Available online at www.sciencedirect.com

SciVerse ScienceDirect

Mathematics and Computers in Simulation 99 (2014) 95–107

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.com/locate/matcom

Original article

Recovering functions: A method based on domain decomposition

Mira Bozzini [a], Licia Lenarduzzi [b,*]

[a] *Dip. Mat. Appl., Università di Milano Bicocca, via Cozzi 53, 20125 Milano, Italy*
[b] *IMATI CNR, via Bassini 15, 20133 Milano, Italy*

Received 27 October 2011; received in revised form 18 October 2012; accepted 4 April 2013
Available online 13 May 2013

Abstract

When the data are unevenly distributed and the behaviour of a function changes abruptly, the approximant can present undue oscillations. We present an algorithm to identify a domain decomposition, such that on each subdomain the behaviour of the function is sufficiently homogeneous in order to calculate separate approximants and to blend them together.

MSC: 65D15; 65D18

Keywords: Scattered data fitting; Contour tree; Computational topology

1. Introduction

In this paper we present an algorithm that makes use of an adaptive geometric technique based on a *contour tree* strategy.

The contour tree is typically used in computational geometry to identify topological changes of the contours, see [9]. In our context it is the tool to subdivide a domain in subdomains.

The algorithm that we propose is suitable to recover a surface when the modulus of the gradient of the function takes large values in some regions of the domain and it is smooth in regions nearby, that is when there are abrupt changes of behaviour.

In such a situation, an approximating function of global type on the whole domain can present false oscillations, even when we make use of a radial basis with low regularity, such as the thin plate spline (TPS) of class C^1.

The same drawback is also evident when using techniques of local type. In fact also the local approximant can present false oscillations if the function has abrupt behaviour locally, see [2].

In order to avoid this drawback, we need to identify a split of the domain as union of subdomains such that, within each one, there is no abrupt change of behaviour of f.

As it will be shown, the method provides effective results not only for the problem here presented but also in the more common case in which the sample is obtained on the basis of the behaviour of the function; it is effective in some applications of classification of subdomains.

The paper is organized as follows.

* Corresponding author. Tel.: +39 0223699526; fax: +39 0223699538.
E-mail addresses: mira.bozzini@unimib.it (M. Bozzini), licia.lenarduzzi@mi.imati.cnr.it, licia@mi.imati.cnr.it (L. Lenarduzzi).

0378-4754/$36.00
http://dx.doi.org/10.1016/j.matcom.2013.04.004

In Section 2 we describe the problem and we show the construction of a contour tree for gridded data.

In Section 3 we construct the contour tree for the current problem, and in Section 4 we define how to identify the subdomains by the tree. In Section 5 we present the construction of the approximant. At the end, in Section 6, we present some examples relevant to real data.

2. Definition of the problem and construction of a contour tree

We want to reconstruct a surface on a domain $\Omega \subset \mathcal{R}^2$ where it presents very different behaviours. In particular in some subregions of Ω the surface can have a smooth behaviour while in other parts of Ω it can present abrupt changes. Such a problem arises, for instance, when reconstructing surfaces relevant to geophysical problems. In these cases, the sampling is obtained by devices that collect the data not always in relation with the behaviour of the surface. Among all, a good example is the reconstruction of oceanographic surfaces where the data collection is entrusted to boats. The surface presents very heterogeneous behaviour and the data are not sampled according to the behaviour of it. In addition their locations are very irregular with varying density.

With this assumption, our purpose is to provide a faithful reconstruction of the surface with a moderate computational cost. For this aim, we present a two-step method.

In the first step, we organize the information from the data by a contour tree strategy. Hence we identify the subdomains of Ω where the behaviour of the function is reasonably homogeneous. In the second step, we recover the surface by a partition of the unity method, by using the subdomains just identified.

2.1. Construction of the contour tree T for gridded data

Let a function $z = f(\mathbf{x})$ be defined on a domain $\Omega \subset \mathcal{R}^2$.

Having fixed a value $z^\star$ belonging to the range of $f(\mathbf{x})$ on Ω, the plane $z = z^\star$ intersects the surface along one or more curves, which can in some cases be reduced to points.

In order to obtain closed curves, it is necessary to extend f out of the domain Ω by considering a domain $\Omega^\star \supset \Omega$ and such that a constant value $\overline{z} \leq \min_{x\in\Omega} f(\mathbf{x})$ is set at the border of $\Omega^\star$.

In such a way the plane $z = \overline{z}$ intersects the extended function $\tilde{f}$ on $\Omega^\star$ along a closed curve and each curve, obtained by the intersection between $z = z^\star$ and the surface, is closed on $\Omega^\star$.

In the following, we call: *contour* the set of curves associated to $z^\star$ and we denote it by Γ, while we call *component* each one of these curves, which component we denote by γ. We call *inner set* the region contained within such a component and we denote it by R.

When a curve γ is reduced to a point, we call it *singular component*.

Having assigned a sample of $\tilde{f}$ on a regular grid G, placed on $\Omega^\star$, we subdivide the interval $\omega_{\tilde{f}} = [\overline{z}, \max_{x_i\in G} f(\mathbf{x}_i)]$ in $(n+1)$ equal parts.

Let

$$M := \{m_j = \overline{z} + (j+1)\cdot \omega_{\tilde{f}}/(n+1), \quad j = -1, \ldots, n\}$$

be the set of the heights to be used to identify the contours. The *contour tree* T is a tree consisting of an ordered sequence of points (nodes of the tree). Each node is associated to one and only one component.

The association between node of the tree and component is given by the vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ whose components are counters:

- α_1 counts the number of bifurcations
- α_2 enumerates the branches originated at the α_1-th bifurcation
- α_3 enumerates the nodes of the branch (α_1, α_2).

The construction of the contour tree is explained here by an example. Precisely, having assigned a set of contours (Fig. 1), we start with the contour Γ_{-1} that contains the only component $\gamma_{0,a,a}$ and we associate to it the point $P_{0,1,1}$.

Following the order of inclusion, we consider the contour Γ_0 that contains the component $\gamma_{0,1,b}$ to which the node $P_{0,1,2}$ corresponds. The first branch terminates with the node $P_{0,1,3}$ corresponding to $\gamma_{0,1,3}$. Since $\gamma_{0,1,3}$ is followed