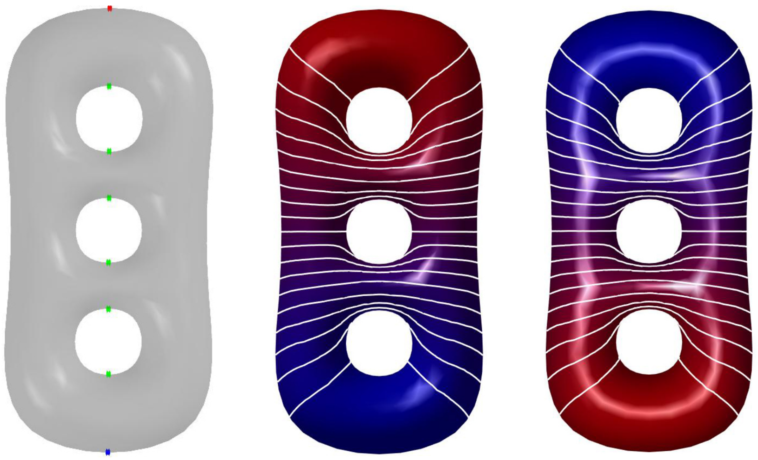
3.3. Meshless design of scalar functions with constrained critical points

Analogously to Eq. (2), we compute the function $u : \mathbb{R}^3 \rightarrow \mathbb{R}$ that satisfies the conditions $u(\mathbf{p}_i) = f_i$, $i \in \mathcal{I}$, and is defined as a linear combination $u(\mathbf{p}) := \sum_{i\in\mathcal{I}} a_i\phi_i(\mathbf{p})$ [6] of RBFs $\phi_i(\mathbf{p}) := \phi(\|\mathbf{p}-\mathbf{c}_i\|_2)$ generated by the kernel $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$. The coefficients $\mathbf{a} := (a_i)_{i\in\mathcal{I}}$ are the solution to the linear system $\Phi\mathbf{a} = \tilde{\mathbf{f}}$, where Φ is the Gram matrix induced by $(\phi,\mathcal{C})$, and $\tilde{\mathbf{f}} := (f(\mathbf{p}_i))_{i\in\mathcal{I}}$. The solution to our problem is the piecewise linear function f whose values at the vertices are defined as $f(\mathbf{p}_i) := u(\mathbf{p}_i)$, $i = 1,\ldots,n$. An important constraint on the kernel is its global support; in fact, using only compactly-supported RBFs would provide artefacts where the supports of the basis functions intersect and a low regularity of the level-sets. Indeed, for our experiments we have selected the Gaussian and cubic kernels (Fig. 4).

Selecting the RBFs induced by the harmonic kernel $\phi(t) := t^{-1}$ and centred at the points in $\mathcal{C} := \{\mathbf{c}_i\}_{i\in\mathcal{I}}$, u is harmonic in $\mathcal{D} := \mathbb{R}^3\backslash\mathcal{C}$, as a linear combination of harmonic functions, is the unique solution to the Laplace equation $\Delta u(\mathbf{p}) = 0$, with initial conditions $u(\mathbf{p}_i) = f(\mathbf{p}_i)$, $i \in \mathcal{I}$, and minimises the Dirichlet energy. Since the function u is evaluated at $\mathbf{p}_i$ and ϕ_i is not defined at $\mathbf{c}_i$, as centres of the RBFs we select the offset points $\mathbf{c}_i := \mathbf{p}_i + \delta\mathbf{n}_i$, $\delta > 0$, which are obtained by translating $\mathbf{p}_i$ along the corresponding normal $\mathbf{n}_i$ to the surface.

3.4. Properties of the designed scalar functions

We discuss how the selected critical points affect the overall number of critical points of the designed function and the main criteria for the selection of interpolating or least-squares constraints.



(a) (b) $\phi(\sigma, r) = \exp(-\sigma r^2)$ (c) $\phi(r) = r^3$

Figure 4: (a) Selected critical points: 1 maximum (red dot), 1 minimum (blue dot), 6 saddles (green dots) and level-sets of the scalar function designed with the meshless approach and (b,c) two generating kernels.

Critical points of the designed function. Recalling the Euler formula (Sect. 2), the number of saddles of the designed scalar function is $s = 2(g-1)+m+M$, where g is the genus of the input domain. Combining this relation with the smoothness of the approximation due to the minimisation of the Laplacian energy, we expect a low number of additional critical points according to the Euler formula, the smoothness and regular distribution of the level-sets on the input surface. Choosing 1 maximum and 1 minimum, the minimal number of saddles is $s = 2g$. The selection of up-to $2g$ saddles and interpolating constraints, together with the minimisation of the $\mathcal{L}^2$-norm (*mesh design*) or the Dirichlet energy (*meshless design*), provide the smoothest scalar function that satisfies the selected conditions on the number and location of the critical points. Selecting more than $2g$ saddles induces additional extrema, which are necessary to guarantee the validity of the Euler formula. Assuming that the input scalar function is general (i.e., $f(\mathbf{p}_i) \neq f(\mathbf{p}_j)$, (i, j) edge), from the Euler formula we get that the additional critical points $\tilde{m}$, $\tilde{M}$, $\tilde{s}$, satisfy the "*nullity relation*" $\tilde{m} - \tilde{s} + \tilde{M} = 0$.

Constraints' selection. Least-squares constraints are preferable to interpolating constraints in case of a large number of selected critical points, in order to minimise the number of additional critical points that are necessary to satisfy the Euler formula. Reducing the trade-off ε in the mesh-based design with least-squares constraints (c.f., Eq. (3)) will remove low persistent critical points.

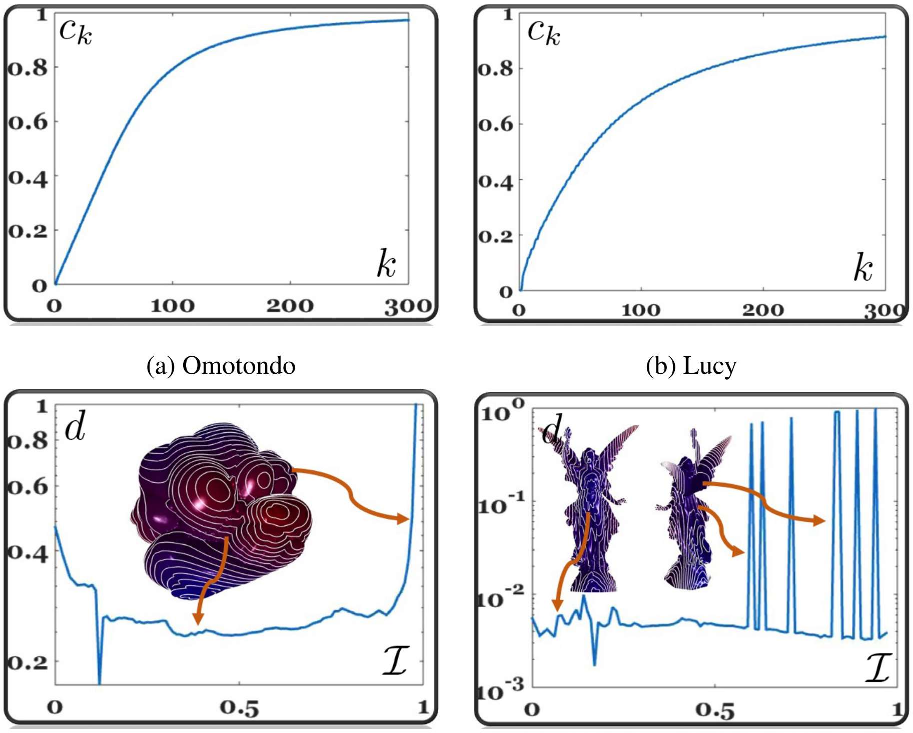
An alternative to the user selection of the f-values at the designed critical points is to apply pre-defined templates of the f-values for maxima, minima, and saddles, which can be defined only once (e.g., on a 6-neighbourhood stencil) or imported from the critical points of template functions (e.g., Laplacian eigenfunctions, diffusion maps). In fact, any template can be adapted to a different configuration of a neighbour of a critical point by applying a local simplification/subdivision to the corresponding 1-star and a piecewise linear approximation to extend the f-values to the updated neighbours.

4. Adaptive approximation of scalar functions

Given a scalar function $f : \mathcal{P} \rightarrow \mathbb{R}$, we tackle the problem of computing a smooth and meshless approximation $g : \mathcal{M} \rightarrow \mathbb{R}$ of f, which preserves its persistent critical points and its local behaviour, as encoded by the level-sets. As novelty with respect to previous work, we focus on the preservation of the spatial distribution and shape of the level-sets, which are useful to correctly represent and analyse the behaviour of f. In order to preserve the persistent critical points and the distribution of the level-sets, the proposed approach combines the techniques introduced in Sect. 3 with descriptors of the local behaviour of functions on surfaces. The resulting approximation is independent of the domain discretisation and provides a compact representation of the input function, which saves input/output space. To characterise the distribution of the level-sets and guide the meshless and adaptive approximation of the input function, we introduce the *global coverage value*, which measures the surface area covered by the level-sets, and the *distortion map*, which measures their spatial distribution.

Discrete and global coverage value. Given a sampling $\mathcal{I}_k$ of the image of f with k iso-values, we define the *coverage value of* $(\mathcal{M}, f)$ *with respect to* $\mathcal{I}_k$ as

$$c_k := \frac{\sum_{\alpha \in \mathcal{I}_k} \text{area}(\gamma_\alpha)}{\text{area}(\mathcal{M})} \in [0,1],$$



(a) Omotondo

(b) Lucy

(c)

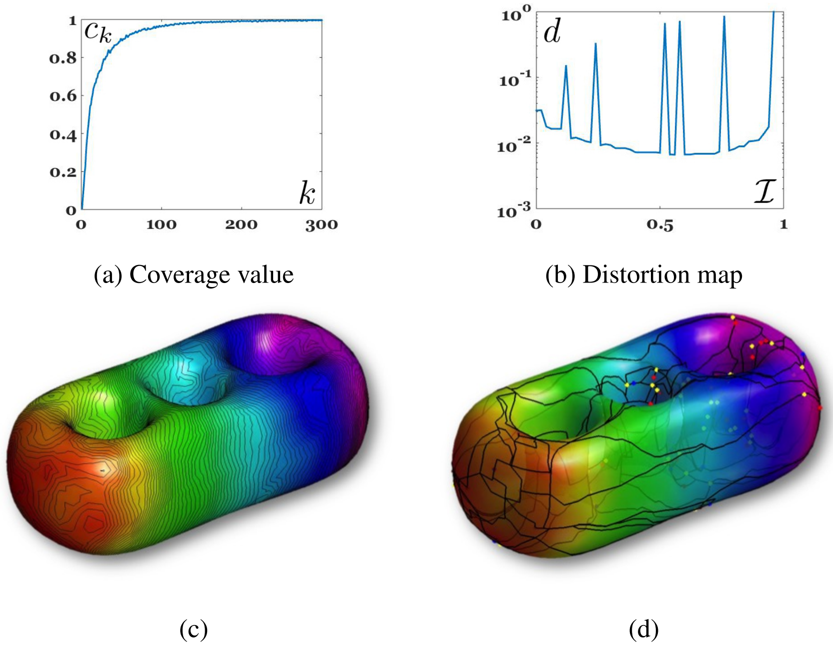
(d)

Figure 5: (a,b) Coverage values (y-axis) with respect to different samplings (x-axis) of the image of f. (c,d) Distortion values (y-axis) with respect to the partitioning (x-axis) of the image $[0,1]$ of the input scalar function. In (b), the global coverage is lower than 1; in fact, the scalar function has a number of irregularly-distributed level-sets higher than (c). This aspect is further stressed by the spikes of the distribution map.

where $\gamma_\alpha := f^{-1}(\alpha) = \{\mathbf{p} \in \mathcal{M} : f(\mathbf{p}) = \alpha\}$ is the level-set of f associated with the iso-value α, $\text{area}(\gamma_\alpha) = \sum_{t \cap \gamma_\alpha \neq \emptyset} \text{area}(t)$ is the sum of the areas of the triangles t intersected by γ_α, and each intersected triangle is counted only once. For its evaluation, we identify the triangles t intersected by γ_α in linear time; i.e., at least one edge (i,j) of t satisfies the relation $(f(\mathbf{p}_i) \leq \alpha \,\&\, f(\mathbf{p}_j) \geq \alpha)$ or vice-versa. To make the coverage value independent of the selected sampling, we apply a dihcotomic subdivision of the image of the input function into 2^k samples and define the *global coverage* as $c := \lim_{k \rightarrow +\infty} c_k$. In Fig. 5(a,b), we report the coverage values c_k (y-axis) with respect to a sampling of the image of the input scalar function f with k samples (x-axis). The subdivision stops when c_k becomes stationary; i.e., the difference between two consecutive coverage values is lower than a given threshold (e.g., 10^{-4}, in our implementation). The corresponding image partitioning is called *uniform optimal partitioning*; here, the optimality refers to the fact that the level-sets induced by the optimal partitioning accurately represent the behaviour of the input function. On the one hand, a uniform partitioning of the image of f with a number of parts lower than the optimal partitioning will provide a less accurate representation of the behaviour of f in terms of the corresponding level-sets. On the other hand, a uniform partitioning of f with a number of parts higher than the optimal partitioning will provide a redundant representation of the behaviour of f. If the global coverage is close to 1, then the level-sets are distributed on the whole surface; as far as it becomes close to 0, larger regions of $\mathcal{M}$ do not contain level-sets.

Distortion map. Given the optimal partitioning $\mathcal{I}$, we further characterise the distribution of the level-sets by defining the *distortion map* $d : \mathcal{I} \rightarrow \mathbb{R}^+$ at α_i as the Hausdorff distance between two consecutive level-sets $f^{-1}(\alpha_i)$, $f^{-1}(\alpha_{i+1})$, where $[\alpha_i, \alpha_{i+1}]$ does not contain iso-values associated with saddles (Fig. 5(c,d)). Removing these critical iso-values, the Hausdorff distance is evaluated between two level-sets with the same topology. To this end, each level-set is discretised as a poly-line, which is computed as the intersection between the mesh edges and the level-set; then, the Hausdorff distance between two consecutive poly-lines is evaluated among these intersection points. If the distortion map is constant, then the level-sets are uniformly distributed on $\mathcal{M}$; otherwise, a local irregularity of the map at α_i indicates that the interval $[\alpha_i, \alpha_{i+1}]$ is distorted when it is mapped on the input surface and this distortion is measured by the Hausdorff distance between $f^{-1}(\alpha_i)$ and $f^{-1}(\alpha_{i+1})$. Alternatively, we can replace the Hausdorff distance with the geodesic distance [32], which is more appropriate to take into account the geometry of the surface. We notice that the Hausdorff distance between two consecutive level-sets becomes closer to their geodesic distance as we proceed with the dichotomic sub-division of the image of the input scalar function.

Meshless approximation. We define the approximation $g := u|_{\mathcal{M}} : \mathcal{M} \rightarrow \mathbb{R}$ of the input scalar function as the piecewise linear function that interpolates the values of u



(a) Coverage value

(b) Distortion map

(c)

(d)

Figure 6: (a) Coverage values (y-axis) of a noisy input scalar function $f:\mathcal{M}\rightarrow[0,1]$ over the surface $\mathcal{M}$ (28 minima, 22 maxima, 54 saddles) with respect to different samplings (x-axis) of the image $[0,1]$ of f and (b) distortion map (y-axis) with respect to the optimal partitioning of the image of f (x-axis). Since the global coverage converges to 1, the level-sets are distributed on the whole surface and spikes of the distribution map locates irregular distribution of the level-sets (due to noise). (c) Level-sets and (d) Morse-Smale complex of the meshless approximations of f with interpolating constraint at a different set of persistent critical points (12 minima, 10 maxima, 26 saddles) and least-squares constraints at the level-sets. The $\mathcal{L}^{\infty}$ approximation error is lower than 2.3×10^{-4}. The colours begin with red, pass through yellow, green, cyan, blue, and magenta, and return to red. See also Fig. 7.

at the vertices of $\mathcal{M}$. Here, the function $u:\mathbb{R}^3\rightarrow\mathbb{R}$, which is a linear combination $u(\mathbf{p}):=\sum_{i\in\mathcal{I}}a_i\phi_i(\mathbf{p})$ of the RBFs $\phi_i(\mathbf{p}):=\phi(\|\mathbf{p}-\mathbf{c}_i\|_2)$, generated by the globally-supported kernel $\phi:\mathbb{R}^+\rightarrow\mathbb{R}$ and centered $\mathcal{C}:=\{\mathbf{c}_i\}_{i\in\mathcal{I}}$ [6]. To define $\mathcal{C}$, we consider the behavior of the input scalar function, as encoded by the coverage value and the distortion map.

Firstly, we select $\mathcal{C}_{int}:=\mathcal{C}_{cp}\cup\mathcal{C}_{iso}$, where $\mathcal{C}_{cp}$ is the set of persistent critical points [11, 22, 23], together with their 1-star vertices, and $\mathcal{C}_{iso}$ is a sub-sampling (e.g., 1% of initial resolution, in our experiments) of the level-sets associated with the optimal sampling

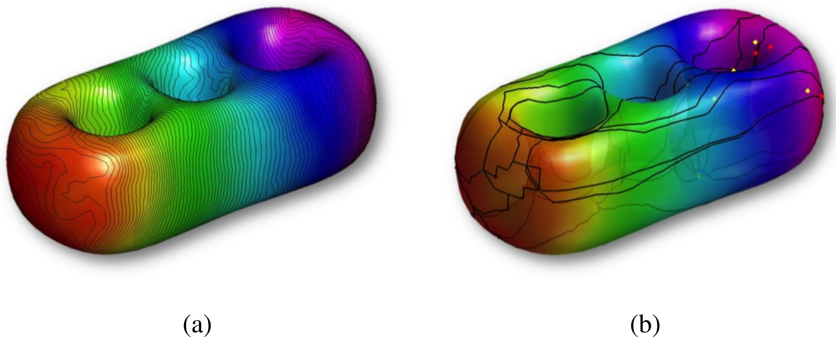


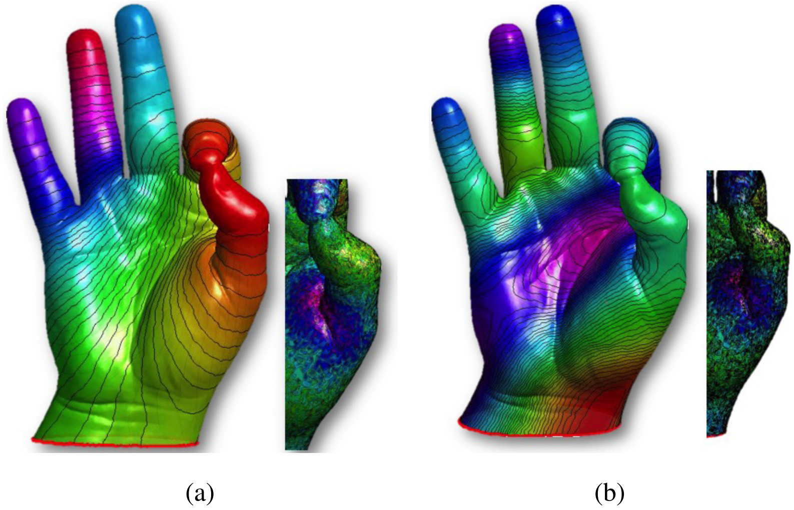
Figure 7: With reference to Fig. 6, (a) level-sets and (b) Morse-Smale complex of the meshless approximations of f with interpolating constraint at a different set of persistent critical points (6 minima, 8 maxima, 18 saddles) and least-squares constraints at the level-sets. The $\mathcal{L}^{\infty}$ approximation error is lower than 4.2×10^{-4}.

of the coverage value. Then, we consider $\mathcal{C}_{int}$ as centres and $\{f(\mathbf{p})\}_{\mathbf{p}\in\mathcal{C}}$ as interpolating constraints. If the global coverage value is close to zero (e.g., in $[0, 0.25)$, in our experiments), then the level-sets are uniformly distributed on the input surface and we do not select additional constraints. Otherwise, we identify the consecutive iso-values α, β such that the values $d(\alpha)$, $d(\beta)$ of the distortion map have an abrupt variation. The set $\mathcal{C}^{\star}_{iso}$ of the points of the surface strip $\mathcal{S} := f^{-1}([\alpha, \beta])$ that is not covered by the level-sets are clustered, thus providing the centres $\mathcal{C}_{ls}$ and the corresponding f-values are used as least-squares constraints. For the computation of the final approximation, we combine the interpolating and least-squares conditions associated with the centres $\mathcal{C}_{int} := \mathcal{C}_{cp} \cup \mathcal{C}_{iso}$ and $\mathcal{C}_{ls}$, respectively (Fig. 6, Fig. 7).

The coefficients $\mathbf{a} := (a_i)_{i=1}^{k}$ solve the linear system

$$\Phi\mathbf{a} = \mathbf{f} \Longleftrightarrow \begin{bmatrix} \Phi_{int} \\ \Phi_{ls} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \tag{4}$$

where k is the number of points in $\mathcal{C}$, $\Phi_{int} := (\phi_i(\mathbf{c}_j))_{i,j}$, $\mathbf{c}_i \in \mathcal{C}$, $\mathbf{c}_j \in \mathcal{C}_{int}$ (*int* stands for interpolation sub-matrix), is the Gram matrix associated with the set of interpolating constraints, $\Phi_{ls} := (\phi_i(\mathbf{c}_j))_{i,j}$, $\mathbf{c}_i \in \mathcal{C}$, $\mathbf{c}_j \in \mathcal{C}_{ls}$ (*ls* stands for least-squares sub-matrix), is the Gram matrix associated with the set of least-squares constraints, and $\mathbf{f}_1$, $\mathbf{f}_2$ are the f-values at the points in $\mathcal{C}_{int}$ and $\mathcal{C}_{ls}$, respectively. Among the possible kernels [21,



(a) (b)

Figure 8: (Right) Level-sets of two noisy scalar functions (a,b, right) with degenerate critical points: (a) ($m=123$, $M=98$, $s=221$) and (b) ($m=149$, $M=121$, $s=270$). Approximation with (a, left) ($m=12$, $M=10$, $s=22$, $\varepsilon_\infty=2.3\times10^{-4}$) and (b, left) ($m=15$, $M=13$, $s=28$, $\varepsilon_\infty=4.9\times10^{-5}$) persistent critical points with least-squares constraints at the level-sets.

37, 64], for our tests we have selected the Gaussian kernel $\phi(r):=\exp(-\sigma r^2)$ with support σ, which is selected according to the local behaviour of $\mathcal{M}$ [19, 38].

Assuming that the f-values have been increasingly sorted in $\mathcal{O}(n\log n)$-time, the samples of the s level-sets associated with the uniform optimal partitioning are computed in $\mathcal{O}(s\log n)$-time. Selecting k sub-samples, the computation of u requires to solve a $k\times k$ linear system with full coefficient matrix in $\mathcal{O}(k^3)$ time. Finally, the classification of the critical points is linear in the number of selected centres (Table 1).

Limitations and generalisation to d-dimensional data. The adaptive approximation of scalar functions (Sect. 4) is generally more time-consuming than those methods (e.g., [9, 14, 23, 61]) that simplify and locally update the input scalar function; in fact, we need to firstly identity the critical points to be preserved and then apply the meshless approximation. However, we are able to handle non-general functions without perturbing their values, to achieve smoother level-sets by imposing least-squares constraints on the f-values, and to support a meshless classification of the critical points (Sect. 5)

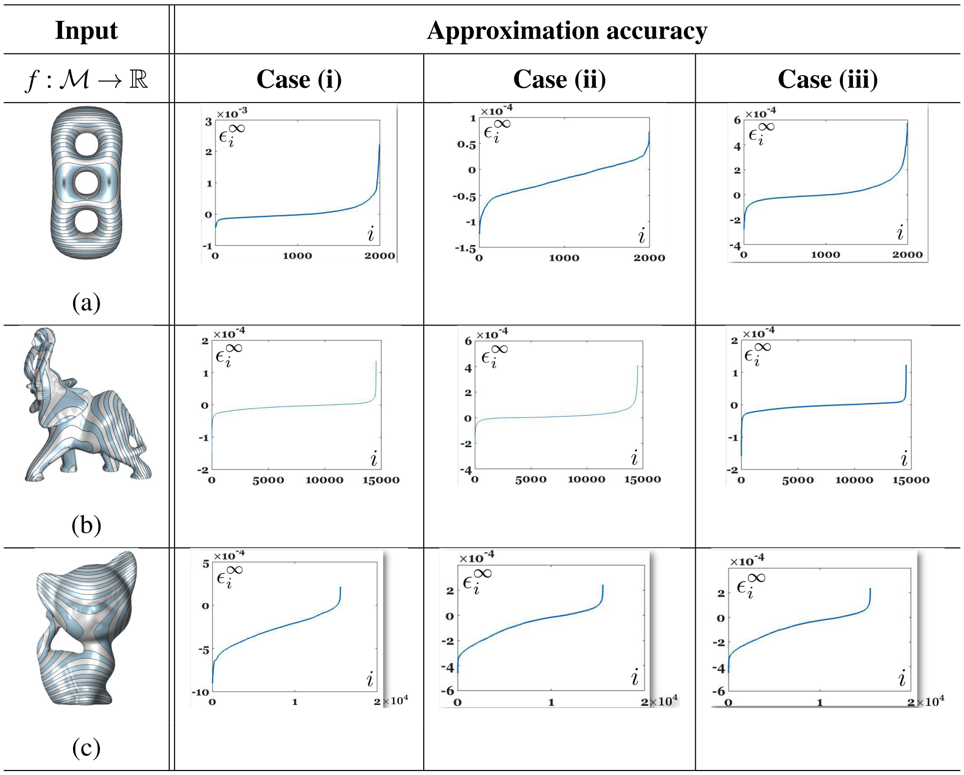


Figure 9: Level-sets and point-wise variation ε_i^∞ (y-axis) of the differences between the input and approximated functions at the vertices (x-axis) of $\mathcal{M}$, $\varepsilon_i^\infty := |f(\mathbf{p}_i) - u(\mathbf{p}_i)|$ is the approximation error at $\mathbf{p}_i$ between the input scalar function and its approximation u. For the approximation, we have imposed interpolating constraints at all the critical points and (i) at the level-sets, or (ii) least-squares constraints at the level-sets, or (iii) at the regular vertices not belonging to the 1-stars of critical points. Statistics are reported in Table 2.

and a meshless iso-contouring (Sect. 6.1). The proposed approach can be applied to scalar functions defined on volumes (e.g., tetrahedral, voxel grids) and more generally to d-dimensional data. In this case, the intersection between the level-set and a given voxel or tetrahedron is computed analogously to the surface case and by linearly interpolating the function values at the mesh vertices. For more details, we refer the reader to the look-up table used by the Marching Cubes algorithm [36]. Then, the coverage value is generalised by considering the percentage of volume contained by the selected level-sets. As done for the surface case, the distribution map is defined as the Hausdorff distance between consecutive level-sets. For the classification of the critical points of

Table 1: Timings (in sec.) for the computation of the meshless approximation, where n is the number of samples.

Test	n	**Time**	**Test**	n	**Time**	**Test**	n	**Time**
Fig. 5(c)	$120K$	5.34	Fig. 9(c)	$60K$	2.12	Fig. 11(a)	$3K$	0.21
Fig. 5(d)	$500K$	17.21	Fig. 14	$2.8M$	145.35	Fig. 11(b)	$4K$	0.36
Fig. 8(a)	$80K$	3.17	Fig. 16	$3K$	0.12	Fig. 11(c)	$2K$	0.12
Fig. 8(b)	$80K$	2.98	Fig. 16	$9K$	0.25	Fig. 11(d)	$21K$	0.64
Fig. 9(a)	$3K$	0.08	Fig. 16	$27K$	0.75	Fig. 10	$120K$	5.87
Fig. 9(b)	$15K$	0.57	Fig. 16	$71K$	2.67	Fig. 17	210	12.01

a scalar function defined on a volumetric domain, we apply previous work [60], which reflects the classification presented in Sect. 3.1. Finally, the approximation with RBFs remains unchanged, as a matter of its meshless and dimension-independent definition.

Examples and discussion. In Fig. 8, the selection of interpolating constraints on persistent critical points and least-squares constraints on the iso-level provides a high approximation accuracy, which is measured as the $\mathcal{L}^\infty$ norm between the approximated and ground-truth functions. Applying interpolating constraints on the f-values at all the critical points and interpolating or least-squares constraints at the level-sets (Fig. 9), the corresponding difference between the input and the approximated scalar functions is between 10^{-3} and 10^{-4}. The highest approximation accuracy is achieved by applying interpolating conditions at the critical points and at the level-sets. Introducing a Gaussian noise (10%) and applying the proposed approximation (Table 2), the selection of interpolating conditions at persistent critical points and least-squares constraints at the level-sets provides an accuracy higher than applying least-squares constraints at the f-values.

If the critical points were simplified by preserving the mutual relation among extrema and saddles in the Euler formula, then the interpolating constraints and the smoothness of the kernel generally guarantee that additional/spurious critical points will not be added. If the input scalar function f is not general (i.e., it is constant along edges), then the smoothness of the kernel guarantees that the number of the additional critical points is minimal to restore the validity of the Euler formula. An analogous

Table 2: Statistics on the critical points (m minima, M maxina, s saddles) and $\mathcal{L}^{\infty}$ approximation error (ε_{∞}) with constraints at the f-values ($\mathcal{M}$), at the critical points ($\mathcal{C}_{cp}$), and at the level-sets ($\mathcal{C}_{iso}$) for a noisy version of the scalar function in Fig. 9.

Approx.	**Critical points**			**Approx. error, ε_{∞}**		
	m	M	s	$\mathcal{M}$	$\mathcal{C}_{cp}$	$\mathcal{C}_{iso}$
Input	**2**	**4**	**9**	**Fig. 9(a)**		
Case (i)	2	4	10	0.000026	0.000000	0.000000
Case (ii)	2	4	10	0.000073	0.000043	0.000040
Case (iii)	2	4	10	0.000576	0.000331	0.000331
Input	**38**	**23**	**65**	**Fig. 9(b)**		
Case (i)	38	23	65	0.000036	0.000000	0.000000
Case (ii)	38	23	65	0.000124	0.000029	0.000022
Case (iii)	38	23	65	0.000136	0.000331	0.000021
Input	**10**	**16**	**26**	**Fig. 9(c)**		
Case (i)	10	16	26	0.000016	0.000000	0.000000
Case (ii)	10	16	26	0.000144	0.000153	0.000075
Case (iii)	10	16	26	0.000239	0.000120	0.000080

discussion applies in case of degenerate critical points or critical points that have been simplified without preserving the Euler formula. Indeed, we avoid the local perturbation of the f-values to remove degenerate cases, as done by previous work. For instance, the scalar function in Fig. 9(a) is not general but the meshless approximation is general and its additional saddle guarantees that the Morse-Euler formula is satisfied.

5. Meshless classification and computation of critical points

Starting from the meshless design (Sect. 3.2) and approximation of scalar functions (Sect. 4) previously introduced, we discuss the classification of their critical points, by combining the differential properties of the meshless function $u:\mathbb{R}^3 \rightarrow \mathbb{R}$ with the geometric properties of the underlying surface, as encoded by the first and second fundamental form. More precisely, let $g := u|_{\mathcal{M}}$ be the restriction of u to $\mathcal{M}$ and let us consider a parameterisation $\mathbf{r}(s,t) := (x_1(s,t), x_2(s,t), x_3(s,t))$ of $\mathcal{M}$ at $\mathbf{p} = \mathbf{r}(s_0,t_0)$,

$r : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$, $r \in \mathcal{C}^2(\Omega)$. Then, $g(s,t) := u(\mathbf{r}(s,t))$ and $\mathbf{p}$ is a critical point of g if and only if $\nabla g(s_0,t_0) = \mathbf{0}$. Applying the derivative of composite functions; i.e.,

$$\partial_s g = \partial_s u(\mathbf{r}(s,t)) = \partial_s u((x_i(s,t))_{i=1}^3) = \sum_{i=1}^3 \partial_{x_i} u \partial_s x_i = \langle \nabla u, \partial_s \mathbf{r} \rangle_2,$$

and analogously for $\partial_u g$, we express ∇g in terms of ∇u as

$$\nabla g(s,t) = \langle \partial_s g, \partial_t g \rangle_2 = [\langle \nabla u, \partial_s \mathbf{r} \rangle_2, \langle \nabla u, \partial_t \mathbf{r} \rangle_2].$$

From the following identities

$$\nabla g(s_0,t_0) = \nabla u(\mathbf{p}) \wedge \nabla \mathbf{r}(s_0,t_0), \qquad \mathbf{n}(\mathbf{p}) = \partial_s \mathbf{r}(s_0,t_0) \wedge \partial_t \mathbf{r}(s_0,t_0),$$

we have that $\mathbf{p}$ is a critical point if and only if $\nabla u(\mathbf{p})$ is parallel to $\mathbf{n}(\mathbf{p})$.

For the computation of the entries of the Hessian matrix $\mathbf{H}_u$, we differentiate $\partial_s g$ with respect to s and t, thus obtaining the following identities

$$\begin{aligned}
\partial_{ss}^2 g &= \sum_{i=1}^3 \partial_s(\partial_{x_i} g \partial_s x_i) \\
&= \sum_{i,j=1}^3 \partial_{x_i x_j}^2 u \partial_s x_i \partial_s x_j + \sum_{i=1}^3 \partial_{x_i} u \partial_{ss}^2 x_i \\
&= \partial_s \mathbf{r}^\top \mathbf{H}_u \partial_s \mathbf{r} + \langle \nabla u, \partial_{ss}^2 \mathbf{r} \rangle_2,
\end{aligned}$$

where $\partial_s \mathbf{r}$ is a 3×1 vector and $\mathbf{H}_u$ is a 3×3 matrix. For the computation of $\partial_{st} g$ and $\partial_{tt} g$, we proceed in an analogous way; indeed, the Hessian matrix is

$$\begin{aligned}
\mathbf{H}_g &= [\partial_s \mathbf{r}, \partial_t \mathbf{r}]^\top \mathbf{H}_u [\partial_s \mathbf{r}, \partial_t \mathbf{r}] + \begin{bmatrix} \langle \nabla u, \partial_{ss}^2 \mathbf{r} \rangle_2 & \langle \nabla u, \partial_{st}^2 \mathbf{r} \rangle_2 \\ \langle \nabla u, \partial_{st}^2 \mathbf{r} \rangle_2 & \langle \nabla u, \partial_{tt}^2 \mathbf{r} \rangle_2 \end{bmatrix}, \\
[\partial_s \mathbf{r}, \partial_t \mathbf{r}] &= \begin{bmatrix} \partial_s x_1 & \partial_t x_1 \\ \partial_s x_2 & \partial_t x_2 \\ \partial_s x_3 & \partial_t x_3 \end{bmatrix}
\end{aligned} \tag{5}$$

and its entries are evaluated at $\mathbf{r}(s,t)$. For the computation of the first and second order derivatives of the parameterisation $\mathbf{r}(s,t)$, we consider a local polynomial approximation of the surface as discussed in [31]. If $\mathbf{p}$ is a critical point of u (i.e., $\nabla u(\mathbf{p}) = \mathbf{0}$), then the matrix in the second part of the previous relation vanishes. Assuming that $\mathbf{p}$

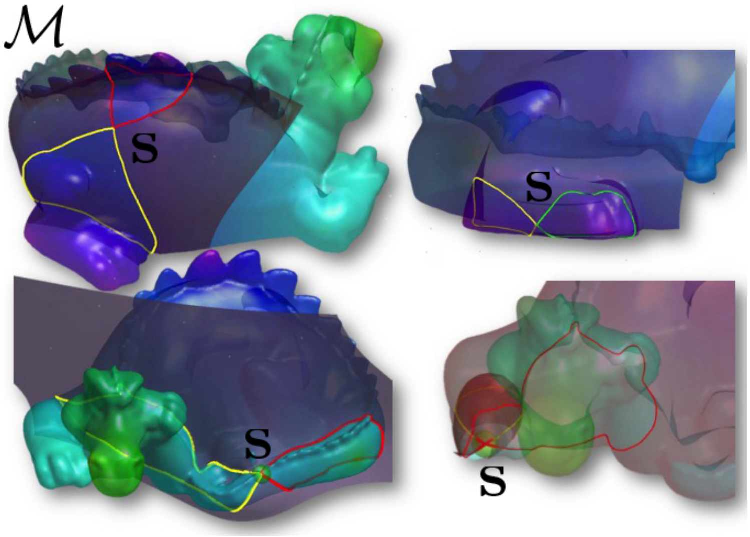


Figure 10: Computation of the saddle $\mathbf{s}$ and level-set $\Sigma_{\mathbf{s}} \cap \mathcal{M}$ at $\mathbf{s}$ achieved by intersecting the (red) iso-surface $\Sigma_{\mathbf{s}} := u^{-1}(f(\mathbf{s}))$ with the input surface $\mathcal{M}$.

is a critical point of g only, we have that $\nabla u(\mathbf{p})$ is parallel to the normal $\mathbf{n}(\mathbf{p})$ to the surface at $\mathbf{p}$ and it can be written as $\nabla u(\mathbf{p}) = \alpha(\mathbf{p})\mathbf{n}(\mathbf{p})$. Then, Eq. (5) is expressed in terms of the second fundamental form $\mathcal{F}_2 = L\mathrm{d}^2 s + 2M\mathrm{d}s\mathrm{d}t + N\mathrm{d}^2 t$ as

$$\mathbf{H}_g = [\partial_s \mathbf{r}, \partial_t \mathbf{r}]^\top \mathbf{H}_u [\partial_s \mathbf{r}, \partial_v \mathbf{r}] + \alpha(\mathbf{p}) \begin{bmatrix} L & M \\ M & N \end{bmatrix},$$

where the terms of the second fundamental form are $L := \langle \mathbf{n}(\mathbf{p}), \partial^2_{ss}\mathbf{r}\rangle_2$, $M := \langle \mathbf{n}(\mathbf{p}), \partial^2_{st}\mathbf{r}\rangle_2$, $N := \langle \mathbf{n}(\mathbf{p}), \partial^2_{tt}\mathbf{r}\rangle_2$.

Computation of the critical points. Assuming a unitary gradient and normal,

$$\pi(\mathbf{p}) := |\langle \nabla u(\mathbf{p}), \mathbf{n}(\mathbf{p})\rangle_2| = |\cos \omega_{\mathbf{p}}|$$

is the probability that $\mathbf{p}$ is critical. Evaluating $\pi(\cdot)$ at the input points, $\mathbf{p}^\star \in \mathcal{M}$ is classified as a *candidate critical point* if $\pi(\mathbf{p}^\star) \approx 1$. Then, we refine this first classification by solving the system of non-linear equations $\langle \nabla u(\mathbf{q}), \mathbf{n}(\mathbf{q})\rangle_2 = \mathbf{0}$ and $h(\mathbf{q}) = 0$, where h is a local representation of the MLS surface [2, 33] underlying $\mathcal{M}$ at $\mathbf{p}^\star$. To this end, we apply an interval analysis [29, 65] and a trust-region solver [17], whose starting point

$\mathbf{p}_{j_1} := \mathbf{p}^{\star}$ of the iterative scheme is the candidate critical point. The search direction of the iterative solver is along the gradient $\nabla u(\mathbf{p}_{j_s})$ at the current approximation $\mathbf{p}_{j_s}$. If the sequence $(\mathbf{p}_{j_s})_s$, generated by iterative solver, converges then the initial guess $\mathbf{p}_{j_1}$ is replaced with the critical point $\mathbf{p} := \mathbf{p}_{j_r}$ such that $\|\nabla u(\mathbf{p}_{j_r})\|_2 \leq \delta$ (e.g., $\delta := 10^{-10}$, in our tests), with δ accuracy threshold. Otherwise, $\mathbf{p}_{j_1}$ is discarded. To ensure the convergence of the iterative scheme, we assume that u and ∇u are defined at all the points of $\mathcal{M}$; to this end, it is enough to consider the Gaussian $\phi(r) := \exp(-r^2)$ or a polynomial kernel. Fig. 10, Fig. 11 show the saddle, the corresponding level-set and iso-surface computed with the meshless classification. If the input domain is represented as a smooth surface instead of a triangle mesh, then we can apply geometric well-constrained [24] or univariate [8] solvers.

The meshless classification of the critical points disambiguates the behaviour of the input scalar function/potential, which is represented in terms of the gradient and Hessian of its meshless approximation, from the geometry of the input domain, which is represented in terms of its normal, first and second fundamental form. This result is also interesting for the design of scalar functions on a surface $\mathcal{M}$ defined as the restriction of an implicit function from $\mathbb{R}^3$ to $\mathcal{M}$ (Sects. 3.3, 4). For the meshless classification and computation of the critical points, we have locally represented the input surface with a MLS approximation, which can be replaced with any other smooth representation, such as Non-Uniform Rational B-Splines [51].

Meshless classification of the critical points. Fig. 12 and Fig. 13 show the computation of the critical points of a noisy function u and its stability with respect to noise; the convergence of the iterative scheme is not affected by a random choice of the initial guesses. Furthermore, these points have been placed far from the critical points of the function $g : \mathbb{R}^2 \rightarrow \mathbb{R}$, which has been sampled to generate u. Note how the critical points that belong to the upper and lower part of the boundary have been identified in spite of their partial neighbours. Statistics related to the iterative scheme used to solve the equation $\nabla u = \mathbf{0}$ are reported in Table 3. For all these examples, each critical point $\mathbf{p}$ of the approximating function u has been computed with an error lower than 10^{-5} and the order of convergence of $\nabla u(\mathbf{p})$ to zero is equal to or lower than

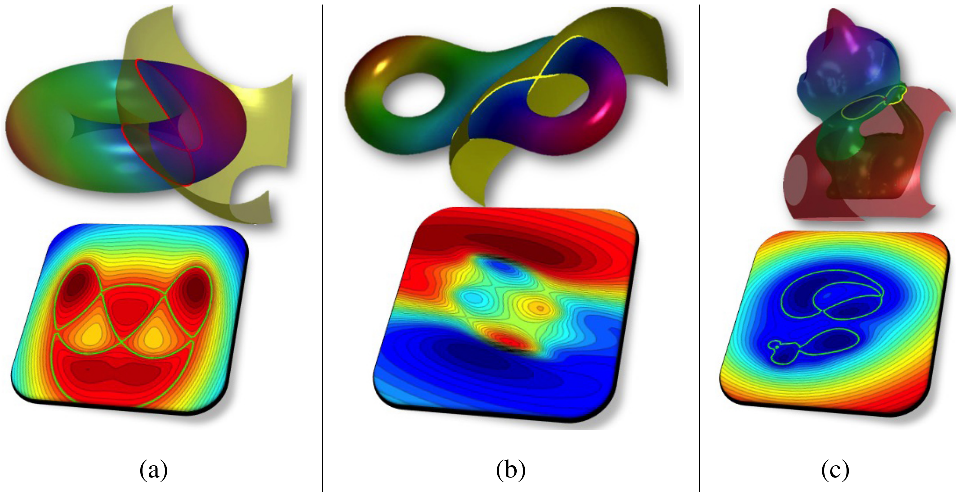


Figure 11: Meshless classification of saddles and iso-contouring of the input scalar functions. The behaviour of the approximation is also visualised on a cutting plane at saddle points and through the iso-surface of the meshless approximation at each saddle.

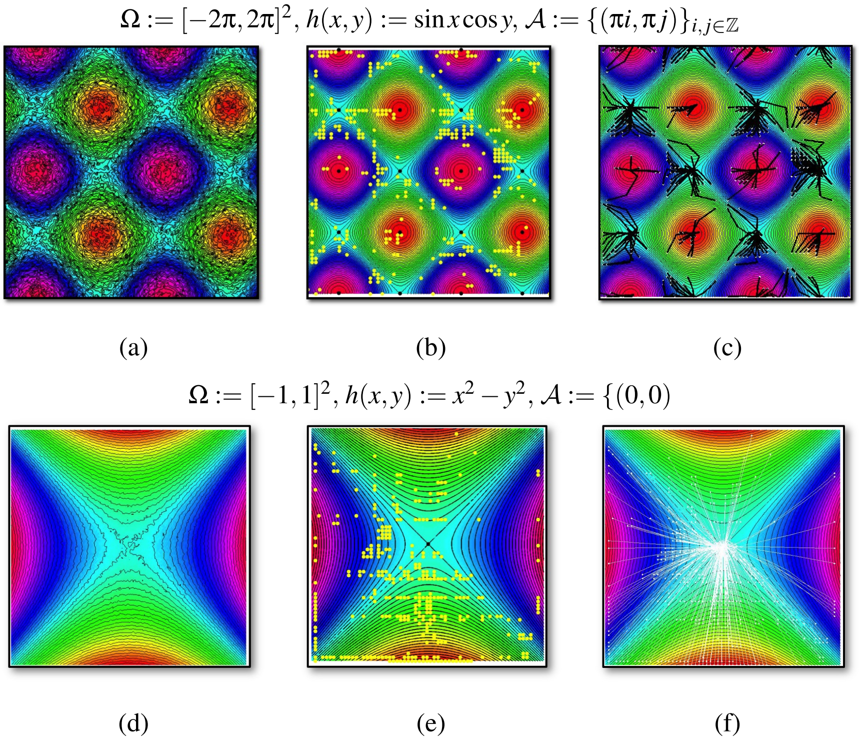
10^{-16}.

6. Applications

We now discuss further applications of the proposed approximation scheme to meshless iso-contouring (Sect. 6.1), space saving (Sect. 6.2), the approximation of rainfalls (Sect. 6.3), the generation of molecular surfaces (Sect. 6.4) and super-resolution images (Sect. 6.5).

6.1. Meshless iso-contouring

The meshless approximation of the input scalar function is useful to compute its level-set at an arbitrary resolution, through a local refinement of the mesh connectivity. To this end, let us consider a triangle t intersected by $\gamma_\alpha := u^{-1}(\alpha)$ and let split t into 4 sub-triangles by joining the mid-points of its edges. To re-sample γ_α in t, we compute the intersection between $u^{-1}(\alpha)$ and the edges of the four sub-triangles. Then, we evaluate u at the new vertices and identify the edges (i,j) intersected by



$\Omega := [-2\pi, 2\pi]^2$, $h(x,y) := \sin x \cos y$, $\mathcal{A} := \{(\pi i, \pi j)\}_{i,j\in\mathbb{Z}}$

(a) (b) (c)

$\Omega := [-1,1]^2$, $h(x,y) := x^2 - y^2$, $\mathcal{A} := \{(0,0)$

(d) (e) (f)

Figure 12: (a,d) Level-sets of a noisy sampling $f : \mathcal{P} \rightarrow \mathbb{R}$ of the function h on Ω, with $\mathcal{A}$ set of critical points. (b,e) Level-sets of its approximation $u : \mathbb{R}^2 \rightarrow \mathbb{R}$; the yellow dots have been randomly selected and used as starting guesses of the iterative scheme that computes the critical points (black dots) of u, which approximates $\mathcal{A}$ with an error lower than 10^{-5}. (c,f) Paths that join the initial guesses and the computed critical points. Statistics are reported in Table 3.

$u^{-1}(\alpha)$ by checking the *intersection condition* $u(\mathbf{p}_i) \leq \alpha$, $u(\mathbf{p}_j) \geq \alpha$. If it is satisfied, the intersection point between $u^{-1}(\alpha)$ and this edge is computed by (i) linearizing $u(\mathbf{p}) := \lambda u(\mathbf{p}_i) + (1-\lambda)u(\mathbf{p}_j)$ along the edge or (ii) computing the solution to the non-linear equation $u(\lambda\mathbf{p}_i + (1-\lambda)\mathbf{p}_j) = 0$, $\lambda \in [0,1]$, which is solved through a trust-region method [17], whose starting value is the solution $\lambda_0 := [\alpha - u(\mathbf{p}_i)]/[u(\mathbf{p}_j) - u(\mathbf{p}_i)]$ to the linearised equation (Fig. 14). Then, the triangle t is marked as visited and we proceed with the triangle t' adjacent to t and intersected by γ_α; i.e, there exists an edge of t' different from (i,j) and that satisfies the intersection condition.

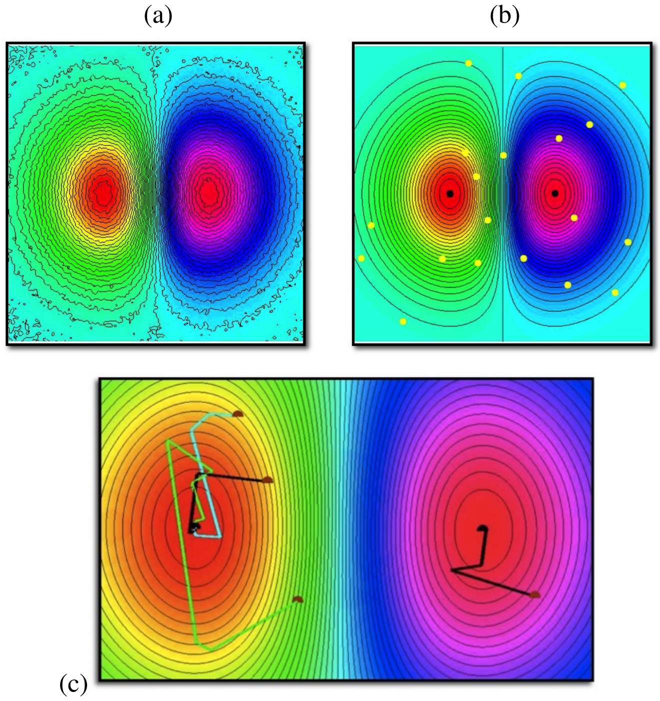
The refinement of the iso-contour stops when the area of the split triangle is lower

Table 3: Statistics for the computation of the critical points (black dots) of u in Fig. 12(c), ’*It*’ is the iteration step, ’*Ev*’ is the number of function evaluations of the of the iterative trust region technique [17], and $\|\nabla u(\mathbf{p})\|_2$ is the order of convergence of ∇u to zero.

It.	Ev.	$\|\nabla u(\mathbf{p})\|_2$	It.	Ev.	$\|\nabla u(\mathbf{p})\|_2$
0	4	0.018528	0	4	0.165408
1	8	0.007241	1	8	0.021796
2	12	0.000812	2	12	0.002461
3	16	$3.3485\,e-05$	3	16	0.000427
4	20	$9.6943\,e-08$	4	20	0.000128
5	24	$2.5456\,e-10$	5	24	$3.8085\,e-05$
6	28	$1.0703\,e-12$	⋮	⋮	⋮
7	32	$4.5536\,e-15$	25	104	$3.8681\,e-15$
8	36	$1.7921\,e-17$	26	108	$1.2274\,e-15$
9	40	$6.9600\,e-20$	27	112	$3.8881\,e-16$

It.	Ev.	$\|\nabla u(\mathbf{p})\|_2$	It.	Ev.	$\|\nabla u(\mathbf{p})\|_2$
0	4	0.0494952	0	4	0.172086
1	8	0.0400076	1	8	0.015144
2	12	0.0231691	2	12	0.002548
3	16	0.0005562	3	16	$5.0169\,e-05$
⋮	⋮	⋮	⋮	⋮	⋮
27	112	$3.53052\,e-16$	17	72	$2.9415\,e-16$
28	116	$1.1088\,e-16$	18	76	$4.7386\,e-17$

than a given threshold (e.g., 1% of the minimum triangle area) or the maximum distance between two consecutive refinements of an arc is not significantly reduced. Once the arc of γ_α has been approximated inside the triangle t and within the selected accuracy, the resampling of the level-set considers the adjacent triangles and the procedure stops when we come back to the first visited triangle and the whole level-set has been re-sampled. If there is a non-visited triangle whose edges satisfy the intersection condition, then we apply the previous procedure and extract the second connected component of the level-set. The meshless iso-contouring stops when all the triangles whose edges satisfy the intersection condition have been visited, thus guaranteeing that we



$$\Omega := [-2,2]^2, \quad g(x,y) := x\exp\left[-(x^2+y^2)\right], \quad \mathcal{A} := \{(\pm\sqrt{2}/2)\}$$

Figure 13: (a) Level-sets of a map achieved by adding a Gaussian noise to h in Ω, with $\mathcal{A}$ set of critical points. (b) Approximation and (c) critical points (black dots) with paths computed by the iterative scheme from random guesses (yellow dots).

have extracted all the connected components of the level-set.

For a regular level-set (i.e., not associated with a critical point), the previous procedure will draw all its connected components at a given resolution. For a non-degenerate saddle $\mathbf{p}_i$, we notice that the connected component β of $f^{-1}(\alpha)$ that contains $\mathbf{p}_i$ is the union of $m+1$ closed curves $\beta_1,\ldots,\beta_{m+1}$ that intersect at $\mathbf{p}_i$, i.e. $\beta := \cup_{l=1}^{m+1}\beta_l \ni \mathbf{p}_i$. As discussed in [50], the classification of a saddle point of multiplicity m (c.f., Sect. 3.1 and Eq. (1)) implies that the cardinality of the mixed link is equal to $2+2m$ and provides the directions that originate at $\mathbf{p}_i$ and that are used to trace all the loops β_l at $\mathbf{p}_i$. Finally, for dense surfaces the iso-contouring is applied to the map u that approximates the f-values only at the vertices $\{\mathbf{p}_i\}_{i\in\mathcal{I}}$ of the edges intersected by γ_α (Fig. 15, Fig. 16).

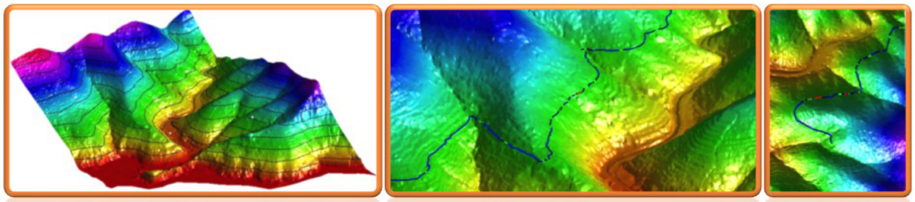


Figure 14: Local iso-contouring of a terrain model with 2.8M points and zoom on the extracted level-sets.

Table 4: Space saving achieved by the meshless approximation and comparison among the original mesh size, the size of the meshless approximation, and the corresponding error.

Test	n	$\mathcal{L}^\infty$ **approx. accuracy**	**Num. centres**	**Space saving**
Fig. 10	150K	10^{-4}	1251	$99,99\%$
Fig. 11(a)	8K	10^{-5}	934	$99,88\%$
Fig. 11(b)	9K	10^{-5}	589	$99,93\%$
Fig. 11(c)	12K	10^{-6}	446	$99,96\%$

6.2. Space saving

Working only with the discrete f-values, the classification and simplification of the critical points depend on the discretisation and samples of the input surface, thus being affected by local perturbations of the surface connectivity/sampling and of the f-values. While a discrete function on different resolutions of the same surface is represented by an array whose dimension depends on the number of vertices, the meshless approximation is represented as a set of coefficients and centres, and depends only on the behaviour of the input scalar function. Decoupling the representation of the function from the discretisation of the input domain is important for space saving (Table 4) and to distinguish the complexity of the geometry of $\mathcal{M}$ from the complexity of the input scalar function.

6.3. Approximation and analysis of rainfalls

As GIS application, we consider the analysis of terrains and rainfalls. Interpreting rainfall measurements on a terrain $\mathcal{M}$ as a scalar function $f : \mathcal{P} \to \mathbb{R}$ at the rain

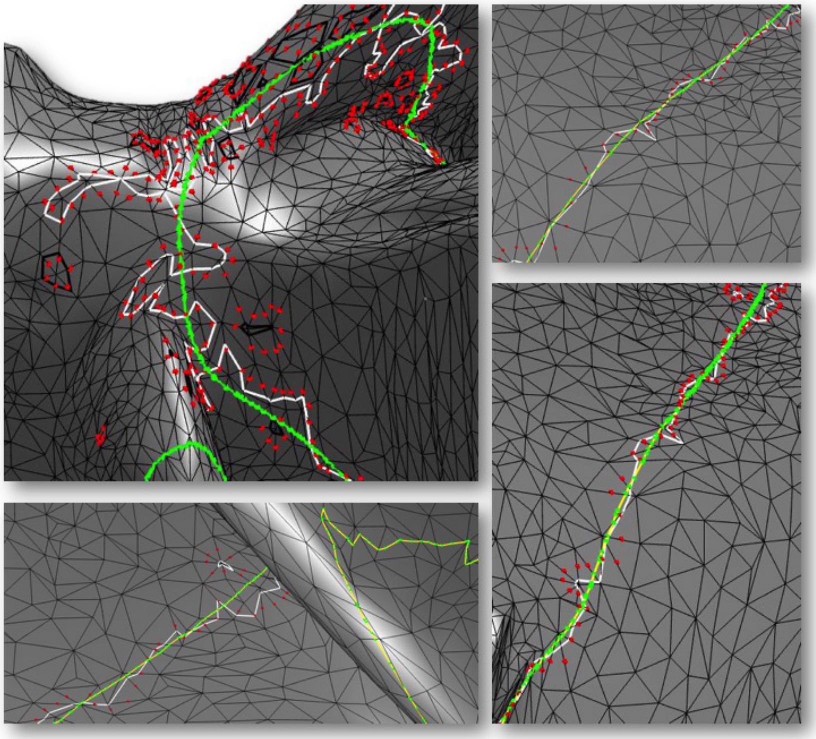


Figure 15: Noisy level-sets (white level-sets) and their meshless approximation (green line) with a local split of the triangles, according to the local approximation error and the reduction of the triangle area.

gauge network $\mathcal{P} := \{\mathbf{p}_i\}_{i=1}^{n}$, we compute an approximation $g := u|_{\mathcal{M}} : \mathcal{M} \rightarrow \mathbb{R}$ that preserves its maxima. In fact, preserving rainfall maxima is crucial to support real-time monitoring and historical data analysis. Sampling u on a terrain model with a resolution higher than the initial sparse network of rainfall station allows us to extrapolate and analyse the rainfall behaviour at different scales and with a higher accuracy. The level-sets close to maxima/minima also locate dangerous/safe regions and saddles identify where main changes of the rainfall happen. Furthermore, the approximation of time-varying rainfalls is easily updated; in fact, the coefficient matrix and the 1-star of each vertex are computed once and only the right-hand side of the linear system and the critical points (c.f., Eq. (4)) are updated at each time step.

For our analysis (Fig. 17, Table 5), we consider the SRTM (Shuttle Radar Topog-

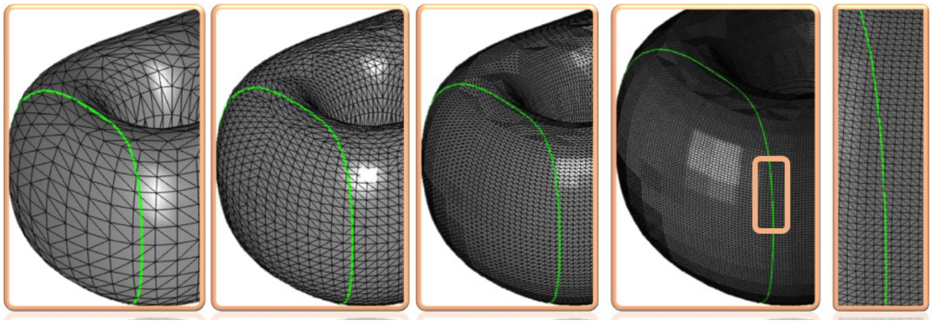


Figure 16: Meshless approximation of a level-set on different resolutions of the same surface. The maximum variation of the resampled level-set with respect to the ground-truth is lower than 0.2%.

Table 5: $\mathcal{L}^{\infty}$ accuracy (%) between the ground-truth rainfalls and three approximations. For the meshless approximation with RBFs, we have selected the Gaussian kernel. Approximation results with RBFs are shown in Fig. 17.

Approximation scheme	$\mathcal{L}^{\infty}$ **Approximation error**							
Piecewise Linear approx.	0.31	0.23	0.27	0.38	0.45	0.57	0.25	1.2
Inverse weight distance	0.21	0.17	0.22	0.32	0.22	0.17	0.11	0.37
Radial Basis Funct. (Fig. 17)	0.12	0.09	0.10	0.11	0.21	0.17	0.07	0.32

raphy Mission [25]) digital terrain model with 100K vertices and a set of time-varying rainfall data measured at 148 rainfall stations. During the whole temporal interval (48 hours), two local thunderstorms occurred and only localized weather stations recorded this heavy event (samples every 30 minutes). We compare our approach with previous work; i.e., the piecewise linear approximation based on Voronoi regions [58] and the meshless approximation based on weighted inverse distances [57], with no constraints on the preservation of their maxima [46, 49]. To this end, each rain gauge has been turned off and the approximating function has been sampled at this position. Comparing the input value at the turned-off station with the approximated value at the same location, we have evaluated the extrapolation capability as the resulting $\mathcal{L}^{\infty}$ error. These three techniques provide analogous results but the proposed approach shows a lower approximation error, which is mainly due to the presentation of local extrema

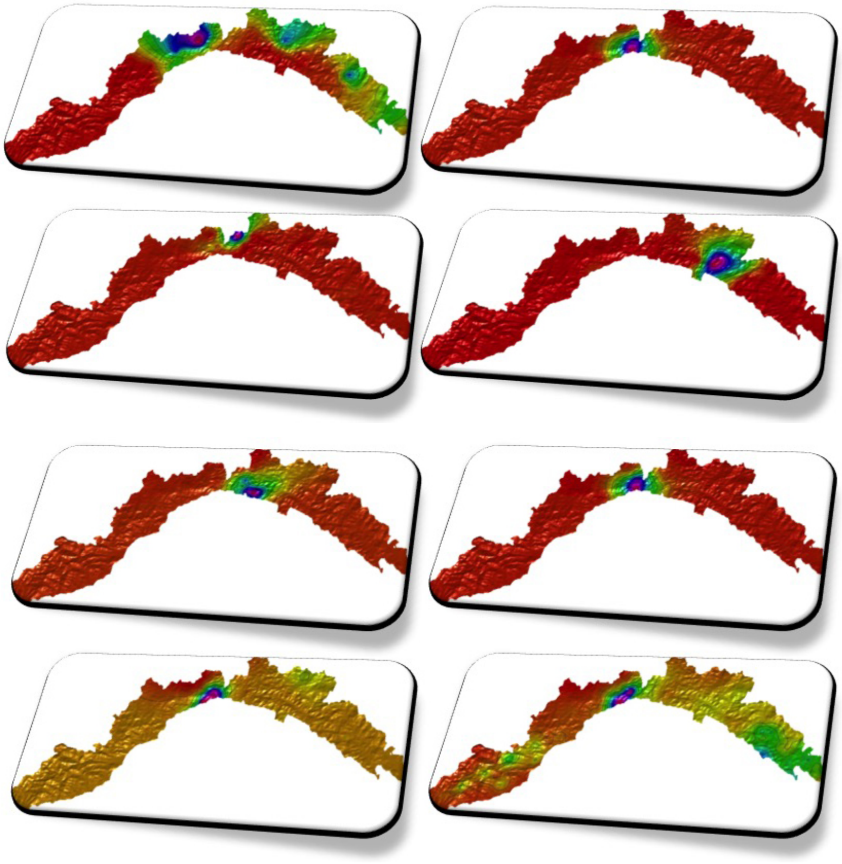


Figure 17: Meshless approximation of rainfalls with preserved maxima: heavy rains are localised in pink areas and drizzles in the remaining green parts. Statistics are reported in Table 5.

and to the capability of the radial basis functions to adapt to the rainfalls' behaviour.

6.4. Molecular surfaces and simulation

As a bioinformatic application, we discuss the re-sampling of the electron density map computed from the centers and radii of a set of atoms, as solution of the Poisson-Boltzmann equation [26] on a low-resolution grid. Since the iterative solver is time-consuming and takes $\mathcal{O}(kn^3)$-time, where n is the number of voxels and k is the number of atoms, the resampling at a higher resolution allows us to improve the quality of the molecular surface. Furthermore, the interpolation of the maxima and minima guarantees that the behavior of the charge is correctly extrapolated, also in

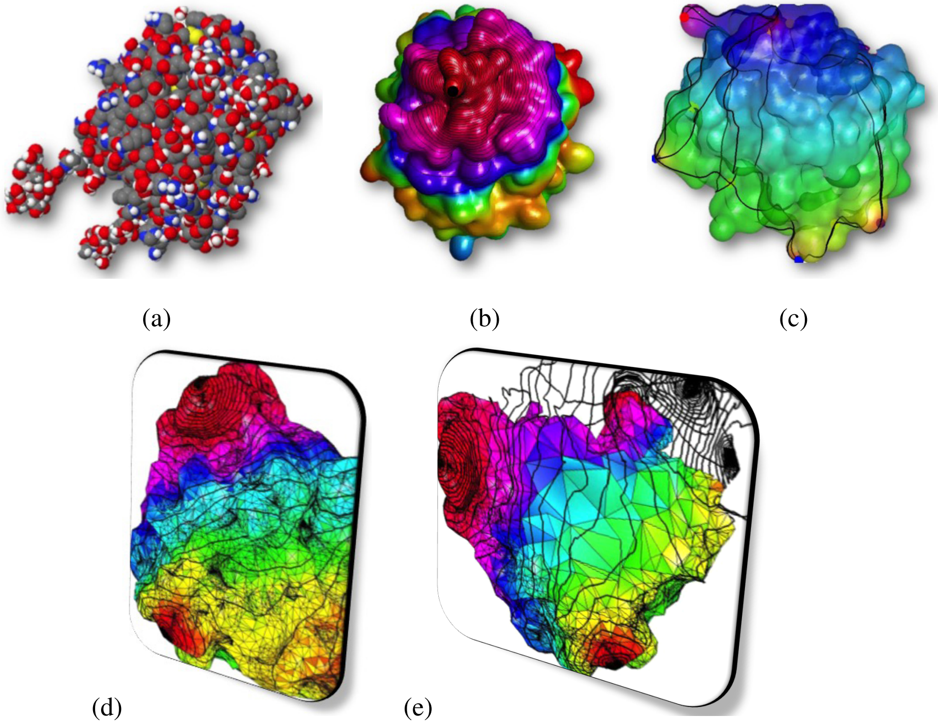


Figure 18: (a) Input atoms and radii, (b) corresponding molecular surface associated with the scalar function $f : \mathcal{M} \to \mathbb{R}$, which represents a local property (i.e., hydrophobicity) of the surface. (c) Morse complex and critical points of f computed through the meshless approximation constrained to the maxima (red), minima (blue), and saddles (green) points of f. (d,e) Level-sets of f and colour-coding of the volumetric approximation at the nodes of a tetrahedral mesh.

terms of preserved local details and small cavities, which are particularly useful for molecular docking (Fig. 18). For details on the application of meshless approximation to the representation and analysis of molecular surfaces, we refer the reader to [48].

6.5. Super-resolution images

For sampling an image at a different resolution, we compute an interpolation of the image pixels, which is then evaluated at the new pixels. In this case, the critical points represent features of the input images, such as corners, intensities' extrema, etc. State-of-the-art methods include the nearest neighborhood [55], linear [16], and cubic [5] interpolations and wavelets [13], meshless approximations [40]. To compare

Table 6: 2× and 4× down-sampling followed by up-samplings; bold text identifies the best results with RBFs generated by the Hardy kernel with respect to previous work. The Gaussian kernel provides results analogous to the Hardy's kernel.

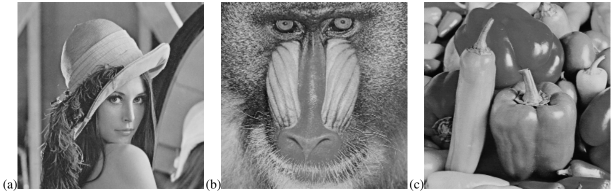


Fig.	Error	Linear $r=1$	Spline $r=3$	Spline $r=4$	Spline $r=5$	RBF Hardy	RBF Gauss
2× down-sampling and up-sampling							
(a)	MRE	284.34	277.52	277.64	278.12	**131.28**	136.43
	PSNR	23.59	23.70	23.70	23.69	**26.95**	26.78
	SSIM	0.80	0.81	0.81	0.81	**0.85**	0.83
(b)	MRE	454.40	448.72	450.54	451.98	**294.37**	308.73
	PSNR	21.56	21.61	21.59	21.58	**23.44**	23.24
	SSIM	0.59	0.62	0.62	0.62	**0.69**	0.62
(c)	MRE	284.00	276.51	276.49	276.69	**104.75**	107.89
	PSNR	23.60	23.71	23.71	23.71	**27.93**	27.80
	SSIM	0.84	0.85	0.85	0.85	**0.88**	0.87
4× down-sampling and up-sampling							
(a)	MRE	494.41	469.56	470.05	470.80	**201.38**	213.25
	PSNR	21.19	21.41	21.41	21.40	**25.09**	24.84
	SSIM	0.70	0.72	0.71	0.71	**0.74**	0.73
(b)	MRE	698.67	694.52	697.62	699.82	**430.78**	442.22
	PSNR	19.69	19.71	19.69	19.68	**21.79**	21.67
	SSIM	0.41	0.43	0.43	0.43	**0.45**	0.44
(c)	MRE	519.14	487.00	486.59	486.99	**180.45**	197.53
	PSNR	20.98	21.26	21.26	21.26	**25.57**	25.17
	SSIM	0.75	0.77	0.76	0.76	**0.79**	0.77

the accuracy of our approach with previous work, an input image (ground-truth) is initially down-sampled and then resampled back to the initial resolution. The two images are then compared with respect to the *mean square error* (MRE); the *peak signal-to-noise ratio* (PSNR) $PSNR_{AB} := 10\log_{10}(\lambda^2/MRE_{AB})$, and the *structural similarity index* (SSIM). The MRE takes values in $[0,\lambda^2]$, where λ is the maximum intensity value and lower values correspond to a higher similarity between images; PSNR returns values in $[10,+\infty)$, which increases as the compared images become similar; SSIM returns a value in $[-1,1]$, where 1 corresponds to the case of two identical images. With respect to these metrics and previous work, Table 6 shows a better performance for the meshless approximation with the Hardy's and Gauss kernels.

7. Conclusions and future work

This paper has tacked the design of scalar functions with constrained critical points and the computation of a smooth approximation of an arbitrary function by preserving its (persistent) critical points and the distribution and shape of the level-sets, and by minimising the approximation error. Both problems have been addressed by applying mesh-based or meshless approximation techniques with interpolating or least-squares constraints on the critical points and their 1-star neighbours.

Through the meshless approximation, which is mainly based on radial basis functions, we have addressed the computation of the level-sets and the classification of the critical points, which can be performed at an arbitrary resolution by applying differential calculus. In this way, our approach is oblivious of any discretisation of differential operators and any assumption on the sampling of the discrete scalar function, the domain connectivity and parameterisation. The underlying meshless approximation allows us to easily extend it to scalar function on $\mathbb{R}^d$.

Future work will be mainly focused on the extension of the proposed approach to the approximation and analysis of vectorial fields defined on surfaces and volumes; in fact, several results introduced for the meshless approximation and classification of the critical points of scalar functions can be extended to vectorial fields.

Acknowledgements. We thank the Reviewers for their constructive comments, which helped us to improve the paper presentation and content. Data sets are courtesy of the SRTM `http://www2.jpl.nasa.gov/srtm/`, the AIM@SHAPE and EG Repositories, Regione Liguria and the FP7 European Project IQmulus. Thanks are due to the members of the *Shape and Semantics Modelling Group* at CNR-IMATI for helpful discussions.

References

[1] A. Adamson and M. Alexa. Approximating and intersecting surfaces from points. In *ACM Symposium on Geometry Processing*, pages 230–239, 2003.

[2] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *IEEE Visualization*, pages 21–28, 2001.

[3] N. Amenta and Y. Kil. The domain of a point set surface. *Symposium on Point-Based Graphics*, pages 139–148, 2004.

[4] N. Amenta and Y. J. Kil. Defining point-set surfaces. In *ACM Siggraph*, pages 264–270, 2004.

[5] L. Andersson and T. Elfving. Interpolation and approximation by monotone cubic splines. *Journal of Approximation Theory*, 66(3):302–333, 1991.

[6] N. Aronszajn. Theory of reproducing kernels. *Trans. of the American Mathematical Society*, 68:337–404, 1950.

[7] T. Banchoff. Critical points and curvature for embedded polyhedra. *Journal of Differential Geometry*, 1:245–256, 1967.

[8] M. Barton, G. Elber, and I. Hanniel. Topologically guaranteed univariate solutions of underconstrained polynomial systems via no-loop and single-component tests. *Computer-Aided Design*, 43(8):1035 – 1044, 2011.

[9] U. Bauer, C. Lange, and M. Wardetzky. Optimal topological simplification of discrete functions on surfaces. *Discrete Computational Geometry*, 47(2):347–377, 2012.

[10] J. Bloomenthal and B. Wyvill, editors. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., 1997.

[11] P. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Trans. on Visualization and Computer Graphics*, 10(4):385–396, 2004.

[12] M. Bronstein and A. Bronstein. Shape recognition with spectral distances. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(5):1065 –1071, 2011.

[13] W. Carey, D. Chuang, and S. Hemami. Regularity-preserving image interpolation. *IEEE Trans. on Image Processing*, 8(9):1293–1297, 1999.

[14] H. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *IEEE Visualization*, pages 497–504, 2004.

[15] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *ACM Siggraph*, pages 67–76, 2001.

[16] M. Chen, C. Huang, and W. Lee. A fast edge-oriented algorithm for image interpolation. *Image and Vision Computing*, 23(9):791–798, 2005.

[17] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods*. Society for Industrial and Applied Mathematics, 2000.

[18] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *ACM Siggraph*, pages 317–324, 1999.

[19] T. K. Dey and J. Sun. An adaptive MLS surface for reconstruction with guarantees. In *ACM Symp. on Geometry Processing*, pages 43–52, 2005.

[20] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Spectral surface quadrangulation. *ACM Siggraph*, pages 1057–1066, 2006.

[21] N. Dyn, D. Levin, and S. Rippa. Numerical procedures for surface fitting of scattered data by radial functions. *SIAM Journal on Scientific and Statistical Computing*, 7(2):639–659, 1986.

[22] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Local and global comparison of continuous functions. In *IEEE Visualization*, pages 275–280, 2004.

[23] H. Edelsbrunner, D. Morozov, and V. Pascucci. Persistence-sensitive simplification functions on 2-manifolds. In *ACM Symp. on Computational Geometry*, pages 127–134, 2006.

[24] G. Elber and M.-S. Kim Geometric constraint solver using multivariate rational spline functions. In *Proc. of ACM Symposium on Solid Modeling and Applications*, pages 1–10, New York, NY, USA, 2001. ACM.

[25] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, and L. Roth. The Shuttle radar topography mission. *Reviews of Geophysics*, 45(2), 2007.

[26] F. Fogolari, A. Brigo, and H. Molinari. The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology. *Journal of Molecular Recognition*, 15(6):377–392, 2002.

[27] G. Golub and G. VanLoan. *Matrix Computations*. John Hopkins University Press, 2nd Edition, 1989.

[28] P. C. Hansen and D. P. O'Leary. The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM Journal of Scientific Computing*, 14(6):1487–1503, 1993.

[29] J. C. Hart, A. Durr, and D. Harsh. Critical points of polynomial metaballs. In *Proc. of Implicit Surfaces*, pages 69–76, 1998.

[30] J. Huang, M. Zhang, J. Ma, X. Liu, L. Kobbelt, and H. Bao. Spectral quadrangulation with orientation and alignment control. *ACM Trans. on Graphics*, 27(5):1–9, 2008.

[31] X. Jiao and H. Zha. Consistent computation of first- and second-order differential quantities for surface meshes. In *Proc. of ACM Symp. on Solid and Physical Modeling*, pages 159–170, 2008.

[32] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. In *Proc. of the National Academy of Sciences*, pages 8431–8435, 1998.

[33] D. Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67(224):1517–1531, 1998.

[34] D. Levin. Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization*, 2003.

[35] Y.-S. Liu, M. Liu, D. Kihara, and K. Ramani. Salient critical points for meshes. In *Proc. of Symp. on Solid and Physical Modeling*, pages 277–282, 2007.

[36] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph*, 21(4):163–169, 1987.

[37] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

[38] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Proc. of Symp. on Computational Geometry*, pages 322–328, 2003.

[39] B. S. Morse, T. S. Yoo, D. T. Chen, P. Rheingans, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *IEEE Shape Modeling and Applications*, pages 89–98, 2001.

[40] M. Natali, G. Tagliafico, and G. Patanè. Local up-sampling and morphological analysis of low-resolution mr images. *Neurocomputing*, To appear, 2017.

[41] X. Ni, M. Garland, and J. C. Hart. Fair Morse functions for extracting the topological structure of a surface mesh. In *ACM Siggraph*, pages 613–622, 2004.

[42] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Siggraph*, 22(3):463–470, 2003.

[43] Y. Ohtake, A. Belyaev, and H.-P. Seidel. 3D scattered data interpolation and approximation with multilevel compactly supported RBFs. *Graphical Models*, 67(3):150–165, 2005.

[44] V. Pascucci, G. Scorzelli, P. Bremer, and A. Mascarenhas. Robust on-line computation of Reeb graphs: simplicity and speed. *ACM Trans. on Graphics*, 26(3):58, 2007.

[45] G. Patanè. Accurate and efficient computation of laplacian spectral distances and kernels. *Computer Graphics Forum*, 36(1):184–196, 2017.

[46] G. Patanè, A. Cerri, V. Skytt, S. Pittaluga, S. Biasotti, D. Sobrero, T. Dokken, and M. Spagnuolo. Comparing methods for the approximation of rainfall fields in environmental applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2016.

[47] G. Patanè and B. Falcidieno. Computing smooth approximations of scalar functions with constraints. *Computer & Graphics*, 33(3):399 – 413, 2009.

[48] G. Patanè and M. Spagnuolo. *State-of-the-Art and Perspectives of Geometric and Implicit Modeling for Molecular Surfaces*, pages 157–176. Springer International Publishing, 2015.

[49] G. Patanè and M. Spagnuolo. *Heterogeneous Spatial Data: Fusion, Modeling, and Analysis for GIS Applications*. Synthesis Lectures on Visual Computing. Morgan & Claypool Publishers, 2016.

[50] G. Patanè, M. Spagnuolo, and B. Falcidieno. A minimal contouring approach to the computation of the reeb graph. *IEEE Trans. on Visualization and Computer Graphics*, 15(4):583–595, 2009.

[51] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 1997.

[52] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

[53] T. Poggio and F. Girosi. Networks for approximation and learning. *Proc. of the IEEE*, 78(9):1481–1497, 1990.

[54] M. Reuter, F.-E. Wolter, M. E. Shenton, and M. Niethammer. Laplace-beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. *Computer-Aided Design*, 41(10):739–755, 2009.

[55] S. S. Rifman. Digital rectification of ERTS multispectral imagery. *In Proc. of Symp. Significant Results Obtained from ERTS-I*, 1:1131–1142, 1973.

[56] G. Taubin. A signal processing approach to fair surface design. In *ACM Siggraph 1995*, pages 351–358, 1995.

[57] R. Teegavarapu and V. Chandramouli. Improved weighting methods, deterministic and stochastic data-driven models for estimation of missing precipitation records. *Journal of Hydrology*, 312(1):191–206, 2005.

[58] A. H. Thiessen. Precipitation averages for large areas. *Monthly weather review*, 39(7):1082–1089, 1911.

[59] J. Tierny, D. Günther, and V. Pascucci. *Optimal General Simplification of Scalar Fields on Surfaces*, pages 57–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[60] J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci. Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1177–1184, 2009.

[61] J. Tierny and V. Pascucci. Generalized topological simplification of scalar fields on surfaces. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2005–2013, 2012.

[62] G. Turk and J. F. O'Brien. Modelling with implicit surfaces that interpolate. *ACM Siggraph*, 21(4):855–873, 2002.

[63] G. Wahba. *Spline Models for Observational Data*, volume 59. SIAM, Philadelphia, 1990.

[64] H. Wendland. Real piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(4):389–396, 1995.

[65] S.-T. Wu and M. De Gomensoro. On improving the search for critical points of implicit functions. *Proc. Implicit Surfaces*, pages 73–80, 1999.

[66] H. Xie, K. T. McDonnell, and H. Qin. Surface reconstruction of noisy and defective data sets. In *IEEE Visualization*, pages 259–266, 2004.