

Remarks on the realization of the Reeb graph

ADELA LUPESCU
(CLUJ-NAPOCA)

ABSTRACT. The Reeb graph came into the attention of mathematicians as the result of studies regarding the level sets of a real valued function defined on a manifold. It was introduced by G.H Reeb [13] as a tool to investigate real Morse functions. Its importance gained weight as the technology progressed and today the Reeb graph is used in many fields like computer graphics, image processing and shape analysis. This paper offers a survey on different approaches on realization of the Reeb graph. We also present a new algorithm that emphasizes the construction of such a graph using pre-defined set of components and given rules for circle-valued functions defined on surfaces.

1 Introduction

The Reeb graph, represents a mathematical instrument, used for studying the evolution of level sets of a function with real values on a given manifold. Formally it is given as the quotient space $M/\sim$, where M is a topological space, $f : M \rightarrow \mathbb{R}$ a real valued function and " $\sim$ " the equivalence relation defined by $x \sim y$ if and only if x and y belong to the same connect component of $f^{-1}(a)$, $a \in \mathbb{R}$.

[◇]Adela Lupescu, Babeş-Bolyai University, Cluj-Napoca
email: adelalupescu@gmail.com

The Reeb graph category allows a visualization way of the evolution of connected components throughout the level sets using a real valued function. The objects of this category are finite graphs associated to functions strictly monotonic on the edges and morphisms that preserve maps between underlying spaces. More explicitly the morphism that preserves maps between the topological spaces (X, f) , (Y, g) is in fact a function φ for which the following diagram commutes:

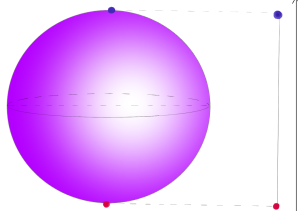
$$\begin{array}{ccc} X & \xrightarrow{\varphi} & Y \\ & \searrow f & \swarrow g \\ & \mathbb{R} & \end{array}$$

Using the terminology in [1], the vertices of the Reeb graph are the classes of critical points x , such that $x \in M_c^s$, where the set M_c^s denotes the set of essential components, meaning $M_c^s \cap C(f) \neq \emptyset$.

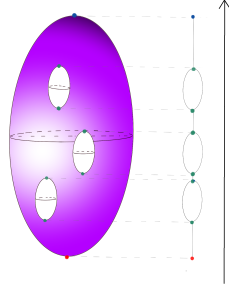
For a better understanding of the Reeb graph it is important to present some particular situations and examples. In the paper [9], the following particularization of the domain of the real valued function is considered:

- a) If M is the n -dimensional unit sphere S^n , ($n \geq 2$) or the n -dimensional real projective space $\mathbb{RP}^n$ or the n -dimensional complex projective space $\mathbb{CP}^n$, the associated Reeb graph is a tree;
- b) If M is an orientable surface of genus g then its Reeb graph contains at most $2g$ loops;
- c) If M is a non-orientable surface of genus g then its Reeb graph contains at most g loops.

Example 1.1 1. *The Reeb graph of the sphere given with f the height function:*

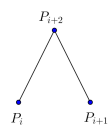
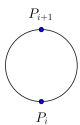
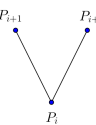


2. The Reeb graph of a tours of genus 3 with f the height function:

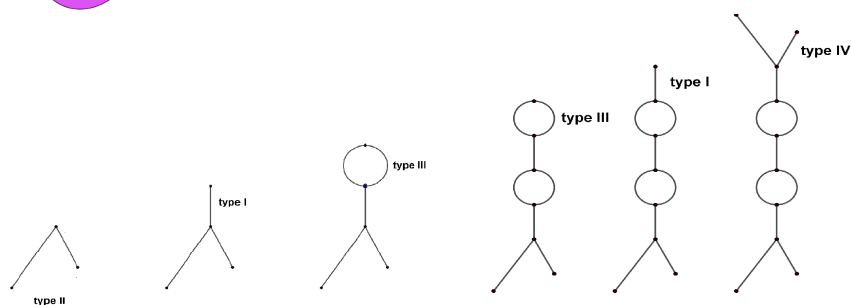
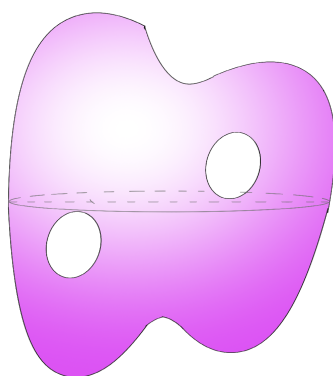


The need for understanding the evolution of level curves on a manifold led to a more rigorous classification of components for a graph and therefor it facilitated the apprehension of parameters required for realization of such a graph. The paper [11] presents an algorithm for constructing the Reeb graph using a scheme of predefined components. We will only present a short sketch of this algorithm and a new example based on the above scheme.

This algorithm identifies the critical points of the function, arranges the critical points in an array in the same order they appear while tracing the level sets and allocates to each point the description $v[i] = (P_i, type)$, where the set of types contains the following elements: minimum/ absolute minimum, saddle(with special subtypes according to the type before and after the saddle) and maximum/absolute maximum. The scheme of components is illustrated in the following table:

Tipul componentei	I	II	III	IV
Reprezentare				

Using the algorithm and pre-defined components we present the construction of the Reeb graph on a surface of genus 2 with ten critical points, considering f to be the height function.



2 Survey on the existing algorithms for a Reeb graph of real valued functions

The Reeb graph's importance has led to an expansion of algorithms and research towards the simplification and summarization of techniques. Given the fact that the volume of data needed to be computed has increased

drastically, the complexity of algorithms is among one of the first concerns when constructing a representation technique for Reeb graphs. In this section we will present a selection of the most common algorithms together with their complexity.

In 2003 Carr, Snoeyink and Axen [4] proposed an algorithm for computing contour trees using a technique of joining and splitting the trees and forming a final contour tree. Given that v is the number of vertices, e the number of edges, n the number of supernodes and $f(n)$ the slow-growing inverse Ackerman function the algorithm provides a complexity of type $O(v \lg(v) + e + nf(n))$.

Another popular method was introduced by Kunii and Shinagawa in the paper [10]. The algorithm works by keeping a record of the triangles that have common points with the level set while tracing the complexity in order of function values. For 2-dimensional manifold the initial complexity was $O(n^2)$ with a generalization with the parameters v -the number of vertices and n the size of vertices, edges and triangles, providing a running time of $O(vn)$. Paper [6] offers an improvement for the special case of 2-dimensional manifolds by obtaining the complexity $O(mln(v))$. Also, using a more complex technique Doraiswamy and Natarajan [7] managed to obtain the complexity $O(mlg(m)(lg(lg(m)))^3)$. The paper [12] provided an algorithm for 3 dimensional manifold considering i_l to be the number of independent loops of the graph and obtaining a complexity of $O(mlg(m) + i_lm)$.

A different approach was offered by J.Cheng [5] by eliminating the old technique of swiping the contour trees and replacing it with a surjection from the manifold to the Reeb graph. And finally the more recent approach of Hajij and Rosen [8] considers an algorithm that requires an easy implementation and takes up the least memory storage compared to existing algorithms.

3 Morse-Bot functions and the associated Reeb graph

According to [3], a smooth sub-manifold $N \subset C(f)$, $f : M \rightarrow \mathbb{R}$ is said to be a critical non-degenerate sub-manifold if the following conditions hold:

- a) N is without boundary;

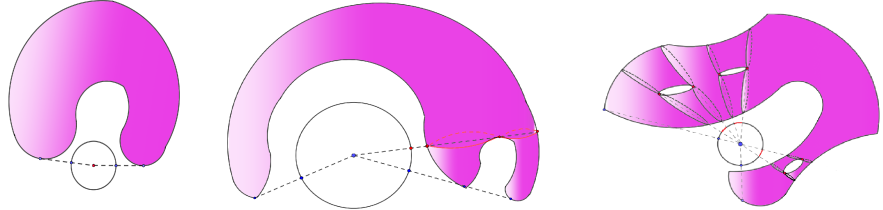
- b) N is compact and connected;
- c) for any $x \in N$, we have $T_x(N) = \ker (H(f)(x))$, where $H(f)(x)$ is the Hessian matrix of function f .

Definition 3.1 Let $f : M \rightarrow \mathbb{R}$ be a real valued function defined on a smooth manifold. We say that f is a Morse-Bott function if the critical set $C(f)$ contains only isolated points or critical non-degenerate smooth sub-manifolds.

Let M be a closed orientable surface and let $f : M \rightarrow S^1$ be a circular function on M . The function f is named circular Morse-Bott function if $C(f)$ contains only isolated points or critical non-degenerate smooth submanifolds. A more formal definition can be stated in the following way:

Definition 3.2 The function $f : M \rightarrow S^1$ is called a circular Morse-Bott function if for any $x \in M$, it exists a neighborhood V of $f(x)$ and a diffeomorphism $\phi : V \rightarrow \mathbb{R}$ such that, for $U = f^{-1}(V)$, the map $\phi \circ (f|_U)$ is a real Morse-Bott function.

Given the above definition, for a Morse-Bott function defined on a surface, the critical point set contains the following components: singular circles, extrema points (minimum/maximum) and critical points of index 1 called saddle points. In the images presented below it can be observed examples of Morse-Bott functions from a surface to the unit circle S^1 through the radial projection:

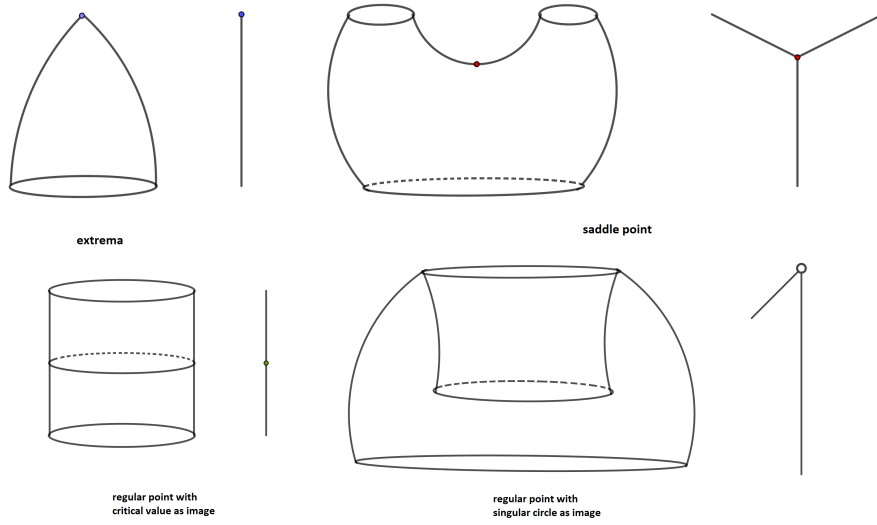


The classification problem for Morse functions was studied by important mathematicians as Arnold [2] or Sharko [14] and it got a new meaning considering the technological evolution in computer science. Therefore the

notion of Reeb graph can be extended to simple, circular Morse-Bott function on a surface M^2 as the quotient set $M^2/\sim$ associated to the simple Morse-Bott function $f : M^2 \rightarrow S^1$ with the following observations:

- a) for a critical value $c \in S^1$, the vertices of the graph are connected components of the level curves $f^{-1}(c)$;
- b) for a regular value $r \in S^1$, the edges of the graph are connected components of the level curves $f^{-1}(c)$.

The predefined components for constructing the Reeb graph for a circular Morse-Bott function are slightly different from the ones for the initial Reeb graph and are illustrated in the following image:



Description of algorithm

In order to give a better understanding to circular Morse-Bott functions and the applicability of this concept for practical purposes we will describe the algorithm and present three examples using the pre-defined components: extrema, saddle point, regular point with critical values as image and regular point with singular circle as image. The basic steps are the following:

- 1) identifying the critical points;
- 2) choosing an orientation for tracing the surface;
- 3) arranging the points into a vector $v[i] = (P_i, type)$;

4) identifying the type to each corresponding critical point:

m = minimum, M = maximum with pre-defined component of type I;

s = saddle point with component type II;

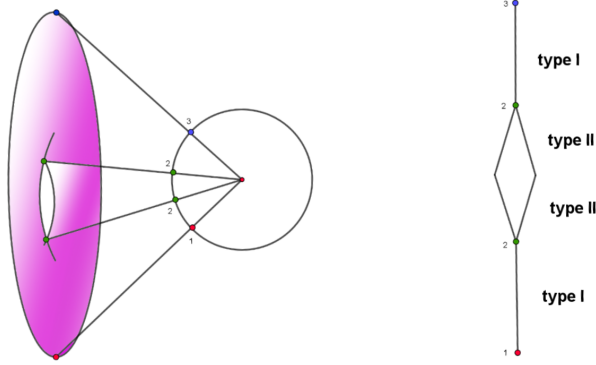
prc = regular point having image a critical value with component III;

prs = regular point having image a singular circle with component IV;

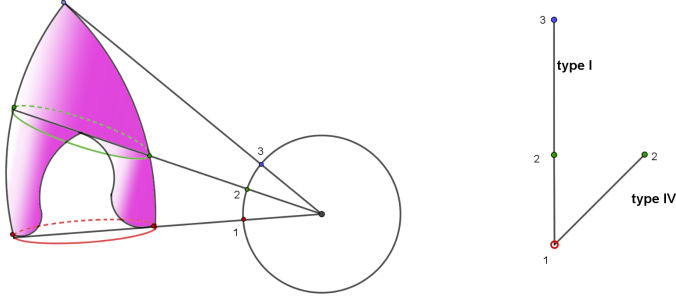
Remark: For components of type IV the Reeb graph will always start and end with symbol of singular circle $\circ$, therefore it will switch the order with the next point appearing on the circle used for making the radial projection. This component will also mark 3 points on the Reeb graph, the one corresponding to the singular circle $\circ$ and another two for each branch. The two double points will be marked as pr_{aux} .

All examples presented underneath consider M^2 to be a surface and f the radial projection.

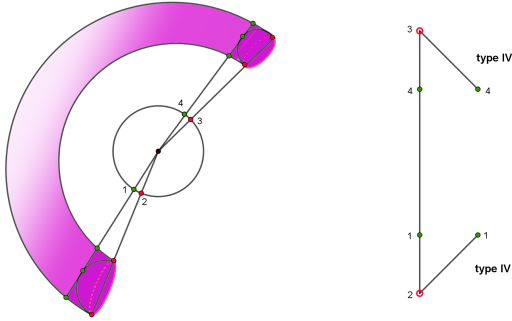
Example 1:



$$v[i] = ((P_1; m); (P_2; s), (P_3, s); (P_4, M))$$

Example 2:

$$v[i] = ((P_1; prs); (P_2; prs_{aux}), (P_3, M))$$

Example 3:

$$v[i] = ((P_1; prs_{aux}); (P_2; prs), (P_3, prs); (P_4, prs_{aux}))$$

References

- [1] D. Andrica. *Critical Point Theory and Some Applications*. Cluj University Press, 2005.
- [2] V.I. Arnold. *Statistics and classification of topologies of periodic functions and trigonometric polynomials*. Proceedings of the Steklov Institute of Mathematics, (S12-S23), 2006.
- [3] E.B. Batista, J.C.F. Costa and I.S. Meza-Sarmiento. *Topological classification of circle-valued simple Morse-Bott functions*. Journal of Singularities, Volume 17(2018), 388-402.

- [4] H. Carr, J. Snoeyink and U. Axen. *Computing contour trees in all dimensions*. Computational Geometry: Theory and Applications, 24(2003): 75-94.
- [5] J. Cheng. *Lecture 16: Computation of Reeb Graphs*. Ohio State University, <http://web.cse.ohio-state.edu/wang.1016/courses/788/Lecs/lec16-jack.pdf>.
- [6] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan and V. Pascucci. *Loops in Reeb graphs of 2-manifolds*. Proceedings of the 19th Annual Symposium on Computational Geometry, 344-350, ACM Press, 2003.
- [7] H. Doraiswamy and V. Natarajan. *Efficient algorithms for computing Reeb graphs*. Computational Geometry: Theory and applications, 46:606-616, 2009.
- [8] M. Hajij and P. Rosen. *An efficient data retrieval parallel Reeb graph algorithm*. arXiv:1810.08310v1, 2018.
- [9] M. Kaluba, W. Marzantowicz and N. Silva. *On representation of the Reeb graph as sub-complex of a manifold*. Topol.Methods Nonlinear Anal., 45(2015), no.1, 287-307.
- [10] T.L. Kunii and Y. Shinagawa. *Constructing a Reeb graph automatically from the cross section*. IEEE Comp. Graph. App. 11,5, 66-78, 1991.
- [11] A. Lupescu. *Note on an algorithm for computing the Reeb graph*. International Journal of Geometry, Volume 6(2017), No.1, 89-94.
- [12] V. Pascucci, G. Scorzelli, P.T. Bremer and A. Mascarenhas. *Robust on-line computation of Reeb graphs: simplicity and speed*. ACM Trans. Graph., 26(3):58 2007.
- [13] G.H. Reeb. *Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique*. Comptes Rendus de l'Académie des Sciences, 222 (1946), 847-849.
- [14] V. Sharko. *About Kronrod-Reeb graph of a function on a manifold*. Methods Funct. Anal. Topology, Volume 1(2006).