Mapping and Pseudoinverse Algorithms for Ocean Data Assimilation

Paul W. Fieguth, *Member, IEEE*, Dimitris Menemenlis, and Ichiro Fukumori, *Member, IEEE*

Abstract—**Among existing ocean data assimilation methodologies, reduced-state Kalman filters are a widely studied compromise between resolution, optimality, error specification, and computational feasibility. In such reduced-state filters, the measurement update takes place on a coarser grid than that of the general circulation model (GCM); therefore, these filters require mapping operators from the GCM grid to the reduced state and vice versa. The general requirements are that the state-reduction and interpolation operators be pseudoinverses of each other, that the coarse state define a closed dynamical system, that the mapping operations be insensitive to noise, and that they be appropriate for regions with irregular coastlines and bathymetry. In this paper, we describe three efficient algorithms for computing the pseudoinverse: a fast Fourier transform algorithm that serves for illustration purposes, an exact implicit method that is recommended for most applications, and an efficient iterative algorithm that can be used for the largest problems. The mapping performance of 11 interpolation kernels is evaluated. Surprisingly, common kernels such as bilinear, exponential, Gaussian, and sinc perform only moderately well. We recommend instead three kernels, smooth, thin-plate, and optimal interpolation, which have superior properties. This study removes the computational bottleneck of mapping and pseudoinverse algorithms and makes possible the application of reduced-state filters to global problems at state-of-the-art resolutions.**

Index Terms—**Climatological models, data assimilation, interscale transforms, pseudoinverse methods, remote sensing, sparse pseudoinverses.**

I. Introduction

DRIVEN BY A desire to understand and predict the general circulation of the oceans and its interaction with anthropogenic forcing and climate, recent years have witnessed a proliferation of ocean data assimilation studies and methodologies [1]. Advancements in computing, modeling, and measurement capabilities make possible increasingly accurate descriptions of the time-varying ocean circulation. Nevertheless, all assimilation methodologies, e.g., adjoint method, feature models, Kalman filters, nudging, and optimal interpolation, remain a compromise between what is required, an optimal estimate with full error description at the highest possible resolution, and computational feasibility.

One such compromise is a class of methodologies known as reduced-state Kalman filters and associated optimal smoothers. These methodologies approximate general circulation model (GCM) errors with fewer degrees of freedom than those of the ocean model. Therefore, these methods require algorithms that project the GCM state onto a reduced state and vice versa. Application examples may be found in [2]–[12]. A key advantage of reduced-state filters is that they provide complete error descriptions for the circulation estimates, albeit on a coarser grid than that of the GCM. Another advantage is their computational efficiency. Were it not for mapping requirements, which can demand large computational and storage resources, a given reduced-state approximation could be applied to arbitrarily high-resolution, global GCMs with minimal computational overhead beyond that required to derive the filter. The present and related work [13] aim to make reduced-state filters feasible at state-of-the-art resolutions.

One computational bottleneck results from a number of requirements relating the fine and coarse fields. In particular, if the filter is to be stable over many time steps, it is normally required that the state reduction and interpolation operators be pseudoinverses of each other [14], which ensures that repeated subsampling and interpolation do not lead to a degradation of the coarse-scale data. It is also required that the coarse state define a closed system, i.e., a perturbation within the coarse space must remain in the coarse state following dynamic evolution. Finally, the mapping operations must be insensitive to noise and be appropriate for nonstationary problems, such as regions having irregular coastlines and bottom bathymetry.

Existing mapping and pseudoinverse schemes that satisfy these requirements often involve the brute-force computation and manipulation of a $n_f \times n_c$ matrix, where n_f and n_c are the fine-grid dimension of the ocean model and the coarse-grid dimension of the reduced state, respectively. Here, grid dimensions may refer to the three-dimensional (3-D) oceanic state vector [4] or, more commonly, to the horizontal grid dimension [2], [6], [11]. To illustrate the magnitude of the computational challenge in computing pseudoinverse matrices by brute force, consider the following hypothetical computation. Suppose the horizontal fine grid is that of a global-ocean GCM with 1/12°-spacing, such that $n_f \simeq 10^7$, and that the coarse grid has approximate grid spacing of 2° in latitude and longitude, i.e., $n_c \simeq 10^4$. The mapping and pseudoinverse operations, stored as a dense matrix, would require the storage and manipulation of a 1-TB matrix, which seems completely disproportionate to the problem at hand.

Clearly what is required is a sparse, implicit, or iterative representation of the pseudoinverse, rather than attempting

Manuscript received May 17, 2002; revised August 25, 2002. This work was supported by the National Ocean Partnership Program and by the Canadian Natural Science and Engineering Research Council. This work is a contribution of the Consortium for Estimating the Circulation and Climate of the Ocean.

P. Fieguth is with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, N2L 3G1 Canada (e-mail: pfieguth@uwaterloo.ca).

D. Menemenlis and I. Fukumori are with the Ocean Science Research Element, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: menemenlis@jpl.nasa.gov; fukumori@jpl.nasa.gov).

Digital Object Identifier 10.1109/TGRS.2002.808058

to compute and store it in dense form. Herein we discuss general requirements for mapping and decimation operators, propose a number of efficient algorithms, and evaluate suitable interpolation kernels for these mapping operators. Section II introduces notation and provides a mathematical description of the mapping problem. Section III describes the three efficient algorithms for computing the pseudoinverse: a fast Fourier transform (FFT) algorithm that serves to illustrate general properties of mapping and pseudoinverse operators, an exact implicit method that is reasonable in most applications, and an efficient iterative algorithm for the largest problems. Various interpolation kernels are tested for suitability to the reduced-state Kalman filter problem in Section IV. Discussion and recommendations follow in Section V.

II. Mathematical Description

Let $\boldsymbol{x}_f(t)$ represent the state vector of an ocean GCM, e.g., temperature, salinity, and velocity on a 3-D grid, at some time t. Algebraically, the GCM can be written as a rule for stepping this state vector forward in time

$$\boldsymbol{x}_f(t+1) = \mathcal{M}[\boldsymbol{x}_f(t),\, \boldsymbol{q}(t)] \tag{1}$$

where $\mathcal{M}$ encodes the system dynamics, and vector $\boldsymbol{q}(t)$ represents the boundary conditions and other model parameters specified at time t. For modern ocean GCMs, the state vector $\boldsymbol{x}_f(t)$ can have a dimension of 10^8 or more. Thus, even for linear systems, the dimensionality of the problem makes a full implementation of the Kalman filter and optimal smoother completely infeasible. To reduce the filter and smoother complexity, model errors are represented on a coarser grid, $\boldsymbol{x}_c$, hence the need for mapping operators between $\boldsymbol{x}_f$ and $\boldsymbol{x}_c$.

A. Pseudoinverse

The fine-to-coarse transformation is

$$\boldsymbol{x}_c = \boldsymbol{B}^* \boldsymbol{x}_f \tag{2}$$

where matrix $\boldsymbol{B}^*$ represents a state reduction operator, intuitively a filter that, for example, attenuates mesoscale-eddy variability but retains planetary-wave and other large-scale circulation features. A pseudoinverse operator $\boldsymbol{B}$ is defined such that

$$\boldsymbol{x}_c = \boldsymbol{B}^* \boldsymbol{B} \boldsymbol{x}_c \tag{3}$$

holds for all vectors $\boldsymbol{x}_c$. $\boldsymbol{B}$ is an interpolation operator that maps the reduced-state vector back to the original grid so that

$$\boldsymbol{x}_f = \boldsymbol{B}\boldsymbol{x}_c + \boldsymbol{\varepsilon} \quad \boldsymbol{B}^*\boldsymbol{\varepsilon} = \boldsymbol{0} \tag{4}$$

where $\boldsymbol{\varepsilon}$ represents the high-frequency/wavenumber components that lie in the null space of transformation $\boldsymbol{B}^*$. Given some selected $\boldsymbol{B}^*$, one possible solution for $\boldsymbol{B}$, satisfying (3), is the Moore–Penrose pseudoinverse [14]

$$\boldsymbol{B} = \boldsymbol{B}^{*T}\left(\boldsymbol{B}^*\boldsymbol{B}^{*T}\right)^{-1}. \tag{5}$$

$\boldsymbol{B}^*\boldsymbol{B}^{*T}$ will be invertible if $\boldsymbol{B}^*$ has full row-rank, implying that the elements in $\boldsymbol{x}_c$ are computed as linearly independent interpolants of $\boldsymbol{x}_f$. This will be the case for all but the most unfortunate choice of $\boldsymbol{B}^*$, since the number of columns in $\boldsymbol{B}^*$ is much greater than the number of rows. A more common approach, in our experience, is to explicitly chose the interpolation operator $\boldsymbol{B}$, in which case $\boldsymbol{B}^*$ can then be computed as the Moore–Penrose pseudoinverse using

$$\boldsymbol{B}^* = \left(\boldsymbol{B}^T\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T. \tag{6}$$

Again, the requirement that $\boldsymbol{B}^T\boldsymbol{B}$ be invertible is normally satisfied. The challenge lies in the computation and representation (storage) of the gigantic matrices $\boldsymbol{B}$ and $\boldsymbol{B}^*$.

In practice, the mapping and pseudoinverse problem for the full GCM state vector is often broken down into a number of smaller horizontal mapping problems using, for example, vertical empirical orthogonal functions [5] or dynamical modes [9]. Symbolically, the 3-D interpolation operator can be written

$$\boldsymbol{B} = \boldsymbol{B}_v\,\boldsymbol{B}_h \tag{7}$$

where $\boldsymbol{B}_h$ and $\boldsymbol{B}_v$ represent horizontal and vertical interpolation operators, respectively. Conversely, the pseudoinverse operation is

$$\boldsymbol{B}^* = \boldsymbol{B}_h^*\,\boldsymbol{B}_v^* \tag{8}$$

with both $\boldsymbol{B}_h^*\,\boldsymbol{B}_h$ and $\boldsymbol{B}_v^*\,\boldsymbol{B}_v$ satisfying the pseudoinverse condition (3). Although the discussion and intuition of the remainder of the paper will apply primarily to the horizontal operators $\boldsymbol{B}_h$ and $\boldsymbol{B}_h^*$, the mapping algorithms that are developed apply to much broader contexts. For large problems, it is often convenient to represent $\boldsymbol{B}$ by specifying an interpolation kernel w, such that

$$x_f(j) = \sum_i x_c(i) w(i,\, j). \tag{9}$$

This permits computational and storage savings when the weights w are stationary, i.e., not a function of i, as further discussed in Section III-C.

B. Noise Sensitivity

A second desired attribute for the mapping operators is that the transformations be stable with respect to errors. Since the reduced-state filter is an approximation, the mapping and interpolation operations must be insensitive to inaccuracies of this approximation. Consider, for example, the degree to which a perturbation $\boldsymbol{\delta}$ at the fine scale affects the coarse-scale coefficients

$$\boldsymbol{x}_c \xrightarrow{\boldsymbol{B}} (\boldsymbol{x}_f + \boldsymbol{\delta}) \xrightarrow{\boldsymbol{B}^*} \overline{\boldsymbol{x}}_c. \tag{10}$$

If $\boldsymbol{B}^*$ were very close to singular, then a small disturbance $\boldsymbol{\delta}$ could give rise to an arbitrarily large difference $(\overline{\boldsymbol{x}}_c - \boldsymbol{x}_c)$. The intuitive noise sensitivity criterion

$$\frac{|\overline{\boldsymbol{x}}_c - \boldsymbol{x}_c|}{|\boldsymbol{\delta}|} \tag{11}$$

is not, however, sufficient because if $\boldsymbol{B}$ and $\boldsymbol{B}^*$ are a pseudoinverse pair, then so are $\gamma\boldsymbol{B}$ and $\boldsymbol{B}^*/\gamma$, in which case (11) can be made arbitrarily small by increasing γ. The properly normalized noise sensitivity criterion is

$$\frac{|\bar{\boldsymbol{x}}_c - \boldsymbol{x}_c|}{|\boldsymbol{\delta}|}\,\frac{|\boldsymbol{x}_f|}{|\boldsymbol{x}_c|} \equiv \frac{|\boldsymbol{B}^*\boldsymbol{\delta}|}{|\boldsymbol{\delta}|}\,\frac{|\boldsymbol{B}\boldsymbol{x}_c|}{|\boldsymbol{x}_c|}. \tag{12}$$

The upper bound for this sensitivity is given by the product of the largest singular values $\sigma_{\max}$ of $\boldsymbol{B}$ and $\boldsymbol{B}^*$; thus, the noise sensitivity is limited by

$$\sigma_{\max}(\boldsymbol{B}) * \sigma_{\max}(\boldsymbol{B}^*) = \mathrm{cond}(\boldsymbol{B}) = \mathrm{cond}(\boldsymbol{B}^*) \geq 1. \tag{13}$$

That is, the noise sensitivity is bounded by the condition number of the pseudoinverse $\boldsymbol{B}^*$, which is equal to the condition number of the interpolator $\boldsymbol{B}$, implying that the noise sensitivity can be evaluated before attempting to compute $\boldsymbol{B}^*$.

C. Shift Sensitivity

A third desired attribute is that the coarse state define a closed dynamical system. This implies that the energy of the large, resolved scales should not degrade or be corrupted over repeated time steps, i.e.,

$$\boldsymbol{B}^*\mathcal{M}[\boldsymbol{x}_f(t),\, \boldsymbol{q}(t)] \approx \boldsymbol{B}^*\mathcal{M}[\mathbf{BB}^*\boldsymbol{x}_f(t),\, \boldsymbol{q}(t)]. \tag{14}$$

A specific example is that the dynamics of $\mathcal{M}$ typically involve advection or propagation of waves on scales considerably smaller than the coarse-scale discretization interval. The mapping-interpolation operation $\boldsymbol{BB}^*$ should therefore be insensitive to spatial shifts, to ensure that a slow, advective flow is not progressively corrupted by repeated mapping-interpolations, i.e.,

$$\mathcal{S}\boldsymbol{BB}^* \approx \boldsymbol{BB}^*\mathcal{S} \tag{15}$$

where $\mathcal{S}$ represents a spatial translation on the fine scale. The condition that an operator be unaffected by fine shifts is effectively an antialiasing or bandlimiting criterion as further discussed in Section III-A and in Appendix I. Experimentally, we define the shift sensitivity as the root-mean-square (rms) ratio

$$\frac{\mathrm{rms}\,\{\mathcal{S}(\boldsymbol{B\delta}_i) - \boldsymbol{BB}^*[\mathcal{S}(\boldsymbol{B\delta}_i)]\}}{\mathrm{rms}\,\{\boldsymbol{B\delta}_i\}} \tag{16}$$

where $\boldsymbol{\delta}_i$ is a coarse unit-vector with pixel i set to one and the rest to zero. The noise and shift sensitivity criteria, described above, are used to evaluate various interpolation kernels in Section IV.

III. Computation of the Pseudoinverse

We next discuss three efficient algorithms for computing the pseudoinverse: an FFT algorithm (which serves to further illustrate noise and shift sensitivities), an exact implicit method (which is recommended for most applications), and an efficient iterative algorithm (which can be used for the largest problems).

A. FFT Algorithm

The straightforward computation of the pseudoinverse, using (5) or (6), is made difficult because even if one operation, such as the interpolation $\boldsymbol{B}$, is chosen to be simple and sparse, the computed pseudoinverse is normally complex and dense. A promising direction is to investigate known reversible operations from signal processing theory. Specifically, the condition of the pseudoinverse (3) can be satisfied using an FFT scheme [5]

$$\boldsymbol{x}_f = \mathcal{F}_2^{-1}[\mathcal{W}(k_x,\, k_y)\,\mathcal{F}_2\,(\uparrow \boldsymbol{x}_c)] \tag{17}$$

$$\boldsymbol{x}_c = \downarrow \mathcal{F}_2^{-1}[\mathcal{W}^*(k_x,\, k_y)\,\mathcal{F}_2\,(\boldsymbol{x}_f)] \tag{18}$$

where $\mathcal{F}_2$ represents the two-dimensional (2-D) Fourier transform, and the arrows $\uparrow$ and $\downarrow$ represent oversampling and subsampling, respectively. $\mathcal{W}$ is a filtering function, the Fourier transform of the kernel w in (9) if the grid is regular and periodic. $\mathcal{W}^*$ is the pseudoinverse filter of $\mathcal{W}$, i.e.,

$$\begin{aligned} \mathcal{W}^* &= \mathcal{W}^{-1}, & &\forall\, k_x,\, k_y \ni \mathcal{W} \neq 0 \\ &= 0, & &\text{otherwise.} \end{aligned} \tag{19}$$

Thus, the interpolation (17) is equivalent to oversampling the coarse state $\boldsymbol{x}_c$ and applying a lowpass filter $\mathcal{W}$. Conversely, the state reduction (18) is equivalent to lowpass filtering the fine state $\boldsymbol{x}_f$ and then subsampling at the coarse-grid locations, the lowpass filter being required to avoid aliasing. Note that in order for the pseudoinverse condition (3) to be satisfied exactly, the number of nonzero elements in $\mathcal{W}$ must be equal to or greater than the number of coarse-grid elements.

The FFT approach provides a simple intuition for the noise and shift sensitivity criteria described in Section II. First, if the filtering function $\mathcal{W}$ is constant for all wavenumbers where it is nonzero, then the noise sensitivity criterion (13) is satisfied with equality. Second, if the interpolation is perfectly band limited, i.e., $\mathcal{W} = 0$ for all scales not resolved by the coarse grid, then the shift-sensitivity criterion (16) is identically zero (see Appendix I for details). That is, noise sensitivity is minimized by choosing interpolation and state-reduction operators that have a flat wavenumber response, and shift sensitivity is minimized by choosing bandlimited operators. In general, however, with the exception of stationary periodic domains, flatness and bandlimitedness are incompatible goals, so the FFT case is the exception in achieving both of these criteria perfectly.

The FFT approach is elegant in that both interpolation and subsampling operations are represented implicitly, rather than as a dense matrix, and because the transformations are very fast. A major drawback, however, stems from the stationary property of the FFT: mapping is possible only on a regular, periodic grid, making this approach ill-suited to global studies with irregular boundaries (coastlines, etc.). Global studies [9] have, instead, relied on defining a suitable sparse interpolation operator, $\boldsymbol{B}$, and then computing the pseudoinverse $\boldsymbol{B}^*$ by brute force (6). Sections III-B and III-C discuss efficient alternatives for the computation of $\boldsymbol{B}^*$.

B. Implicit Algorithm

A first efficient alternative to (6) is based on noting that the specified operator (either $\boldsymbol{B}$ or $\boldsymbol{B}^*$) is usually chosen to be sparse, which suggests a more sensible approach than the brute-force operations of (5) or (6). For example, given a fixed interpolator $\boldsymbol{B}$, if

$$\boldsymbol{Q}^{-1} = \left(\boldsymbol{B}^T\boldsymbol{B}\right)^{-1} \tag{20}$$

is computed off-line, ahead of time, then

$$\boldsymbol{x}_c = \boldsymbol{Q}^{-1}\boldsymbol{B}^T\boldsymbol{x}_f = \boldsymbol{Q}^{-1}\overline{\boldsymbol{x}}_c \tag{21}$$

where the product $\boldsymbol{B}^T\boldsymbol{x}_f$ can be computed efficiently because of the sparsity of $\boldsymbol{B}^T$. The matrix $\boldsymbol{B}^*$ is never computed explicitly; rather, it is represented implicitly, with the following huge advantages.

1) *Computation.* The calculation of $\boldsymbol{Q}^{-1}$ is much simpler than the direct computation of $\boldsymbol{B}^*$.
2) *Storage.* Rather than storing $\boldsymbol{B}^*$ densely, we have an implicit representation of $\boldsymbol{B}^*$ in the form of a dense matrix $\boldsymbol{Q}^{-1}$ and a huge implicit operator $\boldsymbol{B}$. The required storage is a function of *only* the coarse scale; fine scales of *any* size can be accommodated.
3) *Accuracy.* $\boldsymbol{B}^*$ is based on *all* of the fine-scale elements, reducing aliasing problems.

Nevertheless, for large problems (e.g., $n_c = 10^4$) the storage of $\boldsymbol{Q}^{-1}$ requires about 1 GB, the inversion of $\boldsymbol{Q}$ requires ten hours on a modern workstation, and the product $\boldsymbol{Q}^{-1}\overline{\boldsymbol{x}}_c$ is still relatively expensive. Although this may be tolerable for a one-time initialization, it certainly precludes testing a variety of interpolators. The following section develops a novel, faster, iterative approach.

C. Iterative Algorithm

We continue to assume that the interpolant $\boldsymbol{B}$ has been specified, and that $\boldsymbol{B}^*$ needs to be inferred. However, we are not interested in $\boldsymbol{B}^*$ *per se*; our goal is the inference of $\boldsymbol{x}_c$ from $\boldsymbol{x}_f$. Inverting (21), we deduce the equivalent linear system

$$\boldsymbol{Q}\boldsymbol{x}_c = \overline{\boldsymbol{x}}_c. \tag{22}$$

In distinct contrast to the previous section, where a dense matrix $\boldsymbol{Q}^{-1}$ is computed and stored, here $\boldsymbol{Q} = \boldsymbol{B}^T\boldsymbol{B}$ is sparse if the interpolation operator $\boldsymbol{B}$ is sparse. The computation of $\boldsymbol{Q}$

$$\boldsymbol{Q}_{i,\,k} = \sum_j w(i,\,j)w(k,\,j) \tag{23}$$

is particularly rapid if the interpolation kernel, $w(i,\,j)$ in (9), is stationary (independent of i) for most i, e.g., for all but a relatively small number of coastline pixels. The linear system (22) can be solved using any of the usual iterative approaches, with Gauss–Jacobi, Gauss–Seidel, and conjugate–gradient being the most straightforward choices [15]. We recommend the conjugate gradient approach because of its simple implementation and excellent convergence characteristics. Table I presents experimental results showing the average number of conjugate–gradient iterations required to reduce the error in the pseudoinverse to 0.5%, averaged over a variety of interpolation kernels (see Section IV for details). All of the tests were carried out with nonperiodic, square domains of size specified in the first column of the table and with ten fine-scale pixels per coarse-scale pixel in each dimension. Variable τ is a scale parameter that controls the spatial size of the interpolator. Table I shows that the conjugate gradient iterative convergence depends strongly on the scale of the kernel, but is only weakly dependent on problem size.

TABLE I
AVERAGE NUMBER OF CONJUGATE-GRADIENT ITERATIONS TO ACHIEVE A rms ACCURACY OF 0.5%

Problem Size	Density of $\boldsymbol{Q}$	Interpolator Size τ						
(coarse scale		(fine-scale pixels)						
pixels)		2	3	5	8	12	17	28
33×33	0.09	4	6	11	41	174	303	240
29×29	0.12	3	6	11	43	165	291	245
25×25	0.15	3	6	11	41	169	283	233
21×21	0.21	3	6	11	40	158	290	223
17×17	0.30	3	6	11	41	155	238	195
13×13	0.45	3	6	11	38	115	232	168
9×9	0.73	4	6	11	27	117	172	115

TABLE II
COMPARISON OF STORAGE AND COMPUTATIONAL REQUIREMENTS

	Storage	Initialization	Effort Per
	$\boldsymbol{B}^*, \boldsymbol{Q}^{-1}, \boldsymbol{Q}$	Effort	Mapping
Brute	$n_c \cdot n_f$	$n_c^3 + \alpha^2 n_f^2$	$n_c \cdot n_f$
Force	100 GB	10^{13}	10^{10}
Implicit	n_c^2	n_c^3	$n_c^2 + \alpha^2 n_f$
Method	1 GB	10^{12}	10^8
Iterative	$\alpha^2 n_c$	$\alpha^3 n_f/\beta$	$\alpha^2 n_f + i\alpha^2 n_c$
Method	1 MB	10^7	2×10^7

The benefits of the iterative approach can be quite significant for large problems (see Table II). Consider 2-D fine-scale and coarse-scale grids with n_f and n_c pixels, respectively, and let $s = \sqrt{n_f/n_c}$ be the corresponding fine-to-coarse scale subsampling ratio. Finally, let α control the size of the region of support of interpolator w in (9): the region is square and has α coarse-scale or αs fine-scale pixels on a side. Then the initialization of the iterative approach requires the computation of $\boldsymbol{Q}$, which has at most $\alpha^2 n_c$ nonzero elements, as compared to dense matrices $\boldsymbol{Q}^{-1}(n_c \times n_c)$ and $\boldsymbol{B}^*(n_c \times n_f)$ for the implicit and direct approaches, respectively.

The computational effort in deriving $\boldsymbol{Q}$ scales with $\alpha^2 n_c(\alpha s)^2/\beta = \alpha^3 n_f/\beta$: the number of nonzero elements of $\boldsymbol{Q}$, multiplied by the effort per dot product, divided by redundancies in (23). In the worst case of a fully nonstationary problem, $\beta = 1$. However, away from irregularities (e.g., coastlines), the weights are normally stationary so that β typically increases with the size of the coarse state ($\beta \sim 2.5$ for the large 81×121 coarse grid of Fig. 5). In comparison, the computational effort for the inversion to $\boldsymbol{Q}^{-1}$ scales with n_c^3, and the computation of $\boldsymbol{B}^*$ requires on the order of $\alpha^2 n_c n_f$ additional operations.

The direct method requires $n_c n_f$ multiplications per fine-to-coarse mapping. The implicit and iterative methods are initialized by computing $\overline{\boldsymbol{x}}_c$, requiring on the order of $\alpha^2 n_f$ operations. The implicit method completes the mapping by solving (21), an effort of n_c^2. The iterative method solves (22) and requires an effort of $\alpha^2 n_c$ per conjugate gradient iteration.

Table II provides some perspective on the relative cost of the three methods. It lists values for a problem having $n_c = 10^4$ coarse-scale and $n_f = 10^6$ fine-scale elements, an interpolation footprint of $\alpha = 3$ coarse-grid elements, redundancy factor $\beta = 2$, and $i = 100$ conjugate gradient iterations. In

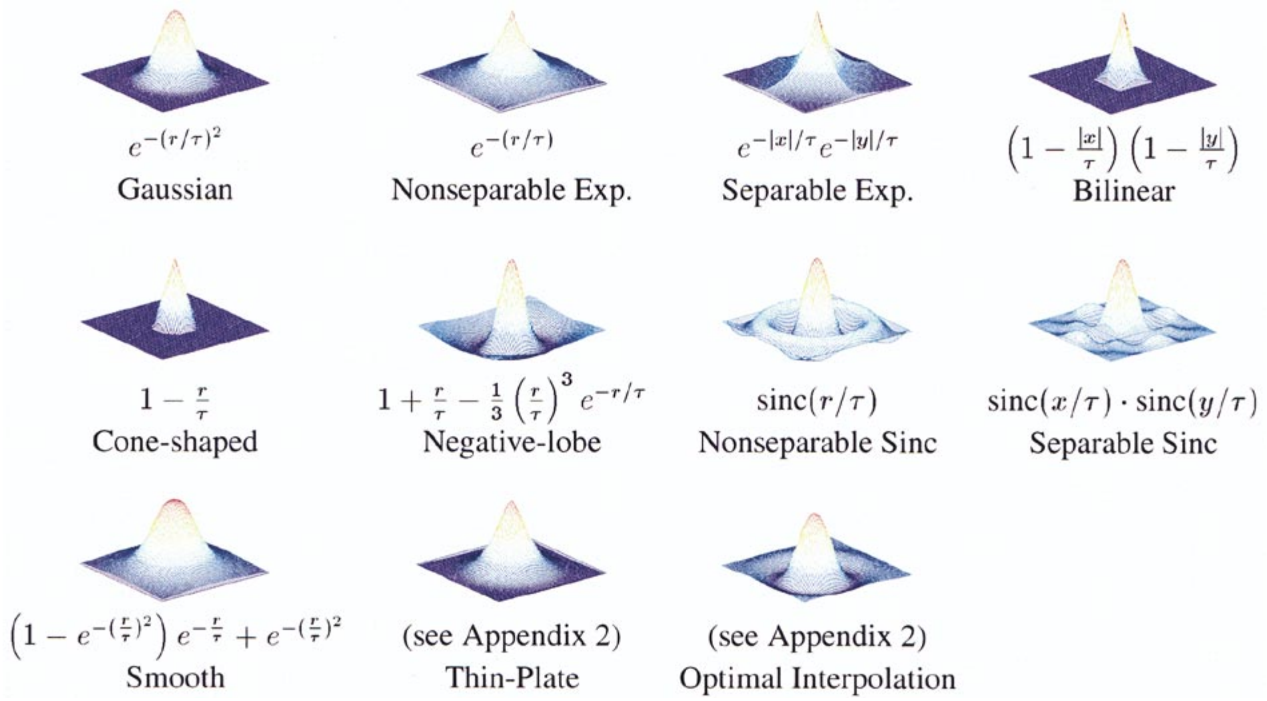

$e^{-(r/\tau)^2}$
Gaussian
$e^{-(r/\tau)}$
Nonseparable Exp.
$e^{-|x|/\tau}e^{-|y|/\tau}$
Separable Exp.
$\left(1-\frac{|x|}{\tau}\right)\left(1-\frac{|y|}{\tau}\right)$
Bilinear
$1-\frac{r}{\tau}$
Cone-shaped
$1+\frac{r}{\tau}-\frac{1}{3}\left(\frac{r}{\tau}\right)^3 e^{-r/\tau}$
Negative-lobe
$\text{sinc}(r/\tau)$
Nonseparable Sinc
$\text{sinc}(x/\tau)\cdot\text{sinc}(y/\tau)$
Separable Sinc
$\left(1-e^{-(\frac{r}{\tau})^2}\right)e^{-\frac{r}{\tau}}+e^{-(\frac{r}{\tau})^2}$
Smooth
(see Appendix 2)
Thin-Plate
(see Appendix 2)
Optimal Interpolation

Fig. 1. Interpolation kernels tested; $r = \sqrt{x^2 + y^2}$ measures the distance to the origin, and τ is a scale parameter that controls the spatial size of the interpolator.

this context, compared to the implicit approach, the new iterative method uses only 0.1% of the memory (a saving of 1 GB), initializes in a minuscule fraction of the time, and computes each pseudoinverse about five times faster. Compared to a direct, brute-force pseudoinverse calculation, the comparison is even more extreme.

IV. Interpolation Kernels

Having determined an extremely efficient algorithm for the pseudoinverse, it is now possible to test a large number of interpolation kernels and parameters in relation to the noise and shift sensitivity criteria of Section II. Fig. 1 displays the 11 interpolation kernels that were tested. The kernels include a variety of standard and heuristic kernels: Gaussian, nonseparable and separable exponential, bilinear, cone-shaped, negative-lobe, nonseparable and separable sinc, smooth, thin-plate, and optimal interpolation-based functions. Of these, the latter two are implicit in their formulation; details are provided in Appendix II. All tests were carried out in 20 × 20 coarse-scale, 200 × 200 fine-scale domains. The region of support for the interpolation kernel was varied from 20 × 20 to 100 × 100 fine-scale pixels (i.e., $\alpha = 2, \ldots, 10$); however, results are shown only for $\alpha = 8, 10$. The scale parameter τ was varied from 2–30 fine-scale pixels.

The two predominant criteria, noise sensitivity (conditioning) and shift sensitivity (aliasing), were tested both experimentally and theoretically. The experimental results were computed empirically using a Monte Carlo approach. Noise sensitivity was computed as the rms noise sensitivity to random fine-scale noise (12). Shift sensitivity was computed using a shift $\mathcal{S}$ of one-half of the coarse-scale discretization for a unit coarse-grid perturbation in the middle of the test domain (16). These results are plotted in Fig. 2.

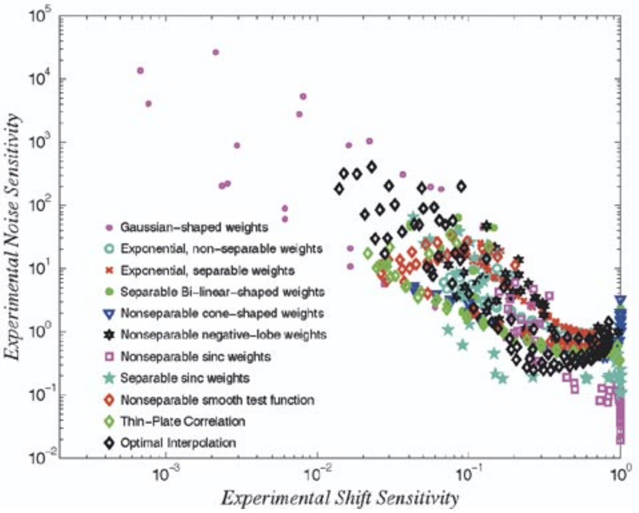


Fig. 2. Comparison of experimental shift sensitivity versus experimental noise sensitivity for the 11 interpolation kernels of Fig. 1. Better performance is down and to the left. Results are shown for $\alpha = 8, 10$.

For comparison purposes and to validate our understanding, the counterpart theoretical results are plotted in Fig. 3. The condition number (13), which represents an upper bound on the noise sensitivity, is plotted versus interpolation kernel aliasing (27). In both cases, the lower left corner of Fig. 3 represents better kernel performance. The fit between the aliasing criterion (27) and experimental shift sensitivity (16) is very close, with the differences primarily attributed to boundary effects. The fit between the condition number and experimental noise sensitivity is less close because the condition number establishes an upper bound rather than an expected value. However, although the theoretical and experimental quantities measure slightly different things, the overall patterns and conclusions from the two

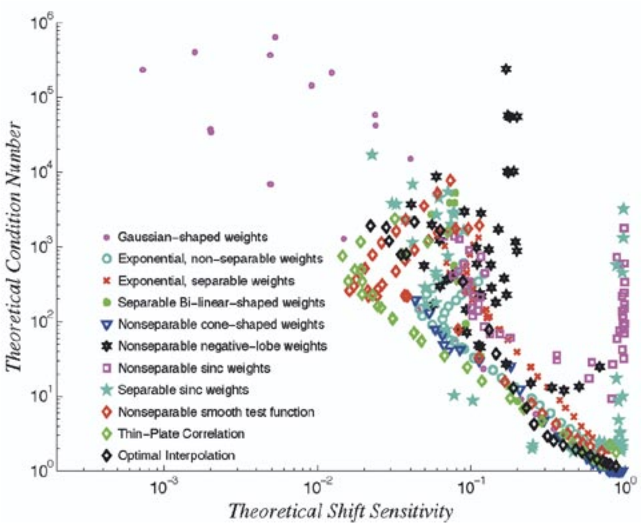


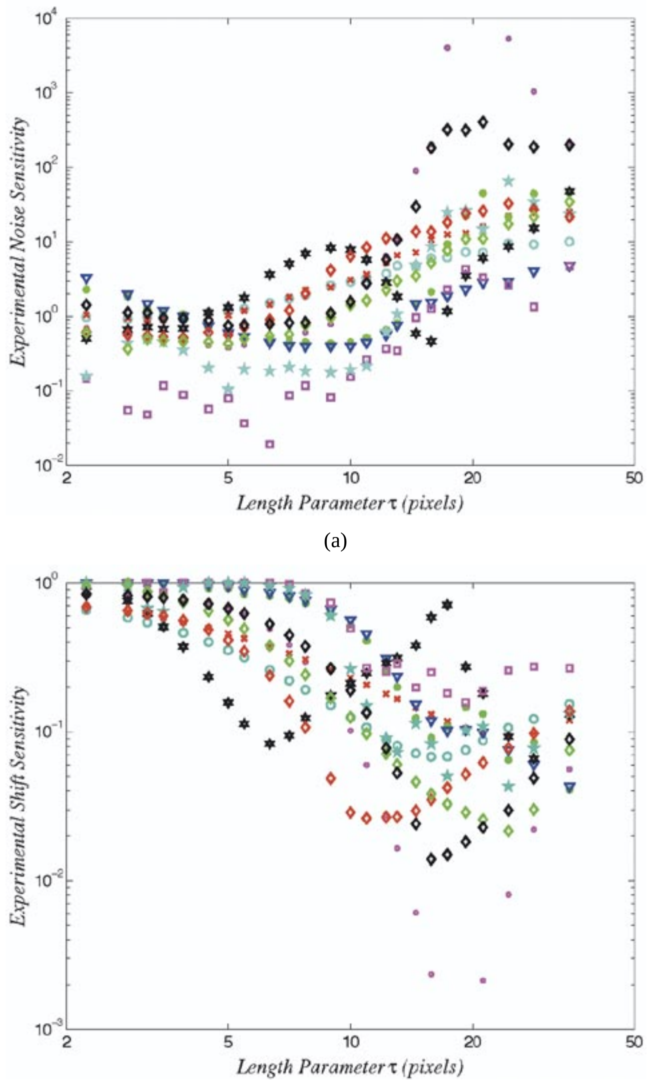
Fig. 3. Theoretical partner to Fig. 2. Shift sensitivity is based on kernel aliasing, and noise sensitivity is measured by the interpolator condition number. Although there are slight differences, qualitatively the same kernels perform well. Results are shown for $\alpha = 8, 10$.

figures are identical: the five kernels along the optimum envelope (lower left) are Gaussian, separable-sinc, smooth, thin-plate, and optimal-interpolation.

To further characterize the properties of the 11 kernels and to assist in the selection of scale parameter τ, Fig. 4 displays experimental noise and shift sensitivity results versus scale parameter τ, for fixed $\alpha = 10$ (i.e., an interpolator size of 100×100 fine-scale pixels). The basic trend, as expected, is that increasing τ produces a smoother kernel, thereby reducing shift and increasing noise sensitivity. Interestingly, the trend reverses for large values of τ: because we have fixed $\alpha = 10$, the interpolator is truncated to 100×100 pixels, and the truncation error increases as τ increases. As $\tau \rightarrow \infty$, all of the truncated interpolants will converge to a constant function over a 100×100 square.

As discussed, for a square, periodic domain, the FFT will have zero shift sensitivity. However, it is interesting to observe that the separable-sinc kernel, which should be equivalent to the FFT, in fact has significant shift sensitivity because of boundary conditions and finite kernel size: the sinc function is very slowly decaying, and our tests are based on finite-sized kernels in a finite-size domain with aperiodic boundaries. On the other hand, the Gaussian function is strongly bandlimited in both the spatial and frequency domains, and so is able to exhibit extremely low shift sensitivities for certain values of τ.

Two additional criteria may be appropriate to consider. First, it may be desirable for the interpolation kernels to be smooth, often for aesthetic reasons. This criterion is already asserted, to some degree, by the shift/aliasing test, which will be sensitive to local or sharply peaked interpolators. Second, if the position of the coarse elements is irregular, or if the interpolation kernel footprint is finite, then a kernel that passes through zero may not at all (or just barely) sample certain fine-scale elements, making the problem nearly singular. Arguably this issue should have been detected as part of noise sensitivity; however, the noise sensitivity test is problem-geometry dependent and may not detect all such singular cases (note that this issue does not apply to the OI kernels, since these kernels adjust themselves to changes in geometry). These criteria, and the performance of the kernels



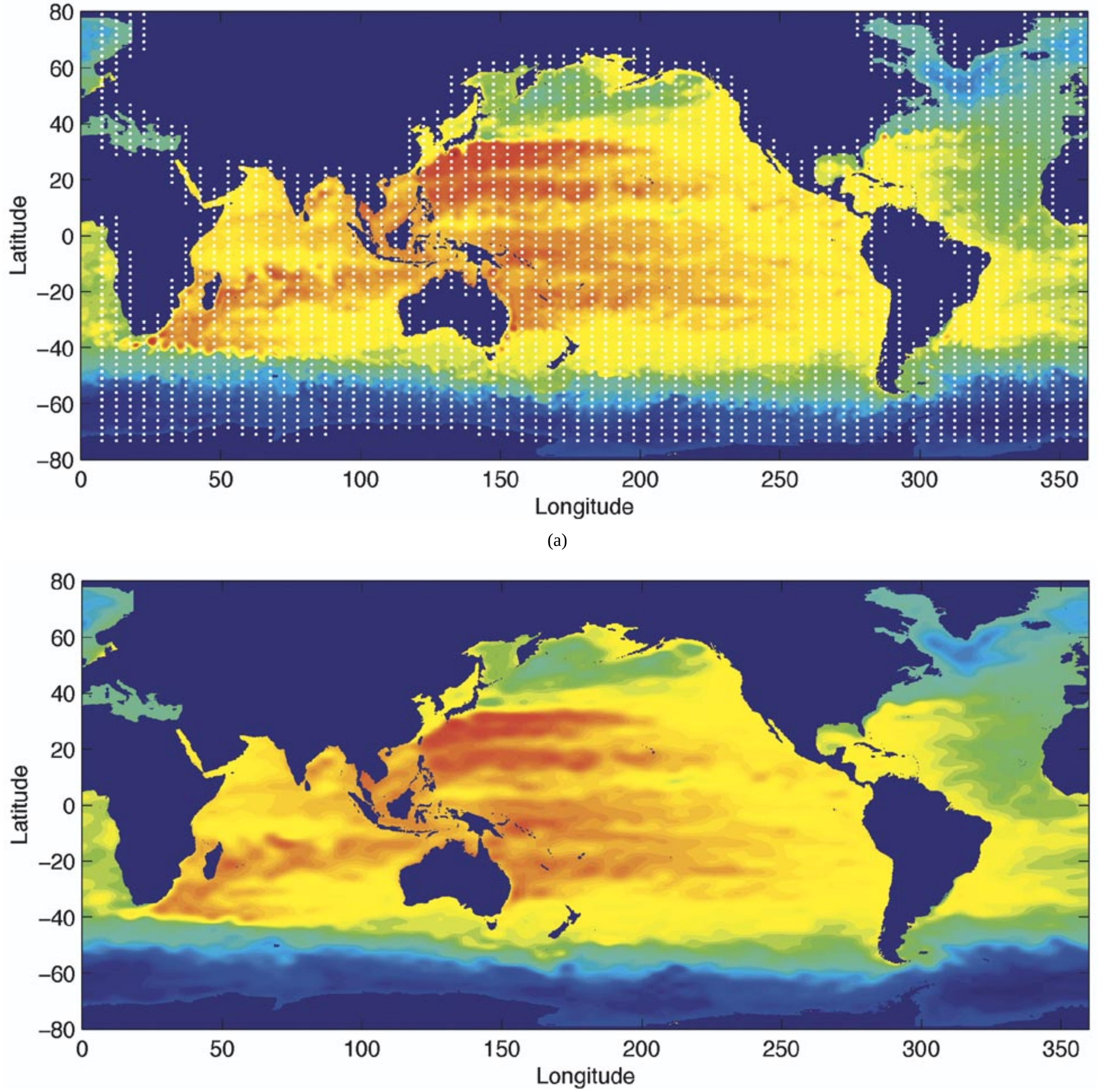
(a)

(b)

Fig. 4. (a) Experimental noise and (b) shift sensitivity results versus scale parameter τ for the 11 interpolation kernels (same symbol legend as in Figs. 2 and 3). In all cases, $\alpha = 10$.

TABLE III
INTERPOLATION KERNEL CONCLUSIONS: UNLESS THE PROBLEM DOMAIN IS A LARGE, REGULAR GRID, AND NUMERICAL CONCERNS ARE NOT AN ISSUE, ONLY THE FINAL THREE KERNELS CAN BE RECOMMENDED FOR GENERAL MAPPING PROBLEMS

Weight	Positivity	Properties	Comments
Gaussian	+	+	OK with good numerics
Nonsep. Exp.	+	−	
Separable Exp.	+		
Bilinear	−		
Cone-shaped	−	−	
Neg.-lobe		−	
Nonsep. Sinc		−	
Sep. Sinc		+	Good if regular grid
Smooth	+	+	Recommended
Thin-Plate	+	+	Recommended
Optimal Interp.	+	+	Recommended



(a)

(b)

Fig. 5. Mapping test for global-scale problem. We have a 71 × 62 coarse grid and a 2160 × 960 fine grid. The centered locations of the 3551 interpolants are shown as white dots in the top panel; each interpolant has a footprint of 121 × 81 pixels, or 20 × 13°. (b) Result of fine-coarse-fine mapping.

with respect to them, are summarized in Table III: the positivity column assesses whether kernels are strictly positive (+) or have zero-crossings (−), and the properties column assesses the extent to which the shift and noise sensitivities lie on the desired (lower left) envelope in Figs. 2 and 3. The Gaussian kernel must be used with caution: although it has outstanding shift sensitivity, it also has the greatest noise sensitivities. The large noise sensitivites can lead to numerical problems if insufficient numerical accuracy is used.

Finally, Fig. 5 shows a mapping test for a global-scale problem. The centered locations of the 3551 interpolants are shown as white dots in the top panel, where each interpolant has a footprint of 20 × 13°. The bottom panel, which shows the result of fine-coarse-fine mapping, shows no evidence of distortions or noise amplifications, despite the use of a Gaussian kernel. Although it appears that a number of weights lie over land, these "land" weights actually capture coastline information because of the size of the kernel footprint.

V. Summary and Recommendations

Three efficient interpolation and pseudoinverse algorithms have been described. The first of these, the FFT algorithm, is only appropriate for regular, periodic grids. Nevertheless, the FFT algorithm provides theoretical intuition for the properties of mapping and pseudoinverse operators. Specifically, noise sensitivity is minimized by choosing interpolation and state-reduction operators that have flat wavenumber response, and shift sensitivity is minimized by choosing bandlimited operators.

A second efficient algorithm is based on an implicit representation of the pseudoinverse operator. This method is recommended for all but the largest problems because it is exact and relatively efficient.

For the largest problems, we have developed an iterative approach that is extremely efficient, but with the drawback that questions of convergence rate, stability, and parameter settings are functions of the interpolant w and therefore need to be tested empirically in a given setting of interest. However, the advantages are astonishing: the initialization step is so fast that one could, if appropriate, choose a *different* interpolator w at *each* time step. Furthermore the memory and computational requirements make it feasible to define multiple interpolants w_k, apply *all* of them, and chose between the results in some context-sensitive manner.

The computational efficiency of the iterative algorithm makes it possible to test a large number of interpolation kernels and parameters in relation to desirable properties of the interpolation and mapping operations. Eleven kernels were tested for a wide range of problem sizes, kernel scale τ, and region of support α. In general, τ trades off noise versus shift sensitivity, and α trades off accuracy versus computational cost. In most problems, where numerical issues may be a concern and where the coarse domain possesses irregularities due to coastlines, only three kernels—smooth, thin plate, and optimal interpolation—can be recommended.

APPENDIX I
SHIFT SENSITIVITY

If the coarse and fine domains are regularly gridded and periodic and if the interpolation operator is stationary, then the fine scale is given by the convolution of weight $w(i, j)$ from (9) with an oversampled version of the coarse scale

$$\boldsymbol{x}_f = \boldsymbol{w} * (\uparrow \boldsymbol{x}_c). \tag{24}$$

Shift insensitivity, (15), implies that for any coarse vector $\boldsymbol{x}_c$ and shift operation $\mathcal{S}$, there exists a new coarse vector $\overline{\boldsymbol{x}}_c$ that satisfies

$$\mathcal{S}\boldsymbol{w} * (\uparrow \boldsymbol{x}_c) = \boldsymbol{w} * (\uparrow \overline{\boldsymbol{x}}_c). \tag{25}$$

Taking the Fourier transform of (25) and using elementary properties of the Fourier transform, we obtain

$$\mathcal{W}\mathcal{F}_2(\uparrow \boldsymbol{x}_c)e^{2\pi i(k_x\delta_x/N_x + k_y\delta_y/N_y)} = \mathcal{W}\mathcal{F}_2(\uparrow \overline{\boldsymbol{x}}_c) \tag{26}$$

where $\mathcal{W}$ is the Fourier transform of $\boldsymbol{w}$; δ_x and δ_y are the spatial shifts in the x and y directions, respectively; and N_x and N_y are the dimensions of the fine grid. Note that although the number of degrees of freedom in $\mathcal{F}_2(\uparrow \boldsymbol{x}_c)$ is small, equal to the number of coarse-grid points, the wavenumber content is much higher, generally spanning the complete range of fine-grid wavenumbers, due to discontinuities in the oversampled signal $(\uparrow \boldsymbol{x}_c)$. Therefore, to satisfy (26) exactly for all possible $\boldsymbol{x}_c$, δ_x, and δ_y, the number of nonzero elements in filter $\mathcal{W}$ must be exactly equal to the number of coarse-grid points, i.e., to achieve zero shift sensitivity (16), $\mathcal{W}$ must be bandlimited with a bandwidth equal to that spanned by the coarse grid. (Because of symmetry constraints at Nyquist, the shift sensitivity will in fact be exactly zero only for coarse domains $m \times n$ where both m and n are odd.) As an approximation, we may measure the shift sensitivity as the fraction of energy that is aliased

$$1 - \frac{\displaystyle\sum_{i=-N_x/2s}^{N_x/2s} \sum_{j=-N_y/2s}^{N_y/2s} \mathcal{W}(i, j)^2}{\displaystyle\sum_i \sum_j \mathcal{W}(i, j)^2}. \tag{27}$$

APPENDIX II
INTERPOLANTS

Of the 11 interpolants tested, two have implicit definitions. First, the "thin-plate" interpolant is found from a second-order constraint. An efficient approach to the generation of such functions is using a 2-D FFT. For a correlation length of `corr_len`, a 64 × 64 interpolant kernel can be generated in Matlab as

```
% Create Thin-Plate convolutional mask
w = zeros (64, 64);
m = [ 0   0   1   0  0;
      0   2  -8   2  0;
      1  -8  20  -8  1;
      0   2  -8   2  0;
      0   0   1   0  0];
for i = 1:5,  for j = 1:5,
  w(1 + rem (61 + i, 64), 1 + rem(61 + j, 64)) = m(i, j);
end;  end;
w(1, 1) = 20 + 10^(-6 - 4 * log 10(corr_len/53));

% Invert this to get correlation
w = real (ifft2(1 ./ (real(fft2(w)))));
w = fftshift (w ./ w(1, 1)).
```

Second, the "optimal-interpolation" interpolant is found as the set of weights for the least-squares estimation of the fine-scale values from the coarse-scale ones. We define $x(i)$, $y(i)$ as the spatial coordinate of the ith element, subscripted for the fine and coarse scales. There are three parameters to specify: the length scales σ_x, σ_y and the weight ψ (for this paper, ψ was varied from 0.01–1.0). The interpolation follows as

$$\begin{aligned} P_{FF}(i, j) &= \psi\delta_{i,j} \\ &\quad + \exp\left\{-\left[\frac{x_f(i) - x_f(j)}{\sigma_x}\right]^2 - \left[\frac{y_f(i) - y_f(j)}{\sigma_y}\right]^2\right\} \end{aligned} \tag{28}$$

$$P_{CF}(i, j) = \exp\left\{-\left[\frac{x_c(i) - x_f(j)}{\sigma_x}\right]^2 - \left[\frac{y_c(i) - y_f(j)}{\sigma_y}\right]^2\right\} \tag{29}$$

from which the interpolant is derived as

$$\boldsymbol{x}_f = \boldsymbol{P}_{CF}\boldsymbol{P}_{FF}^{-1}\boldsymbol{x}_c. \tag{30}$$

In practice, this computation is impossibly large to undertake by brute force, so the problem is usually divided spatially, and only local least squares is performed.

REFERENCES

[1] I. Fukumori, "Data assimilation by models," in *Satellite Altimetry and Earth Sciences*, L.-L. Fu and A. Cazenave, Eds. San Diego, CA: Academic, 2001, ch. 5, pp. 237–265.

[2] I. Fukumori and P. Malanotte-Rizzoli, "An approximate Kalman filter for ocean data assimilation: An example with an idealized Gulf Stream model," *J. Geophys. Res.*, vol. 100, no. C4, pp. 6777–6793, Apr. 1995.

[3] I. Fukumori, "Assimilation of TOPEX sea level measurements with a reduced-gravity, shallow water model of the tropical Pacific Ocean," *J. Geophys. Res.*, vol. 100, no. C12, pp. 25 027–25 039, Dec. 1995.

[4] M. A. Cane, A. Kaplan, R. N. Miller, B. Tang, E. C. Hackert, and A. J. Busalacchi, "Mapping tropical Pacific sea level: Data assimilation via a reduced state space Kalman filter," *J. Geophys. Res.*, vol. 101, pp. 22 599–22 617, 1996.

[5] D. Menemenlis and C. Wunsch, "Linearization of an oceanic circulation model for data assimilation and climate studies," *J. Atmos. Oceanic Technol.*, vol. 14, no. 6, pp. 1420–1443, 1997.

[6] D. Menemenlis, T. Webb, C. Wunsch, U. Send, and C. Hill, "Basin-scale ocean circulation from combined altimetric, tomographic and model data," *Nature*, vol. 385, pp. 618–621, 1997.

[7] D. T. Pham, J. Verron, and M. C. Roubaud, "A singular evolutive extended Kalman filter for data assimilation in oceanography," *J. Mar. Syst.*, vol. 16, pp. 323–340, 1998.

[8] The ATOC Consortium, "The North Pacific general circulation and heat budget from acoustic tomography, satellite altimetry, and a general circulation model," *Science*, vol. 281, pp. 1327–1332, 1998.

[9] I. Fukumori, R. Raghunath, L.-L. Fu, and Y. Chao, "Assimilation of TOPEX/Poseidon altimeter data into a global ocean circulation model: How good are the results?," *J. Geophys. Res.*, vol. 104, no. C11, pp. 25 647–25,665, Nov. 15, 1999.

[10] T. M. Chin, A. J. Mariano, and E. P. Chassignet, "Spatial regression and multiscale approximations for sequential data assimilation in ocean models," *J. Geophys. Res.*, vol. 104, pp. 7991–8014, 1999.

[11] N. Hirose, I. Fukumori, and J.-H. Yoon, "Assimilation of TOPEX/POSEIDON altimeter data with a reduced gravity model of the Japan Sea," *J. Oceanogr.*, vol. 55, pp. 53–64, 1999.

[12] J. Verron, L. Gourdeau, D. T. Pham, R. Murtuguddle, and A. J. Busalacchi, "An extended Kalman filter to assimilate satellite altimeter data into a nonlinear numerical model of the tropical Pacific Ocean," *J. Geophys. Res.*, vol. 104, pp. 5441–5458, 1999.

[13] I. Fukumori, "A partitioned Kalman filter and smoother," *Mon. Weather. Rev.*, vol. 130, pp. 1370–1383, 2002.

[14] A. Albert, *Regression and the Moore–Penrose Pseudoinverse*. New York: Academic, 1972.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed. New York: Cambridge Univ. Press, 1992.

Paul W. Fieguth (S'87–M'96) received the B.A.S. degree from the University of Waterloo, Waterloo, ON, Canada, in 1991, and the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 1995, both in electrical engineering.

He is currently an Associate Professor in Systems Design Engineering at the University of Waterloo. Prior to appointment at the University of Waterloo, he held postdoctoral positions in the Department of Computer Science, University of Toronto, Toronto, ON, Canada, and in the Laboratory for Information and Decision Systems, MIT. His research interests include statistical signal and image processing, hierarchical algorithms, data fusion, and the interdisciplinary applications of such methods, particularly to remote sensing.

Dimitris Menemenlis received the B.Eng. degree from McGill University, Montreal, QC, Canada, in 1984, the M.A.S. degree from Waterloo University, Waterloo, ON, Canada, in 1987, and the Ph.D. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada, in 1993.

He is currently with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena. From 1993 to 1998, he was a Research Scientist at the Massachusetts Institute of Technology, Cambridge. His present research interests include the application of inverse methods to the study of ocean and sea-ice circulations and their interaction with biogeochemical tracer cycles and climate.

Ichiro Fukumori (M'92) received the B.S. degree in geophysics from the University of Tokyo, Tokyo, Japan, in 1983, and the Ph.D. degree in physical oceanography from the Massachusetts Institute of Technology, Cambridge, and the Woods Hole Oceanographic Institution, Boston, MA, in 1989.

He is currently with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena. His research interests include studies of seasonal-to-interannual changes of the ocean and their effect on climate, and development and applications of numerical modeling and data assimilation methodologies.