A mixed finite element discretisation of thin-plate splines

Bishnu P. Lamichhane[1] Stephen Roberts[2]
Linda Stals[3]

(Received 28 January 2011; revised 22 July 2011)

Abstract

Thin-plate splines are a well established technique for the interpolation and smoothing of scattered data. However, the traditional formulation of the method leads to large, dense and often ill-conditioned matrices, which reduces its applicability in practice. We present a new mixed finite element formulation based on the ideas behind the mortar finite element methods. The resulting system of equations is sparse and positive definite, and its size depends only on the number of finite elements not the number of data points.

Contents

http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/3934 gives this article, © Austral. Mathematical Soc. 2011. Published July 29, 2011. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

1 Introduction

Let $\Omega \subset \mathbb{R}^d$ with $d \in \{2, 3\}$ be a closed and bounded region and take $H^2(\Omega) = \{u \in L^2(\Omega),\ \nabla u \in [L^2(\Omega)]^d,\ \nabla\nabla u \in [L^2(\Omega)]^{d\times d}\}$. Given a set $\mathcal{G} = \{\mathbf{x}_i\}_{i=0}^N$ of scattered points in Ω and a set $\{z_i\}_{i=0}^N$ of values, the traditional formulation of a thin-plate spline is a smooth function $u : \Omega \to \mathbb{R}$ that minimises the functional

$$\frac{1}{N}\sum_{i=0}^{N}[u(\mathbf{x}_i) - z_i]^2 + \alpha\int_\Omega \sum_{|\nu|=2}\binom{2}{\nu}(D^\nu u)^2\, d\mathbf{x} \tag{1}$$

over the function space $H^2(\Omega)$, where $\nu = (\nu_1, \dots, \nu_d) \in \mathbb{N}_0^d$ is a d-dimensional vector used for multi-index notation. Note that $|\nu| = \sum_{i=1}^d \nu_i$ and

$$\binom{2}{\nu} = \frac{2}{\nu_1!\nu_2!\cdots\nu_d!}.$$

Further, $D^\nu u$ denotes the usual partial derivative

$$\left(\frac{\partial}{\partial x_1}\right)^{\nu_1}\cdots\left(\frac{\partial}{\partial x_d}\right)^{\nu_d} u\,,$$

and α is a positive constant. Two influential articles on the topic are by Duchon [6] and Wahba [16].

Small values of α result in an interpolation that closely follows the data, but may be sensitive to errors in the data; large values of α result in a smooth fit that may not adequately represent the data. Techniques such as generalised cross validation find an appropriate choice of α. See, for example, articles by Hutchinson [8] and the book by Wahba [16].

Thin-plate splines exhibit many favourable mathematical properties, but the traditional formulation is computationally expensive. So, our main goal is to find an efficient discretisation technique for the minimisation of the functional (1). The obvious approach is to use a H^2-conforming finite element space, but the resulting linear system is difficult to solve as it is ill-conditioned and based on large stencils. Therefore, we aim at replacing the second order derivative in the thin-plate spline formulation with a first order derivative by using a H^1-conforming finite element method. This idea has been exploited to solve the biharmonic equation and to discretise the thin-plate spline [3, 4, 5, 7, 9, 11, 12, 13, 15].

Our new formulation is obtained by introducing an auxiliary variable $\boldsymbol{\sigma} = \nabla u$ such that the minimisation problem (1) is rewritten as [3, 9]

$$\min_{\substack{(u,\boldsymbol{\sigma})\in V\\ \boldsymbol{\sigma}=\nabla u}} \left\{ \frac{1}{N}\sum_{i=0}^{N} [u(\mathbf{x}_i) - z_i]^2 + \alpha \|\nabla \boldsymbol{\sigma}\|^2_{L^2(\Omega)} \right\}, \tag{2}$$

where $V = H^1(\Omega) \times [H^1(\Omega)]^d$. The $\boldsymbol{\sigma}$ variable acts like the gradient of the data u. Since we cannot satisfy the constraint $\boldsymbol{\sigma} = \nabla u$ exactly in the discrete setting, we introduce a variational equation for this constraint using a Lagrange multiplier space. This leads us to a saddle point problem that has three unknowns: the smoother u, the gradient of the smoother $\boldsymbol{\sigma}$, and the Lagrange multiplier $\boldsymbol{\phi}$. It is advantageous if our formulation is such that the auxiliary variables (the gradient of the smoother $\boldsymbol{\sigma}$ and the Lagrange multiplier $\boldsymbol{\phi}$) can easily be eliminated from the system. This is accomplished by using a pair of bases for the gradient of the smoother and the Lagrange multiplier space that satisfy a quasi-biorthogonality property in the discrete setting. We will give detailed analysis elsewhere [10], here we give an overview of the main ideas and write briefly about the solver.

2 Discrete setting

Let $\mathcal{T}_h$ be a globally quasi-uniform and shape regular triangulation of the polygonal domain Ω having the mesh-size h consisting of triangles or tetrahedra. Let

$$S_h = \{u_h \in C^0(\Omega) \mid u_{h|_T} \in \mathcal{P}(T),\ T \in \mathcal{T}_h\} \tag{3}$$

be the standard linear finite element space, and

$$B_h = \left\{ b_h \mid b_h|_T = (d+1)^{d+1} \prod_{i=1}^{d+1} \lambda_i^T\,,\ T \in \mathcal{T}_h \right\},$$

be the space of bubble functions, where $\mathcal{P}(T)$ is the space of linear functions on T, and $\{\lambda_i^T\}_{i=1}^{d+1}$ is the set of barycentric co-ordinates on T. Let $L_h = S_h \oplus B_h$. L_h is explicitly defined on a reference element in Section 4. In a more detailed article [10] we will explain why it is not possible to obtain stability if we discretise the space V by using $S_h \times [S_h]^d$ instead of $S_h \times [L_h]^d$.

Denoting the discrete counterpart of the continuous space V by $V_h = S_h \times [L_h]^d$, our discrete problem is to find

$$\min_{(u_h, \sigma_h) \in V_h} \left\{ \frac{1}{N} \sum_{i=0}^{N} [u_h(\mathbf{x}_i) - z_i]^2 + \alpha \|\nabla \boldsymbol{\sigma}_h\|^2_{L^2(\Omega)} \right\}, \tag{4}$$

subject to

$$\langle \boldsymbol{\sigma}_h, \boldsymbol{\psi}_h \rangle_{L^2(\Omega)} = \langle \nabla u_h, \boldsymbol{\psi}_h \rangle_{L^2(\Omega)}, \quad \boldsymbol{\psi}_h \in [M_h]^d.$$

The space $[M_h]^d$ plays the role of a Lagrange multiplier space. The specific formulation of M_h, as given in Section 4, is influenced by the properties of the resulting systems of equations.

Denoting function values of u at the measurement points $\{x_i\}_{i=0}^N$ by

$$Pu = (u(x_0), u(x_1), \dots, u(x_N))^T,$$

the minimisation problem (4) is equivalent to

$$\min_{(u_h,\sigma_h)\in V_h} \left(\frac{1}{N}\|Pu_h\|^2 + \alpha\|\nabla\sigma_h\|^2_{L^2(\Omega)} - \frac{2}{N}(Pu_h)^T z \right), \tag{5}$$

with the side condition

$$\int_\Omega \sigma_h \cdot \psi_h \, dx = \int_\Omega \nabla u_h \cdot \psi_h \, dx, \quad \psi_h \in [M_h]^d. \tag{6}$$

Here z is a column vector with ith entry z_i for $i = 0, \dots, N$, and $\|\cdot\|$ is the standard Euclidean norm. Hence we have a constrained minimisation problem which gives rise to a saddle point structure.

3 Positive definite formulation

To obtain the positive definite formulation, we introduce a quasi-projection operator, $Q_h : L^2(\Omega) \to L_h$, which is defined as

$$\int_\Omega (Q_h v)\mu_h \, dx = \int_\Omega v\mu_h \, dx, \quad v \in L^2(\Omega), \quad \mu_h \in M_h.$$

This type of operator was introduced by Scott and Zhang [14] to obtain the finite element interpolation of non-smooth functions satisfying boundary conditions, and was used by Bernardi et al. [2] in the context of mortar finite elements. The definition of Q_h allows us to write the weak gradient as

$$\sigma_h = Q_h(\nabla u_h),$$

where the operator Q_h is applied to a vector component-wise.

Using the definition of the operator Q_h, we eliminate the degrees of freedom corresponding to $\boldsymbol{\sigma}_h$ so that our problem is to minimise the functional

$$J(u_h) = \frac{1}{N}\|Pu_h\|^2 + \alpha\|\nabla(Q_h\nabla u_h)\|^2_{L^2(\Omega)} - \frac{2}{N}(Pu_h)^T\boldsymbol{z} \tag{7}$$

over the space S_h. We note the replacement of $\|\nabla\boldsymbol{\sigma}\|^2_{L^2(\Omega)}$ in the continuous problem (1) by $\|\nabla(Q_h\nabla u_h)\|^2_{L^2(\Omega)}$ in the discrete problem (7).

We are currently undertaking a convergence analysis of the above formulation with results to be reported elsewhere [10].

4 Algebraic formulation and construction of M_h

In the following, we use the same notation for the vector representation of the solution and the solution as elements in S_h, $[L_h]^d$ and $[M_h]^d$. Let R, A, B and D be the matrices associated with the bilinear forms

$$\frac{1}{N}(Pu_h)^TPv_h\,, \quad \int_\Omega \nabla\boldsymbol{\sigma}_h : \nabla\boldsymbol{\tau}_h\,dx\,, \quad \int_\Omega \nabla u_h\cdot\boldsymbol{\psi}_h\,dx \quad \text{and} \quad \int_\Omega \boldsymbol{\sigma}_h\cdot\boldsymbol{\psi}_h\,dx\,,$$

respectively, where $u_h, v_h \in S_h$, $\boldsymbol{\sigma}_h, \boldsymbol{\tau}_h \in [L_h]^d$, and $\boldsymbol{\psi}_h \in [M_h]^d$. The matrix D associated with the bilinear form $\int_\Omega \boldsymbol{\sigma}_h\cdot\boldsymbol{\psi}_h\,dx$ is often called a Gram matrix. We recall that $\nabla\boldsymbol{\sigma}_h : \nabla\boldsymbol{\tau}_h$ denotes the dot product of two matrices $\nabla\boldsymbol{\sigma}_h$ and $\nabla\boldsymbol{\tau}_h$.

By using Lagrange multipliers, the minimum of (7) is found by finding u_h, $\boldsymbol{\sigma}_h$ and $\boldsymbol{\phi}_h$ satisfying

$$\frac{1}{N}(Pu_h)^TPv_h - \int_\Omega \nabla v_h\cdot\boldsymbol{\phi}_h\,dx = f(v_h), \quad v_h \in S_h\,,$$

$$\alpha \int_\Omega \nabla \boldsymbol{\sigma}_h : \nabla \boldsymbol{\tau}_h \, dx + \int_\Omega \boldsymbol{\phi}_h \cdot \boldsymbol{\tau}_h \, dx = 0 \,, \quad \boldsymbol{\tau}_h \in [L_h]^d,$$

where $f(\nu_h) = \frac{1}{N}(P\nu_h)^T \boldsymbol{z}$. The algebraic formulation of the equivalent saddle point problem is

$$\begin{bmatrix} R & 0 & -B^T \\ 0 & \alpha A & D^T \\ -B & D & 0 \end{bmatrix} \begin{bmatrix} u_h \\ \boldsymbol{\sigma}_h \\ \boldsymbol{\phi}_h \end{bmatrix} = \begin{bmatrix} f_h \\ 0 \\ 0 \end{bmatrix}, \tag{8}$$

where f_h is the vector form of discretisation of the linear form $f(\cdot)$. The last line in Equation (8) comes from the side condition given in Equation (6).

Each submatrix in (8) is sparse and has size independent of the number of data points, although the sizes of B, A and D grow with the dimension. The time taken to construct the R matrix depends linearly on the number of data points; the other matrices only depend on the size of the finite element grid.

We recall that a Gram matrix of two sets of basis functions $\{\mu_i\}_{1\leqslant i\leqslant n}$ and $\{\varphi_i\}_{1\leqslant i\leqslant n}$ is a matrix G whose ijth entry is

$$\int_\Omega \mu_i \varphi_j \, dx \,.$$

Definition 1 *The basis functions* $\{\mu_i\}_{1\leqslant i\leqslant n}$ *of* M_h *and the basis functions* $\{\varphi_i\}_{1\leqslant i\leqslant n}$ *of* L_h *are called* ***quasi-biorthogonal*** *and the resulting Gram matrix* G *is called* ***quasi-diagonal*** *if* G *is of the form*

$$G = \begin{bmatrix} D_1 & 0 \\ R & D_2 \end{bmatrix} \quad \text{or} \quad G = \begin{bmatrix} D_1 & R \\ 0 & D_2 \end{bmatrix},$$

where D_1 *and* D_2 *are diagonal matrices and* R *is a sparse rectangular matrix.*

If the Gram matrix of two bases $\{\mu_i\}_{1\leqslant i\leqslant n}$ of M_h and $\{\varphi_i\}_{1\leqslant i\leqslant n}$ of L_h is diagonal or quasi-diagonal, the matrix D in (8) is diagonal or quasi-diagonal.

Using a quasi-biorthogonal system, the inverse of the Gram matrix G is

$$G^{-1} = \begin{bmatrix} D_1^{-1} & 0 \\ -D_2^{-1} R D_1^{-1} & D_2^{-1} \end{bmatrix} \quad \text{or} \quad G^{-1} = \begin{bmatrix} D_1^{-1} & -D_1^{-1} R D_2^{-1} \\ 0 & D_2^{-1} \end{bmatrix},$$

and the inverse G^{-1} is sparse.

In the following we show how to construct the local basis functions of L_h and M_h so that they form a quasi-biorthogonal system. The construction is shown on the reference triangle $\widehat{T} = \{(x, y) : 0 \leqslant x\,,\ 0 \leqslant y\,,\ x + y \leqslant 1\}$ in the two dimensional case. Construction of the basis functions in the three dimensional case is to be described later [10].

Let

$$\widehat{\varphi}_1 = 1 - x - y\,, \quad \widehat{\varphi}_2 = x\,, \quad \widehat{\varphi}_3 = y\,, \quad \text{and} \quad \widehat{\varphi}_4 = 27xy\,(1 - x - y)$$

be the local basis functions of L_h on the reference triangle associated with three vertices $(0, 0)$, $(1, 0)$, $(0, 1)$ and one barycentre $(\frac{1}{3}, \frac{1}{3})$. We define the local basis functions of M_h as

$$\begin{aligned} \widehat{\mu}_1 &= \frac{34}{9} - 4\,x - 4\,y - \frac{140}{3}\,xy\,(1 - x - y)\,, \\ \widehat{\mu}_2 &= -\frac{2}{9} + 4\,x - \frac{140}{3}\,xy\,(1 - x - y)\,, \\ \widehat{\mu}_3 &= -\frac{2}{9} + 4\,y - \frac{140}{3}\,xy\,(1 - x - y)\,, \end{aligned}$$

and

$$\widehat{\mu}_4 = 27\,xy\,(1 - x - y)\,,$$

so the two sets of basis functions $\{\widehat{\varphi}_1, \dots, \widehat{\varphi}_4\}$ and $\{\widehat{\mu}_1, \dots, \widehat{\mu}_4\}$ form a quasi-biorthogonal system on the reference triangle.

As these local basis functions for M_h are defined on the reference element, they are mapped by an affine mapping to construct basis functions for an actual element. The global basis functions of M_h are then constructed by glueing these basis functions together. This process is exactly the same as constructing global basis functions for the space S_h.

Given the above formulation for L_h and M_h, the local Gram matrix on the reference triangle $\widehat{T}$ is

$$D_{\widehat{T}} = \begin{bmatrix} 1/6 & 0 & 0 & 0 \\ 0 & 1/6 & 0 & 0 \\ 0 & 0 & 1/6 & 0 \\ \frac{3}{40} & \frac{3}{40} & \frac{3}{40} & \frac{81}{560} \end{bmatrix} .$$

As the local matrix $D_{\widehat{T}}$ is quasi-diagonal, the global matrix D will be quasi-diagonal.

5 Numerical examples

To better understand the theoretical results presented in the previous sections, the mixed finite element method described in Section 4 was implemented in a C++ code developed by Stals. In all of the examples the domain is the unit square and the finite element grid is uniform.

To find the interpolant u_h we eliminate $\boldsymbol{\sigma}_h$ and $\boldsymbol{\phi}_h$ from (8) to obtain

$$\left(R + \alpha B^T D^{-T} A D^{-1} B\right) u_h = f_h. \tag{9}$$

The gradient is then $\boldsymbol{\sigma}_h = D^{-1} B u_h$.

The matrix in (9) is symmetric and positive definite so the preconditioned conjugate gradient (PCG) method was used as a solver. The preconditioner we use is a multiplicative Schwarz preconditioner. Define two subdomains Ω_1 and Ω_2 where Ω_1 is the same as the original domain Ω except the value of u_h at a corner point, such as $(0, 0)$, is set to zero and similarly Ω_2 is the same

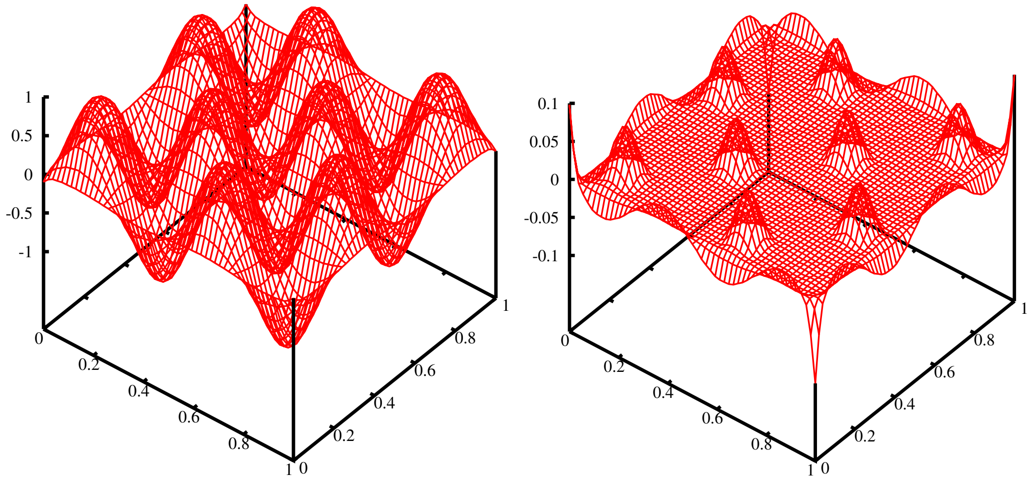

Figure 1: The left plot shows the thin-plate spline approximation u_h of $\sin(4\pi x)\sin(4\pi y)$ with missing data on a finite element grid containing 129×129 nodes and $\alpha = 10^{-8}$, while the right plot shows $\widehat{u}(x,y) - u_h$

as Ω except the value of u_h at another corner point, such as $(1,1)$, is set to zero. Let L be the standard Laplace matrix associated with the bilinear form $\int_\Omega \nabla u_h \cdot \nabla v_h \, \mathbf{dx}$ where $u_h, v_h \in S_h$, and D_1 be the submatrix of D that is specified in Definition 1. Then the matrix $R + \alpha L D_1^{-1} L$ is symmetric and positive definite on Ω_1 and Ω_2. As part of the Schwarz preconditioner (9) must be solved on each subdomain; we use PCG with $\left(R + \alpha L D_1^{-1} L\right)^{-1}$ as the preconditioner for that subdomain.

The stopping criterion used in the PCG solver is to exit when the residual is less than some tolerance. For the example discussed here the tolerance was set to 10^{-12}.

Table 1: Number of PCG iterations on a finite element grid containing n nodes and with the given values of α. The results are recorded in the form a, b (c) where a is the number of times the Schwarz preconditioner was called, b is the total number of PCG iterations required to solve Equation (9) on each subdomain and c the number of PCG iterations required to solve Equation (9) if no preconditioner is used. A value of '*' means that the solver needs more than 1000 iterations

n (h)	α				
	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
9 (1/2)	5, 37 (6)	4, 14 (6)	3, 6 (6)	2, 4 (6)	2, 4 (6)
25 (1/4)	4, 53 (17)	3, 16 (14)	3, 8 (14)	2, 4 (14)	2, 4 (14)
81 (1/8)	4, 76 (64)	4, 35 (25)	3, 9 (25)	3, 4 (27)	2, 4 (27)
289 (1/16)	4, 104 (243)	4, 59 (74)	3, 19 (35)	3, 8 (46)	2, 4 (47)
1089 (1/32)	4, 176 (975)	4, 91 (263)	4, 44 (52)	3, 18 (76)	3, 13 (118)
4225 (1/64)	4, 257 (*)	4, 137 (*)	4, 60 (173)	3, 29 (57)	3, 19 (211)
16641 (1/128)	4, 398 (*)	4, 192 (*)	4, 93 (580)	3, 36 (133)	3, 24 (165)
66049 (1/256)	4, 577 (*)	4, 282 (*)	4, 138 (*)	3, 48 (448)	3, 28 (194)

5.1 Sine function

A data set was created by evaluating the function $\hat{u}(x, y) = \sin(4\pi x)\sin(4\pi y)$ at 998002 points uniformly spaced over the unit square. Next we took the data set and remove all of the data points with values greater than 0.75. The right plot in Figure 1 shows $\hat{u}(x, y) - u_h$. Specifically it shows how the method filled in the missing data. Notice that $|\hat{u}(x, y) - u_h| < 0.1$.

Table 1 records the number of PCG iterations required to solve (9) on the finite element grid with n nodes using the current data set. The cost of each conjugate gradient iteration on Ω_1 and Ω_2 is the same as the cost per iteration on Ω.

The results in the table indicate that the condition of the system deteriorates with larger values of α. For large values of α we expect the condition number to grow like $O(h^{-4})$. The preconditioner proved to be very effective, except for small values of n. The number of iterations required to solve the preconditioned system does not show the strong growth with n that is evident when no preconditioner is used. The number of iterations for the preconditioned system does increase with α; however, for practical applications we would not often need to evaluate the system with larger values of α as the solution does not display any of the characteristics of the data.

To apply the preconditioner it is necessary to solve systems of the form

$$\left(R + \alpha L D_1^{-1} L\right) \mathbf{x} = \mathbf{f}\,, \tag{10}$$

for some given $\mathbf{f}$. The condition number of this system will also grow like $O(h^{-4})$ for large values of α, therefore we use the sparse direct solver UMFPACK to solve the equation. It may be argued that we could use UMFPACK to solve (9) directly, but we have not done that for two reasons. The first reason is that to use UMFPACK the matrix $B^T D^{-T} A D^{-1} B$ must be explicitly constructed. The cost of constructing the matrix is $O(d^6 n^5)$, which is extremely high. Also, the size of the resulting stencil would be large, thus increasing the potential for fill-in. The second reason is that using a direct solver on (9) would limit the size the finite element grids.

5.2 Bathymetry test set

As an example of an application we find the thin-plate spline approximation to bathymetry data of Medicine Lake, California [1]. A contour plot of the original bathymetry data set is given in Figure 2. The thin-plate spline approximations on a finite element grid containing 4225 nodes with $\alpha = 10^{-3}$ and $\alpha = 10^{-4}$ are shown in Figure 3 and Figure 4 respectively.

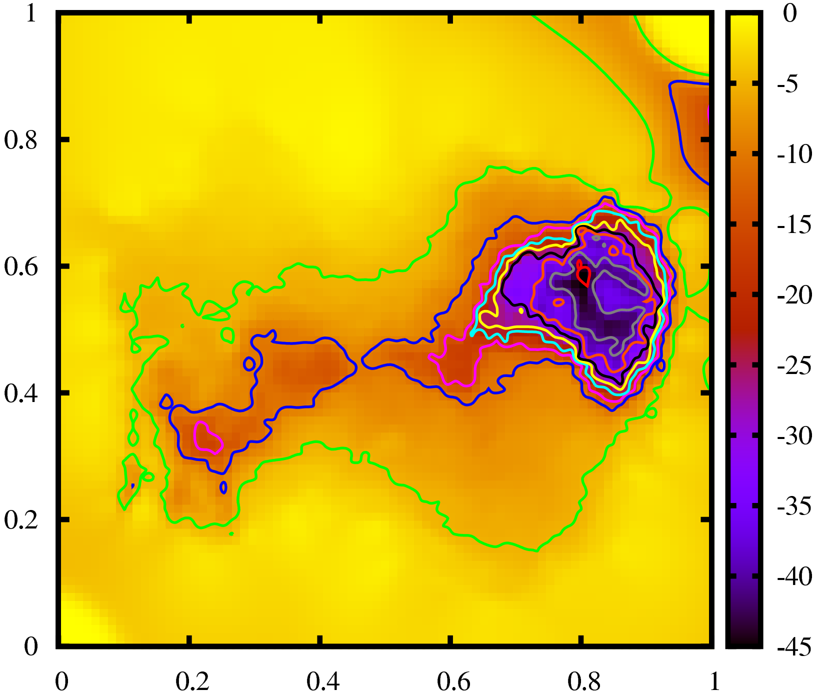

Figure 2: Contour plot of original bathymetry data set.

References

[1] http://geopubs.wr.usgs.gov/open-file/of00-043/bathymetry/appendices.html. C529

[2] C. Bernardi, Y. Maday, and A. T. Patera. A new nonconforming approach to domain decomposition: the mortar element method. In H. Brezis & J.-L. Lions, editor, *Nonlinear Partial Differential Equations and Their Applications*, pages 13–51. Paris, 1994. C522

[3] X. Cheng, W. Han, and H. Huang. Some mixed finite element methods for biharmonic equation. *Journal of Computational and Applied*

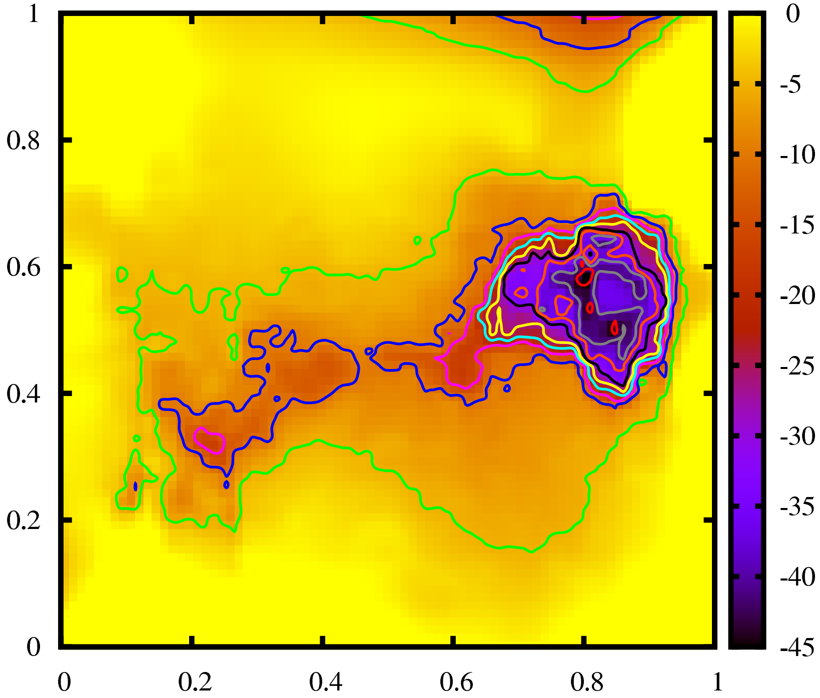

Figure 3: Finite element approximation using 4225 nodes and $\alpha = 10^{-3}$.

Mathematics, 126(1-2):91–109, 2000. http://dx.doi.org/10.1016/S0377-0427(99)00342-8. C520

[4] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North Holland, Amsterdam, 1978. C520

[5] P. G. Ciarlet and P. A. Raviart. A mixed finite element method for the biharmonic equation. In C. De Boor, editor, *Symposium on Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 125–143, New York, 1974. Academic Press. C520

[6] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables,*

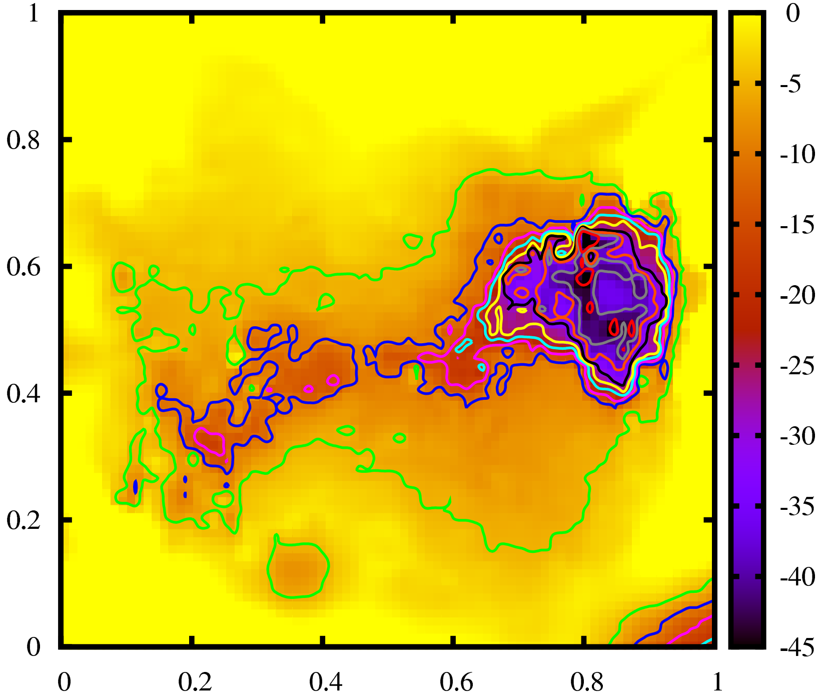

Figure 4: Finite element approximation using 4225 nodes and $\alpha = 10^{-4}$.

Lecture Notes in Mathematics, volume 571, pages 85–100. Springer-Verlag, Berlin, 1977. http://www.springerlink.com/content/g27671q701166031/. C520

[7] R. S. Falk. Approximation of the biharmonic equation by a mixed finite element method. *SIAM Journal on Numerical Analysis*, 15(3):556–567, 1978. http://epubs.siam.org/sinum/resource/1/sjnaam/v15/i3. C520

[8] M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics — Simulation and Computation*, 18(3):1059–1076, 1989. http://www.tandfonline.com/toc/lssp20/18/3. C520

[9] C. Johnson and J. Pitkäranta. Analysis of some mixed finite element methods related to reduced integration. *Mathematics of Computation*, 38(158):375–400, 1982. http://www.ams.org/journals/mcom/1982-38-158/home.html. C520

[10] B. P. Lamichhane, S. Roberts, and L. Stals. A mixed finite element discretization of thin plate splines based on quasi-biorthogonal systems. To be submitted. C521, C523, C525

[11] P. Monk. A mixed finite element method for the biharmonic equation. *SIAM Journal on Numerical Analysis*, 24(4):737–749, 1987. http://epubs.siam.org/sinum/resource/1/sjnaam/v24/i4. C520

[12] T. Ramsay. Spline smoothing over difficult regions. *Journal of Royal Statistical Society. Series B (Statistical Methodology)*, 64(2):307–319, 2002. http://onlinelibrary.wiley.com/doi/10.1111/rssb.2002.64.issue-2/issuetoc. C520

[13] S. Roberts, M. Hegland, and I. Altas. Approximation of a thin plate spline smoother using continuous piecewise polynomial functions. *SIAM Journal on Numerical Analysis*, 41(1):208–234, 2003. http://epubs.siam.org/sinum/resource/1/sjnaam/v41/i1. C520

[14] L. R. Scott and S. Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Mathematics of Computation*, 54(190):483–493, 1990. http://www.ams.org/journals/mcom/1990-54-190/home.html. C522

[15] L. Stals and S. Roberts. Smoothing large data sets using discrete thin plate splines. *Computing and Visualization in Science*, 9(3):185–195, 2006. http://www.springerlink.com/content/1432-9360/9/3/. C520

[16] G. Wahba. *Spline Models for Observational Data*, volume 59 of *Series in Applied Mathematic*. SIAM, Philadelphia, first edition, 1990. C520

Author addresses

1. **Bishnu P. Lamichhane**, School of Mathematical and Physical Sciences, The University of Newcastle, Callaghan, NSW 2308, Australia
   mailto:Bishnu.Lamichhane@newcastle.edu.au

2. **Stephen Roberts**, Department of Mathematics, Australian National University, Canberra, ACT 0200, Australia
   mailto:Stephen.Roberts@anu.edu.au

3. **Linda Stals**, Department of Mathematics, Australian National University, Canberra, ACT 0200, Australia
   mailto:Linda.Stals@anu.edu.au