

Approximation of noisy data using multivariate splines and finite element methods

Bishnu P Lamichhane¹ , Elizabeth Harris¹ and Quoc Thong Le Gia²

Journal of Algorithms & Computational Technology
Volume 15: 1–12
© The Author(s) 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/17483026211008405
journals.sagepub.com/home/act

Abstract

We compare a recently proposed multivariate spline based on mixed partial derivatives with two other standard splines for the scattered data smoothing problem. The splines are defined as the minimiser of a penalised least squares functional. The penalties are based on partial differential operators, and are integrated using the finite element method. We compare three methods to two problems: to remove the mixture of Gaussian and impulsive noise from an image, and to recover a continuous function from a set of noisy observations.

Keywords

Sobolev space, scattered data interpolation, Gaussian noise, impulsive noise

Received 16 November 2020; revised 16 November 2020; accepted 12 February 2021

Introduction

We begin by outlining the scattered data problem. Consider the set of scattered points $\mathcal{G} = \{\mathbf{p}_i\}_{i=1}^N$ in a domain $\Omega \subseteq \mathbb{R}^d$ with $d \in \mathbb{N}$, and the set of noisy observations at those points $\{z_i\}_{i=1}^N$. We want to reconstruct an unknown function u to approximate the given data. Assuming that the underlying data set is corrupted with Gaussian noise, we can assume that the unknown function u satisfies

$$z_i = u(\mathbf{p}_i) + n_i$$

$i = 1, \dots, N$, where $\{n_i\}_{i=1}^N$ is a set of normally distributed random variables with mean 0 and variance σ^2 .

To recover the unknown function u , we will use an approach based on the multivariate L-spline. That is, we will search for a function u that minimises the following least squares functional

$$\sum_{i=1}^N (z_i - u(\mathbf{p}_i))^2 + \lambda \int_{\Omega} (Lu(x, y))^2 dx$$

over a Sobolev space V , where L is a partial differential operator, and λ is a positive smoothing parameter.

We use the standard notation for Sobolev spaces on Ω .^{1–3} We consider three different choices for L . The first choice is to take Lu as the gradient of u . Then we need to have $V = H^1(\Omega)$. However, the continuous problem is not well-posed with this choice for $d \geq 2$, because the point value of a function is not defined in $H^1(\Omega)$ when $d \geq 2$. The second choice is to choose Lu as the Laplacian of u . Then we need to have $V = H^2(\Omega)$. Again, the continuous problem is not well-posed for $d > 3$, because the point value of a function is not defined in $H^2(\Omega)$ when $d > 3$. The third choice is to include mixed partial derivatives of u on the gradient penalty to construct Lu .⁴ Unlike the other two choices, the resulting spline is well defined for any dimension $d \in \mathbb{N}$.

¹School of Mathematical & Physical Sciences, Mathematics Building, University of Newcastle, University Drive, Callaghan, NSW, Australia

²School of Mathematics and Statistics, University of New South Wales, Sydney, NSW, Australia

Corresponding author:

Bishnu P Lamichhane, School of Mathematical & Physical Sciences, Mathematics Building, University of Newcastle, University Drive, Callaghan, NSW 2308, Australia.

Email: Bishnu.Lamichhane@newcastle.edu.au

This is the first time that computational results of the newly proposed multivariate spline⁴ are presented and compared with other existing techniques. Moreover, we observe the instability of the gradient penalty approach in our numerical experiments, which is another novelty of this contribution.

We will apply these methods to two problems. The first problem is to recover an image that have been corrupted with both Gaussian and impulsive noise. We apply a finite element method to compute the solution of the above minimisation problem. Finite element methods have recently become popular in different areas of image processing.^{5–9} Finite element methods are applied in Lamichhane¹⁰ and Lamichhane¹¹ to remove the mixture of Gaussian and impulsive noise using the gradient penalty and total variation penalty, respectively.

The second problem is to recover a continuous function from a set of noisy observations. We consider observations that have been corrupted with Gaussian noise. In this example, we see spurious spikes in the solution using the gradient penalty. This is due to the fact that the gradient penalty does not control the point-wise values of the function. Numerical results show that we can increase the mesh-size to reduce the height of the spikes but they cannot be totally removed.

This paper is organised as follows. In the next section, we present the gradient penalty smoothing technique. In the third section, we present the smoothing technique based on the minimisation of a functional involving mixed partial derivatives. In the fourth section, we compare the three finite element methods in denoising images and recovering continuous functions. We discuss these results in the last section.

Multivariate spline with gradient penalty

The multivariate spline with the gradient penalty is given by the following minimisation problem

$$\min_{u \in V} \left(\sum_{i=1}^N (u(\mathbf{p}_i) - z_i)^2 + \lambda \int_{\Omega} \|\nabla u\|^2 d\mathbf{x} \right) \quad (1)$$

Due to the choice of the minimisation functional it is natural to take $V = H^1(\Omega)$, for which the problem will not be well-posed when $d > 1$.

Now we consider a finite element discretisation of the spline. Let $C^0(\Omega)$ be the space of continuous functions in Ω and $\mathcal{T}_h$ a finite element triangulation of Ω . Note that $\mathcal{T}_h$ is the set of triangles or rectangles.

Then let

$$V_h = \{u_h \in C^0(\Omega) \mid u_h|_T \in P(T), T \in \mathcal{T}_h\} \quad (2)$$

be a finite element space, where $P(T)$ is the linear polynomial space if T is a triangle, and $P(T)$ is the bilinear polynomial space on T if T is a rectangle.¹² The minimisation problem leads to the variational problem of finding $u_h \in V_h$ such that

$$a(u_h, v_h) = \ell(v_h), \quad v_h \in V_h$$

where the bilinear form $a(\cdot, \cdot)$ and the linear form $\ell(\cdot)$ are given by

$$a(u, v) = \sum_{i=1}^N u(\mathbf{p}_i) v(\mathbf{p}_i) + \lambda \int_{\Omega} \nabla u \cdot \nabla v d\mathbf{x}$$

$$\ell(v) = \sum_{i=1}^N v(\mathbf{p}_i) z_i$$

It is easy to show that the above problem has a unique solution under the assumption that the set of scattered points $\mathcal{G}$ is non-empty.¹⁰

Since $a(\cdot, \cdot)$ is positive definite, we can define the energy $\|\cdot\|_a$ on V as

$$\|v\|_a^2 = a(v, v)$$

for all $v \in V$

The following lemma shows that the discrete multivariate spline with the gradient penalty is well-posed for $d=1$ but not well-posed for $d>1$. The point-value of a function is not controlled by the gradient of the function when $d>1$. The well-posedness is exhibited in the stability result first proved in Garcke and Hegland.¹³ For completeness we have given these results in Lemmas 1, 2 and 3, which are taken from Garcke and Hegland.¹³

Lemma 1. (*Discrete Sobolev inequality*). *There exists constant $c_d > 0$ such that for all $u \in V_h$*

1. $|u(x)| \leq c_d \|u\|_{H^1(\Omega)}$ for $d=1$.
2. $|u(x)| \leq c_d (1 + |\log h|) \|u\|_{H^1(\Omega)}$ for $d=2$.
3. $|u(x)| \leq c_d h^{1-d/2} \|u\|_{H^1(\Omega)}$ for $d>2$.

The constant c_d is independent of the mesh-size h but depends on d . These bounds are tight, and for $d>2$ we have that

$$c_d \geq \frac{1}{\sqrt{3d+1}} \left(\frac{3}{2} \right)^{d/2}$$

Lemma 2. (*Discrete Poincaré inequality*). Let $(x_0, y_0) \in \Omega$ and $u_0 = u(x_0, y_0)$ for $u \in V_h$. Then there exist constants $c_d > 0$ such that

1. $\|u - u_0\|_{L^2(\Omega)} \leq c_d \|\nabla u\|_{L^2(\Omega)}$ for $d = 1$
2. $\|u - u_0\|_{L^2(\Omega)} \leq c_d(1 + |\log h|) \|\nabla u\|_{L^2(\Omega)}$ for $d = 2$
3. $\|u - u_0\|_{L^2(\Omega)} \leq c_d h^{1-d/2} \|\nabla u\|_{L^2(\Omega)}$ for $d > 2$.

Lemma 3. (*Discrete V-ellipticity*). There exist constants c_d and C_d such that the energy norm on V_h satisfies

$$\alpha_{d,h} \|u\|_{H^1(\Omega)} \leq \|u\|_a \leq \beta_{d,h} \|u\|_{H^1(\Omega)}$$

for all $u \in V_h$, where $\alpha_{d,h}$ and $\beta_{d,h}$ are given by

1. $\alpha_{d,h} = \left(\frac{c_d}{\sqrt{\lambda}} + \frac{1}{\sqrt{\lambda}} + 1 \right)^{-1}$ and $\beta_{d,h} = C_d + \sqrt{\lambda}$ for $d = 1$
2. $\alpha_{d,h} = \left(\frac{c_d(1+|\log h|)}{\sqrt{\lambda}} + \frac{1}{\sqrt{\lambda}} + 1 \right)^{-1}$ and $\beta_{d,h} = C_d(1 + |\log h|) + \sqrt{\lambda}$ for $d = 2$
3. $\alpha_{d,h} = (c_d \lambda^{-1/2} h^{1-d/2} + \lambda^{-1/2} + 1)^{-1}$ and $\beta_{d,h} = C_d h^{1-d/2} + \sqrt{\lambda}$ for $d > 2$.

The above results imply that for the solution $u_h \in V_h$ of the spline with the gradient penalty we have

$$\|u_h\|_a \leq \frac{\beta_{d,h}}{\alpha_{d,h}} \|\ell\|_{L^2(\Omega)}$$

where $\|\ell\|_{L^2(\Omega)}$ is the L^2 -norm of the linear functional ℓ .

Remark 4. We can see that the ill-posedness is exhibited in the stability constant being not independent of the mesh-size h . There is no easy way to remove this dependency.

Multivariate spline with Laplacian penalty

The multivariate spline with the Laplacian penalty is the solution of the following minimisation problem

$$\min_{u \in V} \left(\sum_{i=1}^N (u(\mathbf{p}_i) - z_i)^2 + \lambda \int_{\Omega} (\Delta u)^2 d\mathbf{x} \right) \quad (3)$$

where $V = H^2(\Omega)$. The minimisation problem is well-posed for $d \in \{1, 2, 3\}$ since $H^2(\Omega) \subset C^0(\Omega)$ when $d \in \{1, 2, 3\}$. More details of this spline can be found in Ramsay.¹⁴ This spline will be called Laplacian spline in the following.

Remark 5. The Laplacian penalty imposes higher smoothness in the solution. Since we look for a solution, whose second derivatives are square integrable, point-values of the solution are well-defined in contrast to the

gradient penalty spline. However, a finite element approximation of the Laplacian spline is more expensive than the gradient penalty spline.

While we can use a direct finite element approximation using a low order finite element space V_h to approximate the gradient penalty spline, we cannot directly use this space to approximate the Laplacian spline as Δu_h for $u_h \in V_h$ is not well-defined. We use a mixed finite element method proposed by Lamichhane.¹⁵ We first introduce a new variable $\phi = \Delta u$ in the minimisation formulation (3) and then write a weak equation as

$$\int_{\Omega} \phi \mu d\mathbf{x} = \int_{\Omega} \nabla u \cdot \nabla \mu d\mathbf{x}, \quad \mu \in H^1(\Omega) \quad (4)$$

Now choosing $\phi \in L^2(\Omega)$ and $u, \mu \in H^1(\Omega)$, we have a well-defined formulation for which we can use the space V_h to discretise u and μ , whereas we use a discontinuous piecewise polynomial space W_h to discretise ϕ . The basis functions of W_h and V_h satisfy a biorthogonality relationship¹⁵ so that the associated matrix corresponding to the L^2 -inner product $\int_{\Omega} \phi \mu d\mathbf{x}$ in (4) is diagonal. In this way, we arrive at a very efficient finite element method to approximate the solution of the minimisation problem (3). The discrete problem is then to compute

$$\min_{[u_h, \phi_h] \in V_h \times W_h} \left(\sum_{i=1}^N (u_h(\mathbf{p}_i) - z_i)^2 + \lambda \int_{\Omega} \phi_h^2 d\mathbf{x} \right)$$

subject to the constraint

$$\int_{\Omega} \phi_h \mu_h d\mathbf{x} = \int_{\Omega} \nabla u_h \cdot \nabla \mu_h d\mathbf{x}, \quad \mu_h \in V_h$$

Since $C^0(\Omega) \not\subset H^1(\Omega)$ when $d > 1$, the gradient penalty spline formulation does not provide a well-posed problem. The problem is well-posed only in one dimension. Similarly, $C^0(\Omega) \subset H^2(\Omega)$ for $d \in \{1, 2, 3\}$ but $C^0(\Omega) \not\subset H^2(\Omega)$ when $d > 3$. This motivates us to find a well-posed spline formulation for any $d \in \mathbb{N}$, which is given in the next section.

New multivariate spline with mixed derivative penalty

In order to define the new multivariate spline, we define the associated Sobolev space. Let $\mathcal{B} = \{0, 1\}^d \setminus \{\mathbf{0}\}$, where $\mathbf{0} \in \mathbb{R}^d$ is a zero vector. We use a standard multi-index notation with $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathcal{B}$ so that

a mixed derivative of a sufficiently smooth function u is denoted by

$$D^\alpha u = \frac{\partial \sum_{i=1}^d \alpha_i u}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}$$

where we use the usual Cartesian coordinate system with $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$.

We now define our Sobolev space for the multivariate spline problem as

$$H_m^1(\Omega) := \{u \in L^2(\Omega) : D^\alpha u \in L^2(\Omega), \alpha \in \mathcal{B}\}$$

which is equipped with the norm

$$\|u\|_{H_m^1(\Omega)} = \sqrt{\|u\|_{L^2(\Omega)}^2 + \sum_{\alpha \in \mathcal{B}} \|D^\alpha u\|_{L^2(\Omega)}^2}$$

and the semi-norm

$$|u|_{H_m^1(\Omega)} = \sqrt{\sum_{\alpha \in \mathcal{B}} \|D^\alpha u\|_{L^2(\Omega)}^2}$$

The semi-norm for $d=2$ is simply

$$|v|_{H_m^1(\Omega)} = \sqrt{\int_{\Omega} \|\nabla v\|^2 + \left(\frac{\partial^2 v}{\partial x \partial y}\right)^2 d\mathbf{x}}$$

where $\|\nabla v\|^2$ is the Euclidean norm of the vector $\nabla v \in \mathbb{R}^2$.

We note that the space $H_m^1(\Omega)$ is a Hilbert space, and $H_m^1(\Omega) \subset C^0(\Omega)$.¹⁶ The new multivariate spline is then obtained as a solution of the minimisation problem

$$\min_{u \in H_m^1(\Omega)} \left(\sum_{i=1}^N (u(\mathbf{p}_i) - z_i)^2 + \lambda |u|_{H_m^1(\Omega)}^2 \right) \quad (5)$$

The associated inner-product for the semi-norm on H_m^1 -space for $d=2$ is

$$\langle u, v \rangle_{H_m^1(\Omega)} = \int_{\Omega} \nabla u^T \nabla v + \frac{\partial^2 u}{\partial x \partial y} \frac{\partial^2 v}{\partial x \partial y} d\mathbf{x}$$

We can define a bilinear form $b(\cdot, \cdot)$ and a linear form $\ell(\cdot)$ as

$$\begin{aligned} b(u, v) &= (Pu)^T (Pv) + \lambda \langle u, v \rangle_{H_m^1(\Omega)} \\ \ell(v) &= (Pv)^T \mathbf{z} \end{aligned}$$

where

$$Pu = (u(x_1, y_1), u(x_2, y_2), \dots, u(x_N, y_N))^T$$

is a column vector of the function values of u at the scattered points $\mathcal{G} = \{\mathbf{p}_i\}_{i=1}^N$, and $\mathbf{z} \in \mathbb{R}^N$ is a column vector with i th component z_i . Then the multivariate spline problem is to find $u \in V$ such that

$$b(u, v) = \ell(v) \quad (6)$$

for all $v \in V$.

Let Ω be a rectangle in $\mathbb{R}^2$ and $\mathcal{T}_h$ the tensor product partition of the domain with mesh size h , such that each element $T \in \mathcal{T}_h$ is a rectangle. Then we define a finite element space V_h as

$$V_h = \{u_h \in C^0(\Omega) : u_h|_T \in \mathcal{P}(T), T \in \mathcal{T}_h\},$$

where $\mathcal{P}(T)$ is the space of bilinear polynomials on T . We can now write our discrete multivariate spline problem as

$$\min_{u_h \in V_h} \left(\sum_{i=1}^N (u_h(\mathbf{p}_i) - z_i)^2 + \lambda |u_h|_{H_m^1(\Omega)}^2 \right).$$

That is, the discrete problem is to find $u_h \in V_h$ such that

$$b(u_h, v_h) = \ell(v_h) \quad (7)$$

for all $v_h \in V_h$.

The discrete problem is shown to be well-posed in Lamichhane.¹¹ Here we recall some of the important results. We first show that the bilinear form $b(\cdot, \cdot)$ is positive definite on V_h .

Lemma 6. *Let $\lambda > 0$ and let the set of scattered points $\mathcal{G}$ be non-empty. Then the bilinear form $b(\cdot, \cdot)$ is positive definite on the vector space V_h .*

Proof. If $u_h = 0$, then clearly $b(u_h, u_h) = 0$. Conversely, let $b(u_h, u_h) = 0$. Then

$$Pu_h = 0, \quad \nabla u_h = 0, \quad \text{and} \quad \frac{\partial^2 u_h}{\partial x \partial y} = 0.$$

Since u_h is a continuous function, $\nabla u_h = 0$ gives that u is a constant function in Ω . Further, since $\mathcal{G}$ is non-empty and $Pu_h = 0$, we have that $u_h = 0$.

Since $b(\cdot, \cdot)$ is positive definite, we can define the energy norm $\|\cdot\|_b$ on V_h as

$$\|v_h\|_b^2 = b(v_h, v_h)$$

for all $v_h \in V_h$. Since $b(\cdot, \cdot)$ and $\ell(\cdot)$ satisfy the conditions of the Lax-Milgram lemma,^{2,3} the unique minimiser is the solution of the discrete problem (7). In addition, the following holds.

Lemma 7. Let $\lambda > 0$ and let $\mathcal{G}$ be non-empty. Then the discrete problem (7) admits a unique solution which depends continuously on the data with respect to the energy norm $\|\cdot\|_b$.

Proof. We have that

$$|b(u_h, v_h)| \leq \|u_h\|_b \|v_h\|_b, \text{ and } |\ell(v_h)| \leq \|\mathbf{z}\| \|v_h\|_b$$

for all $u_h, v_h \in V_h$. Hence $b(\cdot, \cdot)$ and $\ell(\cdot)$ are continuous on V_h . We also have that

$$b(u_h, u_h) = \|u_h\|_b^2$$

for all $u_h \in V_h$. Hence $b(\cdot, \cdot)$ is coercive on V_h . By the Lax-Milgram lemma,^{2,3} there exists a unique solution u_h of the discrete problem 7. Additionally, the solution depends continuously on the data $\mathbf{z}$.

In addition, a direct application of the Céa lemma provides an optimal *a priori* estimate of the discrete solution.

Lemma 8. Let u be the solution to the continuous problem (6), and let u_h be the solution to the discrete problem (7). Then

$$\|u - u_h\|_b \leq \inf_{v_h \in V_h} \|u - v_h\|_b$$

Each finite element basis function is associated with a point in the tensor product partition $\mathcal{T}_h$. Assuming there are mn points, we have mn basis functions. Let $\{\phi_i\}_{i=1}^{mn}$ be the set of finite element basis functions, which span V_h . Then we can write our solution $u_h \in V_h$ as a linear combination of these basis functions, namely

$$u_h(x, y) = \sum_{i=1}^{mn} u_i \phi_i(x, y)$$

Let $\mathbf{u} = (u_1, u_2, \dots, u_{mn})^T$ and let K be the finite element stiffness matrix, where $K_{ij} = \int_{\Omega} \nabla u_h^T \nabla v_h d\mathbf{x}$. Let M be a mixed partial derivative matrix, where $M_{ij} = \int_{\Omega} \frac{\partial u_h}{\partial x \partial y} \frac{\partial v_h}{\partial x \partial y} d\mathbf{x}$. Then the finite element problem leads to the linear system

$$(A^T A + \lambda(K + M))\mathbf{u} = A^T \mathbf{z}$$

where A is a matrix of size $N \times mn$, with entries $A_{ij} = \phi_j(\mathbf{p}_i)$.

Computation of K and M

We will use a reference element to construct the matrices K and M . For rectangles, we choose our reference

element $\hat{T}$ to be the square with vertices $(0, 0), (1, 0), (1, 1)$, and $(0, 1)$. We then construct local basis functions at these vertices. These are

$$\begin{aligned} \hat{\phi}_1(\xi, \eta) &= (1 - \xi)(1 - \eta), \\ \hat{\phi}_2(\xi, \eta) &= \xi(1 - \eta), \\ \hat{\phi}_3(\xi, \eta) &= \xi\eta, \\ \hat{\phi}_4(\xi, \eta) &= (1 - \xi)\eta \end{aligned}$$

Now consider the rectangular element T with vertices $(x_1, y_1), (x_2, y_1), (x_2, y_2)$, and (x_1, y_2) . We construct global basis functions at these vertices. When restricted to the element, these are given by

$$\begin{aligned} \phi_1(x, y) &= \frac{(y - y_2)(x - x_2)}{(y_1 - y_2)(x_1 - x_2)}, \\ \phi_2(x, y) &= \frac{(y - y_2)(x - x_1)}{(y_1 - y_2)(x_2 - x_1)}, \\ \phi_3(x, y) &= \frac{(y - y_1)(x - x_1)}{(y_2 - y_1)(x_2 - x_1)}, \\ \phi_4(x, y) &= \frac{(y - y_1)(x - x_2)}{(y_2 - y_1)(x_1 - x_2)} \end{aligned}$$

Let $F_T: \hat{T} \rightarrow T$ be the bijective map from the reference element to T . This mapping is given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & 0 \\ 0 & y_2 - y_1 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

Let the matrix in this transformation be denoted B_T . We note that $\phi_i = \hat{\phi}_i \circ F_T^{-1}$.

We will use the chain rule to calculate the derivatives. We have

$$\begin{aligned} \frac{\partial \phi_i}{\partial x} &= \frac{1}{b_1} \cdot \frac{\partial \hat{\phi}_i}{\partial \xi} = \frac{1}{b_1} \cdot \frac{\partial \hat{\phi}_i}{\partial \xi} \circ F_T^{-1}, \\ \frac{\partial \phi_i}{\partial y} &= \frac{1}{b_2} \cdot \frac{\partial \hat{\phi}_i}{\partial \eta} = \frac{1}{b_2} \cdot \frac{\partial \hat{\phi}_i}{\partial \eta} \circ F_T^{-1} \end{aligned}$$

where $b_1 = x_2 - x_1$ and $b_2 = y_2 - y_1$.

Applying the chain rule again we obtain

$$\frac{\partial^2 \phi_i}{\partial x \partial y} = \frac{1}{b_1 b_2} \cdot \frac{\partial^2 \hat{\phi}_i}{\partial \xi \partial \eta} = \frac{1}{b_1 b_2} \cdot \frac{\partial^2 \hat{\phi}_i}{\partial \xi \partial \eta} \circ F_T^{-1}$$

Now, we'll calculate K . Let $K_T \in \mathbb{R}^{4 \times 4}$ be the local stiffness matrix associated with the element T . Then

$$(K_T)_{ij} = \int_T \frac{\partial \phi_i}{\partial x} \cdot \frac{\partial \phi_j}{\partial x} + \frac{\partial \phi_i}{\partial y} \cdot \frac{\partial \phi_j}{\partial y} d\mathbf{x}$$

Now

$$\int_T \frac{\partial \phi_i}{\partial x} \cdot \frac{\partial \phi_j}{\partial x} d\mathbf{x} = \int_{\hat{T}} \left(\frac{\partial \phi_i}{\partial x} \circ F_T \right) \left(\frac{\partial \phi_j}{\partial x} \circ F_T \right) |\det B_T| d\hat{\mathbf{x}} = \frac{|\det B_T|}{b_1^2} \int_{\hat{T}} \frac{\partial \hat{\phi}_i}{\partial \xi} \cdot \frac{\partial \hat{\phi}_j}{\partial \xi} d\hat{\mathbf{x}} = \frac{y_2 - y_1}{x_2 - x_1} \int_{\hat{T}} \frac{\partial \hat{\phi}_i}{\partial \xi} \cdot \frac{\partial \hat{\phi}_j}{\partial \xi} d\hat{\mathbf{x}}$$

Similarly

$$\int_T \frac{\partial \phi_i}{\partial y} \cdot \frac{\partial \phi_j}{\partial y} d\mathbf{x} = \frac{x_2 - x_1}{y_2 - y_1} \int_{\hat{T}} \frac{\partial \hat{\phi}_i}{\partial \eta} \cdot \frac{\partial \hat{\phi}_j}{\partial \eta} d\hat{\mathbf{x}}$$

Hence we have

$$K_T = \frac{y_2 - y_1}{6(x_2 - x_1)} \begin{bmatrix} 2 & -2 & -1 & 1 \\ -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ 1 & -1 & -2 & 2 \end{bmatrix} + \frac{x_2 - x_1}{6(y_2 - y_1)} \begin{bmatrix} 2 & 1 & -1 & -2 \\ 1 & 2 & -2 & -1 \\ -1 & -2 & 2 & 1 \\ -2 & -1 & 1 & 2 \end{bmatrix}$$

We then assemble each K_T into the global stiffness matrix, by relating the local nodal numbering to the global numbering. Let the global numbering of the vertices of T be i, j, k, ℓ . Then the local matrix K_T will be stored in the submatrix $K[i, j, k, \ell]$ of the global matrix K . (Note here that the submatrix is formed by keeping the i th, j th, k th and ℓ th rows and columns of the matrix K .) The global matrix is obtained by adding all the contributions from the local matrices.

We now calculate the mixed partial derivative matrix M . Let $M_T \in \mathbb{R}^{4 \times 4}$ be the local matrix associated with the element T . Then

$$\begin{aligned} (M_T)_{ij} &= \int_T \frac{\partial^2 \phi_i}{\partial x \partial y} \cdot \frac{\partial^2 \phi_j}{\partial x \partial y} d\mathbf{x} \\ &= \int_{\hat{T}} \left(\frac{\partial^2 \phi_i}{\partial x \partial y} \circ F_T \right) \cdot \left(\frac{\partial^2 \phi_j}{\partial x \partial y} \circ F_T \right) |\det B_T| d\hat{\mathbf{x}} \\ &= \frac{|\det B_T|}{(b_1 b_2)^2} \int_{\hat{T}} \frac{\partial^2 \hat{\phi}_i}{\partial \xi \partial \eta} \cdot \frac{\partial^2 \hat{\phi}_j}{\partial \xi \partial \eta} d\hat{\mathbf{x}} \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} \int_{\hat{T}} \frac{\partial^2 \hat{\phi}_i}{\partial \xi \partial \eta} \cdot \frac{\partial^2 \hat{\phi}_j}{\partial \xi \partial \eta} d\hat{\mathbf{x}} \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} \frac{\partial^2 \hat{\phi}_i}{\partial \xi \partial \eta} \cdot \frac{\partial^2 \hat{\phi}_j}{\partial \xi \partial \eta} \int_{\hat{T}} d\hat{\mathbf{x}} \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} \frac{\partial^2 \hat{\phi}_i}{\partial \xi \partial \eta} \cdot \frac{\partial^2 \hat{\phi}_j}{\partial \xi \partial \eta} \end{aligned}$$

Hence we have

$$M_T = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

We then assemble each M_T into the global matrix M in the same way that the stiffness matrix was assembled.

Numerical results

Real life images

We would like to recover some real life images. Consider an image of size $m \times n$. Then we define a tensor product partition $\mathcal{T}_h$ of the square $[0, 1] \times [0, 1]$ using the collection of points $\mathcal{N}_h = \{(a_i, b_j)\}_{i=1, j=1}^{n, m}$, where $a_i = \frac{i-1}{n-1}$, and $b_j = \frac{j-1}{m-1}$. Then each pixel of the image is associated with a grid point in $\mathcal{N}_h$.

Since we know the images before they have noise applied to them, we will use peak signal-to-noise ratio (PSNR) to compare the results. Let the original image be given by I , and the recovered image be given by $\hat{I}$. Then the PSNR is given by

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) = 20 \log_{10} \left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right)$$

where MAX_I is the maximum pixel value of the image, and MSE is the mean square error. We note that MSE is given by

$$\text{MSE} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \|I_{ij} - \hat{I}_{ij}\|^2.$$

We now consider two test images. These images are the Lena image and Baboon image (see Figure 1). We will apply both Gaussian and impulsive noise to these images. The Gaussian noise has zero mean and variances 0.05 and 0.1, and the salt and pepper noise has densities from 30% through to 80%.

We will now use the three different splines to reconstruct the images. As an example, we will first consider



Figure 1. Lena image (a) and Baboon image (b).

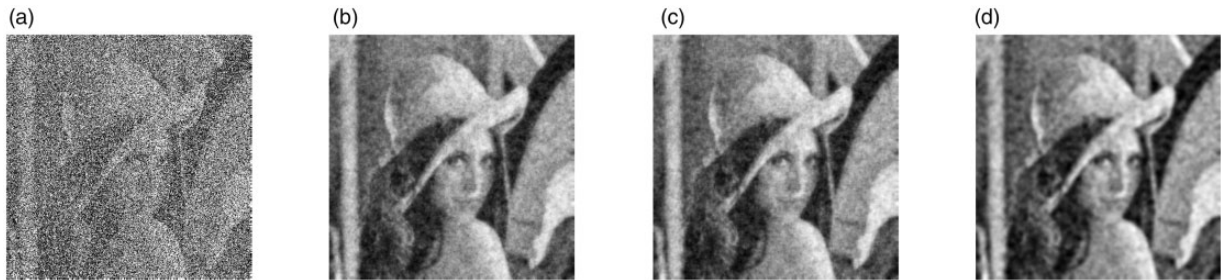


Figure 2. Noisy Lena image ($\sigma^2 = 0.05$, 60% salt and pepper noise density) (a), recovered image using gradient penalty spline (b), recovered image using mixed derivative spline (c), recovered image using Laplacian spline (d).

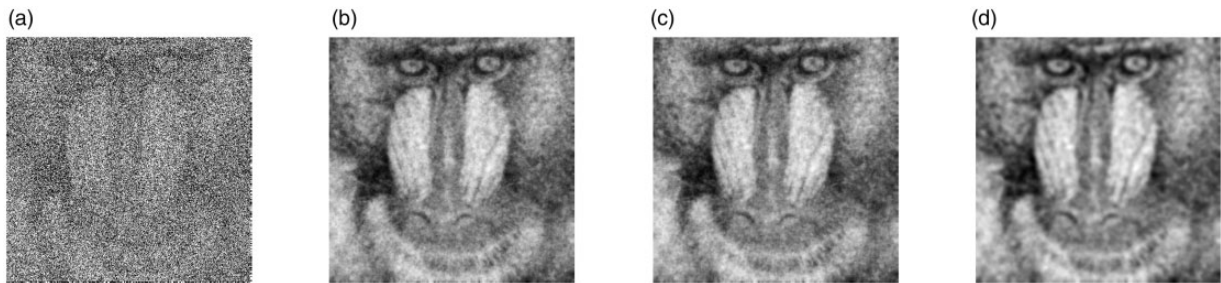


Figure 3. Noisy Baboon image ($\sigma^2 = 0.05$, 60% salt and pepper noise density) (a), recovered image using gradient penalty spline (b), recovered image using mixed derivative spline (c), recovered image using Laplacian spline (d).

Table 1. Lena image PSNR for Gaussian (variance 0.05) and different impulsive noise densities.

	Lena image PSNR					
	Noise density					
	30%	40%	50%	60%	70%	80%
Grad.	22.13	22.01	22.22	21.90	21.42	20.91
Mixed	22.29	22.28	22.42	21.93	21.59	20.80
Biharm.	22.98	22.95	22.30	22.09	21.74	21.48

Table 2. Baboon image PSNR for Gaussian (variance 0.05) and different impulsive noise densities.

	Baboon image PSNR					
	Noise density					
	30%	40%	50%	60%	70%	80%
Grad.	19.14	19.09	18.81	18.48	18.47	18.37
Mixed	19.14	19.08	18.74	18.48	18.35	18.32
Biharm.	18.89	18.71	18.47	18.12	17.84	17.83

Table 3. Lena image PSNR for Gaussian (variance 0.1) and different impulsive noise densities.

Lena image PSNR						
	Noise density					
	30%	40%	50%	60%	70%	80%
Grad.	21.45	21.40	20.81	20.34	20.21	19.82
Mixed	21.77	21.69	20.80	20.44	20.21	19.85
Biharm.	22.20	22.09	21.56	21.10	20.48	20.14

Table 4. Baboon image PSNR for Gaussian (variance 0.1) and different impulsive noise densities.

Baboon image PSNR						
	Noise density					
	30%	40%	50%	60%	70%	80%
Grad.	18.24	18.20	17.93	18.09	17.89	17.65
Mixed	18.18	18.17	17.89	18.06	17.86	17.56
Biharm.	18.06	17.85	17.66	17.54	17.40	17.06

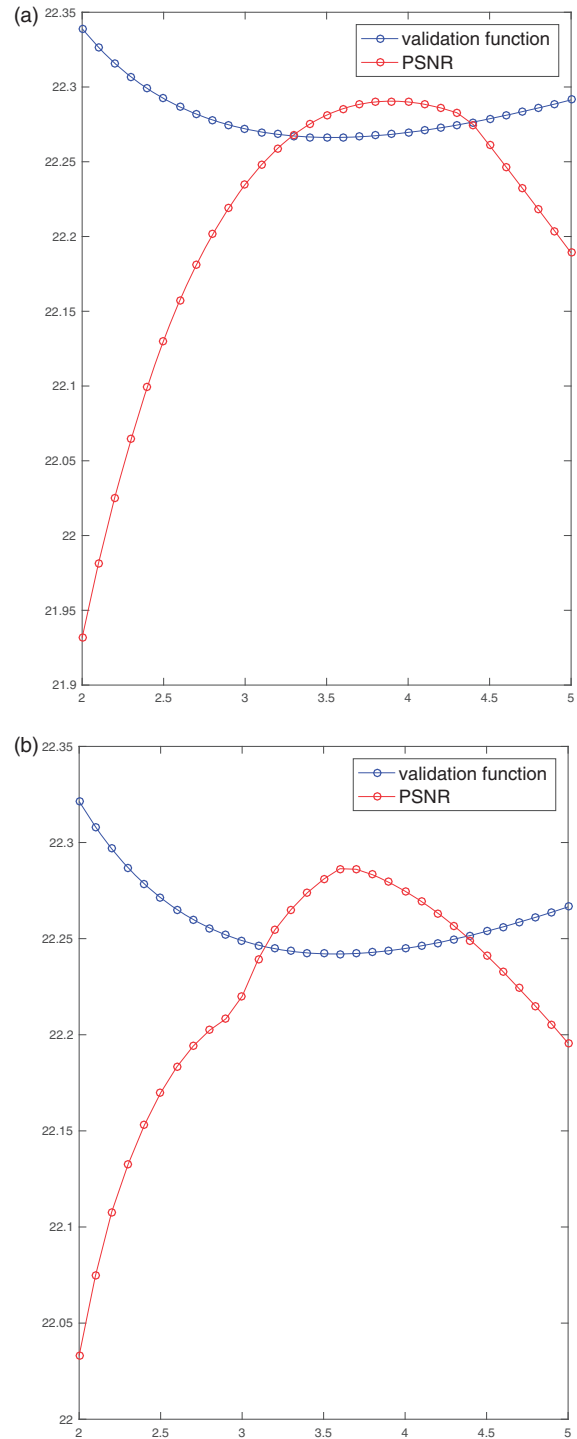
the images corrupted with Gaussian noise with variance 0.05, and impulsive noise of density 60%. In the first image of Figure 2, we show the noisy Lena image. The next three images show the reconstructed images obtained by the three splines. The results for the Baboon image are shown in Figure 3.

We will now show the PSNR for the reconstructed images in Tables 1 to 4.

Note that we have chosen our parameter λ using generalised cross validation¹⁷ and the stochastic trace estimator proposed by Hutchinson.¹⁸ We note that this gives a good estimate of the optimal parameter. In Figure 4 we have plotted the PSNR and the generalised cross validation function versus λ . Note that the validation function has been scaled for visualisation purposes. For both plots, the Lena image has been corrupted with Gaussian noise with variance 0.05, and has been recovered with the mixed derivative spline. In the left plot, the image has been corrupted with impulsive noise of density 30% while in the right plot the density is 40%.

Binary image

We will now apply the same methods to a binary test image (see Figure 5). As an example, consider the image is corrupted with Gaussian noise of variance 0.05, and impulsive noise of density 60%. We show

**Figure 4.** Generalised cross validation function and PSNR versus λ for Gaussian noise with variance 0.05 and impulsive noise with densities 30% (a) and 40% (b).

the noisy image and the reconstructed images in Figure 6.

We will now show the PSNR for the reconstructed image in Tables 5 and 6.

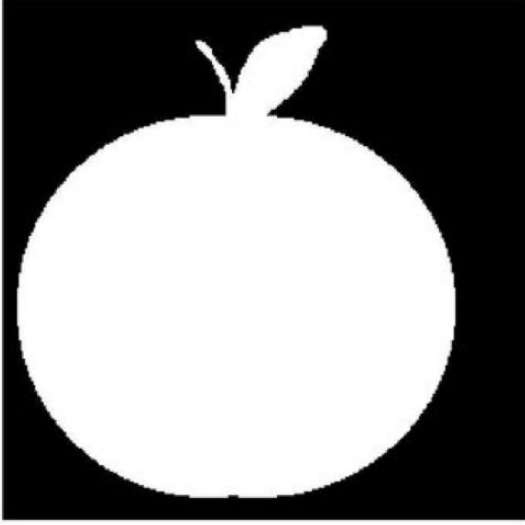


Figure 5. Binary image.

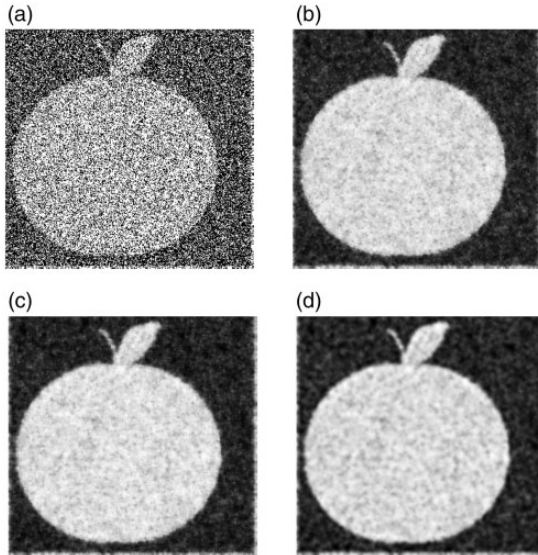


Figure 6. Noisy binary image ($\sigma^2 = 0.05$, 60% salt and pepper noise density) (a), recovered image using gradient penalty spline (b), recovered image using mixed derivative spline (c), recovered image using Laplacian spline (d).

Continuous functions

We would now like to recover continuous functions. We define a tensor product partition $\mathcal{T}_h$ of the square $[-1, 1] \times [-1, 1]$ using the set of points

$$\mathcal{N}_h = \{(a_i, b_j)\}_{i=1, j=1}^{n, m},$$

where $a_i = \frac{2(i-1)}{n-1} - 1$, and $b_j = \frac{2(j-1)}{m-1} - 1$

Table 5. Binary image PSNR for Gaussian (variance 0.05) and different impulsive noise densities.

Binary image PSNR						
Noise density						
	30%	40%	50%	60%	70%	80%
Grad.	14.77	14.66	14.37	14.19	13.77	13.47
Mixed	15.37	15.21	15.03	14.84	14.43	14.00
Biharm.	15.17	14.93	14.86	14.49	14.39	13.72

Table 6. Binary image PSNR for Gaussian (variance 0.1) and different impulsive noise densities.

Binary image PSNR						
Noise density						
	30%	40%	50%	60%	70%	80%
Grad.	13.24	13.05	13.01	12.61	12.49	12.20
Mixed	13.58	13.38	13.27	13.03	12.86	12.50
Biharm.	13.36	13.10	12.23	12.88	12.89	12.56

We sample the function value at each point in the partition, and then apply Gaussian noise of variance 0.05. We then refine the partition several times, which halves the mesh size h in each iteration. We will now consider the first test function. Let the function f be given by $f(x, y) = \sin(3x)e^{-x^2-y^2}$ over the domain $[-1, 1] \times [-1, 1]$ (see Figure 7).

We compare the PSNR values for the recovered function and the original function for different steps of refinement in Table 7, where the refinement step is given by the step-size h .

We can see that PSNR values do not increase or decrease for the spline with the mixed derivative penalty and the spline with the Laplacian penalty, whereas the PSNR values decrease for the spline with the gradient penalty. This is due to the fact that the stability constant depends on the mesh-size h for the spline with the gradient penalty.

We show the functions recovered after the fifth iteration in Figure 7. We can see that gradient penalty spline produces a recovered function that overfits the noisy data. On the other hand, both the mixed derivative and the Laplacian splines produce smoother recovered functions.

We want to see the effect of the mesh-size on the spurious spikes of the recovered function. In Figure 8, we show the functions recovered using the gradient penalty spline using the coarser mesh-sizes $h=2/19$ and $h=1/19$. These pictures

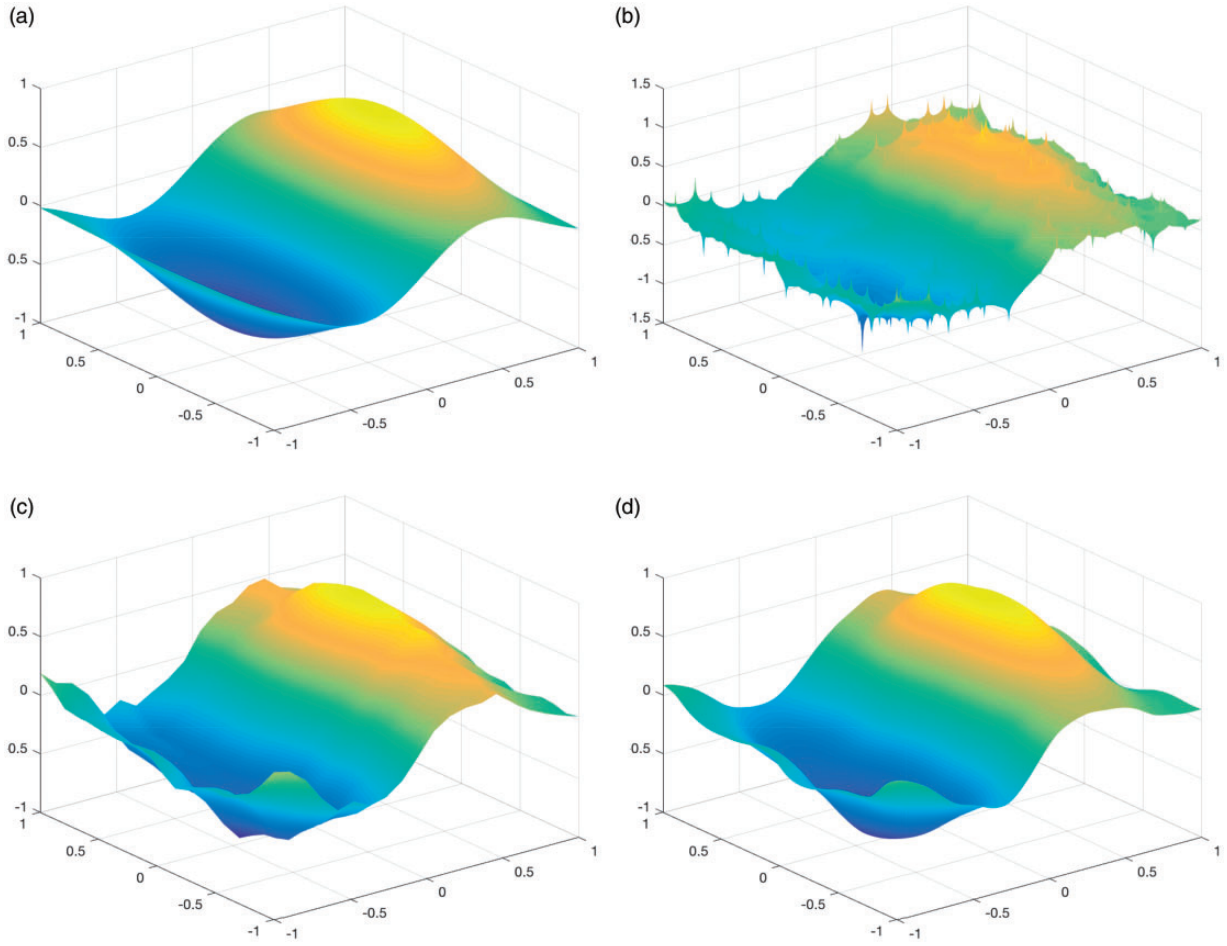


Figure 7. $f(x, y) = \sin(3x)e^{-x^2-y^2}$ restricted to $[-1, 1] \times [-1, 1]$ (a), function recovered using gradient penalty spline (b), function recovered using mixed derivative spline (c), function recovered using Laplacian spline (d).

Table 7. PSNR for f using different penalty terms.

i	h	Grad.	Mixed	Bihar.
0	2/19	19.85	24.31	25.54
1	1/19	20.01	24.48	26.44
2	1/38	19.59	24.61	26.61
3	1/76	19.16	24.67	26.69
4	1/152	18.51	24.70	26.72
5	1/304	17.95	24.71	26.73

show that the spurious spikes are still present although the spikes are slightly smaller in the coarser mesh results.

We will now provide a second test function. Let the function g be given by $g(x, y) = -x^2 - xy^2$ over the domain $[-1, 1] \times [-1, 1]$ (see Figure 9). We have tabulated the PSNR values for different splines at different levels of refinement in Table 8. The results are similar to the first example but the spurious spikes of the recovered function using the gradient penalty

formulation have not affected the PSNR values much in this example.

We show the functions recovered after the fifth iteration in Figure 9. Again, we see that the gradient penalty spline overfits the data.

Discussion

We compared three different bivariate L-spline approaches for removing the mixture of Gaussian and impulsive noise from images. We found that for the Lena image, the Laplacian penalty produced recovered images with the best PSNR. However, we found that the Laplacian penalty performed the worst when recovering the Baboon image. The gradient and mixed derivative penalties performed very similarly to each other when recovering the two real life images. For the Binary image, we found that the mixed derivative penalty performed the best, followed by the Laplacian penalty and then the gradient penalty.

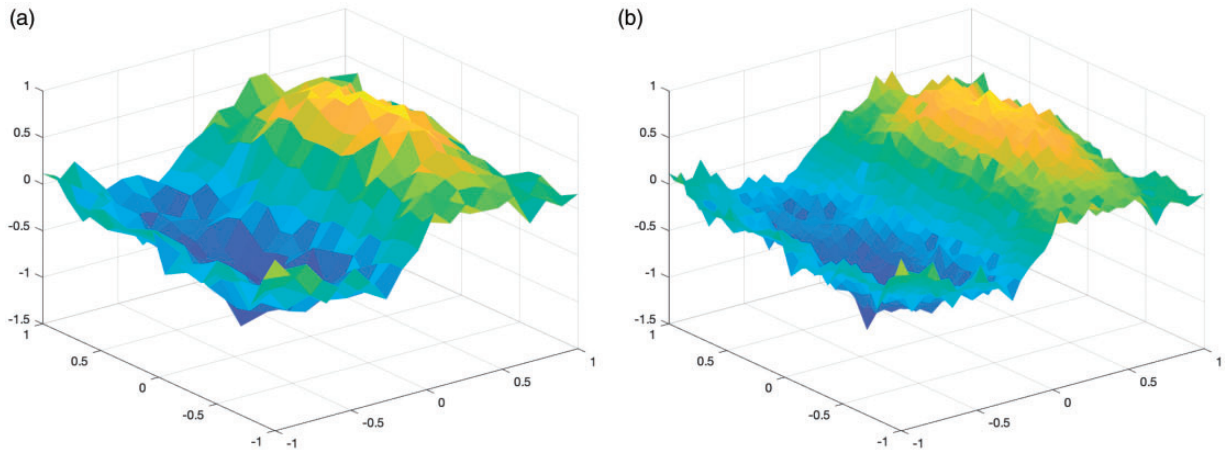


Figure 8. Function recovered using gradient penalty spline with $h = 2/19$ (a), function recovered using gradient penalty spline with $h = 1/19$ (b).

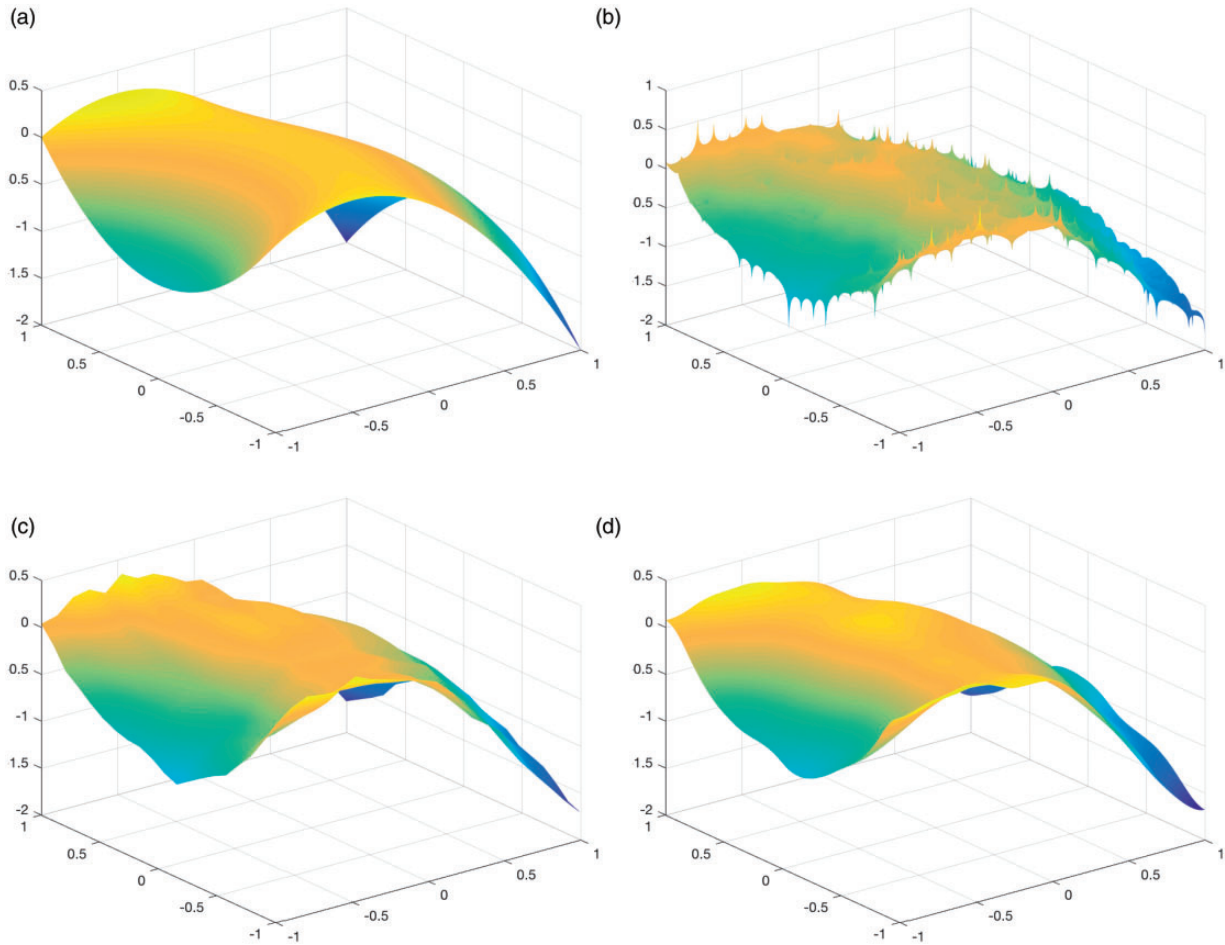


Figure 9. $g(x, y) = -x^2 - xy^2$ restricted to $[-1, 1] \times [-1, 1]$ (a), function recovered using gradient penalty spline (b), function recovered using mixed derivative spline (c), function recovered using Laplacian spline (d).

Table 8. PSNR for g using different penalty terms.

i	h	Grad.	Mixed	Biharm.
0	2/19	19.49	21.24	22.19
1	1/19	19.01	21.17	22.29
2	1/38	18.76	21.15	22.33
3	1/76	18.33	21.13	22.31
4	1/152	18.16	21.12	22.30
5	1/304	18.82	21.11	22.29

We then applied the same approaches to recover two continuous functions from a set of noisy observations. We found that for both functions, the Laplacian penalty produced the best recovered functions, closely followed by the mixed derivative penalty. The gradient penalty produced recovered functions that overfitted the data.

The overfitting occurred because the gradient penalty formulation is not well-posed in the continuous setting. For dimensions $d \geq 2$, we have that $H^1(\Omega) \not\subset C^0(\Omega)$ (by the Sobolev embedding theorem). This ill-posedness exhibits itself when the mesh size goes to zero.¹⁹ The other formulations, however, are well-posed in the continuous setting. This is because $H_m^1(\Omega) \subset C^0(\Omega)$ for any dimension d ,¹⁶ and $H^2(\Omega) \subset C^0(\Omega)$ for dimensions $d \leq 3$.

Overall, the gradient penalty was the simplest spline to implement and the most computationally efficient. However, as this is not well-posed for $d > 1$, it often produces spurious results. The computational cost of the spline with the mixed derivative penalty is very close to the gradient penalty and this is well-posed for all dimensions.¹¹ The Laplacian penalty was the least simple to implement, and was the least computationally efficient. Moreover, the spline with the Laplacian penalty is also not well-posed when $d > 3$. Therefore, we find that the spline with the mixed derivative penalty is the best choice among the presented three splines.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Bishnu P Lamichhane  <https://orcid.org/0000-0002-9184-8941>

References

1. Adams R. *Sobolev spaces*. New York: Academic Press, 1975.

2. Brenner S and Scott L. *The mathematical theory of finite element methods*. New York: Springer-Verlag, 1994.
3. Ciarlet P. *The finite element method for elliptic problems*. North Holland, Amsterdam, 1978.
4. Lamichhane B, Roberts S and Hegland M. A new multivariate spline based on mixed partial derivatives and its finite element approximation. *Appl Math Lett* 2014; 35: 82–85.
5. Preußner T and Rumpf M. An adaptive finite element method for large scale image processing. *J Visual Commun Image Represent* 2000; 11: 183–195.
6. Ferrant M, Nabavi A, Macq B, et al. Registration of 3-d intraoperative MR images of the brain using a finite-element biomechanical model. *IEEE Trans Med Imaging* 2001; 20: 1384–1397.
7. Besdok E. Impulsive noise suppression from images by using Anfis interpolant and Lillietest. *EURASIP J Appl Signal Process* 2004; 2423–2433.
8. Wang Z, Qi F and Zhou F. 2006. A discontinuous finite element method for image denoising. In: *Image analysis and recognition, lecture notes in computer science*. Vol. 4141. Berlin/Heidelberg: Springer, pp. 116–125.
9. Demaret L and Iske A. Adaptive image approximation by linear splines over locally optimal Delaunay triangulations. *IEEE Signal Process Lett* 2006; 13: 281–284.
10. Lamichhane B. Finite element techniques for removing the mixture of Gaussian and impulsive noise. *IEEE Trans Signal Process* 2009; 57: 2538–2547.
11. Lamichhane BP. Removing a mixture of Gaussian and impulsive noise using the total variation functional and split Bregman iterative method. In: Sharples J and Bunder J (eds) *Proceedings of the 17th biennial computational techniques and applications conference, CTAC-2014, ANZIAM J*, vol. 56, 2015, pp. C52–C67, <http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/9316> (accessed 27 October 2015).
12. Quarteroni A and Valli A. *Numerical approximation of partial differential equations*. Berlin: Springer-Verlag, 1994.
13. Garcke J and Hegland M. Fitting multidimensional data using gradient penalties and the sparse grid combination technique. *Computing* 2009; 84: 1–25.
14. Ramsay T. Spline smoothing over difficult regions. *J R Stat Soc Series B* 2002; 64: 307–319.
15. Lamichhane B. A mixed finite element method for the biharmonic problem using biorthogonal or quasi-biorthogonal systems. *J Sci Comput* 2011; 46: 379–396.
16. Schmeisser HJ and Triebel H. *Topics in Fourier analysis and function spaces*. 1st edition. Chichester: Wiley, 1987.
17. Wahba G. *Spline models for observational data. Series in applied mathematics*. Vol. 59, 1st ed. Philadelphia: SIAM, 1990.
18. Hutchinson M. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Commun Statist Simul Comput* 1989; 18: 1059–1076.
19. Utreras F. Convergence rates for multivariate smoothing spline functions. *J Approx Theory* 1988; 52: 1–27.