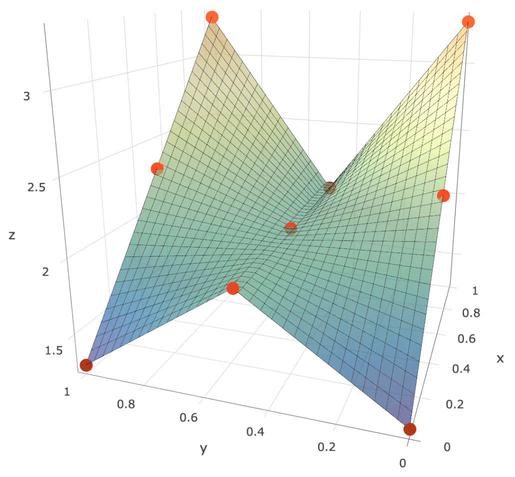
Report on Box Spline Interpolation and Approximation

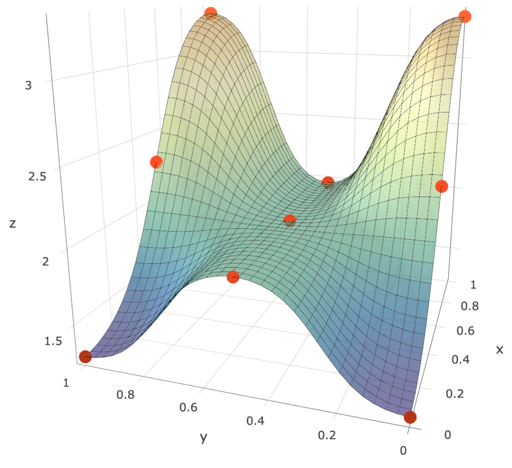
May 29th, 2017

Thomas Lux

Linear Box Mesh Interpolation



Quadratic Box Mesh Interpolation



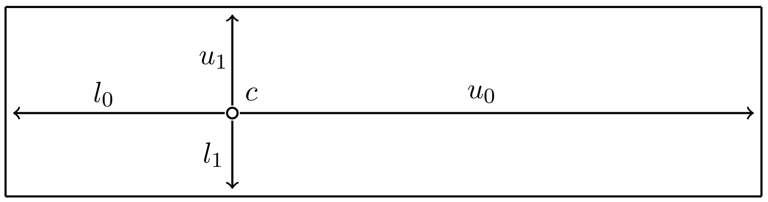
1 Introduction

This documents contains a brief explanation and analysis of the *Box Mesh* algorithm for generating an interpolating and/or approximating surface to data given a desired error tolerance and smoothness. The box mesh algorithm attempts to generate the smallest set of overlapping boxes with the largest minimum side length (least skinny) such that each box contains only one data point and the error at all data points is below a given tolerance. After constructing this set of boxes, we can use a variety of different functions to generate the weight of a control point within each individual box. Taking a weighted sum of control point values using these box functions generates the surfaces seen above.

First we discuss the specification of each individual box and analyze the use of linear and quadratic Box Splines as the box functions. Then, we outline the box mesh construction algorithm, providing a visual example. Following the example we discuss some mechanisms for smoothing the surface. Finally, we conclude with a runtime analysis of the box mesh algorithm.

2 Box Specification

A box mesh is composed of individual boxes. Each n-dimensional box has a *center* (c), *lower width* (l), and *upper width* (u). Given we are working in d dimensions, a point p is inside of a box b with center c, lower width l, and upper width u if $(\forall\ i \mid 0 \leq i < d)(c_i - l_i < p_i < c_i + u_i)$. See the following visualized example of a box in 2 dimensions.



Box Functions

Allowing for the conditional scaling of points, we can assume that every box is a unit cube with center $(\frac{1}{2}, \frac{1}{2})$. All of our box functions then only need to be defined on the unit cube in d dimensions. The two functions that we utilize are: the linear box spline defined by the direction vector set $[II]$ where I is the d-dimensional identity matrix; and the quadratic box spline defined by the direction vector set $[III]$. The convenience of restricting ourselves to box splines of this form is that the box spline value at any x is simply the multaplicative sum $\Pi_{i=0}^{d-1} f(x_i)$ where f is the piecewise polynomial that defines the associated linear or quadratic B-Spline in one dimension.

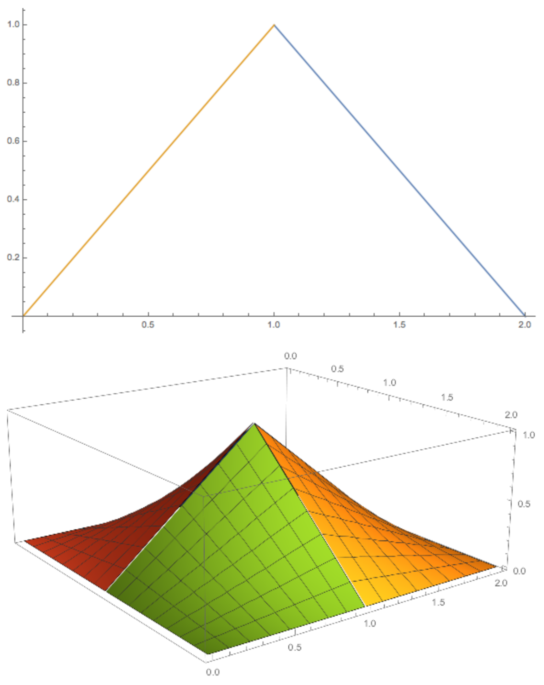
We chose these two box functions because the linear box spline is c_0 and the quadratic box spline is c_1. When normalizing any arbitrary box into the unit cube with center $(\frac{1}{2}, \frac{1}{2})$ it is necessary that values be scaled differently depending on which side of the center they are on. It is clear that scaling the x has no effect on the function value (we maintain c_0), and since the first derivative of any box spline at the center coordinate is 0 we do not break c_1 continuity with this non-uniform scaling. But, the continuity of any derivative that is non-zero at the center would lost with higher order box splines.

The output of the box functions will be used as the weight for the control point value at the center of each box. Thus, the response value associated with a given x is computed by taking:

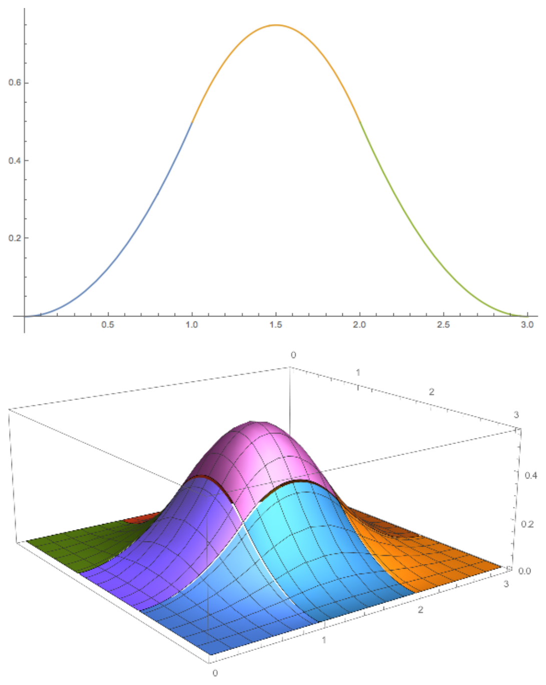
$$\frac{\sum_{i=0}^{\mathbf{b}-1} \mathbf{b}_i(x) \cdot \text{boxval}_i}{\sum_{i=0}^{\mathbf{b}} \mathbf{b}_i(x)}$$

where $\mathbf{b}$ is the set of boxes that contain x, $\mathbf{b}_i(x)$ is the value of the i^{th} box function evaluated at x, and boxval$_i$ is the response value associated with the i^{th} box.

Linear Box Splines in 1 and 2 Dimensions



Quadratic Box Splines in 1 and 2 Dimensions



3 Box Mesh Construction

Given a data set with N points in D dimensions, we want to construct a set of B boxes centered about B points from the data, referred to as control points. In order to create good response approximations, we want to ensure that the boxes are both reasonably shaped (not extremely skinny) and maximally overlapping. The box mesh construction algorithm creates a set of boxes that each meet these two criteria as well as contain strictly *one* control point. This final extra condition allows for strict interpolation.

The algorithm for the box mesh construction proceeds as follows:

```
Data: N by D+1 matrix of data where the last dimension is the response.
Result: B-box mesh
Initialize one box with upper and lower widths of infinity at median response point;
while max error > tolerance do
    Identify point $p$ with max error;
    Identify boxes $\mathbf{b}$ that contain $p$;
    Initialize box new with center $p$ that contains all boxes in $\mathbf{b}$;
    for $b$ in $\mathbf{b}$ do
        split dimension := $\max_i$ |b-center$_i$ − new-center$_i$|;
        Adjust the widths of $b$ and new in split dimension
        such that $b$ and new do not contain each other's center;
    end
end
```

Algorithm 1: Box Mesh Construction

Visual Example of Box Mesh Construction

The following 3 examples are all for x with dimension 2. The initial boxes seen on the left are the $\mathbf{b}$. These boxes contain the new point that will be added. The premise is that the new point has already been identified as the one with the most error in response. The resulting boxes seen on the right demonstrate the reshaping of the new and old boxes.

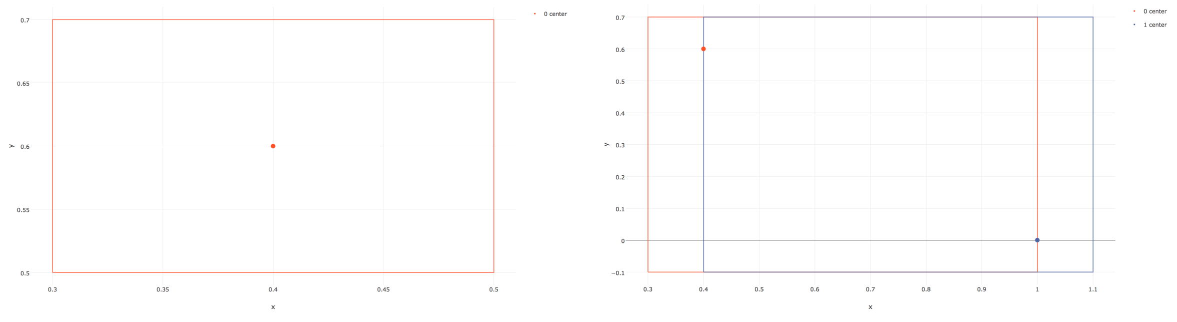


Table 1: Adding the blue point, reshaping the x-dimension of the orange box.

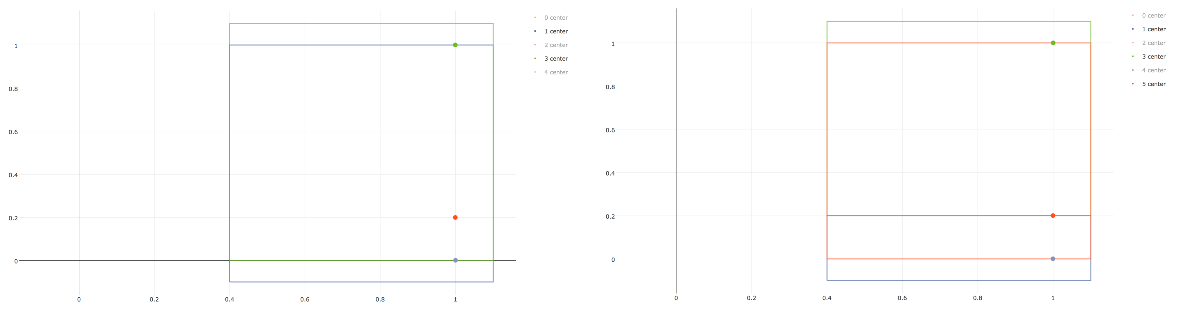


Table 2: Adding the orange point, reshaping the y-dimension of the blue and green boxes.

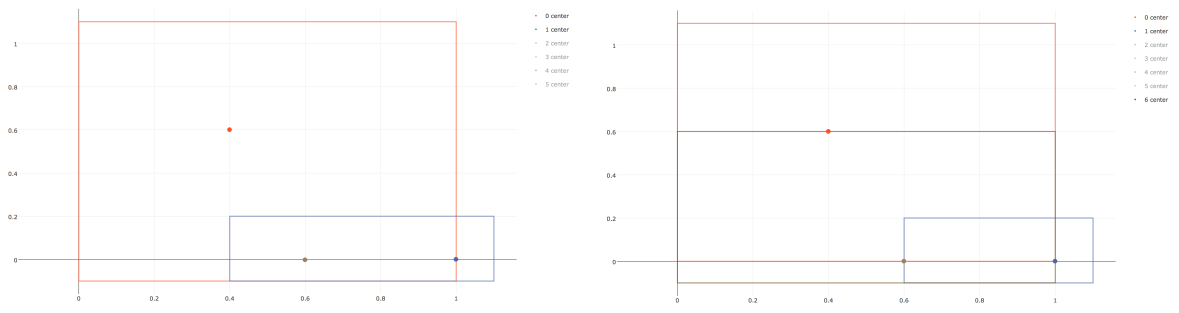


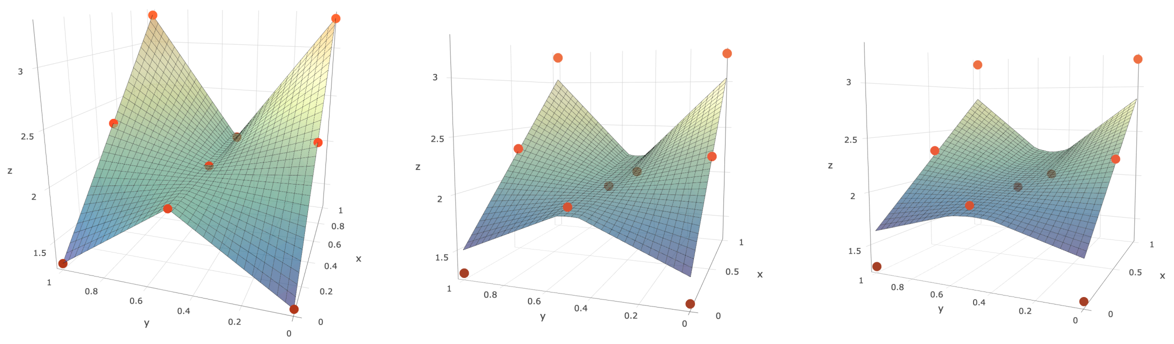
Table 3: Adding the brown point, reshaping the x dimension of the blue box and the y dimension of the orange box.

4 Smoothing the Box Mesh Surface

Given our box mesh as it is computed, we have created a strictly interpolating surface. It might be of interest to create a surface that approximates instead of interpolating in order to create greater smoothness and hence robustness to noise in the data.

Initially, we propose the use of a scaling factor s over the existing widths of each box as a smoothing parameter. An s equal to 1.0 produces the interpolatory surfaces mentioned before. As the smoothing paramater grows, neighboring boxes begin to influence each other more and thus the response estimates tend towards average values of larger neighborhoods of points.

Take note of what happens as we increase the smoothing paramater of our initial 3D surface plots.



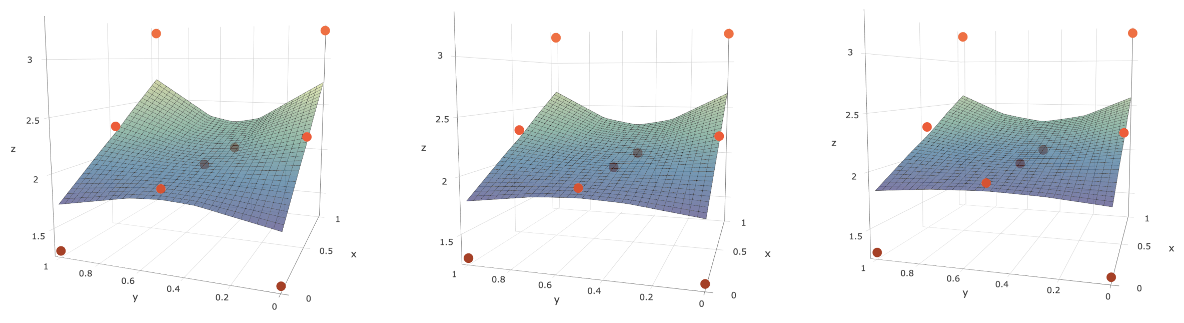


Table 4: Linear box mesh increasing smoothness from 1.0 to 1.5 by increments of 0.1

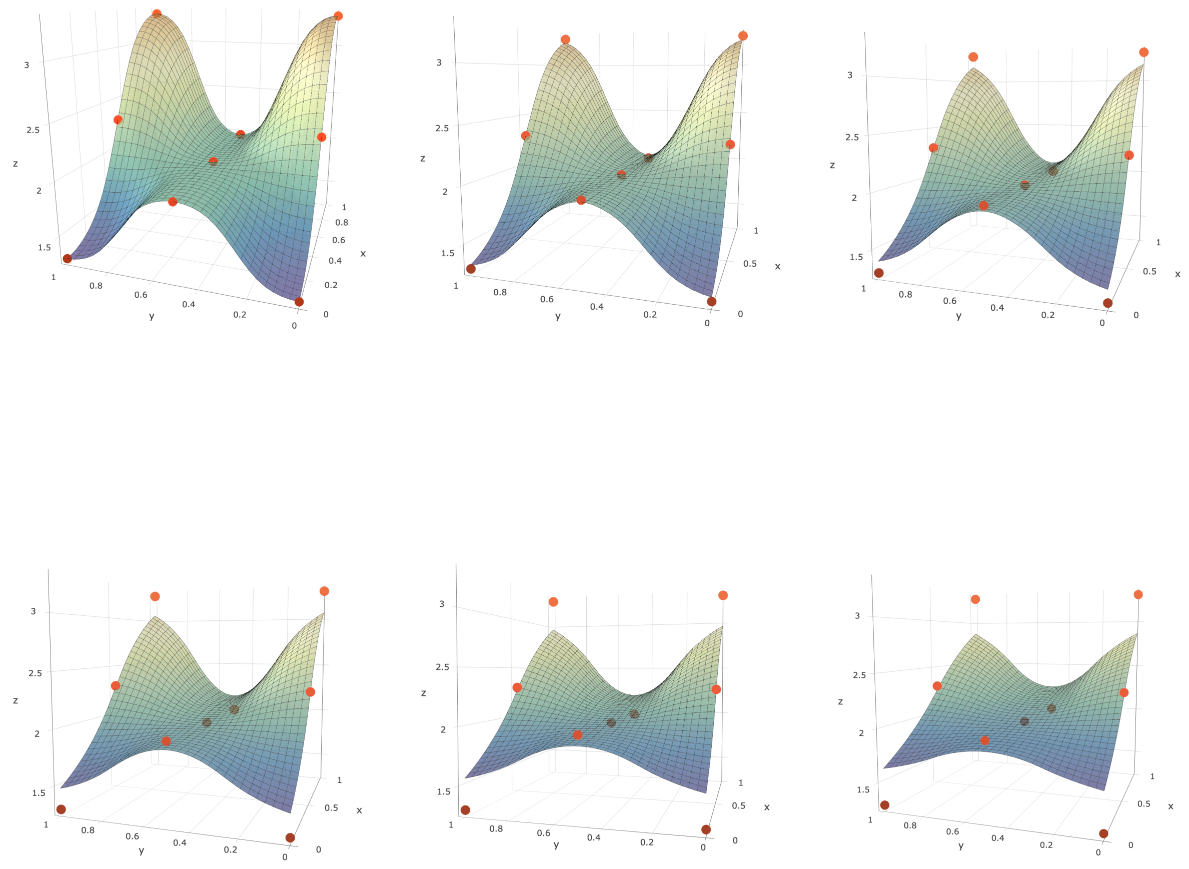


Table 5: Quadratic box mesh increasing smoothness from 1.0 to 1.5 by increments of 0.1

5 Runtime Analysis

The complexity of generating a box mesh is $O(N^2 + ND^2)$ with respect to the number of points N and the dimension of the data D. This is a worst case bound and further analysis needs to be done in order to determine (a) tighter and more accurate / insightful bound(s).

In the following analysis, B is the current number of boxes. Note that at each iteration, the number of boxes increases as more points from the data are designated control points.

Table 6: Runtime Analysis of Box Mesh construction

Initialize BM with single point		–
Check model error (at modified boxes)	(N-B) x 2D x D	
Add new point p to the Box Mesh		
1. Identify boxes b that contain p (at most 2D)	B x D	N
2. Get single bounding box for all boxes in b	2D x D	
3. Split largest dimension between box and boxes	2D x D	

6 Summary

Overall, the benefits of using the box mesh algorithm are its low computational complexity and flexibility in terms of interpolating and smoothing. Further testing needs to be done in order to better quantify the relative performance of the box mesh versus other interpolation and approximation techniques.

Potential Alternative Names:

– Incremental Box Spline Interpolation and Approximation

– Max Box Interpolation and Approximation with Box Splines

– Adaptive Box Partitioning for Interpolation and Approximation

– Adaptive Box Spline Regression

– Box Mesh Regression Interpolation and Approximation

– Adaptive Box Mesh

– Minimal Box Mesh

– Full Box Mesh

– Dynamic Box Mesh

– Constructive Box Mesh

– Incremental Box Mesh