


RESEARCH ARTICLE

On the ensemble-based linearization of numerical models

Max Yaremchuk¹  | Dmitri Nechaev² | Sergey Frolov³¹Naval Research Laboratory, Stennis Space Center, Mississippi, USA²Department of Marine Science, University of Southern Mississippi, Hattiesburg, Mississippi, USA³Cooperative Institute for Research in Environmental Sciences, University of Colorado, Boulder, Colorado, USA**Correspondence**Max Yaremchuk, Naval Research Laboratory, 1009 Balch Blvd., Stennis Space Center, MS 39522, USA.
Email: max.yaremchuk@nrlssc.navy.mil**Funding information**

Office of Naval Research National Science Foundation, 0646352N, 0602435N, and 0603207N, 0065342

Abstract

Current parallelization trends in computer technology facilitates development of the algorithms that retrieve linear approximations of the model operators and their adjoints from ensembles of model simulations. In this study we address the problem of obtaining exact linearizations in the presence of semi-implicit numerics of the parent model under realistic constraints on the ensemble size. The method is based on factorization of the model into a sequence of local and non-local linear operators and employs prior information on the structure of the respective sparse matrices. The performance of the method is tested using 28 perturbed solutions of the shallow-water equations with a moderate size (10^4) state vector. Numerical experiments have shown feasibility of the approach under relatively general constraints on the structure of the parent model. Because of the substantial expense of the ensemble-based linearization, special focus is made on the assessment of the optimal frequency of such computations within the time intervals between data injections in typical operational systems.

KEYWORDS

ensemble methods, variational data assimilation

1 | INTRODUCTION

Over the last several decades, ensemble methods of data assimilation (DA) were among the major developing trends in computational geophysical fluid dynamics. Since the four-dimensional variational (4D-Var) technique is still one of the most skilful DA methods (Lorenc *et al.*, 2015; Bannister, 2016), a lot of efforts were made to improve 4D-Var performance by combining the advantages of 4D-Var with the wealth of statistical information carried by the ensembles (e.g., Zhang and Zhang, 2012; Kuhl *et al.*, 2013; Fairbairn *et al.*, 2014; Buehner *et al.*, 2015; Bonavita *et al.*, 2017). These efforts may mitigate certain disadvantages of the 4D-Var, which include intrinsic instability of the tangent linear models (TLMs) and their adjoints (AMs) in strongly nonlinear regimes, and poor differentiability

of certain sub-grid parametrizations. While the hybrid methods inherit benefits of both ensemble and 4D-Var approaches, they are also subject to the costs associated with their maintenance and development, including the issues of development and maintenance of the TLMs and adjoint models and their limited scalability compared to the ensemble techniques. Although the problem of 4D-Var development without AMs has been addressed in many studies (e.g., Liu *et al.*, 2008; Yaremchuk *et al.*, 2009; 2017), operational systems with AMs tend to provide better forecast skill than the methods based on approximations of the cost function gradients without using the AMs (Lorenc and Jardak, 2008).

In recent years, several attempts have been made to extract linearized representation of the model operator from the information contained in the ensemble of model

trajectories (Frolov and Bishop, 2016; Allen *et al.*, 2017; Bishop *et al.*, 2017; Frolov *et al.*, 2018). The approach is different from the ones mentioned above because it does not involve projection of the gradient on the range of the (localized) background-error covariance or on other subspaces predetermined by the model run, but attempts to directly employ the ensemble statistics for reconstructing (an approximation to) the TLM matrix. The underlying assumption of the approach is that the number of ensemble members is comparable to the grid-point stencil size of the matrix representation of the linearized model operator (Bishop *et al.*, 2017). Although numerical experiments with a shallow-water model performed by Allen *et al.* (2017) have shown promising results, extensions of the method to more general dynamical constraints are hindered by non-locality of the model numerics, often featuring non-local parametrizations and implicit schemes for filtering fast processes.

This study contributes to the methodology of the ensemble-based linearization of the model (ELM) operators. Specifically, we make an attempt to bypass non-locality by factorizing the structure of a one-step propagator into a sequence of sparse matrices utilized later for assembling the ELM approximation that is applicable for both explicit and implicit time stepping. The algorithm is tested using a shallow-water model in a periodic domain.

The paper is organized as follows. In the next section we describe the model, factorization of its one-step propagator, and its approximation over the finite time interval. Setting of the numerical experiments with an idealized model is described in Section 3. Section 4 contains the major results, including comparison of the CPU times with exact TLAM, skill of the respective 4D-Var experiments, and assessment of the optimal frequency of the ELM computation. Summary and discussion of further development conclude the paper.

2 | LINEAR APPROXIMATIONS OF A NUMERICAL MODEL

2.1 | Propagation of the ensemble average

Consider an ensemble of m trajectories of a numerical model propagating a state vector $\mathbf{x} \in \mathbb{R}^N$ from the k th time step to the next one

$$\mathbf{x}_{k+1}^i = \mathbf{m}(\mathbf{x}_k^i); \quad i = 1, \dots, m, \quad (1)$$

where $\mathbf{m}(\mathbf{x})$ is a sufficiently smooth vector-valued function of \mathbf{x} . In the following treatment, we also assume that

the numerical model conserves a scalar property $E = \langle \mathbf{x}, \mathbf{x} \rangle$ (e.g., energy) and the ensemble spread around the mean trajectory $\bar{\mathbf{x}}_k$ is small,

$$|\mathbf{x}_k^i - \bar{\mathbf{x}}_k| \equiv |\delta \mathbf{x}_k^i| \ll |\bar{\mathbf{x}}_k|, \quad (2)$$

with respect to the scalar product $\langle \cdot, \cdot \rangle$ introduced by the conservation law. Consider ensemble evolution on the time interval $[t_0, t_n]$ corresponding to a typical DA window in operational systems, and assume that this interval is short enough to have no sizable impact on the ensemble spread (so that validity of Equation (2) is not violated). Under these conditions, one-step evolution of an ensemble member can be represented by the linear transformation

$$\mathbf{x}_{k+1}^i = \mathbf{M}_k^i \mathbf{x}_k^i, \quad (3)$$

where $\mathbf{M}_k^i \equiv \mathbf{M}(\mathbf{x}_k^i)$ is the unitary matrix specifying rotation of the ensemble member \mathbf{x}_k^i on the surface of the energy sphere $|\mathbf{x}|^2 = E$. Hereinafter, we use the notation $\mathbf{M}(\mathbf{x})$ to specify, where necessary, the vector argument \mathbf{x} of the respective matrix-valued functions. As the matrix elements of \mathbf{M}_k^i depend on \mathbf{x}_k^i , they can be expanded in $\delta \mathbf{x}_k^i$ in the vicinity of $\mathbf{M}(\bar{\mathbf{x}}_k)$,

$$\mathbf{M}(\mathbf{x}_k^i) = \mathbf{M}(\bar{\mathbf{x}}_k) + \frac{\delta \mathbf{M}}{\delta \mathbf{x}}(\bar{\mathbf{x}}_k) \delta \mathbf{x}_k^i + \dots, \quad (4)$$

so that the evolution of the ensemble mean can be approximated by

$$\bar{\mathbf{x}}_{k+1} = \mathbf{M}(\bar{\mathbf{x}}_k) \bar{\mathbf{x}}_k. \quad (5)$$

Note that the ‘‘ensemble mean propagator’’ $\mathbf{M}(\bar{\mathbf{x}}_k)$ (denoted $\bar{\mathbf{M}}_k$ hereinafter) keeps the unitary property of \mathbf{M}_k^i intact, and, therefore, is less susceptible to numerical instabilities than the 4D-Var TL operator \mathbf{M}_k^* which governs evolution of small perturbations of Equation (3),

$$\delta \mathbf{x}_{k+1} = \mathbf{M}_k^* \delta \mathbf{x}_k = \left[\bar{\mathbf{M}}_k + \frac{\delta \mathbf{M}}{\delta \mathbf{x}}(\bar{\mathbf{x}}_k) \bar{\mathbf{x}}_k \right] \delta \mathbf{x}_k, \quad (6)$$

and contains an additional term which may destroy the conservative properties of \mathbf{M}_k , resulting in TLAM instabilities. These instabilities have to be suppressed by introduction of the additional diffusion (Hoteit *et al.*, 2005), which parameterizes the effect of the higher-order moments of the ensemble perturbations (Yaremchuk and Martin, 2014). In the above discussed approximation Equation (5) of the ensemble average evolution, the second term in the r.h.s. of Equation (6) is removed via averaging over the perturbations. In the hybrid 4D-Var methods, the matrix $\bar{\mathbf{M}}^T$ could be used to update the ensemble mean and keep it

being constrained by the conservation laws of the numerical scheme. In that respect, it would be reasonable to refer $\overline{\mathbf{M}}$ and \mathbf{M}^* as the ensemble-based linear model (ELM) and the ensemble-based tangent linear model (ETLM). The latter notation is adopted to distinguish presented technique from the earlier local ETLM (LETLM) method of Frolov and Bishop (2016), which does not directly account for the non-local structure of the constituents of \mathbf{M} , and, therefore, provides a significantly less accurate approximation to the TLM.

Since modern DA systems often treat evolution of $\bar{\mathbf{x}}$ as the best approximation to the truth, it is useful to develop approximations to $\overline{\mathbf{M}}_k$, \mathbf{M}_k^* and their adjoints for assessment of the ensemble mean sensitivities with respect to control variables. A formal way to do this for a large ($m > N$) ensemble is to retrieve the unknown elements of the respective matrices by solving the matrix equations (cf. Equations (5, 6))

$$\overline{\mathbf{M}}_k \mathbf{X}_k = \mathbf{X}_{k+1}, \quad \mathbf{M}_k^* \delta \mathbf{X}_k = \delta \mathbf{X}_{k+1}, \quad (7)$$

where \mathbf{X} stands for the $N \times m$ matrices listing the ensemble members columnwise. This approach is surely prohibitive, since ensembles rarely exceed 100 members in size, while the typical values of N range within 10^7 – 10^9 for operational models. However, the curse of dimensionality could be circumvented if $\overline{\mathbf{M}}_k$ and \mathbf{M}_k^* are represented by sparse matrices, whose stencil sizes (the maximum number of non-zero elements in a row) n^s do not exceed the ensemble size m . The latter constraint appears to be valid for the numerical models of fluid dynamics, because these models are formulated in terms of discretized differential operators that are local in nature and, therefore, represented by sparse matrices. For explicit numerical schemes the constraint $n^s < m$ could be satisfied because most of the models employ small stencil discretizations of the differential and locally averaging operators to approximate evolution of the state vector on the model grid. Non-zero elements of the corresponding matrices occupy relatively compact areas in physical space, so that these elements can, in principle, be retrieved from Equation (7) by solving N systems of $m \times m$ linear equations if the centres and sizes of the stencils are known, and n^s does not exceed the ensemble size (Appendix A). This assumption was central to the development of the original LETLM method by Frolov and Bishop (2016). However, the numerical schemes of more realistic models often feature semi-implicit methods and integral transforms that are non-local in nature and lead to stencil sizes $n^s \gg m$, resulting in violation of the resolvability condition. Several attempts were made to regularize the respective underdetermined problems (e.g., Frolov *et al.*, 2018), with limited success.

2.2 | Factorization of the one-step propagator

In this study we explore the possibility to approximate E(T)LM operators for numerical models employing semi-implicit schemes. Assume for simplicity that the one-step model propagator can be factorized in the form

$$\mathbf{M} = \mathbf{L}^{-1} \mathbf{L}_0, \quad (8)$$

where the (unknown) linear operators \mathbf{L} , \mathbf{L}_0 are represented by sparse matrices which satisfy the conditions $n^s, n_0^s \leq m$, and the locations/sizes of the respective stencils are known. The matrix \mathbf{L} accounts for the implicit part of the numerical scheme and acts on the components of \mathbf{x}_{k+1} . This action is usually available as a code of matrix multiplication in the respective implicit solver of the linear system $\mathbf{L} \mathbf{x}_{k+1} = \mathbf{L}_0 \mathbf{x}_k$.

The structure of \mathbf{M} can then be implicitly obtained through the two-step procedure involving $2N$ solutions of the $m \times m$ systems of linear equations derived from the rows of the auxiliary ensembles $\mathbf{X}_0^* = \mathbf{L}_0 \mathbf{X}_0$ and $\mathbf{X}_1^* = \mathbf{L} \mathbf{X}_1$. The input and output ensembles \mathbf{X}_0 , \mathbf{X}_0^* , \mathbf{X}_1 , \mathbf{X}_1^* do not require additional computations because they can be extracted during the ensemble run as intermediate outputs in the appropriate places of the code. The retrieved matrices \mathbf{L}_0 , \mathbf{L} can be viewed as a compressed storage form of \mathbf{M} , whose multiplication by an arbitrary vector \mathbf{x} can be executed in two steps:

- compute $\mathbf{x}' = \mathbf{L}_0 \mathbf{x}$;
- apply the model's implicit solver to the system matrix \mathbf{L} with the right-hand side \mathbf{x}' to obtain $\mathbf{M} \mathbf{x}$.

The above algorithm can be easily generalized for a sequence (Equation 8)) of arbitrary length $n_\ell > 2$

$$\mathbf{M} = \prod_{j=0}^{n_\ell} \mathbf{L}_j, \quad (9)$$

where \mathbf{L}_j represent either sparse matrices or their inverses. To clarify presentation, we further limit ourselves to the case $n_\ell = 2$ described by Equation (8).

The overhead expense of computing the compressed representation (Equation 9)) consists of i/o operations for storing n_ℓ auxiliary ensembles and the expense of solving $n_\ell N$ systems of $m \times m$ linear equations, which could be costly compared to the ensemble run if \mathbf{M} is computed on every time step. To mitigate extra cost, the elements of \mathbf{M}_k can be computed on a sparser grid in both space and time and then interpolated to the original grid to obtain an approximation to $\mathbf{M} \mathbf{x}$.

Similarly, the action \mathbf{M}^T on a state vector can be accomplished using the reverse sequence (cf. Equation (8))

$$\mathbf{M}^T \mathbf{x} = \mathbf{L}_0^T \mathbf{L}^{-T} \mathbf{x}. \quad (10)$$

Note that transposition of the factors \mathbf{L}_0 , \mathbf{L} is not computationally expensive, since both of them are directly stored in the sparse format after their row-by-row retrieval from the auxiliary ensembles.

2.3 | Approximation of the time evolution

In the practical 4D-Var applications, observations arrive into the assimilation system at fixed time intervals $\tau^j = t_{k_j} - t_{k_{j-1}}$ within the DA window, so that the key component of a 4D-Var system is propagation of the model–data misfits over the data accumulation intervals τ^j by the adjoint code. Since the number of time steps n between the data injections could be quite large, the above-mentioned sparsification of \mathbf{M} may not be enough to ease the burden of computing $\mathbf{M}_1, \dots, \mathbf{M}_n$ at every time step. In that respect it might be reasonable to evolve model–data misfits over τ^j using the adjoint operators interpolated in time. In many practical cases, the model state does not change dramatically over τ^j and the evolution of \mathbf{M}_k over the data accumulation intervals between injections can be well approximated by a linear function

$$\mathbf{M}_k = \left(1 - \frac{k}{n}\right) \mathbf{M}_{k_{j-1}} + \frac{k}{n} \mathbf{M}_{k_j}, \quad k = 1, \dots, n, \quad (11)$$

which requires, in the long-term average, just one retrieval of \mathbf{M} from the ensemble per data accumulation interval.

Furthermore, if \mathbf{M}_k could be represented as perturbations $\mathbf{I} + \varepsilon \mathbf{M}'_k$ of the identity matrix \mathbf{I} , with ε small enough to assume that expansion of the finite-time propagator

$$\mathbf{P} = \mathbf{M}_{n-1} \dots \mathbf{M}_1 \mathbf{M}_0 = \mathbf{I} + \varepsilon \sum_{k=0}^{n-1} \mathbf{M}'_k + \varepsilon^2 \sum_{k<l} \mathbf{M}'_k \mathbf{M}'_l + \dots \quad (12)$$

converges at a reasonable rate, the lengthy time integration over τ^j could be approximated by a nonlinear function of \mathbf{M}'_0 and \mathbf{M}'_n through recursive applications of the operator $\hat{\mathbf{M}} = (\mathbf{M}'_0 + \mathbf{M}'_n)/2$:

$$\mathbf{P} = \mathbf{I} + n\varepsilon \hat{\mathbf{M}} \left[\mathbf{I} + \frac{n\varepsilon}{2} \hat{\mathbf{M}} + \dots \right]. \quad (13)$$

That is, instead of computing a series of intermediate ETLMs, evolution of \mathbf{M} over a multi-step time interval can be approximated by the nonlinear function (Equation (13)) of the ETLMs pre-computed at the beginning and the end

of the interval. If the expansion (Equation (12)) does not converge fast enough, time interpolation could be used on smaller intervals within τ^j to optimize the trade-in between the computational cost and accuracy of the approximation. In the present study, we test the efficiency of the more simple linear approximation (Equation (11)).

In the next section we describe setting of the numerical experiments used for testing the linear approximation (Equation (11)) with a nonlinear numerical model described below.

3 | EXPERIMENTAL SETTING

3.1 | Numerical model

The shallow-water equations were discretized on a C-grid to simulate dynamics of a barotropic flow on the f -plane:

$$\eta_t = -\nabla \cdot (h + \eta) \mathbf{u}, \quad (14)$$

$$\mathbf{u}_t = -g\nabla\eta - f\mathbf{k} \times \mathbf{u} + (\nu\nabla^2 - \mu - \mathbf{u} \cdot \nabla) \mathbf{u}. \quad (15)$$

Here η is the sea surface height, \mathbf{u} is the horizontal velocity vector, ∇ is the gradient operator, \mathbf{k} is the vertical unit vector, $h(x)$ is the ocean depth ($x \in \mathbb{R}^2$), $f = 10^{-4} \text{s}^{-1}$ is the Coriolis parameter, $g = 9.8 \text{m}\cdot\text{s}^{-2}$ is the gravity acceleration and μ, ν are the Newtonian friction and horizontal viscosity coefficients. In the horizontal, a grid step δx was 10 km in both directions, and the periodic boundary conditions were used.

To keep consistency with Equation (1) and layout (Equation (8)), the time integration was performed by the implicit Crank–Nicolson type scheme with a time step $\delta t = 1$ hr, so that the model operator \mathbf{M} was given by

$$\mathbf{M} = \mathbf{L}^{-1} \mathbf{L}_0 \equiv \left\{ \mathbf{I} + \frac{1}{2} [\mathbf{N}(\mathbf{x}) - \mathbf{Q}] \right\}^{-1} \left[\mathbf{I} + \frac{1}{2} \mathbf{Q} \right], \quad (16)$$

where \mathbf{N} is a sparse matrix whose elements depend on \mathbf{x} , and \mathbf{Q} is a sparse matrix with constant (\mathbf{x} -independent) elements. Details of the numerics can be found in Appendix B.

Using the above notation, the expressions for the LM and TLM model operators are given by (Appendix C)

$$\bar{\mathbf{M}} = \left\{ \mathbf{I} + \frac{1}{2} [\mathbf{N}(\bar{\mathbf{x}}) - \mathbf{Q}] \right\}^{-1} \left[\mathbf{I} + \frac{1}{2} \mathbf{Q} \right] \equiv \bar{\mathbf{L}}^{-1} \mathbf{L}_0, \quad (17)$$

$$\mathbf{M}^* = \bar{\mathbf{M}} - \frac{1}{2} \bar{\mathbf{L}}^{-1} \frac{\delta \mathbf{N}}{\delta \mathbf{x}}(\bar{\mathbf{x}}) \bar{\mathbf{L}}^{-1} \mathbf{L}_0 \bar{\mathbf{x}}. \quad (18)$$

Equations (15) and (15) were discretized on a homogeneous 59×59 grid with the basic parameters set as follows: $\mu = 10^{-6} \text{day}^{-1}$, $\nu = 10^{-6} \delta x^2 / \delta t \sim 1.4 \times 10^4 \text{m}^2 \cdot \text{s}^{-1}$, and $h = 20\{5 + \xi(x)\} \text{m}$, where $\xi(x)$ is a realization of the

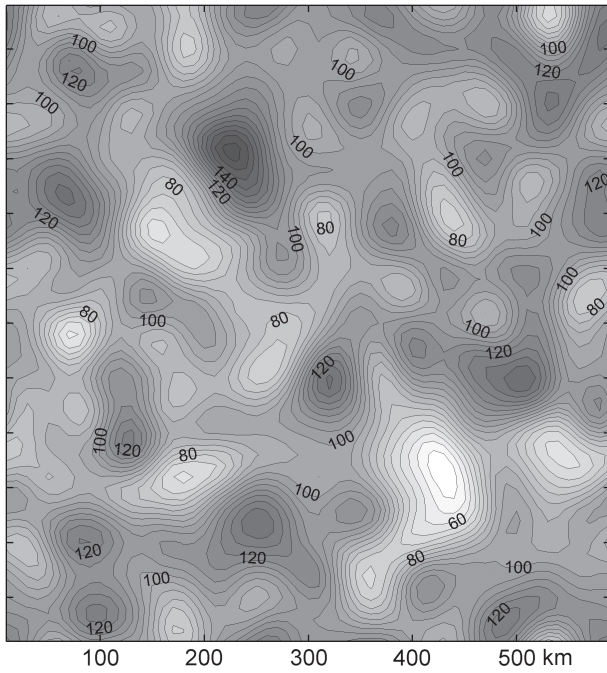


FIGURE 1 Bottom topography (m)

random field with zero mean, unit variance and decorrelation scale $\rho = 30$ km (Figure 1). Different realizations of the same random field were also used to initialize the ensemble of model states.

The length of the model integration (assimilation window) and the data accumulation interval τ for the experiments were set to 4 days and 24 hr respectively.

The basic (ensemble mean) state $\mathbf{x}(0)$ at the beginning of integration was specified as follows. The geostrophic component $\mathbf{x}_g(0)$ was defined by setting $\eta_g(x, 0) = \eta_0 \xi(x)$ and computing the geostrophic velocities (Figure 2). With the value of $\eta_0 = 0.2$ m, the respective geostrophic currents had typical velocities of $\mathbf{u}_0 \sim 0.9 \text{ m} \cdot \text{s}^{-1}$. After that, an ageostrophic component was added, by generating another realization of $\xi(x)$ and setting $\eta_a(x, 0) = 0.1 \eta_0 \xi(x)$, $\mathbf{u}_a = 0.1 \mathbf{u}_0 \xi(x)$. With this formulation, the model trajectory (Figure 2) had a substantial degree of nonlinearity: the respective Rossby number

$$Ro = \frac{|\mathbf{u} \cdot \nabla \mathbf{u}|}{f|\mathbf{u}|}$$

varied between 0 and 1.35 within the domain with the average value of 0.32.

The ensemble was generated in exactly the same manner, using different realizations of the random field ξ . After specification of the ensemble members, the ensemble-mean value was subtracted from the members, then each member was multiplied by the spread factor γ and then the basic state was added. The parameter

γ was varied between 10^{-4} and 3 in the course of the experiments.

Since the maximum stencil size in the discretizations of the differential and averaging operators was 3×3 (Appendix B), the ensemble size was set to $m = 28$ members to guarantee capturing all non-zero elements in the rows of the matrices \mathbf{L}_0 and \mathbf{L} factorizing $\overline{\mathbf{M}}$ and \mathbf{M}^* .

3.2 | 4D-Var setting

Performance of the TLAM approximations derived from the ensembles was assessed in the 4D-Var environment using a simple set of twin data assimilation experiments. For a given assimilation window, the data (the values of \mathbf{u} and η at randomly selected observation points, e.g., Figure 2b) were picked from the ensemble mean trajectory and contaminated by the white noise with specified variances σ_η , σ_u . These variances did not vary in space and defined the diagonal of the observation-error covariance matrix \mathbf{R} . The inverse square root of the background-error covariance matrix was specified (Xu, 2005; Yaremchuk *et al.*, 2013) as the block-diagonal matrix with the three cells for each state vector component given by

$$\mathbf{B}^{-1/2} = \frac{\rho}{\delta x} \mathbf{D} \left[\mathbf{I} - \frac{\rho^2}{2} \nabla^2 \right], \quad (19)$$

where \mathbf{D} is the inverse square root of the respective part of the diagonal extracted from the ensemble covariance matrix.

The first-guess solution was obtained from a randomly picked ensemble member and then optimized through minimization of the cost function

$$J = \frac{1}{2} \left[\mathbf{x}_0^T \mathbf{B}^{-1} \mathbf{x}_0 + \sum_{k=1}^4 (\mathbf{H} \mathbf{P}_k \mathbf{x}_0 - \mathbf{d}_k)^T \mathbf{R}^{-1} (\mathbf{H} \mathbf{P}_k \mathbf{x}_0 - \mathbf{d}_k) \right] \quad (20)$$

with respect to the initial state \mathbf{x}_0 . Here \mathbf{H} is the identity matrix with diagonal elements replaced by zeros at locations without data, \mathbf{d}_k are the observation vectors, and \mathbf{P}_k is the nonlinear model propagator from $t = 0$ to the k th data injection time, separated from the initial condition by k days. The spatial and temporal density of observations varied between the different sets of experiments.

The minimization was performed by a quasi-Newtonian descent algorithm featuring limited-memory BFGS updating (Schmidt, 2005) with the gradient supplied by

$$\frac{\delta J}{\delta \mathbf{x}_0} = \mathbf{B}^{-1} \mathbf{x}_0 + \sum_{k=1}^4 \mathbf{M}_k^T \mathbf{H} \mathbf{R}^{-1} (\mathbf{H} \mathbf{P}_k \mathbf{x}_0 - \mathbf{d}_k), \quad (21)$$

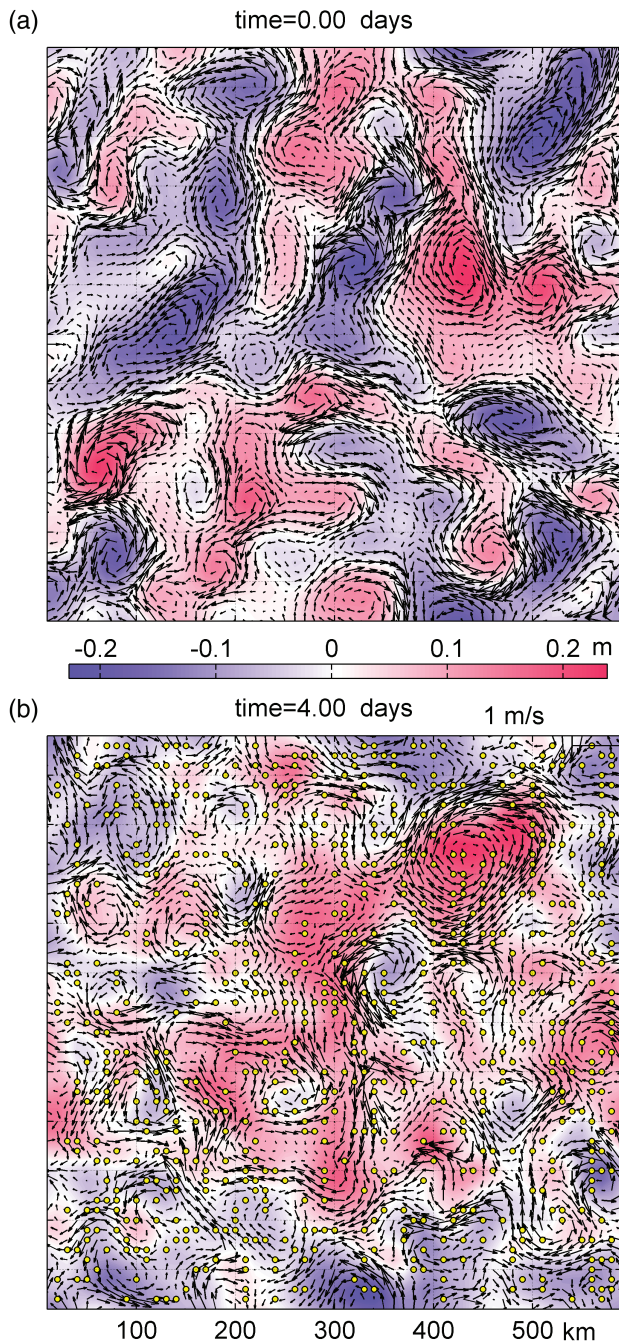


FIGURE 2 Evolution of the ensemble mean total currents (arrows) and surface elevations (colour) for the case $Ro = 0.3$ at (a) time=0, days and (b) 4 days. In (b), circles denote locations of the observation points at the end of the integration ($t = 4\tau$) [Colour figure can be viewed at wileyonlinelibrary.com].

where \mathbf{M}_k^T is either the adjoint of the exact TL propagator in Equation (6), or its ELM/ETLM approximations obtained using Equations (8), (10), and (11).

4 | RESULTS

In what follows we analyze the inaccuracy A of E(T)LM retrievals and their computational cost c . The former is

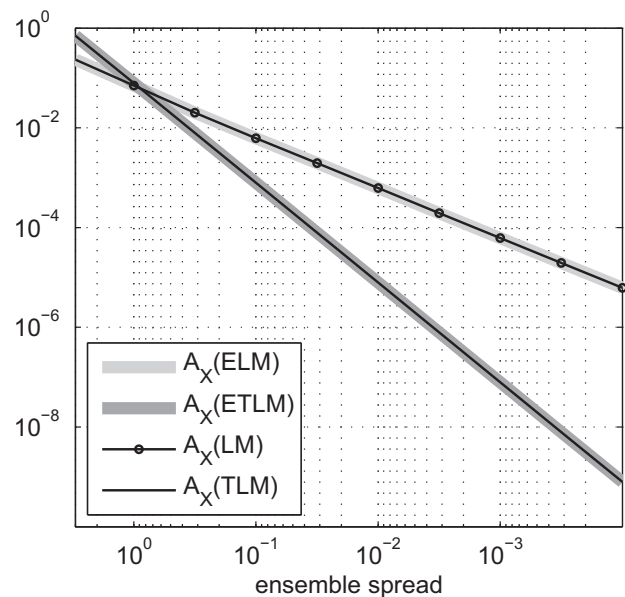


FIGURE 3 Inaccuracy A_X in approximating the model evolution by the E(T)LM operators as a function of the ensemble spread γ for the case $Ro = 0.3$

characterized by the parameters

$$A_o = \frac{\|\mathbf{O}_e - \mathbf{O}\|}{\|\mathbf{O}\|}, \quad A_X = \frac{\|\mathbf{O}_e \mathbf{X} - \mathbf{O} \mathbf{X}\|}{\|\mathbf{O} \mathbf{X}\|}, \quad (22)$$

where $\|\cdot\|$ denotes the Frobenius norm and \mathbf{O} , \mathbf{O}_e respectively stand for the exact TLM operator and the operator retrieved from the ensemble. The computational cost c_τ of the operator retrievals is measured in per cent and defined by $c_\tau = 100\tau_r/\tau_X$, where τ_r , τ_X are the CPU times required by the operator retrieval process and by the non-linear propagation of the ensemble over the time interval between the retrievals, respectively.

4.1 | E(T)LM approximation errors

Computation of the sparse constituents (\mathbf{L}_0 and \mathbf{L}) of the LM matrix $\overline{\mathbf{M}}$ (Equation (18)) requires two solutions of the matrix equations involving intermediate ensembles dumped in the process of the ensemble integration (Section 2.2). Since stencil sizes and locations of \mathbf{L}_0 and \mathbf{L} are known, and the ensemble size m is larger than n^s , respective matrix elements are retrieved exactly, so that approximations of $\overline{\mathbf{M}}$ and \mathbf{M}^* are always obtained within the machine accuracy ($A_o = 10^{-14}$). Figure 3 shows the errors in approximating ensemble evolution by the E(T)LM approximations of $\overline{\mathbf{M}}$ and \mathbf{M}^* . The curves are visually indistinguishable from the ones obtained by propagating the ensemble by the exact (T)LM operators coded using analytical derivations (thin solid lines in Figure 3).

Equations (16) and (18) show that practical factorization of $\bar{\mathbf{M}}$ requires ensemble retrievals of \mathbf{Q} and $\bar{\mathbf{N}}$. For the considered dynamical system, the computational cost of these two retrievals was close to 5.2% if they were performed on every time step, with variations of 0.3% depending on the point along the model trajectory within the assimilation window.

To assess the computational efficiency of retrieving \mathbf{M}^* , we first note that since the matrix elements of \mathbf{N} are homogeneous linear functions of \mathbf{x} , Equation (18) can be rewritten in the form

$$\mathbf{M}^* = \bar{\mathbf{M}} - \frac{1}{2} \bar{\mathbf{L}}^{-1} \frac{\delta \mathbf{N}}{\delta \mathbf{x}} \bar{\mathbf{M}} \bar{\mathbf{x}} = \bar{\mathbf{L}}^{-1} \left[\mathbf{L}_0 - \frac{1}{2} \mathbf{N}(\bar{\mathbf{M}} \bar{\mathbf{x}}) \right] \quad (23)$$

and thus requires an additional retrieval of $\mathbf{N}(\bar{\mathbf{M}} \bar{\mathbf{x}})$ from the ensemble. The respective computational expense is mostly defined by the necessity to compute the product of (already available) $\bar{\mathbf{M}}$ by $\bar{\mathbf{x}}$ followed by the ensemble retrieval of the elements of $\mathbf{N}(\bar{\mathbf{M}} \bar{\mathbf{x}})$. This procedure adds $7.1 \pm 0.6\%$ to $c_\tau(\bar{\mathbf{M}})$, resulting in the overall cost $c_\tau(\mathbf{M}^*) = 12.3 \pm 0.8\%$ for the e ETLM retrieval at every time step.

Although the values of $c_\tau(\bar{\mathbf{M}})$ and $c_\tau(\mathbf{M}^*)$ appear to be a relatively moderate price for computing the exact LM and TLM operators, the respective cost could be much higher in more realistic applications. Therefore, it is worth considering their approximations via time interpolation (Equation (11)), an inexpensive ($c_\tau(i) = 0.01\%$) procedure for sparse matrices. If the retrievals are performed every m th time step, the overall cost of computing the ETLM operator will reduce to $mc_\tau(i) + c_\tau(\mathbf{M}^*)/m$. In that respect, it is instructive to assess the accuracy if such interpolation within the data acquisition interval as a function of the number of time steps n_i between the ETLM retrievals.

Figures 4 and 5 indicate that, in terms of accuracy, time interpolation could be feasible if E(T)LM retrievals are performed not less than two times ($dt = 12\delta t = 12$ hr, dark grey lines) per data acquisition interval. In this case ELM and ETLM approximations are accurate within 5–7% and 0.5–1% respectively. Performing the retrieval only once ($dt = 24\delta t$) inflates the error to tens of percent in the case of ELM and to several percent for TLM, which could have a strong impact on the 4D-Var descent process. In the following section we assess this impact in a series of numerical experiments.

4.2 | 4D-Var experiments

The numerical experiments were performed using two basic ensembles. The first one (described in Section 3.1) was characterized by the maximum and time-averaged Rossby numbers of 1.35 and 0.32, characteristic for

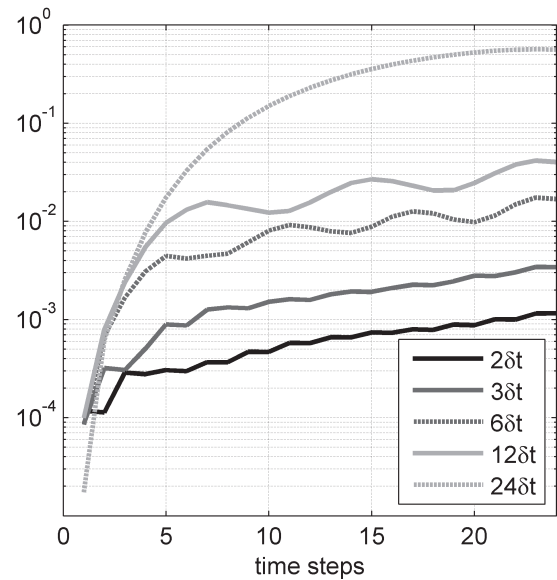


FIGURE 4 Inaccuracy A_X in approximating the ensemble mean evolution by the ELM operator interpolated in time within the data acquisition interval ($\tau = 24$ hr). The values of the interpolation time intervals n_i are shown at the lower right

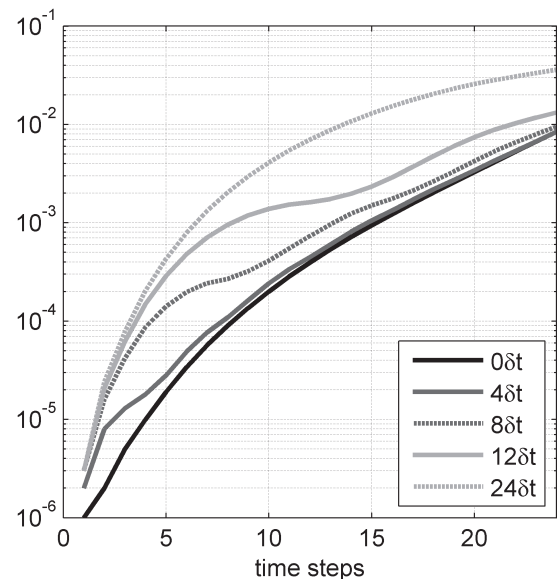


FIGURE 5 As Figure 4, but for the ETLM operator

sub-mesoscale turbulence. The second ensemble was generated in a similar manner, but using a three times smaller value of η_0 , so that the resulting maximum and mean Rossby numbers (0.4 and 0.1) were more typical for the oceanic conditions observed in the regions of the western boundary currents.

4.2.1 | Benchmark runs

For each basic ensemble, we performed two benchmark 4D-Var assimilation experiments. The first one was

targeted at the assessment of the optimal frequency of the TLAM extractions from the ensembles within a single data acquisition window. In this experiment, the data were specified in every grid point ($\mathbf{H} = \mathbf{I}$ in Equation (20)) without noise contamination ($\sigma_\eta = \sigma_u = 0$) at $t = \tau$, and no regularization was used ($\mathbf{B}^{-1} = 0$).

The second series of 4D-Var experiments were performed to assess the impact of time interpolation on the descent process in a more realistic environment: the model solution was optimized within the 4-day assimilation window with four data acquisitions performed at the end of each day in four sets of random points (an example is shown in Figure 2). The simulated observations were taken from the “true” solution and contaminated with 10% noise with observation and background (Equation (19)) error covariances specified accordingly.

In both series, the optimization process was interrupted at 50 iterations. The “quality” of optimization was estimated by computing the ratio

$$e = \frac{|\mathbf{x}_0^{\text{opt}} - \mathbf{x}_0^{\text{tr}}|}{|\mathbf{x}_0^{\text{fg}} - \mathbf{x}_0^{\text{tr}}|},$$

where \mathbf{x}_0^{fg} and \mathbf{x}_0^{tr} are respectively the first guess and true states at $t = 0$.

Figure 6 illustrates the 4D-Var performance for the four benchmark solutions characterized by two Rossby numbers ($Ro = 0.1, 0.3$) and two assimilation windows (Figure 2). For perfect observations at the end of the data acquisition interval ($\tau = 1$ day), the true solution is reconstructed within the accuracy of $e = 2.1 \times 10^{-4}$ in 50 iterations when $Ro = 0.1$. At higher nonlinearity ($Ro = 0.3$), the convergence rate slows down (grey lines in Figure 6) and the reconstruction error after 50 iterations increases to $e = 0.24$, although after 200 iterations it becomes less than 0.1 (not shown).

A similar value of the reconstruction error $e = 0.28$ is obtained with noisy data and 4-day assimilation window at $Ro = 0.1$ (solid black line in Figure 6), while with the higher level of nonlinearity ($Ro = 0.3$), only a slight improvement of the first-guess solution is achieved in 50 iterations ($e = 0.87$, solid grey line).

Since the ETLM operators are reconstructed with machine precision, the above numbers and plots remain virtually unchanged, when the exact (analytically derived and coded) adjoints were used in the 4D-Var computations.

4.2.2 | Impact of time interpolation

The situation changes when the adjoint operators \mathbf{M}_k^T are replaced by their approximations (Equation (11)).

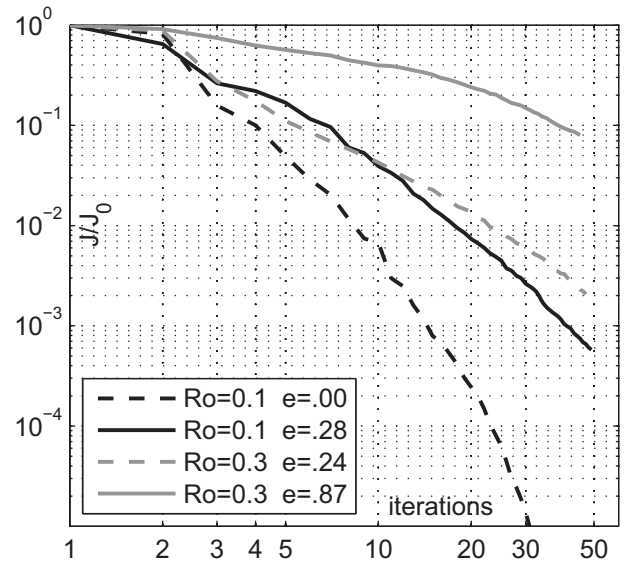


FIGURE 6 Impact of observation density and model nonlinearity on the reconstruction of the true state, showing the reduction of the cost function J with iterations for the benchmark 4D-Var runs with various values of Ro (listed in the lower left corner) and assimilation periods. The cost function is normalized by its initial value J_0 . Errors e of approximating the true solution are shown in the key. Solid lines correspond to a 4-day assimilation window with noisy randomly distributed data. Dashed lines show the reduction of J/J_0 for the 1-day run with complete observation of the state vector at the end of the assimilation interval

Although the ETLM errors exposed in Figure 5 are generally below 1% over the data acquisition interval, they can grow substantially at longer integration times, and, more importantly, have a tendency to accumulate with 4D-Var iterations.

Figures 7 and 8 demonstrate dependence of the 4D-Var convergence rate on the time interpolation interval $n_i \delta t$ for the “realistic” case (4-day assimilation window with noisy data). The overall impact of time interpolation becomes noticeable at $n_i = 8\delta t$, when the improvement e of the optimal solution drops from 0.87 to 0.91 for $Ro = 0.3$ (Figure 7), and from 0.29 to 0.34 for $Ro = 0.1$ (Figure 8). For the larger intervals between operator retrievals ($n_i = 12\delta t$), the descent process quickly loses efficiency after approximately ten iterations, is clearly visible in the behaviour of both the cost function and the gradient (thick black lines in Figure 7), and yields virtually no gain ($e = 0.99$) in approximating the true solution compared to the first guess for $Ro = 0.3$. For the lower nonlinearity level $Ro = 0.1$ (Figure 8), the loss of efficiency also becomes noticeable at $n_i = 12\delta t$ (thick grey lines in Figure 8), but demonstrates a behaviour similar to Figure 7 only when the ETLM retrievals occur at the data acquisition intervals $n_i = 24\delta t$, (thick black lines in Figure 8).

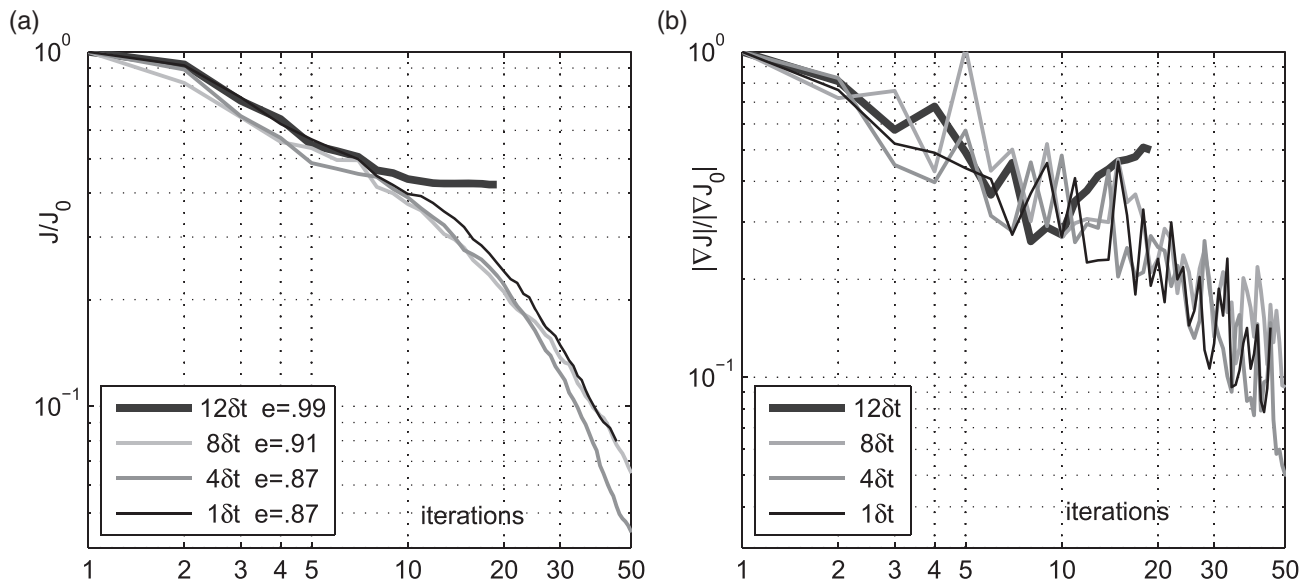


FIGURE 7 Reduction of the (a) cost function and (b) gradient with iterations for 4D-Var runs with $Ro = 0.3$, $\tau = 4$ days, and various time interpolation intervals of the adjoint ETLM operator (listed in the lower left corners). Errors e of approximating the true solution are shown in (a). The case labelled $1\delta t$ carries no approximation caused by time interpolation of the TL operators

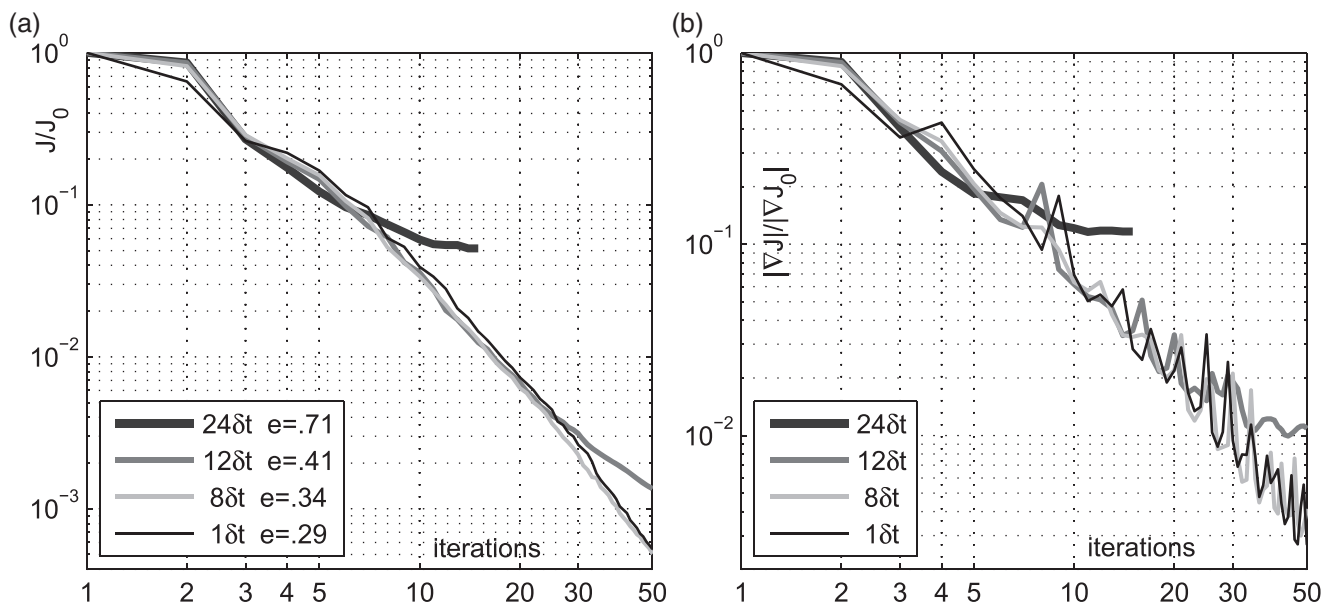


FIGURE 8 As Figure 7, but for $Ro = 0.1$

We should also note that smaller values of $n_i < 8\delta t$ allow us to attain a considerably better ($e \sim 0.65$) approximation to the truth in the case of strong nonlinearity $Ro = 0.3$ after a considerably larger (500) number of iterations, which is impractical, because in realistic 4D-Var applications the descent process is terminated much earlier.

A qualitatively similar behaviour of the 4D-Var descent process has been observed in assimilation experiments within the data acquisition window $\tau = 1$ day and

perfectly observed state at $t = \tau$ (not shown). Quantitatively, the true solution in this case was reconstructed with a much higher accuracy ($e < 0.001$ for $Ro = 0.1$ and $e = 0.24$ for $Ro = 0.3$) up to the value of $n_i = 8\delta t$, evidently due to the absence of noise in the observations and their specification in every grid point at $t = \tau$. At $n_i = 12\delta t$, the descent process stagnated after approximately 25 iterations, but these were enough to reach the accuracy of $e = 0.03$ for $Ro = 0.1$. At $n_i = 24\delta t$, stagnation occurred after seven iterations for both levels of nonlinearity,

producing the final inaccuracies of $e = 0.33$ ($Ro = 0.1$) and $e = 0.90$ ($Ro = 0.3$). It is noteworthy that stagnation of the descent process at $n_i = 24\delta t$ at $Ro = 0.1$ resulted in approximately the same reconstruction accuracy $e = 0.33$ as in the case when the gradient was computed using the adjoint of the ELM model ($e = 0.39$).

As mentioned earlier, the case $Ro = 0.1$ (with the maximum values of $Ro = 0.4$) is more typical for the regions of western boundary currents, which occupy a minor part of the world ocean. To assess the value of ELM operator and its adjoint in optimizing the model state in the majority of the open ocean, a separate series of the 4D-Var experiments were conducted with the nonlinearity level $Ro = 0.03$. As expected, the results demonstrated less efficiency of the descent process than ETLM (thick lines in Figure 9) and much weaker dependence on the frequency of ELM retrievals. The latter can be explained by the fact that, at these levels of nonlinearity, TLMs tend to stabilize and stagnation of the descent process largely depends on the accuracy of approximating the nonlinear state evolution by a linear model. In that sense the ELM approximation becomes slightly disadvantageous because it is less accurate than ETLM (Figure 3).

One may expect that, at lower levels of nonlinearity, ELM-based 4D-Var becomes more competitive to ETLM 4D-Var due to the diminishing role of time variability of the E(T)LTM operators. The result of the experiment exposed in Figure 9 indicates that ELM-based assimilation could be a reasonable alternative to the ETLM-based 4D-Var, especially if we take into the account the conservation properties of the ELM operator retrieved from ensembles with weak spread.

5 | SUMMARY AND DISCUSSION

An Ensemble TLM (ETLM) generalization of the Local Ensemble TLM (LETLM) technique of Frolov and Bishop (2016) has been proposed. The method relaxes LETLM constraint on the locality of the one-step propagator of a numerical model by assuming that the model can be factorized into a sequence of local and non-local operators, constrained by the condition that inverses of the non-local operators are represented by sparse matrices (Section 2.2). The performance of the method has been tested in a series of twin-data experiments with a 28-member ensemble integration of a nonlinear semi-implicit numerical model with 10^4 degrees of freedom described in Section 3. The results demonstrate that exact TLM and its adjoint could be retrieved from the ensemble at the extra computational cost of 12–15% compared to the cost of the one-step ensemble integration (Section 4.1). It is noteworthy that our attempts to apply Frolov and Bishop's (2016)

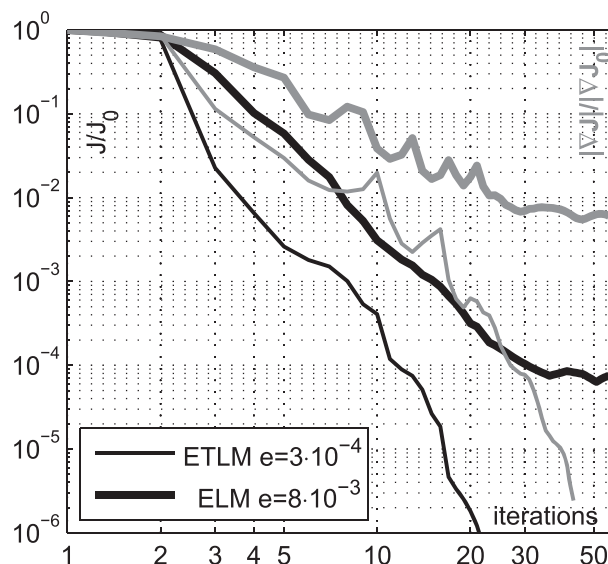


FIGURE 9 Reduction of the cost function (black lines) and gradient (grey lines) for ELM-based and ETLM-based 4D-Var experiments with $Ro = 0.03$ and $\tau = 1$ day

LETLM technique to the semi-implicit numerics of the model described in the paper, failed to recover a useful approximation to \mathbf{M}^* , even at a very small level of nonlinearity. This could be explained by the fact that the number of non-zero elements n^s in the rows of $\bar{\mathbf{M}}$ and \mathbf{M}^* was an order of magnitude larger than the ensemble size m due to the presence of the inverse matrices in Equations (18) and (18). The alternative approach was to inflate the ensemble size which dramatically reduced the computational efficiency of the LETLM retrievals.

In addition, we investigated the possibility of reducing the frequency of ETLM retrievals during the ensemble integration by linearly interpolating the extracted operators in time between the retrievals. Results of the experiments in Section 4.2 indicate that the 4D-Var optimization process is weakly affected if ETLM retrievals are performed less than every 12 time steps in the case of moderate nonlinearity ($Ro = 0.1$) of the background solution, and less than every 6 time steps for stronger nonlinearity ($Ro = 0.3$). This result indicates the possibility to considerably reduce the cost of ETLM extraction in more realistic settings, where the stencil sizes of the sparse matrices in the factorization can be significantly larger than 28.

An important advantage of the ETLM technique over LETLM is that it is capable of producing the exact TLM matrices (and their adjoints) in the case of quadratic nonlinearity of the parent model. For higher-order nonlinearities, the method could be even more advantageous, as it may implicitly mitigate TLM instabilities by processing the fully nonlinear ensemble perturbations, thus keeping

intact the conservative properties of the one-step linear approximation to the parent model.

Apart from the capability to semi-automatically extract ETLMs from the ensembles, a significant advantage of the technique is that it produces ETLM approximations in the form of sparse matrices, which can be efficiently multiplied by the model state using standard math libraries on high-performance massively parallel computers. Furthermore, our results indicate that the input/output cost of reading the ensembles and ETLM operators could be further reduced by employing time interpolation (Equations (11), (13)).

Despite promising results, the presented study was done with a simplified model, and many aspects of the ETLM approach should be investigated before the method could be applied to realistic models. Among those are the proper treatment of higher-order nonlinearities and accommodation of different types of non-local operators, such as convolutions, projections, and other types of integrals applied to the model state (and routinely used, for example, in the mixing/radiation schemes or pressure computations). In that respect, one may think of further generalization of the approach by representing the integrals via (pseudo)inverses of the appropriate differential operators which can be represented by sparse matrices.

The availability of the exact ELMs and ETLMs in the sparse format together with ensemble statistics provides an opportunity to impose conservation laws of the parent model by constraining ETLMs to the respective manifold. In applications, this has been done by adding a heuristic diffusion term to the adjoint models (e.g., Hoteit *et al.*, 2005; Yaremchuk and Martin, 2014). We speculate that this kind of regularization procedure could be done in a more succinct way by combining ensemble statistics with an opportunity to quickly estimate the leading spectral components of the respective sparse matrices.

An equally important issue for ETLM methodology is the rapid growth of the retrieval cost with the stencil size. As an example, an ocean model with the number $n_f = 4$ of three-dimensional prognostic fields (temperature, salinity and two horizontal velocity components), and the stencil half-width $n_s = 2$ will require an ensemble of around $n_s = n_f(2n_s + 1)^3 = 4 \times 125 = 500$ members for the exact retrieval of the ETLM operator. This size appears to be close to the upper limit of the current computer capabilities. Although the problem of n_s reduction can be mitigated to some extent by employing more information on the stencil structure, the cost of retrieving non-zero elements in a TLM row will be at least an order in magnitude larger than in the present study, and will become comparable to the cost of the ensemble integration. However, this

expense could be reduced by the ETLM interpolation in time and/or spatial sparsification of the retrievals.

Another interesting aspect of the ETLM methodology is its similarity to the machine learning (ML) technique, which extracts an unknown mapping from the training ensemble. The difference is that ML methods are well developed for largely unknown maps and huge ensembles of training samples, while ETLMs operate with relatively well-known maps and much smaller training sets. The presented study can be considered as an attempt to improve LETLM technique by employing additional information on the general structure of the retrieved map in the operator extraction algorithm. In that respect, the latest ML developments in sparse recovery (Wang *et al.*, 2018) could improve the computational efficiency of ETLM retrievals. One can anticipate that application of the appropriately modified versions of the ML algorithms are able to bring an extra benefit to ETLM development, especially in the treatment of poorly differentiable operators (such as vertical mixing schemes and cloud/convective physics) which parametrize fast sub-grid processes in the ocean/atmosphere models.

In general, we strongly believe that ETLM methodology, as a new branch of the ensemble DA techniques, have a very good prospect for development within the current parallelization trends in computer technologies which have significantly propelled, in particular, the ML/AI techniques in recent years.

ACKNOWLEDGEMENTS

This study was supported by the Office of Naval Research projects (program elements 0646352N, 0602435N, and 0603207N). Helpful discussions with Prof. C. Beattie and Dr. G. Pantelev are acknowledged.

ORCID

Max Yaremchuk  <https://orcid.org/0000-0002-3280-5490>

REFERENCES

- Allen, D., Bishop, C. and Frolov, S. (2017) Hybrid 4D-Var with a local ensemble tangent linear model: application to the shallow-water model. *Monthly Weather Review*, 145, 97–145.
- Bannister, R.N. (2016) A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143, 607–633.
- Bishop, C.H., Frolov, S., Allen, D.R., Kuhl, D.D. and Hoppel, K. (2017) The local ensemble tangent linear model: an enabler for coupled model 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 143, 1009–1020.
- Bonavita, M., Tremolet, Y., Holm, E., Lang, S., Chrust, M., Janiskova, M., Lopez, P., Laloyaux, P., Rosnay, P., Fisher, M., Harmud, M.

- and English, S. (2017). A strategy for data assimilation. Technical Memorandum 800, ECMWF, Reading, UK.
- Buehner, M., McTaggart-Cowan, R., Beaulne, A., Charette, C., Garand, L., Heilliette, S., Lapalme, E., Laroche, S., Macpherson, S.P., Morneau, J. and Zadra, A. (2015) Implementation of deterministic weather forecasting systems based on ensemble-variational data assimilation in Environment Canada. Part I: the global system. *Monthly Weather Review*, 143, 2532–2559.
- Fairbairn, D., Pring, S.R., Lorenc, A.C. and Roulstone, I. (2014) A comparison of 4D-Var with ensemble data assimilation methods. *Quarterly Journal of the Royal Meteorological Society*, 140, 281–294.
- Frolov, S. and Bishop, C. (2016) Localized ensemble-based tangent linear models and their use in propagating hybrid error covariance models. *Monthly Weather Review*, 144, 1383–1405.
- Frolov, S., Allen, D., Bishop, C., Langland, R., Hoppel, K.W. and Kuhl, D.D. (2018) First application of the local ensemble tangent linear model (LETLM) to a realistic model of the global atmosphere. *Monthly Weather Review*, 146, 2247–2270.
- Hoteit, I., Cornuelle, B., Kohl, A. and Stammer, D. (2005) Treating strong adjoint sensitivities in tropical eddy-permitting variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 131, 3659–3682.
- Kuhl, D.D., Rosmond, T.E., Bishop, C.H., McLay, J. and Baker, N. (2013) Comparison of hybrid ensemble/4D-Var and 4D-Var within the NAVDAS-AR data assimilation framework. *Monthly Weather Review*, 141, 2740–2758.
- Lorenc, A.C., Bowler, N.E., Clayton, A.M., Pring, S.R. and Fairbairn, D. (2015) Comparison of hybrid-4DVar and hybrid-4DVar data assimilation methods for global NWP. *Monthly Weather Review*, 143, 212–229.
- Lorenc, A.C. and Jardak, M. (2008) A comparison of hybrid variational data assimilation methods for global NWP. *Quarterly Journal of the Royal Meteorological Society*, 144, 2748–2760. <https://doi.org/10.1002/qj.3401>
- Liu, C., Xiao, Q. and Wang, B. (2008) An ensemble-based four-dimensional variational data assimilation scheme. Part I: technical formulation and preliminary test. *Monthly Weather Review*, 136, 3363–3373.
- Schmidt, M. (2005). MinFunc: unconstrained differentiable multivariate optimization in Matlab. Available at: <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html>; accessed 14 December 2019.
- Wang, Y., Meng, D. and Yuang, M. (2018) Sparse recovery: from vectors to tensors. *National Science Review*, 5, 756–767.
- Xiu, Q. (2005) Representations of inverse covariances by differential operators. *Advances in Atmospheric Sciences*, 22(2), 181–198.
- Yaremchuk, M., Nechaev, D. and Pantelev, G. (2009) A method of successive corrections of the control subspace in the reduced-order variational data assimilation. *Monthly Weather Review*, 137, 2966–2978.
- Yaremchuk, M., Carrier, M., Smith, S. and Jacobs, G. (2013). Background error correlation modeling with diffusion operators, in *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications*, vol. 3, pp. 83–114. Park, S., Xu, L. (eds). Springer, Berlin.
- Yaremchuk, M. and Martin, P. (2014) On sensitivity analysis within the 4DVar framework. *Monthly Weather Review*, 142, 774–787.
- Yaremchuk, M., Martin, P. and Beattie, C. (2017) A hybrid approach to generating search subspaces in the dynamically constrained data assimilation. *Ocean Modelling*, 117, 41–51.
- Zhang, M. and Zhang, F. (2012) E4DVar: coupling an ensemble Kalman filter with four-dimensional variational data assimilation in a limited-area weather prediction model. *Monthly Weather Review*, 140, 587–600.

How to cite this article: Yaremchuk M, Nechaev D, Frolov S. On the ensemble-based linearization of numerical models. *QJR Meteorol Soc.* 2020;146:1026–1039. <https://doi.org/10.1002/qj.3723>

APPENDICES

A. RETRIEVING A LOCAL LINEAR OPERATOR FROM THE ENSEMBLE

Denote non-zero elements in a j th row of \mathbf{M} by m_i , $i = 1, \dots, n^s$ and assume that \mathbf{M} is local, that is, all these elements are located not farther than l grid steps from the point x in physical space corresponding to the row under consideration. Let n_f be the number of field constituents in the state vector and $i_k, k = 1, \dots, n_f(2l + 1)^2$ be the indices enumerating global positions of the columns of \mathbf{M} containing non-zero elements in the above-mentioned vicinity ω_x of x . Then the ensemble elements $\mathbf{x}^j(x)$ in the j th row of the output ensemble matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times m}$ are given by

$$\tilde{X}^j = \sum_{k=1}^{n^s} m_{i_k} X_{i_k}^j, \quad j = 1, \dots, m, \quad (\text{A1})$$

where $X_{i_k}^j$ are the local elements of the input ensemble $\mathbf{X}(\omega_x)$ and $n^s = n_f(2l + 1)^2$ is the upper limit of the stencil size.

The relationships (Equation (A1)) can be considered as a system of m linear equations with n^s unknown values of the matrix elements m_{i_k} mapping the input ensemble elements $\mathbf{X}(\omega_x)$ to $\tilde{\mathbf{X}}(x)$. The system is overdetermined if $m > n^s$ and has a unique solution if the ensemble members are locally linearly independent, which is usually the case. In the present study, $n_f = 3$ and $l = 1$, so that the threshold ensemble size is $m = 27$.

It is necessary to note that in real applications (e.g., featuring upwind finite differences, or semi-Lagrangian

advection schemes), the stencil's structure and size depend on x , requiring additional calculations of the stencil element locations which is not a straightforward numerical procedure in practice. This procedure could be omitted if a reasonably good estimate is available of the stencil size, shape and position relative to the diagonal of the system matrix. In the present study, we kept ω_x constant, solving $m \times m$ systems of equations in every grid point.

For 3D models, n^s can be as high as several hundred. However, this number could be reduced considerably by more accurate accounting of the number of field dependencies at a point and of the stencil shapes, which may not, for example, contain non-zero elements in the vertices/edges of the stencil cube.

B. NUMERICAL SCHEME

Adopt the units $\delta x = \delta t = 1$, let $l_x = l_y$ be the domain size and \mathbf{s}_x denote the $l_x \times l_x$ "right shift" matrix obtained from the identity matrix \mathbf{i} by displacing its diagonal one step to the right and placing 1 at the bottom of the first column to account for the periodicity. The 1D "forward" and "backward" finite difference and averaging operators are then given by

$$\mathbf{d}_x^+ = \mathbf{s}_x^+ - \mathbf{i}, \quad \mathbf{d}_x^- = -\mathbf{d}_x^T, \quad \mathbf{a}_x^+ = (\mathbf{i} + \mathbf{s}_x)/2 \quad \text{and} \\ \mathbf{a}_x^- = \mathbf{a}_x^{+T}.$$

Using the above notation and \otimes for the Kronecker product, the $l_x l_y \times l_x l_y$ matrix representations of the differential and averaging operators in two dimensions are defined by

$$\partial_x^\pm = \mathbf{d}_x^\pm \otimes \mathbf{i}_y, \quad \partial_y^\pm = \mathbf{i}_x \otimes \mathbf{d}_y^\pm, \quad (\text{B1})$$

$$\nabla^2 = (\mathbf{d}_x^+ - \mathbf{d}_x^-) \otimes \mathbf{i}_y + \mathbf{i}_x \otimes (\mathbf{d}_y^+ - \mathbf{d}_y^-), \quad (\text{B2})$$

$$\mathbf{A}_x^\pm = \mathbf{a}_x^\pm \otimes \mathbf{i}_y, \quad \mathbf{A}_y^\pm = \mathbf{i}_x \otimes \mathbf{a}_y^\pm, \quad (\text{B3})$$

$$\mathbf{A}_v = \mathbf{A}_x^- \mathbf{A}_y^+, \quad \mathbf{A}_u = \mathbf{A}_x^+ \mathbf{A}_y^-. \quad (\text{B4})$$

The Crank–Nicolson scheme is specified by time-centring the result of action by a system matrix \mathbf{S} on a state vector:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \frac{1}{2} [(\mathbf{S}\mathbf{x})_{k+1} + (\mathbf{S}\mathbf{x})_k]. \quad (\text{B5})$$

Since the system matrix \mathbf{S} is linear in \mathbf{x} , its action on \mathbf{x} can be represented in the form $\mathbf{S}\mathbf{x} = \mathbf{S}_0\mathbf{x} + \mathbf{N}'\mathbf{x}\mathbf{x}^T$, where the entries of \mathbf{S}_0 and \mathbf{N}' are independent of \mathbf{x} . Introducing the notation $\delta\mathbf{x} = \mathbf{x}_{k+1} - \mathbf{x}_k$, and assuming that $|\delta\mathbf{x}| \ll |\mathbf{x}_k|$ yields the following approximation for the nonlinear

part of the r.h.s. in Equation (B5),

$$\mathbf{N}' [(\mathbf{x}_k + \delta\mathbf{x})(\mathbf{x}_k + \delta\mathbf{x})^T + \mathbf{x}_k\mathbf{x}_k^T] \approx \mathbf{N}' [\mathbf{x}_k\mathbf{x}_{k+1}^T + \mathbf{x}_{k+1}\mathbf{x}_k^T].$$

The numerical scheme (Equation (B5)) can thus be rewritten in the form:

$$\left[\mathbf{I} + \frac{1}{2}(\mathbf{N}_k - \mathbf{Q}) \right] \mathbf{x}_{k+1} = \left[\mathbf{I} + \frac{1}{2}\mathbf{Q} \right] \mathbf{x}_k,$$

where

$$\mathbf{Q} = - \begin{bmatrix} \mathbf{0} & \partial_x^+ \langle \mathbf{A}_x^- h \rangle & \partial_y^+ \langle \mathbf{A}_y^- h \rangle \\ g \partial_x^- & \mu \mathbf{I} + \nu \Delta & f \mathbf{A}_v \\ g \partial_y^- & -f \mathbf{A}_u & \mu \mathbf{I} + \nu \Delta \end{bmatrix}, \\ \mathbf{N}_k = \begin{bmatrix} \mathbf{Z} & \partial_x^+ \langle \mathbf{A}_x^- \eta_k \rangle & \partial_y^+ \langle \mathbf{A}_y^- \eta_k \rangle \\ \mathbf{0} & \mathbf{U} & \langle \partial_y^c u_k \rangle \mathbf{A}_v \\ \mathbf{0} & \langle \partial_x^c v_k \rangle \mathbf{A}_u & \mathbf{V} \end{bmatrix}.$$

Here $\partial^c = (\partial^+ + \partial^-)/2$ is the central difference operator, and the following notations are adopted:

$$\langle w \rangle = \text{diag}(w), \quad \mathbf{I} = \mathbf{i} \otimes \mathbf{i}, \\ \mathbf{Z} = \partial_x^+ \langle u_k \rangle \mathbf{A}_x^- + \partial_y^+ \langle v_k \rangle \mathbf{A}_y^-, \quad (\text{B6})$$

$$\mathbf{U} = \langle \partial_x^c u_k \rangle + \langle u_k \rangle \partial_x^c + \langle \mathbf{A}_v v_k \rangle \partial_y^c, \quad (\text{B7})$$

$$\mathbf{V} = \langle \partial_y^c v_k \rangle + \langle v_k \rangle \partial_y^c + \langle \mathbf{A}_u u_k \rangle \partial_x^c. \quad (\text{B8})$$

C. THE TANGENT LINEAR OPERATOR

By definition, the tangent linear model \mathbf{M}^* is linearization of $\mathbf{M}(\mathbf{x})\mathbf{x}$ in the vicinity of $\bar{\mathbf{x}}$:

$$\mathbf{M}^* \delta\mathbf{x} = \left[\bar{\mathbf{M}} + \frac{\delta\mathbf{M}}{\delta\mathbf{x}} \bar{\mathbf{x}} \right] \delta\mathbf{x}. \quad (\text{C1})$$

In the above equation and hereinafter, the derivatives are taken at $\bar{\mathbf{x}}$.

Using the notation of Section 3.1 (Equation (16)) and the relationships

$$\frac{\delta\mathbf{L}^{-1}\mathbf{L}}{\delta\mathbf{x}} = \frac{\delta\mathbf{L}^{-1}}{\delta\mathbf{x}}\mathbf{L} + \mathbf{L}^{-1}\frac{\delta\mathbf{L}}{\delta\mathbf{x}} = 0, \quad \frac{\delta\mathbf{L}}{\delta\mathbf{x}} = \frac{1}{2}\frac{\delta\mathbf{N}}{\delta\mathbf{x}},$$

the second term in Equation (C1) can be rearranged in the form

$$\frac{\delta\mathbf{M}}{\delta\mathbf{x}} \bar{\mathbf{x}} = \frac{\delta\mathbf{L}^{-1}\mathbf{L}_0}{\delta\mathbf{x}} \bar{\mathbf{x}} = \frac{\delta\mathbf{L}^{-1}}{\delta\mathbf{x}} \mathbf{L}_0 \bar{\mathbf{x}} \\ = -\bar{\mathbf{L}}^{-1} \frac{\delta\mathbf{L}}{\delta\mathbf{x}} \bar{\mathbf{L}}^{-1} \mathbf{L}_0 \bar{\mathbf{x}} = -\frac{1}{2} \bar{\mathbf{L}}^{-1} \frac{\delta\mathbf{N}}{\delta\mathbf{x}} \bar{\mathbf{L}}^{-1} \mathbf{L}_0 \bar{\mathbf{x}}. \quad (\text{C2})$$

Substitution of the r.h.s. from Equation (C2) into (C1) yields the expression for \mathbf{M}^* in Equation (18):

$$\mathbf{M}^* = \overline{\mathbf{M}} - \frac{1}{2} \overline{\mathbf{L}}^{-1} \frac{\delta \mathbf{N}}{\delta \mathbf{x}} \overline{\mathbf{L}}^{-1} \mathbf{L}_0 \overline{\mathbf{x}}. \quad (\text{C3})$$

In this study, the matrix elements of \mathbf{N} are linear in \mathbf{x} (Equations (5)–(B8)). As a consequence, the product of $\delta \mathbf{N} / \delta \mathbf{x}$ by an arbitrary vector \mathbf{a} is the value of \mathbf{N} at \mathbf{a} :

$$\frac{\delta \mathbf{N}}{\delta \mathbf{x}} \mathbf{a} = \mathbf{N}(\mathbf{a}). \quad (\text{C4})$$

Assuming that $\mathbf{a} = \overline{\mathbf{L}}^{-1} \mathbf{L}_0 \overline{\mathbf{x}} \equiv \overline{\mathbf{M}} \overline{\mathbf{x}}$ in Equation (C3), and taking Equation (C4) into account, Equation (18) takes the form Equation (23).