Representation of Natural Terrain By Cubic L_1 Splines

John E. Lavery
David E. Gilsinn

Abstract. Cubic L_1 and L_2 interpolating splines based on C^1 smooth, piecewise cubic Sibson elements on a tensor-product grid are investigated. Computational tests were carried out for a 102.4 km by 102.4 km area of Fort Hood, Texas represented by a 1025×1025 set of 100-meter-spacing (posting) DTED1 terrain data obtained from the National Imagery and Mapping Agency. L_1 and L_2 interpolating splines were calculated for this area using data at coarser spacings of 800 m, 1600 m, 3200 m, 6400 m, 12800 m and 25600 m. The ℓ_1 and ℓ_2 errors of the L_1 spline for a given spacing are always smaller than the ℓ_1 and ℓ_2 errors of the L_2 spline for the same spacing. In half of the cases, the ℓ_∞ error of the L_1 spline is smaller than the ℓ_∞ error of the corresponding L_2 spline. In the other half of the cases, it is larger.

§1. Introduction

Recently, univariate and bivariate cubic L_1 interpolating splines, the coefficients of which are calculated by minimizing the L_1 norm of the second derivatives of the spline, have been developed [2, 4]. These splines preserve shape for smooth data as well as for data with abrupt changes in magnitude and spacing and for smooth sets of spline nodes as well as for those with abrupt changes in spacing. In the present paper, we investigate the multiresolution capabilities of L_1 splines on terrain elevation data for Fort Hood, Texas and compare these capabilities with those of conventional cubic L_2 splines.

§2. L_1 and L_2 Splines Based on Sibson Elements

The cubic L_1 and L_2 splines that will be used in the present paper are based on piecewise cubic C^1 Sibson elements on tensor-product grids. The tensor-product grids are given by strictly monotonic partitions $\{x_i\}_{i=0}^{I}$ and $\{y_j\}_{j=0}^{J}$ of the finite real intervals $[x_0, x_I]$ and $[y_0, y_J]$, respectively.

Trends in Approximation Theory
Kirill Kopotun, Tom Lyche, and Mike Neamtu (eds.), pp. 1–10.

ISBN 0-8265-xxxx-x.

To create Sibson elements, one proceeds as follows. One first divides each rectangle $(x_i, x_{i+1}) \times (y_j, y_{j+1})$ into four triangles by drawing the two diagonals of the rectangle. The Sibson element $z(x, y)$ in this rectangle is cubic in each of these four triangles, is C^1 on the two diagonals and is C^1 with the Sibson elements in the adjacent rectangles. The derivative $\partial z/\partial x$ of the Sibson element is linear in y along the edges $x = x_i,\ x_{i+1}$; the derivative $\partial z/\partial y$ is linear in x along the edges $y = y_j,\ y_{j+1}$. The Sibson element z in a rectangle is determined by the values of z, $\partial z/\partial x$ and $\partial z/\partial y$ at the corners of that rectangle as is described in [1, 4].

A cubic L_1 spline on the domain $D = [x_0, x_I] \times [y_0, y_J]$ is the surface $z = z(x, y)$ that minimizes

$$\iint_D \left[\left| \frac{\partial^2 z}{\partial x^2} \right| + 2 \left| \frac{\partial^2 z}{\partial x \partial y} \right| + \left| \frac{\partial^2 z}{\partial y^2} \right| \right] dx\, dy \, + \, \varepsilon \sum_{i=0}^{I} \sum_{j=0}^{J} \left[|z_{ij}^x| + |z_{ij}^y| \right] \tag{1}$$

over all Sibson-element surfaces z that interpolate the data

$$z_{ij} = z(x_i, y_j), \ \ i = 0, 1, \ldots, I, \ \ j = 0, 1, \ldots, J\,. \tag{2}$$

In expression (1), ε is a small positive "regularization" number that assists in making the L_1 spline coefficients unique. For further information about ε, see Sec. 3 of [4]. The cubic L_1 spline defined here is the same as the cubic L_1 spline of type A_2 defined in Sec. 3 of [4]. No boundary conditions are used here, although they could be added without changing the theory or computational procedure in any significant way.

A cubic L_2 spline (of type A_2) on the domain $D = [x_0, x_I] \times [y_0, y_J]$ is the surface $z = z(x, y)$ that minimizes

$$\iint_D \left[\left(\frac{\partial^2 z}{\partial x^2} \right)^2 + 4 \left(\frac{\partial^2 z}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 z}{\partial y^2} \right)^2 \right] dx\, dy \, + \, \varepsilon^2 \sum_{i=0}^{I} \sum_{j=0}^{J} \left[(z_{ij}^x)^2 + (z_{ij}^y)^2 \right] \tag{3}$$

over all Sibson-element surfaces z that interpolate the data (2). Expression (3) is the same as expression (1) except for the fact that the terms are measured in the squares of the L_2 and ℓ_2 norms rather than in the L_1 and ℓ_1 norms. The ε in expressions (1) and (3) are the same.

§3. Algorithm and Computational Examples

In Sec. 4 of [4], nonlinear and linear programming procedures suitable for minimizing functional (1) are described. The computational procedure adopted in these papers and in the present paper is to discretize the integral in (1) and to carry out the minimization by the primal affine method of Vanderbei, Meketon and Freedman [5, 6], which is described in detail in Sec. 4 of [4]. The integral in (1) as well as that in (3) was discretized in the following manner. Express the integral as the sum of the integrals over the rectangles $(x_i, x_{i+1}) \times (y_j, y_{j+1})$ of the tensor-product grid. Divide each rectangle into N^2 equal subrectangles, where $N \geq 2$. The integral over the rectangle is approximated by $1/[2N(N-1)]$ times the sum of the $2N(N-1)$ values of the integrand at the midpoints of the sides of the subrectangles that are in the interior of the rectangle.

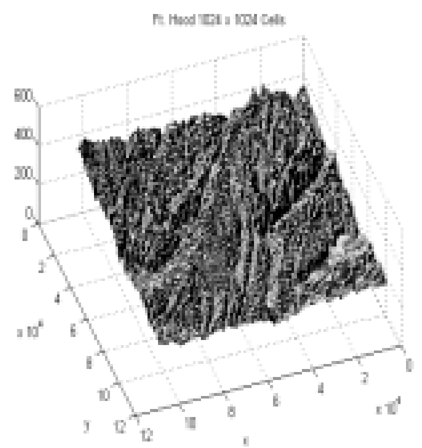

Fig. 1. Surface based on 100-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

Computational tests were carried out on a 1025×1025 set of terrain data that consists of the northwest 102.4 km by 102.4 km portion of a 1201×1201 set of 100-meter-spacing (posting) DTED1 digital elevation data for Fort Hood near Killeen, Texas. This data set was obtained from the Terrain Resource Repository of the Terrain Modeling Project Office (TMPO) on the WWW home page of the National Imagery and Mapping Agency (URL `http://www.nima.mil /geospatial/geospatial.html`). For all of these computational results, $N = 3$ and $\varepsilon = 10^{-4}/(2N(N-1))$.

In Fig. 1, we present the surface for the 102.4 km by 102.4 km, 100-meter-spacing subset of the Fort Hood data set mentioned above. This surface, which was plotted using bilinear elements, is a visual reference for the L_1 and L_2 splines presented below in Figs. 2–13. Figs. 2–13 were plotted using bilinear elements on 100 m by 100 m cells, with spline z values at the corners.

In the even numbered Figs. 2–12, we present for the 102.4 km by 102.4 km area of Fort Hood represented in Fig. 1 the cubic L_1 interpolating splines calculated on coarse spline grids at spacings (postings) of 800 m, 1600 m, 3200 m, 6400 m, 12800 m and 25600 m. We denote these splines by $z_{[L_1,800]}$, $z_{[L_1,1600]}$, $z_{[L_1,3200]}$, $z_{[L_1,6400]}$, $z_{[L_1,12800]}$ and $z_{[L_1,25600]}$, respectively. In the odd numbered Figs. 3–13, we present the cubic L_2 interpolating splines for the 102.4 km by 102.4 km area of Fort Hood represented in Figs. 2–12. The splines in Figs. 8–13 were calculated on coarse spline grids at spacings (postings) of 800 m, 1600 m, 3200 m, 6400 m, 12800 m and 25600 m. We denote these splines by $z_{[L_2,800]}$, $z_{[L_2,1600]}$, $z_{[L_2,3200]}$, $z_{[L_2,6400]}$, $z_{[L_2,12800]}$ and $z_{[L_2,25600]}$, respectively. We emphasize here that the splines of Figs. 2–7 are interpolating splines that use only the data at the given coarse spacings and completely ignore the presence of intermediate data points at lower, 100 m spacing.

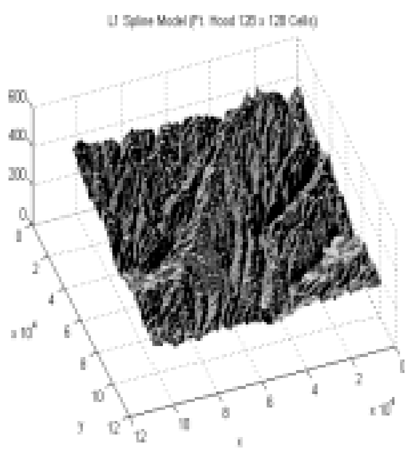

Fig. 2. L_1 spline $z_{[L_1,800]}$ based on 800-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

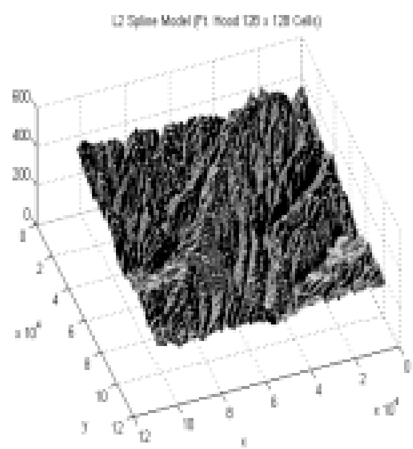

Fig. 3. L_2 spline $z_{[L_2,800]}$ based on 800-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

To measure the performance of the splines, we will use the following discrete norms calculated using the data at the original 1025^2 data points: 1) the (normalized) ℓ_1 norm $||\cdot||_{\ell_1}$ (sum of the absolute values of the 1025^2 points divided by 1025^2), 2) the (normalized) ℓ_2 norm $||\cdot||_{\ell_2}$, also known as the RMS or root-mean-square norm (square root of the quotient that consists of the sum of the squares of the 1025^2 points divided by 1025^2) and 3) the ℓ_∞ norm $||\cdot||_{\ell_\infty}$ (maximum absolute value of the 1025^2 points). In Table 1, we present the ℓ_1, ℓ_2 and ℓ_∞ norms of the error between the L_1 splines and the original set of 1025^2 data points.

Table 1. Norms of differences between cubic L_1 splines on coarse grids and original data

spacing s	=	800	1600	3200	6400	12800	25600

$\|\|z_{[L_1,s]} - \text{data}\|\|_{\ell_1} =$	2.380	4.517	7.512	12.03	16.82	26.48
$\|\|z_{[L_1,s]} - \text{data}\|\|_{\ell_2} =$	3.766	6.636	10.46	16.07	22.37	35.07
$\|\|z_{[L_1,s]} - \text{data}\|\|_{\ell_\infty} =$	63.41	70.42	73.22	91.72	104.4	122.0

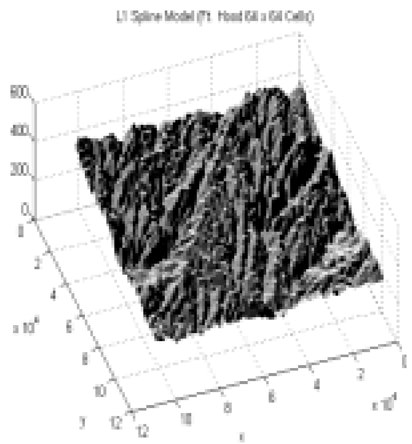

Fig. 4. L_1 spline $z_{[L_1,1600]}$ based on 1600-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

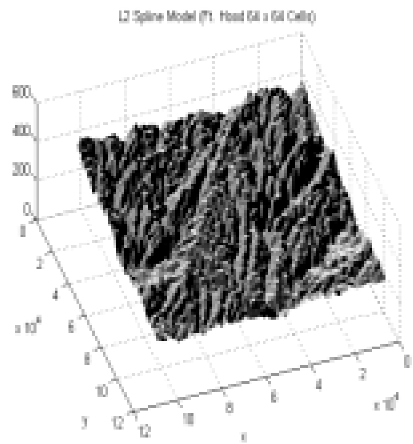

Fig. 5. L_2 spline $z_{[L_2,1600]}$ based on 1600-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

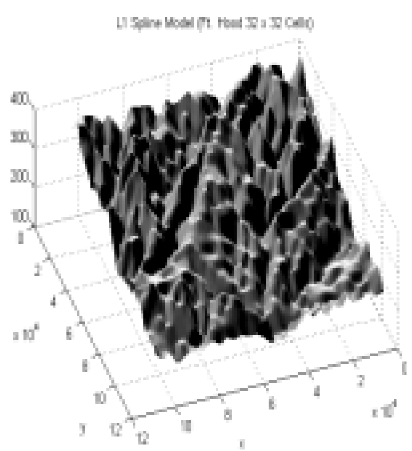

Fig. 6. L_1 spline $z_{[L_1,3200]}$ based on 3200-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

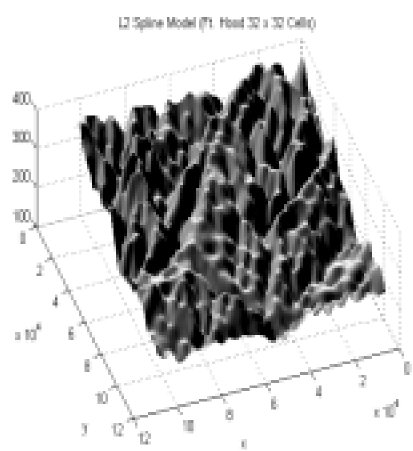

Fig. 7. L_2 spline $z_{[L_2,3200]}$ based on 3200-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

In Table 2, we present the ℓ_1, ℓ_2 norms and ℓ_∞ norm of the error between these L_2 splines and the original set of 1025^2 data points.

Table 2. Norms of differences between cubic L_2 splines on coarse grids and original data

spacing s =	800	1600	3200	6400	12800	25600
$\lVert z_{[L_2,s]} - \text{data}\rVert_{\ell_1} =$	2.408	4.568	7.537	12.04	16.91	26.77
$\lVert z_{[L_2,s]} - \text{data}\rVert_{\ell_2} =$	3.771	6.694	10.52	16.10	22.51	35.37
$\lVert z_{[L_2,s]} - \text{data}\rVert_{\ell_\infty} =$	61.76	70.66	73.38	87.97	106.9	120.4

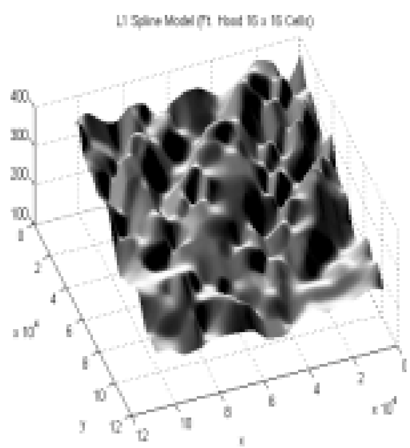

Fig. 8. L_1 spline $z_{[L_1,6400]}$ based on 6400-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

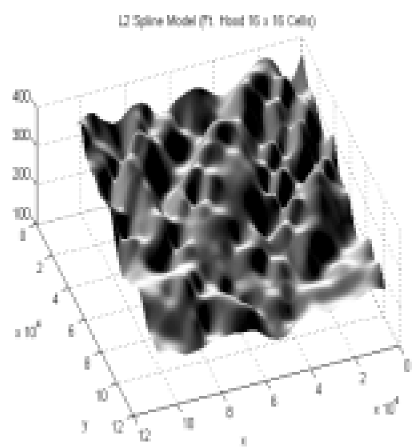

Fig. 9. L_2 spline $z_{[L_2,6400]}$ based on 6400-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

By careful visual inspection of the figures, one can see differences in the L_1 and L_2 splines for the same spacing. These differences consist mainly of additional oscillation in the L_2 splines. However, one is not able to determine by visual inspection which type of spline, L_1 or L_2, is more accurate. Some information about the accuracy can be gathered from the norms of the errors in Tables 1 and 2. In these tables, the ℓ_1 and ℓ_2 errors of the L_1 spline for a given spacing are always smaller than the ℓ_1 and ℓ_2 errors of the L_2 spline for the same spacing. In three cases, the ℓ_∞ error of the L_1 spline is smaller than the ℓ_∞ error of the corresponding L_2 spline. In the other three cases, it is larger.

In Table 3 we present estimated processing times for the interpolating spline runs that produced the images in Figs. 2–13. Any processing times less than one second are reported as one second since the time function used in the interpolation program did not return any time less than one second.

It should be noted that the interior point algorithm used always produced the L_2 interpolation coefficients at the first iteration of the algorithm. The L_1 interpolation coefficients for all of the runs were produced in the range of between nineteen and thirty iterations.

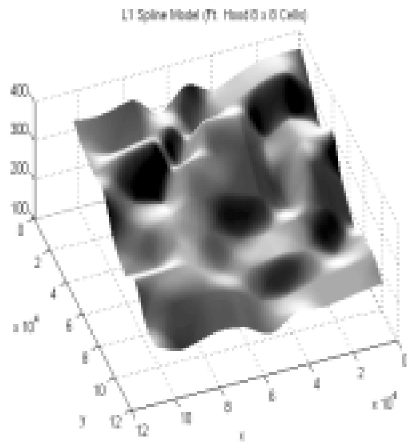

Fig. 10. L_1 spline $z_{[L_1,12800]}$ based on 12800-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

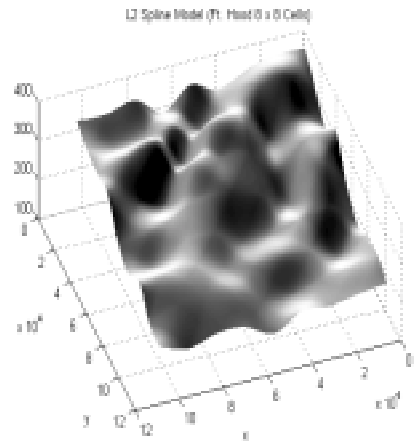

Fig. 11. L_2 spline $z_{[L_2,12800]}$ based on 12800-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

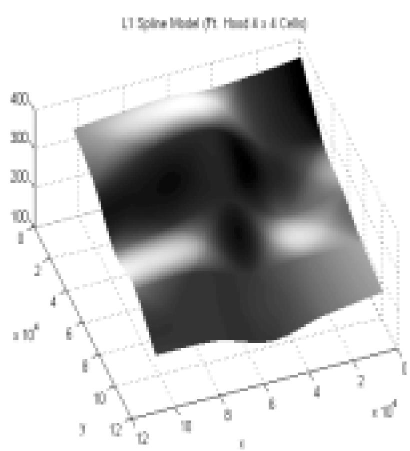

Fig. 12. L_1 spline $z_{[L_1,25600]}$ based on 25600-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

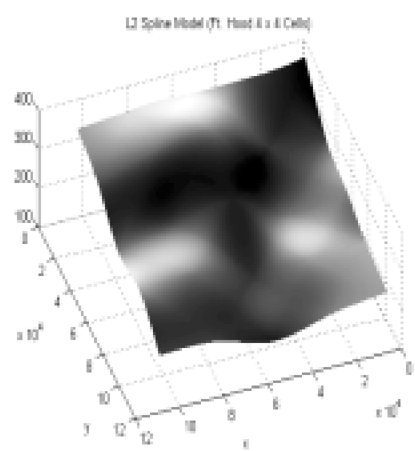

Fig. 13. L_2 spline $z_{[L_2,25600]}$ based on 25600-meter-spacing data for 102.4 km by 102.4 km area of Fort Hood, Texas

Table 3. Estimated Processing time, in seconds for computing L_1 and L_2 interpolating splines on coarse grids. The computer was a 933 MHz PC workstation with 1.5 GB of RAM.

spacing s	$=$	800	1600	3200	6400	12800	25600
$z_{[L_1,s]}$	$=$	2240	102	1	1	1	1
$z_{[L_2,s]}$	$=$	76	7	1	1	1	1

§4. Conclusion

It was noted in [2, 4] that cubic L_1 interpolating splines preserve shape much better than cubic L_2 interpolating splines. The results of the present paper indicate that more evidence is needed before final conclusions about the relative performance of cubic L_1 and L_2 interpolating splines for irregular natural terrain can be made.

One unsolved issue that will be a large factor in further investigations is the metric in which the error should be measured. The ℓ_1, ℓ_2 and ℓ_∞ norms are widely used to measure shape preservation. However, it is well known that the magnitudes of these norms do not correspond well to degrees of shape preservation as perceived by most observers. Alternatives such as the BV norm (norm of the space of functions of bounded variation) also do not express well what human beings understand by shape preservation. Shape preservation is not yet quantitatively understood. A closely related issue is determining the function spaces or classes to which terrain surfaces belong. Those spaces or classes, which may be different for different types of surfaces (for example, natural terrain and urban terrain) and for different human uses, are still unknown in spite of many efforts in the past to characterize terrain using classical measures of smoothness, fractal dimensions and other techniques. It is likely that theoretical justification of the advantages of L_1 splines will go hand in hand with quantification of the concept of shape preservation and with clarification of the function spaces or classes to which various terrain surfaces belong.

In the present paper, we have investigated the behavior of cubic L_1 interpolating splines. However, when the spline grid is coarser than the data grid, smoothing splines, which approximate rather than interpolate, are often a more appropriate tool than interpolating splines. The authors will soon carry out computational tests for L_1 smoothing splines on large terrain data sets. These smoothing splines will be bivariate extensions of the L_1 smoothing splines introduced in [3].

Acknowledgments. David Gilsinn's research was supported in part by the Army Model and Simulation Office under the Army Model Improvement Program and by the Defense Modeling and Simulation Office under the Modeling and Simulation Initiative.

References

1. Han, L. and L. L. Schumaker, Fitting monotone surfaces to scattered data using C^1 piecewise cubics, SIAM J. Numer. Anal. **34** (1997), 569–585.
2. Lavery, J. E., Univariate cubic L_p splines and shape-preserving, multiscale interpolation by univariate cubic L_1 splines, Comput. Aided Geom. Design **17** (2000), 319–336.
3. Lavery, J. E., Shape-preserving, multiscale fitting of univariate data by cubic L_1 smoothing splines, Comput. Aided Geom. Design **17** (2000), 715–727.
4. Lavery, J. E., Shape-preserving, multiscale interpolation by bi- and multivariate cubic L_1 splines, submitted to Comput. Aided Geom. Design.
5. Vanderbei, R. J., Affine-scaling for linear programs with free variables, Mathematical Programming **43** (1989), 31–44.
6. Vanderbei, R. J., M. J. Meketon and B. A. Freedman, A modification of Karmarkar's linear programming algorithm, Algorithmica **1** (1986), 395–407.

John E. Lavery
Computing and Information Sciences Division
Army Research Office
Army Research Laboratory
P.O. Box 12211
Research Triangle Park, NC 27709-2211
`lavery@arl.aro.army.mil`

David E. Gilsinn
National Institute of Standards and Technology
100 Bureau Drive, Stop 8220
Gaithersburg, MD 20899-8220
`dgilsinn@nist.gov`