The Multigrid Beta Function Approach for Modeling of Background Error Covariance in the Real-Time Mesoscale Analysis (RTMA)

R. JAMES PURSER,[a] MIODRAG RANCIC,[a] AND MANUEL S. F. V. DE PONDECA[a]

[a] *IMSG at NOAA/NCEP/EMC, College Park, Maryland*

(Manuscript received 15 December 2020, in final form 7 December 2021)

ABSTRACT: We describe a method for the efficient generation of the covariance operators of a variational data assimilation scheme, which is suited to implementation on a massively parallel computer. The elementary components of this scheme are what we call "beta filters," since they are based on the same spatial profiles possessed by the symmetric beta distributions of probability theory. These approximately Gaussian (bell-shaped) polynomials blend smoothly to zero at the ends of finite intervals, which makes them better suited to parallelization than the present quasi-Gaussian "recursive filters" used in operations at NCEP. These basic elements are further combined at a hierarchy of different spatial scales into an overall multigrid structure formulated to preserve the necessary self-adjoint attribute possessed by any valid covariance operator. This paper describes the underlying idea of the beta filter and discusses how generalized Helmholtz operators can be enlisted to weight the elementary contributions additively in such a way that the covariance operators may exhibit realistic negative sidelobes, which are not easily obtained through the recursive filter paradigm. The main focus of the paper is on the basic logistics of the multigrid structure by which more general covariance forms are synthesized from the basic quasi-Gaussian elements. We describe several ideas on how best to organize computation, which led us to a generalization of this structure that made it practical so that it can efficiently perform with any rectangular arrangement of processing elements. Some simple idealized examples of the applications of these ideas are given.

KEYWORDS: Filtering techniques; Variational analysis; Numerical analysis/modeling; Multigrid models

1. Introduction

We present a technique for generating the covariance operators of an optimal data assimilation (DA) scheme that overcomes several limitations of the present operational recursive-filter-based technique (Wu et al. 2002; De Pondeca et al. 2011) used at the National Centers for Environmental Prediction (NCEP). The variational assimilation of atmospheric data requires that the error in the forecast background field (generally supplied by a previous short-term forecast) be modeled. An extensive review of methods used for modeling of the background error covariance is found for example in Fisher (2003), and (Bannister 2008a,b). At NCEP, the basis for the modulation of the amplitude of the background error variance has been for years the so-called NMC method of Parrish and Derber (1992) that uses the differences of a pair of lagged forecasts. The covariance, being a function of the dynamical field errors at two spatial locations, serves as the kernel of a filtering operator and it is in this specialized role that it appears in the formulation of the cost-function minimization process at the core of any implementation of the variational principle that defines the intent of the DA process. This filter is generally considered to be spatially smooth, though not necessarily spatially homogeneous, or even horizontally isotropic. Since the meteorological fields, and therefore their increments, are known to exhibit multivariate regularities associated with dynamical balance, we demand that the covariance filter also exhibits, as weak or strong constraints, these same multivariate regularities. Another expected property of the covariance of background error is its eventual diminishing value with increasing spatial separation. The analogy between a covariance and the solution of a diffusion operator was exploited in Derber and Rosati (1989), and has been developed into more versatile and computationally efficient generalizations by Weaver and Courtier (2001), Mirouze and Weaver (2010), Weaver and Mirouze (2013), and Guillet et al. (2019), where the iterated diffusion with an anisotropic and inhomogeneous diffusivity tensor, being based on a local operator, is particularly well-adapted to ocean model assimilation where lateral boundaries of each quasi-horizontal layer can be quite irregular.

In NCEP's Gridpoint Statistical Interpolation (GSI) the multivariate features mentioned above have been accommodated by projecting the idealized dynamically balanced and unbalanced contributions to the analysis increments into separate scalar fields, which can then be treated as if they are statistically independent. In this way, the linearized aspects of dynamical balance become an implicit characteristic of the analysis. These quasi-independent combinations are then smoothed using a filter of the recursive kind. The recursive filter (Hayden and Purser 1995; Wu et al. 2002; Purser et al. 2003a) is a sequence of one-sided line filters applied across large regular-gridded domains, in which the sequence involves both directions of travel along each line orientation, and along a sufficient set of line orientations to enable the net result of such filtering to fill out the full dimensionality of the given grid.

a. Limitations associated with the recursive filter

The recursive filter method strives to emulate the Gaussian profile in each opposing pair of back-and-forth applications of each basic recursive filtering operator along each line. When

Corresponding author: Miodrag Rancic, miodrag.rancic@noaa.gov

DOI: 10.1175/MWR-D-20-0405.1

this pair, applied along each of the generalized grid lines threading through the lattice in a common parallel direction, is further compounded sequentially with other correspondingly back-and-forth sweeps along transverse families of lines, the resulting response retains the quasi-Gaussian character in the higher-dimensionality of the grid. On a serial computer architecture, this gives rise to an exceptionally efficient numerical algorithm for generating quasi-Gaussian smoothing responses (much more efficient than could be done by a direct point-by-point evaluation of the filter response whenever the characteristic scale of this response is significantly larger than the grid scale) and it allows variations in the shape of the local quasi-Gaussian response to occur over the spatial extent of the domain that would not be easily achieved otherwise, for example, by using spectral means. Moreover, the quasi-Gaussian shapes need not be restricted to kernels of a locally isotropic shape in the horizontal directions; for the Real-Time Mesoscale Analysis (RTMA; Carley et al. 2020, e.g.,) it is especially important that the quasi-Gaussian contribution to the covariance operator be allowed to exhibit significantly anisotropic form, with stretching of the shape along the contours of the terrain, for example. In the general formulation of the recursive filter method, a number (though typically not more than three) of such quasi-Gaussian contributions, computed independently, and at different characteristic scales, are combined additively, to allow covariances whose spatial profiles are of non-Gaussian fat-tailed form, which now comprise a more realistic broad range of spatial scales.

The recursive filter algorithms have proven to be a successful approach to addressing the challenging problem of numerically simulating background error covariance in a computationally efficient manner on machines with small to moderate degrees of parallelism. However, being intrinsically sequential and by having infinite impulse-response functions (e.g., Hamming 1989), the recursive procedures are not optimally adapted to the modern, massively parallel architectures of the new generation of computers. Neither is the recursive filter formulation able easily to accommodate the more generic multivariate correlations that must clearly be present between dynamically "balanced" and "unbalanced" components in an evolving adaptive scenario, since these components are presently treated, erroneously, as statistically independent.

b. The beta filter and its advantages

The new massively parallel-computer architectures therefore require us to review the basic procedure of covariance generation and find a reformulation to overcome the inherent limitations of the recursive filters. It is to address these challenges that we have formulated the multigrid beta filter covariance method described in this paper. The "support" of a filter denotes the portion of its range where the response to an impulse is nonzero. Thus, the Gaussian function itself has *infinite* support—its calculation also involves evaluating exponential functions, so it is doubly handicapped as a practical choice for a covariance component. Recursive filters involve only simple computations, but unfortunately they also have the handicap of infinite support. The filters at the core of our new scheme remain quasi-Gaussian but, instead of being recursive, and therefore burdened with effectively infinite support, the new filters are based on finite-impulse-response "beta distributions" applied as explicit filters. In this respect they are like the explicit compact-support covariance functions proposed by Gaspari and Cohn (1999) who, by modeling families of covariances as piece-wise rational analytic functions, were successful in constructing useful covariances, allowing general point-to-point evaluations, exhibiting realistic departures from the thin-tailed Gaussian. In our case, however, we have the multigrid machinery at our disposal to take care of the shaping of the final covariance profiles from quasi-Gaussian ingredients (together with their second derivatives in a slightly more sophisticated "Helmholtz-weighting" extension that we shall briefly touch upon), and we have the advantage of a regular grid to operate on. Therefore, we need only emulate, in the simplest algebraic way, the compact-support gridded approximations to Gaussian in our basic filters. These filters can consequently be of an algebraically and computationally simpler polynomial form.

A beta distribution, in probability and statistics, is one with support of finite width, and which, by a judicious choice of the pair of shape-controlling parameters, can be made a filter of a bell-shaped profile. The "beta" name arises from the fact that, in order to ensure that the integral of a beta distribution is one over the standardized centered interval of unit width, the normalizing constant is exactly the Euler beta function of the pair of filter parameters (Abramowitz and Stegun 1972). In this sense, it might be legitimate to regard the distribution as the "incomplete Euler beta function." When, as we always assume, the pair of standard parameters of the beta function are equal, then one less than this shared standard parameter will be taken to be our own beta filter parameter p, and then the profile is symmetrical within its interval of support. When the parameter p is not too small the shape resembles a Gaussian (the resemblance improving as the parameter increases). When the parameter is an integer, the functional form is simply a polynomial of degree $2p$, which is therefore quite easy and cheap to apply; thus, we always assume the shape parameter to be some positive integer. Since the beta function line filter mimics the Gaussian, it can be used in place of the recursive filter as a basic component of the same compounded filters that the recursive filters led to. Although, by itself, iterating a beta line filter is computationally more expensive than the recursive filter, its advantage of having a finite impulse response enables the domain to be dissected into overlapping regions to which we can apply efficient parallelization, which becomes the decisive factor in determining the speed of the algorithm overall.

The beta distribution with common integer parameter, being symmetric, is clearly an even polynomial evaluated about the center of its support interval, so it can therefore be immediately generalized to a radially symmetric distribution in any number of higher dimensions. Since the parameter is an integer, the response function remains a polynomial, in the more general multivariate sense, in the Cartesian spatial coordinates. A linear transformation of these coordinates, by application of a shape-controlling symmetric "aspect tensor,"

will result in another quasi-Gaussian response, but of the anisotropic kind. There is a spectacularly high computational cost of applying an explicit multidimensional filter whose characteristic scale significantly exceeds the grid scale, but for a smoothing component, which comprises a relatively coarse scale, this cost can be significantly mitigated by engaging the second characteristic feature of the new approach—the multigrid feature.

c. Multigrid formulation

In a multigrid method (Brandt 1977, 1997; Hackbusch 2013), whether it is to solve an elliptic problem (its traditional and most familiar application) or an integral equation (such as the present filtering problem), the portions of the problem that can be dealt with at a coarse-scale grid are dealt with on a grid of commensurate spacing; portions of the increment solution that require a fine scale are efficiently computed on a grid of commensurate spacing. Additive contributions of the covariances that have small-scale details and are therefore composed of quasi-Gaussians of small scale, are also efficiently computed because, although they are calculated on a finer grid, the smaller physical scale of the elliptical or ellipsoidal footprints of these quasi-Gaussians still involve about the same number of grid points in each impulse response function. In this way, the combination of a multigrid computational structure with explicit quasi-Gaussian filters, is able to overcome the limitations that accompany single grid implementations of an explicit smoothing filter when the increments possess some characteristic scales of components that are large compared to the grid and some that are comparable with the grid spacing.

Although this is a topic whose detailed presentation will be reserved for future publication, we mention that the ability of a finite-support filter to allow domain decomposition also makes it computationally easier to introduce nontrivial multivariate couplings between pairs of the analysis variables. In the simplest instances, this can be done by generalizing the multigrid "scale-weights" at each grid generation. Instead of restricting these weight fields to being scalars operating separately on each of the analysis variables, they can be made matrix weight fields that couple the different variables. As we indicate briefly in section 3a, we can formally generalize these weights further to be differential operators, at the highest generation at least, which provides a practical way of producing covariances possessing negative side lobes, if these are desired.

d. Outline of the paper

Section 2 presents in more details the beta filter. Our approach to the multigrid (MG hereafter) paradigm for application of the beta filter is discussed in the following section 3, where we also show a few preliminary examples of performance derived in stand-alone test cases. An efficient organization of calculation, that enables generalization and practical application of the MG filtering is described in section 4. Several examples of method performance, derived in a stand-alone version, where the filter is tested outside of the data assimilation framework, and a preliminary implementation in the GSI, are shown in section 5. The paper concludes with discussion, summary and a brief outline of plans contained in section 6. More technical details describing options for further speedup, and a strategy for application of the MG beta filter on the cubed sphere are reserved for appendices.

2. The beta filter

a. Background error covariance

The background error covariance ($\mathbf{B}$) is an operator whose inverse defines the effective weight attributed to the background field in the variational DA, just as the inverse of the observation error covariances defines the effective weights applied to those observation values. It is impractical to hold an explicit representation of $\mathbf{B}$ when this covariance is spatially inhomogeneous, and equally impractical to attempt to apply direct matrix methods in the solution of a variational problem. Instead, we break down the approximate representation of the matrix into small manageable filters and apply iterative methods, such as a conjugate gradient method (e.g., Gill et al. 1981), which help to approach the desired solution in about 100–150 iterative steps. Each step requires only the work whose dominant part is equivalent to multiplying vectors by $\mathbf{B}$, or by the simpler factors into which it can be broken. This process does require some conditions to be upheld on the structure of the approximation for the operator, such as strict self-adjointness (essentially equivalent to matrix symmetry) and nonnegativity (in terms of the signs of the eigenvalues of the approximation to $\mathbf{B}$).

Thus, in modeling of $\mathbf{B}$, one only needs to construct a self-adjoint operator (in matrix terms, $\mathbf{B} = \mathbf{C}\mathbf{C}^{\mathrm{T}}$) such that when acting in the expression:

$$\mathbf{B}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d},$$

it can produce a smooth, bell-shaped response. Terms of a similar kind occur in the conjugate gradient algorithm. Here, we follow the traditional notation from Ide et al. (1997), where

$\delta\mathbf{x} = \mathbf{x}^a - \mathbf{x}^b$ is the analysis increment,

$\mathbf{d} = \mathbf{y} - \mathbf{H}(\mathbf{x}^b)$ is the innovation,

$\mathbf{y}$ is the observation vector,

$\mathbf{B}$ is the background error covariance matrix,

$\mathbf{R}$ is the observation error covariance matrix,

$\mathbf{H}$ is the linearized observation (forward) operator.

b. Recursive filters

Recursive filters (e.g., Wu et al. 2002; Purser et al. 2003a,b), which have, for years, been used in the GSI, represent an efficient and an exceptionally good approximation to the Gaussian (see Fig. 1), allowing inhomogeneity and anisotropy of analyzed fields to be accounted for.

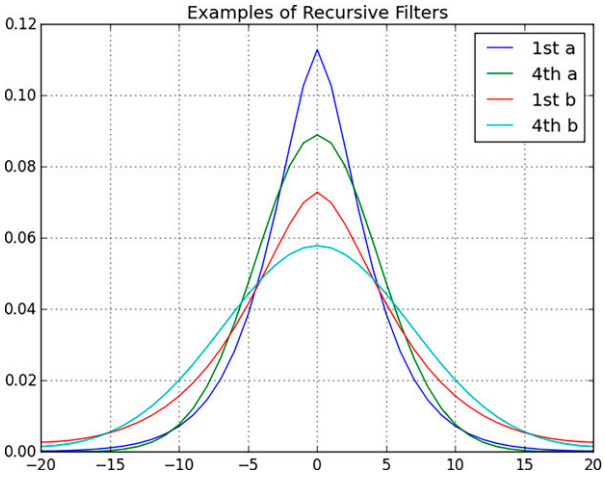

FIG. 1. Examples of recursive filters applied twice to a unit point impulse: first- and fourth-order filters are shown using a relatively narrow filtering scale of 3.2 grid units for each filter (labeled "a") and a relatively broad filtering scale of 4.9 units in each application (labeled "b"). The horizontal axis is in grid units. The fourth-order filter uses a higher degree of approximation to the Gaussian but requires a correspondingly larger number of floating-point operations to achieve this.

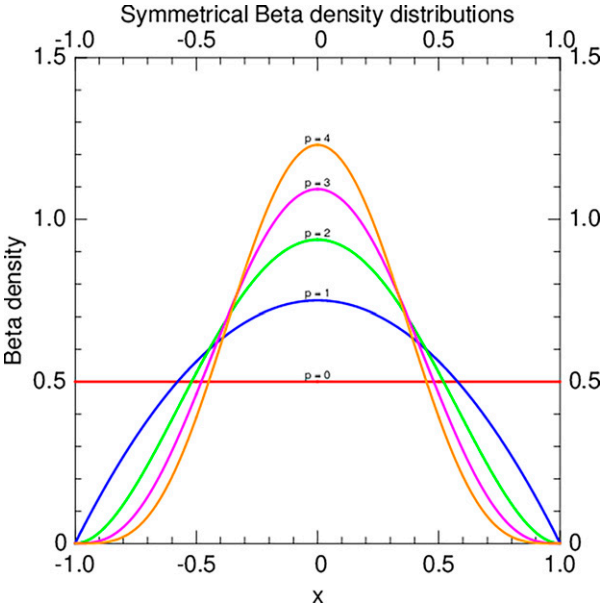

FIG. 2. Example of a homogeneous beta filter with compact support.

On the downside, recursive filters are inherently sequential operators, making them exceedingly difficult to successfully parallelize, and they have an infinite support. In addition, without the benefits of a multigrid structure, the filters, as presently formulated:

- cannot describe covariances across various scales,
- neither can they consider cross correlation,
- nor do they provide negative lobes, which realistic covariances in some situations may possess.

c. Beta filters

Our alternative to recursive filters is based on the beta distribution functions. We shall use the semicolon separator for elements of a column vector, such as the 2D displacement vector of position, $\mathbf{r} = \mathbf{x}_j - \mathbf{x}_i \equiv (x_j - x_i; y_j - y_i)$. Then the radial beta filter response is defined by

$$\beta(\mathbf{r}) \propto (1 - \rho)^p, \quad \rho \leq 1. \tag{1}$$

Here p, the already mentioned beta filter parameter, is a small positive integer. A larger p implies a more Gaussian shape, but also a narrower one. In the isotropic case, ρ can be expressed as

$$\rho = \frac{\mathbf{r}^{\mathrm{T}}\mathbf{r}}{s^2(2p + 4)}, \tag{2}$$

where s is a radial scale [the now finite disk of support has a slightly larger radius of support $s(2p + 4)^{1/2}$]. Such quasi-Gaussian functions are illustrated, for the first few integers p, in Fig. 2.

The second moment of the normalized beta distribution, which has units of the square of distance, can be shown to be equal to s^2 in each direction in the isotropic case, but is a 2×2 symmetric positive-definite tensor more generally. It is then informally referred to as the quasi-Gaussian's "aspect tensor" $\mathbf{A}$ and the formula for ρ, modified to accommodate it, becomes

$$\rho = \frac{\mathbf{r}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{r}}{(2p + 4)}. \tag{3}$$

The 3D (three-dimensional) radial beta filter has a similar formulation, except that the scaling factor $(2p + 4)$ relating the footprint to the aspect tensor is replaced by $(2p + 5)$.

The aspect tensor controls the shape of covariances. In the 2D case it is possible for the symmetric aspect tensor to be written in the following form:

$$\mathbf{A} = A_0 \begin{bmatrix} \cosh(\phi) + \cos(\theta)\sinh(\phi), & \sin(\theta)\sinh(\phi) \\ \sin(\theta)\sinh(\phi), & \cosh(\phi) - \cos(\theta)\sinh(\phi) \end{bmatrix}. \tag{4}$$

In this formula, the quantity A_0 is essentially the characteristic area of the response; the degree of anisotropy is controlled by the magnitude of the parameter ϕ, and if this parameter is positive, the major principal axis is oriented at an angle of $2/\theta$ to the x axis (as we show in Fig. 11).

d. Direct and adjoint beta filtering of scalar fields

Since $\mathbf{A}$ generally varies spatially, let $\beta_{(i)}$ specify that the aspect tensor involved is the one appropriate to the location of grid point $\mathbf{x}_i$. Then let $\beta^*_{ij} = \beta_{(i)}(\mathbf{x}_j - \mathbf{x}_i)$. Then the direct beta filtering of a scalar field $\psi_i \equiv \psi(\mathbf{x}_i)$, where the grid point $\mathbf{x}_i$ is assigned a measure τ_i, is done according to

$$\overline{\psi}_i = \sum_j F_{ij}\psi_j, \tag{5}$$

where

$$F_{ij} = \frac{\beta^*_{ij}\tau_j}{\sum_k \beta^*_{ik}\tau_k}. \tag{6}$$

The action of F defined as in (6) is clearly that of a "smoothing" operator. Then the adjoint filtering of another field $\chi_i \equiv \chi(\mathbf{x}_i)$ consistent with the inner product norm of χ and ψ being that associated with the assigned measure τ,

$$(\chi, \overline{\psi}) \equiv \sum_i \chi_i \overline{\psi}_i \tau_i \equiv \sum_j \overline{\chi}_j \psi_j \tau_j = (\overline{\chi}, \psi), \tag{7}$$

is

$$\overline{\chi}_j = F^\dagger_{ji}\chi_i, \tag{8}$$

where the formal definition of the coefficients of this adjoint beta filter is therefore,

$$F^\dagger_{ji} = F_{ij}\frac{\tau_i}{\tau_j}. \tag{9}$$

The action of $\mathbf{F}^\dagger$ defined as in (8) is clearly that of a "conserving" operator for the integrated "substance" that the operator acts upon. We shall find it convenient later to maintain the "smoothing" and "conserving" attributes in the direct and adjoint operators of interpolation between grids when we deal with the multigrid structure. The measure τ is usually the geographical area of the grid cell in two dimensions, and in three dimensions it also includes a factor for the vertical spacing of grid points in some suitable choice of coordinates. In practice, it is possible to avoid including the measure terms in the filtering computations by predividing the field to be filtered by the τ value at each grid point, and then finally multiplying the adjoint-filtered results at each grid point by this same τ measure. Of course, associated with the basic beta filtering operation is some form of amplitude weighting needed to prescribe each filter's relative contribution to the final $\mathbf{B}$. This weight operator, denoted $\mathbf{W}$, is just a scalar positive number, and therefore expressible as a square, $\mathbf{W} = \mathbf{G}^2$, in the simplest case. This allows a single beta filter to be weighted in such a way that it factors neatly into square root factors:

$$(\mathbf{FG})(\mathbf{GF}^\dagger) \equiv (\mathbf{FG})(\mathbf{FG})^\dagger. \tag{10}$$

But, as we shall show in the next section, $\mathbf{W}$ can also be a differential operator of the generalized Helmholtz type, $\mathbf{W} = \mathbf{GG}^\dagger$, where $\mathbf{G}$ is now itself a vectorial differential operator of the first order. The next section explains this in greater detail and shows how the notation is generalized to allow the "square root" factoring form of $\mathbf{B}$ to be extended to the multigrid case.

3. Multigrid approach to the application of the beta filter

Examples of the application of the multigrid method in DA are found in: Kang et al. (2014), as a method for minimization of the cost function; Li et al. (2008) for assimilation of ocean temperature; Li et al. (2010) for two-dimensional Doppler radar radial velocity assimilation; Zhang and Tian (2018), Zhang et al. (2020) for a nonlinear least squares four-dimensional variational data assimilation, etc. In our case, the beta filter is used at a hierarchy of different scales, combined into a parallel multigrid scheme to achieve a larger coverage and potentially a more versatile synthesis of anisotropic covariances, allowing a greater control over the shape. In this section, we will first describe the core of our multigrid approach, and in the following discuss a technique that we developed to ensure the general applicability of the code. By that, we refer to the capability of the code to perform optimally under various grid decompositions across a multiprocessing computing platform.

a. Self-adjoint operators

Recall that $\mathbf{B}$ needs to be a self-adjoint operator which converts an impulse into a weighted superposition of quasi-Gaussian shapes, or of derivatives of these shapes. The construction of $\mathbf{B}$ achieving nonnegativity and self-adjointness in a factorization, $\mathbf{B} = \mathbf{B}^\dagger = \mathbf{CC}^\dagger$, does not require operator $\mathbf{C}$, as a matrix, to be "square." The weighting operator in the superposition may be a positive scalar function of position, sandwiched between the conserving and smoothing beta filter stages. This construction obviously remains compatible with the desired $\mathbf{B} = \mathbf{CC}^\dagger$ factoring.

A more general scale-weight function, in which this positive scalar function is replaced by a self-adjoint "Helmholtz" elliptic operator, is also, but less obviously, compatible with this formal factoring, as we noted in the previous section. We will set out the main ideas here, although a detailed examination of this method will await a future paper dedicated to this topic. We need to emphasize that our use of an uniterated Helmholtz operator as a weighting function is quite distinct from the use, in other covariance generation schemes, of elliptic operators iterated to emulate a diffusive process. In our scheme, the Helmholtz operator acts as a de-smoothing operator (reducing the second moments of the impulse response function) and can introduce negative side lobes, which a scalar positive weight cannot do. To see how this works for an isotropic example in two dimensions, define at each point a first-order composite, or multicomponent, derivative operator $\mathbf{G}^\dagger$acting on some generic field φ:

$$\boldsymbol{\gamma} \equiv [\gamma_0, \gamma_x, \gamma_y]^{\mathrm{T}} = \mathbf{G}^\dagger(\varphi) = \left[a\varphi, -b\,\frac{\partial\varphi}{\partial x}, -b\,\frac{\partial\varphi}{\partial y}\right]^{\mathrm{T}}, \tag{11}$$

where a and b are real coefficients that are, in general, spatially varying. This vector valued $\mathbf{G}^\dagger(\varphi)$ with a scalar argument φ is the formal adjoint of another, scalar valued operator $\mathbf{G}(\boldsymbol{\gamma})$ with a vector argument, $\boldsymbol{\gamma} \equiv [\gamma_0, \gamma_x, \gamma_y,]^{\mathrm{T}}$. Explicitly,

$$\mathbf{G}(\boldsymbol{\gamma}) = a\gamma_0 + \frac{\partial(b\gamma_x)}{\partial x} + \frac{\partial(b\gamma_y)}{\partial y}. \tag{12}$$

Therefore, a combination $\mathbf{W}(\varphi) = \mathbf{GG}^\dagger(\varphi)$ determines a positive self-adjoint operator of the Helmholtz type. In this case,

$$\mathbf{W}(\varphi) = \mathbf{G}\mathbf{G}^{\dagger}(\varphi) = a^2\varphi - \frac{\partial b^2 \partial \varphi}{\partial x^2} - \frac{\partial b^2 \partial \varphi}{\partial y^2}. \quad (13)$$

The advantage of introducing a Helmholtz weighting of this form is that it opens a greater possible range of the covariance shapes that can be formed from positive superpositions of such contributions at each generation of the multigrid combination. For example, the derivative terms, sufficiently weighted, give rise to "side-lobes" of negative values in the response to a positive impulse input, which cannot possibly be obtained when the individual multigrid contributions to this response are restricted to being purely the self-convolved beta functions at each scale (since these are positive-valued functions within their support).

In practice, we would wish the Helmholtz operator to be, like our aspect tensor, anisotropic. In this case, the operator $\mathbf{G}^{\dagger}$ would take on the more general form:

$$\mathbf{G}^{\dagger}(\varphi) = \left[a\varphi, -b_{xx}\frac{\partial \varphi}{\partial x} - b_{xy}\frac{\partial \varphi}{\partial y}, -b_{yx}\frac{\partial \varphi}{\partial x} - b_{yy}\frac{\partial \varphi}{\partial y}\right]^{\mathrm{T}} \quad (14)$$

(but still using no higher than the first-order derivatives). Without loss of generality, we can take the matrix of these new b coefficients to be symmetric ($b_{xy} = b_{yx}$) and positive definite, but other styles of the decomposition of a general nonisotropic Helmholtz operator into mutually adjoint factors are also possible. If the square of this matrix of b coefficients is kept proportional to the aspect tensor $\mathbf{A}$, then the covariance shape is consistently a stretched image of a response that results from applying an isotropic Helmholtz operator to an isotropic beta filter.

The formalism for depiction of the background error covariance is exactly satisfied in the case of the described multigrid scheme, where we can, respectively, denote contributions from the adjoint interpolations ($\mathbf{I}^{\dagger}$), the adjoint (conserving) filtering ($\mathbf{F}^{\dagger}$), and the Helmholtz weight right-factor ($\mathbf{G}^{\dagger}$), as $\mathbf{C}^{\dagger} = \mathbf{G}^{\dagger}\mathbf{F}^{\dagger}\mathbf{I}^{\dagger}$ for a single additional generation. Then we can form the modulating weight's left-factor ($\mathbf{G}$), the direct (smoothing) filtering ($\mathbf{F}$), and the direct interpolations ($\mathbf{I}$), as $\mathbf{C} = \mathbf{IFG}$. Finally, we express the covariance of background error, in the idealized case where it only involves the single coarser scale of filtering, as follows:

$$\mathbf{B} = \mathbf{IFG}(\mathbf{IFG})^{\dagger}. \quad (15)$$

A schematic representation of that procedure is shown in Fig. 3.

b. Grids and decompositions

It may be useful at this point to clarify some of the conventions used throughout the paper:

- Grid: represents a distribution of grid points over a computational domain
- Analysis grid: contains control variables
- Filter grids: handle filter variables

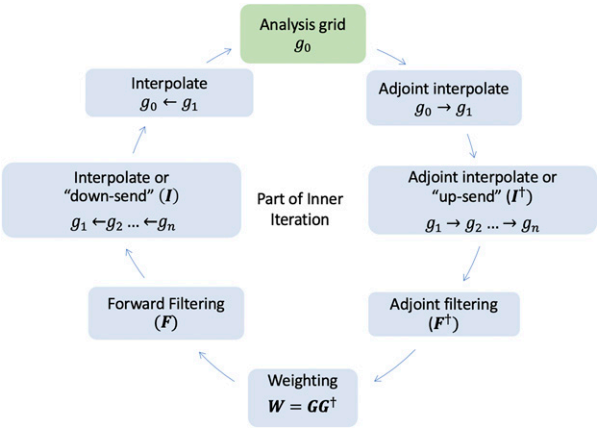

FIG. 3. Schematic illustration of filtering procedure. Here, g_0 denotes the analysis grid, and $g_1, \ldots, g_n$ denotes the successive filter grids within the multigrid structure. See text for explanation of other used symbols.

Filter grids are defined at several different resolutions, referred to as "generations," and denoted here as $g_1, g_2, \ldots$, where generation 1, that is, g_1, has the highest resolution; generation-2, or g_2, has half the resolution of g_1 in each direction, etc. In the simplest case, all grids from this multigrid structure cover the same computational domain. In the generalized version of the algorithm, higher generations of the filter grid may cover somewhat fictively larger domains, created by padding with zeros, for reasons that will become clear later on. Generally, the filter grid at generation-1 has a resolution equal to, or lower than, that of the analysis grid, conveniently denoted by g_0. In addition, we will use the term "decomposition" or "distribution" to describe the arrangement of grid points over processing elements (PEs) of a parallel-computing platform.

Generally, the computational domain is decomposed in such a way that all PEs at all generations have the same number of grid points. Mapping from the analysis to the filter grid is done using adjoint interpolations, and mapping back following filtering is done using direct interpolations. In our case, we use linearly weighted quadratic interpolation so as to preserve continuity of derivatives. The whole cycle of filtering in a variational optimization procedure is enclosed within each inner iteration of the minimization procedure, as shown in Fig. 3. It starts with mapping from the analysis to the first generation of filter grid, filtering, and mapping back to the analysis grid. Filtering itself consists of two successive steps: adjoint ("conserving" stage) and direct ("smoothing" stage). The "conservative" and "smoothing" terminology comes from the attributes of the normalized (unweighted) inhomogeneous filters. A conserving filter, like an "adjoint interpolator," has the property of ensuring that the integrated substance after the filter's application is identical to the integrated substance of the input distribution upon which it acts, even as the aspect tensor varies. This "conserving" attribute may be thought of as the consequence of the filter, or adjoint-interpolation, being the exact adjoint of a filter, or interpolator, whose action upon an input that happens to be perfectly uniform,

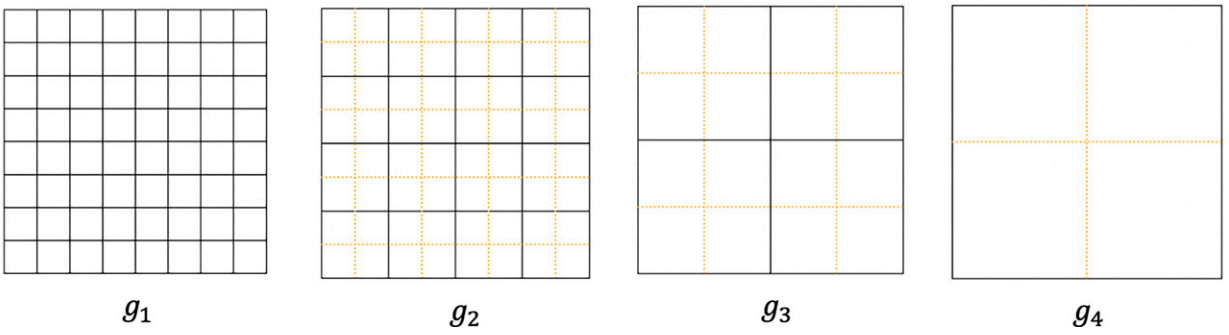

FIG. 4. Schematic presentation of successive generations in the case of simplest multigrid scheme. The squares represent computation domains. Each next generation has half of resolution of the previous in each direction, decomposed in proportionally half of processors.

leads to an identical uniform value in the output. Of course, it would be desirable in many applications if a single filter could be fashioned to always possess both the exactly "conserving" and "smoothing" attributes simultaneously (e.g., with recursive filters, this is possible even when the response is required to be both anisotropic and spatially inhomogeneous), but for our purposes, this dual attribute is not a necessary requirement and, in the case of our beta filters, is not something that is easily achieved either.

In the context of building a practical covariance operator to characterize the spatial distribution of errors of a suitably chosen variable in an atmospheric forecast background field, then apart from the smooth positive weighting that always modulates the individual quasi-Gaussian contributions in order to match the intended background variance (and, to some degree, the intended covariance "shape"), it is the attribute of "smoothness" that we most want to preserve in the final output. For this reason, we invariably perform the self-adjoint filtering combinations in such a way that the "conserving" adjoints of the "smoothing" filters and interpolators are applied *first* in each factored self-adjoint combination, and the corresponding filters and direct interpolators themselves are applied *last*. These guaranteed self-adjoint contributions are then further combined, but now additively, or "in superposition," to synthesize the final covariance operators for each control variable of the assimilation, thus guaranteeing smoothness, self-adjointness, and nonnegativity.

It is acceptable for the $\mathbf{B}$ operator to possess a null space as long as the null degrees of freedom do not correspond to scales or structures where a nontrivial analysis increment is required. As a spectral analysis of the continuum beta filters reveals (Purser 2020a), there will be wavenumbers where the spectral transform changes sign or momentarily vanishes, implying a null space for the corresponding self-convolved filter. But such sets of wavenumbers are of zero measure and are unlikely to play a role in the discrete filters, especially once a few such self-convolved filters, at different characteristic scales, have been positively superposed in the multigrid scheme. Where there almost certainly is a substantial null space is at those scales unresolved by the generation-1 filter grid of the multigrid's structure, but resolved by the analysis grid itself, since these barely resolved components of the analysis grid will be invisible to (i.e., in the null space of) the adjoint interpolation operator that goes from the analysis grid to the first generation of the filter grids.

In its simplest form, our parallel multigrid scheme assumes that all generations cover the same area and that each consecutive grid generation halves the resolution (in each direction) of the previous, lower generation. The computational domain is decomposed in such a way that all PEs at all generations have the same number of grid points.

A typical scheme fitting this situation is shown in Fig. 4, where successive grid generations, g_1, g_2, g_3, and g_4, cover the same domain with successively 64, 16, 4, and 1 PEs, respectively, each with the same number of grid points.

Consistently with our convention, we declare the highest generation to be g_n, and in our example in Fig. 4 index n is equal to 4. Suppose we write the interpolation operator from the filter grid of g_k to the analysis grid g_0 as $\mathbf{I}_k$, with adjoint $\mathbf{I}_k^\dagger$, and correspondingly write the direct and adjoint beta filters at generation g_k as $\mathbf{F}_k$ and $\mathbf{F}_k^\dagger$, and the weighting operators there as, $\mathbf{G}_k$ and $\mathbf{G}_k^\dagger$. Then the mutually adjoint factors, $\mathbf{C}$ and $\mathbf{C}^\dagger$, of covariance $\mathbf{B}$ with n multigrid generations can be expressed in the block-vector forms:

$$\mathbf{C} = [\mathbf{I}_1\mathbf{F}_1\mathbf{G}_1, \quad \ldots, \quad \mathbf{I}_1\ldots\mathbf{I}_n\mathbf{F}_n\mathbf{G}_n] \tag{16}$$

(a comma separator indicating a block-row-vector) and

$$\mathbf{C}^\dagger = \left[\mathbf{G}_1^\dagger\mathbf{F}_1^\dagger\mathbf{I}_1^\dagger;\ldots;\mathbf{G}_n^\dagger\mathbf{F}_n^\dagger\mathbf{I}_n^\dagger\ldots\mathbf{I}_1^\dagger\right], \tag{17}$$

(semicolon separators indicating a block-column-vector). In this way, we see clearly how $\mathbf{B}$ becomes a self-adjoint superposition of weighted filter applications for all n different scales of the multigrid hierarchy. Also, by maintenance of the explicit $\mathbf{C}\mathbf{C}^\dagger$ factorization we ensure that the scheme is amenable to conjugate gradient minimizers using the so-called square root basic preconditioning, if desired.

c. Stages of multigrid procedure

Our multigrid algorithm, as schematically described in the boxes in Fig. 3, consists of the following steps:

1) Adjoint interpolate filter fields (or "up-send") them from g_1 to g_2, then from g_2 to g_3, and so on, all the way to g_n ($\mathbf{I}^\dagger$)

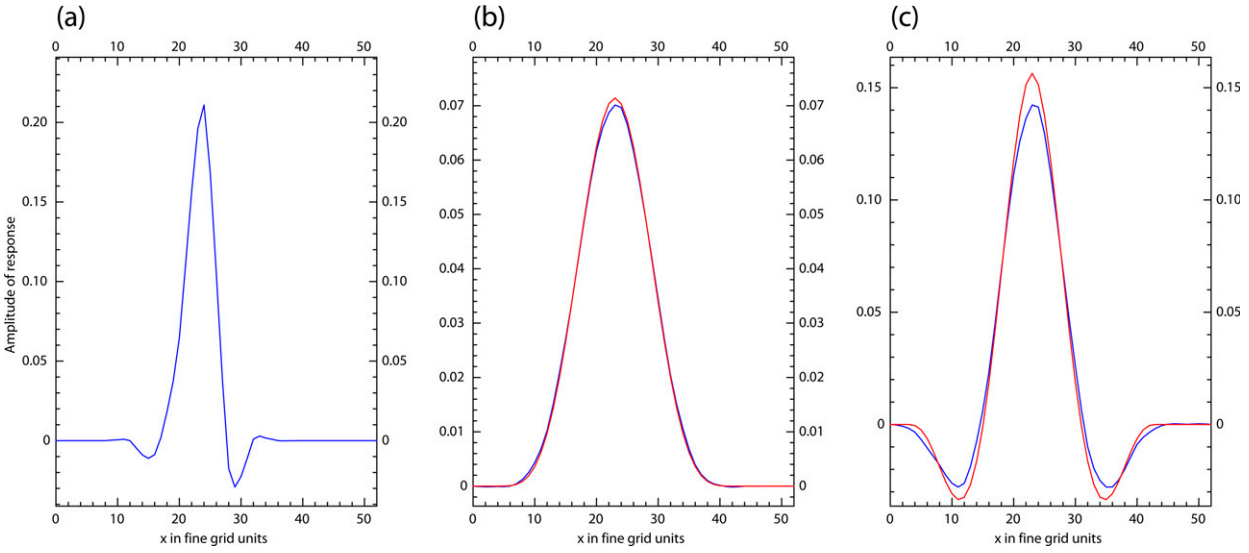

FIG. 5. Idealized examples in 1D of operator response function at fine filter grid g_1 for an initial unit impulse at a location that does not correspond to a coarser grid point in the MG hierarchy. (a) The result for adjoint and direct interpolation across two generations to g_3 and back without additional operators. (b) As in (a), but with beta filters $\mathbf{FF}^{\dagger}$ applied at coarse grid g_3 (blue curve) compared to the equivalent filtering without interpolations at fine grid g_1 (red curve), showing that any additional smoothing effect of the interpolation operators is minimal. (c) As in (b), but with an additional Helmholtz weighting to engender sidelobes.

2) Apply the adjoint beta filter at all generations ($\mathbf{F}^{\dagger}$)
3) Apply weighting at all generations ($\mathbf{W} = \mathbf{GG}^{\dagger}$)
4) Apply the direct beta filter at all generations ($\mathbf{F}$)
5) Interpolate (or, "down-send") result of the adjoint of the beta filter from g_n to g_{n-1}, and add it to the existing adjoint of the beta filter at that lower generation. Then, repeat the procedure all the way to g_1 ($\mathbf{I}$)

Steps 1 and 5, up-sending and down-sending, are sequential, while step 2 (adjoint beta filtering), step 3 (weighting, possibly with the Helmholtz style), and step 4 (direct beta filtering) are parallel. Technically, in the up-sending stage we halve the resolution and correspondingly reduce the number of the processing elements, while in the down-sending stage we double the resolution and increase the number of processing elements.

For example, in a two-generation multigrid scheme, the covariance operator would be written

$$\mathbf{B} = \mathbf{I}_1\left[\left(\mathbf{I}_2\mathbf{F}_2\mathbf{W}_2\mathbf{F}_2^{\dagger}\mathbf{I}_2^{\dagger}\right) + \left(\mathbf{F}_1\mathbf{W}_1\mathbf{F}_1^{\dagger}\right)\right]\mathbf{I}_1^{\dagger}. \tag{18}$$

Here, we use $\mathbf{W}_k$ to express the scale-weight at generation k.

To give some insight into the effect that these operators have, Fig. 5 shows the response function in the case of a three-generation MG scheme to a unit impulse, in the 1D case. Figure 5a shows the effect at the fine grid of operating with the combination only of adjoint and direct interpolation operators from an input pulse at the fine grid that does not coincide with the points of the coarser grids. Figure 5b shows, in the red curve, the application of a broad beta filter combination, $\mathbf{FF}^{\dagger}$ on the fine grid, together with the equivalent amount of smoothing of the adjoint-interpolated initial impulse done at the coarse grid and interpolated back down again to the fine grid. Figure 5c shows the same comparison except with the inclusion of a Helmholtz operator. The effect of the adjoint and direct interpolation operators on these smooth and broad functions is minimal. The ability of the Helmholtz operators to form sidelobes is apparent in Fig. 5c.

Since we always deal with several 2D and 3D variables, it is useful to create a composite variable for filtering which simplifies coding and speeds up the execution by avoiding short messages and redundant repetitions.

4. Organization of computation

The basic MG structure, as described in the previous section, assumes that all grid generations reserve separate processors, which guarantees a parallel execution of the beta filter. However, though appealing, a direct implementation of such a paradigm is impractical, and is especially hardly applicable across a large range of processing elements. Thus, we had to carefully investigate how best to apply the MG beta filter and at the same time efficiently utilize available processing resources. To illustrate the nature of the problem, we will first briefly describe various solutions, preliminarily outlined in Rančić et al. (2020), that we have considered along the way, before showing the one that was able to optimally satisfy most of the necessary requirements.

Let us first denote by N and M the numbers of processors which hold control fields in x and y directions, respectively. Let us also assume that each of the analysis fields is evenly spread among processors, which can always be accomplished using padding toward ends of the domain using one or another method for extrapolation. Alternatively, one can slightly modify the size of input background fields so that they can be uniformly spread across given processors. Thus, we

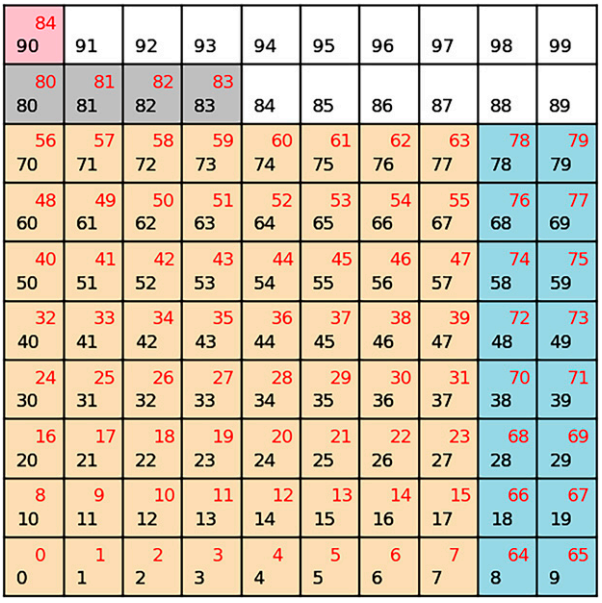

FIG. 6. Schematic presentation of option 1 for decomposition of the domain on the analysis and filter grid processors, which enables a parallel execution among MG generations. The analysis grid is decomposed over 100 PEs. Filter grid generations g_1, g_2, g_3, and g_4 occupy the same space with 64 (yellow), 16 (blue), 4 (gray), and 1 (pink) PEs, respectively. Note that 15 PEs that hold the analysis grid do not participate in filtering.

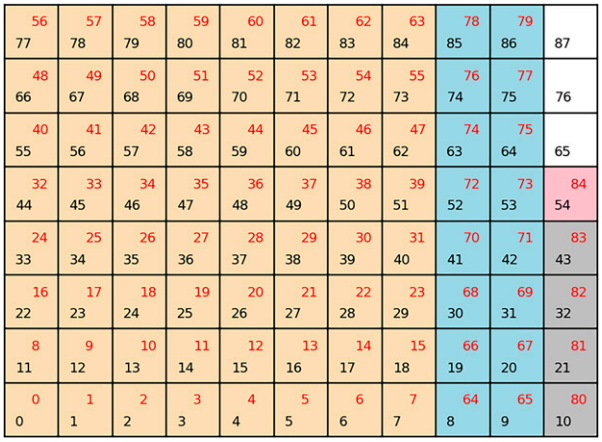

FIG. 7. Schematic presentation of option 2 for decomposition of the domain on the analysis and filter grid processors, which enables a parallel execution among MG generations. The analysis grid is distributed over 80 PEs. Unlike option 1, there is no need for re-decomposition in the shorter dimension's y direction. Filter grid generations g_1, g_2, g_3, and g_4 occupy the same space with 64 (yellow), 16 (blue), 4 (gray), and 1 (pink) Pes, respectively. Only 3 PEs that hold the analysis grid now do not participate in filtering.

assume after such regularization, the analysis variables (g_0) will be uniformly spread with $p_0 = N \times M$ processors.

At the same time, filtering through generations g_1 to g_n will require a specific number of processors. If the highest generation g_n requires just one PE (i.e., if $p_n = 1$) and assuming that each lower generation has 4 times more processors, the overall number of distinct processors for filtering will be

$$\begin{aligned} p_f &= p_n + p_{n-1} + \cdots p_1 \\ &= 1 + 4 + \cdots + 4^{n-1} \\ &= \frac{4^n - 1}{3}. \end{aligned} \tag{19}$$

For example, for the case from Fig. 4, where $n = 4$, we get that the entire number of processors, $p_f = 85$. That also means that the number of processors p_0, which hold the analysis grid, must be large enough so that it can handle all 85 processors for filtering. In addition, we need to: (1) find out how to arrange these 85 processors for filtering among processors which hold the analysis grid; and (2) develop an efficient algorithm which will provide not only mapping between the analysis and first generation of filter grid, but also the re-decomposition between processors that host these two grids.

a. Option 1

Figure 6 shows one solution for the organization of the calculation. For simplicity, we assume that analysis grid (g_0) is decomposed over 10×10 PEs. 64 processors handling grid g_1 are arranged from the lower left corner of the arrangement for g_0 in a topological order that corresponds to the geometry of the domain. We developed a relatively efficient code for remapping and re-decomposition of g_0 to g_1 and back. The higher generations are then fit among free PEs of g_0. The problem with that approach is that too much of the computation time is spent on data motion in re-decomposition between grids g_0 and g_1.

b. Option 2

Since the RTMA mainly runs on rectangular domains, with typically one dimension significantly exceeding the other, we next tried another solution, shown schematically in Fig. 7. This solution assumes that g_0 and g_1 use the same number of processors in one, typically y direction, which eliminates the need for re-decomposition in that direction, reducing in this way the computation time. If we want to use more processors for g_0 we can do that by adding them to the right side of the presented arrangement.

c. Option 3

The arrangement of processors in the next solution, shown in Fig. 8, is based on the idea that in order to maximize use of the processors, we can progressively reduce the number of vertical levels, ($L/2$, $L/2$, …, $L/2^{n-1}$) for higher generations (g_2, g_3, …, g_n), where L is the number of vertical levels of generation-1. If we place higher generations (g_2, g_3, …, g_n) in the processors of generation g_2 and execute generations higher than g_1 successively, but in parallel with generation g_1, then the overall clock time (τ) would be about the same as if all generations were run in parallel. That is because the time

56 70	57 71	58 72	59 73	60 74	61 75	62 76	63 77	78 78	79 79
48 60	49 61	50 62	51 63	52 64	53 65	54 66	55 67	76 68	77 69
40 50	41 51	42 52	43 53	44 54	45 55	46 56	47 57	74 58	75 59
32 40	33 41	34 42	35 43	36 44	37 45	38 46	39 47	72 48	84 73 49
24 30	25 31	26 32	27 33	28 34	29 35	30 36	31 37	70 38	83 71 39
16 20	17 21	18 22	19 23	20 24	21 25	22 26	23 27	68 28	82 69 29
8 10	9 11	10 12	11 13	12 14	13 15	14 16	15 17	66 18	81 67 19
0 0	1 1	2 2	3 3	4 4	5 5	6 6	7 7	64 8	80 65 9

FIG. 8. Schematic presentation of option 3 for decomposition of the domain on the analysis and filter grid processors. The analysis grid is distributed over 80 PEs. Unlike option 1, there is no need for re-decomposition in the y direction. Here, higher generations have progressively lower vertical resolution, and are executed sequentially but in parallel with generation g_1. Filter grid generations g_1, g_2, g_3, and g_4 occupy the same space with 64 (yellow), 16 (blue), 4 (gray), and 1 (pink) PEs, respectively. Now all processors participate in filtering.

spent in successive calculations for generations with progressively vertically halved resolutions is

$$\frac{\tau}{2} + \frac{\tau}{2^2} + \cdots + \frac{\tau}{2^{n-1}} \leq \tau. \tag{20}$$

In that case, with no processors outstanding, all of them would be engaged in filtering and thus would be spreading the computational burden in the most efficient way.

d. Option 4—Generalized solution

None of these solutions turned out to be fully satisfactory. Essentially, the main problem with all of them is the lack of flexibility to enable an efficient generalization of the code for operation on a different number of processors. A truly effective code must not be hardwired but must be able to effectively adjust to different numbers of processors and various resolutions. The final option that we will show next satisfies all these important criteria. However, to that end, we had to give up a full parallelism of the multigrid generations, and instead require that the higher generations are calculated in parallel among themselves, but sequentially with generation-1.

We explain this solution, which will be referred to as "generalized," with the aid of stencils in Fig. 9. We assume that the analysis grid (g_0) operates on 11×8 PEs (as in option 2). However, the generalized method will work with any rectangular arrangement of processors at generation-0, with the only condition that it is possible to uniformly distribute grid boxes among them. The essence of the method is that generation g_1 of the filter grid is distributed across all given processors, just as the analysis grid (g_0) is. Thus, mapping between them (the first and the last tasks on Fig. 3) is done only using adjoint and direct interpolations within the same processors, no longer requiring re-decomposition and motion of data, which eliminates the main slowdown of all other previously shown options. Additionally, this allows that calculation of generation g_1 of the filter grid is spread among all processing

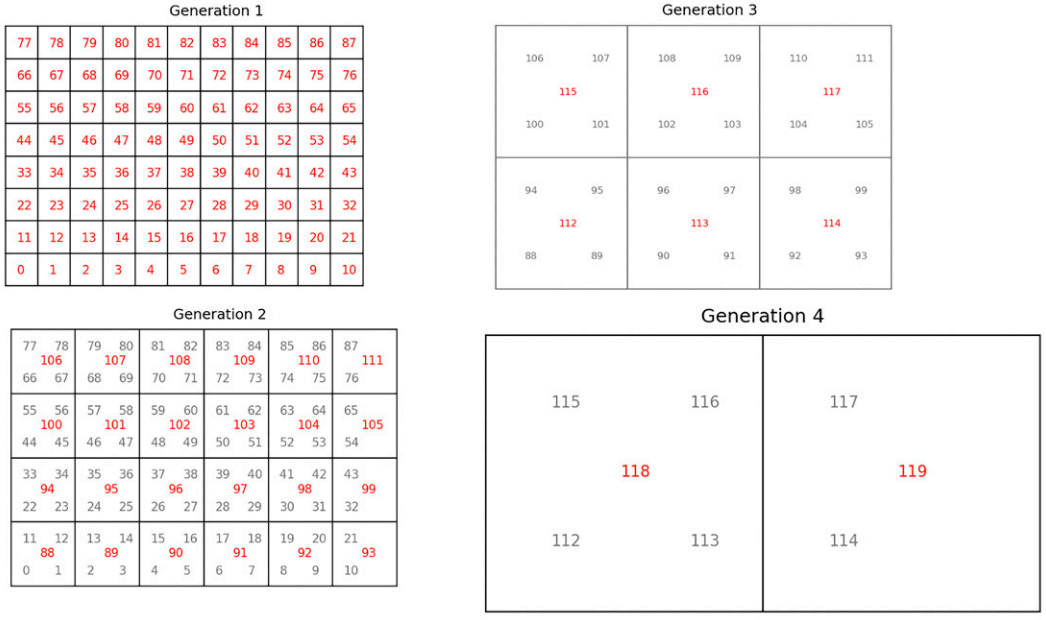

FIG. 9. Option 4: Higher generations are created by always grouping 2×2 PEs from the lower one in the generalized version. In the case that 2 PEs are missing at the right or upper boundary (or both), we replace their content with zeroes, keeping the number of grid points in the processor of the new generation the same as in all other processors.

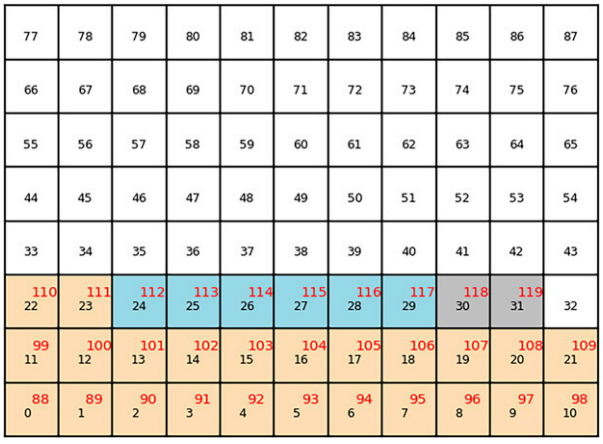

FIG. 10. Option 4: Physical location of processors of higher generations in processors of generation-1 (and analysis) grid.

elements, not just a portion of them. In the solutions discussed so far, we group four processors from the lower generation to access the next higher generation. To this end, the number of processors of generation-1 had to be divisible by 4. Within the generalized method, we again form higher generations by grouping the content of 2×2 processors through adjoint interpolations. However, if the number of processors in one or both directions is odd, we simply add in that direction an extra virtual processor by supplying its content with zeros, as shown in Fig. 9.

Note that the effective computational domain size stays the same, and that there is no computation in the portion of processors that cover fictive (or "ghost") space padded with zeros. All higher filter generations are executed concurrently, but sequentially with generation-1. Thus, we can physically place the content at these processors in the processors of generation-1, as it is done in Fig. 10, which supplements Fig. 9.

In this paradigm, all generations have the same data load, and g_0 is executed sequentially with higher generations. However, since the generation-1 workload is spread among all available processors, and since there is no need for re-decomposition in mapping between g_0 and g_1, based on timings that we got in the preliminary testing, we estimate that this method, in addition to providing a full generalization of MG beta filtering, would actually perform faster than any of the previous versions. The coding of this approach can be organized in such a way that identification of higher generations, and the processors in charge of them, is done automatically. Practically, once the requirement that data of g_0 are evenly spread across given processors, and resolution of the filter grid is set, we can just ask for any number of rectangular arrangement of processors, and the code will automatically handle the higher generations. The only other requirement is that resolution of filter grid within a processor (defined in this case as the number of grid spaces in each direction) is divisible by 2^{n-1} where n is the number of generations. That option guarantees that we can use mirror boundary conditions for the increments at the same physical location for all generations, which is a simple way of defining them in a manner that guarantees self-adjointness.

Assuming that data of both the analysis and the first generation of filter grid can be evenly distributed among the given $N \times M$ processors, these numbers become the only input in the code. In addition to increasing the versatility of the method, option 4 showed the best computational efficiency, as will be demonstrated later in section 5, and therefore it became our primary solution for organization of the computation in application of the beta filter for modeling of covariances. Various opportunities for further speedup of the code are discussed in appendix A.

5. Test results

In this section we present results of several preliminary tests that we ran during the development of this method. All tests are run on the Weather and Climate Operational Supercomputing System (WCOSS).

a. Case 1

In this set of tests, we were not concerned with the decomposition but rather with various effects that the parameters of the aspect tensor and the scale-weight operators may have on the shape of covariances. The results of the idealized homogeneous 2D MG beta filter acting on a delta function impulse, summarized in Fig. 11, are derived using the simplest multigrid scheme shown before in Fig. 4. Parameters A_0, ϕ, and θ come from the definition of the aspect tensor in (4), and the coefficients a and b defining the isotropic Helmholtz style of the scale weight operators, are introduced in (11). Note that only the last of the illustrated examples uses a nontrivial Helmholtz operator (and then, only at the highest generation). The large weights at the higher generations spread a covariance resulting from filtering of an initial delta function impulse, and at the same time increase its amplitude. The definition of the aspect tensor can produce various rotations of covariance subjected to stretching. The areal parameter A_0 spreads or shrinks the effective area of the covariance, but the larger values require larger halos, increasing the cost of filtering.

b. Case 2

In the next set of tests, we investigate the 3D problem through a setting that closely fits the realistic conditions in the GSI. We ran a series of tests of a stand-alone version of the multigrid beta filter with a resolution of the analysis grid that corresponds to the RTMA domain (1804×1072 grid distances) and with a filter grid at resolution (1760×960 grid distances), both with 50 vertical levels. In terms of the decomposition of the aspect tensor given by (4), the characteristic areal parameter A_0 was 1, which required eight points of halo in all directions. The other, anisotropy parameters, θ and ϕ, of the aspect tensor were set to zero. A filtering option with application of the 2D version of the radial filter in the horizontal and 1D in the vertical direction was used. The code used six 3D variables and four 2D, one of which did not share common boundaries

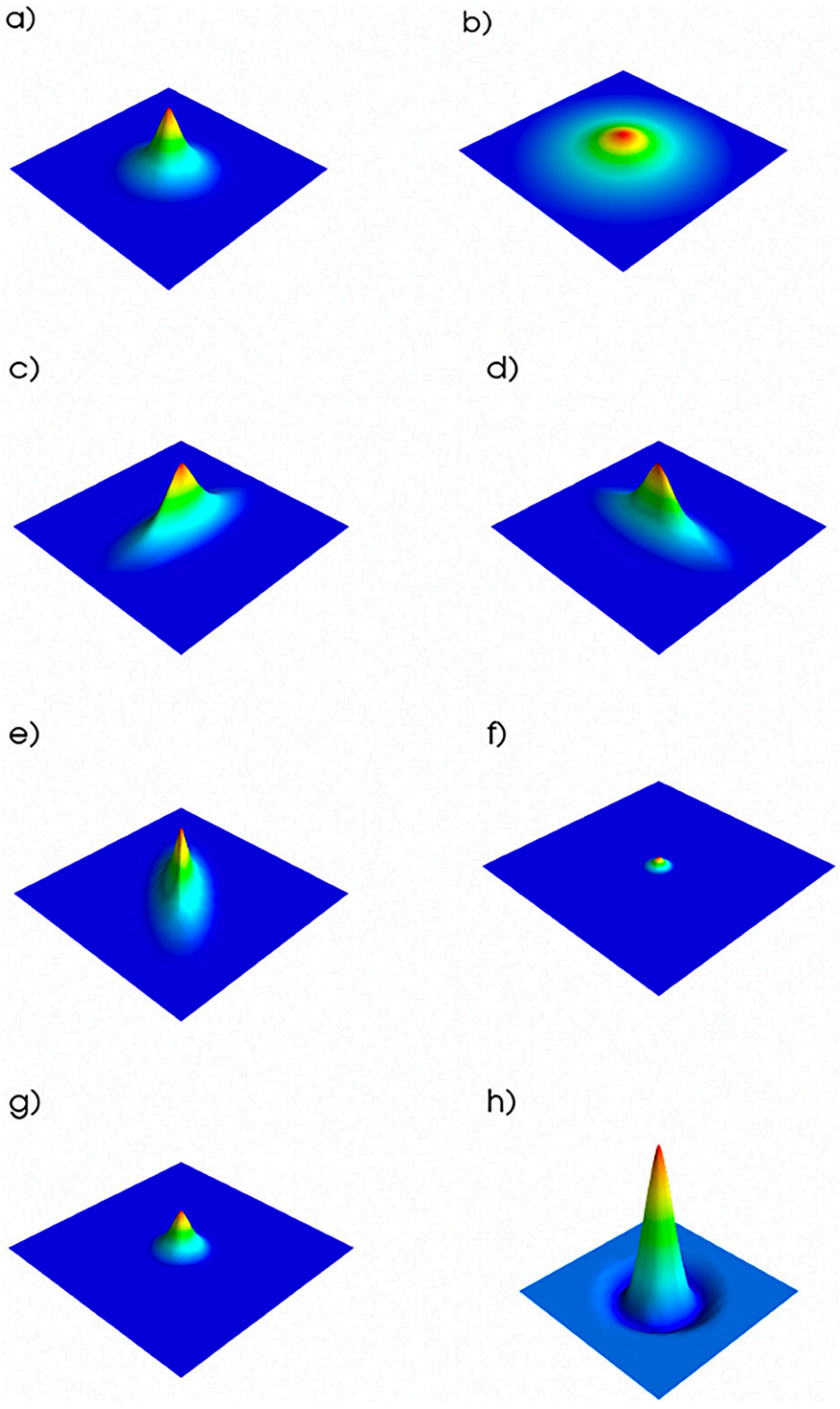

FIG. 11. Effects of parameters of aspect tensor and Helmholtz "scale-weights" of the MG beta filter acting on a delta function impulse. (a) A standard set of parameters is applied for the aspect tensor, $A_0 = 0.125$, $\phi = 0$ and $\theta = 0$, and for the set of the Helmholtz "scale weight" operators, $a = \{1, 1, 1, 1\}$ and $b = \{0, 0, 0, 0\}$. All other covariances are referred to that one. (b) $A_0 = 4$; (c) $\phi = 1$, $\theta = 0$; (d) $\phi = 1$, $\theta = \pi$; and (e) $\phi = 1$, $\theta = \pi/2$. The last three panels show effects of the "scale-weight" operators. (f) $a = \{1, 1, 0, 0\}$; (g) $a = \{1, 1, 1, 0\}$; and (h) with the negative values has $b = \{0, 0, 0, 5\}$.

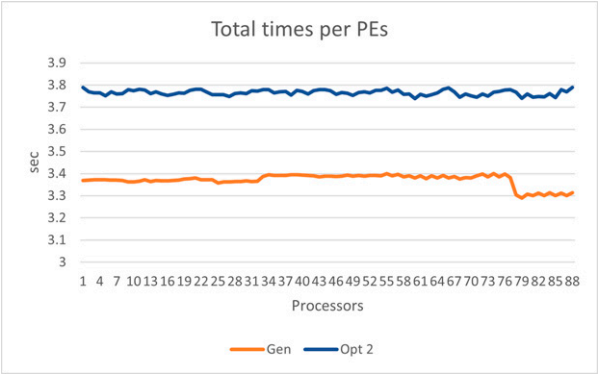

FIG. 12. Time taken for each PE derived in tests on 88 PEs with option 2 and the generalized scheme introduced as option 4. There is a gain of about 10% in efficiency.

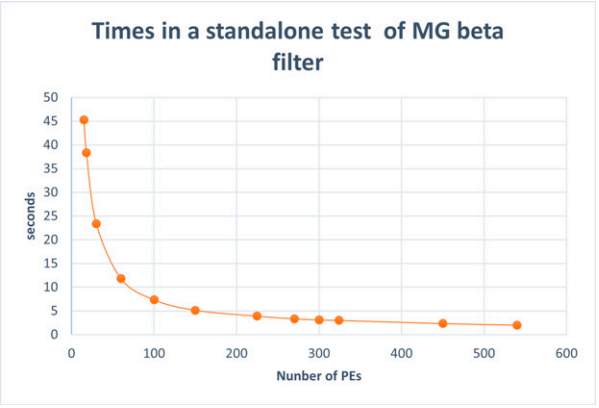

FIG. 13. Times derived in tests of a stand-alone version of the multigrid beta filter code using generalized option 4.

with neighbors, which mimics the situation that exists in the GSI. The six 3D variables used for filtering in GSI are streamfunction, velocity potential, temperature, specific humidity, ozone, and cloud condensate. The four 2D variables are surface pressure, sea surface temperature, land surface temperature and ice surface temperature. At the end of each inner iteration the GSI combines these last three distinct variables, with the help of land/sea/ice masks, to get the skin temperature.

Figure 12 shows timings derived in a test run on 11 × 8 PEs in which the generalized option 4 is compared against identical test run with option 2, the most promising among all other considered. (Recall that the other two options were presented only to better explain the problem of decomposition that we encountered while developing this method.) In option 2, all filter generations are run in parallel, but that required a redecomposition of data in mapping between the analysis grid and the first generation of filter grid and back, and the first generation of filter grid ran on fewer processors. The motivation for introducing option 4 was to generalize the MG beta filter. However, it turned out that this approach also increases the efficiency of calculation, in this case by about 10%. Note that option 2 would be very difficult to apply across various arrangements of processors because of the need to tailor the code that describes the multigrid structure specifically for each new decomposition. In contrast, the generalized version can run on various constellations of processors, without any need for additional interventions.

The way that option 4 scales with the number of processors in a test with the generalized version of the stand-alone version derived using resolution with 2430 × 1080 grid intervals on the analysis grid and 2160 × 900 on the filter grid, with up to 540 PEs running on the WCOSS, is shown in Fig. 13.

c. Case 3

The MG beta filter was implemented in the GSI over the RTMA domain and a series of preliminary integrations were run using a single observation test case, only using the generalized option 4. At this time, we were focused on potential gains in computational efficiency in comparison with the parallel tests run with the recursive filter, while other potential benefits that are expected to come through a better and a more versatile description of covariances across spatial scales of the MG beta filter will be analyzed at a later stage.

In this exercise we used a slightly reduced version of the RTMA domain with a resolution of 1792 × 1056 grid intervals and prescribed various uniform decompositions across processors in a more measured way as summarized in Table 1. The value of beta filter parameter p was set to 2, and the aspect tensor was adjusted so that the resulting covariance had a similar shape as in the test case with the recursive filter. In this case, the size of the halo was eight grid intervals, which we found experimentally. Note that we used here a slightly lower resolution for the filter grid, making sure that in each direction the number of intervals is divisible by 8, which enables the running of four multigrid generations in all tests.

The derived results are presented in Table 2 and Fig. 14, which show, respectively, times, and a plot of inverses of times spent on filtering per iteration, as derived in the described tests for the recursive filter (RF) and the MG beta filter. In addition, according to Fig. 14, the MG beta filter scales much better by continuing to increase the efficiency as the number

TABLE 1. Decompositions ($N \times M$) and resolutions of analysis ($n \times m$) and filter ($i \times j$) grids in each processor for different decompositions used in tests of the GSI.

No. of PEs	168	256	336	448	512	704	768	924	1056
$N \times M$	14 × 12	16 × 16	28 × 12	28 × 16	32 × 16	32 × 22	32 × 24	28 × 33	32 × 33
$n \times m$	128 × 88	112 × 66	64 × 88	64 × 66	56 × 66	56 × 48	56 × 44	64 × 32	56 × 32
$i \times j$	120 × 80	96 × 64	56 × 80	56 × 64	48 × 64	48 × 40	48 × 40	56 × 24	48 × 24

TABLE 2. Times per single iteration in seconds derived in a single observation test of the GSI using recursive filtering (RF) and the multigrid beta filter (MF) running with different numbers of processors (PEs).

No. of PEs	168	256	336	448	512	704	768	924	1056
RF	17.162	8.942	8.769	8.556	8.492	8.399	8.367	8.363	8.363
MG	7.593	4.951	3.781	3.057	2.660	1.857	1.746	1.333	1.155
RF/MG	2.260	1.806	2.319	2.799	3.192	4.522	4.792	6.273	7.238

of processors increases. In contrast, the RF quickly reaches a saturation point, and further increase in the number of processors does not significantly improves its performance. Thus, with an increasing number of processors, the MG beta filter becomes more and more efficient relative to the RF (see last row in Table 2). With the set parameters, MG beta filter in these tests becomes even *7 times more efficient* when running with 1056 PEs.

6. Conclusions

This paper summarizes work on development of a new method for modeling of background error covariance for application in the 3D RTMA system as a replacement for the recursive filter. The key prerequisite for the success of this enterprise, consisting of 3D analysis at a horizontal resolution of 2.5 km in frequent time intervals of 15 min, is a vastly improved efficiency. The new approach to modeling of the background error covariance, which was discussed here, is one of the key components for the success of that effort.

The new approach uses a beta function for construction of the filter, which has a compact support and thus is much better adapted for parallelization than the recursive filters. The effect of various scales is taken into account through a multigrid procedure, which is generalized so that it can run on an arbitrary number of processing elements, assuming only that the number of grid points on the filter grid in each direction can be presented as a product of two integers. A more detailed analysis of effects of various parameters of the MG beta filter on the covariances, separate from computational

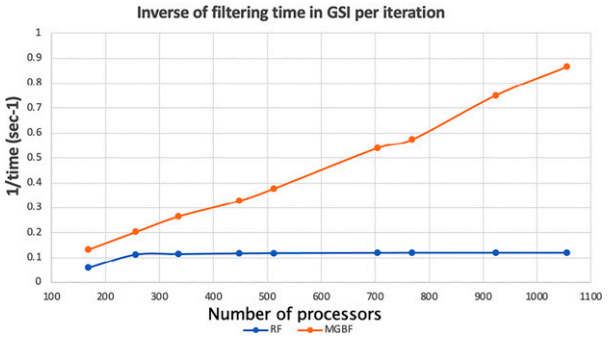

FIG. 14. Inverse of time for filtering per iteration in a single observation test for the recursive filter (RF) and for the multigrid filter (MG) plotted against the number of processors. The capability of the RF for further improvement quickly becomes saturated, while the MG filter exhibits a trend that promises to keep increasing the efficiency with the addition of more processors.

implementation, would require much more space, but we hope to provide such an analysis in future, maybe in the context of response to a single observation forcing within the GSI.

The described MG beta filter comes in several flavors, some of them still in their final developmental stage, which will be described in this section.

The radial beta filter itself has 1D, 2D and 3D versions. Thus, we can apply filtering in:

- Horizontal directions only
- All three directions, using a sequence of 2D filtering in horizontal and 1D in vertical for 3D variables. (Timings shown in Figs. 12 and 13 are derived with this combination.)
- All three directions, using a 3D version of the radial beta filter

Potentially highly efficient "line" versions of the suite of beta filters, the "Triad" (three sequentially applied components, in 2D), and the "Hexad" (six sequentially applied components in 3D), are also being developed, which may replace the present "radial" versions of our beta filter. These advantageous alternatives are achieved by exploiting the symmetries of a regular grid using the powerful and elegant methods of group theory (Purser 2020b,c), and will be described in a future article. All described types of the radial beta filter (two and three-dimensional, isotropic and anisotropic) are also available with the line filter. A more detailed comparison of the effects of these various filtering flavors is also reserved for future investigations.

We only foresee applying the method on structured grids, which is a category that includes cubic and icosahedral global grids, since these grids are well-adapted to multigrid treatments. Although the radial beta filter, with some effort, could be made to apply to an unstructured grid, it is not clear how the multigrid architecture could generalize in the unstructured case.

The primary objective here was to improve efficiency through a better utilization of multiprocessing computing resources. Yet the developed method has the potential to generally improve performance of the analysis. For example, a consistent extension of this approach led us to a fully 4D extension (the so-called "Decad" algorithm, Purser 2020b) giving us a tool that would enable future extension of the RTMA procedure into a fully 4D scheme.

The scalar versions of the scale-weights $\mathbf{W}$ for each generation can be generalized to supply the scheme with cross covariances among the different analysis variables. This can be accomplished by replacing, at each generation, the scalar scale-weight, $\mathbf{W} = \mathbf{G}^2$, separately defined for each analysis variable, by positive-definite symmetric matrices, allowing factoring, $\mathbf{W} = \mathbf{G}\mathbf{G}^{\dagger}$, that simultaneously couple the in situ vector of the different analysis variables together, assuming that all analysis variables are being operated on in parallel. Furthermore, the

more general factored Helmholtz weights themselves can also be generalized into corresponding factored matrices, **G** and their formal adjoints $\mathbf{G}^{\dagger}$, such that they contain both scalar and differential operator elements that cross-couple the different analysis variables. In principle, this should enable the coupled increments to be correlated and to exhibit directional phase shifts in these pairs of coupled increments, as is generally expected for realistic error increments. However, the detailed elaboration of these generalizations of our approach will be left for future studies. Among other things, in the future we also plan to apply machine learning (ML) machinery for estimation of cross covariances using ensembles for training. This, we hope, will allow us to step toward a new paradigm of running a pure variational DA instead of a hybrid, with potentially a huge saving in computational time at some future stage. We also started development of a cubed-sphere version of the MG beta filter, targeting application within the Joint Effort for Data assimilation Integration (JEDI) (https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/index.html), which will allow us to add our approach to the "BUMP" covariance initiative (https://github.com/benjaminmenetrier/bump). More details of this development are supplied in appendix B.

The implementation of a MG beta filter in the GSI (as a backup version for running 3D RTMA) is in the final stage. Since the primary goal of 3D RTMA is application in JEDI, a preliminary implementation of the MG beta filter in JEDI is also under way. An analysis of the effects of the new beta filter on the analysis will be reported elsewhere.

Acknowledgments. This material is based upon work initially supported by the EPIC Program within the NOAA/OAR Office of Weather and Air Quality under the title "Development of a multigrid background error covariance model for high resolution data assimilation," and presently by the Unified Forecast System Research to Operation (UFS R2O) Project, which is jointly funded by NOAA's Office of Science and Technology Integration (OSTI) of National Weather Service (NWS) and Weather Program Office (WPO), [Joint Technology Transfer Initiative (JTTI)] of the Office of Oceanic and Atmospheric Research (OAR). We are grateful to Dr. Jacob Carley for his unwavering encouragement and support of this project and to Drs. Ting Lei and Kristen Bathmann for their helpful suggestions following internal reviews of the manuscript. We also would like to recognize the unknown reviewers for their dedicated efforts, which helped us improve the paper.

APPENDIX A

Further Speedup of Generalized Version of MG Filter

Further potential speedup in application of a generalized version of MG beta filter may come from vertically splitting the 3D arrays of higher generations and sharing their load among processors. Moreover, we can combine this method with a judiciously chosen lower vertical resolution of higher generation, which give us a chance to devise a scheme that will better utilize the processors and further speed up the execution.

TABLE A1. Budgeting of the computational load in the case with 11 × 8 PEs, where the higher grid generations use lower vertical resolutions and vertically split arrays. Originally, the higher grid generations g_2, g_3, and g_4, with the same vertical resolutions, occupy 24, 6, and 2 PEs, respectively. After reduction of their vertical resolutions and vertical partition, they can be evenly redistributed among, respectively, 72, 12, and 4 PEs, to populate all 88 PEs, as shown in Fig. A1.

Generation	Levels	Vertical split	No. of PEs
g_1	50	1	88
g_2	45	3	24 × 3 = 72
g_3	30	2	6 × 2 = 12
g_4	30	2	2 × 2 = 4

In the considered example with 11 × 8 PEs, generation g_1, which occupies all 88 PEs, has 50 vertical levels. Let us assume that generation g_2, which in Fig. 10 occupies 24 PEs is instead of 50 given 45 vertical levels, and divided in 3 layers, each with 15 levels. Generation g_2 can now be evenly distributed among 72 PEs. Similarly, let us assume that generation g_3, that occupies 6 PEs, and generation g_4, that occupies 2 PEs are given 30 vertical levels, and that they are both divided in 2 layers, each with 15 levels. This budgeting is shown in Table A1.

Using this method, we can fully exploit all available processors as show in Fig. A1.

APPENDIX B

Extension of the MG Beta Filter for the Cubed Sphere

Following an early suggestion by Sadourny (1972), the cubed sphere became one of the standard grid geometries in numerical modeling of the atmosphere in general, and is

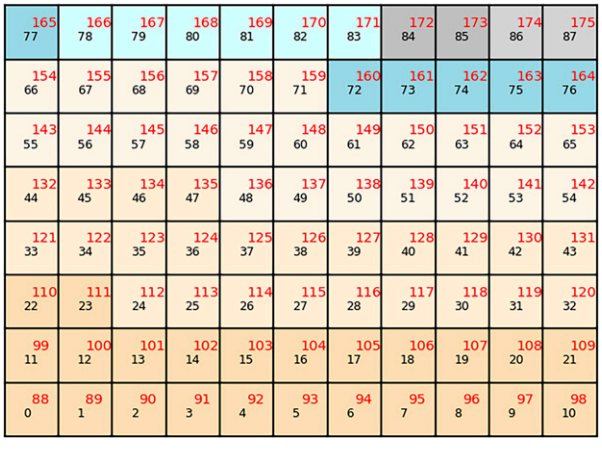

FIG. A1. Distribution of high generations with vertically reduced resolution and sliced subdomains in the considered case with 11 × 8 processors. Higher filter generations in this case perform about 30% faster than the filter generation-1, and all available processors participate in filtering.

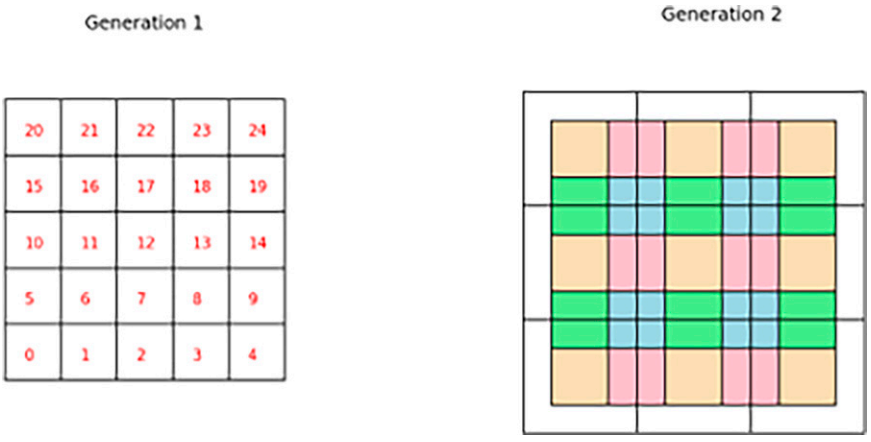

FIG. B1. Scheme used for location of higher generations on the cubed sphere. If number of processors on (left) the lower generation in one direction is odd, then we arrange processors of (right) the higher generation as shown. (See text for further clarification.)

used as the underlying grid geometry in the global atmospheric model of the Unified Forecasting System (http://ufs-dev.rap.ucar.edu/), finite volume model, FV3 (e.g., Putman and Lin 2007). The straight grid lines in a gnomonic projection map back to great circle arcs on the sphere, but there is some freedom to choose the profile of spacing between the grid lines in each family. An example of modeling the background error covariance directly on the cubed-sphere geometry is found in Song et al. (2017). The performance of the MG beta filter in DA for a global version of the FV3 will be presented and discussed somewhere else. Here, we just outline the approach used to apply the MG beta filter on this grid geometry.

Since the filtering can be performed on a slightly different version of the gnomonic family of cubed sphere grids from the one used for the final analysis increments, we can choose the computationally advantageous "equiangular" version as a grid of choice for filtering. The reason is that, when applying a beta filter, one must always supply a sufficiently large halo (whose extent depends on the spatial span of filter) and populate it with data from the surrounding processors. In the case of the gnomonic cubed sphere used in the FV3, the coordinate lines are arcs of great circles which are interrupted at the cube edges. Not only are the grid lines interrupted where they intercept the cube edges, but the smooth progression of the family of grid arcs that are approximately parallel to one edge will exhibit an interruption of the profile of spacing in passing from one side of this edge to the other. This necessitates that the process of matching the results of filtering spilling over from one side of the edge to the other would involve two-dimensional interpolations in the general case. However, as Ronchi et al. (1996) pointed out, using the equiangular version of the gnomonic grid, where this second kind of interruption is not present, one is able to perform the needed interpolations one-dimensionally, along those same grid arcs. Nevertheless, since the analysis grid is defined on an equal-along-edges version of the gnomonic cubed sphere grid (slightly different from the equiangular version), and at a higher resolution, we do need to remap (within the same processors), in fully two dimensions, between these two grids at the beginning and at the end of the filtering procedure.

The multigrid scheme for filtering used in the regional 3D RTMA extends the domains of higher generations, if the number of processors is odd, only in one (east or north) direction, which on the cubed sphere clearly will be inappropriate. Therefore, we came up with the following scheme: If the number of processors in each direction (N in x direction and M in y direction) of one generation is divisible by 2, the domain of the next generation stays the same and the number of processors in each direction is divided by 2, so that number of processors dealing with that next generation is $(N/2) \times (M/2)$. However, if the number of processors at one generation in, for example, the x direction, is not divisible by 2, the domain of the next generation will be slightly extended in that direction (with the extra domain populated with zeros), and the number of processors for this generation will be $[(N + 1)/2] \times M/2$. We illustrate this situation in Fig. B1.

In this case, generation-1 has $5 \times 5 = 25$ PEs, and next, generation-2, will have 9 PEs. Processors 0, 2, 4, 22, 24, 26, 44, 46, and 48 are, after adjoint interpolation to lower resolution, completely mapped into the middle portion of virtual processors of the next generation. Processors 1, 3, 23, 25, 45 and 47 are divided vertically; 11, 13, 15, 33, 35 and 37 are divided horizontally; and 12, 14, 34 and 36 are divided into four pieces.

REFERENCES

Abramowitz, M., and I. A. Stegun, 1972: *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover, 1046 pp.

Bannister, R. N., 2008a: A review of forecast error covariance statistics in atmospheric variational data assimilation. I: Characteristics and measurements of forecast error covariances. *Quart. J. Roy. Meteor. Soc.*, **134**, 1951–1970, https://doi.org/10.1002/qj.339.

——, 2008b: A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics. *Quart. J. Roy. Meteor. Soc.*, **134**, 1971–1996, https://doi.org/10.1002/qj.340.

Brandt, A., 1977: Multilevel adaptive solutions of boundary value problems. *Math. Comput.*, **31**, 333–390, https://doi.org/10.1090/S0025-5718-1977-0431719-X.

——, 1997: Multiscale algorithm for atmospheric data assimilation. *SIAM J. Sci. Comput.*, **18**, 949–956, https://doi.org/10.1137/S106482759528942X.

Carley, J. R., and Coauthors, 2020: A description of the v2.8 RTMA/URMA upgrade and progress toward 3D RTMA. *10th Conf. on Transition of Research to Operations*, Boston, MA, Amer. Meteor. Soc., 8A.3, https://ams.confex.com/ams/2020Annual/webprogram/Paper364378.html.

De Pondeca, M. S., and Coauthors, 2011: The real-time mesoscale analysis at NOAA's National Centers for Environmental Prediction: Current status and development. *Wea. Forecasting*, **26**, 593–612, https://doi.org/10.1175/WAF-D-10-05037.1.

Derber, J. C., and A. Rosati, 1989: A global ocean data assimilation system. *J. Phys. Oceanogr.*, **19**, 1333–1347, https://doi.org/10.1175/1520-0485(1989)019<1333:AGODAS>2.0.CO;2.

Fisher, M., 2003: Background error covariance modelling. *ECMWF Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean*, Reading, United Kingdom, ECMWF, 45–64.

Gaspari, G., and S. Cohn, 1999: Construction of correlation functions in two and three dimensions. *Quart. J. Roy. Meteor. Soc.*, **125**, 723–757, https://doi.org/10.1002/qj.49712555417.

Gill, P. E., W. Murray, and M. H. Wright, 1981: *Practical Optimization.* Academic Press, 401 pp.

Guillet, O., A. T. Weaver, X. Vasseur, Y. Michel, S. Gratton, and S. Gürol, 2019: Modelling spatially correlated observation errors in variational data assimilation using a diffusion operator on an unstructured mesh. *Quart. J. Roy. Meteor. Soc.*, **145**, 1947–1967, https://doi.org/10.1002/qj.3537.

Hackbusch, W., 2013: *Multi-Grid Methods and Applications.* Springer, 376 pp.

Hamming, R. W., 1989: *Digital Filters.* 3rd ed. Dover, 284 pp.

Hayden, C. M., and R. J. Purser, 1995: Recursive filter objective analysis of meteorological fields: Applications to NESDIS operational processing. *J. Appl. Meteor.*, **34**, 3–15, https://doi.org/10.1175/1520-0450-34.1.3.

Ide, K., P. Courtier, M. Ghil, and A. C. Lorenc, 1997: Unified notation for data assimilation: Operational, sequential and variational. *J. Meteor. Soc. Japan*, **75**, 181–189, https://doi.org/10.2151/jmsj1965.75.1B_181.

Kang, Y.-H., D. Y. Kwak, and K. Park, 2014: Multigrid methods for improving the variational data assimilation in numerical weather prediction. *Tellus*, **66A**, 20217, https://doi.org/10.3402/tellusa.v66.20217.

Li, W., Y. Xie, Z. He, G. Han, K. Liu, J. Ma, and D. Li, 2008: Application of the multigrid data assimilation scheme to the China Seas' temperature forecasting. *J. Atmos. Oceanic Technol.*, **25**, 2106–2116, https://doi.org/10.1175/2008JTECHO510.1.

——, ——, S.-M. Deng, and Q. Wang, 2010: Application of the multigrid method to the two-dimensional Doppler radar radial velocity data assimilation. *J. Atmos. Oceanic Technol.*, **27**, 319–332, https://doi.org/10.1175/2009JTECHA1271.1.

Mirouze, I., and A. T. Weaver, 2010: Representation of correlation functions in variational assimilation using an implicit diffusion operator. *Quart. J. Roy. Meteor. Soc.*, **136**, 1421–1443, https://doi.org/10.1002/qj.643.

Parrish, D. F., and J. C. Derber, 1992: The National Meteorological Center's Spectral Statistical-Interpolation analysis system. *Mon. Wea. Rev.*, **120**, 1747–1763, https://doi.org/10.1175/1520-0493(1992)120<1747:TNMCSS>2.0.CO;2.

Purser, R. J., 2020a: Description and some formal properties of beta filters: Compact support quasi-Gaussian convolution operators with applications to the construction of spatial covariances. NOAA/NCEP Office Note 498, 21 pp., https://doi.org/10.25923/qvfm-js76.

——, 2020b: A formulation of the Decad algorithm using the symmetries of the Galois field, GF(16). NOAA/NCEP Office Note 500, 28 pp., https://doi.org/10.25923/4nhy-x681.

——, 2020c: A formulation of the Hexad algorithm using the geometry of the Fano projective plane. NOAA/NCEP Office Note 499, 13 pp., https://doi.org/10.25923/xrzp-x016.

——, W.-S. Wu, D. F. Parrish, and N. M. Roberts, 2003a: Numerical aspects of the application of recursive filters to variational statistical analysis. Part I: Spatially homogeneous and isotropic Gaussian covariances. *Mon. Wea. Rev.*, **131**, 1524–1535, https://doi.org/10.1175//1520-0493(2003)131<1524:NAOTAO>2.0.CO;2.

——, ——, ——, and ——, 2003b: Numerical aspects of the application of recursive filters to variational statistical analysis. Part II: Spatially inhomogeneous and anisotropic general covariances. *Mon. Wea. Rev.*, **131**, 1536–1548, https://doi.org/10.1175//2543.1.

Putman, W., and S.-J. Lin, 2007: Finite-volume transport on various cubed-sphere grids. *J. Comput. Phys.*, **227**, 55–78, https://doi.org/10.1016/j.jcp.2007.07.022.

Rančić, M., M. Pondeca, R. J. Purser, and J. R. Carley, 2020: MPI re-decomposition and remapping algorithms used within a multigrid approach to modeling of the background error covariance for high-resolution data assimilation. *30th Conf. on Weather Analysis and Forecasting/26th Conf. on Numerical Weather Prediction*, Boston, MA, Amer. Meteor. Soc., J59.3, https://ams.confex.com/ams/2020Annual/webprogram/Paper363505.html.

Ronchi, C., R. Iacono, and P. S. Paolucci, 1996: The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.*, **124**, 93–114, https://doi.org/10.1006/jcph.1996.0047.

Sadourny, R., 1972: Conservative finite-differencing approximations of the primitive equations on quasi-uniform spherical grids. *Mon. Wea. Rev.*, **100**, 136–144, https://doi.org/10.1175/1520-0493(1972)100<0136:CFAOTP>2.3.CO;2.

Song, H.-J., I.-H. Kwon, and J. Kim, 2017: Characteristics of a spectral inverse of the Laplacian using spherical harmonic functions on a cubed-sphere grid for background error covariance modeling. *Mon. Wea. Rev.*, **145**, 307–322, https://doi.org/10.1175/MWR-D-16-0134.1.

Weaver, A., and P. Courtier, 2001: Correlation modelling on the sphere using a generalized diffusion equation. *Quart. J. Roy. Meteor. Soc.*, **127**, 1815–1846, https://doi.org/10.1002/qj.49712757518.

——, and I. Mirouze, 2013: On the diffusion equation and its application to isotropic and anisotropic correlation modelling in variational assimilation. *Quart. J. Roy. Meteor. Soc.*, **139**, 242–260, https://doi.org/10.1002/qj.1955.

Wu, W.-S., R. J. Purser, and D. F. Parrish, 2002: Three-dimensional variational analysis with spatially inhomogeneous covariances. *Mon. Wea. Rev.*, **130**, 2905–2916, https://doi.org/10.1175/1520-0493(2002)130<2905:TDVAWS>2.0.CO;2.

Zhang, H. Q., and X. J. Tian, 2018: Multigrid nonlinear least squares four-dimensional variational data assimilation scheme with the Advanced Research Weather Research and Forecasting Model. *J. Geophys. Res. Atmos.*, **123**, 5116–5129, https://doi.org/10.1029/2017JD027529.

——, ——, W. Cheng, and L. P. Jiang, 2020: System of multigrid nonlinear least-squares four-dimensional variational data assimilation for numerical weather prediction (SNAP): System formulation and preliminary evaluation. *Adv. Atmos. Sci.*, **37**, 1267–1284, https://doi.org/10.1007/s00376-020-9252-1.