

Constrained smoothing and interpolating spline surfaces using normalized uniform B-splines

HIROYUKI FUJIOKA, HIROYUKI KANO, AND CLYDE F. MARTIN

We consider the problem of constructing optimal smoothing and interpolating spline surfaces with equality and/or inequality constraints. By using normalized uniform B-splines as the basis functions, the problem of constructing optimal surfaces is to determine the so-called control points optimally. In particular, following a similar approach as in the case of curves, we formulate various types of equality and inequality constraints as linear functions of the control points. Included are constraints on the value at isolated points, those over intervals or over regions, or on integral value on a region, and their combinations. Concise expressions are derived for these constraints and it is shown that they can be incorporated easily to smoothing and interpolating spline problems. The splines can be of arbitrary degree, and the problem is reduced to convex quadratic programming (QP) problem. Some efficient algorithms are available for solving the QP problems numerically, thus the proposed method is useful for many applications. The performance is examined by numerical examples of interpolating function with boundary constraints, approximating probability density functions, and of smoothing digital image data.

KEYWORDS AND PHRASES: B-splines, optimal smoothing splines, optimal interpolating splines, equality/inequality constraint, convex quadratic programming.

1. Introduction

The goal of constructing smooth approximations is central to many applications. In statistics much of the credit for the field must go to Grace Wahba and her seminal construction of smoothing splines. Her work has set the stage for many if not most of the advancements that have taken place in the last decade. Recently there has been work in engineering that has rediscovered some of her techniques and has pushed the science further in complimentary directions. The work in engineering has had a different goal than the primary work in statistics. The work as described in [6] is aimed

more at path planning than in data analysis. The work by H. Kano and his group has been directed at obtaining realistic depictions of curves and surfaces using a few data points. The original work of C. Martin and his collaborators was motivated by the question of how many points must be stored to reproduce typographical fonts. However as the work of Kano and Martin developed it became clear that it was necessary to understand the relations with the existing statistical literature.

A natural problem common to both the engineering and statistical work was the construction of shape preserving approximations. The authors in [16] showed a method of dealing with this problem on spline curves using the method of B-splines. Other work in this direction is contained in [3, 22] and a chapter on monotone smoothing splines in [6]. It would be remiss not to mention the new book of Yuedong Wang, [30]. It is an up to date summary of what is known in the statistical literature.

On the other hand, the problem of constructing curves and surfaces from a given set of discrete observational data arises in many other fields – such as numerical analysis, computer aided design (CAD), computer graphics (CG), computer vision, robotics, and image processing, etc. Thus, the spline functions have been studied and used extensively (e.g. [2, 29, 32]). Moreover, in addition to traditional approximating or interpolating splines, there are a large class of problems where we need to impose various constraints on splines. For example, when the observational data is considerably corrupted by some noises, we may have difficulties to maintain the shape properties of the underlying functions by splines without imposing any constraints. In addition, if such observational data is sparsely distributed in some interval or domain of interest, the difficulties may increase more and more. Thus, many researchers have recently dealt with the issues of constrained splines. Monotone and convex splines form important class of shape preserving splines, where the first and second derivatives are kept e.g. nonnegative over a given interval respectively. Examples of other constraints are as follows: Boundary conditions as initial and terminal conditions are usually specified as equalities at the boundaries of interval, the so-called interval interpolation is specified as inequality at given points, probability density function is approximated by splines under unit area constraint, etc.

Specifically, Egerstedt and Martin in [21] have developed the method of constructing smoothing spline curves with inequality constraints at isolated points by employing the control theoretic approach. An advantage of this approach is that, by an appropriate choice of the system matrix, various types of functions can be used for splines – such as exponential functions, trigonometric functions, polynomial functions and their combinations. This

method is used for monotone splines [5], but in general, such a problem with constraints over interval leads to an infinite dimensional problem and not be easily solved. Thus, they formulated and solved the monotone spline problem as a dynamic programming problem but the method is specific to the cubic splines. Meyer in [23] has not only constructed monotone splines by using M-splines [26] but also extended the construction to the case of convex constraints. Elfving and Anderson in [7], treated a combination of monotonicity and convexity constraints simultaneously. Both constructions in [23] and [7] are however limited to the cubic splines. Employing B-spline approach, Kano et al. in [15, 16] have developed a method for designing smoothing spline curves with various types of equality and/or inequality constraints in a unified framework. The constraints can be imposed on function value or its derivatives at isolated point or over interval, and on integral values, and moreover they can be imposed in combinations. Shape preserving splines as monotone or convex splines are easily incorporated into this framework by imposing the sign of first and second derivatives e.g. nonnegative over an interval. Also note that the splines are treated as of arbitrary degree.

While the above work is for spline curves, similar problems for the spline surface have been studied by various authors. Beatson and Ziegler in [1] have developed an algorithm for constructing monotone surfaces for a given set of data on some domain. It is particularly worth noting that the necessary and sufficient conditions for the spline surface to be monotone have been analyzed and derived, but it is also limited to the case of quadratic splines. Although we find a lot of similar works (see e.g. [8, 11, 12, 13, 17]), they are also specific to the cubic case or lower. From the viewpoints of constrained splines, the types of constraints are restricted to monotonicity and convexity in these studies. As for the control theoretic splines with constraints, Maad et al. in [19] developed a method of fitting spline surfaces with boundary conditions which arise in the problem of toxicology and testing drugs. On the other hand, some applications using the constrained spline surfaces often arise in various problems – such as estimation of density functions which arises from bivariate histograms (see e.g. [4]), and skinning method which is frequently used in CAD and CG modeling (see e.g. [31, 24]). These surfaces have generally been constructed by employing B-spline interpolating methods. When the data is corrupted by noises, the smoothing counterpart of this method will be desirable. Also, inequality constraints at a finite number of isolated points are considered in [28]. To the authors' knowledge, however, there has been no study that deals with smoothing spline surfaces of arbitrary degree providing a framework where various types of constraints can be incorporated as they are required. Possible constraints types are on

points, over lines (as boundary conditions), or over region (as monotonicity, convexity, or integral values), and moreover they may better be used in combination either as equalities or inequalities. This gives a motivation for the present study. It is an extension of our results on curves [15] to surfaces and moreover with an intension of extending further as trivariate splines and n -variate splines in general.

Specifically, in this study, we develop a systematic method for constructing optimal smoothing and interpolating spline surfaces with equality and/or inequality constraints. The splines are constructed by using normalized uniform B-splines as the basis functions. Then, various types of constraints are formulated as linear function of the so-called control points, and the problems are reduced to convex quadratic programming problem. Following the line of approach given in [15], the central issues here are firstly the derivation of basic formula e.g. of partial derivatives of arbitrary degree, secondly how to formulate various constraints in terms of control point vector originally given as matrix, and after all to develop an algorithm in the form where the QP problem solver as the function ‘quadprog’ in MATLAB can be readily applied. The performance is examined by numerical examples of interpolating given function with boundary conditions, approximating probability density functions, and of smoothing operation for digital image data.

For designing surfaces $x(s, t)$, we employ normalized, uniform B-spline function $B_k(t)$ of degree k as the basis functions,

$$(1) \quad x(s, t) = \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} \tau_{i,j} B_k(\alpha(s - s_i)) B_k(\beta(t - t_j))$$

on a domain $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}] \subset \mathbf{R}^2$. Here, $\tau_{i,j}$ are the weighting coefficients called control points, $\alpha, \beta (> 0)$ are constants, $m_1, m_2 (> 2)$ are integers, and s_i 's, t_j 's are equally spaced knot points with

$$(2) \quad s_{i+1} - s_i = \frac{1}{\alpha}, \quad t_{j+1} - t_j = \frac{1}{\beta}.$$

Remark 1. *We define the spline surface $x(s, t)$ on $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}]$, not on $[s_{-k}, s_{m_1+k}] \times [t_{-k}, t_{m_2+k}]$ where $x(s, t)$ takes nonzero values, since the spline $x(s, t)$ of degree k is constituted from k B-spline bases only on the smaller region \mathcal{D} . This difference arises since we are using uniform B-splines as the basis.*

We summarize some of the symbols that will be used throughout the paper: $\nabla^2 = \frac{\partial^2}{\partial s^2} + \frac{\partial^2}{\partial t^2}$ denotes the Laplacian operator, and \otimes the Kronecker

Table 1: The basis elements $N_{j,k}(t)$ ($j = 0, 1, \dots, k$) for $k = 0, 1, \dots, 5$

	$N_{j,0}(t)$	$N_{j,1}(t)$	$(2!) \times N_{j,2}(t)$	$(3!) \times N_{j,3}(t)$
$j = 0$	1	$1 - t$	$(1 - t)^2$	$(1 - t)^3$
$j = 1$		t	$1 + 2t - 2t^2$	$4 - 6t^2 + 3t^3$
$j = 2$			t^2	$1 + 3t + 3t^2 - 3t^3$
$j = 3$				t^3

	$(4!) \times N_{j,4}(t)$	$(5!) \times N_{j,5}(t)$
$j = 0$	$(1 - t)^4$	$(1 - t)^5$
$j = 1$	$11 - 12t - 6t^2 + 12t^3 - 4t^4$	$26 - 50t + 20t^2 + 20t^3 - 20t^4 + 5t^5$
$j = 2$	$11 + 12t - 6t^2 - 12t^3 + 6t^4$	$66 - 60t^2 + 30t^4 - 10t^5$
$j = 3$	$1 + 4t + 6t^2 + 4t^3 - 4t^4$	$26 + 50t + 20t^2 - 20t^3 - 20t^4 + 10t^5$
$j = 4$	t^4	$1 + 5t + 10t^2 + 10t^3 + 5t^4 - 5t^5$
$j = 5$		t^5

Then $N_k(t)$ is written as

$$(8) \quad N_k(t) = S_k h_k(t),$$

where $S_k \in \mathbf{R}^{(k+1) \times (k+1)}$ is a matrix whose i -th row consists of the coefficients of the polynomial $N_{i-1,k}(t)$. When $k = 3$, for example, we obtain the matrix S_3 from Table 1 as

$$(9) \quad S_3 = \frac{1}{3!} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Differentiating $h_k(t)$ in (7), we get

$$(10) \quad h_k^{(1)}(t) = C_k h_{k-1}(t), \quad h_k^{(2)}(t) = C_k C_{k-1} h_{k-2}(t), \quad \dots$$

where $C_k \in \mathbf{R}^{(k+1) \times k}$ is defined by

$$(11) \quad C_k = \begin{bmatrix} k & & & \\ & k-1 & & \\ & & \ddots & \\ & & & 1 \\ 0 & \dots & & 0 \end{bmatrix}.$$

In (11), empty spaces denote zero entries. We write the i -th derivative $h_k^{(i)}(t)$ as

$$(12) \quad h_k^{(i)}(t) = C_{k,i} h_{k-i}(t), \quad i = 1, 2, \dots, k,$$

where $C_{k,i} \in \mathbf{R}^{(k+1) \times (k-(i-1))}$ is defined by

$$(13) \quad C_{k,i} = C_k C_{k-1} \cdots C_{k-(i-1)}.$$

2.2. Optimal smoothing and interpolating surfaces

The control points $\tau_{i,j}$ in (1) may be determined by the theory of smoothing splines. Here we briefly describe the cost functions for the optimal design.

Suppose that we are given a set of data $\{d_{ij}\}$ at points $(u_i, v_j) \in \mathcal{D}$ as

$$(14) \quad \{(u_i, v_j; d_{ij}) : u_i \in [s_0, s_{m_1}], v_j \in [t_0, t_{m_2}], d_{ij} \in \mathbf{R}, \\ i = 1, 2, \dots, N_1, j = 1, 2, \dots, N_2\}$$

and let $\tau \in \mathbf{R}^{M_1 \times M_2}$ be the control point matrix defined by

$$(15) \quad \tau = \begin{bmatrix} \tau_{-k,-k} & \tau_{-k,-k+1} & \cdots & \tau_{-k,m_2-1} \\ \tau_{-k+1,-k} & \tau_{-k+1,-k+1} & \cdots & \tau_{-k+1,m_2-1} \\ \vdots & \vdots & \cdots & \vdots \\ \tau_{m_1-1,-k} & \tau_{m_1-1,-k+1} & \cdots & \tau_{m_1-1,m_2-1} \end{bmatrix}$$

with $M_1 = m_1 + k$ and $M_2 = m_2 + k$. Then, given the data set in (14), a standard smoothing spline problem is to find such a spline $x(s, t)$ in (1) or the associated matrix τ in (15) minimizing the following cost function

$$(16) \quad J(\tau) = \lambda \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (\nabla^2 x(s, t))^2 ds dt + \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} w_{ij} (x(u_i, v_j) - d_{ij})^2,$$

where ∇^2 is the Laplacian, $\lambda (> 0)$ is a smoothing parameter, and w_{ij} ($0 \leq w_{ij} \leq 1$) are the weights for approximation errors.

Remark 2. In (16), the smoothing parameter λ can be determined by a method as generalized cross validation method [29], yielding a proper balance between the smoothness of function and goodness of fit. Moreover although a similar method was developed for the case with inequality constraint [28], we need careful examination on applying the method to the present case

with various types of constraints and hence is not used here. A guideline for determining w_{ij} is, for example, based on an observation that larger w_{ij} yields better approximation of data d_{ij} , or putting larger w_{ij} for more reliable data d_{ij} when it contains noise, e.g. by taking w_{ij} inverse proportionally of noise magnitude for d_{ij} .

The above problem can be solved as follows (see e.g. [9] for details): Let $b_1(s) \in \mathbf{R}^{M_1}$ and $b_2(t) \in \mathbf{R}^{M_2}$ be

$$(17) \quad b_1(s) = [B_k(\alpha(s - s_{-k})) \ B_k(\alpha(s - s_{-k+1})) \ \cdots \\ \cdots \ B_k(\alpha(s - s_{m_1-1}))]^T,$$

$$(18) \quad b_2(t) = [B_k(\beta(t - t_{-k})) \ B_k(\beta(t - t_{-k+1})) \ \cdots \\ \cdots \ B_k(\beta(t - t_{m_2-1}))]^T.$$

Then, letting $\hat{\tau} \in \mathbf{R}^{M_1 M_2}$ be a vector defined using the vec function (cf. the last paragraph in Section 1) as

$$(19) \quad \hat{\tau} = \text{vec } \tau,$$

the spline function $x(s, t)$ in (1) is expressed as

$$(20) \quad x(s, t) = (b_2(t) \otimes b_1(s))^T \hat{\tau}.$$

Using the expression in (20), the cost function $J(\tau)$ in (16) can be rewritten as a quadratic function of $\hat{\tau}$,

$$(21) \quad J(\hat{\tau}) = \hat{\tau}^T G \hat{\tau} - 2\hat{\tau}^T g + d^T W d.$$

Here, the matrix $G \in \mathbf{R}^{M_1 M_2 \times M_1 M_2}$ and the vector $g \in \mathbf{R}^{M_1 M_2}$ are given by

$$(22) \quad G = \lambda Q + \Gamma W \Gamma^T$$

$$(23) \quad g = \Gamma W d,$$

where $Q \in \mathbf{R}^{M_1 M_2 \times M_1 M_2}$ is a Gram matrix defined by

$$(24) \quad Q = \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (\nabla^2(b_2(t) \otimes b_1(s))) (\nabla^2(b_2(t) \otimes b_1(s)))^T ds dt.$$

This Gram matrix can be computed easily as shown in [9] once the relevant parameters $k, \alpha, \beta, m_1, m_2$ are given. Moreover, in (22), the matrix $\Gamma \in \mathbf{R}^{M_1 M_2 \times N_1 N_2}$ is defined

$$(25) \quad \Gamma = \bar{B}_2 \otimes \bar{B}_1$$

with matrices $\bar{B}_1 \in \mathbf{R}^{M_1 \times N_1}$ and $\bar{B}_2 \in \mathbf{R}^{M_2 \times N_2}$ defined by

$$(26) \quad \begin{aligned} \bar{B}_1 &= \begin{bmatrix} b_1(u_1) & b_1(u_2) & \cdots & b_1(u_{N_1}) \end{bmatrix}, \\ \bar{B}_2 &= \begin{bmatrix} b_2(v_1) & b_2(v_2) & \cdots & b_2(v_{N_2}) \end{bmatrix}. \end{aligned}$$

Also, $W \in \mathbf{R}^{N_1 N_2 \times N_1 N_2}$ and $d \in \mathbf{R}^{N_1 N_2}$ are given by

$$(27) \quad \begin{aligned} W &= \text{diag}\{w_{11}, w_{21}, \cdots, w_{N_1 1}, \cdots, \\ &\quad w_{1N_2}, w_{2N_2}, \cdots, w_{N_1 N_2}\} \\ d &= [d_{11}, d_{21}, \cdots, d_{N_1 1}, \cdots, \\ &\quad d_{1N_2}, d_{2N_2}, \cdots, d_{N_1 N_2}]^T. \end{aligned}$$

Notice here that $G \geq 0$ (i.e. positive-semidefinite) in (21) since $\lambda > 0$, $Q \geq 0$ and $W \geq 0$, and hence the cost $J(\hat{\tau})$ is a convex function. Thus, if we design the smoothing surfaces without imposing any constraints, the optimal solution $\hat{\tau}$ minimizing the cost function in (16) is given as a solution of

$$(28) \quad G\hat{\tau} = g.$$

For convenience, we summarize an algorithm for constructing optimal smoothing spline surface $x(s, t)$ described above.

Algorithm 2. (*Smoothing spline surface without constraints*) Suppose that we are given a set of data in (14). Then, $x(s, t)$ is constructed in the following steps (S1)–(S6).

- (S1) Set k, α, β, m_1 and m_2 in (1), and let $M_1 = m_1 + k$ and $M_2 = m_2 + k$.
- (S2) Set λ and w_{ij} in (16).
- (S3) Compute the matrices Q in (24), Γ in (25), W in (27) and the vector d in (27).
- (S4) Compute the matrix G in (22) and the vector g in (23).
- (S5) Solve (28) in terms of the control point vector $\hat{\tau}$.
- (S6) Compute the spline $x(s, t)$ in (1).

On the other hand, when we are given a function $f(s, t)$, $s \in [s_0, s_{m_1}]$, $t \in [t_0, t_{m_2}]$ instead of the discrete data in (14), it can be approximated by smoothing splines $x(s, t)$, e.g. by using the cost function given by

$$(29) \quad \begin{aligned} J(\tau) &= \lambda \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (\nabla^2 x(s, t))^2 ds dt \\ &\quad + \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (x(s, t) - f(s, t))^2 ds dt. \end{aligned}$$

Similarly as above, this cost function is rewritten as

$$(30) \quad J(\hat{\tau}) = \hat{\tau}^T G \hat{\tau} - 2\hat{\tau}^T g + f_0,$$

where

$$(31) \quad G = \lambda Q + Q_0$$

$$(32) \quad g = \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} f(s, t) (b_2(t) \otimes b_1(s)) ds dt$$

$$(33) \quad f_0 = \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} f^2(s, t) ds dt$$

with

$$(34) \quad Q_0 = \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (b_2(t) \otimes b_1(s)) (b_2(t) \otimes b_1(s))^T ds dt.$$

It holds that $G > 0$ (positive-definite) in (30) since $Q_0 > 0$ (see [9]), hence the cost $J(\hat{\tau})$ in (30) is strictly convex in $\hat{\tau}$, yielding a unique optimal minimizer. The corresponding smoothing spline $x(s, t)$ is computed by following the similar steps as in Algorithm 2.

In the case of interpolation, the problem of constructing optimal splines is treated as that of minimizing the cost function

$$(35) \quad J(\tau) = \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (\nabla^2 x(s, t))^2 ds dt$$

subject to the equality constraints $x(u_i, v_j) = d_{ij}$, $i = 1, 2, \dots, N_1$, $j = 1, 2, \dots, N_2$. The cost function becomes

$$(36) \quad J(\hat{\tau}) = \hat{\tau}^T Q \hat{\tau}$$

and the constraints are written as

$$(37) \quad \Gamma \hat{\tau} = d$$

with Γ and d given in (25) and (27) respectively. Obviously, the cost $J(\hat{\tau})$ in (36) is also convex since $Q \geq 0$.

Note that this is a special class of interpolation problems since we require the function $x(s, t)$ to be smooth by minimizing the cost (35) in addition to the constraint (37) for conventional interpolation problem. Thus, this can be

Table 2: Physical meaning of constraints, where $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}] \subset \mathbf{R}^2$

constraint type	physical meaning	example
pointwise constraint (Sec. 3.2)	constraint on function and its derivative value at given point $(s, t) \in \mathcal{D}$	$x(s_0, t_0) = 0,$ $\frac{\partial^2}{\partial s \partial t} x(s_0, t_0) \geq 0.$
constraints over knot point interval (Sec. 3.3)	constraint on function value over an interval as $[s_\kappa, s_{\kappa'}]$ for s	$x(s, t_0) = s, \forall s \in [s_0, s_{m_1}].$
constraints over knot point region (Sec. 3.4)	constraint on function value over some region $[s_\kappa, s_{\kappa'}] \times [t_\mu, t_{\mu'}] \subset \mathcal{D}$	$x(s, t) \geq 0, \forall (s, t) \in \mathcal{D}.$
constraints on integral value (Sec. 3.5)	constraint on value of surface volume over domain \mathcal{D}	$\int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} x(s, t) ds dt = 1.$

regarded also as a problem of smoothing spline surfaces $x(s, t)$, and treated in particular as the one with pointwise equality constraints as described in the next section.

3. Optimal spline surfaces with constraints

There are various types of constraints that we need to take into account for constructing spline surfaces. Included are pointwise constraints on $x(s, t)$ and its derivatives, interval constraints on $x(s, t)$ in \mathcal{D} , and constraints on $x(s, t)$ and its integral values over some domains in \mathcal{D} , all as either equality or inequality. Using B-splines approach, we here show that such constraints are formulated as linear functions of the control points. In the sequel, we first develop basic formula in Section 3.1. Then in Sections 3.2–3.5, we show that the basic formula yield expressions for some typical constraints as shown in Table 2 with the physical meanings and examples. These constraints can be mixed as desired and used in wide range of applications. Based on such developments, it is shown that the formulation for optimal spline surfaces with constraints is very suitable for numerical solutions as convex quadratic programming problems in Section 3.6.

3.1. Basic formula

We develop basic formula for expressing the constraints. Noting that $x(s, t)$ is constructed as a product of two piecewise polynomials, we examine the

polynomial in each knot point region $\mathcal{D}_{\kappa,\mu} = [s_\kappa, s_{\kappa+1}) \times [t_\mu, t_{\mu+1})$ for $\kappa = 0, 1, \dots, m_1 - 1$ and $\mu = 0, 1, \dots, m_2 - 1$. For $\mathcal{D}_{\kappa,\mu}$, the function $x(s, t)$ in (1) is written as

$$(38) \quad x(s, t) = \sum_{i=-k+\kappa}^{\kappa} \sum_{j=-k+\mu}^{\mu} \tau_{i,j} B_k(\alpha(s - s_i)) B_k(\beta(t - t_j)),$$

and, by (3), we get

$$(39) \quad x(s, t) = \sum_{i=0}^k \sum_{j=0}^k \tau_{\kappa-k+i, \mu-k+j} N_{i,k}(\alpha(s - s_\kappa)) N_{j,k}(\beta(t - t_\mu)).$$

Then, by introducing new variables u and v defined by

$$(40) \quad u = \alpha(s - s_\kappa), \quad v = \beta(t - t_\mu),$$

the region $\mathcal{D}_{\kappa,\mu}$ is normalized to the unit region $\mathcal{E} = [0, 1) \times [0, 1)$ for (u, v) . Now $x(s, t)$ is expressed in terms of (u, v) as $x(s, t) = \hat{x}(u, v)$ with

$$(41) \quad \hat{x}(u, v) = \sum_{i=0}^k \sum_{j=0}^k \tau_{\kappa-k+i, \mu-k+j} N_{i,k}(u) N_{j,k}(v).$$

Now let $T_{\kappa\mu} \in \mathbf{R}^{(k+1) \times (k+1)}$ be the submatrix of τ in (15) as

$$(42) \quad T_{\kappa\mu} = \begin{bmatrix} \tau_{\kappa-k, \mu-k} & \tau_{\kappa-k, \mu-(k-1)} & \cdots & \tau_{\kappa-k, \mu} \\ \tau_{\kappa-(k-1), \mu-k} & \tau_{\kappa-(k-1), \mu-(k-1)} & \cdots & \tau_{\kappa-(k-1), \mu} \\ \vdots & \vdots & & \vdots \\ \tau_{\kappa, \mu-k} & \tau_{\kappa, \mu-(k-1)} & \cdots & \tau_{\kappa, \mu} \end{bmatrix},$$

then $\hat{x}(u, v)$ in (41) is expressed as

$$(43) \quad \hat{x}(u, v) = N_k^T(u) T_{\kappa\mu} N_k(v) = (N_k(v) \otimes N_k(u))^T \hat{T}_{\kappa\mu}$$

with

$$(44) \quad \hat{T}_{\kappa\mu} = \text{vec } T_{\kappa\mu}.$$

Here we used the formula $\text{vec}(AXB) = (B^T \otimes A)\text{vec}X$ and $(A \otimes B)^T = A^T \otimes B^T$ for Kronecker product. Moreover, by introducing the matrices

$E_\kappa \in \mathbf{R}^{(k+1) \times M_1}$ and $E_\mu \in \mathbf{R}^{(k+1) \times M_2}$ as

$$(45) \quad E_\kappa = [0_{k+1, \kappa} \ I_{k+1} \ 0_{k+1, M_1 - \kappa - (k+1)}], \quad E_\mu = [0_{k+1, \mu} \ I_{k+1} \ 0_{k+1, M_2 - \mu - (k+1)}],$$

we rewrite $T_{\kappa\mu}$ in terms of τ as

$$(46) \quad T_{\kappa\mu} = E_\kappa \tau E_\mu^T.$$

Then, the vector $\hat{T}_{\kappa\mu}$ is written in terms of $\hat{\tau}$ as

$$(47) \quad \hat{T}_{\kappa\mu} = \text{vec}(E_\kappa \tau E_\mu^T) = (E_\mu \otimes E_\kappa) \hat{\tau}$$

Thus, by (43), we have the following equation for $\hat{x}(u, v)$, and hence $x(s, t)$ expressed in terms of $\hat{\tau}$.

$$(48) \quad x(s, t) = \hat{x}(u, v) = [(E_\mu^T N_k(v)) \otimes (E_\kappa^T N_k(u))]^T \hat{\tau}$$

Next, letting $x^{(l_1, l_2)}(s, t)$ be the derivatives of $x(s, t)$ as

$$(49) \quad x^{(l_1, l_2)}(s, t) = \frac{\partial^{l_1 + l_2}}{\partial s^{l_1} \partial t^{l_2}} x(s, t),$$

and $\hat{x}^{(l_1, l_2)}(u, v)$ be defined similarly, they are related by

$$(50) \quad x^{(l_1, l_2)}(s, t) = \alpha^{l_1} \beta^{l_2} \hat{x}^{(l_1, l_2)}(u, v), \quad l_1, l_2 = 0, 1, \dots$$

Then, by (43), we have the following general expression of the derivatives $x^{(l_1, l_2)}(s, t)$ for $(s, t) \in \mathcal{D}_{\kappa, \mu}$

$$(51) \quad x^{(l_1, l_2)}(s, t) = \alpha^{l_1} \beta^{l_2} \left(N_k^{(l_2)}(v) \otimes N_k^{(l_1)}(u) \right)^T \hat{T}_{\kappa\mu},$$

with u, v given as (40). Using (47), equation (48) is generalized as

$$(52) \quad x^{(l_1, l_2)}(s, t) = \alpha^{l_1} \beta^{l_2} \left[\left(E_\mu^T N_k^{(l_2)}(v) \right) \otimes \left(E_\kappa^T N_k^{(l_1)}(u) \right) \right]^T \hat{\tau}.$$

Note that the derivatives of $N_k(t)$ in the above expression can be evaluated from (8) as

$$(53) \quad N_k^{(l)}(t) = S_k C_{k, l} h_{k-l}(t),$$

where $C_{k, l} \in \mathbf{R}^{(k+1) \times (k-l)}$ is defined by (11).

We are now in the position to derive various types of constraints on the surface $x(s, t)$.

3.2. Pointwise constraints

We consider to impose constraints on the value of $x^{(l_1, l_2)}(s, t)$ for given point $(s, t) \in \mathcal{D}$ and l_1, l_2 ($0 \leq l_1, l_2 \leq k$). Examples of the constraint are $x^{(l_1, l_2)}(s, t) = c$, $x^{(l_1, l_2)}(s, t) \geq c$, etc.

First let $(s, t) \in \mathcal{D}_{\kappa, \mu}$ for some κ ($0 \leq \kappa < m_1$) and μ ($0 \leq \mu < m_2$), and let u, v be defined as in (40). Then we see from (52) that any linear constraints on the value of $x^{(l_1, l_2)}(s, t)$ is specified as a linear constraint of the vector $\hat{\tau}$. Specifically, $x^{(l_1, l_2)}(s, t)$ is written as

$$(54) \quad x^{(l_1, l_2)}(s, t) = a^T \hat{\tau},$$

where $a \in \mathbf{R}^{M_1 M_2}$ is defined by

$$(55) \quad a = \alpha^{l_1} \beta^{l_2} (a_2 \otimes a_1)$$

and $a_1 \in \mathbf{R}^{M_1}$ and $a_2 \in \mathbf{R}^{M_2}$ are obtained from (52) as

$$(56) \quad a_1 = E_{\kappa}^T N_k^{(l_1)}(u) = \begin{bmatrix} 0_{\kappa}^T & N_k^{(l_1)}(u)^T & 0_{M_1 - \kappa - (k+1)}^T \end{bmatrix}^T,$$

$$(57) \quad a_2 = E_{\mu}^T N_k^{(l_2)}(v) = \begin{bmatrix} 0_{\mu}^T & N_k^{(l_2)}(v)^T & 0_{M_2 - \mu - (k+1)}^T \end{bmatrix}^T.$$

The only points in $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}]$ that are not covered by $\mathcal{D}_{\kappa, \mu}$ for any choice of κ, μ are those on the boundary (s, t) with either $s = s_{m_1}$ or $t = t_{m_2}$. It is obvious, however, that the function $x^{(l_1, l_2)}(s, t)$ is continuous on \mathcal{D} for $l_1, l_2 = 0, 1, \dots, k-1$, and the values at the boundary are obtained readily by setting, for instance, $\kappa = m_1 - 1$ and $u = 1$ for the boundary $s = s_{m_1}$. In this case, we only need to replace the vector a_1 in (56) by

$$(58) \quad a_1 = \begin{bmatrix} 0_{M_1 - (k+1)}^T & N_k^{(l_1)}(1)^T \end{bmatrix}^T.$$

If we need to constrain the k -th partial derivative, e.g. $x^{(k, l_2)}(s, t)$ at the boundary $s = s_{m_1}$, which is piecewise constant in s and discontinuous at the knot points $s = s_i$, we regard that $x^{(k, l_2)}(s_{m_1}, t) = \lim_{s \rightarrow s_{m_1}} x^{(k, l_2)}(s, t)$. Then the above result based on (58) holds also for $l_1 = k$.

Thus for any point $(s, t) \in \mathcal{D}$, the point constraint $x^{(l_1, l_2)}(s, t) = c$, $x^{(l_1, l_2)}(s, t) \geq c$, or $c \leq x^{(l_1, l_2)}(s, t) \leq c'$ for example is specified in terms of

$\hat{\tau}$ as $a^T \hat{\tau} = c$, $a^T \hat{\tau} \geq c$, or $c \leq a^T \hat{\tau} \leq c'$ respectively where the vector a is given by (55). Typical cases of imposing pointwise constraints are at the knot points (s_κ, t_μ) , yielding $u = 0$ and $v = 0$. We then need the vector of the form $N_k^{(l)}(0)$ in (56) and (57), which is computed from (53). For instance, $N_k^{(l)}(0)$ for $k = 3$, i.e. $N_3^{(l)}(0)$, is obtained as

$$(59) \quad N_3^{(l)}(0)^T = \begin{cases} \frac{1}{6} [1 & 4 & 1 & 0] & l = 0 \\ \frac{1}{2} [-1 & 0 & 1 & 0] & l = 1 \\ [1 & -2 & 1 & 0] & l = 2 \\ [-1 & 3 & -3 & 1] & l = 3. \end{cases}$$

In addition, if we impose constraints at (s_0, t_0) as $x(s_0, t_0) = x_0$, $x^{(1,1)}(s_0, t_0) = x_0^{(11)}$ and $x^{(2,2)}(s_0, t_0) = x_0^{(22)}$, we simply write the condition as $A\hat{\tau} = p$. For the case of $k = 3$, the matrix $A \in \mathbf{R}^{3 \times M_1 M_2}$ is derived from (56) and (57) with $\kappa, \mu = 0$ and (59) as

$$(60) \quad A = \begin{bmatrix} [N_3(0)^T \ 0_{M_2-4}^T] \otimes [N_3(0)^T \ 0_{M_1-4}^T] \\ \alpha\beta [N_3^{(1)}(0)^T \ 0_{M_2-4}^T] \otimes [N_3^{(1)}(0)^T \ 0_{M_1-4}^T] \\ \alpha^2\beta^2 [N_3^{(2)}(0)^T \ 0_{M_2-4}^T] \otimes [N_3^{(2)}(0)^T \ 0_{M_1-4}^T] \end{bmatrix}$$

$$= \begin{bmatrix} [\frac{1}{6} \ \frac{4}{6} \ \frac{1}{6} \ 0 \ \cdots \ 0] \otimes [\frac{1}{6} \ \frac{4}{6} \ \frac{1}{6} \ 0 \ \cdots \ 0] \\ \alpha\beta [-\frac{1}{2} \ 0 \ -\frac{1}{2} \ 0 \ \cdots \ 0] \otimes [-\frac{1}{2} \ 0 \ -\frac{1}{2} \ 0 \ \cdots \ 0] \\ \alpha^2\beta^2 [1 \ -2 \ 1 \ 0 \ \cdots \ 0] \otimes [1 \ -2 \ 1 \ 0 \ \cdots \ 0] \end{bmatrix}$$

and $p \in \mathbf{R}^3$ is defined by

$$(61) \quad p = \begin{bmatrix} x_0 & x_0^{(11)} & x_0^{(22)} \end{bmatrix}^T.$$

3.3. Constraints over knot point interval

We consider constraints on $x(s, t)$ over a knot point interval of s or t . Such a constraint can be used, e.g. to specify boundary conditions. Let a function $f(s)$ be given for $s \in [s_\kappa, s_{\kappa+1}]$, and moreover let it be described using B-

splines as

$$(62) \quad f(s) = \sum_{i=-k}^{m_1-1} c_i B_k(\alpha(s - s_i)), \quad s \in [s_\kappa, s_{\kappa+1}].$$

Then our problem is to find the condition on $\hat{\tau}$ such that the spline $x(s, t)$ satisfies, e.g. in the case of equality constraint,

$$(63) \quad x(s, t_\mu) = f(s) \quad \forall s \in [s_\kappa, s_{\kappa+1}]$$

for given integer μ ($0 \leq \mu \leq m_2$). Thus t is taken as a knot point t_μ . For example, the cases of $\mu = 0$ and $\mu = m_2$ may be used to specify the boundary conditions of $x(s, t)$ at the two boundaries of $t = t_0$ and $t = t_{m_2}$ of the domain \mathcal{D} .

Now, by (1), (40) and (41), the function $x(s, t_\mu)$ is given as

$$(64) \quad \begin{aligned} x(s, t_\mu) &= \hat{x}(u, 0) \\ &= \sum_{i=0}^k \sum_{j=0}^k \tau_{\kappa-k+i, \mu-k+j} N_{i,k}(u) N_{j,k}(0). \end{aligned}$$

Similarly, the function $f(s)$, $s \in [s_\kappa, s_{\kappa+1}]$ in (62) is expressed by

$$(65) \quad f(s) = \hat{f}(u) = \sum_{i=0}^k c_{\kappa-k+i} N_{i,k}(u).$$

Here if we let

$$(66) \quad \tau_{i, \mu-k} = \tau_{i, \mu-k+1} = \cdots = \tau_{i, \mu} = c_i \quad i = \kappa - k, \kappa - k + 1, \dots, \kappa,$$

then, we have

$$(67) \quad \begin{aligned} x(s, t_\mu) &= \hat{x}(u, 0) = \sum_{i=0}^k c_{\kappa-k+i} N_{i,k}(u) \sum_{j=0}^k N_{j,k}(0) \\ &= \hat{f}(u) = f(s) \end{aligned}$$

by (5) and (65).

Thus the constraint (63) is realized by the condition (66), which is expressed in terms of $\hat{\tau}$ as follows. Using the matrix $T_{\kappa, \mu}$ in (42), the condition

(66) is written as

$$(68) \quad T_{\kappa,\mu} = C$$

where $C = [\overbrace{\mathbf{c} \ \mathbf{c} \ \cdots \ \mathbf{c}}^{k+1}]$ and $\mathbf{c} = [c_{\kappa-k} \ c_{\kappa-k+1} \ \cdots \ c_{\kappa}]^T$. Applying ‘vec’ function and using (44) and (47), this condition is expressed in terms of $\hat{\tau}$ as

$$(69) \quad E_{\kappa\mu}\hat{\tau} = \hat{\mathbf{c}},$$

where $E_{\kappa\mu} \in \mathbf{R}^{(k+1)^2 \times M_1 M_2}$ is defined by

$$(70) \quad E_{\kappa\mu} = E_{\mu} \otimes E_{\kappa}$$

and $\hat{\mathbf{c}} \in \mathbf{R}^{(k+1)^2}$ by $\hat{\mathbf{c}} = \text{vec } C$, i.e. $\hat{\mathbf{c}} = \mathbf{1}_{k+1} \otimes \mathbf{c}$ with $\mathbf{1}_{k+1} = [1 \ 1 \ \cdots \ 1]^T \in \mathbf{R}^{k+1}$. Thus the constraint (63) is realized by the condition (69). Note that, when the other constraint as $x(s, t_{\mu}) \geq f(s)$ or $f(s) \leq x(s, t_{\mu}) \leq f'(s)$ is required instead of the equality $x(s, t_{\mu}) = f(s)$ in (63), we only need to replace (69) by $E_{\kappa\mu}\hat{\tau} \geq \hat{\mathbf{c}}$ or $\hat{\mathbf{c}} \leq E_{\kappa\mu}\hat{\tau} \leq \hat{\mathbf{c}}'$ respectively, where the vector $\hat{\mathbf{c}}'$ is defined similarly as $\hat{\mathbf{c}}$.

The above arguments for one knot point interval $[s_{\kappa}, s_{\kappa+1}]$ are easily generalized to larger knot point interval, say $[s_{\kappa}, s_{\zeta}]$ for arbitrary $\zeta (> \kappa)$, for a constraint as

$$(71) \quad x(s, t_{\mu}) = f(s) \quad \forall s \in [s_{\kappa}, s_{\zeta}].$$

Similarly as (66), this constraint is realized by the condition $\tau_{i,\mu-k} = \tau_{i,\mu-k+1} = \cdots = \tau_{i,\mu} = c_i \quad i = \kappa - k, \kappa - k + 1, \cdots, \zeta - 1$. Equation (68) now becomes $T'_{\kappa,\mu} = C'$, where $T'_{\kappa,\mu} = [\tau_{ij}]_{i=\kappa-k, j=\mu-k}^{\zeta-1, \mu}$ (the matrix consisting of

τ_{ij} for $\kappa - k \leq i \leq \zeta - 1$ and $\mu - k \leq j \leq \mu$) and $C' = [\overbrace{\mathbf{c}' \ \mathbf{c}' \ \cdots \ \mathbf{c}'}^{k+1}]$ with $\mathbf{c}' = [c_{\kappa-k} \ c_{\kappa-k+1} \ \cdots \ c_{\zeta-1}]^T$. Here note that the matrix $T'_{\kappa,\mu}$ is written as a submatrix of τ in (15) as $T'_{\kappa,\mu} = E'_{\kappa} \tau E_{\mu}$, where $E'_{\kappa} \in \mathbf{R}^{(k+\zeta-\kappa) \times M_1}$ is defined by

$$(72) \quad E'_{\kappa} = [0_{\zeta-\kappa+k, \kappa} \ I_{\zeta-\kappa+k} \ 0_{\zeta-\kappa+k, M_1-\zeta-k}].$$

Then, by $\text{vec } T'_{\kappa,\mu} = \text{vec } C'$, the condition (69) is generalized to

$$(73) \quad E'_{\kappa\mu}\hat{\tau} = \hat{\mathbf{c}}',$$

where,

$$(74) \quad E'_{\kappa\mu} = E_\mu \otimes E'_\kappa$$

and $\hat{\mathbf{c}}' = \text{vec } C' = \mathbf{1}_{k+1} \otimes \mathbf{c}'$.

The constraints along t axis such as $x(s_\kappa, t) = \hat{f}(t)$, $t \in [t_\mu, t_\eta]$ ($\eta > \mu$), can be treated similarly. When $\hat{f}(t)$ is given by $\hat{f}(t) = \sum_{i=-k}^{m_2-1} c_i B_k(\beta(t-t_i))$, the constraints are realized by the condition on control points as $E''_{\kappa\mu} \hat{\tau} = \hat{\mathbf{c}}''$, where $E''_{\kappa\mu} = E'_\mu \otimes E_\kappa$ with $E'_\mu \in \mathbf{R}^{(k+\eta-\mu) \times M_2}$ defined by

$$(75) \quad E'_\mu = [0_{\eta-\mu+k, \mu} \quad I_{\eta-\mu+k} \quad 0_{\eta-\mu+k, M_2-\eta-k}].$$

Remark 3. *If we impose the constraint $x(s, t_\mu) = f(s)$ on the entire interval $[s_0, s_{m_1}]$, then setting $\kappa = 0, \zeta = m_1$ yields $E'_\kappa = I_{M_1}$ in (72). Hence, the condition is expressed as $E'_{\kappa\mu} \hat{\tau} = \hat{\mathbf{c}}'$, where $E'_{\kappa\mu} = E_\mu \otimes I_{M_1}$ and $\hat{\mathbf{c}}' = \mathbf{1}_{k+1} \otimes \mathbf{c}'$ with $\mathbf{c}' = [c_{-k} \ c_{-k+1} \ \cdots \ c_{m_1-1}]^T$. A boundary condition $x(s, t_0) = c(\text{constant}) \ \forall s \in [s_0, s_{m_1}]$, for example, is realized by $E'_{\kappa\mu} \hat{\tau} = \hat{\mathbf{c}}'$ with $E'_{\kappa\mu} = [I_{(k+1)M_1} \ 0_{(k+1)M_1, (M_2-k-1)M_1}]$ since $\mu = 1$ and $\hat{\mathbf{c}}' = c \mathbf{1}_{(k+1)M_1}$ since $c_i = c \ \forall i$.*

3.4. Constraints over knot point region

We next consider constraints over knot point region. Specifically, first we consider an equality constraint on a basic knot point region as follows,

$$(76) \quad x(s, t) = f(s, t) \quad \forall (s, t) \in \mathcal{D}_{\kappa, \mu} = [s_\kappa, s_{\kappa+1}] \times [t_\mu, t_{\mu+1}],$$

where we assume that $f(s, t)$ is expressed in the same form as $x(s, t)$ as

$$(77) \quad f(s, t) = \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} c_{i,j} B_k(\alpha(s-s_i)) B_k(\beta(t-t_j)).$$

The cases of $x(s, t) \geq f(s, t)$, $x(s, t) \leq f(s, t)$, etc. follow similarly. Such a constraint may be realized by imposing the condition $\tau_{i,j} = c_{i,j}$ for $i = \kappa - k, \kappa - k + 1, \dots, \kappa$ and $j = \mu - k, \mu - k + 1, \dots, \mu$, which may be written as

$$(78) \quad T_{\kappa\mu} = C_{\kappa\mu},$$

where $T_{\kappa\mu}$ is given by (42), and $C_{\kappa\mu}$ is defined similarly as $C_{\kappa\mu} = [c_{i,j}]_{i=\kappa-k, j=\mu-k}^{\kappa, \mu}$ (the matrix consisting of $c_{i,j}$ with $\kappa - k \leq i \leq \kappa, \mu - k \leq j \leq \mu$).

The condition (78) guarantees that the constraint (76) be satisfied, since for (s, t) in $\mathcal{D}_{\kappa, \mu}$ it holds that

$$\begin{aligned}
 x(s, t) &= \hat{x}(u, v) = \sum_{i=0}^k \sum_{j=0}^k \tau_{\kappa-k+i, \mu-k+j} N_{i,k}(u) N_{j,k}(v) \\
 &= \sum_{i=0}^k \sum_{j=0}^k c_{\kappa-k+i, \mu-k+j} N_{i,k}(u) N_{j,k}(v) \\
 &= \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} c_{i,j} B_k(\alpha(s - s_i)) B_k(\beta(t - t_j)) \\
 (79) \quad &= f(s, t).
 \end{aligned}$$

We readily see that the condition (78) is expressed in terms of $\hat{\tau}$ analogously as (69),

$$(80) \quad E_{\kappa\mu} \hat{\tau} = \hat{c},$$

with the only difference in the definition of $\hat{c} \in \mathbf{R}^{(k+1)^2}$ as $\hat{c} = \text{vec } C_{\kappa\mu}$.

The special case of $f(s, t)$ in the constraint (76) is that it is constant, namely

$$(81) \quad x(s, t) = c \quad \forall (s, t) \in \mathcal{D}_{\kappa, \mu}.$$

We can realize $f(s, t) = c \quad \forall (s, t) \in \mathcal{D}_{\kappa, \mu}$ in (77) by setting $\tau_{i,j} = c_{i,j}$ for $i = \kappa - k, \kappa - k + 1, \dots, \kappa$ and $j = \mu - k, \mu - k + 1, \dots, \mu$. This is because, for $(s, t) \in \mathcal{D}_{\kappa, \mu}$, it holds that

$$\begin{aligned}
 f(s, t) &= \hat{f}(u, v) = \sum_{i=0}^k \sum_{j=0}^k c_{\kappa-k+i, \mu-k+j} N_{i,k}(u) N_{j,k}(v) \\
 &= \sum_{i=0}^k \sum_{j=0}^k c N_{i,k}(u) N_{j,k}(v) = c \left(\sum_{i=0}^k N_{i,k}(u) \right) \left(\sum_{j=0}^k N_{j,k}(v) \right) \\
 (82) \quad &= c.
 \end{aligned}$$

In this case, the vector \hat{c} in the condition (80) becomes $\hat{c} = \mathbf{c}_{(k+1)^2}$, where we introduced the notation $\mathbf{c}_i = [c \ c \ \dots \ c]^T \in \mathbf{R}^i$.

The above results on $\mathcal{D}_{\kappa, \mu} = [s_\kappa, s_{\kappa+1}] \times [t_\mu, t_{\mu+1}]$ may readily be extended to broader region $[s_\kappa, s_\zeta] \times [t_\mu, t_\eta]$ for arbitrary $\zeta (> \kappa)$ and $\eta (> \mu)$.

That is, the constraint

$$(83) \quad x(s, t) = f(s, t) \quad \forall (s, t) \in [s_\kappa, s_\zeta] \times [t_\mu, t_\eta]$$

is satisfied by the condition $\tau_{i,j} = c_{i,j}$ for $i = \kappa - k, \kappa - k + 1, \dots, \zeta - 1$ and $j = \mu - k, \mu - k + 1, \dots, \eta - 1$. The constraint corresponding to (80) is then derived as

$$(84) \quad E'_{\kappa\mu} \hat{\tau} = \hat{c}',$$

where $E'_{\kappa\mu} \in \mathbf{R}^{(\zeta - \kappa + k)(\eta - \mu + k) \times M_1 M_2}$ is defined by $E'_{\kappa\mu} = E'_\mu \otimes E'_\kappa$ with (72) and (75), and $\hat{c}' = \text{vec } C'_{\kappa\mu}$ with $C'_{\kappa\mu} = [c_{i,j}]_{i=\kappa-k, j=\mu-k}^{\zeta-1, \eta-1}$.

Remark 4. As noted earlier, if an inequality constraint like $x(s, t) \geq f(s, t)$ is required instead of equality constraint $x(s, t) = f(s, t)$ in above, then we simply replace the condition on $\hat{\tau}$ accordingly, e.g. replace (84) by $E'_{\kappa\mu} \hat{\tau} \geq \hat{c}'$ in the case of (83). Moreover if we impose the constraint on the entire region $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}]$, then setting $\kappa = \mu = 0, \zeta = m_1, \eta = m_2$ yields $E'_\kappa = I_{M_1}$ and $E'_\mu = I_{M_2}$ in (72) and (75), and hence $E'_{\kappa\mu} = I_{M_1 M_2}$ in (84). Thus $x(s, t) \geq f(s, t) \quad \forall (s, t) \in \mathcal{D}$ for example is realized by the condition $\hat{\tau} \geq \hat{c}'$, and in particular $\hat{c}' = \mathbf{c}_{M_1 M_2}$ if $f(s, t) = c$ (constant).

3.5. Constraints on integral value

We consider equality or inequality constraints on the integral value

$$(85) \quad S = \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} x(s, t) ds dt,$$

such as $S \geq c, S = c$, etc. The value S is expressed as a linear function of $\hat{\tau}$ as we show in below.

Recalling the expression (20), we can evaluate the integral value as

$$(86) \quad \begin{aligned} S &= \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} (b_2(t) \otimes b_1(s))^T \hat{\tau} ds dt \\ &= \left(\int_{t_0}^{t_{m_2}} b_2(t) dt \otimes \int_{s_0}^{s_{m_1}} b_1(s) ds \right)^T \hat{\tau} \end{aligned}$$

or simply

$$(87) \quad S = a^T \hat{\tau}.$$

Here the vector $a \in \mathbf{R}^{M_1 M_2}$ is defined by

$$(88) \quad a = a_2 \otimes a_1$$

with $a_1 \in \mathbf{R}^{M_1}$ and $a_2 \in \mathbf{R}^{M_2}$ given by

$$(89) \quad a_1 = \int_{s_0}^{s_{m_1}} b_1(s) ds, \quad a_2 = \int_{t_0}^{t_{m_2}} b_2(t) dt.$$

Thus the constraint on the integral value as $S = c$, $S \geq c$, or $c \leq S \leq c'$ is realized respectively by $a^T \hat{\tau} = c$, $a^T \hat{\tau} \geq c$ or $c \leq a^T \hat{\tau} \leq c'$.

The computations of a_1 and a_2 are simplified by changing the integration variables in (89) as $s' = s - s_0$ and $t' = t - t_0$, and they are given by

$$(90) \quad a_1 = \frac{1}{\alpha} \int_0^{m_1} \hat{b}_1(s) ds, \quad a_2 = \frac{1}{\beta} \int_0^{m_2} \hat{b}_2(t) dt,$$

where $\hat{b}_1(s)$ and $\hat{b}_2(t)$ are the non-scaled version of $b_1(s)$ and $b_2(t)$ in (17) and (18) respectively,

$$(91) \quad \hat{b}_1(s) = [B_k(s - (-k)) B_k(s - (-k + 1)) \cdots B_k(s - (m_1 - 1))]^T,$$

$$(92) \quad \hat{b}_2(t) = [B_k(t - (-k)) B_k(t - (-k + 1)) \cdots B_k(t - (m_2 - 1))]^T.$$

By the definition of $B_k(s)$ in (3), the integral e.g. for a_1 in (90) is expressed as

$$(93) \quad \int_0^{m_1} \hat{b}_1(s) ds = \left[\sum_{i=0}^0 I_{i,k} \sum_{i=0}^1 I_{i,k} \cdots \sum_{i=0}^k I_{i,k} \cdots \sum_{i=0}^k I_{i,k} \sum_{i=1}^k I_{i,k} \cdots \sum_{i=k}^k I_{i,k} \right]^T,$$

where $I_{i,k} = \int_0^1 N_{i,k}(s) ds$.

When $k = 3$, for example, a_1 and a_2 in (90) are obtained as

$$(94) \quad a_1 = \frac{1}{24\alpha} [1 \ 12 \ 23 \ 24 \ \cdots \ 24 \ 23 \ 12 \ 1]^T,$$

$$(95) \quad a_2 = \frac{1}{24\beta} [1 \ 12 \ 23 \ 24 \ \cdots \ 24 \ 23 \ 12 \ 1]^T,$$

using the integral values in $\int_0^1 N_3(s) ds$ computed from (9) as follows.

$$(96) \quad \int_0^1 N_3(s) ds = [I_{0,3} \ I_{1,3} \ I_{2,3} \ I_{3,3}]^T = S_3 \int_0^1 h_3(s) ds = \frac{1}{24} [1 \ 11 \ 11 \ 1]^T.$$

3.6. Constrained spline surfaces

We examined various types of constraints on $x(s, t)$ in the form that can be incorporated to optimal spline problems in Section 2. The optimal smoothing or interpolating spline surfaces without constraints are obtained by minimizing the convex quadratic cost $J(\hat{\tau})$ in (21), (30) and (36), whereas various types of constraints on the splines are developed in Sections 3.2–3.5 as linear constraints on the vector $\hat{\tau}$, either equality or inequality or both. Then, a general form of problems can be written as convex quadratic programming problems as follows:

(QP1)

$$(97) \quad \min_{\hat{\tau} \in \mathbf{R}^{M_1 M_2}} J(\tau) = \frac{1}{2} \hat{\tau}^T G \hat{\tau} + g^T \hat{\tau}$$

subject to the constraints of the form

$$(98) \quad A\hat{\tau} = p, \quad f_1 \leq E\hat{\tau} \leq f_2, \quad h_1 \leq \hat{\tau} \leq h_2,$$

for some matrices and vectors of appropriate dimensions.

Here are some remarks for setting up the constraints in (98). If we impose an equality constraint $x(s_0, t_0) = 1$ at point (s_0, t_0) for example, we simply introduce the condition $a^T \hat{\tau} = 1$ for the vector a computed by (55) with $l_1 = l_2 = 0$, and similarly $a^T \hat{\tau} \geq 1$ for inequality constraint $x(s_0, t_0) \geq 1$. Repeating such a process for all the required constraints based on the formula in Sections 3.2–3.5 yields combination of conditions of the forms $A_i \hat{\tau} = p_i$, $E_i \hat{\tau} \geq f_i$, $E_j \hat{\tau} \leq f'_j$, $\hat{\tau} \geq h_i$, and $\hat{\tau} \leq h_j$, which may be integrated as the three types of conditions given in (98). Note here that we set the constraints (98) so that the inconsistency in (98) is not caused among such constraints.

A very efficient numerical algorithm, such as active set method and interior point method, etc. is available for solving the problem (QP1) (see e.g. [20] and [25]). We use a function ‘quadprog’ of Matlab optimization toolbox for this purpose. When the constraints in (98) are inconsistent, this function gives some warning that solution is infeasible.

For convenience, we summarize an algorithm of constructing constrained smoothing spline surfaces as follows.

Algorithm 3. (*Smoothing spline surface with constraints*) Suppose that we are given a set of data in (14), and constraints as described in Sections 3.2–3.5. Then, $x(s, t)$ with required constraints is constructed in the following steps (S1’)–(S5’).

- (S1') Compute the matrix G in (22) and the vector g in (23) following the steps (S1)-(S4) in Algorithm 2.
- (S2') Set up all the equality constraints $A_i \hat{\tau} = p_i$ in one form as $A \hat{\tau} = p$, and similarly inequality constraints as $f_1 \leq E \hat{\tau} \leq f_2$.
- (S3') For inequality constraints on the vector $\hat{\tau}$ itself (e.g. Remark 4), if any, set them up as $h_1 \leq \hat{\tau} \leq h_2$.
- (S4') Solve convex QP problem (QP1) in terms of $\hat{\tau}$.
- (S5') Compute $x(s, t)$ in (1).

On the other hand, if only equality constraints are imposed on the spline as summarized as $A \hat{\tau} = p$, we can proceed further by using Lagrange function,

$$(99) \quad L(\hat{\tau}, \mu) = J(\hat{\tau}) + \mu^T (A \hat{\tau} - p).$$

By the standard procedure, we get a system of equations in $\hat{\tau}$ and the Lagrange multipliers μ ,

$$(100) \quad \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \hat{\tau} \\ \mu \end{bmatrix} = \begin{bmatrix} -g \\ p \end{bmatrix}.$$

Such a case arises when we specify the boundary conditions $x(s, t_\mu) = f(s)$ as described in Section 3.3 as the equality constraint (69). Periodic spline surfaces can also be treated in this settings [10]. For example, if the first k columns and the last k columns are the same in the control point matrix τ , then we get the relation $x^{(0, l_2)}(s, t_0) = x^{(0, l_2)}(s, t_{m_2}) \forall s \in [s_0, s_{m_1}]$ for $l_2 = 0, 1, \dots, k - 1$, hence the periodicity in t when $[t_0, t_{m_2}]$ is extended to entire real line. Such a condition on τ is described as $\tau \bar{E}_1^T = \tau \bar{E}_2^T$ with

$$(101) \quad \bar{E}_1 = [I_k \ 0_{k, M_2 - k}], \quad \bar{E}_2 = [0_{k, M_2 - k} \ I_k],$$

or $\tau(\bar{E}_1 - \bar{E}_2)^T = 0$, which is rewritten in terms of $\hat{\tau}$ as the equality $E \hat{\tau} = 0_k$ with $E = (\bar{E}_1 - \bar{E}_2) \otimes I_{M_1}$.

4. Numerical examples

We examine the design method presented in the previous sections numerically. In all the cases, optimal smoothing splines by (16) or (35) and cubic splines, i.e. $k = 3$, are used. The types of constraints used in the following three Sections 4.1–4.3 are constraints on points (Section 3.2) and over lines (Section 3.3) both as equalities in Section 4.1, those over region (Section 3.4)

as inequality and integral value (Section 3.5) as equality in Section 4.2, and finally two constraints over region (Section 3.4) both as inequalities in Section 4.3.

4.1. Interpolation of function with boundary conditions

We construct optimal smoothing spline surfaces $x(s, t)$ with boundary conditions as an approximation of the following function $f(s, t)$,

$$(102) \quad f(s, t) = 2 + 5 \exp \left\{ \frac{-|(s-5)(t-5)|}{10} \right\} + \sin(s) \cos(2t)$$

for $(s, t) \in \mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}] = [0, 10] \times [0, 10]$. With $N_1 = N_2 = 10$, the set of 100(= $N_1 N_2$) data d_{ij} in (14) are generated by sampling the function $f(s, t)$ as $d_{ij} = f(u_i, v_j)$, $i, j = 1, 2, \dots, 10$, where both u_i and v_j are equally spaced in $[1, 9]$. The design parameters are set as $\alpha = \beta = 1$ and $m_1 = m_2 = 13$ in (1). On designing the surface $x(s, t)$, we impose the pointwise constraints in Section 3.2

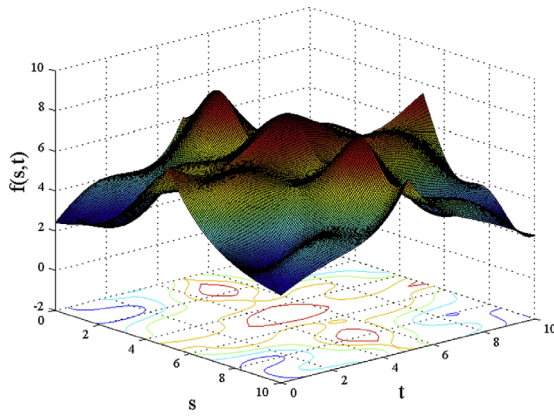
$$(103) \quad x(u_i, v_j) = d_{ij}, \quad i, j = 1, 2, \dots, 10.$$

Thus the surface $x(s, t)$ exactly passes through the given set of data points, and this is a smooth interpolation problem based on the cost given in (35). The constraint (103) is realized by the condition of the form $A_0 \hat{\tau} = p_0$, and moreover here we impose the following boundary conditions. Namely, at the two boundary points of variable t (i.e. $t = 0$ and $t = 10$), we impose on the surface $x(s, t)$ the equality constraints over the interval $[s_0, s_{m_1}] (= [0, 10])$ (cf. Section 3.3) as

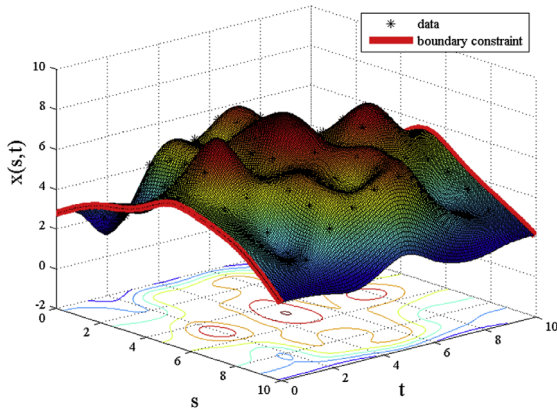
$$(104) \quad x(s, 0) = \bar{f}_1(s), \quad x(s, 10) = \bar{f}_2(s), \quad \forall s \in [0, 10].$$

Here, the boundary functions $\bar{f}_1(s)$ and $\bar{f}_2(s)$ are determined in the form of (62) by approximating the original function $f(s, t)$ at $t = 0$ and $t = 10$ respectively using the method in [9]. The constraints in (104) are realized as $A_1 \hat{\tau} = p_1$ and $A_2 \hat{\tau} = p_2$, which together with $A_0 \hat{\tau} = p_0$ for (103) are put in the form $A \hat{\tau} = p$.

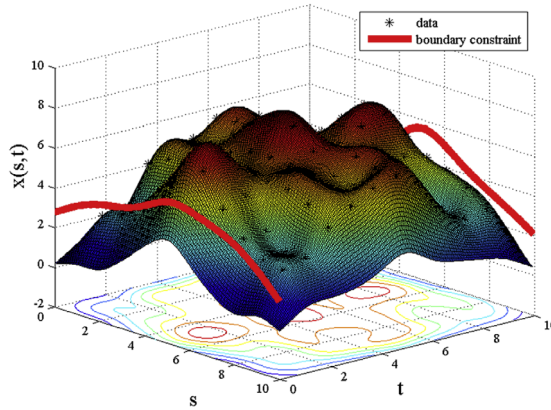
Figure 1 shows the function $f(s, t)$ in (102) (top) and the results for constrained (middle) and unconstrained (bottom) spline surfaces with data points (*). Here the constrained spline indicates that both (103) and (104) are imposed, whereas none of them is used in the unconstrained case. Note



(a) given function $f(s,t)$



(b) constrained case



(c) unconstrained case

Figure 1: Data points (*), $f(s,t)$ in (102) (top), constrained (middle) and unconstrained (bottom) smoothing splines.

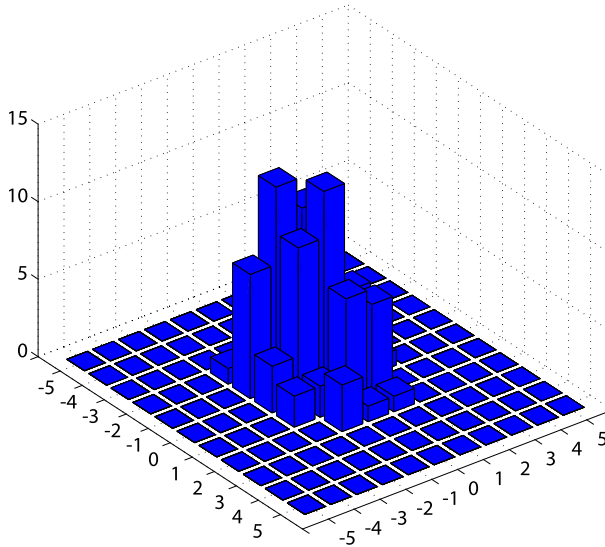


Figure 2: Histogram of 100 Gaussian random numbers with zero means and 1 variance.

that, for the unconstrained case, the optimal spline $x(s, t)$ is computed using the parameters $\lambda = 10^{-8}$ and $w_i = 1/N_1 N_2 \forall i$ in (16). We confirm that constrained spline not only recovers the original function $f(s, t)$ fairly precisely but also satisfies all the constraints in (103) and (104) unlike the unconstrained case.

4.2. Approximation of probability density functions

Next we approximate a probability density function $f(s, t)$ from the histogram of random samples. In Figure 2, we plot the histogram of 100 Gaussian random numbers with zero mean and variance 1. The density function is obtained in the region $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}] = [-5, +5] \times [-5, +5]$. Then we are given data $(u_i, v_j; d_{ij})$, $i = 1, 2, \dots, N_1$, $j = 1, 2, \dots, N_2$ in (14) as follows: First, the center of each bin in the histogram is set as data point, i.e. $(u_i, v_j) = (i - 6, j - 6)$, and hence $N_1 = N_2 = 11$. Let H_{ij} be the value of histogram at (u_i, v_j) , $i, j = 1, 2, \dots, 11$. Then the data d_{ij} is obtained by rescaling H_{ij} , so that the volume covered by the histogram over \mathcal{D} is normalized to one, i.e. $d_{ij} = H_{ij}/100$, $i, j = 1, 2, \dots, 11$.

Based on the cost function (16), we construct an optimal smoothing spline surface $x(s, t)$ with $\alpha = \beta = 1$ and $m_1 = m_2 = 10$. The parameters λ

and w_i are set as $\lambda = 10^{-6}$ and $w_i = 1/N_1N_2$. Also, since we are constructing spline surface $x(s, t)$ as approximation of 2D probability density function of random variable taking the values only in $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}]$, it is natural that we impose the equality and inequality constraints as

$$(105) \quad \int_{s_0}^{s_{m_1}} \int_{t_0}^{t_{m_2}} x(s, t) ds dt = 1, \quad x(s, t) \geq 0 \quad \forall (s, t) \in \mathcal{D},$$

yielding the expression of linear constraints in $\hat{\tau}$ by Section 3.4 and 3.5. Specifically the first equality constraint is written in the form of $a^T \hat{\tau} = 1$, and the second inequality is given simply as $\hat{\tau} \geq 0$ (see Remark 4 in Section 3.4).

Figure 3 shows the results for constrained (Figure 3 (a)) and unconstrained (Figure 3 (b)) smoothing spline surfaces with data points (*). We confirmed that the surface $x(s, t)$ in (a) approximates the original Gaussian surface quite well while maintaining the constraints (105), which is not the case in (b). In fact, we computed the integral values S in (85) using (87), and the results were $S = 1$ for (a) as we expect and $S = 1.2106$ for (b). Also, by increasing the numbers of samples, bins and basis functions, i.e. N_1, N_2, α, β , we confirmed that the above approximation was improved.

4.3. Smoothing of digital image

As a practical use of smoothing surfaces with inequality constraints, we consider a smoothing problem for digital image corrupted by noise.

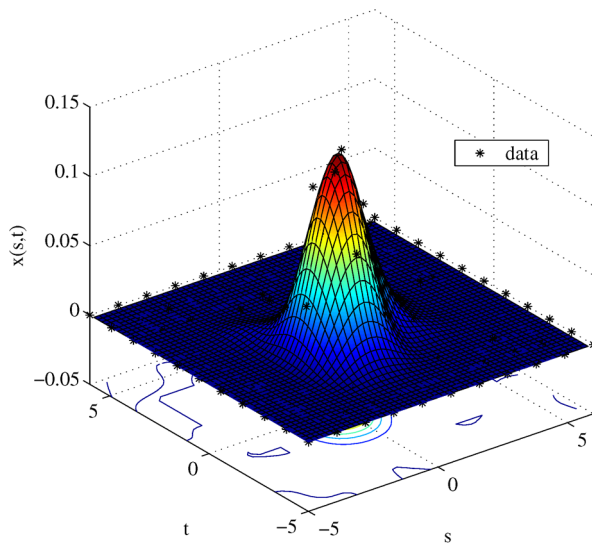
Assume that we are given an image $f(i, j)$, $i = 1, 2, \dots, N_1$, $j = 1, 2, \dots, N_2$, with $f(i, j)$ denoting the gray level of the ij -th pixel. Then, letting the noisy data $(u_i, v_j; d_{ij})$ be given as $(u_i, v_j) = (i, j)$, $d_{ij} = f(i, j)$ for $i = 1, 2, \dots, m_1$, $j = 1, 2, \dots, m_2$, we construct optimal spline surfaces $x(s, t)$ with $m_1 = N_1$ and $m_2 = N_2$.

The test image $f(i, j)$ shown in Figure 4 (a) is taken from [27], where it is of size 50×50 [pixel], i.e. $N_1 = N_2 = 50$, with the gray levels in $[0, 255]$. Figure 4 (b) is the noisy test image, where Gaussian white noise with mean zero and standard deviation 30 was added to $f(i, j)$ to generate the data d_{ij} .

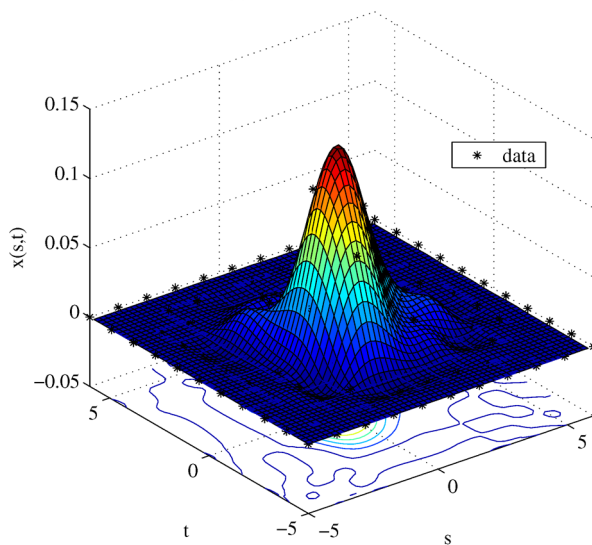
For constructing optimal spline surface $x(s, t)$ for smoothing this noisy image, we introduce the following inequality constraint over the entire domain.

$$(106) \quad 51 \leq x(s, t) \leq 153 \quad \forall (s, t) \in [0, 50] \times [0, 50],$$

Note that, in actual smoothing, the constraint should be $0 \leq x(s, t) \leq 255$, but we restricted the levels in $[51, 153]$ for the sake of comparison between the surfaces designed with and without such constraints. The constraints



(a) constrained case



(b) unconstrained case

Figure 3: Data points (*), constrained (top) and unconstrained (bottom) smoothing splines for histogram generated from Gaussian probability density function $f(s, t)$.

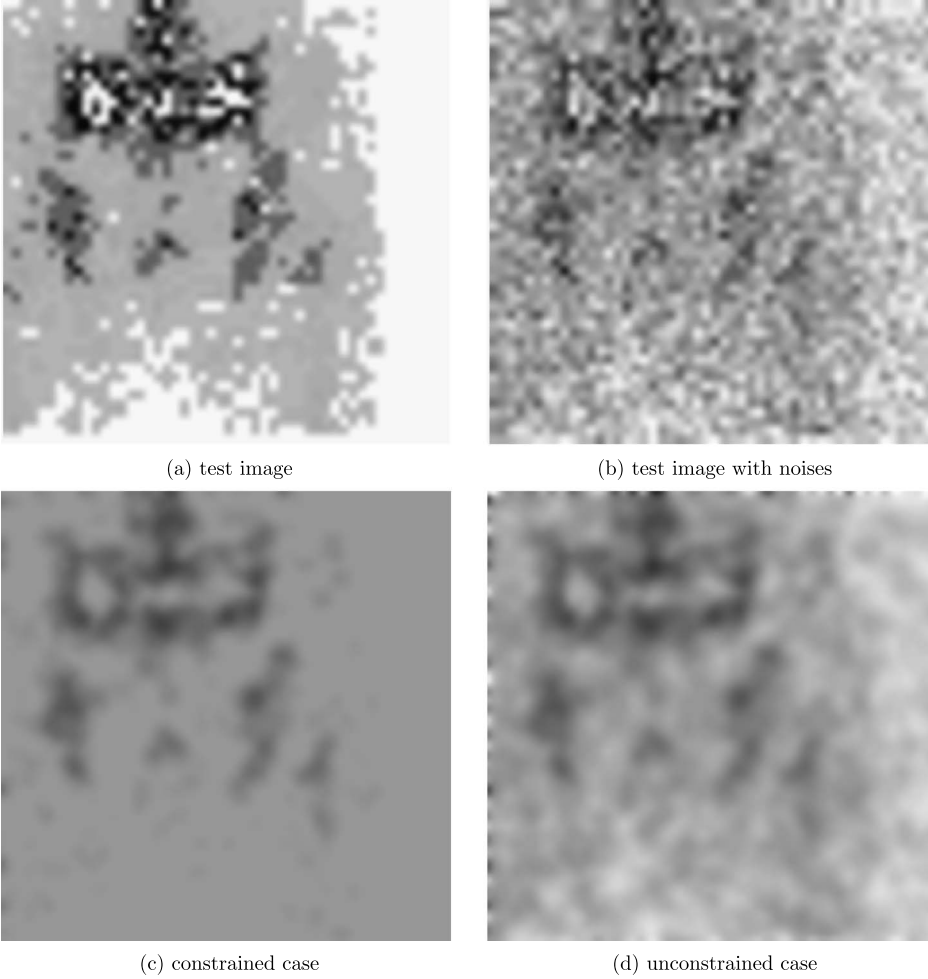


Figure 4: (a) test image (bony nuclear medicine image used for analyzing a patient), (b) test image with noises, (c) the smoothing image by constrained smoothing spline surface, and (d) the smoothing image by unconstrained smoothing spline surface.

in (106) are realized by the inequality $51 \times \mathbf{1} \leq \hat{\tau} \leq 153 \times \mathbf{1}$, where $\mathbf{1} = [1 \ 1 \ \dots \ 1]^T \in \mathbf{R}^{M_1 M_2}$. The design parameters are set as $\lambda = 10^{-4}$, $\alpha = \beta = 1$, $w_i = 1/N_1 N_2$, and an optimal smoothing spline surface $x(s, t)$ is constructed based on the cost function (16).

Figure 4 (c) and (d) show the results of the smoothing operation, where (c) is the case of constrained spline surface $x(s, t)$ with the inequality con-

straints (106), and (d) is the case without constraints. From these results, we observe that the present method works quite well in the presence of noises. In particular, noises remaining in Figure 4 (d) by only the smoothing operation are considerably suppressed by introducing the constraint as we see in (c).

5. Concluding remarks

We considered the problem of designing optimal smoothing and interpolating spline surface with equality and/or inequality constraints. The splines are constructed by using normalized uniform B-splines as the basis functions. Then we formulated various types of equality and inequality constraints including those at isolated point (s, t) , those over knot point interval as $[s_\kappa, s_\zeta]$ for fixed $t = t_\mu$, and those over region $[s_\kappa, s_\zeta] \times [t_\mu, t_\eta]$. These constraints are expressed as linear functions of the control points, and the problem is reduced to quadratic programming problem. We examined performance by numerical examples of smooth interpolation of given function with boundary constraints and of approximating two dimensional probability density function from the underlying histogram, and smoothing of noisy digital image data. In all the cases, the significance of introducing the constraints was confirmed.

Future work includes further extension of this result to higher dimensional case, in particular to the treatment of splines with arbitrary number of variables or n -variate splines for general n . In this case, important issues will be not only the derivation of the algorithms for optimal smoothing splines with constraints, but also a derivation of practical numerical algorithms. In particular, since the n -variate case requires in principle to find control points formed as an n -dimensional array, it will require more detailed studies to obtain practical and efficient algorithm for the numerical computation.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments that greatly improved this paper. This work was supported in part by Japan Ministry of Education, Culture, Sports, Science and Technology under Grant No. 23560541.

References

- [1] R. K. Beatson and Z. Ziegler, Monotonicity Preserving Surface Interpolation. *SIAM J. Numer. Anal.* **22**, 2 (1985). 401–411. [MR0781327](#)

- [2] C. de Boor, *A Practical Guide to Splines*. Revised Edition, Springer-Verlag, New York. (2001). [MR1900298](#)
- [3] J. Charles, S. Sun and C. Martin, Cumulative Distribution Estimation via Control Theoretic Smoothing Splines. *Three Decades of Progress in Control Sciences*. X. Hu, U. Jonsson, B. Wahlberg and B. Ghosh (Eds.), Springer, Berlin (2010). 95–104. [MR3052609](#)
- [4] P. Costantini and F. Pelosi, Shape Preserving Histogram Approximation. *Advances in Computational Mathematics* **26**, 1–3 (2007). 205–230. [MR2350352](#)
- [5] M. Egerstedt and C. F. Martin, Optimal control and monotone smoothing splines. *New Trends in Nonlinear Dynamics and Control, and Their Applications (Lecture Notes in Control and Inform. Sci.)* **295**, Springer, Berlin (2003). 279–294. [MR2024215](#)
- [6] M. Egerstedt and C. Martin, *Control Theoretic Splines: Optimal Control, Statistics and Path Planning*. Princeton University Press, Princeton, NJ. (2010). [MR2584170](#)
- [7] T. Elfving and L. E. Anderson, An Algorithm for Computing Constrained Smoothing Spline Functions. *Numerische Mathematik* **52**, (1988). 583–595. [MR0945101](#)
- [8] R. Farwig, Multivariate Interpolation of Scattered Data By Moving Least Squares Methods. *Algorithms for Approximations*, Clarendon Press, Oxford. (1987). 193–211. [MR0911314](#)
- [9] H. Fujioka, H. Kano, M. Egerstedt and C. Martin, Smoothing Spline Curves and Surfaces for Sampled Data. *Int. J. of Innovative Computing, Information and Control* **1**, 3 (2005). 429–449. [MR2731007](#)
- [10] H. Fujioka and H. Kano, Periodic Smoothing Spline Surface and Its Application to Dynamic Contour Modeling of Wet Material Objects. *IEEE Trans. Systems, Man and Cybernetics Part A* **39**, 1 (2009). 251–261.
- [11] L. Han and L. L. Schumaker, Fitting Monotone Surfaces to Scattered Data Using C^1 Piecewise Cubics. *SIAM J. Numerical Analysis* **34**, 2 (1997). 569–585. [MR1442929](#)
- [12] M. Z. Hussain, D. Zaidi, M. Hussain and A. A. Majid, Monotone Surface Data Visualization. *European J. of Scientific Research* **38**, 3 (2009). 351–361.

- [13] B. Jüttler, Surface fitting using convex tensor-product splines. *J. of Computational and Applied Mathematics* **84**, 1 (1997). 23–44. [MR1474401](#)
- [14] H. Kano, H. Nakata, and C. F. Martin, Optimal Curve Fitting and Smoothing Using Normalized Uniform B-Splines: A tool for studying complex systems. *Applied Mathematics and Computation* **169**, 1 (2005). 96–128. [MR2169137](#)
- [15] H. Kano, H. Fujioka, and C. F. Martin, Optimal Smoothing and Interpolating Splines with Constraints. *Applied Mathematics and Computation* **218**, 5 (2011). 1831–1844. [MR2831414](#)
- [16] H. Kano, H. Fujioka, and C. F. Martin, Optimal Smoothing Spline with Constraints on Its Derivatives. *Proc. of the 49th IEEE Conference on Decision and Control*, Atlanta, GA, USA, Dec. 15–17 (2010). 6785–6790.
- [17] A. Kouibia and M. Pasadas, Variational bivariate interpolating splines with positivity constraints. *Applied Numerical Mathematics* **44**, 4 (2003). 507–526. [MR1957691](#)
- [18] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*. Second Edition, Academic Press. (1985). [MR0792300](#)
- [19] S. Maad, C. F. Martin, T. Strömberg and J. Byström, Surface fitting with boundary data. *Proc. of IEEE Conf. on Decision and Control, Paradise Island*. Bahamas, Dec. 14–17 (2004). 3649–3653.
- [20] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall. (2002).
- [21] C. F. Martin, S. Sun and M. Egerstedt, Optimal control, statistics and path planning. *Math. Comput. Modelling* **33** (2001). 237–253. [MR1812548](#)
- [22] R. Meegaskumbura, *Control theoretic splines and derivative constraints*. PhD Dissertation, Texas Tech University, Lubbock, TX. (2010). [MR2584170](#)
- [23] M. C. Meyer, Inference Using Shape-Restricted Regression Splines. *Annals of Applied Statistics* **2**, 3 (2008). 1013–1033. [MR2516802](#)
- [24] P. Michalik and B. D. Brüderlin, Constraint-based Design of B-spline Surfaces from Curves. *Proc. of the 9th ACM Symposium on Solid Modeling and Applications*, Genova, Italy, Jun. 9–11 (2004). 213–223.

- [25] J. Nocedal and S. J. Wright, *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)*. Second Edition, Springer. (2006). [MR2244940](#)
- [26] J. O. Ramsay, Monotone Regression Splines in Action. *Statist. Sci.* **3**, 4 (1988). 385–493.
- [27] W. E. Snyder, *NC State University Image Analysis Laboratory Database*, available at <http://www.ece.ncsu.edu/imaging/Archives/ImageDatabase/index.html>. (2002).
- [28] M. Villalobos and G. Wahba, Inequality-Constrained Multivariate Smoothing Splines with Application to the Estimation of Posterior Probabilities. *J. American Statistical Association* **82**, 397 (1987). 239–248. [MR0883352](#)
- [29] G. Wahba, *Spline models for observational data*. CBMS-NSF Regional Conference Series in Applied Mathematics, **59**, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. (1990). [MR1045442](#)
- [30] Y. Wang, *Smoothing Splines: Methods and Applications*. CRC Press, New York. (2011). [MR2814838](#)
- [31] K. Yano and K. Harada, Reconstruction of B-spline Skinning Surface from Generalized Cylinder Mesh. *Visual Computer* **26**, 1 (2010). 31–40.
- [32] Z. Zhang, J. Tomlinson, and C. F. Martin, Splines and linear control theory. *Acta Appl. Math.* **49**, 1 (1997). 1–34. [MR1482878](#)

HIROYUKI FUJIOKA
DEPARTMENT OF SYSTEM MANAGEMENT
FUKUOKA INSTITUTE OF TECHNOLOGY
FUKUOKA 811-0295
JAPAN
E-mail address: fujioka@fit.ac.jp

HIROYUKI KANO
SCHOOL OF SCIENCE AND ENGINEERING
TOKYO DENKI UNIVERSITY
SAITAMA 350-0394
JAPAN
E-mail address: kano@mail.dendai.ac.jp

Clyde F. Martin
Department of Mathematics and Statistics
Texas Tech University
Lubbock, TX 79409
USA

E-mail address: clyde.f.martin@ttu.edu

RECEIVED JUNE 27, 2012