Mapping and Pseudo-Inverse Algorithms for Data Assimilation

Paul Fieguth
Systems Design Engineering, University of Waterloo

Dimitris Menemenlis and Ichiro Fukumori
Ocean Science Research Element, Jet Propulsion Laboratory

I. Overview

Abstract:

Among existing ocean data assimilation methodologies, reduced-state Kalman filters are a widely-studied compromise between resolution, and computational feasibility. Such reduced-state filters require mapping operators from the fine grid to the reduced state and vice-versa; that is, that the state-reduction and interpolation operators be pseudo-inverses of each other.

This poster investigates a variety of approaches to computing the pseudoinverse and also evaluates the mapping performance of eleven interpolation kernels.

Introduction:

Goal: to understand and predict the general circulation of the oceans.

Existing approaches remain a compromise between resolution, optimality, error specification, and computational feasibility. Widely-studied compromise: reduced-state Kalman filter in which the measurement update takes place on a reduced state compared to the full state of the Ocean General Circulation Model (OGCM).

Main challenge: require mapping operators from the fine (OGCM) state to the reduced state and vice-versa.

Let $\mathbf{x}_f$ and $\mathbf{x}_c$ represent the fine and coarse state vectors. State reduction $\mathbf{B}^*$ and interpolation $\mathbf{B}$ operations defined such that

$$\mathbf{x}_c = \mathbf{B}^*\mathbf{x}_f, \qquad \mathbf{x}_f = \mathbf{B}\mathbf{x}_c + \boldsymbol{\epsilon}, \qquad \mathbf{B}^*\mathbf{B} = \mathbf{I}. \tag{1}$$

$\mathbf{B}^*$ and $\mathbf{B}$ are pseudoinverses, a condition which ensures that repeated subsampling and interpolation do not lead to a degradation of the coarse-scale data:

$$\mathbf{x}_c = \mathbf{B}^*\mathbf{B}\mathbf{x_c} \tag{2}$$

Objective: to define fast, storage-efficient methods of finding $\mathbf{B}^*$ from $\mathbf{B}$.

Existing mapping and pseudo-inverse schemes often involve the brute-force computation:

$$\boldsymbol{B} = \boldsymbol{B}^{*T}\left(\boldsymbol{B}^*\boldsymbol{B}^{*T}\right)^{-1}, \quad \boldsymbol{B}^* = \left(\boldsymbol{B}^T\boldsymbol{B}\right)^{-1}\boldsymbol{B}^T. \tag{3}$$

Where the matrices are of size $n_f \times n_c$, 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.

Magnitude of challenge: Suppose we have a global problem with 1/12°-spacing: $n_f \simeq 10^7$. Suppose the coarse grid has grid spacing of 2°: $n_c \simeq 10^4$. Then the mapping and pseudo-inverse operations, stored as dense matrices, are each 1 TERABYTE in size!

Inversion Criteria:

In addition to a computationally efficient approach to identifying a pseudoinverse, the interpolation kernel in $\mathbf{B}$ must satisfy at least two other requirements.

First: sensitivity to lateral translations must be minimized, to ensure that a slow, advective flow is not progressively corrupted by repeated mapping-interpolations:

$$\mathcal{S}\boldsymbol{B}\boldsymbol{B}^* \approx \boldsymbol{B}\boldsymbol{B}^*\mathcal{S}, \tag{4}$$

where $\mathcal{S}$ represents a spatial translation on the fine scale. This is effectively an antialiasing or bandlimiting criterion.

Second: insensitivity to noise, that is, we wish to limit the coarse-scale amplification of fine-scale perturbations. The noise sensitivity is proportional to

$$\frac{|\bar{\boldsymbol{x}}_c - \boldsymbol{x}_c|}{|\boldsymbol{\delta}|}\frac{|\boldsymbol{x}_f|}{|\boldsymbol{x}_c|} = \frac{|\boldsymbol{B}^*\boldsymbol{\delta}|}{|\boldsymbol{\delta}|}\frac{|\boldsymbol{B}\boldsymbol{x}_c|}{|\boldsymbol{x}_c|}. \tag{5}$$

The upper bound for this sensitivity is given by the condition number of $\mathbf{B}$ or $\mathbf{B}^*$:

$$\sigma_{\max}(\boldsymbol{B}) * \sigma_{\max}(\boldsymbol{B}^*) = \mathrm{cond}(\boldsymbol{B}) = \mathrm{cond}(\boldsymbol{B}^*) \geq 1. \tag{6}$$

II. Fast Inversion

FFT:

Computing the pseudoinverse by brute force requires enormous storage and computational effort. A simple intuitive approach is to use the FFT:

$$\boldsymbol{x}_f = \mathcal{F}_2^{-1}\left[\,\mathcal{W}(k_x,k_y)\,\mathcal{F}_2(\uparrow \boldsymbol{x}_c)\right], \tag{7}$$

$$\boldsymbol{x}_c = \downarrow \mathcal{F}_2^{-1}\left[\,\mathcal{W}^*(k_x,k_y)\,\mathcal{F}_2(\boldsymbol{x}_f)\right]. \tag{8}$$

Very efficient and fast, however it makes strict stationarity and periodicity assumptions, are incompatible with irregularities (e.g., coastlines).

Subsampling:

Subsampling methods allow a straightforward alternative to the brute-force approach; define $\boldsymbol{x}_s$ of intermediate resolution:

$$\begin{array}{ccccc} \boldsymbol{x}_c & \xleftarrow{\boldsymbol{B}_0^*} & \boldsymbol{x}_s & \xleftarrow{\boldsymbol{B}_1^*} & \boldsymbol{x}_f \\ \boldsymbol{x}_c & \xrightarrow{\boldsymbol{B}_0} & \boldsymbol{x}_s & \xrightarrow{\boldsymbol{B}_1} & \boldsymbol{x}_f \end{array} \tag{9}$$

Key Idea — The pseudoinverse of $\boldsymbol{B}_1^*$ is very easily found:

$$\boldsymbol{B}_1 = \boldsymbol{B}_1^{*T} + \boldsymbol{B}_2, \tag{10}$$

such that a row in $\boldsymbol{B}_2$ is zero if the corresponding row of $\boldsymbol{B}_1^{*T}$ is non-zero. Problem: the subsampling operator introduces aliasing and leads to substantial shift-sensitivities.

Implicit Inversion:

Implicit methods avoid explicitly computing $\mathbf{B}^*$ from $\mathbf{B}$, i.e.,

$$\mathbf{x}_c = \mathbf{B}^*\mathbf{x}_f = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{x}_f = \mathbf{Q}^{-1}(\mathbf{B}^T\mathbf{x}_f) \tag{11}$$

However even the "small" dense matrix $\mathbf{Q}^{-1}$ can be unwieldy, both for storage and inversion complexity, for global-sized problems.

Iterative Inversion:

Instead, we propose to iteratively solve the linear system

$$\mathbf{Q}\mathbf{x}_f = \bar{\mathbf{x}}_f \tag{12}$$

which is vastly simpler because of the sparsity of $\mathbf{Q}$. We apply the Conjugate Gradient method because of its efficiency and simplicity.

Following table compares storage and computational complexity for 100×100 coarse-scale and 1000×1000 fine-scale problem:

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

The iterative approach offers *tremendous* reduction in storage and computational complexity!

Actual reduction in complexity depends on sparsity of Q and i, the number of conjugate-gradient iterations required for convergence:

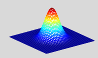
Problem Size	$\boldsymbol{Q}$ Density	Interpolator Size τ (fine-scale pixels)						
		2	3	5	8	12	17	28
33×33	0.09	4	6	11	41	174	303	240
29×29	0.12	3	6	11	43	165	291	245
25×25	0.15	3	6	11	41	169	283	233
21×21	0.21	3	6	11	40	158	290	223
17×17	0.30	3	6	11	41	155	238	195
13×13	0.45	3	6	11	38	115	232	168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

We show the average number of conjugate-gradient iterations to achieve a root-mean-squared accuracy of 0.5%.

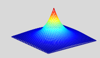
III. Kernels

Kernels Tested:

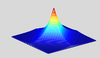
We have evaluated the shift and noise sensitivities for eleven different interpolation kernels:



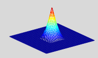
$e^{-(r/\tau)^2}$
Gaussian



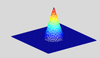
$e^{-(r/\tau)}$
Nonsep. Exp.



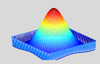
$e^{-|x|/\tau}e^{-|y|/\tau}$
Sep. Exp.



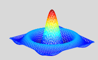
$\left(1-\frac{|x|}{\tau}\right)\left(1-\frac{|y|}{\tau}\right)$
Bilinear



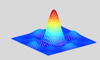
$1-\frac{r}{\tau}$
Nonsep. Linear



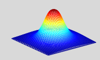
$1+\frac{r}{\tau}-\frac{1}{3}\left(\frac{r}{\tau}\right)^3 e^{-r/\tau}$
Smooth



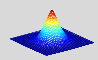
$\mathrm{sinc}(r/\tau)$
Nonsep. Sinc



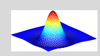
$\mathrm{sinc}(x/\tau)\cdot\mathrm{sinc}(y/\tau)$
Separable Sinc



$\left(1-e^{-(\frac{r}{\tau})^2}\right)e^{-\frac{r}{\tau}}+e^{-(\frac{r}{\tau})^2}$
Hybrid



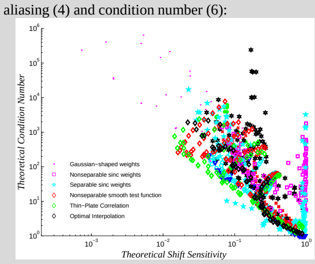
(Implicit)
Thin-Plate



(Implicit)
Obj. Analysis

Kernel Assessments:

All tests were carried out in 20×20-coarse-scale, 200×200-fine-scale domains. The theoretical tests measure aliasing (4) and condition number (6):



We can validate these tests experimentally. The shift sensitivity is defined as the root-mean-square ratio

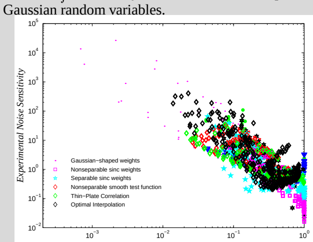
$$\frac{\mathrm{rms}\left\{\mathcal{S}(\boldsymbol{B}\boldsymbol{\delta}_i) - \boldsymbol{B}\boldsymbol{B}^*[\mathcal{S}(\boldsymbol{B}\boldsymbol{\delta}_i)]\right\}}{\mathrm{rms}\left\{\boldsymbol{B}\boldsymbol{\delta}_i\right\}}, \tag{13}$$

where $\boldsymbol{\delta}_i$ is a coarse unit-vector with pixel i set to one and the rest to zero.

Noise sensitivity is measured by computing the reaction to noise:

$$\frac{\mathrm{rms}(\boldsymbol{B}^*\boldsymbol{N}_f)}{\left\{\Sigma_i\, w^2(i)\right\}^{1/2}}, \tag{14}$$

where $\boldsymbol{N}_f$ is an array of unit-variance, independent, Gaussian random variables.

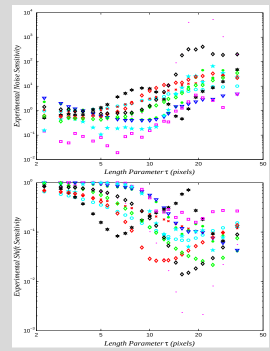


Surprisingly, common kernels such as bilinear, exponential, Gaussian, and sinc functions performed only moderately well.

IV. Results

Scale Sensitivity:

A summary illustration of the sensitivity of various interpolants to the choice of scale. Generally, a larger scale leads to smoother interpolants, less aliasing (shift sensitivity), and larger condition number:

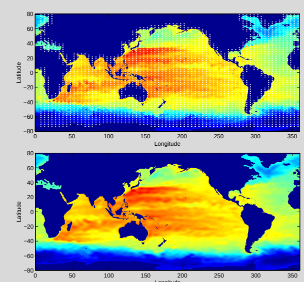


Kernel Conclusions:

Based on our test results we propose that the Hybrid, Thin-Plate, or Objective Analysis kernels have superior properties and should be recommended for mapping exercises:

Weight	Positivity	Properties	Comments
Gaussian	+	+	Numeric issues
Nonsep. Exp.	+	–	
Separable Exp.	+		
Bilinear	–		
Cone-shaped	–	–	
Neg.-lobe		–	
Nonsep. Sinc		–	
Sep. Sinc		+	Regular Grid
Smooth	+	+	Recommended
Thin-Plate	+	+	Recommended
Optimal Interp.	+	+	Recommended

Real Data Example:



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 degrees. The bottom panel shows the result of fine-coarse-fine mapping.