Progress in Oceanography 91 (2011) 34–49

Contents lists available at ScienceDirect

Progress in Oceanography

journal homepage: www.elsevier.com/locate/pocean

The Regional Ocean Modeling System (ROMS) 4-dimensional variational data assimilation systems
Part I – System overview and formulation

Andrew M. Moore[a,*], Hernan G. Arango[b], Gregoire Broquet[c], Brian S. Powell[d], Anthony T. Weaver[e], Javier Zavala-Garay[b]

[a] Department of Ocean Sciences, University of California, 1156 High Street, Santa Cruz, CA 95064, United States
[b] Institute of Marine and Coastal Sciences, Rutgers University, 71 Dudley Road, New Brunswick, NJ 08901-8521, United States
[c] Laboratoire des Sciences du Climat et de l'Environnement, CEA-Orme des Merisiers, F-91191 GIF-SUR-YVETTE CEDEX, France
[d] Department of Oceanography, University of Hawai'i at Manoa, Honolulu, HI 96822, United States
[e] Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, Toulouse, France

ARTICLE INFO

Article history:
Received 14 November 2010
Received in revised form 20 May 2011
Accepted 31 May 2011
Available online 7 June 2011

ABSTRACT

The Regional Ocean Modeling System (ROMS) is one of the few community ocean general circulation models for which a 4-dimensional variational data assimilation (4D-Var) capability has been developed. The ROMS 4D-Var capability is unique in that three variants of 4D-Var are supported: a primal formulation of incremental strong constraint 4D-Var (I4D-Var), a dual formulation based on a physical-space statistical analysis system (4D-PSAS), and a dual formulation representer-based variant of 4D-Var (R4D-Var). In each case, ROMS is used in conjunction with available observations to identify a best estimate of the ocean circulation based on a set of *a priori* hypotheses about errors in the initial conditions, boundary conditions, surface forcing, and errors in the model in the case of 4D-PSAS and R4D-Var. In the primal formulation of I4D-Var the search for the best circulation estimate is performed in the full space of the model control vector, while for the dual formulations of 4D-PSAS and R4D-Var only the sub-space of linear functions of the model state vector spanned by the observations (i.e. the dual space) is searched. In oceanographic applications, the number of observations is typically much less than the dimension of the model control vector, so there are clear advantages to limiting the search to the space spanned by the observations. In the case of 4D-PSAS and R4D-Var, the strong constraint assumption (i.e. that the model is error free) can be relaxed leading to the so-called weak constraint formulation. This paper describes the three aforementioned variants of 4D-Var as they are implemented in ROMS. Critical components that are common to each approach are conjugate gradient descent, preconditioning, and error covariance models, which are also described. Finally, several powerful 4D-Var diagnostic tools are discussed, namely computation of *posterior* errors, eigenvector analysis of the *posterior* error covariance, observation impact, and observation sensitivity.

1. Introduction

Data assimilation is used in meteorology and oceanography to combine observations and numerical models to obtain the best linear unbiased estimate (BLUE) of the circulation, and for other related applications, such as parameter estimation. The circulation estimates are usually defined as those which minimize, in a least-squares sense, the difference between the model state and the observations and a background (or *prior*) subject to *a priori* hypotheses about errors in the background, model, and observations. If the errors are truly Gaussian, the BLUE corresponds to a maximum likelihood estimate. Excellent reviews and seminal texts on data assimilation include Bengtsson et al. (1981), Tarantola (1987), Daley (1991), Ghil and Malanotte-Rizzoli (1991), Bennett (1992, 2002) and Wunsch (1996, 2006).

A common method used to identify the best least-squares estimate is the calculus of variations, and some of the first notable applications in meteorology and oceanography are those of Lewis and Derber (1985), Derber (1987), Le Dimet and Talagrand (1986), Talagrand and Courtier (1987), Courtier and Talagrand (1987), Thacker and Long (1988) and Thacker (1989). Three-dimensional variational data assimilation (3D-Var) attempts to identify the best circulation estimate at a single time using observations from a narrow time window. On the other hand, 4-dimensional variational data assimilation (4D-Var) identifies the best circulation estimate

* Corresponding author. Tel.: +1 831 459 4632; fax: +1 831 459 4882.
E-mail address: ammoore@ucsc.edu (A.M. Moore).

0079-6611/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.pocean.2011.05.004

over a finite time interval using all observations available during the interval, and uses a model to dynamically interpolate information in space and time (see also Talagrand, 1997).

Both 3D-Var and 4D-Var are used routinely in oceanography. For example, Stammer et al. (2002) have applied a 4D-Var approach for ocean state estimation using the MITgcm as part of the "Estimating the Circulation and Climate of the Ocean" (ECCO) project. An incremental 3D-Var and 4D-Var approach are both used in the Ocean Parallelise (OPA) model as described by Weaver et al. (2003). The incremental 4D-Var system described here for the Regional Ocean Modeling System (ROMS) closely parallels that used in OPA. A 3D-Var approach has been developed independently for ROMS by Li et al. (2008). Bennett (1992, 2002) describes a variant of 4D-Var based on the method of representers which has been applied in a wide range of ocean models (Muccino et al., 2008), including ROMS (Di Lorenzo et al., 2007; Kurapov et al., 2009). The representer-based method of 4D-Var described in this paper closely follows that of Chua and Bennett (2001).

This paper is the first in a three part sequence which serves firstly as a review of modern data assimilation methods and diagnostic analyses that are currently available to the oceanographic community, and second as an indispensable reference and demonstration of the power and utility of the ROMS 4D-Var system for regional analyses of the ocean. In this Part I of the sequence, the methods and algorithms employed in ROMS will be described in sufficient detail so as to be understandable by readers with a background in ocean data assimilation. While the algorithmic descriptions presented here are not exhaustive, the reader is referred where appropriate to other, more comprehensive sources. While some aspects of the ROMS incremental 4D-Var system are summarized elsewhere (Powell et al., 2008; Powell and Moore, 2009; Broquet et al., 2009a; Broquet et al., 2009b; Broquet et al., 2011), this paper represents the only comprehensive description of the entire system, particularly in the case of the ROMS 4D physical-space statistical analysis system (PSAS), and the community code version of the ROMS representer-based 4D-Var system. In two companion papers, Moore et al. (in press-a), Moore et al. (in press-b), we present a comparison of the performance of all three ROMS 4D-Var algorithms applied to the California Current System.

The paper begins with a brief overview of ROMS and introduces the notation that will be used throughout. The three ROMS 4D-Var algorithms are introduced in Section 3 based on a search for the best circulation estimate in either the space spanned by the control vector or in the dual space spanned by the observations. The search for the best circulation estimate is facilitated using the Lanczos formulation of the conjugate gradient method and is discussed in Section 4. A critical component of each 4D-Var algorithm is the model of background error covariance matrices which is discussed in Section 5. Preconditioning of the Lanczos algorithm is discussed in Section 6, and some powerful 4D-Var diagnostic tools are described in Section 7.

2. The Regional Ocean Modeling System (ROMS)

ROMS is an hydrostatic, primitive equation, Boussinesq ocean general circulation model designed primarily for coastal applications. Terrain-following vertical coordinates are employed which allow for greater vertical resolution in shallow water and regions with complex bathymetry. Orthogonal curvilinear coordinates are used in the horizontal allowing for increased horizontal resolution in regions characterized by irregular coastal geometry. Even though ROMS is designed with coastal applications is mind, it has also been applied in deep water regions (e.g. Curchitser et al., 2005), on basin scales (e.g. Haidvogel et al., 2000), and to the global ocean (Auad, 2009, personal communication).

One particularly powerful feature of ROMS is the extensive suite of numerical algorithms that are available for solving the momentum and tracer equations. In addition, a wide variety of state-of-the-art physical parameterizations for horizontal and vertical mixing are available, and ROMS has several options for open boundary conditions. A complete description of ROMS is beyond the scope of this paper, however thorough descriptions of the numerical schemes, physical parameterizations, and open boundary condition options can be found in Shchepetkin and McWilliams (2003), Shchepetkin and McWilliams (2005), Marchesiello et al. (2001), and Haidvogel et al. (2008).

In the following sections, it will be necessary to refer explicitly to the nonlinear ROMS equations. This will be done symbolically so that subsequent mathematical developments are less cumbersome, and where possible, we will follow the notation recommended by Ide et al. (1997). A complete list of the mathematical symbols used and their definition is given in Table 1. The ROMS prognostic variables are potential temperature (T), salinity (S), horizontal velocity (u, v), and sea surface displacement (ζ). When the primitive equations are discretized and arranged on the ROMS grid, the individual gridpoint values at time t_i define the components of a state-vector $\mathbf{x}(t_i) = (T,S,\zeta,u,v)^T$ where superscript T denotes the vector transpose. The state-vector is propagated forward in time by the discretized nonlinear ocean model subject to surface boundary conditions, denoted $\mathbf{f}(t_i)$, for momentum, heat and freshwater fluxes, and lateral open boundary conditions, denoted $\mathbf{b}(t_i)$. Following Daget et al. (2009), the surface forcing and boundary conditions can be written as time-tendency terms on the right-hand side (rhs) of the discretized model equations, and the state-vector evolves according to:

$$\mathbf{x}(t_i) = M(t_i, t_{i-1})(\mathbf{x}(t_{i-1}), \mathbf{f}(t_i), \mathbf{b}(t_i)) \tag{1}$$

where $M(t_i,\ t_{i-1})$ represents nonlinear ROMS acting on $\mathbf{x}(t_{i-1})$, and subject to forcing $\mathbf{f}(t_i)$, and boundary conditions $\mathbf{b}(t_i)$ during the time interval $[t_{i-1}, t_i]$. Eq. (1) will hereafter be referred to as NLROMS with initial conditions $\mathbf{x}(t_0)$, surface forcing $\mathbf{f}(t)$, and open boundary conditions $\mathbf{b}(t)$. The time interval under consideration is $[t_0, t_N]$.

In order to apply 4D-Var, three other versions of ROMS are required, all derived directly from the discretized version of NLROMS. All of the 4D-Var data assimilation algorithms currently employed in ROMS are based on departures of the state-vector, surface forcing, and open boundary conditions from a reference background solution, also referred to as the *prior*. Specifically:

$$\begin{aligned}
\mathbf{x}(t_i) &= \mathbf{x}^b(t_i) + \delta\mathbf{x}(t_i) \\
\mathbf{f}(t_i) &= \mathbf{f}^b(t_i) + \delta\mathbf{f}(t_i) \\
\mathbf{b}(t_i) &= \mathbf{b}^b(t_i) + \delta\mathbf{b}(t_i)
\end{aligned} \tag{2}$$

where $\mathbf{x}^b(t_i)$, $\mathbf{f}^b(t_i)$ and $\mathbf{b}^b(t_i)$ are the background fields for the circulation, surface forcing, and open boundary conditions respectively. The increments $\delta\mathbf{x}$, $\delta\mathbf{f}$ and $\delta\mathbf{b}$ are assumed to be small compared to the background fields, in which case they are approximately described by a first-order Taylor expansion of NLROMS in (1), namely:

$$\begin{aligned}
\mathbf{x}^b(t_i) + \delta\mathbf{x}(t_i) = M(t_i, t_{i-1})(&\mathbf{x}^b(t_{i-1}) + \delta\mathbf{x}(t_{i-1}), \mathbf{f}^b(t_i) + \delta\mathbf{f}(t_i), \mathbf{b}^b(t_i) \\
&+ \delta\mathbf{b}(t_i)) \simeq M(t_i, t_{i-1})(\mathbf{x}^b(t_{i-1}), \mathbf{f}^b(t_i), \mathbf{b}^b(t_i)) \\
&+ \mathbf{M}(t_i, t_{i-1})\delta\mathbf{u}(t_{i-1})
\end{aligned}$$

so that:

$$\delta\mathbf{x}(t_i) \simeq \mathbf{M}(t_i, t_{i-1})\delta\mathbf{u}(t_{i-1}). \tag{3}$$

Here $\delta\mathbf{u}(t_{i-1}) = (\delta\mathbf{x}^T(t_{i-1}),\ \delta\mathbf{f}^T(t_i),\ \delta\mathbf{b}^T(t_i))^T$, and $\mathbf{M}(t_i,t_{i-1})$, represents the perturbation tangent linear model for the time interval $[t_{i-1}, t_i]$, linearized about the time evolving background $\mathbf{x}^b(t_{i-1})$ with forcing $\mathbf{f}^b(t_i)$ and boundary conditions $\mathbf{b}^b(t_i)$. Eq. (3) will hereafter be

Table 1
A summary of all the mathematical symbols referred to in the main text and in Part II (Moore et al., in press-a) and Part III (Moore et al., in press-b). Unless otherwise specified, lower case bold characters represent vectors, upper case bold characters are matrices, and scripted or italicized characters are functions. Where applicable, the number of the corresponding equation where each variable is first introduced is also indicated in the third column. Some symbols have different meanings in Parts II and III, and this is indicated in the third column by II and III respectively.

Symbol	Description	Equation/ part
$\mathbf{A}$	A generic matrix to denote the matrix to be inverted during 4D-Var; $\mathcal{H}$ in the primal form and $(\mathbf{GDG}^T + \mathbf{R})$ in the dual form	
$\mathbf{b}$, $\mathbf{b}^b$, $\mathbf{b}^a$	Lateral open boundary conditions. Superscripts denote background/prior (b), and analysis/posterior (a)	2
$\boldsymbol{\beta}$	The vector of representer coefficients	12, 13
$\delta\mathbf{b}$, $\delta\mathbf{b}^*$	Increments to lateral open boundary conditions. Superscript $*$ denotes the adjoint increments	2
$\mathbf{B}_b$	Background/prior error covariance matrix for lateral open boundary conditions	6
$\mathbf{B}_f$	Background/prior error covariance matrix for surface forcing	6
$\mathbf{B}_x$	Background/prior error covariance matrix for state vector initial conditions	6
$\mathcal{B}(\mathbf{x})$	Biconjugate gradient solver	
$\partial\mathcal{B}/\partial\mathbf{x}$	Tangent linearization of the biconjugate gradient solver	
$(\partial\mathcal{B}/\partial\mathbf{x})^T$	Adjoint of the biconjugate gradient solver	
$\mathbf{C}$	Univariate correlation matrix	16
$\mathbf{C}_{h,v}$	Univariate correlation matrix in the horizontal (h) or vertical (v)	18, 19
$\mathbf{d}$	Innovation vector	7
$\mathbf{D}$	Block diagonal background/prior error covariance matrix, diag $(\mathbf{B}_x,\mathbf{B}_f,\mathbf{B}_b,\mathbf{Q})$	7
e_τ	The squared forecast error $(\mathcal{J}^f - \mathcal{J}^t)^2$ in the functional $\mathcal{J}$	III
$\mathbf{e}_i$	Unitary vector with all zero elements, except the ith element which is 1	
$\mathbf{E}^\mathbf{a}$	Analysis/posterior error covariance matrix	21, 22
$\widetilde{\mathbf{E}}^\mathbf{a}$	Reduced rank approximation of $\mathbf{E}^\mathbf{a}$ based on the Lanczos vectors	23
E	Expectation operator	
$\boldsymbol{\varepsilon}_b$	Lateral open boundary condition error	
$\boldsymbol{\varepsilon}_f$	Surface forcing error	
$\boldsymbol{\varepsilon}_i$	State vector initial condition error	
$\boldsymbol{\varepsilon}_m$	Model error	
$\boldsymbol{\varepsilon}_o$	Observation error	
$\delta\mathbf{F}$	Surface freshwater flux increment	II
$\mathbf{f}$, $\mathbf{f}^b$, $\mathbf{f}^a$	Surface forcing (wind stress, heat flux, freshwater flux). Superscripts denote background/prior (b), and analysis/posterior (a)	2
$\delta\mathbf{f}$, $\delta\mathbf{f}^*$	Increments to surface forcing. Superscript $*$ denotes the adjoint increments	2
$\mathbf{g}(t_{i-1})$	Vector of state variables, surface forcing and lateral open boundary conditions for the finite amplitude tangent linearization of ROMS (RPROMS)	5
$\mathbf{G}$	The operator that maps the tangent linear model solution to the observation points	7
$\mathbf{G}^T$	The adjoint of $\mathbf{G}$	7
$\mathbf{GDG}^T$	The representer matrix	
$(\mathbf{GDG}^T + \mathbf{R})$	The stabilized representer matrix	
$\mathbf{h}$	Generic representation for either $\mathbf{G}^T\mathbf{R}^{-1}\mathbf{d}$ in primal form or $\mathbf{d}$ in dual form	
H	Observation operator	6
$\mathbf{H}$	Tangent linearization of H	6
$\mathcal{H}$	Hessian of J	10
$\mathcal{H}_v$	Hessian in v-space	
I	A generic cost/penalty function for which $\mathcal{I}$ and J are the dual and primal forms	
$\mathcal{I}$	Auxiliary function that is directly minimized during dual 4D-Var	
$\boldsymbol{\eta}$, $\boldsymbol{\eta}^b$, $\boldsymbol{\eta}^a$	Control vector corrections for model error. Superscripts denote background/prior (b), and analysis/posterior (a)	
J	Cost/penalty function	6
$\mathbf{J}_0$	The matrix that isolates the contribution of the initial condition state-vector increment to the state-vector increment at a future time	I, III
$\mathcal{J}$	Scalar differentiable function used to characterize some aspect of the circulation for observation impact and observation sensitivity calculations. Superscripts denotes the analysis/posterior (a), background/prior (b), forecast (f), and truth (t)	26, 28
$\mathbf{K}$	Kalman gain matrix	
$\widetilde{\mathbf{K}}$	Reduced rank approximation of gain matrix based on the Lanczos vectors	
$\mathbf{K}_b$	Balance operator	16
$\mathbf{K}_{xy}$	The linear balance describing the relation between variables x and y	
$\mathcal{K}(\mathbf{d})$	Generic function representing the entire dual or primal 4D-Var procedure	
$\partial\mathcal{K}/\partial\mathbf{d}$	The tangent linearization of 4D-Var	
$(\partial\mathcal{K}/\partial\mathbf{d})^T$	Adjoint of 4D-Var	
$\mathbf{L}_{h,v}$	Horizontal 2D diffusion operator (h) or vertical 1D diffusion operator (v)	18, 19

$(\lambda_i, \hat{\mathbf{w}}_i)$	Eigenpairs of $\widehat{\mathcal{P}}$	II
λ_i	Eigenvalues of $\widehat{\mathcal{P}}$ or $\mathbf{T}$	II
$\mathbf{\Lambda}_{h,v}$	Diagonal matrix of normalization coefficients for the horizontal (h) and vertical (v) diffusion operators	18, 19
$\mathbf{\Lambda}$	Diagonal matrix of eigenvalues of $\widehat{\mathcal{P}}$	II
$M(t_i, t_{i-1})$	Nonlinear ROMS for interval $[t_{i-1}, t_i]$ (NLROMS)	1
$\mathbf{M}(t_i, t_{i-1})$	Perturbation tangent linear ROMS (TLROMS)	3
$\mathbf{M}^T(t_{i-1}, t_i)$	Adjoint of tangent linear ROMS (ADROMS)	4
$\boldsymbol{\mathcal{M}}(t_i, t_0)$	Perturbation tangent linear model propagator that maps $\delta\mathbf{z}$ into $\delta\mathbf{x}(t_i)$	
$\boldsymbol{\mathcal{M}}^T(t_0, t_i)$	Adjoint propagator that maps $\mathbf{p}(t_i)$ into $\delta\mathbf{z}^*$	
N_{obs}	Total number of observations	
$\mathbf{p}$	Adjoint state vector increments	
$\mathcal{P}$	The stabilized representer matrix ($\mathbf{GDG}^T + \mathbf{R}$)	II
$\widehat{\mathcal{P}}$	The preconditioned stabilized representer matrix ($\mathbf{R}^{-1/2}\mathbf{GDG}^T\mathbf{R}^{-1/2} + \mathbf{I}$)	II
$\mathbf{Q}$	Background/prior model error covariance matrix	6
$\delta\mathbf{Q}$	Surface heat flux increment	II
$\mathbf{q}_i$	Lanczos vectors (dual or primal)	
$(\theta_i, \mathbf{y}_i)$	Ritz eigenpairs of $\mathbf{T}_m$	
$\mathbf{r}_m(t)$	The representer function for the *mth* observation at time *t*	12
$\mathbf{R}$	Observation error covariance matrix	6
$\mathcal{R}(t)$	The matrix of representer functions at time *t*	12
$\mathbf{s}$	Generic representation of $\delta\mathbf{z}$ in primal form, or $\mathbf{w}$ and $\boldsymbol{\beta}$ in dual form	
S	Salinity	
$\mathbf{\Sigma}$	Matrix of background/prior standard deviations for all control variables	16
$\mathbf{\Sigma}_x$	Matrix of background/prior standard deviations of the unbalanced state vector variables	II
T	Temperature	
t, t_0, t_N	Time variable, with initial time t_0 and final time t_N	
τ_d	Interval of time over which the pseudo-diffusion equation for the covariance models is integrated	
$\delta\boldsymbol{\tau}$	Surface wind stress increments	II
$\mathbf{T}_m$, $\mathbf{T}_p$, $\mathbf{T}_d$	Tridiagonal matrix of Lanczos recursion relation coefficients resulting from m inner-loops. Subscript denotes the primal (p) or dual (d) formulation	
(u, v)	Horizontal velocity	
$\delta\mathbf{u}(t_{i-1})$	Vector of increments for the interval $[t_{i-1}, t_i]$	3
$\delta\mathbf{u}^*(t_{i-1})$	Vector of adjoint increments for the interval $[t_{i-1},t_i]$	
$\mathbf{u}$	Transformed variable $\mathbf{v} = \mathbf{Uu}$	I
$\mathbf{u}(t)$	Representer function	
$\mathbf{U}$	Square root factorization of the preconditioner $\mathbf{X}$	I
$\mathbf{U}(t)$	Matrix of representers	II9
$\mathbf{v}$	The transformed unbalanced state vector increment $\mathbf{D}^{-1/2}$ $\delta\mathbf{x}$ in primal form or $\mathbf{R}^{-1/2}$ $\mathbf{w}$ in dual form	
$\mathbf{V}_m$, $\mathbf{V}_p$, $\mathbf{V}_d$	Matrix of Lanczos vector resulting from m inner-loops. Subscript denotes Lanczos vector matrix for the primal (p) or dual (d) formulation	
$\mathbf{w}$, $\mathbf{w}^a$	An intermediate vector in the dual space. Superscript denotes the analysis/posterior	11
$\mathbf{W}_{h,v}$	Diagonal weight matrix for the horizontal (h) diffusion operator $\mathbf{L}_h$ or vertical (v) diffusion operator $\mathbf{L}_v$	18, 19
$\mathbf{W}$	A diagonal matrix used to include only selected state variable covariances in $\mathbf{Q}$	II9
$\mathbf{x}$, $\mathbf{x}^b$, $\mathbf{x}^a$, $\mathbf{x}^f$, $\mathbf{x}^t$	State vector $(T,S,\varsigma,u,v)^T$. Superscripts denote background/prior (b), analysis/posterior (a), forecast (f), and truth (t)	2
$\delta\mathbf{x}$	Increments to state vector	2
$\delta\mathbf{x}_B$, $\delta\mathbf{x}_U$	The balanced (B) and unbalanced (U) components of $\delta\mathbf{x}$	
$\mathbf{X}$	Preconditioning matrix for $\mathbf{A}$	
$\mathbf{y}^o$	Observation vector	
$\delta\mathbf{y}^o$	Perturbation to observation vector	28
$\hat{\mathbf{y}}_i$	Transformed Ritz vector $\mathbf{V}_m\mathbf{y}_i$	
$\mathbf{\Psi}_i$	Array modes	II10
$\mathbf{z}^b$	Vector of background/prior control variables	
$\mathbf{z}^t$	Vector of true control variables	
$\hat{\mathbf{z}}$	Vector of analysis/posterior control variables	
δ $\mathbf{z}$, δ $\mathbf{z}^a$	Vector of all control variable increments. Superscript denotes the analysis/posterior	
ζ	Free surface height	
ζ_{ref}	Baroclinic reference free surface height used as a balanced reference field for $\mathbf{K}_b$	20

referred to as TLROMS, and with initial conditions $\delta\mathbf{x}(t_0)$, forcing $\delta\mathbf{f}(t)$ and open boundary conditions $\delta\mathbf{b}(t)$.

An operator of central importance to 4D-Var is $\mathbf{M}^T(t_{i-1}, t_i)$ the matrix transpose of the tangent linear model which is integrated backwards, hence the order of the time arguments for $\mathbf{M}^T$ is reversed. The transpose of TLROMS represents the matrix adjoint with respect to the L_2 inner-product, and is associated with the adjoint equation:

$$\delta\mathbf{u}^*(t_{i-1}) = \mathbf{M}^T(t_{i-1}, t_i)\mathbf{p}(t_i) \tag{4}$$

where $\delta\mathbf{u}^*(t_{i-1}) = (\mathbf{p}^T(t_{i-1}), \delta\mathbf{f}^{*T}(t_i), \delta\mathbf{b}^{*T}(t_i))^T$ with $\mathbf{p}$ being the adjoint state-vector increment, and $\delta\mathbf{f}^*$ and $\delta\mathbf{b}^*$ the adjoint of the surface forcing and open boundary condition increments. Eq. (4) will hereafter be referred to as ADROMS, and integrations of ADROMS always start with $\mathbf{p}(t_N) = 0$.

While the incremental approach to 4D-Var described later relies on NLROMS in (1) to propagate $\mathbf{x}^b$ forward in time, the representer method of Bennett (2002) employs a finite-amplitude linearization of ROMS. Specifically, if $\mathbf{x}_k$ denotes the kth member of a linear sequence of k-iterates (identified later as "outer-loops"), then:

$$\begin{aligned}\mathbf{x}_k(t_i) = M(t_i, t_{i-1})(\mathbf{x}_{k-1}(t_{i-1}), \mathbf{f}_{k-1}(t_i), \mathbf{b}_{k-1}(t_i)) + \mathbf{M}_{k-1}(t_i, t_{i-1})\\ \times (\mathbf{g}_k(t_{i-1}) - \mathbf{g}_{k-1}(t_{i-1}))\end{aligned} \tag{5}$$

where $\mathbf{M}_{k-1}$ is TLROMS linearized about $\mathbf{x}_{k-1}$ over the time interval $[t_{i-1}, t_i]$. Eq. (5) is linear in $\mathbf{g}_k = ({\mathbf{x}_k}^T, {\mathbf{f}_k}^T, {\mathbf{b}_k}^T)^T$, and represents the finite-amplitude tangent linearization of ROMS, hereafter referred to as RPROMS. The linear sequence described by (5) is also commonly referred to as Picard iterations (Bennett, 2002) and more generally is used to establish the existence of solutions of nonlinear differential equations. If we denote $(\mathbf{g}_k(t_{i-1}) - \mathbf{g}_{k-1}(t_{i-1}))$ as $\delta\mathbf{g}_k(t_{i-1})$, then the second term on the rhs of (5) is of the form $\mathbf{M}_{k-1}(t_i, t_{i-1})\delta\mathbf{g}_k(t_{i-1})$ which is mathematically equivalent to TLROMS in (3). RPROMS therefore differs from TLROMS by the addition of the first term on the rhs of (5) which represents the NLROMS operators applied to the previous iterate $\mathbf{g}_{k-1}$. A detailed discussion of RPROMS is beyond the scope of this paper, although some additional information can be found in Di Lorenzo et al. (2007). The linearization employed in RPROMS is not unique and there are several possible choices as discussed by Bennett (2002) where illustrative examples are also presented. In the case of ROMS, RPROMS is constructed by adding to TLROMS the appropriate NLROMS terms computed from $\mathbf{g}_{k-1}$.

TLROMS, ADROMS and RPROMS currently exist for all of the commonly used numerical and physical options employed in ROMS. Notable exceptions include most of the vertical mixing and turbulence closure schemes where some of the tangent linear terms must be excluded to prevent the growth of highly unstable modes. The tangent linear and adjoint versions of the surface bulk flux formulations are also linearly unstable in many cases, so are generally not used. TLROMS and ADROMS are described in more detail by Moore et al. (2004).

3. Incremental 4D-Var

The goal of 4D-Var is to identify the best estimate circulation, also commonly referred to as the analysis or *posterior*, namely $\mathbf{x}^a(t)$, that minimizes in a least-squares sense, the difference between the model and the observations and a background, subject to *prior* hypotheses about errors and possibly additional constraints. The solution, $\mathbf{x}(t_i)$, of NLROMS that describes $\mathbf{x}^a$ will depend upon the choice of initial conditions, $\mathbf{x}(t_0)$, surface forcing, $\mathbf{f}(t)$, and boundary conditions, $\mathbf{b}(t)$, all of which are subject to errors and uncertainties. As such, $\mathbf{x}(t_0)$, $\mathbf{f}(t)$ and $\mathbf{b}(t)$ are referred to as control variables, and the problem in 4D-Var is reduced to identifying the appropriate combination of control variables that yield the best estimate $\mathbf{x}^a(t)$. In general, there will be other sources of error and uncertainty associated with the model dynamics and numerics, and unresolved scales of motion (see Daley (1991) and Cohn (1997) for in-depth discussions), which we collectively denote as $\epsilon_m(t_i)$, and introduce an additional vector of control variables $\boldsymbol{\eta}(t_i)$ on the rhs of (1) depending on the 4D-Var approach adopted. To account for model errors, the vector of increments $\delta\mathbf{u}$ in (3) is augmented so that $\delta\mathbf{u}(t_{i-1}) = (\delta\mathbf{x}(t_{i-1})^T, \delta\mathbf{f}^T(t_i), \delta\mathbf{b}^T(t_i), \boldsymbol{\eta}^T(t_i))^T$, and the tangent linear operator $\mathbf{M}(t_i, t_{i-1})$ also advances the correction for model error forward in time, assuming a background correction $\boldsymbol{\eta}^b(t_i) = \mathbf{0}$. Similarly, the adjoint vector associated with (4) is given by $\delta\mathbf{u}^*(t_{i-1}) = (\mathbf{p}^T(t_{i-1}), \delta\mathbf{f}^{*T}(t_i), \delta\mathbf{b}^{*T}(t_i), \boldsymbol{\eta}^{*T}(t_i))^T$.

The development of incremental 4D-Var presented here is based on Courtier et al. (1994) and Courtier (1997), but in an expanded form in which the uncertainties in the surface forcing and lateral boundary conditions are explicitly identified following Daget et al. (2009). The incremental approach to 4D-Var consists of minimizing an objective function, J, given by:

$$\begin{aligned}&J(\delta\mathbf{x}(t_0), \delta\mathbf{f}(t_1), \ldots, \delta\mathbf{f}(t_k), \ldots, \delta\mathbf{b}(t_1), \ldots, \delta\mathbf{b}(t_k), \ldots, \boldsymbol{\eta}(t_1), \ldots, \boldsymbol{\eta}(t_k), \ldots)\\ &\quad = \frac{1}{2}\delta\mathbf{x}^T(t_0)\mathbf{B}_x^{-1}\delta\mathbf{x}(t_0) + \frac{1}{2}\sum_{k=1}^{N}\sum_{j=1}^{N}\Big\{\delta\mathbf{f}^T(t_k)\mathbf{B}_f^{-1}(t_k, t_j)\delta\mathbf{f}(t_j)\\ &\quad\quad + \delta\mathbf{b}^T(t_k)\mathbf{B}_b^{-1}(t_k, t_j)\delta\mathbf{b}(t_j) + \boldsymbol{\eta}^T(t_k)\mathbf{Q}^{-1}(t_k, t_j)\boldsymbol{\eta}(t_j)\Big\}\\ &\quad\quad + \frac{1}{2}\sum_{i=0}^{n}\sum_{l=0}^{n}(\mathbf{H}_i\delta\mathbf{x}(t_i) - \mathbf{d}_i)^T\mathbf{R}_{i,l}^{-1}(\mathbf{H}_l\delta\mathbf{x}(t_l) - \mathbf{d}_l)\end{aligned} \tag{6}$$

where from (2) the increment $\delta\mathbf{x}(t_k) = \mathbf{x}(t_k) - \mathbf{x}^b(t_k)$, and t_i and t_l are identified here as the n observation times. The innovation vector $\mathbf{d}_i = \mathbf{y}_i^o - H_i(\mathbf{x}^b(t_i))$ represents the difference between the vector of observations $\mathbf{y}_i^o$ at time t_i and the model analogue of the observations computed from the background circulation $\mathbf{x}^b(t_i)$ according to the observation operator H_i. In general, H_i will be nonlinear and serves to transform the model state vector to observed variables and to interpolate them to observation points in space and time. Errors arising from H_i are included in $\mathbf{R}$ Janjic and Cohn (2006), and the operator $\mathbf{H}_i$ is the tangent linearization of H_i. While the sequence of times t_k and t_j associated with the forcing, boundary conditions and model errors will, in general, correspond to each of the N model timesteps spanning the assimilation interval, in practice t_k and t_j are typically evaluated less frequently than every timestep.

If $\boldsymbol{\epsilon}_i$, $\boldsymbol{\epsilon}_f(t)$, $\boldsymbol{\epsilon}_b(t)$, $\boldsymbol{\epsilon}_o$ and $\boldsymbol{\epsilon}_m(t)$ denote errors in $\mathbf{x}^b(t_0)$, $\mathbf{f}^b(t)$, $\mathbf{b}^b(t)$, $\mathbf{y}^o$ and the model respectively, then implicit in (6) is the assumption of random, unbiased errors so that $E(\boldsymbol{\epsilon}_i) = \mathbf{0}$, $E(\boldsymbol{\epsilon}_f(t)) = \mathbf{0}$, $E(\boldsymbol{\epsilon}_b(t)) = \mathbf{0}$, $E(\boldsymbol{\epsilon}_m(t)) = \mathbf{0}$ and $E(\boldsymbol{\epsilon}_o) = \mathbf{0}$, where E denotes the expectation operator. The associated error covariances are denoted: $E(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i^T) = \mathbf{B}_x$, the initial condition background error covariance matrix; $E(\boldsymbol{\epsilon}_f(t_k)\boldsymbol{\epsilon}_f^T(t_j)) = \mathbf{B}_f(t_k, t_j)$, the surface forcing background error covariance matrix; $E(\boldsymbol{\epsilon}_b(t_k)\boldsymbol{\epsilon}_b^T(t_j)) = \mathbf{B}_b(t_k, t_j)$, the lateral boundary condition background error covariance matrix; $E(\boldsymbol{\epsilon}_m(t_k)\boldsymbol{\epsilon}_m^T(t_j)) = \mathbf{Q}(t_k, t_j)$, the model error covariance matrix; and $E(\boldsymbol{\epsilon}_o(t_i)\boldsymbol{\epsilon}_o^T(t_j)) = \mathbf{R}_{i,j}$, the observation error covariance matrix. In addition, it is assumed that the individual sources of error are uncorrelated, so that $E\left(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_f^T(t_k)\right) = \mathbf{0}$, $E(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_b^T(t_k)) = \mathbf{0}$, $E(\boldsymbol{\epsilon}_i\boldsymbol{\eta}^T(t_k)) = \mathbf{0}$, $E(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_o^T(t_k)) = \mathbf{0}$, $E(\boldsymbol{\epsilon}_f(t_j)\boldsymbol{\epsilon}_b^T(t_k)) = \mathbf{0}$, etc. Finally, we assume no temporal autocorrelations, and time invariant background error covariances in which case $\mathbf{B}_f(t_k, t_j) = \delta_{k,j}\mathbf{B}_f$, $\mathbf{B}_b(t_k, t_j) = \delta_{k,j}\mathbf{B}_b$, $\mathbf{Q}(t_k, t_j) = \delta_{k,j}\mathbf{Q}$, and $\mathbf{R}_{i,l} = \delta_{i,l}\mathbf{R}_i$, and the double summations in (6) are replaced by a single summation. These statements about the error statistics of the background, observations, forcing, boundary conditions and model represent a precise statement about our *prior* hypotheses.

The objective function J in (6) is referred to as the cost function (or penalty function), and it is customary to write (6) in a more compact form (Courtier, 1997) by introducing the vector $\delta\mathbf{z} = (\delta\mathbf{x}(t_0)^T, \delta\mathbf{f}^T(t_1), \ldots, \delta\mathbf{f}^T(t_k), \ldots, \mathbf{b}^T(t_1), \ldots, \delta\mathbf{b}^T(t_k), \ldots, \boldsymbol{\eta}^T(t_1), \ldots, \boldsymbol{\eta}^T(t_k)\ldots)^T$ which describes all of the control variable increments.

The increment vector $\delta\mathbf{z}$ differs from $\delta\mathbf{u}(t)$ introduced earlier in that $\delta\mathbf{z}$ is comprised of all the elements of the control vector, while $\delta\mathbf{u}(t)$ describes only a subset of the control vector elements. Furthermore, the interpolated and/or transformed increments $\mathbf{H}_i\delta\mathbf{x}(t_i)$ can be expressed as $\mathbf{H}_i\boldsymbol{\mathcal{M}}(t_i,t_0)\delta\mathbf{z}=\mathbf{G}_i\delta\mathbf{z}$, where $\boldsymbol{\mathcal{M}}(t_i,t_0)$ is an alternative form of the tangent linear operator that isolates the state vector increment as described in the appendix. Finally, we introduce the following: the matrix $\mathbf{G}=\left(\ldots,\mathbf{G}_i^T,\ldots\right)^T$; the vector $\mathbf{d}=\left(\ldots,\mathbf{d}_i^T,\ldots\right)^T$ of innovations of length N_{obs}; the block diagonal matrix $\mathbf{R}$ with blocks $\mathbf{R}_i$; and the block diagonal matrix $\mathbf{D}$ with blocks $\mathbf{B}_x$, $\mathbf{B}_f$, $\mathbf{B}_b$ and $\mathbf{Q}$. The cost (penalty) function can then be written as:

$$J(\delta\mathbf{z})=\frac{1}{2}\delta\mathbf{z}^T\mathbf{D}^{-1}\delta\mathbf{z}+\frac{1}{2}(\mathbf{G}\delta\mathbf{z}-\mathbf{d})^T\mathbf{R}^{-1}(\mathbf{G}\delta\mathbf{z}-\mathbf{d}). \tag{7}$$

The desired analysis increment, $\delta\mathbf{z}^a$, that minimizes (7) corresponds to the solution of the equation $\partial J/\partial\delta\mathbf{z}=0$, and is given by:

$$\delta\mathbf{z}^a=(\mathbf{D}^{-1}+\mathbf{G}^T\mathbf{R}^{-1}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{R}^{-1}\mathbf{d} \tag{8}$$

$$=\mathbf{D}\mathbf{G}^T(\mathbf{G}\mathbf{D}\mathbf{G}^T+\mathbf{R})^{-1}\mathbf{d}. \tag{9}$$

Eq. (9) is algebraically equivalent to optimum interpolation (Lorenc, 1986), and is referred to as the dual form, while (8) is some times referred to as the primal form. If the vector $\mathbf{z}^b=(\mathbf{x}^{bT},\ \mathbf{f}^b(t_1)^T,\ldots,\ \mathbf{f}^b(t_k)^T,\ldots,\mathbf{b}^b(t_1)^T,\ldots,\mathbf{b}^b(t_k)^T,\ldots,\mathbf{0}^T)^T$ represents the background control vector, then the best estimate of the circulation is given by $\hat{\mathbf{z}}=\mathbf{z}^b+\delta\mathbf{z}^a$.

3.1. Strong versus weak constraint 4D-Var

It is common in 4D-Var to neglect errors in the model, in which case $\boldsymbol{\epsilon}_m(t_k)=\mathbf{0}$. In addition, the assumption is often made that the surface and lateral boundary conditions are error-free, in which case $\boldsymbol{\epsilon}_f(t_k)=\mathbf{0}$ and $\boldsymbol{\epsilon}_b(t_k)=\mathbf{0}$ also. Following Sasaki (1970), such applications of 4D-Var are subject to a "strong constraint" imposed by the model dynamics which amounts to neglecting the second term on the rhs of (6). In oceanographic applications, the term "strong constraint" is also generally used when only model error is neglected, and we will adopt this convention here. When model errors are admitted ($\boldsymbol{\epsilon}_m(t_k)\neq\mathbf{0}$), 4D-Var is said to be subject to a "weak constraint" imposed by the model dynamics. Additional recent discussions of the issues surrounding model error and weak constraint 4D-Var can be found in Trémolet (2006, 2007).

3.2. Model space versus observation space

Consider Eq. (8) where identifying $\delta\mathbf{z}^a$ is equivalent to solving the linear equation $\boldsymbol{\mathcal{H}}\delta\mathbf{z}^a=\mathbf{G}^T\mathbf{R}^{-1}\mathbf{d}$ where:

$$\boldsymbol{\mathcal{H}}=(\mathbf{D}^{-1}+\mathbf{G}^T\mathbf{R}^{-1}\mathbf{G}) \tag{10}$$

is the Hessian of J. In practice, the estimate of $\delta\mathbf{z}^a$ is computed iteratively by minimizing the cost (penalty) function in (7), which recall corresponds to finding $\delta\mathbf{z}$ that satisfies $\partial J/\partial\delta\mathbf{z}=\mathbf{0}$. According to (8), this is a very challenging problem because the dimension n_z of $\delta\mathbf{z}$ is very large, especially in the case of weak constraint 4D-Var where the correction for model error $\boldsymbol{\eta}(t_k)$ must be identified, in principle, for every model timestep. For this reason, Eq. (8) is usually employed only for strong constraint 4D-Var, in which case the dimension of $\delta\mathbf{z}$ is equal to the number of model gridpoint variables plus all the points in space and time that define the surface and lateral boundary conditions. Strong constraint 4D-Var using (8) therefore involves a search for the optimal circulation estimate in the space spanned by the model control vector.

3.2.1. Incremental strong constraint 4D-Var (I4D-Var)

It is important to realize that while (8) is written in terms of a series of matrix operations, the matrices are never explicitly computed. Instead, matrix–vector products are evaluated using the appropriate model components, including the covariance matrices that are modeled as solutions of diffusion equations as discussed in Section 5. Thus the solution of (8) can be expressed as an iterative sequence of matrixless operations as illustrated in Fig. 1 (see also Courtier, 1997). The solution of (8) proceeds by solving an equivalent system of linear equations using a preconditioned Lanczos formulation of the conjugate gradient method, and the steps in Fig. 1 associated with the "Lanczos algorithm" and "preconditioner" are described in Sections 4 and 5. Steps (ii)–(vi) in Fig. 1 are referred to as an inner-loop, and when repeated represent the minimization of J in the space spanned by the control vector in the vicinity of $\mathbf{z}^b$.

To account for nonlinearities, it is often advantageous to also refine the nonlinear model solution about which TLROMS and ADROMS are initialized. In this case, then using (2), step (vii) of Fig. 1 is performed with the new initial condition $\mathbf{x}_k(t_0)=\mathbf{x}^b(t_0)+\delta\mathbf{x}^k(t_0)$, $\mathbf{f}_k(t)=\mathbf{f}^b(t)+\delta\mathbf{f}^k(t)$ and $\mathbf{b}_k(t)=\mathbf{b}^b(t)+\delta\mathbf{b}^k(t)$, where k refers to the new run of NLROMS, and $\delta\mathbf{x}^k(t_0)$, $\delta\mathbf{f}^k(t)$ and $\delta\mathbf{b}^k(t)$ are the increments computed during step (vi) of the previous inner-loop. Step (vii) is referred to as an outer-loop, and TLROMS and ADROMS are linearized about $\mathbf{x}_k$ during the next sequence of inner-loops. A subscript k is therefore also implied for $\mathbf{G}$ and $\mathbf{G}^T$ but has been omitted for the sake of clarity. It is also important to note that the innovation vector $\mathbf{d}$ is always computed relative to the background circulation $\mathbf{x}^b(t)$ and is not updated between outer-loops. This aspect of 4D-Var in ROMS follows Bennett (2002) and differs from the practice followed in some other models where $\mathbf{d}$ is sometimes updated between outer-loops.

3.2.2. A physical-space statistical analysis system (4D-PSAS)

The best estimate increment $\delta\mathbf{z}^a$ is also given by the dual form (9), which can be written equivalently as $\delta\mathbf{z}^a=\mathbf{D}\mathbf{G}^T\mathbf{w}^a$ where $\mathbf{w}^a$ is a vector in the dual of $\mathbf{x}$ and satisfies:

$$(\mathbf{G}\mathbf{D}\mathbf{G}^T+\mathbf{R})\mathbf{w}^a=\mathbf{d}. \tag{11}$$

The dual form has the advantage that the dimension of $\mathbf{w}^a$ is equal to the number of observations which, in oceanographic applications, is typically several orders of magnitude smaller than the dimension of the control vector. Therefore, solving (11) may be far less challenging than solving (8). Once $\mathbf{w}^a$ has been identified, $\delta\mathbf{z}^a=\mathbf{D}\mathbf{G}^T\mathbf{w}^a$ can be computed by a single integration of ADROMS, $\mathbf{G}^T$, followed by a multiplication by the error covariance matrix $\mathbf{D}$.

More importantly, however, the dimension of $\mathbf{w}^a$ is independent of the strong and weak constraint; despite the enormous difference in the dimension of $\delta\mathbf{z}^a$ when comparing the strong and weak constraint case, $\mathbf{w}^a$ always has the dimension of the number of observations. The dual form (9) therefore involves a search for the best circulation estimate in the space of linear functions (i.e. $\mathbf{G}$) of the state vector $\delta\mathbf{x}$ spanned only by the observations, which makes the weak constraint estimation problem more tractable.

Since (8) and (9) are equivalent, the primal and dual formulations yield identical solutions $\delta\mathbf{z}^a$. Courtier (1997) notes that the solution of (11) minimizes the function $\mathcal{I}(\mathbf{w})=\frac{1}{2}\mathbf{w}^T(\mathbf{G}\mathbf{D}\mathbf{G}^T+\mathbf{R})\mathbf{w}-\mathbf{w}^T\mathbf{d}$ which is equivalent to the physical-space statistical analysis system (PSAS) proposed by Da Silva et al. (1995). The solution of (9) via (11) (*i.e.* by minimizing of $\mathcal{I}(\mathbf{w})$) in the presence of either the weak or strong constraint proceeds iteratively as shown in Fig. 2. Repeated application of the inner-loops (steps (ii)–(vi)), followed by steps (vii) and (viii) is equivalent to minimization of J but now in observation space. Analogous to I4D-Var, the "outer-loop" step (ix) may be repeated to update the circulation about which TLROMS and ADROMS are linearized, but $\mathbf{d}$ is always computed relative to the background $\mathbf{x}^b(t)$.

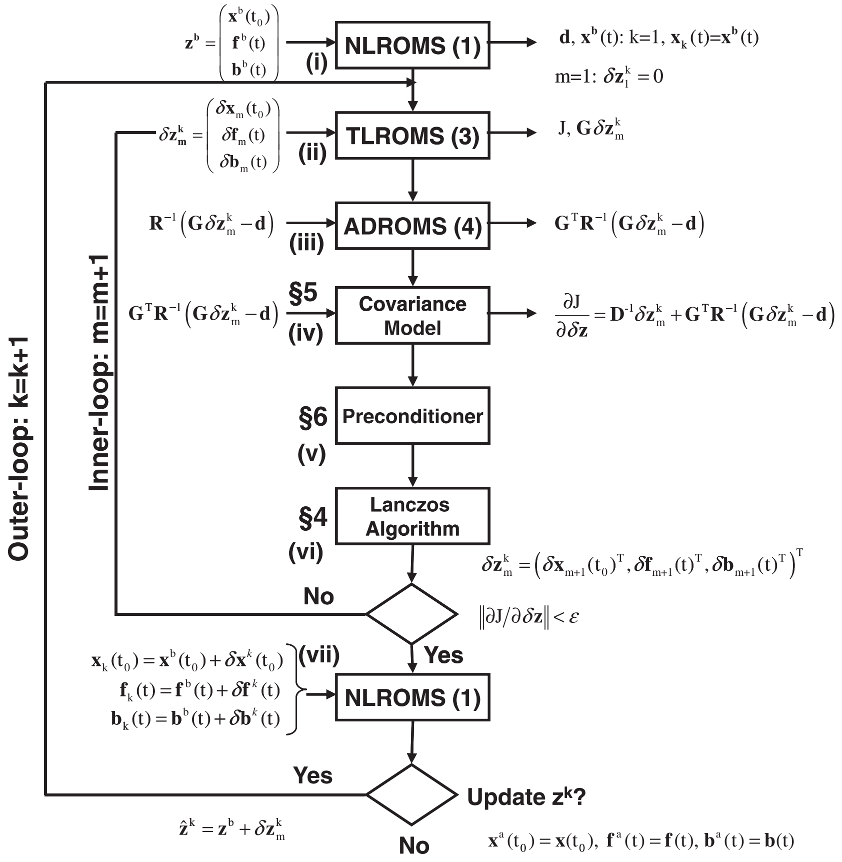
$\mathbf{z}^b = \begin{pmatrix}\mathbf{x}^b(t_0)\\ \mathbf{f}^b(t)\\ \mathbf{b}^b(t)\end{pmatrix}$
(i)
NLROMS (1)
$\mathbf{d}, \mathbf{x}^b(t)$: k=1, $\mathbf{x}_k(t)=\mathbf{x}^b(t)$
m=1: $\delta\mathbf{z}_1^k = 0$
$\delta\mathbf{z}_m^k = \begin{pmatrix}\delta\mathbf{x}_m(t_0)\\ \delta\mathbf{f}_m(t)\\ \delta\mathbf{b}_m(t)\end{pmatrix}$
(ii)
TLROMS (3)
J, $\mathbf{G}\delta\mathbf{z}_m^k$
$\mathbf{R}^{-1}(\mathbf{G}\delta\mathbf{z}_m^k - \mathbf{d})$
(iii)
ADROMS (4)
$\mathbf{G}^T\mathbf{R}^{-1}(\mathbf{G}\delta\mathbf{z}_m^k - \mathbf{d})$
$\mathbf{G}^T\mathbf{R}^{-1}(\mathbf{G}\delta\mathbf{z}_m^k - \mathbf{d})$
§5
(iv)
Covariance Model
$\frac{\partial J}{\partial \delta\mathbf{z}} = \mathbf{D}^{-1}\delta\mathbf{z}_m^k + \mathbf{G}^T\mathbf{R}^{-1}(\mathbf{G}\delta\mathbf{z}_m^k - \mathbf{d})$
§6
(v)
Preconditioner
§4
(vi)
Lanczos Algorithm
$\delta\mathbf{z}_m^k = (\delta\mathbf{x}_{m+1}(t_0)^T, \delta\mathbf{f}_{m+1}(t)^T, \delta\mathbf{b}_{m+1}(t)^T)^T$
Outer-loop: k=k+1
Inner-loop: m=m+1
No
$\|\partial J/\partial\delta\mathbf{z}\| < \varepsilon$
Yes
$\mathbf{x}_k(t_0) = \mathbf{x}^b(t_0) + \delta\mathbf{x}^k(t_0)$
$\mathbf{f}_k(t) = \mathbf{f}^b(t) + \delta\mathbf{f}^k(t)$
$\mathbf{b}_k(t) = \mathbf{b}^b(t) + \delta\mathbf{b}^k(t)$
(vii)
NLROMS (1)
Yes
Update z^k?
$\hat{\mathbf{z}}^k = \mathbf{z}^b + \delta\mathbf{z}_m^k$
No
$\mathbf{x}^a(t_0) = \mathbf{x}(t_0)$, $\mathbf{f}^a(t) = \mathbf{f}(t)$, $\mathbf{b}^a(t) = \mathbf{b}(t)$

Fig. 1. A flow chart illustrating the strong constraint ROMS I4D-Var algorithm. The individual model components NLROMS, TLROMS and ADROMS are all identified, and the Arabic numbers in parentheses refer to the appropriate equation numbers in the main text. The Roman numerals refer to the different components of the inner- and outer-loops described in Section 3.2.1. The calling sequence of the sub-algorithms for the covariance model, Lanczos algorithm and preconditioner in relation to the integration of each model component is also indicated, and § indicates the section number in the main text where each sub-algorithm is described. The arrows to the left (right) of the boxes denote inputs (outputs) associated with the individual model components. The integer *m* (*k*) refers to the number of inner- (outer-) loops, and all symbols are defined in the main text and Table 1.

3.2.3. The method of representers (R4D-Var)

Bennett (1992) describes an alternative approach to dual 4D-Var where the best estimate state-vector $\mathbf{x}^a$ is expressed as:

$$\mathbf{x}^a(t) = \mathbf{x}^b(t) + \mathcal{R}(t)\boldsymbol{\beta} \quad (12)$$

where each column of the matrix $\mathcal{R}(t)$ is a representer function denoted $\mathbf{r}_m(t)$, with one representer function associated with each of the $m = 1,\ldots,N_{\text{obs}}$ observations, and $\boldsymbol{\beta}$ is the $(N_{\text{obs}} \times 1)$ vector of representer coefficients. If $\tilde{\mathbf{x}}(t)$ denotes the response of the model to random forcing with statistics that are consistent with the *prior* hypotheses of Section 3, then each representer $\mathbf{r}_m(t)$ describes the covariance between the circulation $\tilde{\mathbf{x}}(t)$ sampled at the space–time location of the *m*th observation and the field $\tilde{\mathbf{x}}(t)$ at all other points and times (Bennett, 2002, Sections 2.2 and 2.4).

Computation of each column of $\mathcal{R}$ requires one integration of ADROMS and one integration of TLROMS, so if N_{obs} is large, it is not practical to compute all of the representers explicitly. However, Eq. (12) indicates that all that is required is the product $\mathcal{R}(t)\boldsymbol{\beta}$ which can be cleverly evaluated using the indirect representer method introduced by Egbert et al. (1994). The vector of representer coefficients, $\boldsymbol{\beta}$, is in fact the solution of:

$$(\mathbf{GDG}^T + \mathbf{R})\boldsymbol{\beta} = \mathbf{d} \quad (13)$$

(Chua and Bennett, 2001), where (13) is analogous to (11). The procedure for solving (13) subject to either the strong or weak constraint is illustrated in Fig. 3, and is similar to that used for 4D-PSAS except now $\mathbf{x}(t)$ about which TLROMS and ADROMS are linearized is computed using RPROMS. Fig. 3 shows that RPROMS is run at step (ii) and step (x) which mark the start and end of each outer-loop, *k*. The RPROMS solution at step (ii) is always obtained using the background initial condition, $\mathbf{x}^b(t_0)$, surface forcing, $\mathbf{f}^b(t)$, and boundary conditions, $\mathbf{b}^b(t)$. Therefore during the first outer-loop ($k = 1$) the RPROMS solution at step (ii) and the NLROMS solution $\hat{\mathbf{x}}(t)$ at step (i) (about which step (ii) is linearized when $k = 1$) are identical, in which case (11) and (13) are equivalent and 4D-PSAS and R4D-Var yield the same inner-loop solutions. Therefore during the first outer-loop, step (ii) is redundant, but is included in Fig. 3 for completeness. Conversely, at step (x) in Fig. 3 the increments computed at the end of the last inner-loop are applied to RPROMS, which yields a new outer-loop circulation estimate $\mathbf{x}_k(t)$ according to (5). It is at this point that solutions of R4D-Var and 4D-PSAS diverge. During outer-loops $k > 1$, RPROMS at steps (ii) and (x) is linearized about the circulation $\hat{\mathbf{x}}_k(t) = \mathbf{x}^b(t) + \delta\mathbf{x}^k(t)$ identified at the end of the previous inner-loop, and forced by $\mathbf{f}_k(t) = \mathbf{f}^b(t) + \delta\mathbf{f}^k(t)$ and subject to $\mathbf{b}_k(t) = \mathbf{b}^b(t) + \delta\mathbf{b}^k(t)$.

The matrix $\mathbf{GDG}^T$ is referred to as the representer matrix and is the covariance between the model fields sampled at each observation space–time location. For closely spaces observations, such as satellite data, $\mathbf{GDG}^T$ may be poorly conditioned in which case the addition of the observation error covariance matrix $\mathbf{R}$ improves the conditioning of (13), and $(\mathbf{GDG}^T + \mathbf{R})$ is called the stabilized representer matrix.

As discussed by Bennett (1992, 2002), the best circulation estimate $\mathbf{x}^a(t)$ given by (12) satisfies the nonlinear Euler–Lagrange equations, and the R4D-Var algorithm summarized above

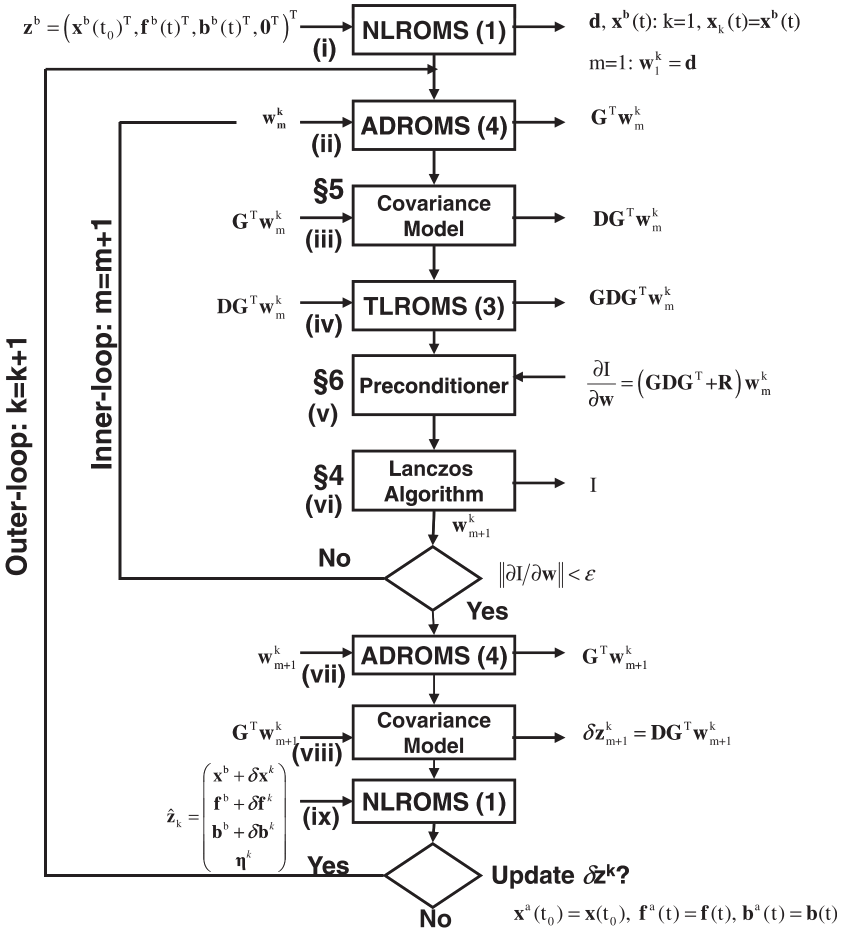
$\mathbf{z}^b = (\mathbf{x}^b(t_0)^T, \mathbf{f}^b(t)^T, \mathbf{b}^b(t)^T, \mathbf{0}^T)^T$
NLROMS (1)
(i)
$\mathbf{d}$, $\mathbf{x}^b(t)$: k=1, $\mathbf{x}_k(t)=\mathbf{x}^b(t)$
m=1: $\mathbf{w}_1^k = \mathbf{d}$
$\mathbf{w}_m^k$
ADROMS (4)
(ii)
$\mathbf{G}^T\mathbf{w}_m^k$
§5
$\mathbf{G}^T\mathbf{w}_m^k$
Covariance Model
(iii)
$\mathbf{DG}^T\mathbf{w}_m^k$
$\mathbf{DG}^T\mathbf{w}_m^k$
TLROMS (3)
(iv)
$\mathbf{GDG}^T\mathbf{w}_m^k$
§6
Preconditioner
(v)
$\frac{\partial I}{\partial \mathbf{w}} = (\mathbf{GDG}^T + \mathbf{R})\mathbf{w}_m^k$
§4
Lanczos Algorithm
(vi)
I
$\mathbf{w}_{m+1}^k$
No
$\|\partial I/\partial \mathbf{w}\| < \varepsilon$
Yes
Inner-loop: m=m+1
Outer-loop: k=k+1
$\mathbf{w}_{m+1}^k$
ADROMS (4)
(vii)
$\mathbf{G}^T\mathbf{w}_{m+1}^k$
$\mathbf{G}^T\mathbf{w}_{m+1}^k$
Covariance Model
(viii)
$\delta\mathbf{z}_{m+1}^k = \mathbf{DG}^T\mathbf{w}_{m+1}^k$
$\hat{\mathbf{z}}_k = \begin{pmatrix} \mathbf{x}^b + \delta\mathbf{x}^k \\ \mathbf{f}^b + \delta\mathbf{f}^k \\ \mathbf{b}^b + \delta\mathbf{b}^k \\ \boldsymbol{\eta}^k \end{pmatrix}$
NLROMS (1)
(ix)
Yes
Update $\delta\mathbf{z}^k$?
No
$\mathbf{x}^a(t_0) = \mathbf{x}(t_0)$, $\mathbf{f}^a(t) = \mathbf{f}(t)$, $\mathbf{b}^a(t) = \mathbf{b}(t)$

Fig. 2. A flow chart illustrating the ROMS 4D-PSAS algorithm. The individual model components NLROMS, TLROMS and ADROMS are all identified, and the Arabic numbers in parentheses refer to the appropriate equation numbers in the main text. The Roman numerals refer to the different components of the inner- and outer-loops described in Section 3.2.2. The calling sequence of the sub-algorithms for the covariance model, Lanczos algorithm and preconditioner in relation to the integration of each model component is also indicated, and § indicates the section number in the main text where each sub-algorithm is described. The arrows to the left (right) of the boxes denote inputs (outputs) associated with the individual model components. The integer *m* (*k*) refers to the number of inner- (outer-) loops, and all symbols are defined in the main text and Table 1.

represents a linear iterative approach for solving the nonlinear Euler–Lagrange equations for ROMS. The outer-loops of R4D-Var are based on Picard iterates, commonly used to establish the existence of a solution to a nonlinear differential equation (Gesztesy and Sticka, 1998), and it is in this sense that R4D-Var fundamentally differs from 4D-PSAS of Section 3.2.2. During R4D-Var, both the inner- and outer-loops are linear, and during the outer-loops dynamical information is conveyed using RPROMS. As discussed in detail by Bennett (2002), a complete linearization of the problem is essential for identifying the true optimal circulation estimate. This is in contrast to 4D-PSAS where NLROMS is used in the outer-loops, in which case a suboptimal estimate will result.

4. Conjugate gradients and the Lanczos Algorithm

Identification of the best circulation estimate using either I4D-Var, 4D-PSAS or R4D-Var involves the solution of a sequence of linear least-squares minimizations. Each 4D-Var algorithm attempts to solve a linear equation by minimizing the cost (penalty) functional *J* in (7). In I4D-Var *J* is minimized directly in the full space spanned by the control vector, while in 4D-PSAS and R4D-Var the minimum of *J* is identified indirectly by minimizing an auxilliary function $\mathcal{I}$ (via (11) or (13)) in observation space. Both *J* and $\mathcal{I}$ can be written in a generic quadratic form as $I(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T\mathbf{A}\mathbf{s} - \mathbf{s}^T\mathbf{h} + \mathbf{c}$. The minimum of *I* corresponds to the condition $\partial I/\partial \mathbf{s} = 0$ for which **s** satisfies the linear equation **As** = **h**. For I4D-Var, inspection of (7) shows that $\mathbf{s} = \delta\mathbf{z}$, $\mathbf{h} = \mathbf{G}^T\mathbf{R}^{-1}\mathbf{d}$, $\mathbf{c} = \frac{1}{2}\mathbf{d}^T\mathbf{R}^{-1}\mathbf{d}$, and $\mathbf{A} = \mathcal{H} = (\mathbf{D}^{-1} + \mathbf{G}^T\mathbf{R}^{-1}\mathbf{G})$ the Hessian of *J* given by (10). Eqs. (11) and (13) show that for 4D-PSAS and R4D-Var, **s** is equivalent to the intermediate solution in observation space given by **w** and $\boldsymbol{\beta}$ respectively, and in both cases **c** = 0, **h** = **d** and $\mathbf{A} = (\mathbf{GDG}^T + \mathbf{R})$, the stabilized representer matrix.

In ROMS, a conjugate gradient descent method recast in the form of a Lanczos algorithm is employed to solve **As** = **h** in all three cases (Fisher and Courtier, 1995). The close connection between the conjugate gradient descent method and the Lanczos algorithm was first made by Paige and Saunders (1975) and is discussed at length by Golub and van Loan (1989). The Lanczos algorithm is favored here because it offers tremendous additional utility to ROMS 4D-Var, as described in Section 7 and the companion papers Moore et al. (in press-a), Moore et al. (in press-b).

Following Lanczos (1950), a reduced rank factorization of the symmetric, positive definite matrix **A** can be found which satisfies the recurrence relation given by:

$$\mathbf{AV}_m = \mathbf{V}_m\mathbf{T}_m + \gamma_m\mathbf{q}_{m+1}\mathbf{e}_m^T \tag{14}$$

where $\mathbf{V}_m = (\ldots, \mathbf{q}_i, \ldots)$, $i = 1, \ldots, m$, is a matrix composed of the orthonormal vectors $\mathbf{q}_i$, referred to as the Lanczos vectors. The matrix **T** is a $(m \times m)$ symmetric tridiagonal matrix with leading diagonal elements δ_i, $i = 1, \ldots, m$, and off-diagonal elements γ_i,

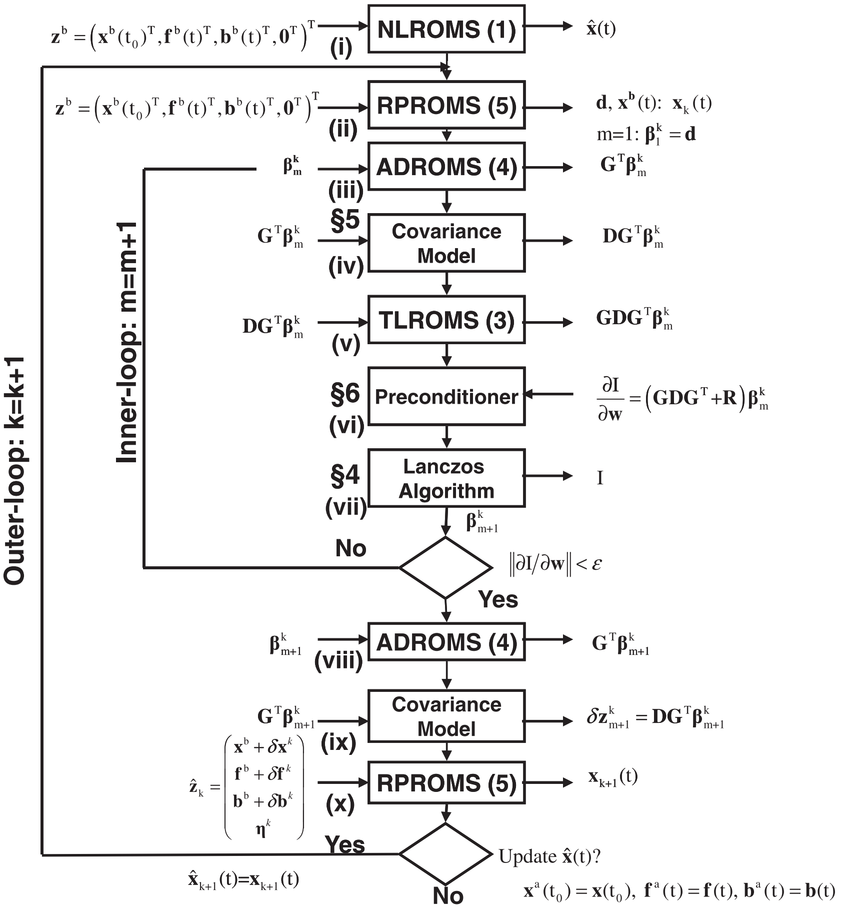
$\mathbf{z}^b = (\mathbf{x}^b(t_0)^T, \mathbf{f}^b(t)^T, \mathbf{b}^b(t)^T, \mathbf{0}^T)^T$
(i)
NLROMS (1)
$\hat{\mathbf{x}}(t)$
$\mathbf{z}^b = (\mathbf{x}^b(t_0)^T, \mathbf{f}^b(t)^T, \mathbf{b}^b(t)^T, \mathbf{0}^T)^T$
(ii)
RPROMS (5)
$\mathbf{d}, \mathbf{x}^b(t)$: $\mathbf{x}_k(t)$
m=1: $\boldsymbol{\beta}_1^k = \mathbf{d}$
$\boldsymbol{\beta}_m^k$
(iii)
ADROMS (4)
$\mathbf{G}^T\boldsymbol{\beta}_m^k$
§5
$\mathbf{G}^T\boldsymbol{\beta}_m^k$
(iv)
Covariance Model
$\mathbf{DG}^T\boldsymbol{\beta}_m^k$
$\mathbf{DG}^T\boldsymbol{\beta}_m^k$
(v)
TLROMS (3)
$\mathbf{GDG}^T\boldsymbol{\beta}_m^k$
§6
(vi)
Preconditioner
$\frac{\partial I}{\partial \mathbf{w}} = (\mathbf{GDG}^T + \mathbf{R})\boldsymbol{\beta}_m^k$
§4
(vii)
Lanczos Algorithm
I
$\boldsymbol{\beta}_{m+1}^k$
No
$\|\partial I/\partial \mathbf{w}\| < \varepsilon$
Yes
Inner-loop: m=m+1
Outer-loop: k=k+1
$\boldsymbol{\beta}_{m+1}^k$
(viii)
ADROMS (4)
$\mathbf{G}^T\boldsymbol{\beta}_{m+1}^k$
$\mathbf{G}^T\boldsymbol{\beta}_{m+1}^k$
Covariance Model
$\delta\mathbf{z}_{m+1}^k = \mathbf{DG}^T\boldsymbol{\beta}_{m+1}^k$
(ix)
$\hat{\mathbf{z}}_k = \begin{pmatrix} \mathbf{x}^b + \delta\mathbf{x}^k \\ \mathbf{f}^b + \delta\mathbf{f}^k \\ \mathbf{b}^b + \delta\mathbf{b}^k \\ \boldsymbol{\eta}^k \end{pmatrix}$
(x)
RPROMS (5)
$\mathbf{x}_{k+1}(t)$
Yes
Update $\hat{\mathbf{x}}(t)$?
$\hat{\mathbf{x}}_{k+1}(t) = \mathbf{x}_{k+1}(t)$
No
$\mathbf{x}^a(t_0) = \mathbf{x}(t_0)$, $\mathbf{f}^a(t) = \mathbf{f}(t)$, $\mathbf{b}^a(t) = \mathbf{b}(t)$

Fig. 3. A flow chart illustrating the ROMS R4D-Var algorithm. The individual model components NLROMS, TLROMS, ADROMS and RPROMS are all identified, and the Arabic numbers in parentheses refer to the appropriate equation numbers in the main text. The Roman numerals refer to the different components of the inner- and outer-loops described in Section 3.2.3. The calling sequence of the sub-algorithms for the covariance model, Lanczos algorithm and preconditioner in relation to the integration of each model component is also indicated, and § indicates the section number in the main text where each sub-algorithm is described. The arrows to the left (right) of the boxes denote inputs (outputs) associated with the individual model components. The integer m (k) refers to the number of inner- (outer-) loops, and all symbols are defined in the main text and Table 1.

$i = 1,\ldots,m-1$, and the vector $\mathbf{e}_m = (0,0,\ldots,1)^T$ is the unitary vector of length m. Eq. (14) can also be written as:

$$\mathbf{Aq}_m = \gamma_m\mathbf{q}_{m+1} + \delta_m\mathbf{q}_m + \gamma_{m-1}\mathbf{q}_{m-1} \tag{15}$$

the more familiar form of the Lanczos recurrence relation.

Given the orthonormal nature of the Lanczos vectors, $\mathbf{q}_i^T\mathbf{q}_j = \delta_{i,j}$, it follows from (15) that $\delta_m = \mathbf{q}_m^T\mathbf{Aq}_m$ and $\gamma_m = (\mathbf{a}_m^T\mathbf{a}_m)^{\frac{1}{2}}$ where $\mathbf{a}_m = \mathbf{Aq}_m - \delta_m\mathbf{q}_m - \gamma_{m-1}\mathbf{q}_{m-1}$. Given $\mathbf{q}_{m-1}$ and $\mathbf{q}_m$, Eq. (15) shows that it is possible to compute the next member of the sequence $\mathbf{q}_{m+1}$. If $\mathbf{s}_0$ is an initial starting guess for the solution of $\mathbf{As} = \mathbf{h}$, the first member of the Lanczos vector sequence is $\mathbf{q}_1 = (\mathbf{As}_0 - \mathbf{h})/|\,\mathbf{As}_0 - \mathbf{h}|$, and the next member being $\mathbf{q}_2 = (\mathbf{Aq}_1 - \delta_1\mathbf{q}_1)/\gamma_1$.

During 4D-PSAS and R4D-Var, the operation $\mathbf{Aq}_m$ in (15) is achieved via a single inner-loop of the respective algorithms, while for I4D-Var, Eq. (7) shows that $\mathbf{Aq}_m \equiv \mathcal{H}\mathbf{q}_m = \nabla J(\mathbf{q}_m) - \nabla J(0)$, so the initial cost/penalty function gradient $\nabla J(0)$ must be subtracted from ∇J during each subsequent inner-loop. Clearly each inner-loop yields one additional member of the Lanczos vector sequence, so that after m inner-loops there are $m + 1$ Lanczos vectors, and the solution estimate for $\mathbf{As} = \mathbf{h}$ is given by $\mathbf{s}_m = \mathbf{s}_0 - \mathbf{V}_m\mathbf{T}_m^{-1}\mathbf{V}_m^T(\mathbf{As}_0 - \mathbf{h})$. Following Tshimanga et al. (2008), $I(\mathbf{s}_m) = I(\mathbf{s}_0) + \frac{1}{2}(\mathbf{As}_0 - \mathbf{h})^T(\mathbf{s}_m - \mathbf{s}_0)$, and the sequence of inner-loops proceeds until I and/or $\partial I/\partial\mathbf{s}$ reach acceptably small values.

5. Error covariance modeling

The error covariance matrices $\mathbf{B}_x$, $\mathbf{B}_f$, $\mathbf{B}_b$ and $\mathbf{Q}$ in (6) are a statement about the prior hypotheses regarding the background fields, and also serve to regularize the resulting estimate by spreading the influence of the observations and background fields in space.[1]

The specification and modeling of the background error covariances presents one of the greatest challenges in 4D-Var. In ROMS, each background error covariance matrix is factorized following Derber and Rosati (1989), Derber and Bouttier (1999) and Weaver and Courtier (2001) according to:

$$\mathbf{B} = \mathbf{K}_b\boldsymbol{\Sigma}\mathbf{C}\boldsymbol{\Sigma}^T\mathbf{K}_b^T \tag{16}$$

where $\mathbf{K}_b$ describes the balanced components of the background errors. Specifically, the state-vector increment in (2) is decomposed as $\delta\mathbf{x} = \delta\mathbf{x}_B + \delta\mathbf{x}_U = \mathbf{K}_b\delta\mathbf{x}_U$ where the subscripts B and U denote the balanced and unbalanced components respectively (Derber and Bouttier, 1999). The underlying assumption here is that the state variables of a balanced circulation (e.g. geostrophic flow) will be mutually correlated, while the unbalanced residual circulation is expected to be largely uncorrelated. The validity of this assumption will clearly depend on the nature of the circulation regime in ques-

[1] Spreading in time via temporal correlation functions is also possible, but is not yet implemented in ROMS 4D-Var.

tion, but experience shows that it is a useful approximation for practical purposes. The balance operator $\mathbf{K}_b$ in (16) is therefore defined so that the unbalanced components $\delta\mathbf{x}_U$ are approximately mutually uncorrelated, in which case the correlation matrix $\mathbf{C}$ of the unbalanced component of the errors in (16) is block diagonal, univariate, with standard deviations given by the diagonal matrix $\mathbf{\Sigma}$.

At scales of the ocean mesoscale and larger, the dominant dynamical balances for the circulation initial condition increments, $\delta\mathbf{x}(t_0) = \mathbf{x}(t_0) - \mathbf{x}^b(t_0)$, are geostrophic balance and hydrostatic balance, while the properties of characteristic water masses provide useful correlations between T and S (Ricci et al., 2005; Weaver et al., 2005). This information can be used to construct informative balance operators $\mathbf{K}_b$ for the initial condition background error covariance matrix $\mathbf{B}_x$. From (2), similar balance requirements can be imposed on the increments $\delta\mathbf{b}(t) = \mathbf{b}(t) - \mathbf{b}^b(t)$ described by $\mathbf{B}_b$ at open boundaries since the open boundary conditions are typically derived from another model calculation in a larger model domain. Surface forcing is usually derived from operational atmospheric forecast model products for which estimates of the analysis error may be available, and such information may prove useful for characterizing the error statistics of surface forcing increments, $\delta\mathbf{f}(t) = \mathbf{f}(t) - \mathbf{f}^b(t)$, described by $\mathbf{B}_f$. Similarly, the dominant dynamical balances of the generally stable atmospheric marine boundary layers that control the fluxes of momentum, heat, and freshwater across the air-sea interface can also be used as process model priors for the $\mathbf{K}_b$ component of $\mathbf{B}_f$. In the current ROMS 4D-Var systems, we account only for the balance constraints on the initial condition and model error increments.

Various approaches for modeling error covariance matrices are documented in the scientific literature, and each has associated advantages and drawbacks. The method used in ROMS follows the diffusion operator approach of Weaver and Courtier (2001) and is summarized next, although there is nothing to preclude the use of alternative approaches in the future.

5.1. The background error correlations, $\mathbf{C}$

It is well known that the action of a correlation matrix $\mathbf{C}$ on a vector $\mathbf{x}$ can be expressed as the solution of a diffusion equation (see Egbert et al. (1994), Weaver and Courtier (2001), and Bennett (2002), p. 64, for a detailed treatment). Specifically, consider the two-dimensional diffusion equation for a scalar quantity θ:

$$\partial\theta/\partial\tau = \kappa\nabla^2\theta \tag{17}$$

where κ is a spatially invariant diffusion coefficient, and ∇^2 is the Laplacian operator in Cartesian coordinates. If (17) is discretized on the model grid, and the grid point values arranged as a vector $\boldsymbol{\theta}$, solutions of (17) over the interval $\tau = [0, \tau_d]$ can be represented as $\boldsymbol{\theta}(\tau) = (4\pi\kappa\tau)^{-\frac{1}{2}}\mathbf{C}\boldsymbol{\theta}(0)$, where the correlation matrix $\mathbf{C}$ is associated with a time invariant, isotropic, homogeneous, Gaussian correlation function, with a squared correlation length scale $L^2 = 2\kappa\tau_d$. The initial conditions $\boldsymbol{\theta}(0)$ are the field on which the correlation matrix operates, appropriately scaled to preserve the properties of a correlation function (see below). Generalizations of the solution of (17) on a sphere are considered by Weaver and Courtier (2001) who show that $L^2 \approx 2\kappa\tau_d$ still holds to good approximation.

In three dimensions, it is usual to factorize the correlation matrix so that $\mathbf{C} = \mathbf{C}_h\mathbf{C}_v$ assuming that the horizontal ($\mathbf{C}_h$) and vertical ($\mathbf{C}_v$) correlation matrices are separable. This is typically done for computational convenience although there is no overwhelming evidence to suggest that the correlations of the real ocean are separable. Nonetheless, $\mathbf{C}_v$ can be modeled in the same way as $\mathbf{C}_h$ by solving a 1-dimensional diffusion equation. In the case of the open boundary conditions for ROMS, the horizontal correlation of the background error covariance matrix $\mathbf{B}_b$ is replaced by a 1D correlation along the boundary. If time dependence of the correlation matrices in (6) is treated as separable from the spatial correlations, it can be modeled as an autoregression process (Bennett, 2002, p. 65). However, as noted earlier, time dependent correlations are not yet a part of ROMS 4D-Var.

Eqs. (6) and (7) for the cost (penalty) function suggest that the inverse covariance matrices are required. However, in practice this is usually not the case; in the space of the control vector the need to invert $\mathbf{D}$ is circumvented by a transformation of variable (see Section 6.1), while in observation space the best estimate given by (9) involves the solution of a linear equation involving $\mathbf{D}$ not its inverse (see Section 4).

The practical implementation of (17) is somewhat involved because of the grid dependence of the calculation. Since the model grid spacing is usually not uniform in the horizontal and vertical, it is necessary to account for variations in the grid cell dimensions when modeling $\mathbf{C}_h$ and $\mathbf{C}_v$ in order to preserve the properties of a correlation function. In practice, $\mathbf{C}_h$ and $\mathbf{C}_v$ are further factorized following Weaver and Courtier (2001) so that:

$$\mathbf{C}_h = \mathbf{\Lambda}_h\mathbf{L}_h^{\frac{1}{2}}\mathbf{W}_h^{-1}\left(\mathbf{L}_h^{\frac{1}{2}}\right)^T\mathbf{\Lambda}_h = \mathbf{C}_h^{\frac{1}{2}}\left(\mathbf{C}_h^{\frac{1}{2}}\right)^T \tag{18}$$

$$\mathbf{C}_v = \mathbf{\Lambda}_v\mathbf{L}_v^{\frac{1}{2}}\mathbf{W}_v^{-1}\left(\mathbf{L}_v^{\frac{1}{2}}\right)^T\mathbf{\Lambda}_v = \mathbf{C}_v^{\frac{1}{2}}\left(\mathbf{C}_v^{\frac{1}{2}}\right)^T \tag{19}$$

where $\mathbf{W}$ is a diagonal matrix with elements corresponding to the grid box areas in the case of $\mathbf{W}_h$ and level thicknesses in the case of $\mathbf{W}_v$; $\mathbf{L}$ represents the action of the matrix obtained by solving either a 1D (for $\mathbf{L}_v$) or 2D (for $\mathbf{L}_h$) diffusion equation; and $\mathbf{\Lambda}$ is a diagonal matrix of normalization coefficients required to ensure that the range of $\mathbf{C}_h$ and $\mathbf{C}_v$ is ±1. A square root factorization is used in (18) and (19) to ensure that $\mathbf{C}_h$ and $\mathbf{C}_v$ remain symmetric in the presence of the inevitable rounding errors that occur during each operation, and the square roots are achieved by integrating (17) over the interval $\tau = [0, \tau_d/2]$.

The most costly part of covariance modeling is the evaluation of the normalization factors $\mathbf{\Lambda}$. However, if the horizontal and vertical correlation lengths do not change from one assimilation cycle to the next, $\mathbf{\Lambda}$ need only be computed once. Determination of the exact normalization factors is generally computationally demanding but possible, and in ROMS the elements of $\mathbf{\Lambda}$ can also be estimated using the randomization method introduced by Fisher and Courtier (1995).

At the present time, ROMS 4D-Var supports only homogeneous, isotropic correlation functions. However, these assumptions can be relaxed by replacing $\kappa\nabla^2\theta$ in (17) with a diffusion tensor formulation and a coordinate rotation as demonstrated by Weaver and Courtier (2001). This option will be available in a future release of ROMS 4D-Var. An alternative approach based on the Kronecker product is discussed in Li et al. (2008).

5.2. The linear balance operator $\mathbf{K}_b$

In ROMS, the set of linear balance relationships relating the balanced ($\delta\mathbf{x}_B$) and unbalanced ($\delta\mathbf{x}_U$) components of the analysis increments $\delta\mathbf{x}^a$ parallel those described by Weaver et al. (2005). Because of the controlling influence of temperature on the ocean circulation via density over much of the ocean (except perhaps at the low temperatures of the deep ocean or high latitudes), all of the balance relations between the variables are based on δT, the temperature increment, and, after all, temperature observations are typically the most abundant observation type in the ocean. For practical purposes, the resulting balance operator $\mathbf{K}_b$ is a lower block triangular matrix which has the advantage that it is easy to invert. Recalling that $\delta\mathbf{x} = (\delta T, \delta S, \delta\zeta, \delta u, \delta v)^T$, Weaver et al. (2005) show that the linear balance equations can be written as:

$$\begin{aligned}
\delta T^k &= \delta T^k \\
\delta S^k &= \mathbf{K}_{ST}^{k-1}\delta T^k + \delta S_U \\
\delta\zeta^k &= \mathbf{K}_{\zeta T}\delta T^k + \mathbf{K}_{\zeta S}\delta S^k + \delta\zeta_U \\
\delta u^k &= \mathbf{K}_{uT}\delta T^k + \mathbf{K}_{uS}\delta S^k + \mathbf{K}_{u\zeta}\delta\zeta^k + \delta u_U \\
\delta v^k &= \mathbf{K}_{vT}\delta T^k + \mathbf{K}_{vS}\delta S^k + \mathbf{K}_{v\zeta}\delta\zeta^k + \delta v_U
\end{aligned}$$

where k refers to the number of the outer-loops, and $\mathbf{K}_{xy}$ represents the linear balance relation between variable x and y. The balance $\mathbf{K}_{ST}^{k-1}$ is based on the water mass properties of the ocean state $\mathbf{x}^b + \delta\mathbf{x}^{k-1}$ of the previous outer-loop, while $\mathbf{K}_{\zeta T}$ and $\mathbf{K}_{\zeta S}$ are derived from the density anomalies computed from the linearized equation of state $\delta\rho^k = -\alpha^{k-1}\delta T^k + \beta^{k-1}\delta S^k$. Refinements of this approach are discussed by Haines et al. (2006) and Balmaseda et al. (2008).

Following Weaver et al. (2005), it is assumed that the balanced and unbalanced components of ζ correspond to the baroclinic and barotropic components of the circulation respectively. The baroclinic contribution to the free surface height, (denoted ζ_{ref}), during outer-loop k satisfies:

$$\begin{aligned}
\nabla\cdot H\nabla\zeta_{\text{ref}} = &-\nabla\cdot\int_{-H}^{0}\int_{z}^{0}\nabla\rho^k(z')/\rho_0 dz'dz \\
&-\nabla\cdot\int_{-H}^{0}(\mathbf{u}^k\cdot\nabla u^k\mathbf{i}+\mathbf{u}^k\cdot\nabla v^k\mathbf{j})dz+\nabla\cdot\int_{-H}^{0}\mathbf{F}^k dz
\end{aligned} \tag{20}$$

where $\mathbf{u}$ is the velocity vector, $\mathbf{i}$ and $\mathbf{j}$ are unit vectors in the zonal and meridional directions, $\mathbf{F}$ represents the net influence of forcing and dissipation, and H is the ocean depth (Fukumori et al., 1998). The field ζ_{ref} is computed in ROMS using a biconjugate gradient (BCG) method to solve (20), and can be expressed as $\zeta_{\text{ref}} = \mathcal{B}(\mathbf{x}_U^k)$, where $\mathcal{B}$ describes the BCG procedure. The function $\mathcal{B}$ is nonlinear since it involves dot-products of the BCG vectors, and because the second term on the rhs of (20) is nonlinear. According to the assumptions of the incremental approach to 4D-Var, the balanced increments $\delta\zeta_B \ll \zeta_{\text{ref}}$, where $\delta\zeta_B \simeq (\partial\mathcal{B}(\mathbf{x})/\partial\mathbf{x}|_{\mathbf{x}_B^k})\delta\mathbf{x}$ and $\partial\mathcal{B}(\mathbf{x})/\partial\mathbf{x}|_{\mathbf{x}_B^k}$ denotes the tangent linearization of the BCG algorithm and solves a tangent linear form of (20). The current default in ROMS 4D-Var is to retain only the first term on the rhs of (20), and this is the case for the experiments presented in the companion papers Moore et al. (in press-a, in press-b). However in general, the other terms may be retained in the tangent linearization $\partial\mathcal{B}(\mathbf{x})/\partial\mathbf{x}|_{\mathbf{x}_B^k}$ at the discretion of the user, depending on the importance of the contributions of each term on the rhs of (20).

The balance relations $\mathbf{K}_{xT}$, $\mathbf{K}_{xS}$ and $\mathbf{K}_{x\zeta}$ for $x = u$ and $x = v$ are computed based on the assumption of geostrophic balance, while $\partial\mathcal{B}(\mathbf{x})/\partial\mathbf{x}$ describes any ageostrophic contributions that may be important to the balance. According to (16), the transpose of the balance operator is required when computing the action of $\mathbf{B}$ on the model state-vector, so the adjoint of the BCG algorithm, $(\partial\mathcal{B}(\mathbf{x})/\partial\mathbf{x})^T|_{\mathbf{x}_B^k}$, is also used.

6. Preconditioning

As described in Section 4, the primal and dual forms of 4D-Var are equivalent to minimizing a function of the form $I(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T\mathbf{A}\mathbf{s} - \mathbf{s}^T\mathbf{h} + \mathbf{c}$. For I4D-Var, I is the cost (penalty) function (7), while for 4D-PSAS and R4D-Var, I is an auxiliary function. In either case, $\mathbf{A}$ is a symmetric, positive-definite matrix with orthogonal eigenvectors that represent linearly independent directions in the space defined by $\mathbf{A}$. In both control space or observation space, $I(\mathbf{s})$ represents a parabola with an aspect ratio determined by the eigenvalues of $\mathbf{A}$. If the eigenvalues are widely separated, then $I(\mathbf{s})$ will be characterized by a parabola that is steep-sided in some directions, and gently sloping in others. In this situation, $\mathbf{A}$ is considered to be poorly conditioned, and the conjugate gradient algorithm will generally converge rapidly in the directions associated with the largest eigenvalues of $\mathbf{A}$, and slowly in other directions, particularly if the largest and smallest eigenvalues differ by many orders of magnitude.

If $\mathbf{A}$ is an $(N \times N)$ matrix then the conjugate gradient algorithm is guaranteed with exact arithmetic to converge to the minimum of I in at most N iterations (Golub and van Loan, 1989). The goal of preconditioning is to achieve convergence in $\ll N$ iterations, which can be facilitated by transforming $\mathbf{A}$ into a new matrix $\mathbf{X}^T\mathbf{A}$ that yields the same minimum solution for I but has clustered eigenvalues and a smaller condition number (i.e. the ratio of the largest to the smallest eigenvalues of $\mathbf{X}^T\mathbf{A}$). In this case the aspect ratio of the parabolic surface I is very similar in many or all directions. The preconditioning matrix $\mathbf{X}$ is symmetric, positive-definite and can be factorized as $\mathbf{X} = \mathbf{U}\mathbf{U}^T$, where $\mathbf{U}$ is usually referred to as the square-root preconditioner. As noted in Section 4, identifying the minimum of $I(\mathbf{s})$ corresponds to solving the linear system $\mathbf{A}\mathbf{s} = \mathbf{h}$, so by defining the change of variable $\mathbf{s} = \mathbf{U}\hat{\mathbf{s}}$, the preconditioned system can be written as $\mathbf{U}^T\mathbf{A}\mathbf{U}\hat{\mathbf{s}} = \mathbf{U}^T\mathbf{h}$. It is beyond the scope of this paper to discuss the required properties of preconditioners, but there are many excellent treatises on the subject (e.g. Golub and van Loan, 1989; Benzi, 2002; Tshimanga, 2007; Tshimanga et al., 2008).

6.1. First-level preconditioning

In 4D-Var, it is customary to apply two levels of preconditioning when minimizing $I(\mathbf{s})$. In the case of I4D-Var, $I(\mathbf{s}) \equiv J(\delta\mathbf{z})$ given by (7), and $\mathbf{A}$ is the Hessian matrix, $\mathcal{H}$, of J. A beneficial first-level of preconditioning is a transformation of variable $\mathbf{v} = \mathbf{D}^{-\frac{1}{2}}\delta\mathbf{x}_U$ via the *prior* error covariance matrix (Lorenc, 2003), where $\delta\mathbf{x}_U$ are the unbalanced components of the increment (Weaver et al., 2005), which are assumed to be mutually uncorrelated. With this transformation all of the transformed variables are then of the same order, and the Hessian in $\mathbf{v}$-space becomes $\mathcal{H}_v = (\mathbf{I} + \mathbf{D}^{\frac{1}{2}}\mathbf{G}^T\mathbf{R}^{-1}\mathbf{G}\mathbf{D}^{\frac{1}{2}})$. Clearly, the lower bound for eigenvalues of $\mathcal{H}_v$ is *1*, and if the number of observations N_{obs} is small compared to the dimension of $\mathcal{H}_v$ (the usual case in oceanographic applications), the eigenvalues will be clustered around *1*, and the first-level preconditioner $\mathbf{U} = \mathbf{D}^{-\frac{1}{2}}$ possesses the desired quality. In general, it is necessary to evaluate the action of both $\mathbf{U}$ and $\mathbf{U}^{-1}$ on a vector. However, as shown by Courtier et al. (1994), if $\delta\mathbf{x} \equiv \mathbf{v} = 0$ is chosen as the starting guess for each outer-loop of I4D-Var, then only the action of $\mathbf{U}^{-1} = \mathbf{D}^{\frac{1}{2}}$ on the state-vector is required and can be readily computed using the methods described in Section 5.

For 4D-PSAS and R4D-Var, $\mathbf{A} \equiv (\mathbf{G}\mathbf{D}\mathbf{G}^T + \mathbf{R})$, and the first-level preconditioner is chosen according to the change of variable $\mathbf{v} = \mathbf{R}^{\frac{1}{2}}\mathbf{w}$ (Courtier, 1997) which transforms the problem into one of solving $\left(\mathbf{R}^{-\frac{1}{2}}\mathbf{G}\mathbf{D}\mathbf{G}^T\mathbf{R}^{-\frac{1}{2}} + \mathbf{I}\right)\mathbf{v} = \mathbf{R}^{-\frac{1}{2}}\mathbf{d}$ with a lower bound of 1 for the eigenvalues.

6.2. Second-level preconditioning

The convergence of the conjugate gradient algorithm to the minimum of $I(\mathbf{s})$ may be further accelerated by a second level of preconditioning applied to $\mathbf{v}$. Two second-level preconditioners are available in ROMS: the first is based on estimates of the eigenvectors of $\mathbf{A}$ and follows Fisher and Courtier (1995), and the second is based on the Ritz vectors of $\mathbf{A}$ and parallels Tshimanga et al. (2008). Tshimanga (2007) refers to these approaches as "spectral" preconditioning and "Ritz" preconditioning respectively, and we adopt the same nomenclature here.

Estimates of the eigenvectors and Ritz vectors of $\mathbf{A}$ can be computed with little extra computational effort during execution of the

Lanczos algorithm described in Section 4 (Golub and van Loan, 1989). Consider the eigenpairs $(\theta_i, \mathbf{y}_i)$ of the tridiagonal matrix $\mathbf{T}_m$ in (14) after m inner-loops of either I4D-Var, 4D-PSAS or R4D-Var. Post-multiplying (14) by the eigenvector $\mathbf{y}_i$ shows that:

$$\mathbf{A}\hat{\mathbf{y}}_i = \theta_i \hat{\mathbf{y}}_i + \gamma_m \mathbf{q}_{m+1}(\mathbf{e}_m^T \mathbf{y}_i)$$

where $\hat{\mathbf{y}}_i = \mathbf{V}_m \mathbf{y}_i$. Clearly, $(\theta_i, \hat{\mathbf{y}}_i)$ are good approximations of the eigenpairs of $\mathbf{A}$ when $\mu_i = \|\gamma_m \mathbf{q}_{m+1}(\mathbf{e}_m^T \mathbf{y}_i)\|$ is small, and as m approaches the dimension of $\mathbf{A}$, then $\mu_i \to 0$ and the eigenpairs $(\theta_i, \hat{\mathbf{y}}_i)$ are exact.

The approximate Hessian eigenpairs $(\theta_i, \mathbf{y}_i)$ are also referred to as Ritz pairs, and Fisher and Courtier (1995) show how they can be used as effective second-level preconditioners during 4D-Var. Following the nomenclature and notation of Tshimanga (2007) and Tshimanga et al. (2008), spectral preconditioning is achieved by a change of variable $\mathbf{v} = \mathbf{U}\mathbf{u}$, where:

$$\mathbf{U}_j = \prod_{i=1}^{m} \left(\mathbf{I} - \left(1 - \left(\theta_i^j\right)^{\frac{1}{2}} \right) \hat{\mathbf{y}}_i^j \left(\hat{\mathbf{y}}_i^j\right)^T \right)$$

$$\mathbf{U}_j^{-1} = \prod_{i=m}^{1} \left(\mathbf{I} - \left(1 - \left(\theta_i^j\right)^{-\frac{1}{2}} \right) \hat{\mathbf{y}}_i^j \left(\hat{\mathbf{y}}_i^j\right)^T \right)$$

Here $\left(\theta_i^j, \mathbf{y}_i^j\right)$ are the Ritz pairs that arise from each sequence of inner-loops $i = 1, \ldots, m$ during outer-loop j. Therefore, each outer-loop yields a new second-level preconditioner $\mathbf{U}_j$ that can be used to precondition subsequent outer-loops, and the resulting sequence of preconditioners $\mathbf{U}_j$, $j = 1, 2, \ldots, k-1$ is applied sequentially during outer-loop k via the change of variable $\mathbf{v}^k = \prod_{j=1}^{k-1} \mathbf{U}_j \mathbf{u}^k$. However, as Tshimanga et al. (2008) demonstrate, spectral preconditioning is only effective if μ_i is sufficiently small. In practice, only those Ritz vectors for which $\epsilon_i = \left\| \mathbf{A}\hat{\mathbf{y}}_i^j - \theta_i^j \hat{\mathbf{y}}_i^j \right\| / \theta_1^j = \mu_i / \theta_1^j$ is sufficiently small should be used, otherwise the second-level preconditioner may actually degrade the convergence of the inner-loops.[2]

As an alternative, Tshimanga et al. (2008) propose Ritz preconditioning for which the error μ_i associated with each $\hat{\mathbf{y}}_i^j$ is formally included in $\mathbf{U}_j$. Following Tshimanga (2007), the associated change of variable $\mathbf{v}^k = \prod_{j=1}^{k-1} \mathbf{U}_j \mathbf{u}^k$ is given by:

$$\mathbf{U}_j = \prod_{i=1}^{m} \left(\mathbf{I} - \left(1 - \left(\theta_i^j\right)^{\frac{1}{2}} \right) \hat{\mathbf{y}}_i^j \left(\hat{\mathbf{y}}_i^j\right)^T + \left(\theta_i^j\right)^{-\frac{1}{2}} \left(\mathbf{e}_m^T \mathbf{y}_i^j\right) \gamma_m \hat{\mathbf{y}}_i^j \mathbf{q}_{m+1}^T \right)$$

$$\mathbf{U}_j^{-1} = \prod_{i=m}^{1} \left(\mathbf{I} - \left(1 - \left(\theta_i^j\right)^{-\frac{1}{2}} \right) \hat{\mathbf{y}}_i^j \left(\hat{\mathbf{y}}_i^j\right)^T - \left(\theta_i^j\right)^{-1} \left(\mathbf{e}_m^T \mathbf{y}_i^j\right) \gamma_m \hat{\mathbf{y}}_i^j \mathbf{q}_{m+1}^T \right)$$

Tshimanga et al. (2008) demonstrate that Ritz preconditioning is less restrictive and more forgiving than spectral preconditioning because even using relatively inaccurate Ritz vectors, as measured by ϵ_i, can lead to a significant reduction in the number of inner-loops required to reduce $I(\mathbf{s})$ to a given level.[2]

7. Tools for 4D-Var post-processing

A number of very useful tools are available for post-processing the output of ROMS 4D-Var. These include estimates of the analysis (*posterior*) error variance, empirical orthogonal functions (EOFs) of the posterior error covariance matrix, and the adjoint of the entire 4D-PSAS and R4D-Var system for computing the impact of observations on the analysis-forecast system as well as the sensitivity to variations in the observations.

7.1. Analysis error estimates

If we denote by $\mathbf{z}^t$ the vector describing the true state vector, surface forcing, and boundary conditions, then the analysis error covariance matrix is given by $\mathbf{E}^a = E((\mathbf{z}^b + \delta\mathbf{z}^a - \mathbf{z}^t)(\mathbf{z}^b + \delta\mathbf{z}^a - \mathbf{z}^t)^T)$. Proceeding from (8), $\mathbf{E}^a$ can be expressed as:

$$\mathbf{E}^a = (\mathbf{D}^{-1} + \mathbf{G}^T \mathbf{R}^{-1} \mathbf{G})^{-1} \tag{21}$$

which is the inverse of the Hessian of the cost (penalty) function given by (10). Alternatively, proceeding from the dual form (9), the analysis error covariance can be written as:

$$\mathbf{E}^a = (\mathbf{I} - \mathbf{K}\mathbf{G})\mathbf{D}(\mathbf{I} - \mathbf{K}\mathbf{G})^T + \mathbf{K}\mathbf{R}\mathbf{K}^T \tag{22}$$

where $\mathbf{K} = \mathbf{D}\mathbf{G}^T(\mathbf{G}\mathbf{D}\mathbf{G}^T + \mathbf{R})^{-1}$ is the Kalman gain matrix.

The dimension of $\mathbf{E}^a$ is very large and it is computationally prohibitive to calculate and save the full matrix. However, the Lanczos vector expansions of the Hessian in (10) and the stabilized representer matrix in (9) can be used to compute a reduced rank approximation to $\mathbf{E}^a$. According to Section 4, a reduced rank approximation $\widetilde{\mathbf{E}}^a$ of $\mathbf{E}^a$ in (21) resultng from the primal formulation can be expressed as:

$$\widetilde{\mathbf{E}}_M^a = \mathbf{D}^{\frac{1}{2}} \mathbf{V}_p \mathbf{T}_p^{-1} \mathbf{V}_p^T \left(\mathbf{D}^{\frac{1}{2}}\right)^T \tag{23}$$

where $\mathbf{V}_p$ is the matrix of primal Lanczos vectors of the Hessian $\mathcal{H}$ resulting from a single outer-loop and M preconditioned inner-loops of I4D-Var, and $\mathbf{T}_p$ is the associated $M \times M$ tridiagonal matrix. Alternatively, using (22) and the dual formulation:

$$\widetilde{\mathbf{E}}_M^a = (\mathbf{I} - \widetilde{\mathbf{K}}\mathbf{G})\mathbf{D} \tag{24}$$

$$= \left(\mathbf{I} - \mathbf{D}\mathbf{G}^T \mathbf{R}^{-\frac{1}{2}} \mathbf{V}_d \mathbf{T}_d^{-1} \mathbf{V}_d^T \left(\mathbf{R}^{-\frac{1}{2}}\right)^T \mathbf{G} \right) \mathbf{D} \tag{25}$$

where $\widetilde{\mathbf{K}} = \mathbf{D}\mathbf{G}^T \mathbf{R}^{-\frac{1}{2}} \mathbf{V}_d \mathbf{T}_d^{-1} \mathbf{V}_d^T \left(\mathbf{R}^{-\frac{1}{2}}\right)^T$, hereafter referred to as the practical gain matrix, is an approximation of the true gain matrix $\mathbf{K}$; $\mathbf{V}_d$ is the matrix of dual Lanczos vectors of the stabilized representer matrix resulting from a single outer-loop and M preconditioned inner-loops of 4D-PSAS or R4D-Var, and $\mathbf{T}_d$ is the associated $M \times M$ tridiagonal matrix. Eq. (24) is simpler than the Joseph form (22) because the space spanned by the uncomputed Lanczos vectors in the sequence (14) for $m > M$ is orthogonal to the space spanned by $\mathbf{V}_d$. If the dual 4D-Var is run to convergence with $M = N_{obs}$ then $\widetilde{\mathbf{K}} = \mathbf{K}$ and (24) has a familiar form (Daley, 1991).

The diagonal elements of $\widetilde{\mathbf{E}}_M^a$ represent estimates of the expected analysis (*posterior*) error variance and can be readily computed from (23) or (25) during ROMS 4D-Var. In fact any column (or row) of $\widetilde{\mathbf{E}}_M^a$ can be computed to yield *posterior* cross-covariance information as well. In addition, the leading eigenpairs, so called principal components and empirical orthogonal functions (EOFs), can also be readily computed during 4D-Var using (23) or (25) and yield information about patterns of uncertainty in the estimate $\hat{\mathbf{z}} = \mathbf{z}^b + \delta\mathbf{z}^a$.

7.2. Observation impact

It is of considerable interest to determine which observations or observation platforms have the greatest impact on the circulation estimate during an analysis or forecast cycle. This has recently attracted considerable attention in meteorology for routine monitoring of observing systems (Langland and Baker, 2004; Cardinali et al., 2004; Zhu and Gelaro, 2008; Daescu, 2008). With this in mind, consider the scalar differentiable function $\mathcal{J}(\mathbf{x}(t_i))$ that quantifies some aspect of the circulation (e.g. transport, heat content, forecast error) during an analysis-forecast cycle. For illustration, we will consider here a function of $\mathbf{x}$ at a single time t_i, although functions involving $\mathbf{x}$ at multiple times (e.g. time

[2] The same is true in ROMS as described at http://myroms.org/wiki/index.php/Variational_Data_Assimilation.

integrals) can also be used as described in Moore et al. (in press-b). The increment in $\mathcal{J}$ due to assimilating observations is given by $\Delta\mathcal{J} = \mathcal{J}(\hat{\mathbf{x}}(t_i)) - \mathcal{J}(\mathbf{x}^b(t_i))$ where $\hat{\mathbf{x}}(t_i) = \mathbf{x}^a(t_i)$ if t_i lies within the analysis cycle, and $\hat{\mathbf{x}}(t_i) = \mathbf{x}^f(t_i)$ if t_i lies within the forecast cycle.

According to (8) and (9), $\hat{\mathbf{z}} = \mathbf{z}^b + \widetilde{\mathbf{K}}\mathbf{d}$ where $\widetilde{\mathbf{K}}$ is the practical gain matrix introduced in Section 7.1. Assuming that the analysis increments $\widetilde{\mathbf{K}}\mathbf{d}$ are small compared to $\mathbf{z}^b$, then using the tangent linear assumption, $\hat{\mathbf{x}}(t_i) = \mathbf{x}^b(t_i) + \boldsymbol{\mathcal{M}}(t_i, t_0)\widetilde{\mathbf{K}}\mathbf{d}$ where $\boldsymbol{\mathcal{M}}(t_i, t_0)$ is the tangent linear operator defined in the appendix. The increment $\Delta\mathcal{J}$ can then be written as:

$$\begin{aligned}\Delta\mathcal{J} &\simeq \mathcal{J}(\mathbf{x}^b(t_i) + \boldsymbol{\mathcal{M}}(t_i, t_0)\widetilde{\mathbf{K}}\mathbf{d}) - \mathcal{J}(\mathbf{x}^b(t_i)) \\ &\simeq \mathbf{d}^T\widetilde{\mathbf{K}}^T\boldsymbol{\mathcal{M}}^T(t_0, t_i)\partial\mathcal{J}/\partial\mathbf{x}|_{\mathbf{x}^b} = \mathbf{d}^T\mathbf{g}\end{aligned} \tag{26}$$

which follows from a first-order Taylor expansion of $\mathcal{J}(\mathbf{x}(t_i))$. Eq. (26) shows that $\Delta\mathcal{J}$ can be expressed as the projection of the innovation vector $\mathbf{d}$ (via the dot-product) onto the vector $\mathbf{g} = \widetilde{\mathbf{K}}^T\boldsymbol{\mathcal{M}}^T\partial\mathcal{J}/\partial\mathbf{x}|_{\mathbf{x}^b}$. Since both $\mathbf{d}$ and $\mathbf{g}$ have dimension of N_{obs}, the contribution of each observation to $\Delta\mathcal{J}$ can be identified (Errico, 2007). The computation of $\mathbf{g}$ is greatly simplified using the Lanczos vector expansion of $\widetilde{\mathbf{K}}$, in which case $\mathbf{g} = \mathbf{R}^{-1}\mathbf{G}\mathbf{D}^{\frac{1}{2}}\mathbf{V}_p\mathbf{T}_p^{-1}\mathbf{V}_p^T\left(\mathbf{D}^{\frac{1}{2}}\right)^T\boldsymbol{\mathcal{M}}^T\partial\mathcal{J}/\partial\mathbf{x}|_{\mathbf{x}^b}$ in the case of I4D-Var, and $\mathbf{g} = \mathbf{R}^{-\frac{1}{2}}\mathbf{V}_d\mathbf{T}_d^{-1}\mathbf{V}_d^T(\mathbf{R}^{-\frac{1}{2}})^T\mathbf{G}\mathbf{D}^T\boldsymbol{\mathcal{M}}^T\partial\mathcal{J}/\partial\mathbf{x}|_{\mathbf{x}^b}$ in the case of 4D-PSAS and R4D-Var. In any case, the contribution of each observation to $\Delta\mathcal{J}$ can be computed for any ROMS analysis-forecast cycle using the saved Lanczos vectors, although an additional run of the adjoint model, $\boldsymbol{\mathcal{M}}^T$, appropriately forced by $\partial\mathcal{J}/\partial\mathbf{x}|_{\mathbf{x}^b}$, is required. When $\mathcal{J}$ is a function of $\mathbf{x}$ at multiple times $\boldsymbol{\mathcal{M}}^T\partial\mathcal{J}/\partial\mathbf{x}|_{\mathbf{x}^b}$ is replaced by a convolution in time $\boldsymbol{\mathcal{M}}^T\star\partial\mathcal{J}/\partial\mathbf{x}|_{\mathbf{x}^b}$. The evaluation of $\mathbf{g}$ also requires an additional integration of the tangent linear model, $\mathbf{G}$, sampled at the observation points, so the computation of $\mathbf{g}$ requires the same computational effort as a single inner-loop, as illustrated in Fig. 4.

7.3. Observation sensitivity

The sensitivity of the circulation analyses arising from 4D-Var to changes in the observations or observation array is also of considerable interest. The best estimate circulation increments are given by (8) or (9), and in both cases the matrix inverse of the Hessian $\boldsymbol{\mathcal{H}}$ or stabilized representer matrix is solved using the Lanczos algorithm as described in Section 4. The Lanczos algorithm is a nonlinear function of the innovation vector $\mathbf{d}$ in the sense that it involves the computation and manipulation of dot-products of linear functions of $\mathbf{d}$. We can therefore express (8) and (9) generically as:

$$\delta\mathbf{z}^a = \mathcal{K}(\mathbf{d}) \tag{27}$$

where the nonlinear function $\mathcal{K}(\mathbf{d})$ represents the entire I4D-Var procedure in the case of (8) and 4D-PSAS or R4D-Var in the case of (9).

The innovation vector $\mathbf{d} = \mathbf{y}^o - H(\mathbf{x}^b)$, and a change $\delta\mathbf{y}^o$ in the observations will yield the first-order change $(\partial\mathcal{K}/\partial\mathbf{d})|_{\mathbf{x}^b}\delta\mathbf{y}^o$ in the control vector, where $(\partial\mathcal{K}/\partial\mathbf{d})|_{\mathbf{x}^b}$ represents the tangent linearization of 4D-Var. Consider again the scalar function $\mathcal{J}(\mathbf{x}(t_i))$, introduced in Section 7.2, describing some aspect of the circulation. To first-order, the change in $\mathcal{J}$ due to a change $\delta\mathbf{y}^o$ in the observations is given by:

$$\delta\mathcal{J} = \delta\mathbf{y}^{oT}\left(\frac{\partial\mathcal{K}}{\partial\mathbf{d}}\right)^T\Bigg|_{\mathbf{z}^b}\boldsymbol{\mathcal{M}}^T(t_0, t_i)|_{\mathbf{z}^b}\star\left(\frac{\partial\mathcal{J}}{\partial\mathbf{x}}\right)\Bigg|_{\mathbf{x}^a} \equiv \delta\mathbf{y}^{oT}(\partial\mathcal{J}/\partial\mathbf{y}^o) \tag{28}$$

where $(\partial\mathcal{J}/\partial\mathbf{y}^o) = (\partial\mathcal{K}/\partial\mathbf{d})^T|_{\mathbf{z}^b}\boldsymbol{\mathcal{M}}^T(t_0, t_i)|_{\mathbf{z}^b}\star(\partial\mathcal{J}/\partial\mathbf{x})|_{\mathbf{x}^a}$ is the sensitivity of $\mathcal{J}$ to variations in the observations. Eq. (28) shows that $(\partial\mathcal{J}/\partial\mathbf{y}^o)$ can be computed by integrating $\boldsymbol{\mathcal{M}}^T\star(\partial\mathcal{J}/\partial\mathbf{x})_{\mathbf{x}^a}$ through the adjoint of 4D-Var. At the present time, however, the adjoint of 4D-Var is only available in ROMS for 4D-PSAS and R4D-Var.

It is important to realize that the change in $\mathcal{J}$ given by (28) is fundamentally different to that described by the observation impact of (26) in Section 7.2. In the case of the observation impact, (26) quantifies the contribution of each individual observation to the actual increment $\Delta\mathcal{J}$ that results from assimilating a given set of observations. Conversely, $\delta\mathcal{J}$ in (28) is the expected change in $\mathcal{J}$ based on the observation sensitivity $(\partial\mathcal{J}/\partial\mathbf{y}^o)$ as a result of a change $\delta\mathbf{y}^o$ in the observation values.

8. Summary

This paper describes a comprehensive and unique community regional ocean model 4D-Var analysis system. All of the 4D-Var applications described here run on parallel computer architectures and can be applied to very large computational domains (Broquet et al., 2009a; Broquet et al., 2009b; Broquet et al., 2011). While 4D-Var systems have been developed for other models, as described in

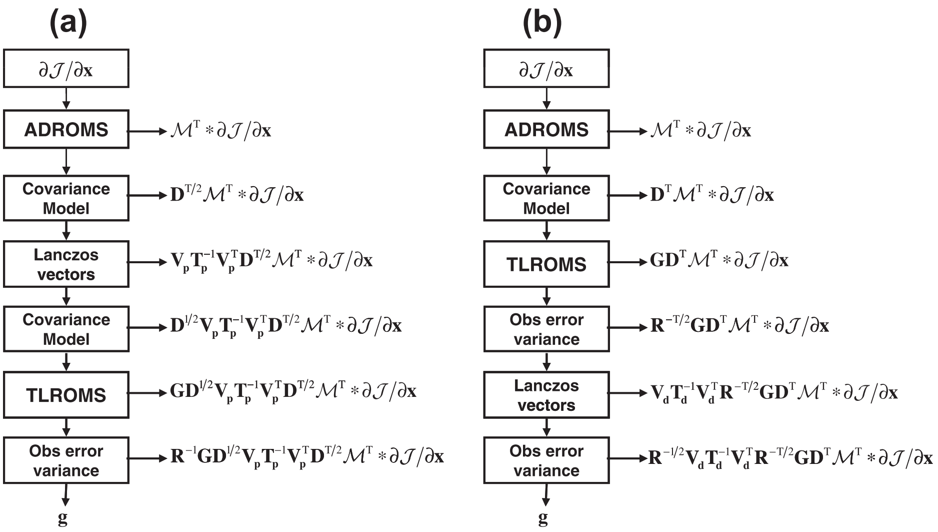
(a)
$\partial\mathcal{J}/\partial\mathbf{x}$
ADROMS
$\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
Covariance Model
$\mathbf{D}^{T/2}\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
Lanczos vectors
$\mathbf{V}_p\mathbf{T}_p^{-1}\mathbf{V}_p^T\mathbf{D}^{T/2}\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
Covariance Model
$\mathbf{D}^{1/2}\mathbf{V}_p\mathbf{T}_p^{-1}\mathbf{V}_p^T\mathbf{D}^{T/2}\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
TLROMS
$\mathbf{G}\mathbf{D}^{1/2}\mathbf{V}_p\mathbf{T}_p^{-1}\mathbf{V}_p^T\mathbf{D}^{T/2}\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
Obs error variance
$\mathbf{R}^{-1}\mathbf{G}\mathbf{D}^{1/2}\mathbf{V}_p\mathbf{T}_p^{-1}\mathbf{V}_p^T\mathbf{D}^{T/2}\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
$\mathbf{g}$
(b)
$\partial\mathcal{J}/\partial\mathbf{x}$
ADROMS
$\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
Covariance Model
$\mathbf{D}^T\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
TLROMS
$\mathbf{G}\mathbf{D}^T\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
Obs error variance
$\mathbf{R}^{-T/2}\mathbf{G}\mathbf{D}^T\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
Lanczos vectors
$\mathbf{V}_d\mathbf{T}_d^{-1}\mathbf{V}_d^T\mathbf{R}^{-T/2}\mathbf{G}\mathbf{D}^T\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
Obs error variance
$\mathbf{R}^{-1/2}\mathbf{V}_d\mathbf{T}_d^{-1}\mathbf{V}_d^T\mathbf{R}^{-T/2}\mathbf{G}\mathbf{D}^T\mathcal{M}^T * \partial\mathcal{J}/\partial\mathbf{x}$
$\mathbf{g}$

Fig. 4. A flow chart illustrating the ROMS observation impact algorithm based on (a) the primal formulation of 4D-Var, and (b) the dual formulation of 4D-Var. The arrows to the right of each box represent the total resulting operation at that stage of the algorithm.

Section 1, the ROMS system offers users the option to apply three 4D-Var approaches to a specific data assimilation application, one in the space of the control vector, Incremental 4D-Var (I4D-Var), and two in observation space, a physical-space statistical analysis system (4D-PSAS) and the indirect representer method (R4D-Var). The choice of approach is problem dependent, and will be dictated by considerations such as strong versus weak constraint, the number of observations, and the linear stability properties of the circulation under investigation.

The control vector space search algorithm of I4D-Var is the one most commonly used by the meteorological community, and is well documented in the literature. Relaxing the strong constraint in I4D-Var is challenging, however, because of the very large increase in the dimension of the problem. For weak constraint problems it is therefore advantageous to search for the best circulation estimate in observation space.

The two approaches to weak constraint data assimilation that have been used are R4D-Var and 4D-PSAS. In the case of R4D-Var, the approach is to formally solve the nonlinear Euler–Lagrange equations for the circulation. This proceeds via a sequence of Picard iterations based on a finite-amplitude linearization of the nonlinear model to generate background circulation estimates. For sufficiently high Rossby number flows, such linearizations can become linearly unstable, at which point the search for the circulation estimate fails (see also Bennett, 2002, Section 5.3.3). However, steps can be taken to stabilize the Picard iterations, such as reducing the length of the assimilation window, or relaxing the solution towards the previous Picard iterate (Chua and Bennett, 2000). The latter necessarily leads to sub-optimal estimates of the circulation but which, nonetheless, may be acceptable. In the case of 4D-PSAS, the iteration procedure is stabilized by using the nonlinear model to generate background circulation estimates instead of the finite-amplitude linearization employed in R4D-Var. While both approaches provide suboptimal circulation estimates in the case of linearly unstable flows, it is by no means clear whether one approach is superior. One advantage of 4D-PSAS, however, is that like I4D-Var the outer-loops will also be stable irrespective of the Rossby number. In all cases however, the linear stability of the inner-loops of 4D-Var cannot be guaranteed, and additional steps may be required to stabilize the algorithm (e.g. reduced model resolution in the inner-loops, simplified inner-loop model physics, increased dissipation, etc.).

A powerful aspect of the ROMS 4D-Var data assimilation algorithms is that they allow adjustments to the surface forcing and open boundary conditions to be made in addition to corrections in the initial conditions. In addition, using 4D-PSAS and R4D-Var the influence of model error can be included in any estimate of the ocean circulation.

Computation of analysis (*posterior*) error estimates is unique to ROMS. A common criticism of 4D-Var methods is that they typically do not provide information about the error or confidence in the resulting circulation estimates. This is because the posterior error covariance matrix $\mathbf{E}^a$ has a very large dimension and is computationally prohibitive to compute. However, for modest additional effort it is possible to compute the diagonal of a reduced rank approximation, $\widetilde{\mathbf{E}}^a$, and its leading eigenvectors which may provide useful information about relative levels of uncertainty in each state variable and the dominant patterns of error.

The observation impact and observation sensitivity features of ROMS are also unique in the oceanographic modeling community, and provide the ability to quantify the impact of individual observations and observation types on the analysis and forecast cycle.

Each of the 4D-Var algorithms described here have been applied to ROMS configured for the California Current System (CCS). Specific case studies that compare the performance of I4D-Var, 4D-PSAS and R4D-Var and the resulting circulation estimates are presented in the companion paper by Moore et al. (in press-a), along with illustrative diagnostic calculations. The second companion paper Moore et al. (in press-b) presents examples of the 4D-Var observation impact and observation sensitivity in the CCS.

While the ROMS 4D-Var system is comprehensive, it is by no means complete, and additional refinements are planned and underway. Some of the most pressing issues include the option for non-isotropic, inhomogeneous correlation functions for the *prior* and observation errors, including correlations in time; implementation of an effective initialization method to suppress gravity waves arising from initialization shocks; and a restart option for each 4D-Var algorithm. It should be clear from the foregoing discussions that the *prior* error and observation error covariances are a critically important and integral component of 4D-Var. The methods used here to model the error covariances may not necessarily be the best, and were chosen because of the considerable experience that already exists within the meteorological and oceanographic communities. However, it is clear that additional flexibility in ROMS to use other approaches, such as EOF based methods, would be desirable. These options will also be considered in the future.

Finally we note that there are few systematic comparisons of the performance characteristics of the primal and dual formulations of 4D-Var, and none that we are aware of for ocean mesoscale circulation environments. ROMS therefore offers the ocean modeling community a comprehensive laboratory in which to explore some of these issues in complex settings, and develop much needed experience and expertise in state-of-the-art ocean data assimilation techniques.

Acknowledgements

The authors are indebted to the unwavering support of the Office of Naval Research (N00014-01-1-0209, N00014-06-1-0406, N00014-08-1-0556, N00014-10-1-0322). Development of R4D-Var was also supported by the National Science Foundation (OCE-0121176, OCE-0121506). Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the National Science Foundation. We are also indebted to the generosity of Mike Fisher at ECMWF for endless discussions and guidance about various aspects of the algorithms described in this paper.

Appendix A

Following the notation of Section 3, the incremental form of the control vector is given by $\delta\mathbf{z} = (\delta\mathbf{x}^T(t_0), \delta\mathbf{f}^T(t_1), \ldots, \delta\mathbf{f}^T(t_k), \ldots, \delta\mathbf{b}^T(t_1), \ldots, \delta\mathbf{b}^T(t_k), \ldots, \boldsymbol{\eta}^T(t_1), \ldots, \boldsymbol{\eta}^T(t_k), \ldots)^T$ where $\delta\mathbf{x}(t_0)$ is the initial condition increment, $\delta\mathbf{f}(t)$ is the surface forcing increment, $\delta\mathbf{b}(t)$ is the open boundary condition increment, and $\boldsymbol{\eta}(t)$ are the corrections for model error. If at each ROMS timestep there are n_m model gridpoint variables, n_f gridpoint surface variables, and n_b gridpoint boundary variables, then $\delta\mathbf{z}$ has the dimensions $n_z \times 1$ where $n_z = n_m + N(n_f + n_b + n_m)$ and N is the number of timesteps in the assimilation interval $[t_0, t_N]$. However, it is often advantageous to reorder the elements of $\delta\mathbf{z}$, and/or isolate particular elements such as the state-vector increment $\delta\mathbf{x}$. With this goal in mind, we introduce the reordered vector $\delta\mathbf{z}' = \mathbf{S}\delta\mathbf{z}$ where $\delta\mathbf{z}' = (\delta\mathbf{x}^T(t_0), \delta\mathbf{f}^T(t_1), \delta\mathbf{b}^T(t_1), \boldsymbol{\eta}^T(t_1), \ldots, \delta\mathbf{f}^T(t_N), \delta\mathbf{b}^T(t_N), \boldsymbol{\eta}^T(t_N))^T$ and $\mathbf{S}$ simply reorders the elements of $\delta\mathbf{z}$ in ascending time. Consider now the matrix $\mathbf{P}(t_i, t_{i-1})$ which transforms the $(n_{i-1} \times 1)$ vector $\mathbf{h}_{i-1}$ into the $(n_i \times 1)$ vector $\mathbf{h}_i$ where $\mathbf{h}_i = (\delta\mathbf{u}^T(t_i), \delta\mathbf{f}^T(t_{i+2}), \delta\mathbf{b}^T(t_{i+2}), \boldsymbol{\eta}^T(t_{i+2}), \ldots, \delta\mathbf{f}^T(t_N), \delta\mathbf{b}_b^T(t_N), \boldsymbol{\eta}^T(t_N))^T$, the vector dimension $n_i = n_z - i(n_f + n_b + n_m)$, and $\delta\mathbf{u}(t_{i-1}) = (\delta\mathbf{x}^T(t_{i-1}), \delta\mathbf{f}^T(t_i), \delta\mathbf{b}^T(t_i), \boldsymbol{\eta}^T(t_i))^T$ introduced in Section 3. Specifically $\mathbf{h}_i = \mathbf{P}(t_i, t_{i-1})\mathbf{h}_{i-1}$ and:

$$\mathbf{P}(t_i,t_{i-1})=\begin{pmatrix}
\mathbf{M}(t_i,t_{i-1}) & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \cdots\\
\mathbf{0} & \mathbf{I}_{n_f} & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\
\cdots & \mathbf{0} & \mathbf{I}_{n_b} & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\
\cdots & \cdots & \mathbf{0} & \mathbf{I}_{n_m} & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \cdots\\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots\\
\cdots & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{I}_{n_f} & \mathbf{0} & \cdots & \cdots & \cdots\\
\cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{I}_{n_b} & \mathbf{0} & \cdots & \cdots\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{0} & \mathbf{I}_{n_m} & \mathbf{0} & \cdots\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots
\end{pmatrix}$$

where $\mathbf{M}(t_i,t_{i-1})$ is the tangent linear model introduced in Section 2 (Eq. (3)); $\mathbf{I}_m$ denotes the $(m \times m)$ identity matrix; each null matrix $\mathbf{0}$ has the appropriate dimensions; and the ellipses denote a continuation of the implied null and identity matrix sequences.

The state-vector increments $\delta\mathbf{x}$ at any time t_i can be isolated according to $\delta\mathbf{x}(t_i)=\mathbf{J}_i\mathbf{h}_i=\mathbf{J}_i\prod_{k=1}^{i}\mathbf{P}(t_k,t_{k-1})\mathbf{S}\delta\mathbf{z}$ where $\mathbf{J}_i=(\mathbf{I}_{n_m}\mathbf{0}_c)$ and $\mathbf{0}_c$ is a null matrix with dimension $(n_m\times(N-i)(n_f+n_b+n_m))$. Similar $\mathbf{J}$ matrices can be defined to isolate $\delta\mathbf{f}$, $\delta\mathbf{b}$ and $\boldsymbol{\eta}$, although these are generally not used here. Just as $\mathbf{J}_i$ isolates the $\delta\mathbf{x}$ component of $\mathbf{h}_i$, it is useful to define an associated tangent linear matrix $\boldsymbol{\mathcal{M}}(t_i,t_0)=\mathbf{J}_i\prod_{k=1}^{i}\mathbf{P}(t_k,t_{k-1})\mathbf{S}$ so that $\delta\mathbf{x}(t_i)=\boldsymbol{\mathcal{M}}(t_i,t_0)\delta\mathbf{z}$, where $\boldsymbol{\mathcal{M}}(t_i,t_0)$ returns the state-vector increment $\delta\mathbf{x}(t_i)$ given a control vector increment $\delta\mathbf{z}$.

References

Balmaseda, M.A., Vidard, A., Anderson, D.L.T., 2008. The ECMWF ocean analysis system: ORA-S3. Monthly Weather Review 136, 3018–3034.

Bengtsson, L., Ghil, M., Kallen, E., 1981. Dynamic Meteorology: Data Assimilation Methods. Springer-Verlag.

Bennett, A.F., 1992. Inverse Methods in Physical Oceanography. Cambridge University Press, Cambridge.

Bennett, A.F., 2002. Inverse Modeling of the Ocean and Atmosphere. Cambridge University Press, Cambridge.

Benzi, M., 2002. Preconditioning techniques for large linear systems a survey. Journal of Computational Physics 182, 418–477.

Broquet, G., Edwards, C.A., Moore, A.M., Powell, B.S., Veneziani, M., Doyle, J.D., 2009a. Application of 4D-variational data assimilation to the California Current System. Dynamics of Atmospheres and Oceans 48, 69–91.

Broquet, G., Moore, A.M., Arango, H.G., Edwards, C.A., Powell, B.S., 2009b. Ocean state and surface forcing correction using the ROMS-IS4DVAR data assimilation system. Mercator Ocean Quarterly Newsletter 34, 5–13.

Broquet, G., Moore, A.M., Arango, H.G., Edwards, C.A., 2011. Corrections to ocean surface forcing in the California Current System using 4D-variational data assimilation. Ocean Modelling 36, 116–132.

Cardinali, C., Pezzulli, S., Andersson, E., 2004. Influence-matrix diagnostic of a data assimilation system. Quarterly Journal of the Royal Meteorological Society 130, 2767–2786.

Chua, B.S., Bennett, A.F., 2001. An inverse ocean modeling system. Ocean Modelling 3, 137–165.

Cohn, S.E., 1997. An introduction to estimation theory. Journal of the Meteorological Society of Japan 75, 257–288.

Courtier, P., 1997. Dual formulation of four-dimensional variational assimilation. Quarterly Journal of the Royal Meteorological Society 123, 2449–2461.

Courtier, P., Talagrand, O., 1987. Variational assimilation of meteorological observations with the adjoint vorticity equation. II: Numerical results. Quarterly Journal of the Royal Meteorological Society 113, 1329–1347.

Courtier, P., Thépaut, J.-N., Hollingsworth, A., 1994. A strategy for operational implementation of 4D-Var using an incremental approach. Quarterly Journal of the Royal Meteorological Society 120, 1367–1388.

Curchitser, E.N., Haidvogel, D.B., Hermann, A.J., Dobbins, E.L., Powell, T.M., Kaplan, A., 2005. Multi-scale modeling of the North Pacific Ocean: assessment and analysis of simulated basin-scale variability (1996–2003). Journal of Geophysical Research 110 (C11021). doi:10.1029/2005JC002902.

Da Silva, A., Pfaendtner, J., Guo, J., Sienkiewicz, M., Cohn, S., 1995. Assessing the effects of data selection with DAO's physical-space statistical analysis system. In: Proceedings of the Second International WMO Symposium on Assimilation of Observations in Meteorology and Oceanography, Tokyo 13–17 March, 1995. WMO.TD 651, pp. 273–278.

Daescu, D.N., 2008. On the sensitivity equations for four-dimensional variational (4D-Var) data assimilation. Monthly Weather Review 136, 3050–3065.

Daley, R., 1991. Atmospheric Data Analysis. Cambridge University Press, Cambridge.

Daget, N., Weaver, A.T., Balmaseda, M.A., 2009. Ensemble estimation of background-error variances in a three-dimensional variational data assimilation system for the global ocean. Quarterly Journal of the Royal Meteorological Society 135, 1071–1094.

Derber, J., 1987. Variational four-dimensional analysis using quasi-geostrophic constraints. Monthly Weather Review 115, 998–1008.

Derber, J., Bouttier, F., 1999. A reformulation of the background error covariance in the ECMWF global data assimilation system. Tellus 51A, 195–221.

Derber, J., Rosati, A., 1989. A global oceanic data assimilation system. Journal of Physical Oceanography 19, 1333–1347.

Di Lorenzo, E., Moore, A.M., Arango, H.G., Cornuelle, B.D., Miller, A.J., Powell, B., Chua, B.S., Bennett, A.F., 2007. Weak and strong constraint data assimilation in the inverse Regional Ocean Modeling System (ROMS): development and application for a baroclinic coastal upwelling system. Ocean Modelling 16, 160–187.

Egbert, G.D., Bennett, A.F., Foreman, M.C.G., 1994. TOPEX/POSEIDON tides estimated using a global inverse method. Journal of Geophysical Research 99, 24821–24852.

Errico, R.M., 2007. Interpretations of an adjoint-derived observational impact measure. Tellus 59A, 273–276.

Fisher, M., Courtier, P., 1995. Estimating the Covariance Matrices of Analysis and Forecast Error in Variational Data Assimilation. ECMWF Technical Memorandum 220, European Centre for Medium Range Weather Forecasts, Shinfield Park, Reading, UK.

Fukumori, I., Raghunath, R., Fu, L., 1998. Nature of global large-scale sea level variability in relation to atmospheric forcing: a modeling study. Journal of Geophysical Research 103, 5493–5512.

Gesztesy, F., Sticka, W., 1998. On the theorem of Picard. Proceedings of the American Mathematical Society 126, 1089–1099.

Ghil, M., Malanotte-Rizzoli, P., 1991. Data assimilation in meteorology and oceanography. Advances in Geophysics 33, 141–266.

Golub, G.H., van Loan, C.F., 1989. Matrix Computations. Johns Hopkins University Press.

Haidvogel, D.B., Arango, H.G., Hedstrom, K., Beckmann, A., Malanotte-Rizzoli, P., Shchepetkin, A.F., 2000. Model evaluation experiments in the North Atlantic basin: simulations in nonlinear terrain-following coordinates. Dynamics of Atmospheres and Oceans 32, 239–281.

Haidvogel, D.B., Arango, H.G., Budgell, W.P., Cornuelle, B.D., Curchitser, E., Di Lorenzo, E., Fennel, K., Geyer, W.R., Hermann, A.J., Lanerolle, L., Levin, J., McWilliams, J.C., Miller, A.J., Moore, A.M., Powell, T.M., Shchepetkin, A.F., Sherwood, C.R., Signell, R.P., Warner, J.C., Wilkin, J., 2008. Ocean forecasting in terrain-following coordinates: formulation and skill assessment of the Regional Ocean Modeling System. Journal of Computational Physics 227, 3595–3624.

Haines, K., Blower, J., Drecourt, J.-P., Liu, C., Vidard, A., Astin, I., Zhou, X., 2006. Salinity using S(T): covariance relationship. Monthly Weather Review 134, 759–771.

Ide, K., Courtier, P., Ghil, M., Lorenc, A.C., 1997. Unified notation for data assimilation: operational, sequential and variational. Journal of the Meteorological Society of Japan 75, 181–189.

Janjic, T., Cohn, S.E., 2006. Treatment of observation error due to unresolved scales in atmospheric data assimilation. Monthly Weather Review 134, 2900–2915.

Kurapov, A.L., Egbert, G.D., Allen, J.S., Miller, R.N., 2009. Representer-based analyses in the coastal upwelling system. Dynamics of Atmospheres and Oceans 48, 198–218.

Lanczos, C., 1950. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of Research of the National Bureau of Standards 45, 255–282.

Langland, R.H., Baker, N., 2004. Estimation of observation impact using the NRL atmospheric variational data assimilation adjoint system. Tellus 56, 189–201.

LeDimet, F.-X., Talagrand, O., 1986. Variational algorithms for analysis and assimilation of meteorological observations. Tellus 38A, 97–110.

Lewis, J., Derber, J., 1985. The use of adjoint equations to solve a variational adjustment problem with advective constraints. Tellus 37, 309–327.

Li, Z., Chao, Y., McWilliams, J.C., Ide, K., 2008. A three-dimensional variational data assimilation scheme for the regional ocean modeling system. Journal of Atmospheric and Oceanic Technology 25, 2074–2090.

Lorenc, A.C., 1986. Analysis methods for numerical weather prediction. Quarterly Journal of the Royal Meteorological Society 112, 1177–1194.

Lorenc, A.C., 2003. Modelling of error covariances by 4D-Var data assimilation. Quarterly Journal of the Royal Meteorological Society 129, 3167–3182.

Marchesiello, P., McWilliams, J.C., Shchepetkin, A.F., 2001. Open boundary conditions for long-term integration of regional oceanic models. Ocean Modelling 3, 1–20.

Moore, A.M., Arango, H.G., Di Lorenzo, E., Cornuelle, B.D., Miller, A.J., Neilson, D.J., 2004. A comprehensive ocean prediction and analysis system based on the tangent linear and adjoint of a regional ocean model. Ocean Modelling 7, 227–258.

Moore, A.M., Arango, H.G., Broquet, G., Edwards, C.A., Veneziani, M., Powell, B.S., Foley, D., Doyle, J.D., Costa, D., Robinson, P., 2011. The Regional Ocean Modeling System (ROMS) 4-dimensional variational data assimilation systems: II Performance and application to the California Current system. Progress in Oceanography 91, 50–73.

Moore, A.M., Arango, H.G., Broquet, G., Edwards, C.A., Veneziani, M., Powell, B.S., Foley, D., Doyle, J.D., Costa, D., Robinson, P., 2011. The Regional Ocean Modeling System (ROMS) 4-dimensional variational data assimilation systems: III Observation impact and observation sensitivity in the California Current system. Progress in Oceanography 91, 74–94.

Muccino, J.C., Luo, H., Arango, H.G., Haidvogel, D.B., Levin, J.C., Bennett, A.F., Chua, B.S., Egbert, G.D., Cornuelle, B.D., Miller, A.J., Di Lorenzo, E., Moore, A.M., Zaron, E.D., . The inverse ocean modeling system. Part II: Applications. Journal of Atmospheric and Oceanic Technology 25, 1623–1637.

Paige, C.C., Saunders, M.A., 1975. Solution of sparse indefinite systems of linear equations. SIAM Journal on Numerical Analysis 12, 617–629.

Powell, B.S., Arango, H.G., Moore, A.M., Di Lorenzo, E., Milliff, R.F., Foley, D., 2008. 4DVAR data assimilation in the intra-Americas Sea with the Regional Ocean Modeling System (ROMS). Ocean Modelling 25, 173–188.

Powell, B.S., Moore, A.M., 2009. Estimating the 4DVAR analysis error from GODAE products. Ocean Dynamics 59, 121–138.

Ricci, S., Weaver, A.T., Vialard, J., Rogel, P., 2005. Incorporating state-dependent temperature-salinity constraints in the background-error covariance of variational ocean data assimilation. Monthly Weather Review 133, 317–338.

Sasaki, Y., 1970. Some basic formulations in numerical variational analysis. Monthly Weather Review 98, 875–883.

Shchepetkin, A.F., McWilliams, J.C., 2003. A method for computing horizontal pressure-gradient force in an oceanic model with a nonaligned vertical grid. Journal of Geophysical Research 108 (C3). doi:10.1029/2001/JC001047.

Shchepetkin, A.F., McWilliams, J.C., 2005. The regional oceanic modeling system (ROMS): a split explicit, free-surface, topography-following-coordinate oceanic model. Ocean Modelling 9, 347–404.

Stammer, D., Wunsch, C., Giering, R., Eckert, C., Heimbach, P., Marotzke, J., Adcroft, A., Hill, C.N., Marshall, J., 2002. The global ocean circulation during 1992–1997 estimated from ocean observations and a general circulation model. Journal of Geophysical Research 107. doi:10.1029/2001JC000888.

Talagrand, O., Courtier, P., 1987. Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. Quarterly Journal of the Royal Meteorological Society 113, 1321–1328.

Talagrand, O., 1997. Assimilation of observations, an introduction. Journal of the Meteorological Society of Japan 75, 191–209.

Thacker, W.C., 1989. The role of the hessian matrix in fitting models to measurements. Journal of Geophysical Research 94, 6177–6196.

Thacker, W.C., Long, R.B., 1988. Fitting dynamics to data. Journal of Geophysical Research 93, 1227–1240.

Tarantola, A., 1987. Inverse Problem Theory: Methods for Data Filtering and Model Parameter Estimation. Elsevier, Amsterdam.

Trémolet, Y., 2006. Accounting for an imperfect model in 4D-Var. Quarterly Journal of the Royal Meteorological Society 132, 2483–2504.

Trémolet, Y., 2007. Model-error estimation in 4D-Var. Quarterly Journal of the Royal Meteorological Society 133, 2671280.

Tshimanga, J., 2007. On a Class of Limited Memory Preconditioners for Large-scale Nonlinear Least-squares Problems. PhD Thesis, Facultés Universitaires Notre-Dame de la Paix, Namur, Belgium.

Tshimanga, J., Gratton, S., Weaver, A.T., Sartenaer, A., 2008. Limited-memory preconditioners with application to incremental variational data assimilation. Quarterly Journal of the Royal Meteorological Society 134, 751–769.

Weaver, A.T., Courtier, P., 2001. Correlation modelling on the sphere using a generalized diffusion equation. Quarterly Journal of the Royal Meteorological Society 127, 1815–1846.

Weaver, A.T., Deltel, C., Machu, E., Ricci, S., Daget, N., 2005. A multivariate balance operator for variational ocean data assimilation. Quarterly Journal of the Royal Meteorological Society 131, 3605–3625.

Weaver, A.T., Vialard, J., Anderson, D.L.T., 2003. Three- and four-dimensional variational assimilation with a general circulation model of the tropical Pacific Ocean. Part I: Formulation, internal diagnostics and consistency checks. Monthly Weather Review 131, 1360–1378.

Wunsch, C., 1996. The Ocean Circulation Inverse Problem. Cambridge University Press, Cambridge.

Wunsch, C., 2006. Discrete Inverse and State Estimation Problems: With Geophysical Fluid Applications. Cambridge University Press, Cambridge.

Zhu, Y., Gelaro, R., 2008. Observation sensitivity calculations using the adjoint of the gridpoint statistical interpolation (GSI) analysis system. Monthly Weather Review 136, 335–351.