CONF-9006293--1

UCRL- JC-105136
PREPRINT

Received

NOV 19 1990

A Note on Piecewise Monotonic
Bivariate Interpolation

R.E. Carlson
F.N. Fritsch

This paper was prepared for submittal to
Conference on Curves and Surfaces,
Chamouix, France, June 21-27, 1990

October 1990

Lawrence
Livermore
National
Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.

A NOTE ON PIECEWISE MONOTONIC BIVARIATE INTERPOLATION

UCRL-JC--105136

DE91 002799

R. E. Carlson and F. N. Fritsch

Abstract

This paper outlines the design of a piecewise monotonic interpolation algorithm for bivariate gridded data

§1. Introduction

Let f be a bivariate function defined on $R = [a, b] \times [c, d]$ and let $p = p_1 \times p_2$ be a partition of R defined by $p_1: a = x_1 < x_2 < \dots < x_{nx} = b$; $p_2: c = y_1 < y_2 < \dots < y_{ny} = d$. Then a function $u_f(x, y)$ which satisfies the set of $nx \cdot ny$ equations

$$u_f(x_i, y_j) = f(x_i, y_j), \quad 1 < i < nx; \quad 1 < j < ny, \quad (1)$$

is said to be an *interpolant* of f .

Because the data lie on a grid, interpolation methods for solving this type of problem are called *gridded methods*, and usually consist of tensor products of univariate functions. Of special interest in this paper are shape preserving interpolation methods. That is, interpolation methods that preserve one or more *shape characteristics* (positivity, monotonicity, and/or convexity) present in the underlying function f .

In most shape preserving applications it is important for u_f to be reasonably smooth, i.e., that $u_f \in C^k[R]$ where $k \geq 1$. Bicubic Hermite (BCH) functions have one continuous derivative and are particularly well suited for solving these types of problems. A BCH interpolant is uniquely determined by the Hermite coordinates at each grid point, i.e. $(u_f)_{i,j}$, $(\partial u_f / \partial x)_{i,j}$, $(\partial u_f / \partial y)_{i,j}$, and $(\partial^2 u_f / \partial x \partial y)_{i,j}$. Because of condition (1) above, a BCH interpolation algorithm is simply a procedure for computing derivative estimates.

The most successful shape preserving interpolation results have involved preserving monotonicity. In [2] the authors derived a BCH interpolation algorithm which preserved monotonicity in both independent variables. The original BCH algorithm was simplified and improved in [3].

MASTER

Also in 1985, Beatson and Ziegler [1] derived a monotone interpolation algorithm in which each rectangular element $R_{i,j} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ was subdivided into sixteen triangular elements. A quadratic polynomial was then constructed in each triangular element which interpolated to the given data.

A second problem of interest involves bivariate data which are monotone in only one variable. A solution to this problem was presented in [4].

The monotone BCH interpolation algorithms noted above start by generating an initial set of values for $(\partial u_f / \partial x)_{i,j}$ and $(\partial u_f / \partial y)_{i,j}$ based on an appropriate univariate interpolation algorithm employed along each horizontal and vertical grid line. This initial set of values for the Hermite coordinates is used to check monotonicity constraints in the form of a set of linear inequalities. If the inequalities are satisfied, then the interpolant will satisfy the appropriate monotonicity conditions (provided $(\partial^2 u_f / \partial x \partial y)_{i,j}$ are chosen properly). If an inequality is not satisfied, the value of one or more partial derivatives is decreased in magnitude so that the new value(s) satisfy the inequality. Difficulties occur because the inequalities involve both $\partial u_f / \partial x$ and $\partial u_f / \partial y$. Care must be taken so that modifying a Hermite coordinate does not produce a violation in an equality that was previously satisfied.

In this paper we build on earlier BCH interpolation results plus some unpublished results by Fritsch to develop a BCH interpolation method for piecewise monotonic data. A complete derivation of the inequalities that must be satisfied and a thorough discussion of algorithm used is not possible in the space available. These results will be published elsewhere. Instead we focus on the underlying ideas and the problems that must be addressed.

It may be argued that all data are piecewise monotone, so that the solution to this problem is a general purpose algorithm. In this context the remark is true, but the strength of the algorithm lies in its ability to solve problems in which regions with steep gradients are adjacent to regions which are relatively flat. Most general purpose algorithms tend to overshoot in the flat areas, producing undesirable oscillations or "ringing". The algorithm presented here eliminates this behavior and produces an accurate, visually pleasing surface.

§2. Outline of the algorithm

To develop a bivariate piecewise monotonic interpolation algorithm it is necessary to answer two questions. First, where should monotonicity be enforced? Second, how do you construct an algorithm to enforce monotonicity in these subregions?

Because a BCH is a tensor product of univariate cubic polynomials, it is a bicubic polynomial in each rectangular element $R_{i,j}$. Therefore, the most reasonable subregion in which to enforce monotonicity is at the element level. Question one may be rephrased as, how should one determine the set of elements in which monotonicity should be enforced?

A simple answer to this question would be to have monotonicity in an element be determined by whether or not the data in that element are monotone. That is, in the element $R_{i,j}$ if the data are monotone nondecreasing in x ,

(i. e. if $(f_{i+1,j+1} - f_{i,j+1})$ and $(f_{i+1,j} - f_{i,j})$ are nonnegative), then the interpolant should also be monotone nondecreasing in x in $R_{i,j}$. (Similarly for the case in which the data are nonincreasing.) A similar test for monotonicity of the data would be made in the y -direction. An interpolation method in which both the interpolant and the data obey the same monotonicity properties in each element is called a *co-monotone* interpolation method.

Unfortunately, co-monotonicity is not a satisfactory approach for determining where monotonicity should be enforced. This view is illustrated in Figure 1. The data change from monotone increasing to monotone decreasing at x_3 . Therefore, a co-monotone interpolant would be monotone increasing on $[x_1, x_3]$, monotone decreasing on $[x_3, x_5]$ and have a zero derivative at x_3 . The interpolant would look similar to the solid curve. A more reasonable solution would be to allow the local maximum to be interior to one of the intervals, in this case in the interval $[x_3, x_4]$, as illustrated by the dotted curve.

Figure 1. Co-monotonicity is not a sufficient criterion for determining elements in which monotonicity should be enforced

This example clearly demonstrates that unless all extrema are known to occur at data points, it is better to permit extrema to occur on the interior of an interval rather than only at the endpoints. Therefore, what is needed is an algorithm that automatically selects an appropriate location for the extremum in order to produce a visually pleasing curve.

This problem was extensively studied by Fritsch in the 1980s for univariate cubic Hermite interpolation. He discovered that developing a general algorithm which was a continuous function of the data and which would produce a visually pleasing curve in regions where monotonicity changed was a nontrivial problem. In some unpublished work he derived an algorithm that depended on the data at five points: the point in question and two points on either side. This algorithm has been implemented in subroutine PCHIC, which is part of the Fortran Piecewise Cubic Hermite Interpolation Package called PCHIP (see [5]).

These univariate results are applied to the bivariate problem by using

PCHIC along each horizontal and vertical grid line. This step produces an initial set of values for $(\partial u_f/\partial x)_{i,j}$ and $(\partial u_f/\partial y)_{i,j}$. If the signs of $\partial u_f/\partial x$ ($\partial u_f/\partial y$) agree at all four corners of element $R_{i,j}$, then monotonicity in x (y) will be enforced in that element.

The answer to the second question above is significantly more complicated and space does not permit a detailed description of the inequalities that must be satisfied. In order to guarantee a bicubic polynomial is monotone in an element it is sufficient that a collection of linear inequalities involving the Hermite coordinates be satisfied. If the inequalities are not satisfied by the initial Hermite derivatives, the magnitude of one (or more) derivatives is (are) decreased. The new set of values then satisfies the inequality. Changing one or more Hermite derivatives may have a cascading effect, in that additional values may have to be changed along a row or a column. In no case is the sign of a Hermite derivative changed. Once an inequality is satisfied, it is never subsequently violated due to a change in one or more Hermite coordinates. Once all of the inequalities are satisfied, an appropriate set of values for $(\partial^2 u_f/\partial x \partial y)_{i,j}$ is computed. The resulting BCH interpolant is piecewise monotone, and the monotonicity regions are consistent with those determined by PCHIC.

A detailed derivation of this algorithm will be contained in a forthcoming paper.

§3. Conclusions

The ability to construct a piecewise monotone interpolant to piecewise monotonic data is the final step in providing a set of solutions to a collection of monotone interpolation problems. All of the solutions are practical and have been (or are being) implemented in portable Fortran software packages.

References

1. Beatson, R., and Z. Ziegler, Monotonicity preserving surface interpolation, *SIAM J. Numer. Anal.* **22** (1985), 401-411.
2. Carlson, R. E., and F. N. Fritsch, Monotone piecewise bicubic interpolation, *SIAM J. Numer. Anal.* **22** (1985), 386-400.
3. Carlson, R. E., and F. N. Fritsch, An algorithm for monotone piecewise bicubic interpolation, *SIAM J. Numer. Anal.* **26** (1989), 230-238.
4. Carlson, R. E., and F. N. Fritsch, A bivariate interpolation algorithm for data which are monotone in one variable, *SIAM J. Sci. Stat. Comput.* (1990), to appear.
5. Fritsch, F. N., PCHIP Final Specifications, Computer Documentation Report UCID-30194, Lawrence Livermore National Laboratory, Livermore, California, 1982.

This work was performed under the auspices of the U. S. Department of Energy by Lawrence Livermore National Laboratory under contract W-7504-Eng-48.

END

DATE FILMED

12 / 12 / 90