

Constrained Visualization Using the Shepard Interpolation Family

K. W. Brodlie,¹ M. R. Asim² and K. Unsworth³

¹School of Computing, University of Leeds, Leeds LS2 9JT, UK

²COMSATS Institute of Information Technology, Off Raiwind Road, Lahore, Pakistan

³Applied Computing, Mathematics and Statistics Group, Division of Applied Management and Computing, P.O.Box 84, Lincoln University, Canterbury, New Zealand

Abstract

This paper discusses the problem of visualizing data where there are underlying constraints that must be preserved. For example, we may know that the data are inherently positive. We show how the Modified Quadratic Shepard method, which interpolates scattered data of any dimensionality, can be constrained to preserve positivity. We do this by forcing the quadratic basis functions to be positive. The method can be extended to handle other types of constraints, including lower bound of 0 and upper bound of 1—as occurs with fractional data. A further extension allows general range restrictions, creating an interpolant that lies between any two specified functions as the lower and upper bounds.

Keywords: visualisation, interpolation, Shepard's method, shape preservation, positivity, constraints

ACM CCS: I.3.5 Computer Graphics—Computational Geometry and Object Modelling G.1.1 Numerical Analysis: *Interpolation* G.1.6 Numerical Analysis: *Optimization*

1. Introduction

Visualization can be seen as a process of visual reconstruction. We create a representation of the overall behavior of the entity we are interested in, from a limited set of sampled information. This reconstruction is achieved by interpolation. However we often have some additional information that we wish to build into the reconstruction: the entity may be subject to certain physical laws that constrain its behavior—for example, we know densities are always positive and any credible visualization must honor this. Another common constraint occurs with data values that are specified as fractions of a whole—here the reconstruction must lie within the range $[0,1]$ to be realistic. In this paper, we examine one particular interpolation approach—the Shepard family of interpolants—and show how this can be adapted to handle constraints on the range of the interpolant.

The problem we are addressing is the interpolation of scattered data. This problem occurs in very many practical situations where data are gathered experimentally (for example,

we shall look later at rainfall measurements gathered at a set of recording stations) or is computed in a simulation process using an unstructured grid. There are many approaches to this general problem—a good review of the whole area is given by Lodha and Franke [1].

Some methods are based on an initial triangulation of the data points (or equivalent in higher dimensions), followed by a piecewise construction of the interpolant—one piece per triangle. A very simple technique of this type is piecewise linear interpolation. This has a nice property of remaining within the bounds of the data, and thus preserving for example positivity in the data. However, it is only C^0 continuous. Smoother interpolants over triangulations will typically fail to remain within the bounds of the data, but several have been modified to incorporate constraints. For example, Asim [2] modifies the Barnhill *et al.* [3] blending method (constrained cubics as suggested by Asim and Brodlie [4] are created along triangle edges and blended in the interior); Ong and Wong [5] create a C^1 interpolant by blending constrained rational cubics along

triangle edges using the Nielson [6] side-vertex method. Mulansky and Schmidt [7] construct a constrained C^1 interpolant using quadratic splines on a Powell–Sabin refinement of the original triangulation. Chan and Ong [8] create a constrained C^1 interpolant as a combination of cubic Bezier triangles.

All these approaches, however, require the points to be triangulated. Another major class of methods for scattered data interpolation does not involve any prior triangulation step, and can be thought of as ‘meshless’. The two main types are radial basis functions (RBFs), which include multiquadrics and thin-plate splines, and Shepard-type methods, which include the modified quadratic Shepard approach and also the moving least squares technique. Both types, RBFs and Shepard, are widely used in practice.

However, there has been relatively little work done on the imposition of constraints for these meshless methods. For RBFs, in the special case of thin-plate splines for 2D data, Utreras [9] has shown how positivity can be imposed as a constraint, but the computational cost is rather high, requiring a global optimization problem to be solved at each step of an iteration. Xiao and Woodbury [10] look at a number of meshless methods for constrained scattered data interpolation for 3D data. In areas where the entity is known to have a particular value, say, zero, extra data points are added in order to ‘encourage’ the interpolant to take values close to zero in these areas. If a physical constraint additionally tells us that the entity is nonnegative, then the interpolant is simply clamped at zero. A difficulty with this approach is that the resulting interpolant will have derivative discontinuity where the clamping is applied. Our aim is to generate a constrained interpolant which is computationally efficient, and which incorporates the constraint as part of the interpolation process, rather than as an *a posteriori* process such as clamping.

We shall adapt the Shepard family of interpolants. Section 2 describes these in detail—a Shepard interpolant is essentially a weighted mean of basis functions. By constraining each basis function to be positive, we immediately obtain a positive interpolant. Section 3 explains how we can impose the positivity constraint in an efficient manner, while Section 4 generalizes the work from simple positivity to arbitrary upper and lower bound constraints. Section 5 discusses the quality of the constrained interpolant, and finally Section 6 concludes and suggests further work.

There is one important word to add on terminology. We shall use the term ‘positivity’ to refer to ‘greater than or equal to zero’, rather than the more rigorous but somewhat awkward ‘non-negativity’. When we mean ‘greater than zero’, we use the term ‘strictly positive’.

2. Shepard Family of Interpolants

The general problem we are addressing is the following. Given a set of N data points \mathbf{x}_i , $i = 1, 2, \dots, N$,

where $\mathbf{x} = (x, y, z, \dots)$, with associated data values f_i , we seek an interpolating function $F(\mathbf{x})$ such that $F(\mathbf{x}_i) = f_i$. Later, we shall be concerned with imposing constraints on the behavior of $F(\mathbf{x})$, but to start with we consider the unconstrained case.

A popular approach to the problem emerged in the 1960s among the contour plotting community, and is now associated with the name of one of its proponents, Shepard [11]. In its basic form, it involves an inverse distance-weighted average of the data values, constructing $F(\mathbf{x})$ as:

$$F(\mathbf{x}) = \sum_{i=1}^N w_i(\mathbf{x}) f_i \quad (1)$$

where the normalized weight function $w_i(\mathbf{x})$ has the form:

$$w_i(\mathbf{x}) = \frac{\sigma_i(\mathbf{x})}{\sum_{j=1}^N \sigma_j(\mathbf{x})} \quad (2)$$

where $\sigma_i(\mathbf{x}) = \frac{1}{d_i^2(\mathbf{x})}$ with $d_i = \|\mathbf{x} - \mathbf{x}_i\|_2$

The weights $w_i(\mathbf{x})$ satisfy:

1. $\sum_{i=1}^N w_i(\mathbf{x}) = 1$
2. $w_i(\mathbf{x}) \geq 0$
3. $w_i(\mathbf{x}_j) = \delta_{ij}$.

Note that we have been able to specify the method for arbitrary dimension of the space; we have not had to specify any connectivity between data points; and we have not had to solve any linear system of equations (as is needed in the radial basis function approach).

In practice, however, this method does not work particularly well, for two reasons:

1. The function $F(\mathbf{x})$ has zero derivative at the data points, exhibiting as ‘flat spots’ in the interpolating curve or surface in 1D or 2D respectively. This led to the suggestion by Franke and Nielson [12] of replacing the constant f_i in the averaging process by a local best-fit quadratic approximation $Q_i(\mathbf{x})$.

2. The method is global in the sense that any interpolation involves a computation involving all data points. This is computationally inefficient. This led to the suggestion by Franke and Little (reported in [13]) that the weighting functions σ_i be subjected to a damping factor λ to reduce them to zero outside a certain radius of the data point.

When these two modifications are taken together, we have the modified quadratic Shepard method (as proposed by Franke and Nielson [12] and further discussed by Nielson [14]). We create an interpolant $F(\mathbf{x})$ as:

$$F(\mathbf{x}) = \sum_{i=1}^N w_i(\mathbf{x}) Q_i(\mathbf{x}) \quad (3)$$

Table 1: Percentage of oxygen in flue gas.

Time (mins)	0	2	4	10	28	30	32
oxygen (%)	20.8	8.8	4.2	0.5	3.9	6.2	9.6

where the normalized weight function $w_i(\mathbf{x})$ has the form:

$$w_i(\mathbf{x}) = \frac{\sigma_i(\mathbf{x})}{\sum_{j=1}^N \sigma_j(\mathbf{x})} \quad (4)$$

where $\sigma_i(\mathbf{x}) = \frac{1}{d_i^2(\mathbf{x})} (1 - \frac{d_i(\mathbf{x})}{r_w})_+^2$ with $d_i = \|\mathbf{x} - \mathbf{x}_i\|$, and where r_w is a constant defining an area of interest around the interpolation point \mathbf{x} , outside of which basis functions have zero weight.

$Q_i(\mathbf{x})$ is the best inverse distance weighted least-squares approximation by a quadratic function to the data points. The least-squares calculation is again restricted to those data points within a specified radius, say r_q , of \mathbf{x}_i , in order that the method is local. We write Q_i as:

$$Q_i(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T A(\mathbf{x} - \mathbf{x}_i) + \mathbf{g}^T(\mathbf{x} - \mathbf{x}_i) + f_i \quad (5)$$

(For simplicity of notation, the subscript i is omitted from the terms A , \mathbf{g} on the RHS.)

The modified quadratic Shepard method is now widely used, and an implementation of the ACM algorithm of Renka [15–16] is available in the NAG Library [17]. The original Shepard method of Equation (1) is rarely, if ever, seen in practice. However the original method does have one useful property which we lose in the modified quadratic version. As explained by Gordon and Wixom [18], the original Shepard method satisfies the following Maximum Principle:

Theorem 1 (Maximum Principle for Shepard’s Method)
 Let $M = \max\{f_i\}$ and $m = \min\{f_i\}$. Then $F(\mathbf{x})$ satisfies:

$$m \leq F(\mathbf{x}) \leq M. \quad (6)$$

Thus the interpolant lies within the range of the data, and one consequence for example is that a positive interpolant is guaranteed if the data values are positive.

We illustrate this with a very simple example, in 1D. The data set in Table 1 shows the percentage of oxygen in the flue gas, as coal burns in a furnace. The oxygen percentage is inherently positive, and we therefore require the interpolant to preserve this property (we have used this data set previously to demonstrate positive curve drawing by piecewise cubics—see [4]). Figure 1 shows the original Shepard interpolant—the ‘flat spots’ are very evident, and indeed the appearance is in general unsatisfactory. However note that the curve does remain positive. As we extrapolate to infinity, the value of the curve tends to the average of the data values.

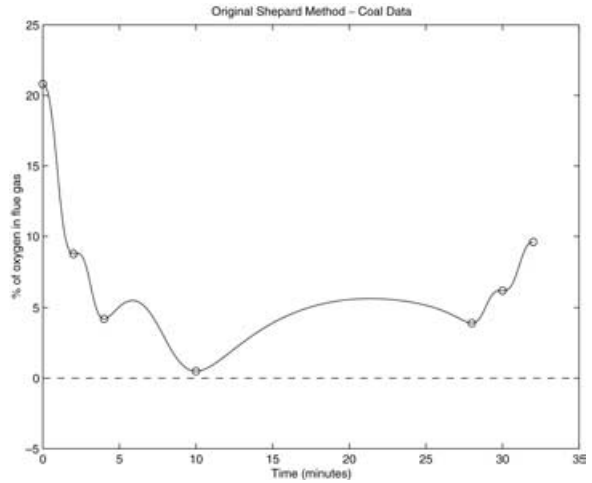


Figure 1: One-dimensional coal burning data—Flat-spots from the Original Shepard Method.

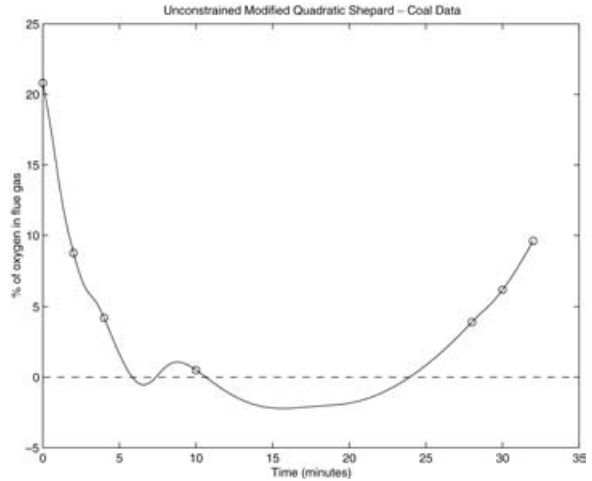


Figure 2: One-dimensional coal burning data—Modified Quadratic Shepard method loses positivity.

Figure 2 shows the modified quadratic Shepard interpolant, applied to the same data set. Generally the behavior is far superior, but the curve now goes beyond the range of the data values and indeed the positivity constraint is violated. We see the same behavior when the method is applied to surface interpolation in 2D, or volumetric interpolation in 3D, or indeed higher dimensions.

This motivates our work. We would like to retain the improved interpolation behavior of the modified quadratic approach, but we would like to be able to impose constraints. Rather than the Maximum Principle of Theorem 1 and the original Shepard method, we would like to express the

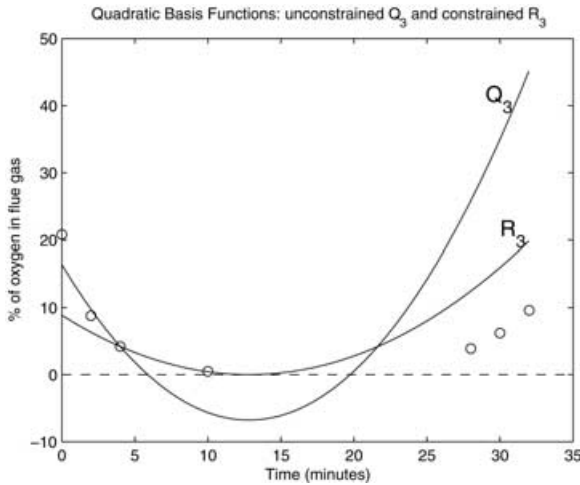


Figure 3: Quadratic basis function Q_3 has negative values while the constrained basis function R_3 (discussed in Section 3.2) is entirely positive.

constraint in a way which is detached from the data. In the first instance we shall consider positivity, and so we seek an interpolant $F(\mathbf{x})$ which will satisfy the constraint

$$F(\mathbf{x}) \geq 0. \tag{7}$$

3. Constrained Modified Quadratic Shepard Method

3.1. How Positivity is Lost

We can gain useful insight into the problem through examining the 1D example of the previous section. In Figure 3, we show the quadratic basis function $Q_3(x)$ that is generated, interpolating at $(x_3, y_3) = (4.0, 4.2)$ and approximating the other data in a weighted least-squares sense. It clearly goes negative within the range of interpolation, and contributes to the loss of positivity exhibited in the overall modified quadratic Shepard, or MQS, interpolant shown in Figure 2.

Remember that F is a positive linear combination of the Q_i values at x . Therefore, we can ensure positivity if we can constrain each basis function to be positive within the range of the interpolation. This sufficient condition is a key point of our approach.

3.2. Positive Quadratic Basis Functions

Our objective then is to constrain the quadratic basis functions to be positive within the region where they are active. For the modified quadratic version, this means that the i th basis function must be positive within a region:

$$\|\mathbf{x} - \mathbf{x}_i\|_2 \leq r_w. \tag{8}$$

We are therefore interested in solving the problem: minimize

$$Q_i(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T A(\mathbf{x} - \mathbf{x}_i) + \mathbf{g}^T(\mathbf{x} - \mathbf{x}_i) + f_i \tag{9}$$

subject to the constraint (8).

If the minimum is positive, then obviously $Q_i(\mathbf{x})$ is positive everywhere it is active in the interpolation calculation and no action need be taken. If the minimum is negative, however, then it is possible that the basis function could contribute a negative component in the evaluation of $F(\mathbf{x})$ in Equation (3). In this case we modify Q_i .

Inspection of Figure 3 lets us motivate the modification. The range of Q_3 is too great, and thus we are led to apply a positive scaling factor, α say, where $\alpha < 1$. The factor α must compress the range of data values $[Q_3^{\min}, f_3]$ (where Q_3^{\min} is the minimum of Q_3) to the range $[0, f_3]$. This scaling will destroy the interpolation condition, $Q_3 = f_3$, and so we also apply a shift, $\beta = (1 - \alpha)f_3$, in order to retain interpolation. In this way, we construct a constrained quadratic basis function, R_3 , which is a scaled and shifted transform of Q_3 , compressing the range of Q_3 , while making sure it still passes through the data point. The constrained R_3 is shown alongside the unconstrained Q_3 in Figure 3.

In general, then, we construct for any Q_i which goes negative, a revised basis function R_i , which is a linear transformation of Q_i :

$$R_i(\mathbf{x}) = \alpha Q_i(\mathbf{x}) + \beta \tag{10}$$

where we apply a scale factor $\alpha \in [0, 1]$ to reduce the range of Q_i and a shift factor β to maintain interpolation. Specifically,

$$\alpha = \frac{f_i}{f_i - Q_i^{\min}}, \quad \beta = (1 - \alpha)f_i \tag{11}$$

where Q_i^{\min} is the minimum of Q_i within the region it is active.

There are two points to note at this stage:

- If $f_i = 0$ for any i , that is, the data value equals the constraint, then we have $\alpha = 0$ and $\beta = f_i$. Thus $R_i(\mathbf{x}) = f_i$, and the basis function reverts to the constant value used in the original Shepard method 1.
- If we want to construct an interpolant that is strictly positive, then we need to choose α such that:

$$0 < \alpha < \frac{f_i}{f_i - Q_i^{\min}}.$$

The smaller the value of α , the less is the range of R_i .

This simple ‘scale-then-shift’ operation has some nice properties in addition to raising the minimum to zero, and preserving the interpolation condition. First we rewrite

Equation (9) in terms of the unique stationary point of Q_i , say \mathbf{x}_s , as:

$$Q_i(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_s)^T A(\mathbf{x} - \mathbf{x}_s) + Q_s \quad (12)$$

where Q_s is the value of Q_i at \mathbf{x}_s . Then we have:

$$\begin{aligned} R_i(\mathbf{x}) &= \alpha Q_i(\mathbf{x}) + \beta \\ &= \frac{1}{2}(\mathbf{x} - \mathbf{x}_s)^T (\alpha A)(\mathbf{x} - \mathbf{x}_s) + \gamma \end{aligned} \quad (13)$$

where $\gamma = \beta + \alpha Q_s$. From Equation (13), it is clear that R_i has the same stationary point, \mathbf{x}_s , as Q_i and moreover, since the Hessian matrix A is scaled by $\alpha \in [0, 1]$, the eigenvectors of the new Hessian are unchanged, and the eigenvalues are scaled by a uniform positive constant α . This implies that the essential nature of the function, in terms of convex and concave regions, is unchanged by the linear transformation to R_i .

Specifically, we have the following property:

Property 1 Suppose \mathbf{x}_A and \mathbf{x}_B are any two points such that

$$Q_i(\mathbf{x}_A) \leq Q_i(\mathbf{x}_B). \quad (14)$$

Then it follows from Equation (10) that

$$R_i(\mathbf{x}_A) \leq R_i(\mathbf{x}_B). \quad (15)$$

In the next subsection, we look at how this works for the one-dimensional case, as a simple illustration of the method. For higher dimensions, the solution of the constrained minimization problem (given by (9)) requires some discussion, as the approach will only be feasible if this can be solved efficiently—so this is described in the following Section (3.4). We then show how the method works in practice on 2D and 3D interpolation problems.

3.3. One-Dimensional Positive MQS

In the one-dimensional case, the problem (9) reduces to: minimize

$$Q_i(x) = \frac{1}{2}a(x - x_i)^2 + g(x - x_i) + f_i \quad (16)$$

subject to the constraint $|x - x_i| \leq r_w$.

If $a \leq 0$, that is, Q_i is concave, then the minimum, x_{\min} , will lie at an end-point of the interval; if $a > 0$, then Q_i is convex and x_{\min} may lie in the interior (if $x_{\min} = x_i - \frac{g}{a} \in [x_i - r_w, x_i + r_w]$) or at an end-point otherwise. Whatever be the case, it is straightforward to calculate the linear transformation:

$$R_i(x) = \alpha Q_i(x) + \beta \quad (17)$$

with α and β chosen according to Equation (11).

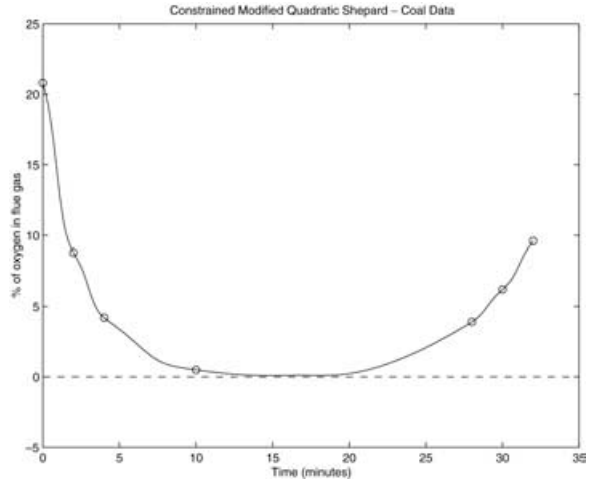


Figure 4: One-dimensional coal burning data—Constrained Modified Quadratic Shepard method.

As mentioned earlier, Figure 3 shows the revised quadratic basis function R_3 for our simple example, with the original Q_3 alongside. Note how the new basis function is positive, and retains the same shape except for being ‘squashed’. In Figure 4, we show the resulting ‘constrained MQS’ interpolant when the revised basis functions are combined in the style of Equation (1).

3.4. Solving the Constrained Minimization Problem

In the one-dimensional case, the constrained minimization problem (9) was easy to solve. For higher dimensions the situation is less trivial, and the success of the constrained interpolant depends on being able to solve this efficiently. Fortunately, just such an efficient solution is provided by the Levenberg–Marquardt algorithm.

Recall that the problem to be solved is the following: minimize

$$Q_i(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T A(\mathbf{x} - \mathbf{x}_i) + \mathbf{g}^T(\mathbf{x} - \mathbf{x}_i) + f_i \quad (18)$$

subject to the constraint (8). The following theorem (see Theorem 5.2.1 of Fletcher [19] for proof) gives the solution to minimizing a quadratic function within a sphere of given radius, r_w , about a given point \mathbf{x}_i :

Theorem 2 (Levenberg–Marquardt) The point

$$\mathbf{x}(\nu) = \mathbf{x}_i - (A + \nu I)^{-1}(-\mathbf{g}) \quad (19)$$

is the solution of the problem (18), if and only if there exists $\nu \geq 0$ such that

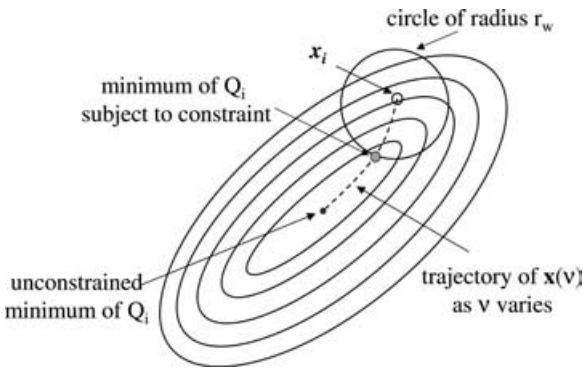


Figure 5: Levenberg–Marquardt algorithm: The dotted line shows the trajectory of $\mathbf{x}(v)$ as v varies. Each point on this trajectory is the solution of a constrained minimization of Q_i for some value of r_w . One instance of r_w is shown in the diagram. For $r_w = 0$, the minimum is at \mathbf{x}_i and as r_w increases the minimum follows the trajectory shown, until eventually the unconstrained minimum is reached. Increasing r_w further does not alter the minimum.

- $A + vI$ is positive semi-definite
- if $v > 0$, then $\|\mathbf{x} - \mathbf{x}_i\|_2 = r_w$

If such a v exists, it is unique.

Levenberg–Marquardt algorithms typically proceed as follows. Equation (19) defines a trajectory $\mathbf{x}(v)$, and we seek the value of v such that

$$\|\mathbf{x}(v) - \mathbf{x}_i\|_2 = r_w. \tag{20}$$

This is a nonlinear equation in one variable, v , and is relatively straightforward to solve.

Insight into the calculation is provided in Figure 5. Here we see the contours of a two-dimensional quadratic, Q_i , associated with data point \mathbf{x}_i . The unconstrained minimum of Q_i is shown, and this is the solution of the problem (19), for sufficiently large r_w . However, as we reduce r_w , the trajectory $\mathbf{x}(v)$ follows the path shown in the dotted line toward \mathbf{x}_i , which it reaches as r_w tends to zero. For any given r_w , the point where the trajectory intersects the circle $\|\mathbf{x} - \mathbf{x}_i\|_2 = r_w$ is the required minimum point.

This example also gives us insight into what happens when we replace Q_i by its modified form R_i . Figure 6 shows the contours of R_i corresponding to the Q_i of Figure 5. Note that the contours are identical in shape, in fact it is only the values attached to contours that change. The contour line through \mathbf{x}_i is unique in being unchanged, those below are increased, those above are decreased. The zero contour goes through the intersection between the trajectory and the constraining circle.

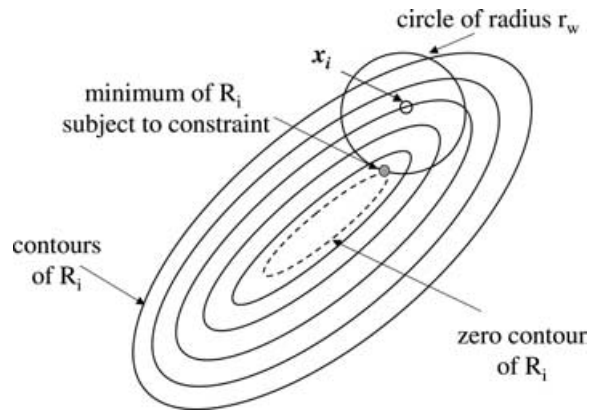


Figure 6: Transformation from Q_i to R_i : The diagram shows the contours for R_i which is a scaled and shifted transform of the Q_i of Figure 5, the factors α, β chosen to achieve positivity using Equation (11). Note that the contours are unchanged in shape. Their values however are transformed: those above f_i are reduced in value; those below f_i are increased in value; the f_i contour line is unchanged in value. The dotted line shows the zero value contour of R_i ; this line, which passes through the constrained minimum point of R_i , has had its value increased from Q_i^{\min} to zero.

Although the figures describe the two-dimensional case, note that the Levenberg–Marquardt algorithm applies to any dimensionality.

3.5. Practical Examples in 2D and 3D

As noted earlier, there are many examples where positive interpolants are important. The case we use here to illustrate the method is rainfall data from sites in New Zealand, supplied by the New Zealand National Institute of Water and Atmospheric Research [20]. The data were collected at some 133 stations throughout New Zealand, and represent the measurement of total rainfall in millimeters, for 2nd May 2002. Figure 7 shows interpolation using the normal, unconstrained MQS method, using all the data for the interpolation but zooming in on a region at the north part of New Zealand’s South Island, Farewell Spit. The contour map shows the interpolant generating negative values, which are clearly unreal. By contrast, Figure 8 shows the constrained method, with all areas showing positive rainfall values. The small circles indicate data points.

Figures 9 and 10 show similar contrasting behavior in a region near Arthur’s Pass on South Island.

These examples are both two-dimensional. In order to illustrate the method in 3D, we included the heights of the weather stations as well as the latitude and longitude, and created a 3D Shepard interpolant. To display the rainfall in

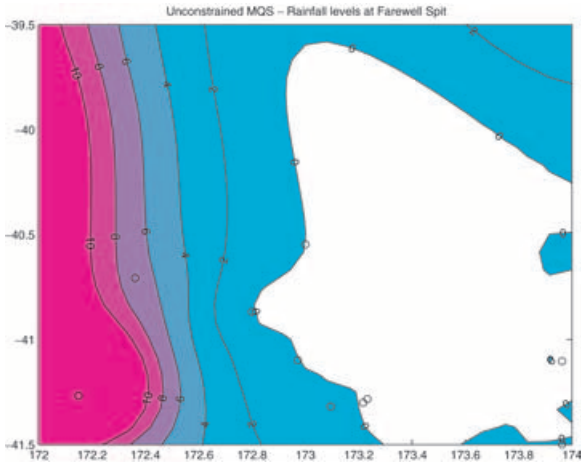


Figure 7: Unconstrained MQS interpolation of rainfall levels near Farewell Spit, New Zealand.

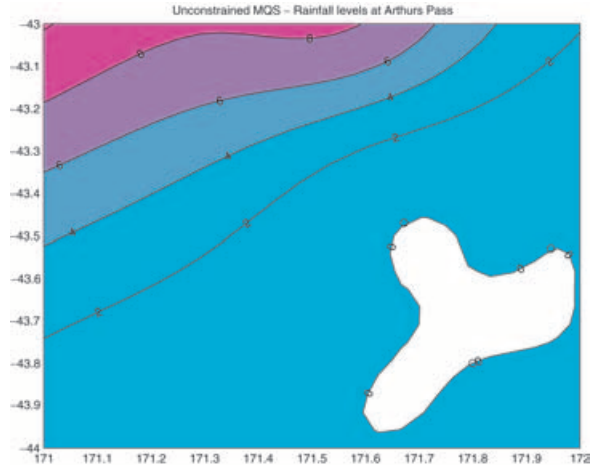


Figure 9: Unconstrained MQS interpolation of rainfall levels near Arthur's Pass, New Zealand.

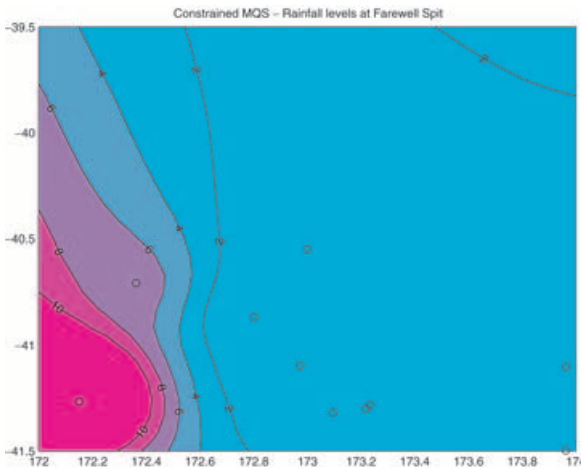


Figure 8: Constrained MQS interpolation of rainfall levels near Farewell Spit, New Zealand.

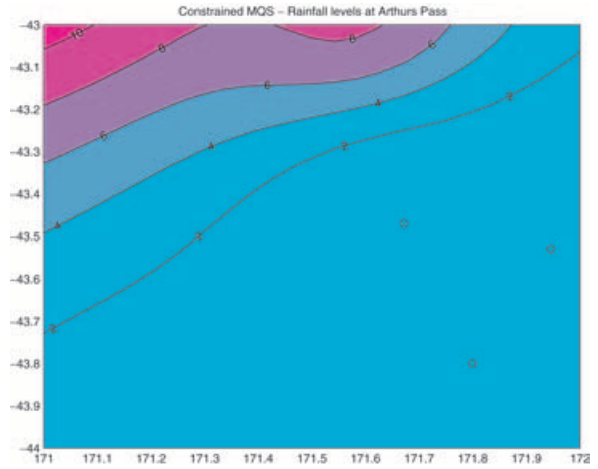


Figure 10: Constrained MQS interpolation of rainfall levels near Arthur's Pass, New Zealand.

the Arthur's Pass region, we used the unconstrained MQS to create a surface approximating the terrain, and then evaluated the 3D rainfall interpolant over this surface—first using the 3D unconstrained interpolant (shown in Figure 11) and second using the 3D constrained interpolant (shown in Figure 12). As expected, negative rainfall values occur with the unconstrained, but not the constrained version.

4. General Constraints

Once we know how to achieve a positivity constraint, we are able to apply **any** constraint on the value of the interpolant, provided the constraint is satisfied by the data. We consider a variety of cases in turn, and then illustrate with an example.

4.1. Arbitrary Lower Bound

Suppose we wish to construct an interpolant $F(\mathbf{x})$ such that

$$F(\mathbf{x}) \geq B(\mathbf{x}) \tag{21}$$

where $B(\mathbf{x})$ is any function of \mathbf{x} . Suppose the data values f_i satisfy the constraints

$$f_i \geq B(\mathbf{x}_i), i = 1, 2, \dots, N. \tag{22}$$

A simple approach (as suggested by Chan and Ong [8] and Asim and Brodlie [4]) is to convert this into an equivalent positivity problem. Using the techniques just described, we construct a positive interpolant $T(\mathbf{x})$ to the positive data values $t_i = f_i - B(\mathbf{x}_i)$, and form the required interpolant $F(\mathbf{x})$

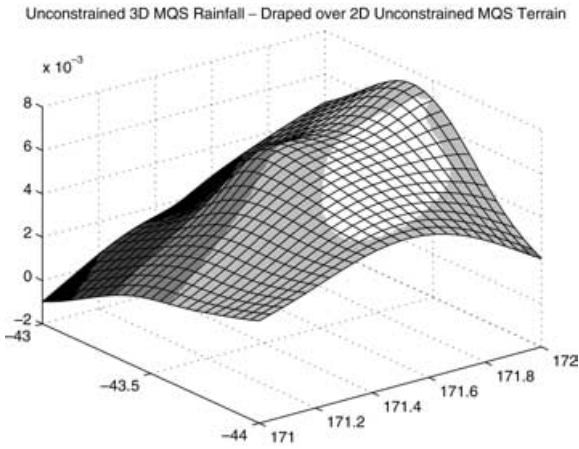


Figure 11: Unconstrained MQS rainfall draped over unconstrained MQS terrain. Note the white area indicates negative values of rainfall.

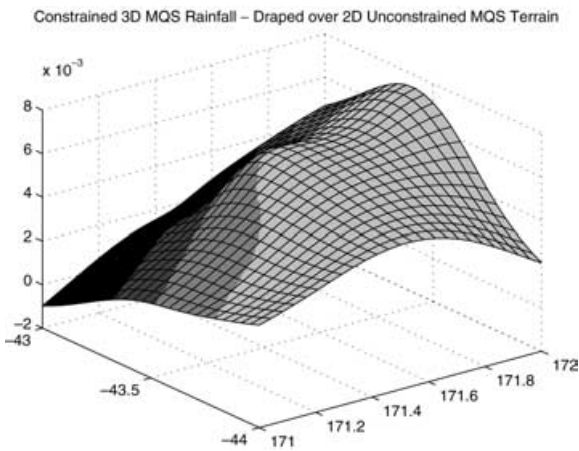


Figure 12: Constrained MQS rainfall draped over unconstrained MQS terrain.

as:

$$F(\mathbf{x}) = T(\mathbf{x}) + B(\mathbf{x}). \tag{23}$$

This has important applications where one wants to construct one surface above another. This would occur for example with borehole measurements, where one wanted to show one strata of rock above another. The case of an arbitrary upper bound follows similarly.

4.2. Upper and Lower Bounds—the [0, 1]-Constraint Problem

Suppose we want to constrain the interpolant to the range [0, 1]—for example, the data might be expressed as fractions. We apply exactly the same approach as for positivity, scaling Q_i to make its range fit within [0, 1], and then shifting so that interpolation is preserved. Specifically, we construct a new basis function R_i as:

$$R_i(\mathbf{x}) = \alpha Q_i(\mathbf{x}) + \beta \tag{24}$$

where we apply a scale factor $\alpha \in [0, 1]$ to reduce the range of Q_i and a shift factor β to maintain interpolation. The required scale factor α is the smaller of the scale factors required to achieve lower bound of 0 and upper bound of 1, namely

$$\alpha = \min\{\alpha_{\text{lower}}, \alpha_{\text{upper}}\}$$

where

$$\alpha_{\text{lower}} = \frac{f_i}{f_i - Q_i^{\min}}, \quad \alpha_{\text{upper}} = \frac{1 - f_i}{Q_i^{\max} - f_i}$$

where Q_i^{\max} , Q_i^{\min} are the maximum and minimum of Q_i within the region it is active. Again the Levenberg–Marquardt method is used to identify the maximum and minimum values. Note that using the smaller of the two α values means that *both* constraints are satisfied: choosing a lower α than required to satisfy a constraint will simply ‘flatten’ the function R_i more than is actually required, and R_i will lie well within the corresponding bound. As in Section 3.2, if any f_i equals 0 or 1, the corresponding $R_i(\mathbf{x})$ will be a constant (0 or 1 respectively). As before, $\beta = (1 - \alpha)f_i$.

We illustrate this technique on a test example from Lancaster and Salkauskas [21]. A function $S(x, y)$ is defined as:

$$S(x, y) = \begin{cases} 1.0 & \text{if } (y - x) \geq 0.5 \\ 2(y - x) & \text{if } 0.5 \geq (y - x) \geq 0.0 \\ \frac{\cos(4\pi r) + 1}{2} & \text{if } r \leq \frac{1}{4} \\ 0 & \text{otherwise} \end{cases}$$

where $r = \sqrt{(x - 1.5)^2 + (y - 0.5)^2}$.

The function $S(x, y)$ is shown in Figure 13. Note that it has areas where it is exactly zero, and a peak and upper shelf where it has a value of 1.0. A contour representation is also shown, as Figure 14, where we additionally show the 40 data points that were used to construct the test data.

We construct a [0, 1]-constraint test problem by evaluating $S(x, y)$ at a random set of 40 points, and requiring the interpolation scheme to reconstruct the function in such a way that it remains within the [0, 1] limits. A sequence of figures shows how the new method performs. In Figure 15, we show the surface recreated by the unconstrained MQS technique. It is clear that it goes below zero and above one, and this is

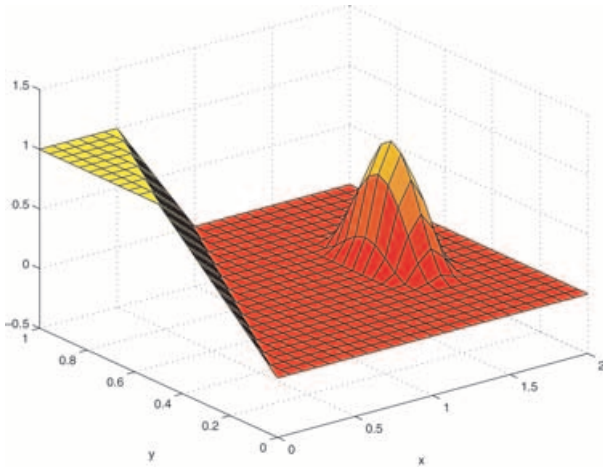


Figure 13: *Lancaster and Salkauskas function $S(x, y)$ —surface view.*

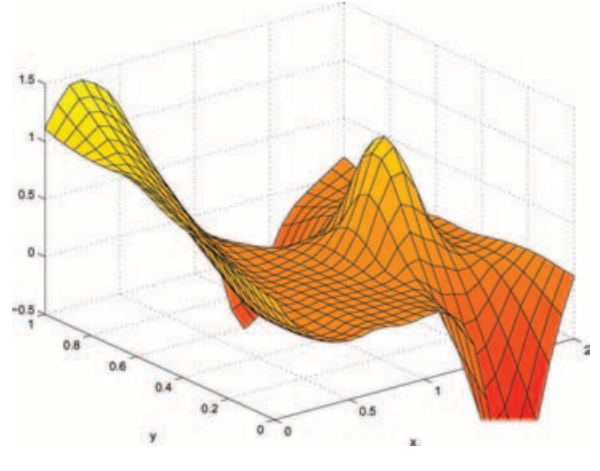


Figure 15: *Unconstrained reconstruction of Lancaster and Salkauskas function $S(x, y)$ —surface view.*

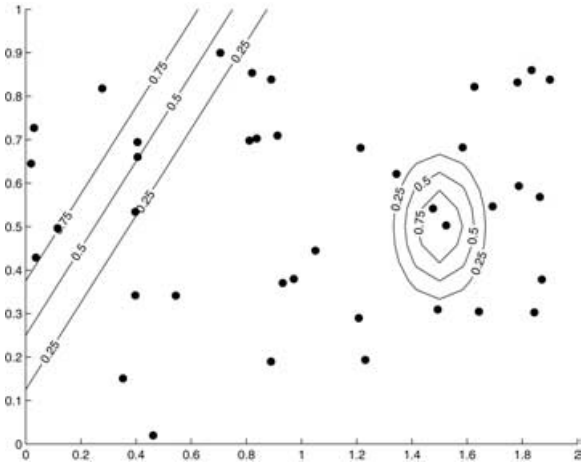


Figure 14: *Lancaster and Salkauskas function $S(x, y)$ —Contour map.*

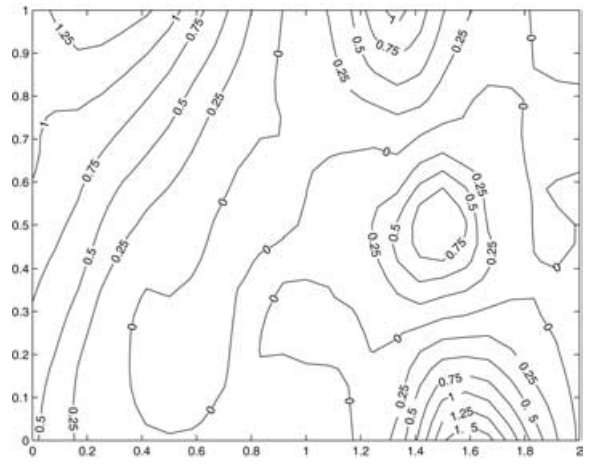


Figure 16: *Unconstrained reconstruction of Lancaster and Salkauskas function $S(x, y)$ —contour map.*

confirmed very clearly in the contour representation, shown in Figure 16.

By contrast, in Figure 17, we show the surface generated by the constrained method. It lies within the $[0, 1]$ limits, as is confirmed by the contour representation shown in Figure 18. Note that the flat plane with zero height, where many of the data values equal the constraint, is reproduced quite well by the algorithm. In this area, many of the R_i functions will be constant, equal to zero, and this enables a good reconstruction of the base plane.

4.3. Arbitrary Upper and Lower Bounds

Having solved the $[0, 1]$ constraint problem, it is then easy to solve the general problem of constructing an interpolant $F(\mathbf{x})$ subject to upper and lower bounds, that is,

$$A(\mathbf{x}) \geq F(\mathbf{x}) \geq B(\mathbf{x}). \tag{25}$$

To achieve this we create a new set of data values

$$t_i = \frac{f_i - B(\mathbf{x}_i)}{A(\mathbf{x}_i) - B(\mathbf{x}_i)}. \tag{26}$$

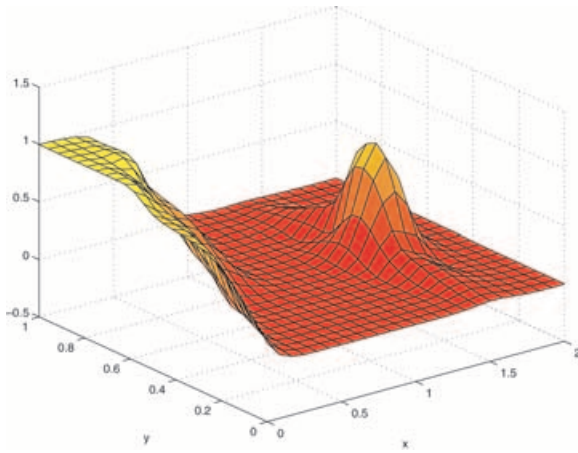


Figure 17: Constrained reconstruction of Lancaster and Salkauskas function $S(x, y)$ —surface view.

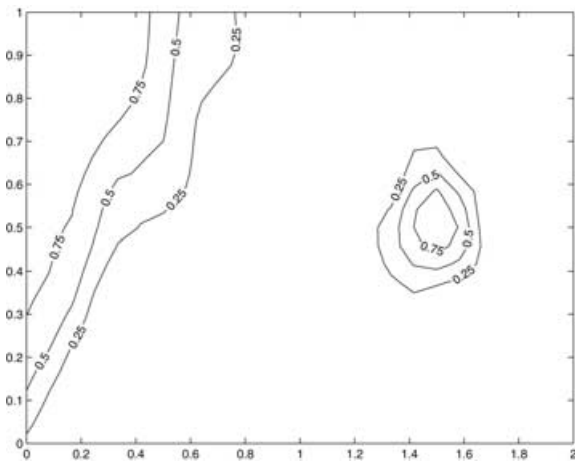


Figure 18: Constrained reconstruction of Lancaster and Salkauskas function $S(x, y)$ —contour map.

We construct a $[0,1]$ -constrained interpolant $T(\mathbf{x})$ to the data points (\mathbf{x}_i, t_i) , and then construct $F(\mathbf{x})$ as:

$$F(\mathbf{x}) = T(\mathbf{x})(A(\mathbf{x}) - B(\mathbf{x})) + B(\mathbf{x}). \quad (27)$$

Note that everything in this constrained section would apply to any interpolation method for which positivity, or $[0, 1]$ constraint, can be proved.

To illustrate the method we show a rather contrived example where we have defined data values randomly between upper and lower quadratic ‘bowls’, and required an interpolant to be created that keeps within the limits imposed by the bowls. The result is shown in Figure 19.

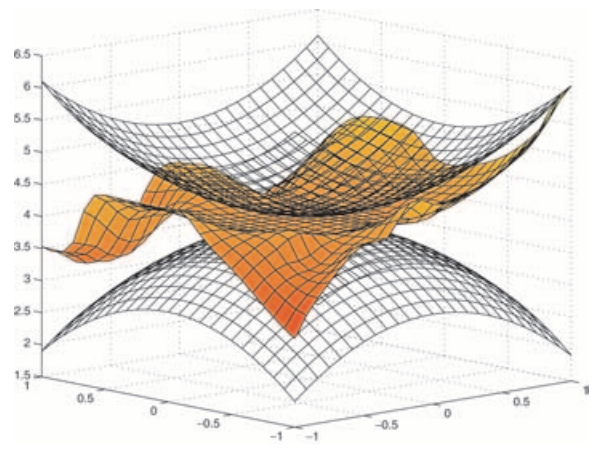


Figure 19: Surface between surfaces. In this figure, we show lower and upper bound quadratic surfaces as a wire-frame mesh, and the constructed interpolant as a shaded surface. The interpolant is constructed from data that are randomly located, with random values in the range between the lower and upper bounds.

5. Interpolation Quality

An important issue concerns the quality of the new interpolant, in comparison with the original, unconstrained interpolant. In the original method, a quadratic basis function is computed as interpolating the associated data point, and as the best weighted least-squares approximation to the other data points. In the constrained interpolant, we keep the interpolation property, and we keep the same shape—but we lose the best least-squares property when we apply the transformation. Moreover, in order to ensure positivity of the overall interpolant, we make sure each basis function is positive within its region of influence—a sufficient rather than necessary condition. Thus we may modify basis functions, losing the least-squares property, when it is not strictly necessary. It is reasonable to ask therefore how much we sacrifice the quality of interpolation in order to preserve positivity.

We evaluate the quality of the new interpolant by the following experiment. For the Lancaster and Salkauskas function we construct the unconstrained and constrained interpolants, based on a series of data sets where the number of randomly chosen points progressively increases. We measure the quality by evaluating the interpolants on a 25×25 grid of points, and calculating the RMS error between the exact and calculated values. The results are shown in Table 2. The column headed ‘Uncon RMS (con fails)’ shows the RMS error followed in parentheses by the number of evaluated points which fail the $[0, 1]$ constraint out of a total of 625. The column headed ‘Con RMS’ shows the RMS error for the constrained interpolant, where of course there are no evaluated points which fail the constraint.

Table 2: Interpolation quality for original unconstrained MQS, constrained MQS and blended MQS.

No. of data points	Uncon RMS (con fails)	Con RMS	Blend RMS
20	1.04 (211)	0.90	0.82
40	1.53 (259)	0.35	0.50
60	0.40 (211)	0.41	0.40
80	0.90 (214)	0.39	0.38
100	0.34 (190)	0.35	0.30
150	0.37 (197)	0.20	0.18
200	0.13 (170)	0.19	0.13
250	0.09 (149)	0.14	0.10
300	0.08 (171)	0.12	0.09
350	0.08 (133)	0.10	0.08
400	0.07 (133)	0.11	0.08

Consider the first three columns of the table. It is clear that for a small number of points, that is, relatively sparse data, the constrained interpolant is superior using RMS error as the criterion. We can see from the images in Figure 15 and Figure 17, that there are some points where the constraints are exceeded by a wide margin (these figures correspond to 40 data points). We find that for up to 150 points, the gain in overall quality from imposing the constraints generally offsets the loss from transforming the basis functions. However as the number of data points increases, then the quality of the unconstrained interpolant is better, and we can infer that the loss of the least-squares property starts to penalise the constrained version. (However, note that the unconstrained method still creates an unsatisfactory interpolant, in the sense that over 20% of evaluated points fail the constraints, in all the experiments we ran.)

The above analysis motivates a modified approach in which we retain more of the least-squares fitting property of the basis functions. Consider the $[0, 1]$ -constraint case which we have here (the other constraint cases are handled similarly). When the Levenberg–Marquardt algorithm determines that a basis function goes outside the $[0, 1]$ interval, rather than use the transformed function R_i directly, we create a basis function which is a blend of the unconstrained Q_i and the constrained R_i . The blend is computed as a linear combination:

$$(1.0 - \theta)Q_i(\mathbf{x}) + \theta R_i(\mathbf{x}). \tag{28}$$

Here, θ is chosen so that the unconstrained interpolant Q_i is selected ($\theta = 0$) for values of \mathbf{x} where $0.25 \leq Q_i \leq 0.75$ and the constrained interpolant R_i is progressively blended in (with θ increasing from 0 to 1) for $0.0 \leq Q_i \leq 0.25$ and $0.75 \leq Q_i \leq 1.0$, with finally $\theta = 1.0$ for $Q_i \leq 0.0$ and $Q_i \geq 1.0$. In this way, we use the unconstrained basis function in regions where it comfortably satisfies the constraints; we use

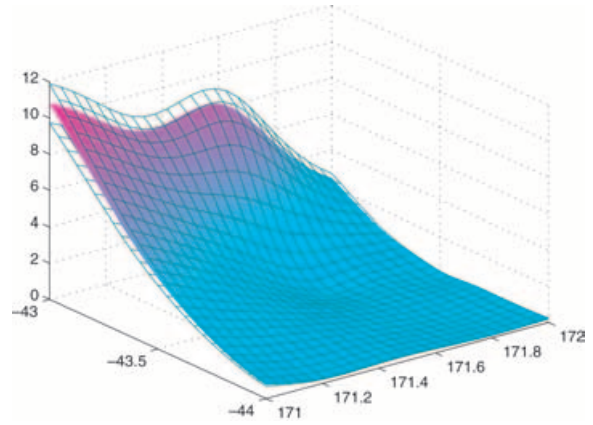


Figure 20: Error surfaces: Arthur's Pass rainfall data with 10% upper and lower error bound surfaces.

the constrained basis function where the unconstrained one fails; and we blend smoothly in the intermediate region. The final column of the table shows the success of the strategy: we achieve consistently good interpolation quality while still maintaining the constraints. Visually there is little perceptible difference in the two interpolants.

6. Conclusions and Future Work

We have shown how the modified quadratic Shepard method for interpolation of scattered data of any dimension, can be constrained to preserve positivity of the data. This has been demonstrated in examples in 1D, 2D and 3D. The method has also been adapted in order to constrain the interpolant within $[0, 1]$ limits, so that it can be used to interpolate fractional data. We have shown how the positivity and the $[0, 1]$ constraints can be generalised to any arbitrary functions as lower and upper bounds.

We would like to be able to extend this work to cover other approaches to data interpolation, such as RBFs, and to data approximation where we do not require the function to pass through the data points.

An interesting application of the work is the visualization of data subject to error, where we wish to show upper and lower bound visualizations that lie entirely above and entirely below the predicted 'surface'. If the error bounds for each data point are uniform, that is, the value at each data point $\mathbf{x}_i, i = 1, 2, \dots, N$ is within the range $[f_i - \delta, f_i + \delta]$, then for any interpolant we can form upper and lower error surfaces as $F(\mathbf{x}) \pm \delta$. A more interesting case, especially relevant to this paper, is when the error is expressed as a relative quantity. Suppose for positive data, f_i , there is a uniform relative error bound, that is, the value is within the range $[(1.0 - \delta)f_i, (1.0 + \delta)f_i]$. In this case, the values always lie in a positive range. If we use the methods described in this paper to construct a positive interpolant F , then the 'error' surfaces

$(1 \pm \delta)F$ are also positive and lie entirely above and below F . In Figure 20, we show these surfaces for the Arthur's Pass rainfall data (shown earlier in Figure 10 as a contour plot), where there is a 10% error bound associated with the data values. Further work is needed to improve the visual representation of error surfaces.

Acknowledgments

Ken Brodlić carried out much of this work during two sabbatical visits: one to Lincoln University, New Zealand; and the other to Caltech, California, USA. He is extremely grateful to both institutions for the warmth of their hospitality and the stimulating environment that allowed this work to be carried out. We are grateful to the referees for a number of suggestions that have improved this paper.

Rafiq Asim acknowledges the Government of Pakistan for providing funds to carry out his part of this research, at the University of Leeds.

References

1. S. K. Lodha and R. Franke. Scattered data techniques for surfaces. In *Scientific Visualization, Dagstuhl 97 Proceedings*. IEEE Computer Society Press, pp. 189–230, 2000.
2. M. Asim. *Visualization of data subject to positivity constraint*. PhD thesis, School of Computing, University of Leeds, 2000.
3. R. Barnhill G. Birkhoff and W. Gordon. Smooth interpolation in triangles. *Journal of Approximation Theory*, 8(1973), 114–128.
4. M. Asim and K. Brodlić. Curve drawing subject to positivity and more general constraints. *Computers and Graphics*, 27(2003), 469–485.
5. B. Ong and H. Wong. A c^1 positivity preserving scattered data interpolation scheme. In *Advanced Topics in Multivariate Approximation*. World Scientific, Singapore, pp. 259–274, 1996.
6. G. Nielson. The side-vertex method for interpolation in triangles. *Journal of Approximation Theory*, 25(1979), 318–336.
7. B. Mulansky and J. Schmidt. Powell-Sabin splines in range restricted interpolation of scattered data. *Computing*, 53(1994), 137–154.
8. E. Chan and B. Ong. Range restricted scattered data interpolation using convex combination of cubic Bezier triangles. *Journal of Computational and Applied Mathematics*, 136(2001), 135–147.
9. F. I. Utreras. Positive thin plate splines. *J. Approximation Theory and its Applications*, 1, 3 (1985), 77–108.
10. Y. Xiao and C. Woodbury. Constraining global interpolation methods for sparse data volume visualization. *International Journal of Computers and Applications*, 21, 2 (1999), 59–64.
11. D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 23rd National Conference* (New York, 1968), ACM, pp. 517–523.
12. R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *International Journal of Numerical Methods in Engineering*, 15(1980), 1691–1704.
13. R. Barnhill. Representation and approximation of surfaces. In *Mathematical Software III*, J. R. Rice (Ed.). Academic Press, New York, pp. 69–120, 1977.
14. G. M. Nielson. Scattered data modelling. *IEEE Computer Graphics and its Applications*, (1993), 60–70.
15. R. J. Renka. Algorithm 660: QSHEP2D: Quadratic Shepard method for bivariate interpolation of scattered data. *ACM Transactions on Mathematical Software* 14, 2 (1988), 149–150.
16. R. J. Renka. Multivariate interpolation of large sets of scattered data. *ACM Transactions on Mathematical Software*, 14, 2 (1988), 139–148.
17. NAG: NAG Library routine E01SGF, 2005. See <http://www.nag.co.uk>.
18. W. J. Gordon and J. A. Wixom. Shepard's method of metric interpolation to bivariate and multivariate interpolation. *Mathematics of Computation*, 32, 141 (1978), 253–264.
19. R. Fletcher. *Practical Methods of Optimization*. Wiley, Chichester and New York, 1987.
20. NIWA: New Zealand National Institute of Water and Atmospheric Research: Rainfall data, 2005. See <http://www.niwa.co.nz>.
21. P. Lancaster and K. Salkauskas. *Curve and Surface Fitting: An Introduction*. Academic Press, London, 1986.