SPECIAL
4D-Var ↔ EnKF
COLLECTION

# Efficient Ensemble Covariance Localization in Variational Data Assimilation

CRAIG H. BISHOP AND DANIEL HODYSS
*Naval Research Laboratory, Monterey, California*

PETER STEINLE AND HOLLY SIMS
*Bureau of Meteorology, Melbourne, Australia*

ADAM M. CLAYTON, ANDREW C. LORENC, AND DALE M. BARKER
*Met Office, Exeter, United Kingdom*

MARK BUEHNER
*Meteorological Research Division, Environment Canada, Dorval, Canada*

(Manuscript received 2 March 2010, in final form 31 August 2010)

ABSTRACT

Previous descriptions of how localized ensemble covariances can be incorporated into variational (VAR) data assimilation (DA) schemes provide few clues as to how this might be done in an efficient way. This article serves to remedy this hiatus in the literature by deriving a computationally efficient algorithm for using nonadaptively localized four-dimensional (4D) or three-dimensional (3D) ensemble covariances in variational DA. The algorithm provides computational advantages whenever (i) the localization function is a separable product of a function of the horizontal coordinate and a function of the vertical coordinate, (ii) and/or the localization length scale is much larger than the model grid spacing, (iii) and/or there are many variable types associated with each grid point, (iv) and/or 4D ensemble covariances are employed.

## 1. Introduction

Work by Lorenc (2003), Buehner (2005), Wang et al. (2007), Buehner et al. (2010a,b), Liu et al. (2009), and others indicate a growing interest in using localized ensemble covariances in three- and four-dimensional variational data assimilation (3D-VAR and 4D-VAR, respectively) schemes. Buehner et al. (2010a,b) found that a 4D-VAR scheme that used localized 3D ensemble covariances for a background error covariance matrix and a tangent linear model (TLM) and adjoint, outperformed versions of operational 4D-VAR and ensemble Kalman filter (EnKF) data assimilation (DA) schemes. They also showed that an ensemble-4D-VAR scheme that only used localized 4D ensemble covariances and did not use a TLM or adjoint, outperformed a version of the operational 4D-VAR scheme both in the tropics and Southern Hemisphere, but not in the northern extratropics. They also mention that, without including the cost of generating the ensemble, ensemble-4D-Var could be performed at a fraction of the cost of a conventional 4D-Var scheme. However, the descriptions of how ensemble covariances can be incorporated into variational schemes given in Lorenc (2003), Buehner (2005), Wang et al. (2007), and Buehner et al. (2010a,b) give scant details or understanding of the factors that influence efficiency. This paper provides understanding vital to the design and implementation of fast ensemble covariance localization algorithms for variational DA. The fast algorithm derived pertains to Buehner et al's (2010a,b) ensemble-4D-VAR in which the 4D ensemble covariances are nonadaptively localized in space alone, with no localization in time or modulation of covariances between variable types. The

*Corresponding author address:* Craig H. Bishop, Naval Research Laboratory, Marine Meteorology Division, 7 Grace Hopper Ave., Stop 2, Building 702, Room 212, Monterey, CA 93943-5502.
E-mail: bishop@nrlmry.navy.mil

DOI: 10.1175/2010MWR3405.1

fast algorithm draws heavily from techniques that have been developed for nonensemble variational DA.

Variational DA algorithms perform a global minimization of a penalty function containing a background or prior term written in terms of a background error covariance matrix $\mathbf{P}_0^f$. This is a nonsparse matrix defining the error covariance between every pair of points and variables in a forecast model, which for modern NWP systems may have a billion degrees of freedom. Not only is it computationally impossible to store and manipulate matrices of this size, but it is also impossible to know them (Dee 1991): there is not enough information in all the forecasts made in the lifetime of an NWP system to accurately estimate even a static covariance. Another problem is that the poor conditioning of the covariance prevents efficient minimization. In 4D-VAR algorithms, these problems are addressed together, by representing $\mathbf{P}_0^f$ in terms of a square root that is implied by a sequence of simple transformations. These simple transformations give the necessary cost savings and, just as importantly, can impart covariances based on a physical understanding of the dynamics governing forecast error. The transformations define the covariance, rather than being chosen to fit a known or hypothesized covariance. Traditional variational assimilation methods use a parameter transform to balanced and unbalanced variables that are expected to be uncorrelated, together with vertical and horizontal transforms that model the three-dimensional correlation structure in each of the transformed variables (e.g., Parrish and Derber 1992).

In traditional variational methods, the designer of the background error covariance matrix selects transform operators that implicitly define the background error covariance matrix. In contrast, designers of ensemble covariance localization methods such as Bishop and Hodyss (2007, 2009a,b) explicitly design localization functions with which to multiply explicit ensemble covariance functions. Apart from Kepert's (2009) sophisticated formulation of balance constraints for ensemble Kalman filters, expressions for localized ensemble covariance matrices are, in general, trivial to write down in matrix form. In contrast, the background error covariance matrix implied by the sequence of transformations and balance operators that define the background error covariance matrices of traditional variational methods are nontrivial to write down in matrix form. This note shows how the easily written down matrix vector product associated with a nonadaptively localized ensemble covariance matrix can be reduced to a sequence of cost-efficient transform operators similar to those used in variational DA schemes.

Section 2 shows how a straightforward coding of the matrix-vector multiplications would be extremely expensive for forecast error covariance matrices based on localized ensemble covariance matrices. Section 3 derives a fast algorithm for performing this multiplication. Conclusions follow in section 4.

## 2. The problem

Consider the localized 4D ensemble covariance matrix given by $\mathbf{P}^f = \mathbf{P}_K^f \odot \tilde{\mathbf{C}}$, where $\odot$ indicates the element-wise matrix product, and $\tilde{\mathbf{C}} = \tilde{\mathbf{C}}^{1/2}\tilde{\mathbf{C}}^{1/2\mathrm{T}}$ is the correlation matrix used to localize the ensemble covariance matrix $\mathbf{P}_K^f = 1/(K-1)\sum_{i=1}^{K}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}$, where $\mathbf{x}_i$ is the $i$th ensemble perturbation (ensemble member minus the ensemble mean) outputted at a series of $n_q$ times across the data assimilation time window, and $\tilde{\mathbf{C}}^{1/2}$ is the left square root of $\tilde{\mathbf{C}}$.

As shown in the appendix and as discussed in Buehner (2005) and Bishop and Hodyss (2009b), the square root of $\mathbf{P}_K^f \odot \tilde{\mathbf{C}}$ is given by $[\mathrm{diag}(\mathbf{x}_1)\tilde{\mathbf{C}}^{1/2}, \mathrm{diag}(\mathbf{x}_2)\tilde{\mathbf{C}}^{1/2}, \ldots, \mathrm{diag}(\mathbf{x}_K)\tilde{\mathbf{C}}^{1/2}]$, where $\mathrm{diag}(\mathbf{x}_i)$ is the diagonal matrix whose diagonal is composed of the elements of $\mathbf{x}_i$.

Variational algorithms that minimize a measure of the distance of the estimated state from observations and the first guess require the repeated evaluation of terms like

$$[\mathrm{diag}(\mathbf{x}_i)\tilde{\mathbf{C}}^{1/2}]\mathbf{f}_i = \mathbf{x}_i \odot (\tilde{\mathbf{C}}^{1/2}\mathbf{f}_i) \qquad (1)$$

and

$$\mathbf{p}_i = [\mathrm{diag}(\mathbf{x}_i)\tilde{\mathbf{C}}^{1/2}]^{\mathrm{T}} \quad \mathbf{h} = [\tilde{\mathbf{C}}^{1/2}]^{\mathrm{T}}(\mathbf{x}_i \odot \mathbf{h}). \qquad (2)$$

Here $\mathbf{f}_i$ has as many elements as there are columns in $\tilde{\mathbf{C}}^{1/2}$ while $\mathbf{h}$ has as many elements as the 4D state vector. The meaning of the vectors $\mathbf{f}_i$ and $\mathbf{h}$ and the order in which the operations (1) and (2) are performed depends on whether the primal or dual forms of variational data assimilation are being used—see El Akkraoui et al. (2008) for details. For the purposes of this paper, $\mathbf{f}_i$ and $\mathbf{h}$ are simply symbols that denote the vector part of the matrix vector products defined by (1) and (2). The symbol $\mathbf{p}_i$ will be used as shorthand for $[\mathrm{diag}(\mathbf{x}_i)\tilde{\mathbf{C}}^{1/2}]^{\mathrm{T}}\mathbf{h}$.

If the sole component of the forecast error covariance matrix is localized ensemble covariances, the analysis correction $\mathbf{x}^a - \mathbf{x}^f$ takes the following form: $\mathbf{x}^a - \mathbf{x}^f = 1/\sqrt{K-1}\sum_{i=1}^{K}\mathbf{x}_i \odot (\tilde{\mathbf{C}}^{1/2}\mathbf{f}_i)$. In other words, the analysis correction is a weighted sum of the ensemble perturbations $\mathbf{x}_i$. The vector of weights for the $i$th ensemble perturbation is given by the vector $(\tilde{\mathbf{C}}^{1/2}\mathbf{f}_i)$. The less the values in the vector $(\tilde{\mathbf{C}}^{1/2}\mathbf{f}_i)$ vary with space, time, and variable type, the more the global analysis approximates a simple linear combination of global ensemble perturbations. The more the values in the vector $(\tilde{\mathbf{C}}^{1/2}\mathbf{f}_i)$ vary with space, time, and variable type, the freer the global analysis is to fit observations. Here, however, we consider the nonadaptive, purely spatial localization used with Buehner et al's (2010a,b) ensemble-4D-VAR scheme. In this scheme, the values of $(\tilde{\mathbf{C}}^{1/2}\mathbf{f}_i)$ depend only on the spatial location of the grid point to which they pertain and are independent of time and variable type. The scale of the correlation functions defining $\tilde{\mathbf{C}}$ determine the penalty associated with allowing the ensemble weights to vary. Broadscale correlation functions in $\tilde{\mathbf{C}}$ only permit broadscale fluctuations of the weights while short-scale correlation functions allow shorter-scale fluctuations of the weights. In this way, the localization correlation matrix $\tilde{\mathbf{C}}$ controls the spatial variation of the ensemble weights defining the analysis correction. In contrast, in the local ensemble transform Kalman filter the variation of ensemble weights is controlled by the assigned sizes of observation volumes and distance-dependent observation error variance inflation (Hunt et al. 2007; Miyoshi and Yamane 2007; Szunyogh et al. 2008; Yang et al. 2009).

We assume that the columns of $\tilde{\mathbf{C}}^{1/2}$ are separable functions; that is, each column is the product of a member of a set of basis functions of the horizontal coordinate and a member of a set of basis functions of the vertical coordinate. Under this assumption, the number of columns $n_c$ in $\tilde{\mathbf{C}}^{1/2}$ is given by the product of the number of horizontal basis functions $L_h$ and the number of vertical basis functions $M$; in other words, $n_c = ML_h$. We will also assume that the model has $\tilde{n}_h$ horizontal grid points and $\tilde{n}_z$ vertical levels. For the sake of simplicity, we shall assume that the number of variable types at each model grid point is equal to the constant $n_\varpi$. The total number of variables is then $\tilde{n}_T = \tilde{n}_h\tilde{n}_z n_q n_\varpi$ (recall that $n_q$ is the number of time levels). The cost of computing (nonsparse) matrix-vector products like those in (1) and (2) is order $ML_h\tilde{n}_T$ operations. Assuming $[\tilde{n}_h, \tilde{n}_z, n_\varpi, n_q] \sim O[10^6, 10^2, 10, 10]$ so that $\tilde{n}_T \sim O(10^{10})$, and assuming that $[L_h, M] \sim O[10^5, 10]$ so that $ML_h \sim 10^6$, it follows that $ML_h\tilde{n}_T \sim O(10^{16})$ operations would be required for the matrix vector products in (1) and (2). This study shows how this cost can be reduced by many orders of magnitude.

## 3. Derivation of a rapid algorithm for ensemble covariance localization

A first step to a rapid algorithm is to recognize that parts of it can be performed on a reduced resolution grid. Consider a coarse-resolution grid that only has $n_h \times n_z$ grid points giving a total of just $n_T = n_q n_h n_z n_\varpi$ variables instead of the $\tilde{n}_h \times \tilde{n}_z$ grid points and $\tilde{n}_T$ variables of the high-resolution grid. Assume that the coarse-resolution grid has $n_h = \tilde{n}_h/u$ grid points in the horizontal and $n_z = \tilde{n}_z/s$ grid points in the vertical, where $u$ and $s$ are scalars.[1] Let $\mathbf{L}$ be a $\tilde{n}_T \times n_T$ sparse rectangular matrix operator that interpolates coarse-resolution $n_T$ vectors to high-resolution $\tilde{n}_T$ vectors. When the horizontal and vertical length scales of the functions that describe the columns of $\tilde{\mathbf{C}}^{1/2}$ are broad, $\tilde{\mathbf{C}}^{1/2} \approx \mathbf{L}\mathbf{C}^{1/2}$, and (1) and (2) can be approximated by

$$[\mathrm{diag}(\mathbf{x})\tilde{\mathbf{C}}^{1/2}]\mathbf{f} = \mathbf{x} \odot (\tilde{\mathbf{C}}^{1/2}\mathbf{f}) \approx \mathbf{x} \odot [\mathbf{L}(\mathbf{C}^{1/2}\mathbf{f})] \qquad (3)$$

and

$$\mathbf{p} = [\mathrm{diag}(\mathbf{x})\tilde{\mathbf{C}}^{1/2}]^{\mathrm{T}} \quad \mathbf{h} = [\tilde{\mathbf{C}}^{1/2}]^{\mathrm{T}}(\mathbf{x} \odot \mathbf{h}) \approx [\mathbf{C}^{1/2}]^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}(\mathbf{x} \odot \mathbf{h}). \qquad (4)$$

Notice that for the sake of conciseness, in (3) and (4) and throughout the rest of the paper, we drop the ensemble index subscript $i$ from **f**, **x**, and **p** (except in section a of the appendix where it is needed again). The cost of mapping from high- to low-resolution grids and vice versa is order $\tilde{n}_T$ operations. Without any further simplification, the use of the coarse grid reduces the cost of (1) and (2) to order $ML_h\tilde{n}_T/(su)$ operations. However, as we will see below, there are even more profound gains in computational cost that can be obtained by judicious use of the separable nature of the functions that define the columns of $\mathbf{C}^{1/2}$.

When each column of $\mathbf{C}^{1/2}$ is proportional to a product of vertical and horizontal structure functions, we can use the modulation product with symbol "$\triangle$"—defined in the appendix and in Bishop and Hodyss (2009b)—to write

$$\mathbf{C}^{1/2} = \mathbf{W}^v \triangle \mathbf{W}^h, \qquad (5)$$

where each column of the $n_T \times L_h$ matrix $\mathbf{W}^h$ consists of a single horizontal structure (such as a scaled spherical harmonic in a global model) replicated at all vertical levels, all time levels, and all of the $n_\varpi$ variable types. Each column of the $n_T \times M$ matrix $\mathbf{W}^v$ consists of a single vertical structure (such as a scaled eigenvector of a vertical correlation matrix) that is the same for all variable types and has no horizontal or temporal variation. In other words, the single vertical structure is replicated in the horizontal and through time for all variables. According to the definition of the modulation product given in the appendix, (5) means that $\mathbf{C}^{1/2}$ has $n_c = L_hM$ columns.

To define $\mathbf{W}^h$ algebraically, let $\mathbf{R}^h$ be a $n_T \times n_h$ replication matrix and let $\mathbf{G}$ be a $n_h \times L_h$ matrix listing the $L_h$ horizontal basis functions defining a square root of the horizontal correlation matrix associated with $\mathbf{C}^{1/2}$. We then have

$$\mathbf{W}^h = \mathbf{R}^h\mathbf{G} = \begin{bmatrix} \mathbf{R}^h_{111} \\ \mathbf{R}^h_{211} \\ \cdot \\ \mathbf{R}^h_{mqr} \\ \cdot \\ \mathbf{R}^h_{n_z n_q n_\varpi} \end{bmatrix}\mathbf{G} = \begin{bmatrix} \mathbf{I} \\ \cdot \\ \mathbf{I} \\ \cdot \\ \mathbf{I} \end{bmatrix}\mathbf{G}, \qquad (6)$$

[1] Recent implementations of efficient localization at Environment Canada and the Met Office did not degrade vertical resolution; however, for the reasons given in section 3, moving to a coarser vertical resolution should provide almost equivalent results at a smaller computational cost.

where the replication matrix $\mathbf{R}^h$ is defined by the submatrices $\mathbf{R}^h_{mqr} = \mathbf{I}$ for all vertical levels $m = 1, 2, \ldots, n_z$, all times $q = 1, 2, \ldots, n_q$, and all variable types $r = 1, 2, \ldots, n_\varpi$, where $\mathbf{I}$ is the $n_h \times n_h$ identity matrix. Note that in (6), we have assumed that the $n_h n_z n_q n_\varpi$ elements of the state vector have been listed such that the first $n_h$ elements of the state vector correspond to the horizontal field defining the variation of the first variable at the first time at model level 1. The second $n_h$ elements of the state vector correspond to the horizontal field defining the variation of the first variable at model level 2 and so on, until all $n_h n_z$ variables defining the 3D state of the first variable at the first time level have been defined. The next $n_h n_z$ variables define the state at the second time level until all $n_h n_z n_q$ variables have been listed that define the 4D state of the first variable. The next $n_h n_z n_q$ define the 4D state of the second model variable, and so on until all $n_h n_z n_q n_\varpi$ elements defining the 4D state of all model variables have been listed.

Here $\mathbf{W}^v$ is completely analogous to $\mathbf{W}^h$, but its algebraic form changes to accommodate the noncontiguous location of the vertical indices. To define $\mathbf{W}^v$ algebraically, let the $n_z \times M$ matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j, \ldots, \mathbf{v}_M]$ be the square root of the vertical correlation matrix associated with $\mathbf{C}^{1/2}$. The $j$th column of the $n_z n_h \times M$ matrix

$$\mathbf{R}^z\mathbf{V} = \begin{bmatrix} \mathbf{1}(1) & \mathbf{0} & ..\,\mathbf{0} & & ..\mathbf{0} \\ \mathbf{0} & \mathbf{1}(2) & & & \\ . & & & & \\ \mathbf{0} & & & \mathbf{1}(m) & ..\mathbf{0} \\ .. & & & & \\ \mathbf{0} & \mathbf{0} & & ..\mathbf{0} & \mathbf{1}(n_z) \end{bmatrix}\mathbf{V}, \quad \text{where}$$

$$\mathbf{1}^{\mathrm{T}}(m) = \overbrace{[1, 1, \ldots, 1]}^{n_h} \quad \text{for any } m, \tag{7}$$

then gives a 3D function with vertical columns equal to $\mathbf{v}_j$ and no horizontal variation. It then remains to replicate this 3D structure function for all variable types and all output times using the replication matrix $\mathbf{R}^v$ defined by

$$\mathbf{W}^v = \mathbf{R}^v\mathbf{V} = \left\{\begin{bmatrix} \mathbf{I}_{11} \\ \mathbf{I}_{21} \\ . \\ \mathbf{I}_{qr} \\ . \\ \mathbf{I}_{n_q n_\varpi} \end{bmatrix}\mathbf{R}^z\right\}\mathbf{V}, \quad \text{where} \quad \mathbf{I}_{qr} \text{ is the}$$

$$n_h n_z \times n_h n_z \text{ identity matrix for all } q \text{ and } r. \tag{8}$$

Using (5) in the RHS of (3) gives

$$\mathbf{x} \odot [\mathbf{L}(\mathbf{C}^{1/2}\mathbf{f})] = \mathbf{x} \odot \{\mathbf{L}([\mathbf{W}^v \vartriangle \mathbf{W}^h]\mathbf{f})\}. \tag{9}$$

The elements of $\mathbf{f}$ are coefficients of the columns of $\mathbf{W}^v \vartriangle \mathbf{W}^h$. Hence, $[\mathbf{W}^v \vartriangle \mathbf{W}^h]\mathbf{f}$ is a mapping from coefficient space to physical space. In section b of the appendix, we prove that $[\mathbf{W}^v \vartriangle \mathbf{W}^h]\mathbf{f} = \sum_{j=1}^M \mathbf{w}^v_j \odot [(\mathbf{R}^h\mathbf{G})\mathbf{f}_j] = \sum_{j=1}^M \mathbf{w}^v_j \odot [\mathbf{R}^h(\mathbf{G}\mathbf{f}_j)]$, where $\mathbf{f}_j^{\mathrm{T}}$ is a $L_h$ vector and $\mathbf{f}^{\mathrm{T}} = [\mathbf{f}_1^{\mathrm{T}}, \mathbf{f}_2^{\mathrm{T}}, \ldots, \mathbf{f}_j^{\mathrm{T}}, \ldots, \mathbf{f}_M^{\mathrm{T}}]$. Using this, (6) and (8) in (9) gives

$$\mathbf{x} \odot [\mathbf{L}(\mathbf{C}^{1/2}\mathbf{f})] = \mathbf{x} \odot \left\langle \mathbf{L}\left\{\sum_{j=1}^{M} (\mathbf{R}^v\mathbf{v}_j) \odot [\mathbf{R}^h(\mathbf{G}\mathbf{f}_j)]\right\}\right\rangle. \tag{10}$$

To rephrase (10) in a form that avoids the explicit use of the replication matrices $\mathbf{R}^v$ and $\mathbf{R}^h$, let $(\mathbf{C}^{1/2}\mathbf{f})_{qrl} = \{\sum_{j=1}^M (\mathbf{R}^v\mathbf{v}_j) \odot [\mathbf{R}^h(\mathbf{G}\mathbf{f}_j)]\}_{qrl}$ denote a vertical column of the vector $\sum_{j=1}^M (\mathbf{R}^v\mathbf{v}_j) \odot [\mathbf{R}^h(\mathbf{G}\mathbf{f}_j)]$ corresponding to the $r$th variable type, $q$th time level, and $l$th horizontal location. One may then deduce from (10) and (6)–(9) that

$$\begin{aligned} (\mathbf{C}^{1/2}\mathbf{f})_{qrl} &= \left\{\sum_{j=1}^{M} (\mathbf{R}^v\mathbf{v}_j) \odot [\mathbf{R}^h(\mathbf{G}\mathbf{f}_j)]\right\}_{qrl} = \sum_{j=1}^{M} (\mathbf{v}_j) \odot \left\{\begin{bmatrix} \mathbf{R}^h_{1qr} \\ \mathbf{R}^h_{2qr} \\ . \\ \mathbf{R}^h_{n_z qr} \end{bmatrix}(\mathbf{G}\mathbf{f}_j)\right\}_l \\ &= \sum_{j=1}^{M} (\mathbf{v}_j) \odot \begin{bmatrix} \mathbf{G}\mathbf{f}_j \\ \mathbf{G}\mathbf{f}_j \\ . \\ \mathbf{G}\mathbf{f}_j \end{bmatrix}_l, \quad \text{because} \quad \mathbf{R}^h_{mqr} = \mathbf{I} \\ &= \sum_{j=1}^{M} \mathbf{v}_j[(\mathbf{G}\mathbf{f}_j)]_l = \sum_{j=1}^{M} \mathbf{v}_j(\mathbf{g}_l^{\mathrm{T}}\mathbf{f}_j) = \mathbf{V}\begin{bmatrix} \mathbf{g}_l^{\mathrm{T}}\mathbf{f}_1 \\ \mathbf{g}_l^{\mathrm{T}}\mathbf{f}_2 \\ . \\ \mathbf{g}_l^{\mathrm{T}}\mathbf{f}_M \end{bmatrix}, \end{aligned} \tag{11}$$

where $\mathbf{g}_l^{\mathrm{T}}$ is the $l$th row of $\mathbf{G}$.

The following list of tasks enables $\mathbf{x} \odot [\mathbf{L}(\mathbf{C}^{1/2}\mathbf{f})]$ to be computed in a computationally efficient way. Each task is associated with an operator or subroutine identified with capitalized script letter.

1) Use the operator $\mathcal{W}_h$ to compute $\mathbf{G}\mathbf{f}_j$ for $j = 1, 2, \ldots, M$ at a cost of order $n_h L_h M$ operations. Note that the operation $\mathbf{G}\mathbf{f}_j$ returns the scalar value $\mathbf{g}_l^{\mathrm{T}}\mathbf{f}_j$ at the $l$th horizontal grid column for $l = 1, 2, \ldots, n_h$, ready for use in (11).
2) Use the operator $\mathcal{W}_z$ to compute the right-hand side of (11) for all horizontal grid points. This operation provides the ensemble weight for the $i$th ensemble perturbation at each grid point and costs order $n_h n_z M$ operations.
3) Use the operator $\mathcal{W}_{tr}$ to replicate the weights at each grid point for all variable types and all output times to obtain the full coarse resolution multivariate weight $n_T$ vector $(\mathbf{C}^{1/2}\mathbf{f})$ at a cost of order $n_q n_h n_\omega n_z = n_T$ operations.
4) Use the operator $\mathcal{L}$ to interpolate the weight vector from the coarse resolution grid to the high-resolution grid (using $\mathbf{L}$) and perform the element-wise product $\mathbf{x} \odot [\mathbf{L}(\mathbf{C}^{1/2}\mathbf{f})]$ at a cost of order $\tilde{n}_T$ operations.

In operator form, the algorithm is $\mathbf{x} \odot [\mathbf{L}(\mathbf{C}^{1/2}\mathbf{f})] \equiv \mathcal{L}\mathcal{W}_{rt}\mathcal{W}_z\mathcal{W}_h(\mathbf{f})$. Using parameter values from section 2, and assuming that the coarse resolution domain has about an order of magnitude fewer horizontal grid points and about the same order of magnitude of vertical grid points gives $u \sim O(10)$ and s $\sim O(1)$, its cost is order $n_h L_h M + n_h n_z M + n_T + \tilde{n}_T \sim O(10^{11} + 10^8 + 10^9 + 10^{10}) \sim O(10^{11})$ operations. This is five orders of magnitude fewer operations than the approach described in section 2.

It now remains to describe how similar approaches can be used to speed the adjoint operation defined by (4). Defining $\mathbf{y} = \mathbf{L}^{\mathrm{T}}(\mathbf{x} \odot \mathbf{h})$ and using (5) in (4) gives

$$\mathbf{p} \approx (\mathbf{C}^{1/2})^{\mathrm{T}}\mathbf{y} = [\mathbf{W}^v \vartriangle \mathbf{W}^h]^{\mathrm{T}}\mathbf{y}. \tag{12}$$

Section c of the appendix shows that

$$[\mathbf{W}^v \vartriangle \mathbf{W}^h]^{\mathrm{T}}\mathbf{y} = \begin{bmatrix} \mathbf{W}^{h\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_1^v) \\ \mathbf{W}^{h\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_2^v) \\ . \\ \mathbf{W}^{h\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_j^v) \\ . \\ \mathbf{W}^{h\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_M^v) \end{bmatrix}. \tag{13}$$

Now from the definition of $\mathbf{W}^h$ given in (6):

$$\begin{aligned} \mathbf{W}^{h\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_j^v) &= (\mathbf{R}^h\mathbf{G})^{\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_j^v) \\ &= \mathbf{G}^{\mathrm{T}}\mathbf{R}^{h\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_j^v) \\ &= \mathbf{G}^{\mathrm{T}}[\mathbf{I}, \mathbf{I}, \ldots, \mathbf{I}](\mathbf{y} \odot \mathbf{w}_j^v) \\ &= \mathbf{G}^{\mathrm{T}}\left[\sum_{q=1}^{n_q}\sum_{m=1}^{n_z}\sum_{r=1}^{n_\varpi}(\mathbf{y}_{mqr} \odot \mathbf{w}_{jmqr}^v)\right], \end{aligned} \tag{14}$$

where $\mathbf{y}_{mqr}$ and $\mathbf{w}_{jmqr}^v$ are each $n_h$ vectors describing horizontal fields of the $q$th model variable type on the $m$th vertical level at the $q$th time level. To see the connection between (14) and vertical functions $\mathbf{V}$, let $y_{mqrl} \odot w_{jmqrl}^v$ denote the $l$th horizontal element of the $n_h$ vector $\mathbf{y}_{mqr} \odot \mathbf{w}_{jmqr}^v$. From (8), $w_{jmqrl}^v = v_{jm}$. Consequently,

$$\begin{aligned} \sum_{q=1}^{n_q}\sum_{m=1}^{n_z}\sum_{r=1}^{n_\varpi}(y_{mqrl} \odot w_{jmqrl}^v) &= \sum_{q=1}^{n_q}\sum_{m=1}^{n_z}\sum_{r=1}^{n_\varpi}(y_{mqrl}v_{jm}) \\ &= \sum_{m=1}^{n_z} v_{jm}\left(\sum_{r=1}^{n_\varpi}\sum_{q=1}^{n_q} y_{mqrl}\right) \\ &= \mathbf{v}_j^{\mathrm{T}}\mathbf{d}_l = \mathbf{d}_l^{\mathrm{T}}\mathbf{v}_j, \end{aligned} \tag{15}$$

where the $n_z$ vector $\mathbf{d}_l^{\mathrm{T}}$ is given by

$$\mathbf{d}_l^{\mathrm{T}} = \left[\sum_{r=1}^{n_\varpi}\sum_{q=1}^{n_q} y_{1qrl}, \sum_{r=1}^{n_\varpi}\sum_{q=1}^{n_q} y_{2qrl}, \ldots, \sum_{r=1}^{n_\varpi}\sum_{q=1}^{n_q} y_{n_z qrl}\right]. \tag{16}$$

Using (16) to define the $n_z \times n_h$ matrix $\mathbf{D}$ gives

$$\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_l, \ldots, \mathbf{d}_{n_h}], \tag{17}$$

it follows that

$$\left[\sum_{q=1}^{n_q}\sum_{m=1}^{n_z}\sum_{r=1}^{n_\varpi}(\mathbf{y}_{mqr} \odot \mathbf{w}_{jmqr}^v)\right] = \mathbf{D}^{\mathrm{T}}\mathbf{v}_j, \tag{18}$$

and using (18) in (14) gives

$$\mathbf{W}_h^{\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_j^v) = \mathbf{G}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{v}_j. \tag{19}$$

We can then use (19) and (14) in (13) to obtain the following:

$$\mathbf{p}_i \approx [\mathbf{W}^v \triangle \mathbf{W}^h]^{\mathrm{T}}\mathbf{y} = [\mathbf{W}^v \triangle \mathbf{W}^h]^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}(\mathbf{x} \odot \mathbf{h}) = \begin{bmatrix} \mathbf{G}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{v}_1 \\ \mathbf{G}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{v}_2 \\ \cdot \\ \mathbf{G}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{v}_j \\ \cdot \\ \mathbf{G}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{v}_M \end{bmatrix}. \tag{20}$$

The above analysis allows us to see that the operation in (4) may be computed by applying the following list of operators.

1) Use the operator $\mathcal{L}^{\mathrm{T}}$ to compute $\mathbf{y} = \mathbf{L}^{\mathrm{T}}[\mathbf{x} \odot \mathbf{h}]$ at a cost of order $\tilde{n}_T$ operations.
2) Use the operator $\mathcal{W}_{tr}^{\mathrm{T}}$ to compute the arithmetic sum $\sum_{r=1}^{n_\varpi}\sum_{q=1}^{n_q} y_{mqrl}$ over variables *and* time [see (16)] at each model grid point defining the 3D grid and hence derive the $n_z \times n_h$ matrix $\mathbf{D}$ [see (17)] at a cost of order $n_h n_z n_q n_\varpi = n_T$ operations.
3) Use $\mathcal{W}_z^{\mathrm{T}}$ to compute $\mathbf{D}^{\mathrm{T}}\mathbf{v}_j$ for $j = 1, 2, \ldots, M$ with order $n_h n_z M$ operations.
4) Use $\mathcal{W}_h^{\mathrm{T}}$ to compute $\mathbf{G}^{\mathrm{T}}(\mathbf{D}^{\mathrm{T}}\mathbf{v}_j)$ for $j = 1, \ldots, M$ and build the vector $\mathbf{p}$ in accordance with (20) at a cost of order $ML_h n_h$ operations.

In operator form, the above algorithm may be summarized as $(\mathbf{C}^{1/2})^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}(\mathbf{x} \odot \mathbf{h}) \equiv \mathcal{W}_h^{\mathrm{T}}\mathcal{W}_z^{\mathrm{T}}\mathcal{W}_{rt}^{\mathrm{T}}\mathcal{L}^{\mathrm{T}}(\mathbf{x} \odot \mathbf{h})$. Unsurprisingly, the cost for these adjoint operations is the same order of magnitude as it was for the forward algorithm at $O(10^{11})$ operations and is again five orders of magnitude fewer operations than the approach described in section 2.

## 4. Conclusions

A fast algorithm for including nonadaptively localized ensemble covariances in variational algorithms has been derived and explained. This algorithm requires several orders of magnitude fewer computations than a simple-minded evaluation of the equations that were originally given in Lorenc (2003), Buehner (2005), and Buehner et al. (2010a,b). Elements of the fast algorithm were actually implemented by both Lorenc and Buehner in their work on localized ensemble covariances. However, their papers did not describe or explain the numerous nontrivial aspects of creating a fast algorithm. This paper serves to provide this information.

Some but not all of the techniques introduced here to speed nonadaptive ensemble covariance localization can also be applied to adaptive localization methods such as those discussed in Bishop and Hodyss (2009b). These include (i) defining the adaptive localization correlation matrix on a coarser resolution grid, and (ii) summing over variable type if the same adaptive localization functions are being applied to all variable types.

*Acknowledgments.* The authors wish to acknowledge the contributions of Ricardo Todling whose comments, as a reviewer, significantly improved the paper. Among other things, Ricardo improved the conciseness of the proof of the square root theorem. The authors also acknowledge the helpful criticism of the Editor, Herschel Mitchell, and the insightful suggestions of an anonymous reviewer. CHB expresses his gratitude to the "Centre for Australia Weather and Climate RESEARCH—a collaboration between the Bureau of Meteorology and CSIRO" for their support of a visit to the Centre during which this paper was conceived. CHB and DH gratefully acknowledge financial support from ONR Project Element 0602435N, Project BE-435-003, and ONR Grant N0001407WX30012.

## APPENDIX

### Definitions and Proofs

*a. Definition of the modulation product and square root theorem*

It is convenient to define a modulation product $\mathbf{u} \triangle \mathbf{W}$ of an $n$-vector $\mathbf{u}$ with a $n \times L$ matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L]$ to be the $n \times L$ matrix that lists the modulations (or element-wise products) of each column of $\mathbf{W}$ by $\mathbf{u}$; in other words:

$$\mathbf{u} \triangle \mathbf{W} = [\mathbf{u} \odot \mathbf{w}_1, \mathbf{u} \odot \mathbf{w}_2, \ldots, \mathbf{u} \odot \mathbf{w}_L] = \mathrm{diag}(\mathbf{u})\mathbf{W}. \tag{A1}$$

We define the modulation product of the $n \times M$ matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$ with an $n \times L$ matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L]$ to be the $n \times (LM)$ matrix $\mathbf{U} \triangle \mathbf{W}$ such that it lists all possible modulations of the columns of $\mathbf{W}$ by the columns of $\mathbf{U}$; in other words,

$$\begin{aligned}\mathbf{U} \triangle \mathbf{W} &= [\mathbf{u}_1 \triangle \mathbf{W}, \mathbf{u}_2 \triangle \mathbf{W}, \ldots, \mathbf{u}_M \triangle \mathbf{W}] \\ &= [\mathrm{diag}(\mathbf{u}_1)\mathbf{W}, \mathrm{diag}(\mathbf{u}_2)\mathbf{W}, \ldots, \mathrm{diag}(\mathbf{u}_M)\mathbf{W}].\end{aligned} \tag{A2}$$

It follows that for a general $LM$ vector $\mathbf{f}$,

$$[\mathbf{U} \triangle \mathbf{W}]\mathbf{f} = \sum_{m=1}^{M}\sum_{l=1}^{L}(\mathbf{u}_m \odot \mathbf{w}_l)f_{lm}, \tag{A3}$$

where $f_{lm}$ represents elements of the $LM$ vector $\mathbf{f}$ partitioned as

$$\begin{aligned}\mathbf{f}^{\mathrm{T}} &= [\mathbf{f}_1^{\mathrm{T}}, \mathbf{f}_2^{\mathrm{T}}, \ldots, \mathbf{f}_M^{\mathrm{T}}] \\ &= [f_{11}, f_{21}, \ldots, f_{L1}, f_{12}, f_{22}, \ldots, f_{L2}, \ldots, f_{LM}].\end{aligned}$$

### SQUAREROOT THEOREM FOR ELEMENTWISE PRODUCTS

If the matrices $\mathbf{U}$ and $\mathbf{V}$ are $n \times M$ matrices and $\mathbf{W}$ and $\mathbf{X}$ are $n \times L$ matrices such that

$$\mathbf{A} = \mathbf{U}\mathbf{V}^{\mathrm{T}} \quad \text{and} \quad \mathbf{B} = \mathbf{W}\mathbf{X}^{\mathrm{T}} \tag{A4}$$

then

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{U} \triangle \mathbf{W}][\mathbf{V} \triangle \mathbf{X}]^{\mathrm{T}}. \tag{A5}$$

*Proof:* The proof is by construction. First, note that

$$[\mathbf{U} \triangle \mathbf{W}][\mathbf{V} \triangle \mathbf{X}]^{\mathrm{T}} = \sum_{m=1}^{M}\sum_{l=1}^{L} (\mathbf{u}_m \odot \mathbf{w}_l)(\mathbf{v}_m \odot \mathbf{x}_l)^{\mathrm{T}}, \tag{A6}$$

where $\mathbf{u}_m$ and $\mathbf{v}_m$ represent the $m$th columns of $\mathbf{U}$ and $\mathbf{V}$, respectively; and $\mathbf{w}_l$ and $\mathbf{x}_l$ represent the $l$th columns of $\mathbf{W}$ and $\mathbf{X}$, respectively. Second, note that if the subscripted variables $\{[\mathbf{U} \triangle \mathbf{W}][\mathbf{V} \triangle \mathbf{X}]^{\mathrm{T}}\}_{ij}$ and $u_{ij}$, $v_{ij}$, $w_{ij}$ and $x_{ij}$ denote the elements on the $i$th row and $j$th column of $[\mathbf{U} \triangle \mathbf{W}][\mathbf{V} \triangle \mathbf{X}]^{\mathrm{T}}$, $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{X}$, respectively, then (A5) implies that

$$\begin{aligned}
\{[\mathbf{U} \triangle \mathbf{W}][\mathbf{V} \triangle \mathbf{X}]^{\mathrm{T}}\}_{ij} &= \sum_{m=1}^{M}\sum_{l=1}^{L} (u_{im}w_{il})(v_{jm}x_{jl}) \\
&= \sum_{m=1}^{M}\sum_{l=1}^{L} (u_{im}v_{jm})(w_{il}x_{jl}) = \sum_{m=1}^{M} (u_{im}v_{jm}) \sum_{l=1}^{L} (w_{il}x_{jl}) \\
&= (\mathbf{U}\mathbf{V}^{\mathrm{T}})_{ij}(\mathbf{W}\mathbf{X}^{\mathrm{T}})_{ij} \\
&= (\mathbf{A} \odot \mathbf{B})_{ij}
\end{aligned} \tag{A7}$$

and, hence, $[\mathbf{U} \triangle \mathbf{W}][\mathbf{V} \triangle \mathbf{X}]^{\mathrm{T}} = \mathbf{A} \odot \mathbf{B}$ as was required.

The square root theorem for localized ensemble covariances is a corollary of the theorem above in the special case when the Schur product involves symmetric, semipositive definite matrices. To be specific, let $\mathbf{Z} = 1/\sqrt{K-1}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K]$ so that

$$\mathbf{P}_K^f = \mathbf{Z}\mathbf{Z}^{\mathrm{T}} \quad \text{and} \quad \tilde{\mathbf{C}} = \tilde{\mathbf{C}}^{1/2}(\tilde{\mathbf{C}}^{1/2})^{\mathrm{T}} \tag{A8}$$

then (A5) implies that

$$\begin{aligned}
&\mathbf{P}_K^f \odot \tilde{\mathbf{C}} = [\mathbf{Z} \triangle \tilde{\mathbf{C}}^{1/2}][\mathbf{Z} \triangle \tilde{\mathbf{C}}^{1/2}]^{\mathrm{T}}, \quad \text{where} \\
&\mathbf{Z} \triangle \tilde{\mathbf{C}}^{1/2} = \frac{1}{\sqrt{K-1}} \\
&\quad \times [\mathrm{diag}(\mathbf{x}_1)\tilde{\mathbf{C}}^{1/2}, \mathrm{diag}(\mathbf{x}_2)\tilde{\mathbf{C}}^{1/2}, \ldots, \mathrm{diag}(\mathbf{x}_K)\tilde{\mathbf{C}}^{1/2}].
\end{aligned} \tag{A9}$$

### b. *Proof that* $[\mathbf{W}^v \triangle \mathbf{W}^h]\mathbf{f} = \sum_{j=1}^{M} \mathbf{w}_j^v \odot [\mathbf{W}^h\mathbf{f}_j]$, *where* $\mathbf{f}_j^{\mathrm{T}}$ *is a* $L_h$ *vector such that* $\mathbf{f}^{\mathrm{T}} = [\mathbf{f}_1^{\mathrm{T}}, \mathbf{f}_2^{\mathrm{T}}, \ldots, \mathbf{f}_j^{\mathrm{T}}, \ldots, \mathbf{f}_M^{\mathrm{T}}]$ *and where* $\mathbf{W}^v$ *and* $\mathbf{W}^h$ *have* $M$ *and* $L_h$ *columns, respectively.*

Let $\mathbf{w}_k^h$ denote the $k$th column of $\mathbf{W}^h$ and $f_{jk}$ be the *kth* element of $\mathbf{f}_j$ then

$$\begin{aligned}
\sum_{j=1}^{M} \mathbf{w}_j^v \odot [\mathbf{W}^h\mathbf{f}_j] &= \sum_{j=1}^{M} \mathbf{w}_j^v \odot \sum_{k=1}^{L_h} \mathbf{w}_k^h f_{jk} \\
&= \sum_{k=1}^{L_h}\sum_{j=1}^{M} (\mathbf{w}_j^v \odot \mathbf{w}_k^h) f_{jk} \\
&= (\mathbf{W}^v \triangle \mathbf{W}^h)\mathbf{f} \text{ (as noted in A3)}
\end{aligned} \tag{A10}$$

as was required.

### c. *Proof of (13)*

To prove (13) note that each column of $\mathbf{W}^v \triangle \mathbf{W}^h$ can be expressed in the form $\mathbf{w}_j^v \odot \mathbf{w}_k^h$—the element-wise product of the $j$th column of $\mathbf{W}^v$ and the $k$th column of $\mathbf{W}^h$. Hence, each column of $\mathbf{W}^v \triangle \mathbf{W}^h$ is associated with two indices $j$ and $k$. Hence, the row element of $(\mathbf{W}^v \triangle \mathbf{W}^h)^{\mathrm{T}}\mathbf{y}$ corresponding to $\mathbf{w}_j^v \odot \mathbf{w}_k^h$ is given by

$$(\mathbf{w}_j^v \odot \mathbf{w}_k^h)^{\mathrm{T}}\mathbf{y} = \sum_{r=1}^{n_T} w_{jr}^v w_{kr}^h y_r = \sum_{r=1}^{n_T} w_{kr}^h (y_r w_{jr}^v) = \mathbf{w}_k^{h\mathrm{T}}(\mathbf{y} \odot \mathbf{w}_j^v) \tag{A11}$$

Equation (13) follows from (A11). [In (A11) "$r$" does not refer to a variable type as it does in the text; it is just an index.]

## REFERENCES

Bishop, C. H., and D. Hodyss, 2007: Flow adaptive moderation of spurious ensemble correlations and its use in ensemble based data assimilation. *Quart. J. Roy. Meteor. Soc.,* **133,** 2029–2044.

——, and ——, 2009a: Ensemble covariances adaptively localized with ECO-RAP. Part 1: Tests on simple error models. *Tellus,* **61,** 84–96.

——, and ——, 2009b: Ensemble covariances adaptively localized with ECO-RAP. Part 2: A strategy for the atmosphere. *Tellus,* **61,** 97–111.

Buehner, M., 2005: Ensemble-derived stationary and flow-dependent background error covariances: Evaluation in a quasi-operational NWP setting. *Quart. J. Roy. Meteor. Soc.,* **131,** 1013–1043.

——, P. L. Houtekamer, C. Charette, H. L. Mitchell, and B. He, 2010a: Intercomparison of variational data assimilation and the ensemble Kalman filter for global deterministic NWP. Part I: Description and single-observation experiments. *Mon. Wea. Rev.,* **138,** 1550–1566.

——, ——, ——, ——, and ——, 2010b: Intercomparison of variational data assimilation and the ensemble Kalman filter for global deterministic NWP. Part II: One-month experiments with real observations. *Mon. Wea. Rev.,* **138,** 1567–1586.

Dee, D. P., 1991: Simplification of the Kalman filter for meteorological data assimilation. *Quart. J. Roy. Meteor. Soc.,* **117,** 365–384.

El Akkraoui, A., P. Gauthier, S. Pellerin, and S. Buis, 2008: Intercomparison of the primal and dual formulations of variational data assimilation. *Quart. J. Roy. Meteor. Soc.,* **134,** 1015–1025.

Hunt, B. R., E. J. Kostelich, and I. Szunyogh, 2007: Efficient data assimilation for a spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D,* **230,** 112–126.

Kepert, J. D., 2009: Covariance localisation and balance in an Ensemble Kalman Filter. *Quart. J. Roy. Meteor. Soc.,* **135,** 1157–1176.

Liu, C., Q. Xiao, and B. Wang, 2009: An ensemble-based four-dimensional variational data assimilation scheme. Part II: Observing system simulation experiments with Advanced Research WRF (ARW). *Mon. Wea. Rev.,* **137,** 1687–1704.

Lorenc, A. C., 2003: The potential of the ensemble Kalman filter for NWP—A comparison with 4D-VAR. *Quart. J. Roy. Meteor. Soc.,* **129,** 3183–3203.

Miyoshi, T., and S. Yamane, 2007: Local ensemble transform Kalman filtering with an AGCM at a T159/L48 resolution. *Mon. Wea. Rev.,* **135,** 3841–3861.

Parrish, D. F., and J. C. Derber, 1992: The National Meteorological Center's spectral statistical-interpolation analysis system. *Mon. Wea. Rev.,* **120,** 1747–1763.

Szunyogh, I., E. J. Kostelich, G. Gyarmati, E. Kalnay, B. R. Hunt, E. Ott, E. Satterfield, and J. A. York, 2008: A local ensemble transform Kalman filter data assimilation system for the NCEP global model. *Tellus,* **60A,** 113–130.

Wang, X., C. Snyder, and T. M. Hamill, 2007: On the theoretical equivalence of differently proposed ensemble–3DVAR hybrid analysis schemes. *Mon. Wea. Rev.,* **135,** 222–227.

Yang, S. C., E. Kalnay, B. R. Hunt, and N. E. Bowler, 2009: Weight interpolation for efficient data assimilation with the Local Ensemble Transform Kalman Filter. *Quart. J. Roy. Meteor. Soc.,* **135,** 251–262.