Optimising the degree of data smoothing for locally adaptive finite element bivariate smoothing splines

M.F. Hutchinson*

(Received 7 August 2000)

Abstract

Finite difference and finite element schemes for bivariate thin plate smoothing splines are described. Nested grid SOR iterative methods are known to be able to solve these systems efficiently for large data sets. An iterative Newton procedure for optimising the smoothing parameter to achieve a prescribed residual sum of squares from the data

*Centre for Resource and Environmental Studies, Australian National University, Canberra ACT 0200, AUSTRALIA. mailto:hutch@cres.anu.edu.au

is obtained. It can be added to the SOR iteration with little additional computational cost and is demonstrated on test data to work for a wide range of smoothing parameters. An apparently more accurate version of this procedure, which requires more memory, converges slightly less quickly than the simpler approximation. The simpler method appears to be directly compatible with the SOR iterative method. The Newton method is shown to also work for locally adaptive versions of finite difference smoothing splines. The roughness penalty can be made locally adaptive to respect process-based constraints, such as minimum profile curvature, which depends on the local aspect of the fitted surface. This can be applied to the interpolation of digital elevation models. The weighted residual sum of squares can be made locally adaptive to allow for positional error in data, whether arising from actual data error, or from a finite difference discretisation. This has given rise to an objective method for optimising the grid resolution to the information content of the data.

Contents

1 Introduction

Finite difference and finite element discretisations, based on a regular two-dimensional grid, provide a means of calculating close approximations to bivariate thin plate smoothing splines fitted to scattered point data. Efficient methods based on a simple nested grid SOR iteration, in which the solution is progressively refined on successive finer grids, have been developed by Inoue [15], Hutchinson [10] and Smith and Wessel [17]. A recent variant on these methods has been developed by Hegland *et al.* [9]. All of these methods

originate from the non-nested grid solution formulated by Briggs [2]. Though sharp results on rates of convergence for nested grid implementations have yet to be obtained, the finite difference method developed by Hutchinson [10] has been shown in practice to have optimal computational cost, in the sense that it is proportional to the number of grid points. It has been routinely applied on standard workstations to problems involving millions of data points and grid points. Such problems are well beyond the means of standard analytic methods for thin plate smoothing splines.

Data smoothing is usually required to allow for data error and for error in the bivariate spline model. Objective methods for setting the degree of smoothing, are difficult to apply for larger data sets. This paper describes two computationally efficient procedures that can be applied to nested iterative methods to adaptively determine the smoothing parameter, so that the fitted bivariate spline has a prescribed residual sum of squares from the data. Both procedures use a Newton iteration scheme that simultaneously updates the smoothing parameter as the smoothing spline is solved for each grid resolution. The methods have similar rates of convergence. They provide a direct counterpart to the Newton procedure for the univariate polynomial smoothing spline obtained by Reinsch [16].

Prescribing the residual sum of squares is appropriate when the data errors are known and are known to dominate the errors in the spline model. When this is not the case, the degree of smoothing is better set by minimising the generalised cross validation [3], although this is also difficult for large data sets. The stochastic trace estimators obtained by Girard [6] and

Hutchinson [11] provide a means for addressing this problem, but cross validation methods are not discussed further here. Applications of the minimum variance trace estimator of Hutchinson [11] to a variety of problems have been described by Golub and von Matt [7].

A particular advantage of finite difference and finite element formulations of smoothing splines is that locally adaptive constraints can be applied to respect known process-based conditions. Such constraints cannot be easily imposed by existing analytic methods. Locally adaptive constraints can be applied to the two components defining the smoothing spline, the roughness penalty and the weighted residual sum of squares. Locally adaptive schemes impose additional computational cost in updating the equations as the solution is generated. These could also affect the proposed Newton procedure, but if the locally adaptive behaviour is fairly stable, its influence on the Newton procedure is minimal. This is verified by applying the Newton procedure to examples that include a locally adaptive roughness penalty and a locally adaptive weighting of the residual sum of squares.

2 The basic finite element formulation and its iterative solution

The data model for which bivariate spline smoothing is appropriate is that there are N noisy data values given by

$$z_i = f(x_i, y_i) + w_i \epsilon_i \quad (i = 1, ..., N) \tag{1}$$

where f is an unknown bivariate function, each ϵ_i is an independent sample from a zero mean random variable with common variance σ^2 and each w_i is a known positive constant. Suppose that f is represented by a regular grid of coefficients, given by a vector u, over a region of the plane containing the points (x_i, y_i). These coefficients may be a regular grid of values of the unknown function f. More generally they denote coefficients of a finite element representation of f. In the interests of computational efficiency, these elements normally have minimal local support, such as that afforded by bilinear or biquadratic polynomial B-splines, as defined by de Boor [4]. Various locally supported non-conforming finite element and finite difference schemes have been devised [1].

Using such representations, the vector of function values $f(x_i, y_i)$ may be written as Pu where P is a sparse $N \times M$ matrix and M is the total number of grid coefficients. When B-spline representations are used, each row of P has one, two or three non-zero entries, depending respectively on whether f is represented by piece-wise constants, or bilinear or biquadratic

polynomial B-splines. An approximation to a thin plate smoothing spline fit to the data in equation (1) may then be determined by finding the vector u that minimises

$$\|W^{-1}(Pu - z)\|^2 + \lambda u^T A u \tag{2}$$

where $z = (z_1, \ldots, z_N)^T$, $W = \mathrm{diag}(w_1, \ldots, w_N)$, A is a sparse symmetric positive semi-definite matrix corresponding to the total curvature of the function f, and λ is a positive smoothing parameter.

Differentiating expression (2) with respect to the vector u gives rise to a sparse, positive definite, system of equations for u given by

$$(P^T V P + \lambda A)u = P^T V z \tag{3}$$

where $V = W^{-2}$. For a given value of the smoothing parameter λ, this system can be solved in a nested grid iteration, proceeding from a suitably coarse initial grid to successively finer grids, using SOR or conjugate gradient iteration at each grid resolution. The SOR method has been adopted here because of its simplicity and suitability for adaptive enhancement. Though it is difficult to determine the optimal relaxation parameter for the SOR method analytically [8], a relaxation parameter of 1.6 has been found in practise to provide significant acceleration of convergence across a range of smoothing parameters. Provided the starting grid is chosen to be sufficiently coarse, 30 SOR iterations at each grid resolution are normally sufficient to ensure convergence [10]. This implies that the computational cost for the procedure is optimal, in the sense that it is essentially proportional to the final

number of grid points, since iteration on the final grid resolution dominates the computation.

Computational efficiency can be enhanced by using parallel vector operations on each row of the grid of coefficients in u. Vector operations may be applied to all contributions to the basic SOR step by all grid coefficients in u that are not in the current row, as well as grid coefficients in the same row that lie ahead of the SOR iteration. If A corresponds to the usual 13 point stencil associated with the biharmonic equation [2], then 10 of these 13 values can be accounted for using vector operations. The same vector strategy can be applied to the contributions from the matrix P^TVP.

Computer memory requirements in solving equation (3) may be minimised by noting that the matrix A need not be stored, as its entries are normally known. The memory requirements of the matrix P^TVP depend on the order of the finite element or finite difference scheme used to represent the function f. If a piece-wise constant finite difference scheme is used, then P^TVP is in fact a diagonal matrix, which can be stored in an array with the same number of entries as the vector u.

Higher order finite elements require significant additional storage. These memory costs can be avoided if the data points are stored off-line, in the same sequential order that the corresponding grid coefficients are accessed in the SOR iteration. This entails the additional computational cost of recalculation of the matrix P^TVP and the vector P^TVz. Since the number of data points is usually much less than the number of grid points, the extra cost can be

small compared to that of the basic SOR iteration.

3 Optimising the residual sum of squares

Two Newton procedures for optimising the weighted residual sum of squares are described. These depend on differentiating with respect to the smoothing parameter λ the weighted residual sum of squares, given by

$$R = \|W^{-1}Pu - W^{-1}z\|^2 = u^T P^T V P u - 2u^T P^T V z + z^T V z. \tag{4}$$

Using the chain rule gives

$$\frac{dR}{d\lambda} = \frac{dR}{du} \cdot \frac{du}{d\lambda} = 2v^T \frac{du}{d\lambda} \tag{5}$$

where

$$v = P^T V P u - P^T V z = P^T V (Pu - z). \tag{6}$$

Differentiating equation (3) implicitly with respect to λ and re-using equation (3) gives

$$(P^T V P + \lambda A)\frac{du}{d\lambda} = -Au = v/\lambda. \tag{7}$$

Setting $\lambda = e^{\theta}$ then gives

$$(P^T V P + \lambda A)\frac{du}{d\theta} = v. \tag{8}$$

Thus $du/d\theta$ satisfies the same system of equations as u, but with the data vector z replaced by the vector $Pu - z$. Moreover, from equations (5, 8) it follows that

$$\frac{dR}{d\theta} = 2v^T(P^T V P + \lambda A)^{-1}v. \tag{9}$$

Thus a simple Newton scheme can be used to achieve a prescribed weighted residual sum of squares S with increments in θ given by

$$\Delta\theta = (S - R)/\frac{dR}{d\theta}. \tag{10}$$

This scheme ensures the positivity of λ and also permits λ to take on a large range of values. It must be adapted to the nested grid SOR iterative scheme employed here. Equation (5) is valid for any value of u, but if u is an approximate solution to equation (3), then equations (7, 8, 9) also hold approximately.

If Gauss-Seidel iteration, is used to update u, then each scalar equation in (3) holds temporarily as each grid value in the vector u is updated. Applying implicit differentiation with respect to λ to this equation yields each corresponding scalar equation in (8). It is therefore valid to update values of $du/d\theta$ using the same Gauss- Seidel procedure.

3.1 First Newton scheme

It remains to investigate options for efficient approximate solution of equations (8, 9). The first option is to subject equation (8) to the same SOR iterative scheme used to solve for u. This should give accurate results, but at the expense of allocating memory to store the vector $du/d\theta$. However, it is not expensive computationally, as the equation coefficients and the entries in the vectors P^TVPu and P^TVz, which make up the right hand side of equation (8), are all available as part of the SOR iteration on u.

3.2 Second Newton scheme

The simplest alternative is to approximately solve equation (9) directly by replacing $(P^TVP + \lambda A)$ by its diagonal elements. This is equivalent to differentiating the basic Gauss-Seidel step in the solution of u while treating all other elements of u as constant. Since the diagonal elements of $(P^TVP+\lambda A)$ are positive, this maintains the positivity of the derivative with respect to θ in equation (9), so that the increments $\Delta\theta$ in the Newton iteration have the correct sign. This estimate is likely to be accurate if λ is relatively small and the matrix $(P^TVP + \lambda A)$ is diagonally dominant. This is indeed the case if there is little data smoothing and piece-wise constants are used to represent the function f, in which case P^TVP is diagonal. This approximation has been used effectively in the method described by Hutchinson [10].

4 Locally adaptive roughness penalties

Many locally adaptive roughness penalties are possible. Two that have direct relevance to digital elevation modelling are described. A common criticism of minimum curvature interpolation is that it cannot match sharp changes in gradients, particularly at peaks. The first locally adaptive penalty simply modifies the curvature roughness penalty by removing, or significantly reducing, the finite difference or finite element contribution to the roughness penalty at each peak. This can be readily implemented in an iterative finite difference framework.

The second, perhaps more generic, roughness penalty is profile curvature in the downslope direction. Minimising profile curvature should be reasonably consistent with fluvial landforming processes, and certainly consistent with normal river profiles. Departures from linear profiles due to strong underlying controls, such as rock outcrops, cliffs and waterfalls, can be accounted for in the interpolation process if these features are sampled by data points. This penalty has been posed by Hutchinson [12], and preliminary implementations are encouraging. The penalty is based on minimising the integral of the square of

$$\frac{d^2 f}{d\alpha^2} = \cos^2(\alpha)\frac{d^2 f}{dx^2} + 2\cos(\alpha)\sin(\alpha)\frac{d^2 f}{dxdy} + \sin^2(\alpha)\frac{d^2 f}{dy^2} \qquad (11)$$

where α is the local aspect angle. Aspect is defined at each grid point as the direction of steepest slope. This penalty is stable enough to support the iterative interpolation process provided the aspect angle is suitably stable as the

iteration proceeds. At peaks, where the aspect angle is not defined, there is no penalty, in accord with the first locally adaptive penalty described above. Other breakpoints in the land surface, such as cliffs, can be similarly accommodated. This penalty is also readily defined in terms of finite differences of the grid coefficients, although optimal strategies are still under development.

5 Locally adaptive weighting of the residual sum of squares

If a finite difference interpolation strategy is employed, data points are normally allocated to the nearest grid point. This introduces a small positional error in the data which may be interpreted as a vertical error in the point placed at the grid point. This is illustrated in Figure 1 where the data point A on a sloping terrain surface is allocated to the centre of the grid cell of width h, leading to a horizontal displacement d, which gives rise to a vertical error z.

The size of this vertical error depends on the slope s of the grid cell, and the magnitude of the horizontal displacement d. If it is assumed that each data point is placed randomly within its associated grid cell, then the standard deviation of the corresponding vertical error is readily shown to be

$$w = h.s/\sqrt{12}\,. \tag{12}$$

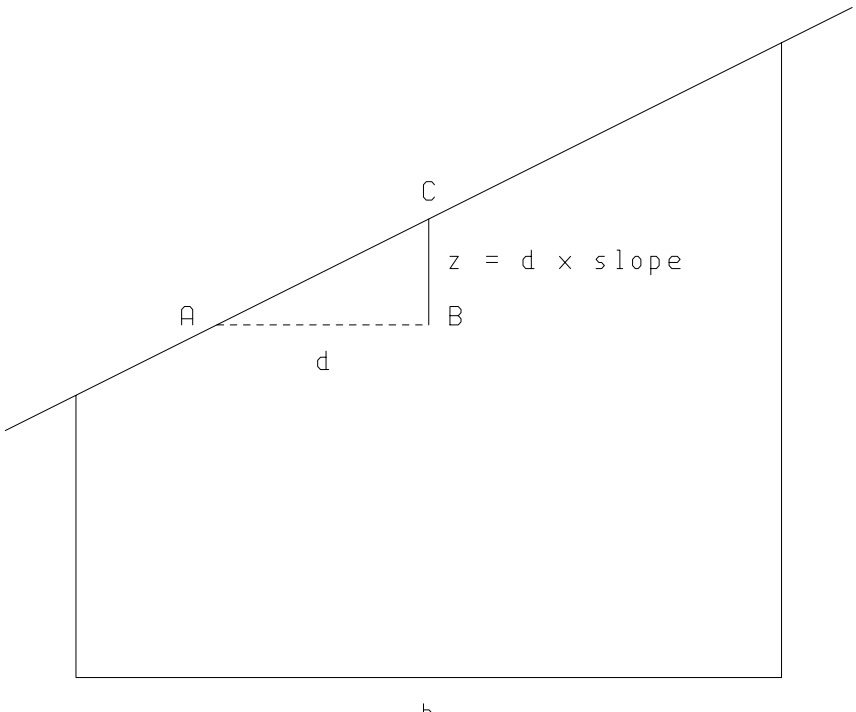

FIGURE 1: Vertical error of a horizontally displaced data point.

The amount of data smoothing can then be determined in a locally adaptive manner by weighting the data points, as in equation (2), according to these standard deviations, and determining the smoothing parameter so that

$$\|W^{-1}Pu - W^{-1}z\|^2 = N \tag{13}$$

where N is the number of data points. This matches the expected value of the weighted residual sum of squares from the true terrain surface. Provided the slopes of the surface are sufficiently stable as the iteration proceeds, this can be achieved using either of the Newton procedures described above. This has been implemented in the ANUDEM elevation gridding procedure [13]. It can be extended to the case where the data points have significant positional error, provided the variance of this error is known.

A significant byproduct of this approach is an objective procedure for optimising grid resolution to the information content of the source data. As the nested grid iteration proceeds from coarse to fine resolution by successively halving the grid spacing, the slopes of the fitted grid at the data points tend to steadily increase in magnitude. At coarse resolutions several data points may be allocated to single grid cells, leading to averaging of the data points and oversmoothing of the fitted surface in comparison to the true terrain surface. Eventually the resolution is sufficiently fine for there to be little or no data averaging, and the slopes of the fitted surface stabilise. At this stage, all information has been extracted from the source data. This can be detected by plotting the root mean square slope of the grid across all data points as a function of grid resolution, as shown in Figure 2 of Hutchinson [12].

6 Examples

6.1 Example 1

The first example demonstrates the efficacy of the proposed Newton procedure applied to data obtained from Franke's principal test function [5]. One hundred data points were randomly selected from this function on the unit square. Three noisy data sets were created by adding to these points samples of Gaussian noise with standard deviations of 1/128, 1/16 and 1/2 respectively. These data sets were submitted to analytic bivariate thin plate spline smoothing using the ANUSPLIN package [14], with the amount of data smoothing determined by minimising the generalised cross validation. The results of these analyses are shown in Table 1. Note the agreement between the standard deviation of the data noise and the estimated standard deviations of data noise in the first and last columns of this table respectively.

A piece-wise constant finite difference iterative bivariate smoothing spline, as described above, was applied to these data sets with root mean square residuals from the data points prescribed to be 0.0036, 0.043 and 0.41 respectively, each augmented by the variance of the locally adaptive discretisation error, as given by equation (12). Grids with spacings of 0.01 were fitted across the unit square. For each data set, there were five nested grids, with successive grid spacings from 0.16 to 0.01.

The number of iterations for each grid resolution was set to 40 and the

TABLE 1: Summary statistics of minimum GCV thin plate smoothing spline analyses of data perturbed from Franke's principal test function.

Standard deviation Gaussian noise	Smoothing parameter	Signal	Square Root of GCV	Root Mean square residual	Estimated standard deviation of noise
0.0078	0.751E-04	73.8	0.014	0.0036	0.0070
0.0625	0.956E-03	36.3	0.067	0.043	0.0535
0.5000	0.616E-01	7.2	0.445	0.41	0.428

smoothing parameter was updated every second iteration, using the two Newton procedures described above. The second more approximate Newton procedure in fact performed slightly better than the first. Root mean square residuals for both procedures converged to within three figure accuracy of the prescribed values after about 20 iterations at each grid level. Comparisons of minimum and maximum values for the analytic analyses and the finite difference analyses are given in Table 2. Differences between the analytic values and the finite difference values are slightly larger for the data sets with the larger noise levels.

TABLE 2: Comparison of maximum and minimum of analytic and finite difference splines fitted to noisy bivariate data.

Root Mean square residual	Analytic thin plate spline		Finite diff. thin plate spline	
	Minimum	Maximum	Minimum	Maximum
0.0036	0.01	1.21	0.02	1.20
0.043	-0.03	1.17	-0.01	1.15
0.41	-0.21	1.22	-0.13	1.19

6.2 Example 2

This example shows the application of the second locally adaptive roughness penalty described above. The source data consisted of elevation contours extending across a 2 km × 2 km square region, as shown in Figure 2(a). A 40 m resolution grid (with 50×50 points) was fitted to this data using a second order finite difference approximation, and minimising profile curvature in the downslope direction. Stability in the aspect of the grid during the iterative solution was ensured by adding to the profile curvature penalty 1/10 th of the standard total curvature penalty. The result is shown in Figure 2(b). The relatively coarse grid used in this preliminary study has removed some of the detail in the data, but plausible trends above and below the data contours have been fitted, particularly in the long ridge extending from the bottom of the figure. The broad structure in the data contours has also been sensibly extended between the data contours.

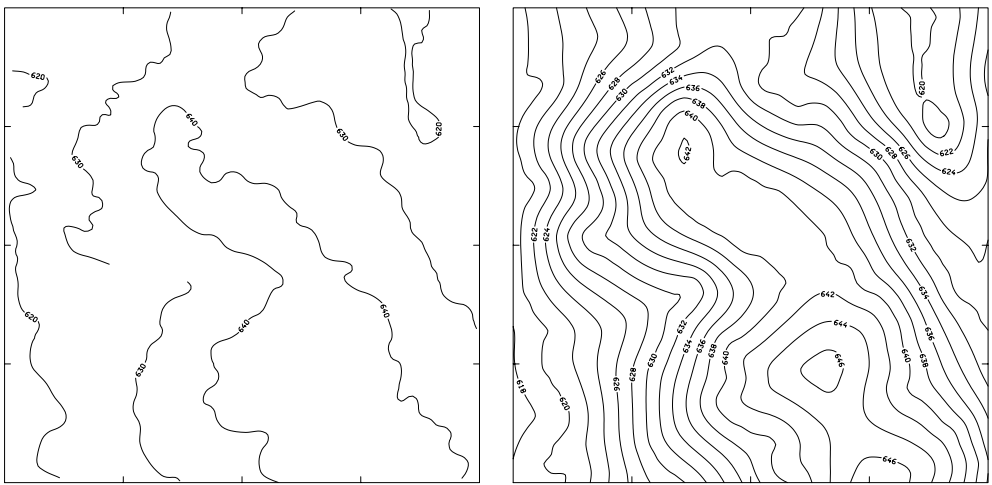

FIGURE 2: Locally adaptive, finite difference thin plate spline analysis of contour elevation data: (a) Contour data; (b) Contours of fitted grid.

7 Discussion and conclusion

The nested grid formulation of bivariate thin plate smoothing splines has been shown to be an effective method for analysing noisy bivariate data. The Newton procedure for optimising the smoothing parameter to achieve a prescribed root mean square residual from the data has also been shown to work for a wide range of smoothing parameters. The simplest version of this method, which uses the diagonal elements of the spline equation coefficients to estimate the derivative of the residual sum of squares with respect to the smoothing parameter, performed slightly better than the apparently closer approximation which iteratively solves the full set of equations for the derivative.

The simpler approximation appears to be directly compatible with the SOR iterative method. It is therefore the recommended procedure, although further investigation of the relative merits of these two procedures is warranted. Using the fuller approximation gives more stable behaviour, specifically, smaller Newton increments in the smoothing parameter, with fewer changes in sign, but, as implemented here, it converges a little more slowly.

Locally adaptive formulations of finite difference splines using nested grid SOR iteration have also been shown to be effective in allowing for positional error in the data and in implementing locally adaptive roughness penalties to respect known process based constraints. The locally adaptive strategies have been demonstrated to be sufficiently stable to allow optimisation of

the residual sum of squares using the Newton scheme described. Further investigation of these locally adaptive strategies is anticipated.

References

[1] D. Braess. *Finite Elements*. University Press, Cambridge, 1997. C779

[2] I.C. Briggs. Machine contouring using minimum curvature. *Geophysics*, 39:39–48, 1974. C777, C781

[3] P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31:377–403, 1979. C777

[4] C. De Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978. C779

[5] R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *International Journal for Numerical Methods in Engineering*, 15:1691–1704, 1980. C789

[6] D.A. Girard. A fast 'Monte-Carlo cross-validation' procedure for large least squares problems with noisy data. *Numerische Mathematik*, 56:1–23, 1989. C777

[7] G. Golub and U. von Matt. Generalized cross-validation for large scale problems. *Journal of Computational and Graphical Statistics*, 6:1–34, 1997. C778

[8] A. Greenbaum. *Iterative Methods for Solving Iterative Systems*. SIAM, Philadelphia, 1997. C780

[9] M. Hegland, S. Roberts, and I. Altas. Finite element thin plate splines for surface fitting. In B.J. Noye, M.D. Teubner, and A.W. Gill, editors, *Computational Techniques and Applications: CTAC97*, pages 289–296, Singapore 1998, World Scientific. C776

[10] M.F. Hutchinson. A new method for gridding elevation and streamline data with automatic removal of spurious pits. *Journal of Hydrology*, 106:211–232, 1989. C776, C777, C780, C784

[11] M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18:1059–1076, 1989. C778, C778

[12] M.F. Hutchinson. A locally adaptive approach to the interpolation of digital elevation models. *Proceedings Third International Conference/Workshop on Integrating GIS and Environmental Modelling*, 1996.
http://www.ncgia.ucsb.edu/conf/SANTA_FE_CD-ROM/santa_fe.html
C785, C788

[13] M.F. Hutchinson. ANUDEM Version 4.6 (1997). http://cres.anu.edu.au/software.html. C788

[14] M.F. Hutchinson. ANUSPLIN Version 4.0 (1999). http://cres.anu.edu.au/software.html C789

[15] H. Inoue. A least-squares smooth fitting for irregularly spaced data: Finite-element approach using the cubic B-spline basis. *Geophysics*, 51:2051–2066, 1986. C776

[16] C.H. Reinsch. Smoothing by spline functions. II. *Numerische Mathematik*, 16:451–454, 1971. C777

[17] W.H.F. Smith and P. Wessel. Gridding with continuous curvature. *Geophysics*, 55:293–305, 1990.