Solution and Linear Estimation of 2-D Nearest-Neighbor Models

BERNARD C. LEVY, SENIOR MEMBER, IEEE MILTON B. ADAMS, MEMBER, IEEE, AND ALAN S. WILLSKY, FELLOW, IEEE

The solution and linear estimation of 2-D nearest-neighbor models (NNMs) are considered. The class of problems that can be described by NNMs is quite large, as models of this type arise whenever partial differential equations are discretized with finite-difference methods. A general solution technique is proposed for 2-D NNMs that relies on converting the system into an equivalent 1-D two-point boundary-value descriptor system (TPBVDS) of large dimension, for which a recursive and stable solution technique is developed. Under slightly restrictive assumptions, an even faster procedure can be obtained by using the Fast-Fourier Transform (FFT), with respect to one of the space dimensions, to convert the 1-D TPBVDS into a set of decoupled TPBVDSs of low order, which can be solved in parallel. The smoothing problem for 2-D random fields described by stochastic NNMs is then examined. The smoother is expressed as a Hamiltonian system of twice the dimension of the original system, and is also in NNM form. NNM solution techniques are therefore directly applicable to this smoother. Our results are illustrated by two examples, corresponding to the discretized Poisson and heat equations, respectively.

I. INTRODUCTION

In two dimensions, a large class of physical processes can be described by nearest neighbor models (NNMs): When finite-difference methods are used to discretize linear 2-D partial differential equations of arbitrary type (hyperbolic, parabolic, or elliptic), and of any order, the resulting finite-difference approximation can usually be expressed in the form of a vector NNM. Consequently, it is not surprising that NNMs have been employed widely to model 2-D stochastic images [1]–[4], particularly for image restoration and coding, as well as for the control and estimation of distributed parameter systems.

This paper is concerned with the development of efficient estimation algorithms for 2-D random fields described by stochastic nearest-neighbor models over a rectangular domain, when local boundary conditions, which include as special cases periodic, Dirichlet, and Neumann conditions, are imposed on the domain boundaries. As NNMs have an acausal structure, we shall focus our attention on the NNM smoothing problem, because this problem is also acausal, in the sense that the measurements need not be produced according to a specific order in 2-D space. A system is said to be acausal if for an arbitrary partition of points in space between "past" and "future," future outputs are allowed to depend on both future and past inputs. Thus, both the class of 2-D estimation problems that we examine and the NNMs that are used to formulate these problems are *completely acausal*. This is in contrast with early attempts at deriving 2-D estimation algorithms, which mimicked the structure of 1-D Kalman filters by introducing artificial 2-D causality concepts, such as quarter-plane or asymmetric half-plane causality (see the discussion appearing in [5, ch. 4]). On the other hand, because our goal is to obtain efficient estimation procedures, the algorithms that we develop for the NNM smoothing problem are *recursive* and are obtained by breaking down noncausal processing steps into parts that are causal. As the original problem is noncausal, there is generally a large amount of flexibility in the choice of recursion directions for the algorithms that we propose and, consequently, causality appears as a computational artifice, not as a modeling assumption.

The approach used here to formulate the NNM smoothing problem relies on the general results developed in [6]–[8] for the solution of estimation problems for boundary-value stochastic processes. From a historical point of view, 1-D boundary-value systems and processes were first introduced by Krener [9]–[11] in order to study the internal structure of acausal systems and to formulate the stochastic realization problem for non-Markov processes such as reciprocal processes. In [6], [7], a general solution technique was developed for the estimation of boundary-value stochastic processes in one or several dimensions. This approach is extremely general, and relies on the so-called method of complementary models introduced by Weinert and Desai [12] for the study of the smoothing problem for 1-D causal systems. Specifically, it is shown that given both an internal model and appropriate boundary conditions for

Manuscript received September 13, 1988; revised June 2, 1989. The research described in this paper was supported in part by the National Science Foundation under Grant ECS-8700903, and by the Army Research Office under Grant DAAL03-86-K-0171.

B. C. Levy is with the Dept. of Electrical Engineering and Computer Science, University of California, Davis, CA 95616, U.S.A.

M. B. Adams is with the Charles Stark Draper Laboratory, 555 Technology Square, Cambridge, MA 02139, U.S.A.

A. S. Willsky is with the Dept. of Electrical Engineering and Computer Science, and the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

IEEE Log Number 9034600.

a boundary-value process, the smoothed estimate satisfies a Hamiltonian system of twice the size, and therefore of twice the order, of the original model. The reason the size is doubled is that it is necessary to estimate not only the state of the internal model of interest, but also the state of the complementary model. This approach was used to study the smoothing problem for 1-D continuous boundary-value processes in [8], and for boundary-value 1-D descriptor systems in [13]. Some rough results for the 2-D NNM smoothing problem were presented in [6, ch. 6], and the present paper is in fact an improved version of this earlier work. Subsequently, the complementary model technique was also used by Riddle and Weinert [14]–[16] to study the 2-D smoothing problem for the Helmholtz equation and for 2-D hyperbolic systems. Together with the present paper, these contributions illustrate the wide applicability of the boundary-value process smoothing solution proposed in [6], [7].

An interesting feature of the NNM smoother is that it is itself in NNM form. Thus, the class of NNM systems is closed under the smoothing operation. This property is rather satisfactory, as it indicates that NNMs are "natural" models for the study of noncausal estimation problems. From a practical point of view, because we seek to develop efficient estimation algorithms, this implies that it is important to obtain efficient NNM solution techniques. The solution proposed in this paper consists in solving the 2-D model in 1-D fashion by writing the 2-D NNM dynamics columnwise in the form of a 1-D boundary-value system of very large dimension. This 1-D system has second-order dynamics, but can be rewritten as a 1-D two-point boundary-value descriptor system (TPBVDS) of the type examined in [17]–[20], for which a number of recursive solution techniques involving different concepts of causality can be employed. Under slightly more restrictive conditions, this 1-D system can be decoupled into a family of low-order 1-D subsystems by a fast-Fourier transform (FFT)-based transformation. This decoupling technique is an extension of a method used by Hockney [21] to obtain fast Poisson solvers, and later applied by Jain and Angel [22] to a 2-D estimation problem.

In Section II, we describe 2-D NNMs, as well as the class of local boundary conditions that are used to specify the solution of these models. These conditions include as special cases periodic, Dirichlet, and Neumann boundary conditions. The transformation of a 2-D NNM into a 1-D TPBVDS is discussed in Section III, and a general solution technique is obtained for the transformed system. The FFT solver is presented in Section IV for the case where the NNM satisfies periodic boundary conditions, or has vertically symmetric dynamics with Dirichlet or Neumann conditions. The smoothing problem for stochastic 2-D NNMs is formulated in Section V, and the Hamiltonian system satisfied by the smoothed estimate is described and shown to be in NNM form. Section VI discusses two examples of 2-D NNM smoothers, corresponding to the discretized 2-D Poisson and heat equations, respectively. It is shown that the FFT decoupling technique of Section IV is applicable to both of these examples.

II. 2-D Nearest-Neighbor Models

The 2-D nearest-neighbor models (NNMs) that will be considered in this paper are of the form

$$x_{i,j} = A_1 x_{i-1,j} + A_2 x_{i+1,j} + A_3 x_{i,j-1} + A_4 x_{i,j+1} + B u_{i,j} \tag{1}$$

$$z_{i,j} = C x_{i,j} \tag{2}$$

where the state x, input u, and output z are vectors of dimension n, m, and p respectively, and A_k with $1 \le k \le 4$, B, and C are matrices of corresponding dimensions. Equation (1) indicates that the state at point (i, j) is specified by $u_{i,j}$, and by the states at points immediately to the left, to the right, above, and below point (i, j). This explains why (1) is called a nearest-neighbor model.

Models such as (1) and (2) arise naturally from the discretization of 2-D partial differential equations by finite difference methods, as as can be seen from the following examples.

Examples: NNM form of finite-difference discretizations of PDEs. For each of the 2-D examples discussed below, the continuous space variables are denoted as t and s, and the corresponding discretized variables are i and j, respectively. Furthermore, except for the heat equation, it is assumed that the same mesh size h is used to discretize t and s.

a) Poisson equation: The discretized form of

$$\nabla^2 x(t, s) = u(t, s) \tag{3}$$

is given by

$$x_{i,j} = \frac{1}{4}(x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}) - \frac{h^2}{4} u_{i,j} \tag{4}$$

which is exactly in the form (1).

b) Heat equation: Let

$$\frac{\partial}{\partial t} x(t, s) = \alpha \frac{\partial^2}{\partial s^2} x(t, s) + u(t, s) \tag{5}$$

where $\alpha > 0$. If t and s are discretized with mesh sizes h and k, that is, $t = ih$ and $s = jk$, and if backwards and central difference schemes [23] are used, respectively, to discretize $\partial x/\partial t$ and $\partial^2 x/\partial s^2$, we obtain

$$m x_{i,j} = x_{i-1,j} + n(x_{i,j-1} + x_{i,j+1}) + b u_{i,j} \tag{6}$$

where $m = 1 + 2\alpha h/k^2$, $n = \alpha h/k^2$ and $b = h$. This model can then be brought to NNM form by dividing both sides of (6) by $m > 0$. Equation (6) corresponds to an implicit discretization of the heat equation (5), where to compute $x_{i,j}$ for increasing values of i, it is necessary for each value of i to solve a linear system of equations for the coupled variables $x_{i,j}$, where j varies over all index values. It is shown in [23, p. 69] that this discretization scheme is unconditionally stable—it is stable for all choices of mesh sizes h and k. The motivation for selecting different meshes h and k to discretize t and s is that, to approximate the first-order derivative of x with respect to t and the second-order derivative with respect to s with the same degree of accuracy, one must have $h = O(k^2)$.

c) Biharmonic equation: Vector NNMs can arise in a variety of ways. One of them is of course from the discretization of higher-order PDEs, such as

$$\nabla^4 x(t, s) = u(t, s). \tag{7}$$

This equation can be decomposed as

$$\nabla^2 x(t, s) = \xi(t, s) \qquad \nabla^2 \xi(t, s) = u(t, s). \tag{8}$$

Then, using the discretization (4) of the Laplacian, and denoting

$$X_{i,j} = \begin{bmatrix} x_{i,j} \\ \xi_{i,j} \end{bmatrix},$$

we obtain

$$\begin{bmatrix} 1 & \frac{h^2}{4} \\ 0 & 1 \end{bmatrix} X_{i,j} = \frac{1}{4}(X_{i-1,j} + X_{i+1,j} + X_{i,j-1} + X_{i,j+1}) + \begin{bmatrix} 0 \\ \frac{h^2}{4} \end{bmatrix} u_{i,j} \tag{9}$$

which, after inversion of the matrix multiplying $X_{i,j}$ is in NNM form.

d) Poisson equation with a crossover term: Vector NNMs can also arise if higher-order schemes are used to discretize second-order PDEs. Sometimes the use of a higher-order scheme is dictated by the structure of the PDE itself. Consider for example

$$\left[\frac{\partial^2}{\partial t^2} + \frac{\partial^2}{\partial s^2} + a\frac{\partial^2}{\partial t \partial s}\right] x(t, s) = u(t, s) \tag{10}$$

which is elliptic, provided that parameter a is such that $|a| < 2$. Then, when a first-order finite-difference discretization scheme is used to approximate the above equation, we obtain the following 9-point stencil model

$$x_{i,j} = \frac{1}{4}(x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}) + \frac{a}{16}(x_{i-1,j-1} + x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1}) - \frac{h^2}{4}u_{i,j} \tag{11}$$

where $x_{i,j}$ depends not only on its four nearest neighbors, but also on values of x at the four corners $(i-1, j-1)$, $(i+1, j+1)$, $(i-1, j+1)$, and $(i+1, j-1)$. It can be transformed to NNM form by state augmentation. Thus, if

$$X_{i,j}^T = [x_{i,j-1} x_{i,j} x_{i,j+1}]$$

the model (11) can be rewritten as

$$X_{i,j} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{a}{16} & \frac{1}{4} & -\frac{a}{16} \\ 0 & 0 & 0 \end{bmatrix} X_{i-1,j} + \begin{bmatrix} 0 & 0 & 0 \\ -\frac{a}{16} & \frac{1}{4} & \frac{a}{16} \\ 0 & 0 & 0 \end{bmatrix} X_{i+1,j} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 \end{bmatrix} X_{i,j-1} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 1 & 0 \end{bmatrix} X_{i,j+1} + \begin{bmatrix} 0 \\ \frac{h^2}{4} \\ 0 \end{bmatrix}, \tag{12}$$

which is now in NNM form. Note that even though the second-order PDE (10) is scalar, the state $X_{i,j}$ has dimension 3. This is due to the presence of the crossover term $a\partial^2 x(t, s)/\partial t \partial s$ in (10).

For simplicity, it will be assumed below that model (1) is defined over the rectangular domain $1 \le i \le I-1$, $1 \le j \le J-1$. Then, in addition to model (1), some *boundary conditions* need to be specified. What constitutes a proper set of boundary conditions depends on the exact type of the partial difference operator (1) or the underlying PDE from which it comes. For example, if this operator is elliptic (noncausal), initial-value problems are ill posed. A general framework for specifying boundary conditions, which can accomodate operators of all types, and which can be used to model a wide class of PDE boundary conditions, consists in assuming that the boundary conditions on the edges of the rectangle $0 \le i \le I$, $0 \le j \le J$ are local in the sense that they involve only neighboring points along the boundary. An exception is that some coupling is allowed between points on opposite sides of the rectangle, enabling us to model periodic boundary conditions. We consider, therefore, the following general form for NNM boundary conditions.

Horizontal conditions:

$$V_L x_{0,j} + W_L x_{1,j} + V_R x_{I,j} + W_R x_{I-1,j} = d_{H,j} \tag{13a}$$

with $0 \le j \le J$.

Vertical conditions:

$$V_B x_{i,0} + W_B x_{i,1} + V_T x_{i,J} + W_T x_{i,J-1} = d_{V,i} \tag{13b}$$

with $1 \le i \le I-1$.

The subscripts L, R, T, and B denote the left, right, top, and bottom edges of the rectangle, respectively.

In (13a) and (13b), it is assumed that the boundary matrices V_E and W_E, with $E = L, R, B, T$ have size $2n \times n$. Thus, in conjunction with NNM model (1), the horizontal boundary conditions (13a) provide enough constraints to specify the states $x_{0,j}$ and $x_{I,j}$ with $0 \le j \le J$ on the left and right edges of the rectangle $\Omega = [0, I] \times [0, J]$. Similarly, the vertical conditions (13b) introduce sufficient constraints to enable the specification of $x_{i,0}$ and $x_{i,J}$ with $1 \le i \le I-1$ on the bottom and top edges of Ω. Note that there is a slightly asymmetry in the above specification, in the sense that the horizontal boundary condition (13a) holds for $j = 0, J$, which has the effect of adding enough constraints to specify the *corner states* $x_{0,0}$, $x_{0,J}$, $x_{I,0}$, and $x_{I,J}$. However, this is clearly an arbitrary convention, and we can just as well use the vertical condition (13b) to specify the corner states.

The conditions (13) are *local* since they involve only pairs of points located on opposite sides of the rectangle Ω. Specifically, the horizontal condition (13a) couples points $(0, j)$, $(1, j)$ located along the left edge of Ω with points (I, j) and $(I-1, j)$ on the right edge, where all these points have the same row index j. Similarly, the vertical condition (13b) couples two pairs of points along the bottom and top edges of rectangle Ω, respectively, and with the same column index i.

The motivation for coupling points located on opposite edges of Ω is that we want to be able to impose *periodic* boundary conditions. For example, if the horizontal condition (13a) takes the form

$$x_{0,j} = x_{I-1,j} \qquad x_{1,j} = x_{I,j} \qquad \text{for } 0 \le j \le J \tag{14}$$

the NNM system (1) can be viewed as being defined over a *discretized cylinder* with index set $\Omega_C = [1, I-1] \times [0, J]$.

Then, after imposing periodic horizontal conditions, if we also select periodic vertical boundary conditions

$$x_{i,0} = x_{i,J-1}, \quad x_{i,1} = x_{i,J} \quad \text{for } 1 \le i \le I-1 \tag{15}$$

the NNM is now defined over a *discretized torus*, with index set $\Omega_T = [1, I-1] \times [1, J-1]$.

Another interesting subclass of boundary conditions (13) corresponds to the case when the boundary conditions on the left and right, and bottom and top edges of Ω are *separable*, in the sense that independent boundary conditions are specified on each edge of Ω. In this case, the boundary conditions (13) take the form

$$\tilde{V}_L x_{0,j} + \tilde{W}_L x_{1,j} = d_{L,j} \quad 0 \le j \le J \tag{16a}$$

$$\tilde{V}_R x_{I,j} + \tilde{W}_R x_{I-1,j} = d_{R,j} \quad 0 \le j \le J \tag{16b}$$

$$\tilde{V}_B x_{i,0} + \tilde{W}_B x_{i,1} = d_{B,i} \quad 1 \le i \le I-1 \tag{16c}$$

$$\tilde{V}_T x_{i,J} + \tilde{W}_T x_{i-1,J} = d_{T,i} \quad 1 \le i \le I-1 \tag{16d}$$

where the boundary matrices $\tilde{V}_E$ and $\tilde{W}_E$ with $E = L, R, B, T$ have size $n \times n$. Boundary conditions of this type arise extremely frequently in the study of PDEs, and in particular can be used to model Dirichlet or Neumann boundary conditions, as is shown by considering several examples.

Examples: Boundary conditions for discretized PDEs in NNM form. The PDEs considered in the following examples are assumed to be defined over the rectangle $[0, T] \times [0, S]$, where if h and k are the mesh sizes used to discretize the continuous variables t and s, we have $T = Ih$ and $S = Jk$. Also, as for the PDE discretization examples considered earlier in this section, it will be assumed that $h = k$, except for the discretization of the heat equation.

a) Consider the Poisson equation (3) with the *mixed* boundary conditions

$$-m_L \frac{\partial}{\partial t} x(0, s) + n_L x(0, s) = d_L(s) \tag{17a}$$

$$m_R \frac{\partial}{\partial t} x(T, s) + n_R x(T, s) = d_R(s) \tag{17b}$$

$$-m_B \frac{\partial}{\partial s} x(t, 0) + n_B x(t, 0) = d_B(t) \tag{17c}$$

$$m_T \frac{\partial}{\partial s} x(t, S) + n_T x(t, S) = d_R(t). \tag{17d}$$

These boundary conditions reduce to Dirichlet conditions when $m_E = 0$ and $n_E = 1$ for $E = L, R, B, T$, and to Neumann conditions when $m_E = 1$ and $n_E = 0$ for all values of index E. Then, a straightforward discretization yields

$$V_E = n_E + m_E/h \qquad W_E = -m_E/h \tag{18}$$

for $E = L, R, B, T$, and the boundary vectors appearing in (16) are given by $d_{E,k} = d_E(kh)$, where the index k varies over $[0, J]$ for $E = L, R$, and over $[0, I]$ for $E = B, T$.

b) Consider now the heat equation (5) with initial condition

$$x(0, s) = f(s) \tag{19a}$$

and boundary conditions

$$x(t, 0) = g_B(t) \qquad x(t, S) = g_T(t). \tag{19b}$$

After discretization, we find

$$V_E = 1 \qquad W_E = 0 \qquad \text{for } E = L, B, T \tag{20a}$$

with

$$d_{L,j} = f(jk) \qquad d_{B,i} = g_B(ih) \qquad d_{T,i} = g_T(ih). \tag{20b}$$

In the above formulation, no boundary condition is specified on the right edge of Ω. This is unsatisfactory, because our NNM formulation requires that there should be as many constraints as there are variables to be computed. The key step is to observe that, as the discretized equation (6) is *causal* with respect to time, the values of $x_{I,j}$ on the right edge do not affect any of the other variables, and can therefore be assigned arbitrarily, so that the boundary condition on the right edge is assumed to have the form

$$x_{I,j} = d_{R,j} \quad 0 \le j \le J \tag{20c}$$

where $d_{R,j}$ is arbitrary.

c) Examine the Poisson equation (11) with a crossover term, and with Dirichlet boundary conditions obtained by setting $m_E = 0$ and $n_E = 1$ in (17). Then, a simple discretization of these conditions is not sufficient to specify the NNM boundary conditions, because, as was observed above, we must consider the vector NNM system (12). Furthermore, owing to the state augmentation procedure used to construct $X_{i,j}$, if the scalar discretized PDE (11) is defined over the domain $[0, I] \times [0, J]$, the domain of definition of NNM (12) is only $[0, I] \times [1, J-1]$. Over this domain, the discretized Dirichlet boundary conditions for the scalar equation can be rewritten in the NNM form (16) as

$$X_{0,j} = \begin{bmatrix} d_L((j-1)h) \\ d_L(jh) \\ d_L((j+1)h) \end{bmatrix} X_{I,j} = \begin{bmatrix} d_R((j-1)h \\ d_R(jh) \\ d_R((j+1)h) \end{bmatrix} \tag{21a}$$

$$X_{i,1} + \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} X_{i,2} = \begin{bmatrix} d_B(ih) \\ 0 \\ 0 \end{bmatrix} \tag{21b}$$

$$X_{i,J-1} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} X_{i,J-2} = \begin{bmatrix} 0 \\ 0 \\ d_T(ih) \end{bmatrix} \tag{21c}$$

III. Solution of Boundary Value Nearest-Neighbor Models

In this section, a method for computing the solution of the boundary-value problem specified by the NNM dynamics (1) and boundary conditions (13) is described. The method employed relies on a column stacking operation, whereby the variables $x_{i,j}$ along the ith column of the rectangular domain Ω are combined to form a large state vector $\mathbf{x}_i$. This procedure is used to transform the 2-D NNM dynamics, as well as the boundary and corner conditions, into an equivalent 1-D two-point boundary value system of very large size with second order dynamics. This 1-D dynamical system is then formulated as a 1-D two-point boundary value descriptor system (TPBVDS) of the type studied in [17]–[20]. Using a TPBVDS solution technique proposed in [17, app. B] and [13], a recursive procedure is obtained for solving NNM models. It relies on decoupling the TPBVDS dynamics into forward and backward stable filters with zero initial and final conditions, respectively. The true boundary conditions are then taken into account by adding a correction

term to the solution obtained for zero boundary conditions.

A. Column Stacking and Well Posedness

This section established the notation that will be used in the remainder of this paper. As indicated above, the first step of our solution is to perform a column-stacking operation, where the state, input, and output vectors along the ith column of rectangle $\Omega = [0, I] \times [0, J]$ are represented by

$$\mathbf{x}_i^T = [x_{i,0}^T \;\; x_{i,1}^T \;\cdots\; x_{i,J-1}^T \;\; x_{i,J}^T], \quad \mathbf{u}_i^T = [u_{i,1}^T \;\cdots\; u_{i,J-1}^T] \tag{22a}$$

$$\mathbf{z}_i^T = [z_{i,0}^T \;\; z_{i,1}^T \;\cdots\; z_{i,J-1}^T \;\; z_{i,J}^T]. \tag{22b}$$

Here $\mathbf{x}_i$, $\mathbf{u}_i$, and $\mathbf{z}_i$ have dimensions $n(J+1)$, $m(J-1)$, and $p(J+1)$, respectively. Note that $\mathbf{x}_i$ and $\mathbf{z}_i$ have two more block entries that $\mathbf{u}_i$, because $x_{i,j}$ and $z_{i,j}$ are defined on the edges of the rectangular domain Ω, whereas $u_{i,j}$ is only defined in the interior. Then, by combining the NNM relations (1) for a fixed value of i and $1 \le j \le J-1$ with the vertical boundary conditions (13b) for the same value of i, we obtain the 1-D dynamics

$$\Phi_+ \mathbf{x}_{i+1} + \Phi_0 \mathbf{x}_i + \Phi_- \mathbf{x}_{i-1} = \mathbf{n}_i \qquad 1 \le i \le I-1 \tag{23}$$

$$\mathbf{z}_i = (I \otimes C)\mathbf{x}_i \tag{24}$$

where $\otimes$ denotes the Kronecker product of two matrices [24], with

$$\Phi_0 = \begin{bmatrix} V_B & W_B & & & & W_T & V_T \\ -A_3 & I & -A_4 & & \mathbf{0} & & \\ & -A_3 & I & -A_4 & & & \\ & & & \cdots & & & \\ & \mathbf{0} & & & \cdots & & \\ & & & & -A_3 & I & -A_4 \end{bmatrix} \tag{25a}$$

$$\Phi_+ = \begin{bmatrix} 0 & & & \\ -A_2 & 0 & & \\ & -A_2 & \ddots & \\ \mathbf{0} & & \ddots & -A_2 \end{bmatrix}$$

$$\Phi_- = \begin{bmatrix} 0 & & & \\ -A_1 & 0 & & \\ & -A_1 & \ddots & \\ \mathbf{0} & & \ddots & -A_1 \end{bmatrix} \tag{25b}$$

and

$$\mathbf{n}_i = \begin{bmatrix} d_{V,i} \\ (I \otimes B)\mathbf{u}_i \end{bmatrix} \tag{25c}$$

As the boundary matrices V_B, V_T, W_B, and W_T have size $2n \times n$, it is easy to check that the matrices Φ_k with $k = 0, -, +$ are square and have dimension $n(J+1)$. The relation (23) defines, therefore, a 1-D system with second-order dynamics evolving over the interval $[0, I]$ and driven by inputs $\mathbf{n}_i$ which are expressed in terms of the inputs $u_{i,j}$ of the NNM and of the boundary vector $d_{V,i}$ associated with the vertical conditions on the bottom and top edges of rectangle Ω.

By considering also the horizontal NNM boundary condition (13a) on the left and right edges of Ω, we obtain the boundary condition

$$\Gamma_L \mathbf{x}_0 + \Delta_L \mathbf{x}_1 + \Gamma_R \mathbf{x}_J + \Delta \mathbf{x}_{J-1} = \mathbf{d}_H \tag{26}$$

for system (23), where

$$\Gamma_L = I \otimes V_L \qquad \Gamma_R = I \otimes V_R \tag{27a}$$

$$\Delta_L = I \otimes W_L \qquad \Delta_R = I \otimes W_R \tag{27b}$$

and

$$\mathbf{d}_H^T = [d_{H,0}^T \;\; d_{H,1}^T \;\cdots\; d_{H,J-1}^T \;\; d_{H,J}^T]. \tag{27c}$$

Noting again that the boundary matrices V_L, V_R and W_L, W_T have size $2n \times n$, it is easy to check that Γ_L, Γ_R, Δ_L, and Δ_R have size $2(J+1)n \times (J+1)n$, and that vector $\mathbf{d}_H$ has dimension $2(J+1)n$. Thus, the boundary conditions (26) and dynamics (23) define a boundary value system over $[0, I]$, where the number of constraints imposed by (23) and (26) equals the total number of variables that need to be computed, namely vectors $\mathbf{x}_i$ for $0 \le i \le I$. One possible method of solving this system consists in combining all the equations that define it into a single matrix equation of very large dimension of the form

$$\Sigma \mathbf{x} = \mathbf{n} \tag{28a}$$

$$\mathbf{x}^T = [\mathbf{x}_0^T \;\; \mathbf{x}_1^T \;\cdots\; \mathbf{x}_{I-1}^T \;\; \mathbf{x}_I^T] \qquad \mathbf{n}^T = [\mathbf{d}_H^T \;\; \mathbf{n}_1^T \;\; \mathbf{n}_2^T \;\cdots\; \mathbf{n}_{I-1}^T] \tag{28b}$$

where

$$\Sigma = \begin{bmatrix} \Gamma_L & \Delta_L & & & & \Delta_R & \Gamma_R \\ \Phi_- & \Phi_0 & \Phi_+ & & \mathbf{0} & & \\ & \Phi_- & \Phi_0 & \Phi_+ & & & \\ & & & \cdots & & & \\ & \mathbf{0} & & & \cdots & & \\ & & & & \Phi_- & \Phi_0 & \Phi_+ \end{bmatrix} \tag{29}$$

is a matrix of size $(I+1)(J+1)n$. Then, the 1-D boundary-value system (23), (26) is well posed over interval $[0, I]$— that is, there exists a unique solution $\mathbf{x}_i$ with $0 \le i \le I$ for all possible choices of inputs $\mathbf{n}_i$ and boundary vector $\mathbf{d}_H$, if and only if Σ is invertible. Because system (23), (26) was obtained from the original NNM by column stacking, the invertibility of Σ is therefore a necessary and sufficient condition for the well-posedness of the NNM (1), (13). Note that the concept of well-posedness is used here in a mathematical sense, where we require only that the matrix equation (28a) admit a unique solution. By contrast, for numerical well-posedness (in order to guarantee that the solution of (28a) does not change significantly for small perturbations of the matrix Σ), Σ would need to have a low condition number [25, p. 27]. By using an argument similar to the one appearing in Theorem 1 of [26], it is also easy to check that the invertibility of Σ implies that the second-order dynamics (23) must be *regular*. Thus, the determinant of the polynomial

matrix

$$\Phi(z) = \Phi_+ z^2 + \Phi_0 z + \Phi_- \tag{30}$$

is not identically zero for all z.

In practice, the matrix Σ has such a huge dimension that it is neither possible nor desirable to invert it directly. In the special case when Σ is obtained by discretizing an elliptic PDE, iterative inversion methods, such as the successive overrelaxation (SOR) [27], preconditioned conjugate gradient [25, ch. 10], or multigrid [28] methods can be employed to solve (28a). Although these solution techniques are limited in scope, they are usually more efficient than the totally general solution technique described below, which applies to NNM operators of all types.

B. Stable Two-Filter Solution

The general solution technique that we propose relies on transforming the 1-D dynamics (23) in such a way that stable forward and backward recursions can be used to compute $\mathbf{x}_i$. In some sense, this method falls within the class of marching methods [29], [30]. Marching methods were originally developed when it was realized that, by column stacking, noncausal 2-D models such as (1) could be transformed into 1-D dynamical systems such as (23). Then, in the special case when Φ_+ is invertible, (23) can be expressed as

$$\mathbf{x}_{i+1} = -\Phi_+^{-1}[\Phi_0 \mathbf{x}_i + \Phi_- \mathbf{x}_{i-1} - \mathbf{n}_i] \tag{31}$$

which is now a causal system that can be used to compute $\mathbf{x}_i$ recursively, provided that the boundary condition (26) is properly taken into account. In addition to requiring that either Φ_+ or Φ_- be invertible, one major drawback of the naive approach described above is that there is no guarantee that the causal system (31) is stable. An important criticism of marching methods, at least in this simplistic form, has therefore been that they are numerically unstable, and are not appropriate for solving NNMs on large lattices. The solution presented here can be viewed as a *stabilized marching method*, where instead of attempting to propagate the whole system (23) in the forward (or backward) direction, we break it into smaller parts, which are stable when propagated in their respective forward and backward directions.

Instead of considering directly the second-order system (23), we transform it into a TPBVDS of the type examined in [17]–[20]. To do so, consider the augmented state

$$\mathbf{q}_i = \begin{bmatrix} \mathbf{x}_{i-1} \\ \mathbf{x}_i \end{bmatrix}. \tag{32}$$

Then, the dynamics (23) and (24) and boundary condition (26) can be expressed as

$$E\mathbf{q}_{i+1} = F\mathbf{q}_i + G\mathbf{n}_i \qquad 1 \le i \le I \tag{33}$$

$$\mathbf{z}_i = H\mathbf{q}_i \tag{34}$$

and

$$U_L \mathbf{q}_1 + U_R \mathbf{q}_I = \mathbf{d}_H \tag{35}$$

respectively, where

$$E = \begin{bmatrix} \Phi_0 & \Phi_+ \\ I & 0 \end{bmatrix} \qquad F = \begin{bmatrix} -\Phi_- & 0 \\ 0 & I \end{bmatrix} \tag{36a}$$

$$G = \begin{bmatrix} I \\ 0 \end{bmatrix} \qquad H = [0 \quad I \otimes C] \tag{36b}$$

$$U_L = [\Delta_L \quad \Gamma_L] \qquad U_R = [\Gamma_R \quad \Delta_R]. \tag{36c}$$

The relations (33)–(36) define a TPBVDS over the interval $[1, I]$. This system has first-order dynamics, and it is easy to check that

$$|zE - F| = |\Phi(z)| \tag{37}$$

where $\Phi(z)$ is the second-order matrix polynomial defined in (30), so that no new dynamics have been introduced by going from (23) to (33). Owing to the simple nature of the augmentation procedure (32), we can also conclude that the TPBVDS (33)–(36) is well posed over the interval $[1, I]$ if and only if the second-order system (23) with boundary condition (26) is well posed over $[0, I]$, which in turn was shown to be equivalent to the well-posedness of the original NNM system. It has been shown [17] that an arbitrary TPBVDS of the form (33), (35) is well posed if and only if the matrix

$$S = U_L E^{I-1} + U_R F^{I-1} \tag{38}$$

is invertible. The invertibility of S in (38) can therefore be used to characterize the well-posedness of the NNM (1), (13). As the size of this matrix is "only" $2(J+1)n$, the invertibility of S is much easier to test than that of the matrix Σ which was used to characterize NNM well-posedness in (28a).

At this point, the NNM problem has been reduced to the solution of a TPBVDS over a finite interval. The system structure of TPBVDSs was studied in detail in [17]–[20], and several solution techniques were proposed in [17, app. B] and [13]. As mentioned previously, the solution described here relies on breaking the descriptor dynamics (33) into smaller parts that are causal and stable in the forward and backward directions, respectively. Specifically, because the NNM that we consider is assumed to be well posed, the matrix pencil $zE - F$ is regular, and according to Weierstrass's canonical decomposition of a regular pencil [31], there are some invertible matrices M and T such that

$$M(zE - F)T = \begin{bmatrix} zI - F_f & 0 \\ 0 & zF_b - I \end{bmatrix} \tag{39}$$

where the eigenvalues of matrices F_f and F_b have magnitude less than or equal to 1. Reliable numerical methods for performing the decomposition (39) are described in [32]. Furthermore, if $|zE - F|$ has no zero on the unit circle, then all the eigenvalues of F_f and F_b are strictly inside the unit circle. If we define B_f and B_b such that

$$MB = \begin{bmatrix} B_f \\ B_b \end{bmatrix} \tag{40}$$

then the transformed state variables

$$\begin{bmatrix} \mathbf{q}_{f,i} \\ \mathbf{q}_{b,i} \end{bmatrix} = T\mathbf{q}_i \tag{41}$$

satisfy the forward and backward recursions

$$\mathbf{q}_{f,i+1} = F_f \mathbf{q}_{f,i} + B_f \mathbf{n}_i \tag{42a}$$

$$\mathbf{q}_{b,i} = F_b \mathbf{q}_{b,i+1} - B_b \mathbf{n}_i. \tag{42b}$$

These recursions are asymptotically stable if $zE - F$ has no zero on the unit circle. Under the transformation (41), the boundary condition (35) takes the form

$$[U_{L,f} \quad U_{L,b}]\begin{bmatrix}\mathbf{q}_{f,1}\\ \mathbf{q}_{b,1}\end{bmatrix} + [U_{R,f} \quad U_{R,b}]\begin{bmatrix}\mathbf{z}_{f,I}\\ \mathbf{q}_{b,I}\end{bmatrix} = \mathbf{d}_H \tag{43}$$

where

$$[U_{L,f} \quad U_{L,b}] = U_L T^{-1}\,[U_{R,f} \quad U_{R,b}] = U_R T^{-1}. \tag{44}$$

Note that although the forward and backward dynamics (42a) and (42b) for $\mathbf{q}_f$ and $\mathbf{q}_b$ are decoupled, the boundary conditions remain coupled, so that $\mathbf{q}_f$ and $\mathbf{q}_b$ cannot be computed separately. Let $\mathbf{q}^0_{f,i}$ and $\mathbf{q}^0_{b,i}$ be the solutions of (42a) and (42b) with zero initial and final conditions, respectively. Then

$$\mathbf{q}_{f,i} = F_f^{i-1}\mathbf{q}_{f,1} + \mathbf{q}^0_{f,i} \tag{45a}$$

$$\mathbf{q}_{b,i} = F_b^{I-i}\mathbf{q}_{b,I} + \mathbf{q}^0_{b,i}. \tag{45b}$$

Substituting (45) inside (43), and solving for $\mathbf{q}_{f,1}$ and $\mathbf{q}_{b,I}$ gives

$$\begin{bmatrix}\mathbf{q}_{f,1}\\ \mathbf{q}_{b,I}\end{bmatrix} = K^{-1}(\mathbf{d}_H - U_{R,f}\mathbf{q}^0_{f,I} - U_{L,b}\mathbf{q}^0_{b,1}) \tag{46}$$

where

$$K = [U_{L,f} + U_{R,f}F_f^{I-1} \quad U_{R,b} + U_{L,b}F_b^{I-1}]. \tag{47}$$

Note that the transformation (39)-(41) does not affect the well-posedness of the TPBVDS, so that the matrix K is invertible if and only if S is invertible in (38). Finally, substituting (46) inside (45), we find

$$\begin{bmatrix}\mathbf{q}_{f,i}\\ \mathbf{q}_{b,i}\end{bmatrix} = \begin{bmatrix}F_f^{i-1} & 0\\ 0 & F_b^{I-i}\end{bmatrix}K^{-1}(\mathbf{q} - U_{R,f}\mathbf{q}^0_{f,I} - U_{L,b}\mathbf{q}^0_{b,1}) + \begin{bmatrix}\mathbf{q}^0_{f,i}\\ \mathbf{q}^0_{b,i}\end{bmatrix}. \tag{48}$$

The solution in the original basis can then be obtained by inverting (41).

From a practical point of view, the solution technique described above consists in propagating the forward and backward filters (42a) and (42b) for $\mathbf{q}^0_{f,i}$ and $\mathbf{q}^0_{b,i}$, and then combining the resulting values with the boundary condition (43) to obtain $\mathbf{q}_{f,i}$ and $\mathbf{q}_{b,i}$ via (48). The most computationally demanding part of this algorithm is the computation of $\mathbf{q}^0_{f,i}$ and $\mathbf{q}^0_{b,i}$.

The above TPBVDS solution is similar to the Mayne-Fraser [33], [34] two-filter formula for the 1-D fixed-interval smoothing problem. Although it may seem that there is little relation between the fixed-interval smoothing problem for discrete-time causal systems and the solution of TPBVDSs, it turns out that the 1-D discrete-time smoother can be expressed as a TPBVDS (see [6, section 5.3]), which expains why the same solution technique can be used for these two problems.

The TPBVDS solution described here is not the only one that can be developed. In [17] an alternative solution method is proposed that relies on stable recursions propagating inwards and outwards with respect to the center of the interval where the TPBVDS is defined. This choice is a manifestation of the fact that, because causality appears here only as a computational device, we are not restricted to process the 2-D NNM data in any particular order.

IV. FFT Solver

One drawback of the NNM solution described above is that the vectors $\mathbf{x}_i$ obtained by column stacking have very large size. The matrices E and F appearing in the TPBVDS (33)-(36) have size $2(J+1)n$, and therefore the matrices F_f and F_b obtained by pencil decomposition have a very large dimension. In addition, even if E and F are sparse, there is no guarantee that F_f and F_b will also be sparse, so that the forwards and backwards recursions (42) require in general a large amount of computation. In this section, we consider several special cases where some additional structure is present, which can be exploited to obtain *fast* NNM solvers. Specifically, we consider the cases where a) the NNM is defined over a discretized cylinder, and b) the NNM dynamics (1) satisfy the symmetry condition $A_3 = A_4$, and the boundary conditions on the bottom and top edges are either of Dirichlet or Neumann type. For these cases, the FFT or the discrete sine and cosine transforms (DST, DCT) can be used to transform the high-order TPBVDS obtained in (33)-(35) into decoupled low-order 1-D TPBVDSs, which can be solved in parallel. As fast algorithms can be used to implement the FFT, DST and DCT and their inverses, this solution technique is very efficient. It is worth noting that the use of the FFT was first proposed by Hockney [21] to obtain a fast Poisson solver. The FFT was later employed by Jain and Angel [22] (see also Jain [35]) to obtain an efficient solution for a 2-D estimation problem expressed in terms of the Poisson equation, and it was used in [14]-[16] to get fast smoothing algorithms for hyperbolic PDE models. The fast NNM solver described here can be viewed as an extension of these earlier results.

A. NNM Over a Discretized Cylinder

In the first case, it is assumed that the vertical boundary conditions (14a) are *periodic*

$$x_{i,0} = x_{i,J-1} \qquad x_{i,1} = x_{i,J} \qquad \text{for } 1 \le i \le I-1 \tag{49}$$

in which case the domain Ω corresponds to a discretized cylinder. Then, it is easy to check that the components $x_{i,0}$ and $x_{i,J}$ need not be included in the stacked vector $\mathbf{x}_i$, whose dimension is therefore only $n(J-1)$, and in (23), we can identify

$$\Phi_0 = I \otimes I - Z_c^T \otimes A_3 - Z_c \otimes A_4 \tag{50a}$$

$$\Phi_- = -I \otimes A_1 \qquad \Phi_+ = -I \otimes A_2 \qquad \mathbf{n}_i = (I \otimes B)\mathbf{u}_i \tag{50b}$$

where Z_c is the $(J-1)\times(J-1)$ circular shift matrix

$$Z_c = \begin{bmatrix}0 & 1 & & & \\ & 0 & 1 & & 0\\ & & 0 & \ddots & \\ 0 & & & \ddots & 1\\ 1 & & & & 0\end{bmatrix}. \tag{51}$$

The special structure of the 1-D system specified by (23), (26), (27), and (50) can be exploited by performing a state transformation on $\mathbf{x}_i$ which decouples this system into $J-1$ subsystems of dimension n. To do so, let D be the $(J-$

1) × (J − 1) discrete Fourier transform (DFT) matrix with entries

$$d_{k,j} = \frac{1}{(J-1)^{1/2}}\,\omega^{(k-1)(j-1)} \qquad 1 \le k, j \le J-1 \tag{52a}$$

where

$$\omega = e^{-i2\pi/(J-1)}. \tag{52b}$$

The matrix D has the property that it is unitary, that is, $DD^H = D^HD = I$, and it diagonalizes Z_c, so that

$$Z_c = D\Lambda D^H \qquad \text{with } \Lambda = \text{diag}\,\{\omega^{j-1}\}. \tag{53}$$

Then, consider the state transformation

$$(D^H \otimes I)\mathbf{x}_i = \xi_i = [\xi_{i,1}^T \cdots \xi_{i,j}^T \cdots \xi_{i,J-1}^T]^T \tag{54a}$$

where the new state vector ξ_i is partitioned into subvectors $\xi_{i,j}$ of size n. Similarly, let

$$(D^H \otimes I)\mathbf{u}_i = v_i \qquad (D^H \otimes I)\mathbf{d}_H = \delta \tag{54b}$$

where v_i and δ are also partitioned into into vector entries $v_{i,j}$ and δ_j. Using the transformation (54), and taking into account (50), (53), as well as the Kronecker product identities

$$(A \otimes B)(C \otimes D) = AC \otimes BD \qquad (A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \tag{55}$$

the 1-D system (23), (26) is transformed into $J-1$ decoupled subsystems of the form

$$(I - \omega^{-(j-1)}A_3 - \omega^{j-1}A_4)\xi_{i,j} = A_1\xi_{i-1,j} + A_2\xi_{i+1,j} + Bv_{i,j} \tag{56}$$

where $1 \le j \le J-1$, and with boundary conditions

$$V_L\xi_{0,j} + W_L\xi_{1,j} + V_R\xi_{I,j} + W_R\xi_{I-1,j} = \delta_j. \tag{57}$$

The dynamics (56) and boundary conditions (57) have exactly the same structure as the second-order boundary value system (23), (26). This system can therefore be expressed in TPBVDS form and solved by the two-filter solution technique of Section III. The advantage of this approach is that the decoupled systems (56), (57) have size n, whereas the system (23), (26) has dimension $(J+1)n$. Thus, the total number of operations required to solve the family of decoupled systems (56), (57) is $O(IJ)$, whereas the complexity of the algorithm presented in Section III is $O(IJ^2)$. In fact, the most computationally demanding step of the fast NNM solver described above is not the solution of subsystems (56), (57). It is the implementation of the transformations (54b) of the original inputs and boundary vectors and of the inverse transformation

$$\mathbf{x}_i = (D \otimes I)\xi_i \tag{58}$$

which relates the solution of the decoupled TPBVDSs to the original coordinate system. Because of its Kronecker product form, the transform (58) consists of n decoupled FFTs of length $J-1$, represented here by D. The number of operations required by (56) is therefore $O(J \log J)$, and, because this transformation as well as transformations (54) must be performed for every value of i, the complexity of the fast NNM solver described above is $O(IJ \log J)$.

B. Vertically Symmetric NNMs

NNMs defined over a discretized cylinder are not the only ones that give rise to fast solvers. When the NNM dynamics (1) have the vertical symmetry $A_3 = A_4$ (which is the case for the Poisson and heat equations, as well as the biharmonic equation described in Section II), and when the boundary conditions on the bottom and top edges are of Dirichlet or Neumann type, it is possible to obtain fast solvers.

We consider first the case of *Dirichlet conditions*. In this case, we have

$$x_{i,0} = d_{B,i} \qquad x_{i,J} = d_{T,i} \tag{59}$$

so that it is not necessary to include $x_{i,0}$ and $x_{i,J}$ in the stacked vector $\mathbf{x}_i$ introduced in (32a). This vector has therefore dimension $n(J-1)$. With this observation, the dynamics (23) take the form

$$\Phi_0 = I \otimes I - \Pi \otimes A_3 \qquad \Phi_- = -I \otimes A_1$$
$$\Phi_+ = -I \otimes A_2 \tag{60a}$$

$$\mathbf{n}_i = (I \otimes B)\mathbf{u}_i + \begin{bmatrix} A_3 d_{B,i} \\ 0 \\ \vdots \\ 0 \\ A_3 d_{T,i} \end{bmatrix} \tag{60b}$$

with

$$\Pi = Z + Z^T \tag{61a}$$

where Z denotes here the $(J-1) \times (J-1)$ truncated shift matrix

$$Z = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & 0 \\ & & \ddots & \ddots & \\ & 0 & & 0 & 1 \\ & & & & 0 \end{bmatrix} \tag{61b}$$

Let S denote the $(J-1) \times (J-1)$ discrete sine transform (DST) matrix with entries

$$s_{k,j} = \left(\frac{2}{J}\right)^{1/2} \sin\left(kj\,\frac{\pi}{J}\right) \qquad 1 \le k, j \le J-1. \tag{62}$$

The matrix S is symmetric and orthonormal, that is, $S = S^T$ and $S^2 = I$, and it diagonalizes Π, so that

$$S\Pi S^T = \Lambda = \text{diag}\,\{\lambda_j\} \tag{63}$$

where $\lambda_j = 2\cos(j\pi/J)$ with $1 \le j \le J-1$. Thus, if we replace D^H by S in the state transformation (54a) and in the definition (54b) of δ, and if we define

$$\nu_i = (S \otimes I)\mathbf{n}_i \tag{64}$$

where $\mathbf{n}_i$ is given by (60b), then the 1-D system (23), (26), whose dynamics and boundary matrices are specified respectively by (60a) and (27), can be decomposed into $J-1$ decoupled subsystems of the form

$$(I - \lambda_j A_3)\xi_{i,j} = A_1\xi_{i-1,j} + A_2\xi_{i+1,j} + \nu_{i,j} \tag{65a}$$

with boundary conditions

$$V_L\xi_{0,j} + W_L\xi_{1,j} + V_R\xi_{I,j} + W_R\xi_{I-1,j} = \delta_j \tag{65b}$$

where $1 \le j \le J-1$. These subsystems can be written in TPBVDS form and solved in parallel. Furthermore, the FFT can be used to implement the discrete sine transform S, so

that the complexity of the resulting fast NNM solver is identical to that of Section IV-A.

Consider now the case where the NNM is such that $A_3 = A_4$, but where the boundary conditions on the bottom and top edges are now *Neumann conditions*, that is

$$x_{i,0} - x_{i,1} = d_{B,i} \qquad x_{i,J} - x_{i,J-1} = d_{T,i} \tag{66}$$

for $1 \le i \le I-1$. In this case, the expressions (60) for the 1-D dynamics remain unchanged, except that the matrix Π appearing in these expressions is now defined as

$$\Pi = Z + Z^T + \text{diag}\,\{1, 0, \cdots, 0, 1\} = \begin{bmatrix} 1 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & 1 & 0 & & \mathbf{0} & \\ & & 1 & & & \\ & \mathbf{0} & & \ddots & 1 & \\ & & & & 0 & 1 \\ & & & & 1 & 1 \end{bmatrix} \tag{67}$$

In order to diagonalize Π, we can use the $(J-1) \times (J-1)$ discrete cosine transform (DCT) matrix K whose entries are

$$k_{s,j} = \begin{cases} \left[\dfrac{1}{J-1}\right]^{1/2} & \text{for } j = 1 \\ \left[\dfrac{2}{J-1}\right]^{1/2} \cos\left[(s - \frac{1}{2})(j-1)\dfrac{\pi}{J-1}\right] & \text{for } 2 \le j \le J-1 \end{cases} \tag{68}$$

with $1 \le I \le J-1$. The matrix K is orthonormal, that is, $KK^T = K^TK = I$, and it diagonalizes Π, so that

$$K^T \Pi K = \Lambda = \text{diag}\,\{\lambda_j\} \tag{69a}$$

with

$$\lambda_j = 2 \cos\left[(j-1)\frac{\pi}{J-1}\right] \qquad 1 \le j \le J-1. \tag{69b}$$

Consequently, if K plays the same role as D and S in the state, input and boundary vector transformations considered earlier in this section, the 1-D system (23), (26) with dynamics and boundary matrices given by (60) and (27) is transformed into $J-1$ decoupled subsystems specified by (65), where the only difference is that the eigenvalues λ_j appearing in these systems are now given by (69b). These subsystems can be solved in parallel, and as the FFT can also be used to implement the DCT, the complexity of the resulting algorithm is $O(IJ \log J)$.

V. NNM Smoother

In this section we examine the smoothing problem for 2-D random fields described by a NNM driven by white Gaussian noise. Note that as NNMs are intrinsically acausal, the only linear estimation problem that preserves the acausality of the system formulation is the smoothing problem. Given noisy NNM observations over the rectangle Ω, the general approach developed in [6], [7] for estimating boundary value processes is used to show that the smoother dynamics and boundary conditions are themselves in the form of a NNM of twice the size of the original NNM. Thus the class of NNMs, unlike say the class of 1-D causal systems, is closed under the smoothing operation. A consequence of this observation is that the two-filter solution techniques described in Sections III and IV can be used to compute the NNM smoothed estimate.

A. Operator Characterization of the NNM Smoother

The NNM smoothing problem can be described as follows. First, assume that the input sequence $u_{i,j}$ driving the NNM (1) is a zero-mean white Gaussian noise sequence defined over the interior $\tilde{\Omega} = [1, I-1] \times [1, J-1]$ of rectangle Ω, and with intensity

$$E[u_{i,j} u_{k,s}^T] = Q\delta_{ik}\delta_{js}. \tag{70}$$

The boundary vectors $d_{H,j}$ and $d_{V,i}$ appearing in boundary conditions (13) are also assumed to be zero-mean white Gaussian noise sequences that are mutually uncorrelated, as well as uncorrelated with the noise $u_{i,j}$ and with intensities

$$E[d_{H,j} d_{H,s}^T] = \Pi_H \delta_{js} \qquad E[d_{V,i} d_{V,k}^T] = \Pi_V \delta_{ik}. \tag{71}$$

Given the definitions of the inputs and boundary vectors above, the state $x_{i,j}$ of NNM (1) is a zero-mean 2-D Gaussian random field. We are given some noisy observations

$$y_{i,j} = Cx_{i,j} + r_{i,j} \qquad (i,j) \in \tilde{\Omega} \tag{72}$$

of this field over the interior domain $\tilde{\Omega}$. Here $r_{i,j}$ is a zero-mean white Gaussian noise sequence uncorrelated with the driving noise $u_{i,j}$ and the boundary and corner vectors, and with intensity

$$E[r_{i,j} r_{k,s}^T] = R\delta_{i,k}\delta_{j,s} \tag{73}$$

where $R > 0$. In addition to the above interior measurements, we may also be given some boundary measurements that have a structure similar to the boundary and corner conditions described in Section II

$$y_{H,j} = H_L x_{0,j} + G_L x_{1,j} + H_R x_{I,j} + G_R x_{I-1,j} + r_{H,j} \tag{74a}$$

$$y_{V,i} = H_B x_{i,0} + G_B x_{i,1} + H_T x_{i,J} + G_T x_{i,J-1} + r_{V,i}. \tag{74b}$$

In the above measurements, $r_{H,j}$ and $r_{V,i}$ are assumed to be zero-mean white Gaussian noises, which are mutually uncorrelated, and uncorrelated with u, r, and the boundary vectors, and with intensity

$$E[r_{H,j} r_{H,s}^T] = R_H \delta_{js} \qquad E[r_{V,i} r_{V,k}^T] = R_V \delta_{ik}. \tag{75}$$

The motivation for considering boundary observations that have a form different from the interior observations is that (74) can be used to model the case where we observe the discretized normal derivative of a PDE along the boundary of a domain. For example, when the normal derivative is observed along the left and right edges of Ω, if h is the discretization mesh, the measurements can be expressed as

$$y_{L,j} = \frac{1}{h}(x_{0,j} - x_{1,j}) + r_{L,j} \qquad y_{R,j} = \frac{1}{h}(x_{I,j} - x_{I-1,j}) + r_{R,j} \tag{76}$$

where $r_{L,j}$ and $r_{R,j}$ are uncorrelated white Gaussian noises. These boundary measurements clearly correspond to a special case of (74a). An example of this type appears in the inverse resistivity problem considered in [36], where a potential distribution is imposed on the boundary of a resis-

tive medium, and the resulting current density, which is proportional to the normal derivative of the potential, is measured on the boundary.

The *NNM smoothing problem* consists in computing the conditional mean

$$\hat{x}_{i,j} = E[x_{i,j}|\mathbf{Y}] \tag{77}$$

where $\mathbf{Y}$ denotes the Hilbert space of zero-mean random variables spanned by the interior observations $y_{i,j}$ for $(i, j) \in \tilde{\Omega}$, and by the boundary observations $y_{H,j}$ with $0 \le j \le J$, and $y_{V,i}$ with $1 \le i \le I - 1$. To solve this problem, we will use the general results obtained in [6], [7] for the estimation of boundary value processes. However, because these results are expressed in abstract operator form, our first step will be to rewrite the NNM (1), (13) and observations (72) and (74) in operator form.

In this framework, the NNM dynamics (1) take the form

$$(L\mathbf{x})_{i,j} = Bu_{i,j} \tag{78}$$

where, if D_1 and D_2 denote, respectively, the backward horizontal and vertical shift operators

$$D_1 x_{i,j} = x_{i-1,j} \qquad D_2 x_{i,j} = x_{i,j-1} \tag{79a}$$

we have

$$L = I - A_1 D_1 - A_2 D_1^{-1} - A_3 D_2 - A_4 D_2^{-1}. \tag{79b}$$

Note that in (78) x and Lx are defined over the domains Ω and $\tilde{\Omega}$, respectively. Let also Δ_b be the restriction operator such that

$$\mathbf{x}_b = \Delta_b \mathbf{x} \tag{80}$$

is the restriction of $\mathbf{x}$ to the first and last two columns and rows of Ω. Define

$$\mathbf{x}_L = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} \qquad \mathbf{x}_R = \begin{bmatrix} \mathbf{x}_I \\ \mathbf{x}_{I-1} \end{bmatrix} \tag{81a}$$

where the vectors $\mathbf{x}_i$ are defined as in (22a), and let

$$\mathbf{x}_B = \begin{bmatrix} \mathbf{x}'_0 \\ \mathbf{x}'_1 \end{bmatrix} \qquad \mathbf{x}_T = \begin{bmatrix} \mathbf{x}'_J \\ \mathbf{x}'_{J-1} \end{bmatrix} \tag{81b}$$

where

$$\mathbf{x}_j'^T = [x_{1,j}^T \quad x_{2,j}^T \quad \cdots \quad x_{I-1,j}^T] \tag{82}$$

is the vector obtained by scanning the states $x_{i,j}$ along the jth row of Ω, where we omit the first and last elements of each row. Then, the restriction of $\mathbf{x}_b$ can be represented in vector form as

$$\mathbf{x}_b^T = [\mathbf{x}_L^T \quad \mathbf{x}_R^T \quad \mathbf{x}_B^T \quad \mathbf{x}_T^T] \tag{83}$$

and the boundary conditions (13) can be written in operator form as

$$V\mathbf{x}_b = \mathbf{d}_b \tag{84a}$$

with

$$V = \begin{bmatrix} \Gamma_L & \Delta_L & \Gamma_R & \Delta_R & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Gamma_B & \Delta_B & \Gamma_T & \Delta_T \end{bmatrix} \tag{84b}$$

and

$$\mathbf{d}_b = \begin{bmatrix} \mathbf{d}_H \\ \mathbf{d}_V \end{bmatrix}. \tag{84c}$$

The matrices Γ_L, Γ_R, Δ_L, and Δ_R, and vector $\mathbf{d}_H$ appearing in the above expressions are defined in (27), and

$$\Gamma_B = I \otimes V_B \qquad \Gamma_T = I \otimes V_T \tag{85a}$$

$$\Delta_B = I \otimes W_B \qquad \Delta_T = I \otimes W_T \tag{85b}$$

$$\mathbf{d}_V^T = [d_{V,1}^T \quad d_{V,2}^T \quad \cdots \quad d_{V,J-1}^T], \tag{85c}$$

where the matrices Γ_B, Γ_T, Δ_B, and Δ_T have size $2(I-1)n \times (I-1)n$ and the vector $\mathbf{d}_V$ has dimension $2(I-1)n$. Finally, the vector $\mathbf{d}_b$ given by (84c) is a zero-mean Gaussian vector with variance

$$\Pi_b = E[\mathbf{d}_b \mathbf{d}_b^T] = \begin{bmatrix} I \otimes \Pi_H & 0 \\ 0 & I \otimes \Pi_V \end{bmatrix}. \tag{86}$$

A minor technical issue connected with the definition (83) of boundary vector $\mathbf{x}_b$ is that several entries, namely $x_{1,0}$, $x_{1,1}$, $x_{1,J-1}$, $x_{1,J}$ and $x_{I-1,0}$, $x_{I-1,1}$, $x_{I-1,J-1}$, $x_{I-1,J}$ appear *twice* in $\mathbf{x}_b$. For example $x_{1,0}$ is the first entry of $\mathbf{x}_1$ in $\mathbf{x}_L$ as well as the first entry of $\mathbf{x}'_0$ in $\mathbf{x}_B$. The objective of this rather odd definition is to make sure that the smoother boundary conditions (91) below provide enough constraints to specify the smoother completely, and in particular at the *four corners* of domain Ω. A consequence of this choice, however, is that constraints have to be added that force the duplicate entries to be the same. As this is primarily a bookkeeping operation, these constraints are only taken into account implicitly.

Similarly, the interior and boundary observations (72) and (74) can be denoted in operator form as

$$\mathbf{y} = C\mathbf{x} + \mathbf{r} \tag{87}$$

$$\mathbf{y}_b = H\mathbf{x}_b + \mathbf{r}_b \tag{88a}$$

where

$$\mathbf{y}_b = \begin{bmatrix} \mathbf{y}_H \\ \mathbf{y}_V \end{bmatrix} \qquad \mathbf{r}_b = \begin{bmatrix} \mathbf{r}_H \\ \mathbf{r}_V \end{bmatrix} \tag{88b}$$

are obtained by scanning the horizontal and vertical boundary observations and noises, and the matrix H has a structure identical to that of V

$$H = \begin{bmatrix} \Theta_L & \Psi_L & \Theta_R & \Psi_R & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Theta_B & \Psi_B & \Theta_T & \Psi_T \end{bmatrix} \tag{88c}$$

with

$$\Theta_E = I \otimes H_E \qquad \Psi_E = I \otimes G_E \qquad \text{for } E = L, R, B, T. \tag{88d}$$

The covariance of the zero-mean Gaussian vector $\mathbf{r}_b$ is given by

$$R_b = E[r_b r_b^T] = \begin{bmatrix} I \otimes R_H & 0 \\ 0 & I \otimes R_V \end{bmatrix}. \tag{89}$$

Then, it was shown in [6], [7] that the smoother dynamics and boundary conditions could be expressed in operator form as

$$\begin{bmatrix} L & -BQB^* \\ C^*R^{-1}C & L^\dagger \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ C^*R^{-1}\mathbf{y} \end{bmatrix} \tag{90}$$

$$[V^*\Pi_b^{-1}V + H^*R_b^{-1}H \quad E] \begin{bmatrix} \hat{\mathbf{x}}_b \\ \hat{\lambda}_b \end{bmatrix} = H^*R_b^{-1}\mathbf{y}_b \tag{91}$$

where B^*, C^*, V^*, and H^* are the adjoint operators of B, C, V, and H, respectively, and where $L^\dagger$ denotes the formal adjoint of the difference operator L. $L^\dagger$ and the boundary operator E are defined through the Green's identity

$$\langle L\mathbf{x}, \lambda\rangle_{S(\tilde{\Omega})} = \langle \mathbf{x}, L^\dagger\lambda\rangle_{S(\tilde{\Omega})} + \langle \mathbf{x}_b, E\lambda_b\rangle_{S_b} \tag{92}$$

where $S(\tilde{\Omega})$ and S_b are the vector spaces of n-vector functions indexed over the domain $\tilde{\Omega}$, and over the first and last two rows and columns of Ω, respectively, and where $\langle ., . \rangle_S$ denotes the inner product over these spaces. The variable $\hat{\lambda}_{i,j}$ appearing in (90) is the conditional mean of $\lambda_{i,j}$ with respect to the space $\mathbf{Y}$ spanned by the observations, where $\lambda_{i,j}$ is the state of the complementary model associated with $x_{i,j}$. The concept of complementary model was originally introduced by Weinert and Desai [12], and it is the key element used in [6], [7] to derive the smoothing equations (90), (91). Note also that (90) has a Hamiltonian structure similar to that of the smoother for 1-D causal processes [37].

Again, because of the specific scanning scheme employed in the definition of $\hat{\mathbf{x}}_b$ and $\hat{\lambda}_b$, both of these vectors contain duplicate entries, which must be constrained to be the same. This is only an artifact of the mathematical derivation employed to obtain the smoother (91), and all boundary-value duplications can be removed in subsequent uses of the smoother.

B. NNM Characterization of the Smoother

As such, the operator characterization (90), (91) describes completely the NNM smoother. However, this characterization can be made more explicit by noting that for the Green's identity (92), it can be shown that

$$(L^\dagger\lambda)_{i,j} = \lambda_{i,j} - A_2^T\lambda_{i-1,j} - A_1^T\lambda_{i+1,j} - A_4^T\lambda_{i,j-1} - A_3^T\lambda_{i,j+1} \tag{93}$$

and

$$\begin{aligned}\langle \mathbf{x}_b, E\lambda_b\rangle_{S_b} &= \sum_{j=1}^{J-1}\left\{[x_{0,j}^T \quad x_{1,j}^T]E_L\begin{bmatrix}\lambda_{0,j}\\ \lambda_{1,j}\end{bmatrix} + [x_{I,j}^T \quad x_{I-1,j}^T]E_R\begin{bmatrix}\lambda_{I,j}\\ \lambda_{I-1,j}\end{bmatrix}\right\}\\ &\quad + \sum_{i=1}^{I-1}\left\{[x_{i,0}^T \quad x_{i,1}^T]E_B\begin{bmatrix}\lambda_{i,0}\\ \lambda_{i,1}\end{bmatrix} + [x_{i,J}^T \quad x_{i,J-1}^T]E_T\begin{bmatrix}\lambda_{i,J}\\ \lambda_{i,J-1}\end{bmatrix}\right\}\end{aligned} \tag{94}$$

with

$$E_L = \begin{bmatrix}0 & -A_1^T\\ A_2^T & 0\end{bmatrix} \qquad E_R = \begin{bmatrix}0 & -A_2^T\\ A_1^T & 0\end{bmatrix} \tag{95a}$$

$$E_B = \begin{bmatrix}0 & -A_3^T\\ A_4^T & 0\end{bmatrix} \qquad E_T = \begin{bmatrix}0 & -A_4^T\\ A_3^T & 0\end{bmatrix} \tag{95b}$$

Substituting (93) into the operator description (90) of the NNM smoother dynamics, we can rewrite these dynamics as

$$\begin{aligned}\alpha_0\begin{bmatrix}\hat{x}_{i,j}\\ \hat{\lambda}_{i,j}\end{bmatrix} &= \alpha_1\begin{bmatrix}\hat{x}_{i-1,j}\\ \hat{\lambda}_{i-1,j}\end{bmatrix} + \alpha_2\begin{bmatrix}\hat{x}_{i+1,j}\\ \hat{\lambda}_{i+1,j}\end{bmatrix} + \alpha_3\begin{bmatrix}\hat{x}_{i,j-1}\\ \hat{\lambda}_{i,j-1}\end{bmatrix}\\ &\quad + \alpha_4\begin{bmatrix}\hat{x}_{i,j+1}\\ \hat{\lambda}_{i,j+1}\end{bmatrix} + \beta y_{i,j}\end{aligned} \tag{96}$$

with

$$\alpha_0 = \begin{bmatrix}I & -BQB^T\\ C^TR^{-1}C & I\end{bmatrix} \qquad \beta = \begin{bmatrix}0\\ C^TR^{-1}\end{bmatrix} \tag{97a}$$

$$\alpha_1 = \begin{bmatrix}A_1 & 0\\ 0 & A_2^T\end{bmatrix} \qquad \alpha_2 = \begin{bmatrix}A_2 & 0\\ 0 & A_1^T\end{bmatrix} \tag{97b}$$

$$\alpha_3 = \begin{bmatrix}A_3 & 0\\ 0 & A_4^T\end{bmatrix} \qquad \alpha_4 = \begin{bmatrix}A_4 & 0\\ 0 & A_3^T\end{bmatrix} \tag{97c}$$

where (96) is almost in NNM form. This relation can be brought to NNM form by noting that α_0 is invertible with

$$\alpha_0^{-1} = \begin{bmatrix}I & BQB^T\\ -C^TR^{-1}C & I\end{bmatrix}\begin{bmatrix}P_1 & 0\\ 0 & P_2\end{bmatrix}, \tag{98a}$$

where

$$\begin{aligned}P_1 &= (I + BQB^TC^TR^{-1}C)^{-1}\\ P_2 &= (I + C^TR^{-1}CBQB^T)^{-1}\end{aligned} \tag{98b}$$

This yields

$$\begin{aligned}\begin{bmatrix}\hat{x}_{i,j}\\ \hat{\lambda}_{i,j}\end{bmatrix} &= \hat{\alpha}_1\begin{bmatrix}\hat{x}_{i-1,j}\\ \hat{\lambda}_{i-1,j}\end{bmatrix} + \hat{\alpha}_2\begin{bmatrix}\hat{x}_{i+1,j}\\ \hat{\lambda}_{i+1,j}\end{bmatrix} + \hat{\alpha}_3\begin{bmatrix}\hat{x}_{i,j-1}\\ \hat{\lambda}_{i,j-1}\end{bmatrix}\\ &\quad + \hat{\alpha}_4\begin{bmatrix}\hat{x}_{i,j+1}\\ \hat{\lambda}_{i,j+1}\end{bmatrix} + \tilde{\beta}y_{i,j}\end{aligned} \tag{99a}$$

with

$$\tilde{\alpha}_k = \alpha_0^{-1}\alpha_k \quad 1 \le k \le 4 \qquad \tilde{\beta} = \alpha_0^{-1}\beta \tag{99b}$$

which is now in NNM form.

Similarly, by using (94), (95) and taking into account the structure (84b), (88c) of boundary matrices V and H, the boundary conditions (91) for the NNM smoother can be rewritten more explicitly as

$$\begin{aligned}&\left\{\begin{bmatrix}V_L^T\\ W_L^T\\ V_R^T\\ W_R^T\end{bmatrix}\Pi_H^{-1}[V_L \quad W_L \quad V_R \quad W_R]\right.\\ &\qquad \left. + \begin{bmatrix}H_L^T\\ G_L^T\\ H_R^T\\ G_R^T\end{bmatrix}R_H^{-1}[H_L \quad G_L \quad H_R \quad G_R]\right\}\begin{bmatrix}\hat{x}_{0,j}\\ \hat{x}_{1,j}\\ \hat{x}_{I,j}\\ \hat{x}_{I-1,j}\end{bmatrix}\\ &\qquad + \begin{bmatrix}E_L & 0\\ 0 & E_R\end{bmatrix}\begin{bmatrix}\hat{\lambda}_{0,j}\\ \hat{\lambda}_{1,j}\\ \hat{\lambda}_{I,j}\\ \hat{\lambda}_{I-1,j}\end{bmatrix} = \begin{bmatrix}H_L^T\\ G_L^T\\ H_R^T\\ G_R^T\end{bmatrix}R_H^{-1}y_{H,j}\end{aligned} \tag{100a}$$

$$\left\{\begin{bmatrix} V_B^T \\ W_B^T \\ V_T^T \\ W_T^T \end{bmatrix} \Pi_V^{-1} [V_B \quad W_B \quad V_T \quad W_T] + \begin{bmatrix} H_B^T \\ G_B^T \\ H_T^T \\ G_T^T \end{bmatrix} R_V^{-1} [H_B \quad G_B \quad H_T \quad G_T] \right\} \begin{bmatrix} \hat{x}_{i,0} \\ \hat{x}_{i,1} \\ \hat{x}_{i,J} \\ \hat{x}_{i,J-1} \end{bmatrix}$$

$$+ \begin{bmatrix} E_B & 0 \\ 0 & E_T \end{bmatrix} \begin{bmatrix} \hat{\lambda}_{i,0} \\ \hat{\lambda}_{i,1} \\ \hat{\lambda}_{i,J} \\ \hat{\lambda}_{1,J-1} \end{bmatrix} = \begin{bmatrix} H_B^T \\ G_B^T \\ H_T^T \\ G_T^T \end{bmatrix} R_V^{-1} y_{V,i}. \tag{100b}$$

But these boundary conditions are precisely in the form (13)! Thus, the NNM solution techniques developed in Sections III and IV are directly applicable to the NNM smoother (99), (100), because the smoother itself is in NNM form. The fact that the class of NNM models is invariant under the smoothing operation is also quite satisfying, as it indicates that these models are perfectly adapted to the study of noncausal estimation problems.

C. Smoothing Error Dynamics

It was also shown in [6], [7] that the smoothing error $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ admits the operator characterization

$$\begin{bmatrix} L & -BQB^* \\ C^*R^{-1}C & L^\dagger \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ -\hat{\lambda} \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & C^*R^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{r} \end{bmatrix} \tag{101}$$

with boundary condition

$$[V^*\Pi_b^{-1}V + H^*R_b^{-1}H \quad E] \begin{bmatrix} \tilde{\mathbf{x}}_b \\ -\hat{\lambda}_b \end{bmatrix} = V^*\Pi_b^{-1}\mathbf{d}_b - H^*R_b^{-1}\mathbf{r}_b. \tag{102}$$

The 2-D NNM that corresponds to the operator expression (101) is identical to (99a), except for the input term

$$\begin{bmatrix} \tilde{x}_{i,j} \\ -\hat{\lambda}_{i,j} \end{bmatrix} = \tilde{\alpha}_1 \begin{bmatrix} \tilde{x}_{i-1,j} \\ -\hat{\lambda}_{i-1,j} \end{bmatrix} + \tilde{\alpha}_2 \begin{bmatrix} \tilde{x}_{i+1,j} \\ -\hat{\lambda}_{i+1,j} \end{bmatrix} \cdot \tilde{\alpha}_3 \begin{bmatrix} \tilde{x}_{i,j-1} \\ -\hat{\lambda}_{i,j-1} \end{bmatrix} + \tilde{\alpha}_4 \begin{bmatrix} \tilde{x}_{i,j+1} \\ -\hat{\lambda}_{i,j+1} \end{bmatrix} + \alpha_0^{-1} \begin{bmatrix} B & 0 \\ 0 & C^TR^{-1} \end{bmatrix} \begin{bmatrix} u_{i,j} \\ r_{i,j} \end{bmatrix}. \tag{103}$$

Similarly, the operator representation (102) of the boundary conditions yields boundary conditions identical to (100a, b), but with different right-hand sides.

The model (103) for the smoothing error can be used to compute the error covariance $P(i, j; k, l) = E[\tilde{x}_{i,j}\tilde{x}_{k,l}^T]$, which is a useful quantity if we want to evaluate the performance of the NNM smoother.

VI. Smoothing Examples

In this section, the results of the previous sections are applied to implement the NNM smoother for two examples, corresponding to the discretized stochastic Poisson and heat equations, respectively. In particular, it is shown that the FFT solver developed in Section IV can be used to implement the NNM smoother for both of these examples.

A. 2-D Poisson Equation

The dynamics of the process to be estimated are given by

$$x_{i,j} = \tfrac{1}{4}(x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}) + u_{i,j} \tag{104}$$

where the variance of the white Gaussian noise process $u_{i,j}$ is q. The boundary conditions are in Dirichlet form

$$\tilde{V}_E = 1 \qquad \tilde{W}_E = 0 \qquad \text{for } E = L, R, B, T \tag{105}$$

in (16), where the variance of the zero-mean boundary vectors $d_{E,k}$ is π_E. The interior observations are simply the process itself plus some additive white Gaussian noise process $r_{i,j}$ of unit variance

$$y_{i,j} = x_{i,j} + r_{i,j} \qquad (i, j) \in \tilde{\Omega} \tag{106}$$

and we assume that the state x is observed exactly on the boundary

$$y_{L,j} = x_{0,j} \qquad y_{R,j} = x_{I,j} \qquad y_{B,i} = x_{i,0} \qquad y_{T,i} = x_{i,J}. \tag{107}$$

Therefore, for this problem the matrices A_k with $1 \le k \le 4$, B, C, Q, and R are all scalars, and, in particular

$$A_1 = A_2 = A_3 = A_4 = 1/4 \tag{108a}$$

$$B = C = R = 1 \qquad Q = q. \tag{108b}$$

Substituting these values inside expression (96) for the NNM smoother, we find

$$\begin{bmatrix} 1 & -q \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{i,j} \\ \hat{\lambda}_{i,j} \end{bmatrix} = \frac{1}{4} \left\{ \begin{bmatrix} \hat{x}_{i-1,j} \\ \hat{\lambda}_{i-1,j} \end{bmatrix} + \begin{bmatrix} \hat{x}_{i+1,j} \\ \hat{\lambda}_{i+1,j} \end{bmatrix} + \begin{bmatrix} \hat{x}_{i,j-1} \\ \hat{\lambda}_{i,j-1} \end{bmatrix} + \begin{bmatrix} \hat{x}_{i,j+1} \\ \hat{\lambda}_{i,j+1} \end{bmatrix} \right\} + \begin{bmatrix} 0 \\ y_{i,j} \end{bmatrix}. \tag{109}$$

Taking also into account the form of the boundary conditions and observations (107) inside (100), it is easy to check that the NNM smoother boundary conditions are of Dirichlet type

$$\hat{x}_{0,j} = y_{L,j} \qquad \hat{x}_{I,j} = y_{R,j} \qquad \hat{x}_{i,0} = y_{B,i} \qquad \hat{x}_{i,J} = y_{T,i} \tag{110a}$$

$$\hat{\lambda}_{0,j} = 0 \qquad \hat{\lambda}_{I,j} = 0 \qquad \hat{\lambda}_{i,0} = 0 \qquad \hat{\lambda}_{i,J} = 0. \tag{110b}$$

As the NNM smoother dynamics are vertically symmetric and the boundary conditions are in Dirichlet form, the FFT solver described in Section IV-B can be used to solve (109), (110). Let $\{\xi_{i,j}\}$, $\{\mu_{i,j}\}$, and $\{\eta_{i,j}\}$ be the sequences obtained by applying the discrete sine transform S given by (62) to the estimates $\{\hat{x}_{i,j}\}$, $\{\hat{\lambda}_{i,j}\}$, and observations $\{y_{i,j}\}$ for a fixed index i. That is,

$$\xi_{i,j} = \left(\frac{2}{J}\right)^{1/2} \sum_{l=1}^{J-1} \hat{x}_{i,k} \sin(kj\pi/J) \qquad 1 \le j \le J-1 \tag{111a}$$

$$\mu_{i,j} = \left(\frac{2}{J}\right)^{1/2} \sum_{l=1}^{J-1} \hat{\lambda}_{i,k} \sin(kj\pi/J) \qquad 1 \le j \le J-1 \tag{111b}$$

$$\eta_{i,j} = \left(\frac{2}{J}\right)^{1/2} \sum_{l=1}^{J-1} y_{i,k} \sin(kj\pi/J) \qquad 1 \le j \le J-1. \tag{111c}$$

Let also

$$\epsilon_{i,j} = \left(\frac{2}{J}\right)^{1/2} \sin(j\pi/J)(y_{B,i} + (-1)^{j-1} y_{T,i}) \tag{111d}$$

be the sequence representing the effect of the DST on the boundary conditions (110c) and (110d) on the bottom and top edges. Then, by applying the DST to the columns of the NNM smoother (109), (110), we obtain the decoupled subsystems

$$\begin{bmatrix} 1 - \frac{1}{2}\cos(j\pi/J) & -q \\ 1 & 1 - \frac{1}{2}\cos(j\pi/J) \end{bmatrix} \begin{bmatrix} \xi_{i,j} \\ \mu_{i,j} \end{bmatrix} = \frac{1}{4}\left\{ \begin{bmatrix} \xi_{i-1,j} \\ \mu_{i-1,j} \end{bmatrix} + \begin{bmatrix} \xi_{i+1,j} \\ \mu_{i+1,j} \end{bmatrix} \right\} + \begin{bmatrix} \epsilon_{i,j} \\ \eta_{i,j} \end{bmatrix} \tag{112}$$

where $1 \le j \le J-1$, with boundary conditions

$$\xi_{0,j} = \eta_{L,j} \qquad \xi_{I,j} = \eta_{R,j} \qquad \mu_{0,j} = \mu_{I,j} = 0 \tag{113}$$

where $\{\eta_{L,j}\}$ and $\{\eta_{R,j}\}$ denote the DST transforms of the boundary measurements $\{y_{L,j}\}$ and $\{y_{R,j}\}$, respectively. These subsystems can then be written in TPBVDS form and solved by decomposing the TPBVDS model into forward and backward stable components. By observing that the modes σ of the system (112) are the zeros of the determinant of the matrix

$$\Phi(w) = \begin{bmatrix} 1 - \frac{1}{4}(w + 2\cos(j\pi/J)) & -q \\ 1 & 1 - \frac{1}{4}(w + 2\cos(j\pi/J)) \end{bmatrix} \tag{114}$$

where $w = \sigma + \sigma^{-1}$, it is clear that if σ is a mode, so is σ^{-1}, so that in the TPBVDS decomposition, there will be two forward stable and two backward stable modes. Unfortunately, even for this simple example, the TPBVDS decomposition cannot be computed in closed form.

B. Discretized Heat Equation

Consider now the discrete heat equation

$$mx_{i,j} = x_{i-1,j} + n(x_{i,j-1} + x_{i,j+1}) + u_{i,j} \tag{115}$$

where the variance of noise $u_{i,j}$ is q. Assume also that the boundary conditions, interior observations, and boundary observations are the same as for the previous example. Then, the NNM smoother takes the form

$$\begin{bmatrix} m & -q \\ 1 & m \end{bmatrix} \begin{bmatrix} \hat{x}_{i,j} \\ \hat{\lambda}_{i,j} \end{bmatrix} = \begin{bmatrix} \hat{x}_{i-1,j} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \hat{\lambda}_{i-1,j} \end{bmatrix} + n\left\{ \begin{bmatrix} \hat{x}_{i,j-1} \\ \hat{\lambda}_{i,j-1} \end{bmatrix} + \begin{bmatrix} \hat{x}_{i,j+1} \\ \hat{\lambda}_{i,j+1} \end{bmatrix} \right\} + \begin{bmatrix} 0 \\ y_{i,j} \end{bmatrix} \tag{116}$$

and the boundary conditions are given by (110a) and

$$\hat{\lambda}_{I,j} = \hat{\lambda}_{i,0} = \hat{\lambda}_{i,J} = 0 \tag{117}$$

with $\hat{\lambda}_{0,j}$ free. This last feature just corresponds to the fact that the $\hat{\lambda}$ dynamics are anticausal in the i direction, so that the values of $\hat{\lambda}_{i,j}$ with $i \ge 1$ are not affected by $\hat{\lambda}_{0,j}$. Again, the NNM smoother dynamics (116) are vertically symmetric, and the vertical boundary conditions are in Dirichlet form, so that the FFT solver of Section IV-B is applicable to this system. Performing the transformations (111a)–(111d), the NNM smoother is decoupled into $J-1$ subsystems of the form

$$\begin{bmatrix} m - 2n\cos(j\pi/J) & -q \\ 1 & m - 2n\cos(j\pi/J) \end{bmatrix} \begin{bmatrix} \xi_{i,j} \\ \mu_{i,j} \end{bmatrix} = \begin{bmatrix} \xi_{i-1,j} \\ \mu_{i+1,j} \end{bmatrix} + \begin{bmatrix} \epsilon_{i,j} \\ \eta_{i,j} \end{bmatrix} \tag{118}$$

with $1 \le j \le J-1$. But equation (118) is equivalent to the TPBVDS system

$$\begin{bmatrix} m - 2n\cos(j\pi/J) & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \xi_{i,j} \\ \mu_{i+1,j} \end{bmatrix} = \begin{bmatrix} 1 & q \\ 0 & m - 2n\cos(j\pi/J) \end{bmatrix} \begin{bmatrix} \xi_{i-1,j} \\ \mu_{i,j} \end{bmatrix} + \begin{bmatrix} \epsilon_{i,j} \\ -\eta_{i,j} \end{bmatrix} \tag{119}$$

where the boundary conditions are given by

$$\xi_{0,j} = \eta_{L,j} \qquad \mu_{I,j} = 0. \tag{120}$$

Thus, in this particular case, no state augmentation is necessary to bring the transformed smoother to TPBVDS form, because the heat equation is causal in the i direction. Thus, if we apply the DST transform to vertical index j in equation (115), the coupling with respect to the j variable is eliminated, and we obtain a standard causal 1-D system, for which the smoother is the standard 1-D smoother, given here by (119). This implies that the usual Riccati equations for the forward and backward filtered and predicted error variances can be used to decouple the dynamics (119) (see [8], and [6, section 5.3.2] for a description of the decoupling transformation), yielding the standard Mayne-Fraser [33], [34] two-filter implementation of the 1-D smoother.

VII. Conclusions

A general smoothing method has been obtained for 2-D random fields described by 2-D NNMs with local boundary conditions. This smoothing procedure relies on a general approach to the formulation of noncausal estimation problems developed in [6], [7]. In this approach, both the state of the system and of its complementary model need to be estimated, and accordingly, the smoother is described by a Hamiltonian system of twice the dimension of the original system. For the NNM case, it turns out that the Hamiltonian is itself in NNM form, with local boundary conditions of the type to specify the class of NNM systems. This property indicates that NNMs capture well the intrinsic noncausality associated with estimation problems in several dimensions.

One of the main themes of this paper is that straightfoward attempts at extending 1-D Kalman filtering techniques to several dimensions are misguided, because random fields in several dimensions are usually not generated causally, and multidimensional random observations are often not obtained sequentially, but all at one time. This

implies that noncausal random field models, such as NNMs, and smoothing problems provide the most natural ways to formulate multidimensional estimation problems. In other words, a *purely noncausal* formulation of multidimensional estimation problems should be employed. However, it is still possible to reintroduce recursiveness at the algorithmic level in order to obtain fast estimation techniques. As causality is in this case a computational device, many different types of recursions are possible, reflecting the great amount of latitude we have in processing the available data.

An important limitation of the results presented here is that we have assumed that the domain of definition of the 2-D NNMs under consideration was rectangular. For practical applications, random fields are usually defined over irregular domains, so that at first sight the results developed here have a limited applicability. However, this impression is incorrect, because recently developed *domain decomposition* techniques for PDEs [38] make it possible to divide an irregular domain in rectangular subdomains, and then to solve the original problem over each subdomain separately, while handling the coupling between subdomains with a preconditioned conjugate gradient algorithm. This approach would lead here to a parallel implementation of 2-D NNM estimation algorithms, were observations over different subdomains could be processed in parallel, and then combined to obtain an overall estimate. In addition to being parallel, this approach also makes it possible, provided that the conditions of Section IV are satisfied, to use FFT solvers over the rectangular subdomains, as shown in [39]. The application of domain decomposition techniques to NNM estimation problems seems therefore to be a promising area for future research.

References

[1] A. K. Jain, "Advances in mathematical models for image processing," *Proc. IEEE*, vol. 69, pp. 502–528, May 1981.

[2] A. K. Jain, "Partial differential equations and finite difference methods in image processing, Part I: Image representation," *J. Optimization Theory and Applic.*, vol. 23, pp. 65–91, Sept. 1977.

[3] A. K. Jain and J. R. Jain, "Partial differential equations and finite difference methods in image processing, Part II: Image restoration," *IEEE Trans. Automat. Control*, vol. AC-23, pp. 817–834, Oct. 1978.

[4] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[5] A. S. Willsky, *Digital Signal Processing and Control and Estimation Theory: Points of Tangency, Areas of Intersection, and Parallel Directions*. Cambridge, MA: MIT Press, 1979.

[6] M. B. Adams, "Linear estimation of boundary value stochastic processes," Sc.D. dissertation, Dept. of Aeron. and Astr., M.I.T., Cambridge, MA, Apr. 1983.

[7] M. B. Adams, A. S. Willsky, and B. C. Levy, "Linear estimation of boundary-value stochastic processes, Part I: The role and construction of complementary models," *IEEE Trans. Automat. Control*, vol. AC-29, pp. 803–811, Sept. 1984.

[8] M. B. Adams, A. S. Willsky, and B. C. Levy, "Linear estimation of boundary-value stochastic processes, Part 2: 1-D smoothing problems," *IEEE Trans. Automat. Control*, vol. AC-29, pp. 811–821, Sept. 1984.

[9] A. J. Krener, "Boundary value linear systems," *Astérisque*, vol. 75–76, pp. 149–165, Oct. 1980.

[10] A. J. Krener, "Acausal realization theory, Part 1: Linear deterministic systems," *SIAM J. Control and Optimization*, vol. 25, pp. 499–525, May 1987.

[11] A. J. Krener, "Reciprocal processes and the stochastic realization problem for acausal systems," in *Modeling, Identification and Control*, C. I. Byrnes and A. Lindquist, Eds. North-Holland, Amsterdam, The Netherlands, 1986, pp. 197–211.

[12] H. L. Weinert and U. B. Desai, "On complementary models and fixed-interval smoothing," *IEEE Trans. Automat. Control*, vol. AC-26, pp. 863–867, Aug. 1981.

[13] R. Nikoukhah, M. B. Adams, A. S. Willsky, and B. C. Levy, "Estimation for boundary-value descriptor systems," *Circuits, Systems, and Signal Processing*, vol. 8, pp. 25–48, 1989.

[14] L. R. Riddle, "Recursive estimation for distributed parameter systems," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, The Johns Hopkins Univ., Baltimore, MD, 1987.

[15] L. R. Riddle and H. L. Weinert, "Recursive linear smoothing for the 2-D Helmholtz equation," in *Proc. IFAC Symp. on Control of Distributed Parameter Systems*, 1986.

[16] L. R. Riddle and H. L. Weinert, "Recursive linear smoothing for dissipative hyperbolic systems," *Mechanical Systems and Signal Processing*, vol. 2, pp. 77–96, Jan. 1988.

[17] R. Nikoukhah, A. S. Willsky, and B. C. Levy, "Boundary-value descriptor systems, well-posedness, reachability and observability," *Int. J. Control*, vol. 46, pp. 1715–1737, Nov. 1987.

[18] R. Nikoukhah, A. S. Willsky, and B. C. Levy, "Reachability, observability and minimality for shift-invariant two-point boundary-value descriptor systems," *Circuits, Systems, and Signal Processing*, vol. 8, pp. 313–340, 1989.

[19] R. Nikoukhah, B. C. Levy, and A. S. Willsky, "Stability, stochastic stationarity, and generalized Lyapunov equations for two-point boundary-value descriptor systems," *IEEE Trans. Automat. Control*, vol. AC-34, pp. 1141–1152, Nov. 1989.

[20] R. Nikoukhah, "A deterministic and stochastic theory for two-point boundary-value descriptor systems," Ph.D. thesis, Dept. of Elec. Eng. and Comp. Science, and Report LIDS-TH-1820, Lab. for Information and Decision Systems, M.I.T., Cambridge, MA, Sept. 1988.

[21] R. W. Hockney, "A fast direct solution of Poisson's equation using Fourier analysis," *J. Assoc. Comput. Mach.*, vol. 12, pp. 95–113, 1965.

[22] A. K. Jain and E. Angel, "Image restoration, modelling, and reduction of dimensionality," *IEEE Trans. Computers*, vol. C-23, pp. 470–476, May 1974.

[23] G. D. Smith, *Numerical Solutions of Partial Differential Equations: Finite Difference Methods*, 3rd Ed. Oxford: Clarendon Press, 1985.

[24] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Systems*, vol. CAS-25, pp. 772–781, Sept. 1978.

[25] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore: The Johns Hopkins Univ. Press, 1983.

[26] D. G. Luenberger, "Time-invariant descriptor systems," *Automatica*, vol. 14, pp. 473–480, Sept. 1978.

[27] D. M. Young, *Iterative Solution of Large Linear Systems*. New York: Academic Press, 1971.

[28] S. McCormick, *Multigrid Methods*. Philadelphia: SIAM, 1987.

[29] R. E. Bank, "Marching algorithms and block Gaussian elimination," in *Sparse Matrix Computations*, J. R. Bunch and D. J. Rose, Eds. New York: Academic Press, 1976, pp. 293–307.

[30] R. E. Bank and D. J. Rose, "Marching algorithms for elliptic boundary value problems. I: The constant coefficient case," *SIAM J. Numerical Analysis*, vol. 14, pp. 792–829, Oct. 1977.

[31] F. R. Grantmacher, *The Theory of Matrices*. New York: Chelsea Publishing Co., New York, N.Y., 1960, vol. 2.

[32] B. Kågström and A. Ruhe, Eds., *Matrix Pencils*, Lecture Notes in Mathematics, Berlin: Springer Verlag, 1983, vol. 973.

[33] D. Q. Mayne, "A solution of the smoothing problem for linear dynamic systems," *Automatica*, vol. 4, pp. 73–92, Nov. 1966.

[34] D. C. Fraser, "A new technique for the optimal smoothing of data," D.Sc. dissertation, Dept. Aeronautics and Astronautics, M.I.T., Cambridge, MA, Jan. 1967.

[35] A. K. Jain, "A fast Karhunen-Loeve transform for digital restoration of images degraded by white and colored noise," *IEEE Trans. Computers*, vol. C-26, pp. 560–571, June 1977.

[36] K. C. Chou, "A multi-resolution approach to an inverse conductivity problem," M.S. thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA, Dec. 1987.

[37] G. C. Verghese, B. Friedlander, and T. Kailath, "Scattering theory and linear least-squares estimation, Part III: The esti-

mates," *IEEE Trans. Automat. Control*, vol. AC-25, pp. 794–802, Aug. 1980.
[38] R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, Eds., *First Int. Symp. on Domain Decomposition Methods for Partial Differential Equations*. Philadelphia: SIAM, 1988.
[39] T. F. Chan and D. C. Resasco, "A domain decomposed fast Poisson solver on a rectangle," *SIAM J. Scientific Stat. Computing*, vol. 8, pp. s14–s26, Jan. 1987.

Bernard C. Levy (Senior Member, IEEE) was born in Princeton, N.J., on July 31, 1951. He received the diploma of Ingénieur Civil des Mines from the Ecole Nationale Supérieure des Mines in Paris, France, and the Ph.D. in electrical engineering from Stanford University, Stanford, CA.

While at Stanford University, he held an INRIA Fellowship, and worked also as Teaching Assistant, Research Assistant, and Instructor. From June 1979 to June 1987, he was Assistant, and then Associate Professor in the Department of Electrical Engineering and Computer Science at MIT. Since July 1987, he has been Associate Professor in the Department of Electrical Engineering and Computer Science at the University of California, Davis. During the past three years, he has also been a consultant for the Charles Stark Draper Laboratory in Cambridge, MA. His research interests are in the areas of multidimensional and statistical signal processing, inverse problems, estimation, detection, and scientific computing.

Milton B. Adams Jr. (Member, IEEE) received the Sc.B. and Sc.M. degrees from Brown University, Providence, RI in 1971 and 1972, respectively and the Sc.D. Degree in Aeronautics and Astronautics from MIT in 1983.

He has been at the C.S. Draper Laboratory since September of 1972 and is currently the leader of the System Sciences Division of the Decision and Control Directorate. Over the last several years his research interests have focused on the application of heuristic search and learning to the development of algorithms for automated, autonomous mission, and trajectory planning, and on the design of hierarchical mission management systems that control autonomous real-time planning and plan execution.

Alan S. Willsky (Fellow, IEEE) received the S.B. degree and the Ph.D. degree from M.I.T. in 1969 and 1973, respectively.

From 1969 through 1973 he held a Fannie and John Hertz Foundation Fellowship. He joined the MIT faculty in 1973. From 1974 to 1981 he served as Assistant Director of the MIT Laboratory for Information and Decision Systems. His present position is Professor of Electrical Engineering. He is also a founder and member of the board of directors of Alphatech, Inc. He has held visiting positions at Imperial College, London, L'Université de Paris-Sud, and the Institut de Recherche en Informatique et Systèmes Aléatoires, Rennes, France. His present research interests are in problems involving multidimensional and multiresolution estimation and imaging, discrete-event systems, and the asymptotic analysis of control and estimation systems.

Dr. Willsky received the Donald P. Eckman Award from the American Automatic Control Council in 1975. He is Editor of the MIT Press series on signal processing, optimization, and control, was program chairman for the 17th IEEE Conference on Decision and Control, has been an associate editor for several journals, including the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, has served as a member of the Board of Governors and Vice President for Technical Affairs of the IEEE Control Systems Society, and was program chairman for the 1981 Bilateral Seminar on Control Systems held in the People's Republic of China. He gave the opening plenary lecture at the 20th IEEE Conference on Decision and Control, and was made a Distinguished Member of the IEEE Control Systems Society. He is the author of the research monograph *Digital Signal Processing and Control and Estimation Theory* and is co-author of the undergraduate text *Signals and Systems*. He was awarded the 1979 Alfred Noble Prize by the ASCE and the 1980 Browder J. Thompson Memorial Prize by the IEEE for a paper excerpted from his monograph.