The Inverse Ocean Modeling System. Part I: Implementation

A. F. BENNETT, B. S. CHUA, AND B. L. PFLAUM

College of Oceanic and Atmospheric Sciences, Oregon State University, Corvallis, Oregon

M. ERWIG

School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, Oregon

Z. FU

Microsoft Corporation, Redmond, Washington

R. D. LOFT

Scientific Computing Division, National Center for Atmospheric Research, Boulder, Colorado

J. C. MUCCINO

College of Civil and Environmental Engineering, Arizona State University, Tempe, Arizona

(Manuscript received 2 August 2006, in final form 16 January 2008)

ABSTRACT

The Inverse Ocean Modeling (IOM) system constructs and runs weak-constraint, four-dimensional variational data assimilation (W4DVAR) for any dynamical model and any observing array. The dynamics and the observing algorithms may be nonlinear but must be functionally smooth. The user need only provide the model and the observing algorithms, together with an interpolation scheme that relates the model numerics to the observer's coordinates. All other model-dependent elements of the Inverse Ocean Modeling assimilation algorithm (see both Chua and Bennett), including adjoint generators and Monte Carlo estimates of posteriors, have been derived and coded as templates in Parametric FORTRAN (Erwig et al.). This language has been developed for the IOM but has wider application in scientific programming. Guided by the Parametric FORTRAN templates, and by model information entered via a graphical user interface (GUI), the IOM generates conventional FORTRAN code for each of the many algorithm elements, customized to the user's model. The IOM also runs the various W4DVAR assimilations, which are monitored by the GUI. The system is supported by a Web site that includes interactive tutorials for the assimilation algorithm.

1. Introduction

Variational assimilation of real observations into numerical models has emerged as an essential activity in meteorology and oceanography. Early studies include Talagrand and Courtier (1987), Courtier and Talagrand (1987), Bennett and McIntosh (1982), and McIntosh and Bennett (1984); recent texts are Talagrand (2008), Lewis et al. (2006), and Bennett (1992, 2002). Major meteorological activities include the weak-constraint, four-dimensional variational data assimilation (W4DVAR) atmospheric operational analyses at the European Centre for Medium Range Weather Forecasting (ECMWF; see http://www.ecmwf.int/products/forecasts/guide/The_four_dimensional_data_ assimilation_4DVAR.html) and at the U.S. Naval Research Laboratory (NAVDAS–AR; Xu et al. 2006). A major oceanographic assimilation activity is the Estimating the Circulation and Climate of the Ocean (ECCO) oceanic research analysis at the Scripps Institution of Oceanography (http://ecco.ucsd.edu/). These variational analyses are, in effect, constrained least squares regressions. The minimized value of the

Corresponding author address: Andrew F. Bennett, 104 COAS Admin. Bldg., Oregon State University, Corvallis, OR 97331-5503.
E-mail: bennett@coas.oregonstate.edu

DOI: 10.1175/2008JTECHO519.1

least squares estimator is in fact the conventional significance test statistic for the hypothesized means and covariances (the priors) for the errors in a model, along with its initial conditions, boundary conditions, and the data (see, e.g., Bennett et al. 1998, 2000, 2006).

While the value of variational assimilation is widely acknowledged, its widespread adoption has been hampered by the need to develop great amounts of software beyond that involved in a conventional forward model. The mostly commonly cited labor is the development and coding of the so-called adjoint model, but many other algorithmic elements must also be developed, coded, and intricately linked.

The Inverse Ocean Modeling (IOM) system minimizes the effort required for W4DVAR. The IOM minimization algorithm is an iterated implementation of the "indirect representer method" (Bennett and Thorburn 1992; Egbert et al. 1994) or "dual-adjoint method" (Amodei 1995). For detailed presentations of the algorithm, see Courtier (1997), Chua and Bennett (2001), or Bennett (2002).

Development of the IOM system has been guided by experiences gained while implementing the IOM algorithm for a variety of contemporary ocean models:

(i) Advanced Circulation Model (ADCIRC; see http://www.nd.edu/~adcirc/), a finite-element shallow-water model, presently used operationally for storm surge forecasting by the U.S. Navy Oceanographic Office (NAVO), the U.S. Army Corps of Engineers, and the U.S. Federal Emergency Management Authority (FEMA);

(ii) Primitive Equation Z-Coordinate Model (PEZ), an efficient coding of the free-surface Bryan and Cox model, used for inverse modeling of ENSO phenomena (after Bennett et al. 2006) and for harmonically analyzed internal tides over the Hawaiian Ocean Ridge (Zaron and Egbert 2006);

(iii) Regional Ocean Modeling System (ROMS; see http://www.myroms.org/), a general purpose community model in widespread use; and

(iv) Spectral Element Ocean Model (SEOM; see http://marine.rutgers.edu/po/index.php?model=seom), a shallow-water model with advanced numerics.

These codes and their numerics might be described as either "legacy" (ADCIRC), "classic" (PEZ), "mature" (ROMS), or "prototypical" (SEOM). Each presented unique problems to the IOM, and so the flexibility of the latter has been well established. A companion paper (Muccino et al. 2008) describes experiences using the IOM with the models (i)–(iv) above.

The user operates the IOM via a graphical user interface (GUI). An extensive IOM user manual (Bennett et al. 2007) is available at the IOM Web site (http://iom.asu.edu), illustrating the software protocols and Network Common Data Format, Input/Output (NetCDF I/O) conventions using several simple ocean models. The user need *not* know the IOM template language (Parametric FORTRAN; Erwig and Fu 2004; Erwig et al. 2006, 2007), in which key components of the IOM have been written. However, there are a few simple Parametric FORTRAN commands that greatly empower the user. Inserting these commands, with user-selected parameters, into the FORTRAN 90 code for the ocean model or the observing sensor converts those codes to Parametric FORTRAN. The compiler for Parametric FORTRAN then generates the FORTRAN 90 code for either the tangent linearization or the adjoint, as required.

The contents of this introductory description of the IOM are as follows:

- *The IOM minimization algorithm* reviews the state of data assimilation methodology, updating the discussion in Chua and Bennett (2001).
- *The IOM algorithm—a sketch* outlines the math in the barest of terms, for a trivial model. Flow charts in later sections support this brief summary. References are provided for the reader seeking further details.
- *What the user must provide* is a list in the broadest of terms. It will be seen that IOM advancements have massively reduced the effort required by the user, especially for the development of the auxiliary models (tangent linear and adjoint) and the preparation of community data [Argo, Global Drifter Program, National Centers for Environmental Prediction (NCEP) Global SST Analysis, *Jason-1*, and Tropical Atmosphere Ocean/Triangle Trans-Ocean Buoy Network (TAO/TRITON)].
- *What the IOM will do for the user* emphasizes automatic code generation, the management of computations, and the preparation of data.
- *The IOM Web site* is the user's first contact with the IOM, and so a moderate level of detail is included here.
- The *Summary* comments on the broader advantages of rationalizing the development of software for oceanic and atmospheric analysis, and the power of modern information technology to that end.

2. The IOM minimization algorithm

a. Preamble

The minimization algorithm for the Inverse Ocean Modeling system is the iterated, indirect representer algorithm. The system of Euler–Lagrange (EL) equations for extrema of a penalty functional is solved in two stages. The first is a Picard iteration that yields a sequence of linear systems of Euler–Lagrange equations (Bennett and Thorburn 1992). The second is an indirect solution of each linear system (Egbert et al. 1994; Amodei 1995). The complete details of the algorithm may be found in Chua and Bennett (2001) and Bennett (2002). A mathematical "sketch" is provided in section 3 below. The choice of algorithm is reviewed and justified here in light of the latest experience in data assimilation.

The IOM is modular: it accepts any functionally smooth model and any functionally smooth measurement functional (the mathematical transformation that extracts a single number from an ocean state). This modularity presents many mathematical and software engineering challenges. Consequently, the IOM was designed to support only one formulation of the data assimilation problem, namely, W4DVAR. All of the constraints including the data, the initial values of the ocean state, the boundary values of the state, the externally imposed stresses, and the dynamical equations themselves are imposed only weakly. That is, the ocean state estimate is only a weighted, least squares best fit to any constraint. Each weight is chosen inversely to the hypothesized error covariance for the constraint. It may be noted that in the so-called strong-constraint formulation, the numerically approximated dynamics are satisfied exactly. Of course, the strong formulation is a special case for the IOM, in which the dynamics are given arbitrarily large weights (vanishingly small dynamical error variances). The IOM minimizes the weighted least squares fit or "cost" or "penalty" functional by the iterated, indirect representer algorithm. The IOM also makes available several key indicators of sensitivities.

The W4DVAR scheme is an example of a fixed-interval smoother defined by a penalty functional. Again, the IOM minimizes the penalty functional in two stages. The first stage (the "outer iteration") addresses any nonlinearity in the dynamics or the measurement functionals by replacing the original penalty with a sequence of penalties. Each penalty in the sequence is quadratic in the ocean state. The replacement is typically achieved by a tangent linearization of the model dynamics and the measurement functionals as necessary, or more generally by any user-selected linearizing Picard iteration. For a recent example, see Muccino and Luo (2005). The second stage of the IOM algorithm finds the unique solution of the linear Euler–Lagrange problem for the minimum of a quadratic penalty. The linear solution exploits an analysis in terms of the M representers, where M is the number of observations, but unlike the original "direct representer" solution, the IOM solution does not require that all the representers be calculated explicitly. Rather, the solution is found indirectly (the "inner iteration") with appropriate accuracy ($\approx 1\%$) and with far less computation ($\ll M/100$ model integrations). The IOM also provides offline Monte Carlo or ensemble estimates of posterior error statistics.

The choice of minimization algorithm is a secondary consideration, relative to the scientific formulation of the penalty and the scientific interpretation of the result. Yet, choosing an inefficient algorithm can enforce scientifically unsatisfactory compromises in both formulation and interpretation. We have repeatedly heard it stated that "there is no one best choice of data assimilation method." Fortunately, this statement has not entirely suppressed careful investigations of method. Recent investigations include Ngodock et al. (2006) and Zaron (2006).

b. *Data assimilation*

It now seems widely agreed that this imprecise term refers to the estimation of the state of the atmosphere or ocean during some time epoch, using dynamics and observations as constraints. The dynamics need not constitute a complete forecast model, and not every state variable need be observed. It *is* necessary that the choice for every control variable be penalized in some sense (e.g., Bennett and Miller 1991).

There are two major motivations for data assimilation. The first motivation is initializing real dynamical forecasts, possibly including the tuning of dynamical parameters. The resulting ocean state estimate is of intrinsic scientific interest, but only so long as it is credible. The second motivation is, therefore, the scientific testing of dynamical models together with the associated statistical hypotheses concerning errors. Success in the second undertaking builds confidence in the first. Quantitative assessments of the efficiency of the observing system can be extracted from most data assimilation methods, but the credibility of these assessments also depends upon the validity of the underlying error hypotheses.

c. *Weak constraints*

The IOM allows all constraints to be weak: errors are admitted in all information or, expressed another way, the state estimate is allowed to leave residuals in each constraint. The IOM assumes that the errors are jointly normally distributed. Thus it requires that only the first and second moments of the error fields be hypothesized. It is a disturbing reflection upon our knowledge of atmospheric and oceanic physics that we all find even the *rough estimation* of *errors* in subgrid-scale parameterizations so difficult.[1] Error statistics for observations are much better known and are as a rule very much smaller, in relative terms.[2] The number of computational degrees of freedom in W4DVAR for a realistic model can be huge ($\approx 10^9$). The effective number is far less if it is hypothesized that the error fields are widely correlated. In general, however, the number of computational degrees of freedom far exceeds the number of observations ($\approx 10^6$).

In the literature, an assimilation scheme is often described as a "strong-constraint" scheme if no errors are admitted in the dynamics. The remaining control variables include the residuals in the observations themselves, in the initial fields, in any lateral boundary data, in the dynamical parameters and, for ocean models, in the surface fluxes. If the observations lie in a time interval that is considerably shorter than the evolution time scales of the dynamics or the surface fluxes, then the dynamical residuals are less important than the initial residuals. That is, the difference between weak- and strong-constraint assimilation is indistinct. On the other hand, it has been compellingly remarked that modifying the synoptic-scale initial analysis could hardly compensate for subsequent errors in a convective adjustment scheme (R. Elsberry 2001, personal communication).

d. *Four dimensions*

This glamorous-sounding terminology does not indicate the entry of advanced physics into meteorology and oceanography; rather, it indicates the very unglamorous and complicating fact that errors in observations, dynamics, surface fluxes, initial conditions, and lateral boundary data, as badly as these errors are known, show significant correlation in time. Accordingly the estimator or penalty functional for these errors should integrate over time as well as space. The resulting estimate of the instantaneous state will be influenced not only by synoptic observations, but also by observations before and after that instant.

e. *Selection of minimization algorithm*

Assuming the penalty functional for the assimilation problem has been soundly formulated (in this case defining a fixed-interval smoothing problem with all controls penalized), it remains to minimize that penalty. There are all-purpose methods such as simulated annealing, which require only that the penalty be bounded from below. These methods come into their own for highly nonlinear or nonsmooth problems of relatively small size. Relying upon "brute force" searches, the methods exploit virtually no information about the structure of the minimization problem, and so are not competitive for major data assimilation.

Fortunately, it may be assumed without great loss of generality that for data assimilation,

(i) the penalty is quadratic in the residuals, and
(ii) the penalty is smooth in state space, having a classically well-defined gradient and also a classically well-defined Hessian form.

If the dynamics or the measurement functional are nonlinear, then the penalty is not quadratic with respect to the state. However, as a consequence of the assumed smoothness of the penalty, there are tangent linearizations of the constraints that render the penalty quadratic. Moreover, if the penalty is smooth then the gradient may be efficiently constructed, using the operators that are adjoint to the linearized dynamics and measurement functionals. The adjointness is defined with respect to the quadratic penalty. From the control-theoretic perspective, smoothness enables the calculus of variations.

Four minimization algorithms are briefly described below, two of which use gradient information.

1) TRACKING ALGORITHMS

The smoothing problem, that is, the solution of the Euler–Lagrange equations for the minimum of the penalty, may be found using the Reich–Tung–Streibel (RTS) algorithm (e.g., Gelb 1974; Bennett 1992; Wunsch 1996). It is first necessary to linearize the problem with an outer iteration. The dynamics must then be augmented, so that each scalar component of the temporally colored system noise (temporally correlated dynamical error) is a new state variable. The new variable is forced to obey Langevin dynamics having a relaxation time equal to the decorrelation time. The Langevin dynamics are forced by temporally white noise. The RTS algorithm has two stages. The first is the computation of the Kalman filter, including the "forward" propagation of the filter error covariance matrix. Next, this evolving matrix must be inverted at the end of the smoothing interval, and then the inverse matrix must be propagated backward to the beginning of the interval. The RTS algorithm has been applied to plane quasigeostrophic dynamics at moderate resolution (Bennett

[1] It seems prudent in general to estimate that the magnitude of the error is comparable to the magnitude of the flux being parameterized, that is, the parameterization error is 100%.

[2] It is usually dynamically consistent to low-pass filter the observations prior to assimilation; such filtering can reduce error variance considerably, and should be done.

and Budgell 1989), but is not feasible even today for major assimilation with full dynamics. The computations may be greatly reduced by computing the covariances at far less resolution than the state (e.g., Fukumori and Malanotte-Rizzoli 1995), but such economies apply equally well to all other candidate methods mentioned here. The Kalman filter by itself is unsatisfactory as a partial solution to the smoothing problem, not simply because it is significantly suboptimal until the final instant and provides no estimate of the dynamical error, but because of the emergence of bizarre spatial structures near the observing sites.[3] These structures are consequences of the sequential nature of the Kalman Filter, and the nonlinear dependence of the posterior error covariance matrix on the prior covariance matrix at each assimilation step (Bennett and Budgell 1987; Bennett 1992, 2002). A single fixed–interval smoother, covering the same time interval as the filter, has no such bizarre structure (Bennett 2002). However, fixed-interval smoothers run the same risk when applied cyclically to operational forecasting. As is the case at each assimilation time in the Kalman filter algorithm, the posterior error covariance matrix at the end of a cycle or smoothing interval becomes the prior error covariance matrix at the beginning of the next cycle (Bennett 2002).

2) Ensemble estimation

The first two moments of the error fields must be specified for any least squares estimation method. Given this information, it is straightforward to combine a pseudorandom number generator and a square root filter in order to synthesize samples of the error population. A forward integration of the original linear or nonlinear model, forced by the pseudorandom synthetic error field, yields a sample of the random state. Sample moments of the synthesized states yield statistics such as the "measured" values of the Kalman filter error covariance matrix, or the complete set of representers.[4] It is thus feasible to construct approximate filters (Evensen 1994) and approximate smoothers (Bennett 1992). Especially worthy of note are (i) the recent text by Evensen (2006) with accompanying modular software for ensemble Kalman filtering, and (ii) the Data Assimilation Research Test Bed (DART), which is freely available modular software also for ensemble Kalman filtering (http://www.image.ucar.edu/DAReS/DART/). The issue of course is the number of samples needed for stable estimates of the state.[5] Tests with baroclinic models, in which the representer estimate is computed using samples and also variational methods (i.e., by solving the Euler–Lagrange equations, which are in fact equations for second moments), make clear that many hundreds of samples are needed for even a rough approximation to the state (Bennett 2002). It is elementary statistical theory for normal distributions (e.g., Brunk 1965, p. 232) that the relative standard deviation in a sample estimate of a second moment is approximately $\sqrt{2/K}$, where K is the number of independent samples. It is almost always the case in practice that at most one hundred samples are synthesized, thus the population is underexplored. This can be immediately investigated, without making model integrations, by comparing sample moments of the synthesized error fields to the hypothesized moments. The sample variance is typically much smaller than the hypothesized or population variance.

The great attraction of Kalman filters, RTS smoothers, and ensemble estimation schemes is that they do not need gradient information, and so do not require adjoint operators. Yet we know of no national weather service that is using either a Kalman filter or an ensemble method for operational data assimilation, rather than synoptic optimal interpolation as described for example by Daley (1991).

3) State space searches

This method is conceptually simple and very popular. It is expedited by the widespread availability of codes for preconditioned conjugate gradient searches. Indeed, the method is so popular that four-dimensional data assimilation with state space searches and strong dynamical constraints is usually referred to as "4DVAR" or *the* adjoint method. See especially Talagrand and Courtier (1987) and Courtier and Talagrand (1987). Note the specific selections of formulation (strong constraints), and of methodology (gradient search in state space). As with any search method, the dimension of the search space is critical, as is the conditioning of the Hessian form. The dimension, N, of the state space is huge in real problems. Inefficient preconditioning is usual, leading to stalling of the search far from the minimum (Zaron 2006). Such stalling can be overcome by exploiting second-order derivative information, as may be found in the Hessian for the full

[3] This phenomenon of "lock on" bedevils engineers' tracking calculations in general.

[4] It may be remarked in passing that these two structures are identical in size.

[5] Fewer are needed for satisfactory approximations to posterior diagnostics, so long as these diagnostics do not influence subsequent state estimates.

penalty or a partial penalty (Le Dimet et al. 1997; Zaron 2006). Exploiting the Hessian is very difficult and computationally intensive, but has proved to be remarkably effective for some simplified yet still interesting problems. The minimum penalty, which may be directly computed for these linear problems using representers, is actually attained by the second-order or quasi-Newtonian search (Zaron 2006). Nevertheless, such an extremely complex search is much more costly than the *direct* representer algorithm, and so is not at all competitive (Zaron 2006) with the *indirect* representer algorithm. Indeed, the last mentioned is sufficiently economical to allow sufficiently many outer iterations, should the inversion problem be nonlinear.

State space searches have largely been restricted to strong-constraint assimilation. That is, the number of control variables is actually the dimension of the initial state plus, in some instances, that of time-dependent ocean surface fluxes, or of lateral boundary conditions, or the number of uncertain dynamical parameters such as eddy viscosities. In any case, the dimension is far smaller than that of the time-dependent state or "trajectory." Principal examples of strong-constraint data assimilation by state space searches are the ECMWF 4DVAR system (see http://www.ecmwf.int/newsevents/training/) for the initialization of global numerical weather prediction, and the ECCO 4DVAR system (see http://www.ecco-group.org/) for the analysis of general circulation of the global ocean.

A difficulty with state space searches is the apparent need to invert the hypothesized error covariance matrices. Such inversions are unfeasible in general, even for strong-constraint assimilations if they are of realistic size. The remedy is to redefine or "prewhiten" the error fields by scaling with the square root matrices (Courtier 1997). The covariance matrices for the prewhitened fields are then diagonal. The optimal estimates of the unscaled errors are recovered from the optimal estimates of the prewhitened errors, by multiplication with the square root matrices. Such multiplications are computationally intensive but feasible; indeed, the IOM performs similar tasks. It would be simpler to assume diagonal matrices in the first place, that is, spatially uncorrelated initial errors (strong-constraint assimilation) and spatiotemporally uncorrelated dynamical errors (weak-constraint assimilation). The matrix inversions would then of course be trivial, but the resulting optimal estimates of errors would in general be unacceptably finely structured near observing sites (e.g., Zahel 1991). The mathematics of this unphysical fine structure is well understood (Wahba and Wendelberger 1980; Bennett and McIntosh 1982).

4) DATA SPACE SEARCHES

The Sherman–Morrison–Woodbury algorithm (Duncan 1944; Riedel 1991) establishes that the least squares estimation of N correlated real scalars, given the values of M linear combinations of the scalars with $M < N$, is reducible to solving an $M \times M$ linear system. That is, the best fit lies in an M-dimensional subspace of the N-dimensional state space. The $M \times M$ system matrix need *not* be known explicitly; it suffices (see section 3 and the appendix) to be able to calculate the action of the matrix on any vector of length M. The calculation enables a gradient search within the subspace. The condition number of the matrix determines the search efficiency, and assuming a larger error variance in the M "observations" improves the stability of the matrix.[6] Preconditioning the search in the subspace is a vastly smaller task than preconditioning the state space search.

The representer algorithm is an instance of the Sherman–Morrison–Woodbury algorithm. The $M \times N$ structure appearing in the former consists of the M representers of length N; these may be calculated using the Euler–Lagrange equations. The columns of the $M \times M$ matrix (the "representer matrix") are the values of the M linear functionals, or measurements, of the representers. The IOM provides a suite of preconditioners for the search in the data subspace.

By implementing the search in the data subspace, the indirect representer algorithm (Egbert et al. 1994) or dual-adjoint algorithm (Amodei 1995) precludes the need to calculate all the M representers. The resulting gains in efficiency are enormous (e.g., Bennett et al. 1996, 1997). Bennett et al. (2006) report needing only 30 search steps (each of one backward integration, one space–time convolution, and one forward model integration) for $N \approx 10^8$, $M \approx 2000$, while Xu et al. (2005) and Rosmond and Xu (2006) report needing only 50 steps for $N \approx 4 \times 10^6$, $M \approx 4 \times 10^5$.

It must also be stressed that even a direct application of the Sherman–Morrison–Woodbury algorithm has no need for the inversion of an $N \times N$ matrix, where N is the dimension of the state space. Indeed, that is the benefit of the algorithm.

The Sherman–Morrison–Woodbury algorithm and its elaborations exploit linearity. They may be applied to least squares estimation problems having nonlinear constraints, by the formal application of a Picard itera-

[6] However, the observational error variance should not be kept artificially large, by failing to filter the observations for variability owing to dynamical processes not included in the ocean model. Moreover, assimilating severely aliased data can never be justified.

tion scheme such as tangent linearization (Bennett 2002). Bennett et al. (2006) make 4–6 such outer iterations, for a total of 120–180 inner iterations or data space search steps. There are, as a rule, no convergence proofs for the outer iterations applied to realistic forward dynamics and adjoint dynamics. Such proofs would also establish the elusive classical well posedness of the forward equations of fluid dynamics in space. There are, however, empirically successful accelerators for convergence of the outer iterations (Ngodock et al. 2000).

The full development of the IOM minimization algorithm may be found in Chua and Bennett (2001), and the reader must be referred there for further detail. The next section offers a brief sketch of the IOM algorithm.

3. The IOM algorithm—A sketch

As stated in the previous section, the IOM algorithm is the indirect representer algorithm (Egbert et al. 1994), or dual-adjoint algorithm (Amodei 1995; Courtier 1997), applied iteratively to nonlinear dynamics and observing systems. The algorithm is now well known to meteorologists either as an "observation space search" or, less clearly, as a "physical space search." Indeed, it is nearly three decades since representers were brilliantly exploited for synoptic optimal interpolation (today's 3DVAR) by Wahba and Wendelberger (1980). The mathematics for dynamically constrained representers have now been given in detail in many places (e.g., Courtier 1997 or Bennett 2002). The details are extensive for even a shallow-water model; in view of the limited space available here, only the following sketch can be offered.

Consider as an "ocean model" the ordinary differential equation (ODE)

$$\frac{d}{dt}u(t) = F(t) \tag{1}$$

over the "domain" $0 < t < T$, subject to $u(0) = I$, where $F(t)$ is a specified function and I is a specified constant. There is a unique solution $u = u_F(t)$ over the domain. Now introduce M measurements of the "real ocean state" $u(t)$: $d_1, d_2, \ldots, d_M$, taken at times t_m, $1 \le m \le M$, where $0 < t_m < T$. In general, the unique solution of the ODE does not agree with the data, that is, $u_F(t_m) \neq d_m$ for some m. It must be assumed that there are errors in the forcing $F(t)$, in the initial value I, and in the data $d_1, \ldots, d_M$. A simple, weighted least squares estimator for the errors is

$$J[u] = V_F^{-1}\int_0^T\left(\frac{du}{dt} - F\right)^2 dt + V_I^{-1}(u(0) - I)^2 + V_d^{-1}\sum_{m=1}^{M}(u(t_m) - d_m)^2, \tag{2}$$

where the specified constants V_F, V_I, V_d are error variances. Then $J[\hat{u}]$ is least if $\hat{u}(t)$ and the auxiliary variable $\lambda(t)$ together satisfy the Euler–Lagrange equations

$$\frac{d\hat{u}}{dt} = F + V_F\lambda, \tag{3}$$

$$-\frac{d\lambda}{dt} = V_d^{-1}\sum_{m=1}^{M}(d_m - \hat{u}(t_m))\delta(t - t_m), \tag{4}$$

subject to $\lambda(T) = 0$ and $\hat{u}(0) = I + V_I\lambda(0)$. Equation (4) is the "adjoint" of Eq. (3) and $\lambda(t)$ is the weighted residual or adjoint variable. Clearly, $\hat{u}$ and λ are linear combinations of the responses of the coupled system (3), (4) to the individual delta function impulses in (4); that is, they are linear combinations of representer functions and their adjoint variables, respectively. The M impulse coefficients $V_d^{-1}(d_m - \hat{u}(t_m))$, $(m = 1, \ldots, M)$ appearing on the right-hand side of (4) constitute the "coupling vector." Owing to the presence of the $\hat{u}(t_m)$ (the M unknown measurements of $\hat{u}(t)$), the coupling vector is unknown and Eq. (4) cannot be integrated backward from $t = T$ before integrating (3) forward from $t = 0$. Rather, the EL equations form a two-point boundary value problem in the time interval $0 < t < T$. This highly complicated implicit problem is resolvable, that is, the solution can be calculated in a finite number of explicit steps. It has been shown (e.g., Bennett 2002) that the components of the unknown coupling vector satisfy an $M \times M$ system of linear equations, having a coefficient matrix (the representer matrix) that depends linearly upon the specified error variances V_F, V_I, V_d. It also depends upon the dynamics (just d/dt here), the domain (just $0 < t < T$ here), and the observational array (just the measurement times t_m here). The details may be found in the appendix. Once the linear system is solved for the coupling vector, the adjoint variable λ may be found with one backward integration, and then the optimal estimate $\hat{u}(t)$, $(0 < t < T)$ follows with one forward integration. Explicit construction of the symmetric, positive-definite representer matrix requires M pairs of integrations, which may be carried out simultaneously given M processors (Bennett and Baugh 1992). The final pair of integrations is a standard "open loop control" (e.g., Wunsch 1996).

Alternatively (Egbert et al. 1994; Amodei 1995), the coupling vector may be found iteratively without explicit construction of the representer matrix. It suffices to know a search direction in the M-dimensional data subspace. The direction is given by the action of the matrix upon an arbitrary M-component vector; the action may be computed with one pair of integrations. The details may also be found in the appendix. An accurate approximation to the coupling vector is usually obtained with about $M/100$ integration pairs. The indirect algorithm is no more than an iterative application of open loop control (Wunsch 1996).

Note that the search for the best-fitting continuous function of time $\hat{u}(t)$ over $0 < t < T$ has been *exactly* reduced to a search for the M real components of the coupling vector. This massive first preconditioning of the search in function space or state space has been affected by restricting the search to an M-dimensional subspace of observable degrees of freedom, the "data subspace." A vast and unobservable null subspace has been suppressed (Wahba and Wendelberger 1980; Bennett 1985, 1992). The state $u(t)$ is unobservable if $u(t_m) = 0$, for $1 \leq m \leq M$. The search in the data subspace may be further preconditioned by explicit construction of economical approximations to the representer matrix. The IOM includes the infrastructure for controlling this so-called inner iteration in the data subspace. It also includes a suite of preconditioners; the user makes a selection via the GUI. The overall preconditioning of this fitting algorithm is unequalled by any other feasible approach (Zaron 2006). Finally, The IOM calculates the leading eigenvectors of the representer matrix indirectly, using the Lanczos Connection (Golub and Van Loan 1989, p. 522). These eigenvectors determine the most stably observable states. That is, they yield a quantitative assessment of the efficiency of the observing array or system.

In realistic inverse models, point measurements may be replaced with spatial or temporal averages. Also, the constant variances V_I, V_F may be replaced with inhomogeneous covariances over space or space–time, respectively, according to our prior assumptions about the errors in the forward ocean model. Thus, products with the adjoint variable λ must be replaced with (fast) convolutions over space and space–time. As discussed in section 4, the IOM includes fast convolvers for certain commonplace correlations. The IOM can estimate, if so required, all posterior error covariances for all quantities with adequate accuracy using offline Monte Carlo methods.

We wish to emphasize that the IOM not only provides a dynamically constrained estimate of ocean state, it also provides an objective test of the hypothesized error statistics (exemplified here by V_F, etc). Indeed, the quantity $J[\hat{u}]$, which is the minimal value of the weighted least squares estimator defined in Eq. (2), is the test statistic for the hypothesis, and is in fact the standard random variable χ^2_M if the hypothesis is correct.

As a final remark, note that the above sketch makes no distinction between errors in the "ocean dynamics" d/dt and in the specified forcing $F(t)$. These errors may be distinguished by the introduction of additional state variables (Jacobs and Ngodock 2003), but the additional effort is only justifiable if the user is confident that the two kinds of error have different statistics.

The preceding sketch of the IOM algorithm is intentionally simple, and all modelers will realize that much detail is involved if the model is realistic. The code generating capability of the IOM absolutely minimizes the coding that must be carried out by a user, over and above the user's forward model, while the IOM GUI prompts the user for all the scientific and computational information required for the user's choice of partial or complete inversion. The user may choose, in the simplest extreme, just a forward integration and comparison with the user-supplied data, or in the other extreme an inversion with all diagnostic and posterior statistics.

4. What the user must provide

The following is a brief description of the preparations that an IOM user must make:

1) *The IOM is not a black box.* The user must have a complete command of the scientific principles involved in formulating a quadratic penalty functional for the estimation of ocean state. The estimator is maximum likelihood for the errors in the ocean dynamics, the external forcings, and the observations. The errors are assumed to be jointly normally distributed. The user must also be completely familiar with the mathematics of oceanic and atmospheric state estimation in general, and the IOM algorithm in particular. The articles by Courtier (1997) and Chua and Bennett (2001) outline the scientific and mathematical concepts; pedagogical presentations may be found in the graduate text by Bennett (2002) and in the interactive online instruction facility on the IOM Web site (http://iom.asu.edu). The IOM includes a suite of tutorial models.
2) The user must of course provide an ocean model. In our experience, these consist of a run script that starts an executable. The latter is compiled source code that drives a subroutine, which in turn advances the ocean state by one time step. The driver calls the subroutine for a desired number of time steps. The driver also effects input and output, usually in the NetCDF format required by the IOM. The user of the IOM must modify this driver to include a flag that selects the forcing, initial conditions, and boundary conditions according to the stage of the IOM algorithm. Examples of modified

drivers may be found in the IOM user manual (Bennett et al. 2007). An essential feature of the IOM is that the user does not have to reorder the model's independent and dependent variables (i.e., the order of the loops over the grids, e.g.); the IOM imposes no protocol for loop ordering. The IOM GUI prompts the user to provide the details of the model's numerical structure (such as the loop ordering), and the IOM then automatically generates a suite of inversion tools that conform to that numerical structure.

3) The user must provide a subroutine that interpolates the ocean state variables, from the finite-difference grid or basis functions of the model, to the longitude, latitude, and depth of an arbitrary point in the ocean at an arbitrary time. It is invariably the case that the ocean modeler has already developed just such an interpolation routine, for display and analysis of model output and for comparison with ocean observations.

4) If the ocean model is nonlinear, then the user must also provide a tangent-linear (TL) model for a *finite-amplitude* ocean state u that differs by an infinitesimal δu from a reference state u_{ref}: $u \equiv u_{\mathrm{ref}} + \delta u$. Such a TL is illustrated by $u^2 \to u_{\mathrm{ref}}^2 + 2u_{\mathrm{ref}}\delta u$. That is, only terms linear in the infinitesimal perturbation field are retained. The Parametric FORTRAN compiler included with the IOM can act as an automatic TL generator (Erwig and Fu 2004; Erwig et al. 2006; Fu 2006), provided there is neither "wetting" nor "drying" (such as in an intertidal zone). The user may employ the compiler, offline from the IOM, to convert the single time-step subroutine to its TL equivalent. The user need only insert a few very simple Parametric FORTRAN commands into the user's FORTRAN 90 subroutine, thereby converting the code to Parametric FORTRAN. The compiler for Parametric FORTRAN then generates the FORTRAN 90 subroutine for the TL of the original subroutine. The user must "wrap" the generated TL subroutine in a driver, and introduce a flag as discussed above. The driver employs virtually the same NetCDF I/O as does the original ocean model. Details may be found in the IOM user manual (Bennett et al. 2007).

   It may be noted that the compiler finds the derivatives of common smooth functions (rational, trigonometric, exponential, hyperbolic, etc.) in a lookup table. This table is readily extended to less common smooth functions (inverse trig, error, gamma, etc.), as required.

5) If the ocean model is nonlinear, then the user must also provide a tangent-linear representer model for an *infinitesimal* ocean state that is a perturbation about a reference state. That is, $u^2 \to 2u_{\mathrm{ref}}\delta u$. This model is known as *the* tangent-linear model to practitioners of incremental 4DVAR (e.g., Courtier 1997). Alternatively, the IOM's Parametric FORTRAN compiler can generate the required subroutine. The code for the representer model needs no flags: it does occur at several stages in the IOM algorithm, but always with the same functionality.

6) The user must provide the adjoint of the TL of the forward model. The representer model has the same adjoint as the TL of the forward model. The Parametric FORTRAN compiler can also generate most of the one-step adjoint subroutine, following the line-by-line approach of Giering and Kaminski (1998). The adjoints of space-loop interiors for finite-difference models are correct to machine precision, but the user may need to correct the adjoint code at the boundaries. These corrections require a sound understanding of adjoint principles, but the IOM does eliminate the potentially massive and tedious effort to construct the adjoints of the loop interiors. The "manually generated" line-by-line adjoint code for the relevant subroutine in ROMS, and in the Naval Research Laboratory Coastal Ocean Model (NCOM; Barron et al. 2006), has been succesfully regenerated using Parametric FORTRAN together with mild corrections (H. Ngodock 2007, personal communication).

   The one-step adjoint subroutine, however it is obtained, must be "wrapped" in a driver but again no flagging is needed.

   The IOM's adjoint generator is in continuous development. Examples may be found in the IOM user manual, demonstrating the automatic generation of the adjoints of time loops as well as space loops. A second adjoint generator is under development; unlike the line-by-line adjoint of Giering and Kaminski (1998), it generates code that resembles computational fluid dynamics.

7) If the user hypothesizes that the initial errors are multivariate, that is, they are jointly covarying (e.g., are geostrophically balanced), then the user must provide subroutines for the initial multivariate covariance operators. In this case, the IOM GUI prompts the user to provide the system paths to the codes for these initial operators, and then implements the codes in the IOM algorithm. The user must hypothesize that the dynamical errors are univariate, but these errors may be autocovarying in space or time or both. The IOM can automatically provide, if requested, fast convolutions for bell-shaped correlations $[\exp(-x^2/X^2)]$ for all spatial er-

ror dependencies, and fast convolutions for Markovian correlations [$\exp(-|t|/s)$] for all temporal error dependencies. The GUI prompts the user for the parameters (variances, and decorrelation scales X, s, etc.) in all these various convolutions.

8) The user must of course provide some ocean data. The IOM requires these to be in the form of single, ordered file of real numbers. If these data (which may be of mixed type: velocity, salinity, continuously varying climate indices, etc.) have been downloaded from one or more of the Web sites for many of the "great ocean observing programs" (Argo, Global Drifter Program, NCEP Global SST Analysis, *Jason-1*, TAO/TRITON), the user merely informs the IOM Data Ingest System (IDIS). This subsystem supplies and applies the appropriate software filters, which convert the many Web sites' formats to the IOM NetCDF format. Then IDIS merges the many downloads into a single IOM file.
9) Finally, the user must provide separate executables that encode the mathematical description of the ocean observing sensor of interest. In the simplest case, the sensor is no more than that for a single, scalar, ocean state variable at a single point in space (expressed as a longitude, a latitude, and a depth), and the sensor being active at just a single time. More complex linear sensors involve integrals over spatial or temporal domains, and may involve linear combinations of state variables. In mathematical terminology, the description is the *kernel* of a linear functional. Anticipating that the user is likely to be interested in data from the "great programs," IDIS contains the various kernels for each of the abovenamed observing systems. If the user introduces a sensor for which the mathematical description is nonlinear, the two required tangent linearizations may be automatically generated using the Parametric FORTRAN compiler. If the ocean model employs thermodynamic coordinates, then all moored subsurface observations are nonlinear. If the ocean model is Eulerian, then all Lagrangian observations are nonlinear. For examples and discussions, see the IOM user manual.

5. What the IOM will do for the user

Given that the user must for now provide the adjoint model, or at least "tweak" the adjoint generated by the IOM, the user might well wonder what more needs to be done in order to carry out W4DVAR. In fact, a massive amount of software infrastructure is needed. Some idea of the scope and complexity is illustrated by the flow chart in Fig. 1, for the IOM applied to a quasigeostrophic model (Bennett 2002).

1) The infrastructure must accept the forward, TL, representer, and adjoint models, plus the parameters for the hypothesized error covariances, plus the observations and their associated kernels, and then compute a W4DVAR analysis with the full set of diagnostics. The IOM provides the entire model-independent infrastructure, and automatically generates a vast amount of model-dependent software, from the Parametric FORTRAN templates and from the parameters entered into the GUI by the user.
2) For example, the IOM combines the user-supplied, model-dependent interpolation scheme and a user-supplied, observing sensor–dependent kernel into a tool that extracts the analogous measurement from the model representation of the ocean state. The mathematics of this "cleavage plane" between the model and the observations is hardly profound, but is of critical importance to the modularity of the IOM. There is an extensive discussion in the IOM user manual, for finite-difference models and basis-function models, with both simple and complex sensors.
3) The IOM automatically generates the "comb" for the observing array. The comb is a linear combination of discrete operators acting on discrete unit-impulse functions [see the right-hand side of Eq. (4) for the continuous analog]; it may be regarded as the adjoint of the observing system, since the comb forces the adjoint model at several stages of the IOM. Great care is needed here, in order to maintain consistency with the user's numerical approximation to the penalty functional or estimator. At issue are the volume elements and time elements of integration, or their spectral equivalents, which the user may or may not choose to include in the estimator. It is critical that the comb be consistent, so that the minimal value of the estimator is in fact the test statistic for the hypothesis test that is made when estimating ocean state. The scaling of the estimator is discussed in detail in the IOM user manual.
4) As already mentioned, the IOM generates univariate covariance operators consistent with the numerical structure of the user's model.
5) Also, IDIS makes virtually effortless the introduction of observations from the great ocean observing programs.
6) The IOM includes the entire model-independent infrastructure for effecting the data space search that is fundamental to the IOM or dual-adjoint algorithm. The inner loop that affects this search is illustrated in Fig. 2.

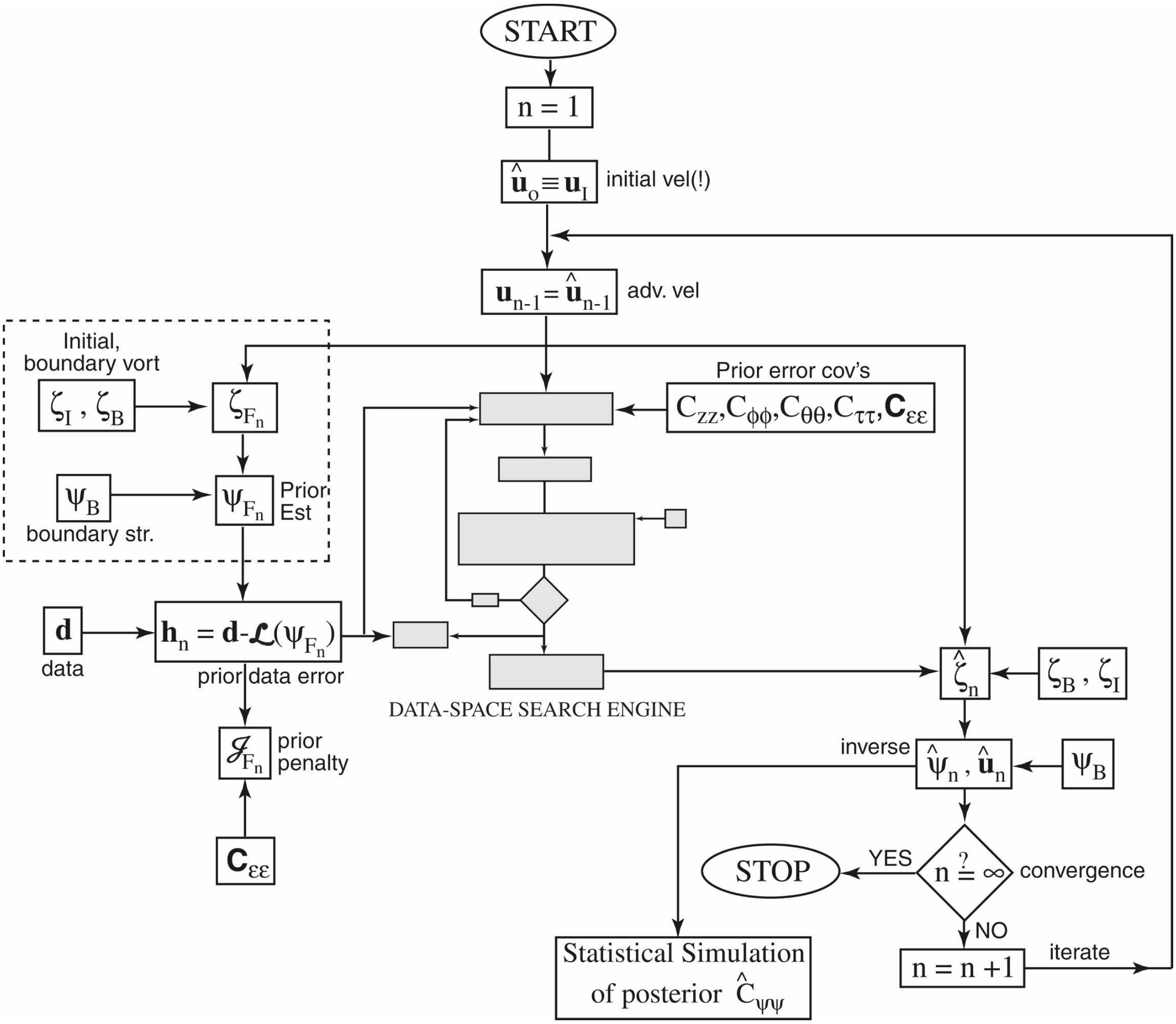

FIG. 1. Flowchart for the IOM algorithm, applied to a quasigeostrophic model (after Bennett 2002). The notation is as follows: ψ, $\mathbf{u}$, ζ are, respectively, the streamfunction, velocity, and total vorticity; $\mathbf{d}$ is the M vector of data; $\mathcal{L}$ is the M vector of linear measurement functionals; F is the prior for the forcing in the vorticity equation; $\mathbf{C}_{ZZ}$, etc. are the prior or hypothesized covariances for the error Z in the vorticity equation, etc.; $\mathbf{C}_{\varepsilon\varepsilon}$ is the prior for the measurement error covariance; $\mathcal{J}$ is a penalty, n is the outer iteration index, and the hat symbol denotes the optimal value. The hatched area indicates the data space search (see Fig. 2).

The user merely makes a selection, via the GUI, from a suite of preconditioners.

7) Once the user has chosen a partial or full W4DVAR analysis, the IOM prompts the user for all the required information, which consists of paths to files and values of parameters. The IOM then generates an Extensible Markup Language (XML) formatted input file for the master Perl script that runs the stages of the IOM algorithm. It should be noted that each stage is coded as a separate executable rather than a subroutine call. The IOM creates a logfile for each experiment. The user has access to the run script and to all the automatically generated FORTRAN 90 codes, should they be of interest.
8) The IOM can provide syntheses of pseudorandom vectors, and pseudorandom fields over space and time, having means and covariances prescribed by the user. At present, the spatial covariances must be bell shaped and the temporal covariances must be Markovian for computational efficiency, and the user merely prescribes parameters via the GUI. These pseudorandom vectors and fields are used by the IOM for Monte Carlo estimates of any pos-

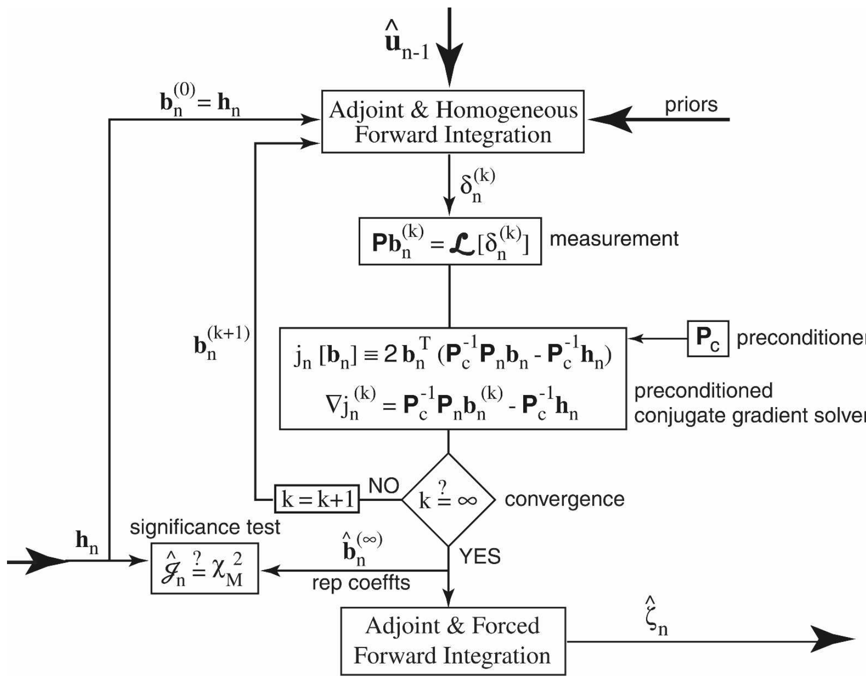

FIG. 2. The inner loop in the IOM algorithm that effects the data space search, after Bennett (2002). The background for the integrations is the last outer iterate $\hat{\mathbf{u}}_{n-1}$ for the optimal velocity field; δ denotes the impulses that, in linear combinations, force the adjoint equation; $\mathbf{P}$ is the unknown representer matrix stabilized by addition of the known measurement error covariance matrix $\mathbf{C}_{\epsilon\epsilon}$; $\mathbf{b}_n^{(k)}$ is the kth inner iterate for the sought-after *nth* coupling vector. The user chooses a preconditioner $\mathbf{P}_c$, via the GUI. Finally, χ_M^2 is the chi-squared random variable with M degrees of freedom. If the hypothesized error covariances are correct, then the expected value of $\hat{\jmath}$ is χ_M^2.

teriors requested by the user, and of other diagnostics needed in the assessment of observing systems or arrays.

9) The IOM can verify adjoint symmetry and perform other algorithmic consistency checks.
10) The IOM can compute sensitivities for linear (or tangent linearized) models and linear (or tangent linearized) quantifiers of ocean state, with respect to dynamical, initial, or boundary perturbations.

6. The IOM Web site

An IOM Web site has been developed (available online at http://iom.asu.edu). The site has two main purposes. The first is to facilitate communication among the IOM community, including distribution of the IOM software and documentation. The second is to provide user-friendly, interactive instructional material with more detail than is feasible in a journal paper or text.

a. *Facilitation of communication*

The IOM Web site has several features to facilitate communication between researchers. The latest version of the software can be downloaded, along with a technical document that details full testing of the IOM with simple one- and two-dimensional models, as well as various IOM technical documents such as the user manual. The Parametric FORTRAN compiler can also be downloaded, along with its own user manual. Also maintained on the site is a full list of references related to the project (both the algorithmic and application oriented papers). This list can be updated by all IOM researchers and viewed by anyone who accesses the site. There is also a repository for written material related to the IOM. Material can be posted securely so that only IOM researchers can access it, or may be made available to the general public. In addition, several discussion boards have been implemented.

b. *Instructional content*

The instructional content is based on the development from Bennett (2002), but is not simply a restatement of that material. While the material assumes that the user has access to that text as a reference, the site is intended to provide details omitted from the book because of space restrictions, solutions to some of the exercises, and additional detailed development for discrete, multidimensional, and nonlinear extensions of the theory. Throughout the instructional material, there are opportunities for interaction with the material. For example, the user can see an animation of adjoint representers and representers (with or without space and/or time convolutions). In addition, the user can effortlessly toggle between the continuous and discrete equation developments, and therefore see their similarities and their differences. Two models serve to illustrate key concepts of the assimilation exercise, each model being examined in both continuous and discrete form:

1) one-dimensional, linear advection equation, and
2) one-dimensional, nonlinear Korteweg–de Vries equation.

7. Summary

The IOM exploits a natural modularization of the W4DVAR algorithm into components that depend only upon models, components that depend only upon observing systems, and components (such as algebraic system solvers) that depend upon neither. Automatic generation of customized code saves the user from a very heavy burden of mathematical derivation and coding. Indeed, the creation of program templates in Parametric FORTRAN enables the precise standardization of algorithms for W4DVAR by inverse theorists, and also the precise standardization of observing algorithms by instrument builders. The user retains the complete freedom (and now has the time) to vary the model, as the IOM can build a new W4DVAR system for a revised model or a new observing system for the cost only of entering the modified information.

New users of the IOM should be encouraged. The IOM will evolve in response to users' experiences. The system design, in terms of separate executables, anticipates a more advanced design as a *framework* such as the Earth System Modeling Framework (http://www.esmf.ucar.edu/).

Acknowledgments. This work is supported by a National Science Foundation (Information Technology Research) Collaborative Research Medium Award, with Subawards OCE–121542 to OSU and OCE-121315 to ASU, and by the Office of Naval Research with Grant N000140410439 to ASU.

APPENDIX

The Representer Solution

The unique solution of Eqs. (3) and (4) in section 3 is given explicitly by

$$\hat{u}(t) = u_F(t) + \sum_{m=1}^{M} \beta_m r_m(t), \tag{A1}$$

where the "first guess" or "prior" solution u_F satisfies the forward model (1); that is,

$$\frac{d}{dt} u_F(t) = F(t), \tag{A2}$$

over the domain $0 < t < T$, subject to $u_F(0) = I$. The M representers r_m and their adjoints ρ_m satisfy

$$-\frac{d\rho_m}{dt} = V_d^{-1} \delta(t - t_m), \tag{A3}$$

subject to $\rho_m(T) = 0$, and

$$\frac{dr_m}{dt} = V_F \rho_m, \tag{A4}$$

subject to $r_m(0) = V_I \rho_m(0)$. Note that each ρ_m may be found independently of finding the corresponding r_m. The M representer coefficients β_m satisfy the $M \times M$ linear system

$$\sum_{m=1}^{M} (R_{nm} + V_d \delta_{nm}) \beta_m = d_n - u_F(t_n), \tag{A5}$$

where δ_{nm} denotes the components of the unit matrix, and the symmetric positive-definite representer matrix component R_{nm} is given by $R_{nm} = r_n(t_m) = r_m(t_n)$ (see, e.g., Bennett 2002). It is easily shown that the coupling $V_d^{-1}(d_m - \hat{u}(t_m))$ in (2) is in fact just β_m. In compact notation, the linear system is

$$(\mathbf{R} + V_d \mathbf{I})\boldsymbol{\beta} = \mathbf{d} - \mathbf{u}_F, \tag{A6}$$

which is the extremal condition for the quadratic penalty

$$J[\mathbf{b}] = \frac{1}{2} \mathbf{b}^{\mathrm{T}} (\mathbf{R} + \mathbf{V}_d \mathbf{I})[\mathbf{b}] - \mathbf{b}^{\mathrm{T}} (\mathbf{d} - \mathbf{u}_F). \tag{A7}$$

Thus $\boldsymbol{\beta}$ may be found by searching for the minimum of $J[\mathbf{b}]$. It suffices to know $(\mathbf{R} + V_d \mathbf{I})\mathbf{b}$ for any $\mathbf{b}$. To this end, replace the coupling vector $V_d^{-1}(\mathbf{d} - \hat{\mathbf{u}})$ in (4) with $\mathbf{b}$; solve (4) for λ and insert the solution into (3); solve (3) for u, with F and I set to zero; finally evaluate the solution of (3) at each t_m, $1 \leq m \leq M$. The result is the required new search direction, in the data subspace, for the coupling vector. The search is accelerated (preconditioned) with an inverse of an approximation to $\mathbf{R}$. The IOM provides a suite of user-selectable preconditioners.

REFERENCES

Amodei, L., 1995: Solution approcheé pour un problème d'assimilation de donneés meteorologiques avec prise en compte de l'erreur de modèlè. *C. R. Acad. Sci. II,* **321,** 1087–1094.

Barron, C. N., A. B. Kara, P. J. Martin, R. C. Rhodes, and L. F. Smedstad, 2006: Formulation, implementation and examination of vertical coordinate choices in the global Navy Coastal Ocean Model (NCOM). *Ocean Modell.,* **11,** 347–375.

Bennett, A. F., 1985: Array design by inverse methods. *Prog. Oceanogr.,* **15,** 129–156.

——, 1992: *Inverse Methods in Physical Oceanography.* Cambridge University Press, 347 pp.

——, 2002: *Inverse Modeling of the Ocean and Atmosphere.* Cambridge University Press, 234 pp.

——, and P. C. McIntosh, 1982: Open ocean modeling as an inverse problem: Tidal theory. *J. Phys. Oceanogr.,* **12,** 1004–1018.

——, and W. P. Budgell, 1987: Ocean data assimilation and the Kalman filter: Spatial regularity. *J. Phys. Oceanogr.,* **17,** 1583–1601.

——, and ——, 1989: The Kalman smoother for a linear quasi–geostrophic model of ocean circulation. *Dyn. Atmos. Oceans,* **13,** 219–267.

——, and R. N. Miller, 1991: Weighting initial conditions in variational assimilation schemes. *Mon. Wea. Rev.,* **119,** 1098–1102.

——, and J. R. Baugh, 1992: A parallel algorithm for variational assimilation in oceanography and meteorology. *J. Atmos. Oceanic Technol.,* **9,** 426–433.

——, and M. A. Thorburn, 1992: The generalized inverse of a nonlinear quasigeostrophic ocean circulation model. *J. Phys. Oceanogr.,* **22,** 213–230.

——, B. S. Chua, and L. M. Leslie, 1996: Generalized inversion of a global numerical weather prediction model. *Meteor. Atmos. Phys.,* **60,** 165–178.

——, ——, and ——, 1997: Generalized inversion of a global numerical weather prediction model. II: Analysis and implementation. *Meteor. Atmos. Phys.,* **62,** 129–140.

——, ——, D. E. Harrison, and M. J. McPhaden, 1998: Generalized inversion of Tropical Atmosphere–Ocean data and a coupled model of the tropical Pacific. *J. Climate,* **11,** 1768–1792.

——, ——, ——, and ——, 2000: Generalized inversion of Tropical Atmosphere–Ocean (TAO) data and a coupled model of the tropical Pacific. Part II: The 1995/96 La Niña and 1997/98 El Niño. *J. Climate,* **13,** 2770–2785.

——, ——, H.-E. Ngodock, D. E. Harrison, and M. J. McPhaden, 2006: Generalized inversion of the Gent–Cane model of the tropical Pacific with Tropical Atmosphere–Ocean (TAO) data. *J. Mar. Res.,* **64,** 1–42.

——, ——, B. L. Pflaum, M. Erwig, Z. Fu, R. D. Loft, and J. C. Muccino, 2007: Inverse Ocean Modeling System user manual. IOM Tech. Doc. 6, 221 pp. [Available online at ftp://ftp.coas.oregonstate.edu/dist/chua/iom/tech_doc/IOM_TD6.pdf.]

Brunk, H. D., 1965: *An Introduction to Mathematical Statistics.* 2nd ed. Blaisdell, 429 pp.

Chua, B. S., and A. F. Bennett, 2001: An inverse ocean modeling system. *Ocean Modell.,* **3,** 137–165.

Courtier, P., 1997: Dual formulation of four-dimensional assimilation. *Quart. J. Roy. Meteor. Soc.,* **123,** 2449–2461.

——, and O. Talagrand, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation. II: Numerical results. *Quart. J. Roy. Meteor. Soc.,* **113,** 1329–1347.

Daley, R. A., 1991: *Atmospheric Data Analysis.* Cambridge University Press, 457 pp.

Duncan, W. J., 1944: Some devices for the solution of large sets of simultaneous equations (with an appendix on the reciprocation of partitioned matrices). *London Edinburgh Dublin Philos. Mag. J. Sci.,* **35,** 660.

Egbert, G. D., A. F. Bennett, and M. G. G. Foreman, 1994: TOPEX/POSEIDON tides estimated using a global inverse model. *J. Geophys. Res.,* **99,** 24 821–24 852.

Erwig, M., and Z. Fu, 2004: Parametric Fortran—A program generator for generic Fortran programming. *Proc. Sixth Int. Symp. on Practical Aspects of Declarative Languages,* Charleston, SC, COMPULOG Americas, 209–223.

——, ——, and B. Pflaum, 2006: Generic programming in Fortran. *Proc. ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation,* Charleston, SC, Association for Computing Machinery, 130–139.

——, ——, and ——, 2007: Parametric Fortran: Program generation in scientific computing. *J. Software Maint. Evol.,* **19,** 155–182.

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.,* **99,** 10 143–10 162.

——, 2006: *Data Assimilation: The Ensemble Kalman Filter.* Springer-Verlag, 280 pp.

Fu, Z., 2006: Automatic program generation for scientific applications. Ph.D. dissertation, Oregon State University, 200 pp.

Fukumori, I., and P. Malanotte-Rizzoli, 1995: An approximate Kalman filter for ocean data assimilation: An example with an idealized Gulf Stream model. *J. Geophys. Res.,* **100,** 6777–6793.

Gelb, A., Ed., 1974: *Applied Optimal Estimation.* MIT Press, 374 pp.

Giering, R., and T. Kaminski, 1998: Recipes for adjoint code construction. *ACM Trans. Math. Software,* **24,** 437–474.

Golub, G. H., and C. F. van Loan, 1989: *Matrix Computations.* The Johns Hopkins University Press, 642 pp.

Jacobs, G. A., and H. Ngodock, 2003: The maintenance of conservative physical laws within data assimilation systems. *Mon. Wea. Rev.,* **131,** 2595–2607.

Le Dimet, F., H. Ngodock, B. Luong, and J. Verron, 1997: Sensitivity analysis in variational data assimilation. *J. Meteor. Soc. Japan,* **75,** 245–255.

Lewis, J. M., S. Lakshmivarahan, and S. Dhall, 2006: *Dynamic Data Assimilation: A Least Squares Approach.* Cambridge University Press, 676 pp.

McIntosh, P. C., and A. F. Bennett, 1984: Open ocean modeling as an inverse problem: M_2 tides in Bass Strait. *J. Phys. Oceanogr.,* **14,** 601–614.

Muccino, J. C., and H. Luo, 2005: Picard iterations for a finite element shallow water equation model. *Ocean Modell.,* **10,** 316–341.

——, and Coauthors, 2008: The Inverse Ocean Modeling system. Part II: Applications. *J. Atmos. Oceanic Technol.,* **25,** 1623–1637.

Ngodock, H. E., B. S. Chua, and A. F. Bennett, 2000: Generalized inverse of a reduced gravity primitive equation model and Tropical Atmosphere–Ocean data. *Mon. Wea. Rev.,* **128,** 1757–1777.

——, G. Jacobs, and M. Chen, 2006: The representer method, the ensemble Kalman filter and the ensemble Kalman smoother: A comparison study using a nonlinear reduced gravity ocean model. *Ocean Modell.,* **12,** 378–400.

Riedel, K. S., 1991: A Sherman Morrison Woodbury identity for rank augmenting matrices with application to centering. *SIAM J. Matrix Anal. Appl.,* **13,** 659–662.

Rosmond, T., and L. Xu, 2006: Development of NAVDAS-AR: Nonlinear formulation and outer loop tests. *Tellus,* **58A,** 45–58.

Talagrand, O., 2008: *Data Assimilation in Meteorology and Oceanography.* Academic Press.

——, and P. Courtier, 1987: Variational assimilation of meteorological observations with the adjoint vorticity equation. I. Theory. *Quart. J. Roy. Meteor. Soc.,* **113,** 1311–1328.

Wahba, G., and J. Wendelberger, 1980: Some new mathematical methods for variational objective analysis using splines and cross validation. *Mon. Wea. Rev.,* **108,** 1122–1143.

Wunsch, C., 1996: *The Ocean Circulation Inverse Problem.* Cambridge University Press, 442 pp.

Xu, L., T. Rosmond, and R. Daley, 2005: Development of NAVDAS-AR: Formulation and initial tests of the linear problem. *Tellus,* **57A,** 546–559.

——, R. Langland, N. Baker, and T. Rosmond, 2006: Development of the NRL 4DVar data assimilation adjoint system. Geophysical Research Abstracts, Vol. 8, Abstract 08773. [Available online at http://www.cosis.net/abstracts/EGU06/08773/EGU06-J-08773.pdf.]

Zahel, W., 1991: Modeling ocean tides with and without assimilating data. *J. Geophys. Res.,* **96,** 20 379–20 391.

Zaron, E. D., 2006: A comparison of data assimilation methods using a planetary geostrophic model. *Mon. Wea. Rev.,* **134,** 1316–1328.

——, and G. D. Egbert, 2006: Verification studies for a *z* coordinate primitive–equation model: Tidal inversions at a mid-ocean ridge. *Ocean Modell.,* **14,** 257–278.