PADOVA UNIVERSITY PRESS

Dolomites Research Notes on Approximation

Volume 19 · 2026 · Pages 31–47

Fast Anisotropic Gaussian Convolution in 3D-Var Data Assimilation

Salvatore Cuomo[a] · Raffaele Farina[a]

Communicated by Gabriele Santin

Abstract

This paper presents an efficient method for modeling horizontal covariances in three-dimensional variational data assimilation (3D-VAR) using a fast anisotropic Gaussian convolution. Unlike conventional isotropic Gaussian convolution, which assumes uniform spatial correlation scales, the proposed approach introduces an adaptive anisotropic diffusion tensor that accounts for spatial heterogeneities. The method is implemented through a non-orthogonal coordinate transformation, enabling an efficient recursive filtering approximation of anisotropic Gaussian convolution. By incorporating spatially varying anisotropic correlation structures, this approach improves background error covariance representation and enhances information propagation in data assimilation. Numerical experiments show that the method effectively captures directional dependencies while maintaining computational efficiency comparable to isotropic convolution. This advancement aligns 3D-VAR with four-dimensional variational assimilation (4D-VAR) by introducing flow-dependent anisotropic corrections within a computationally feasible framework.

Keywords: anisotropic Gaussian convolution, recursive filtering, data assimilation, anisotropic covariance modeling, anisotropic diffusion tensor.

1 Introduction

Data assimilation is a fundamental process in numerical weather prediction and oceanography, aimed at optimally combining observational data with model forecasts to improve state estimation. In three-dimensional variational assimilation (3D-VAR), the background error covariance matrix, **B**, governs how observational information propagates in space [1, 2]. Traditional implementations model horizontal covariances using isotropic Gaussian convolution, which assumes separable filtering along the canonical (x, y) grid axes with either uniform or spatially varying correlation scales, but fails to account for directional dependencies in geophysical flows [3]. Recursive filters and diffusion-based approaches have widely been employed to approximate Gaussian convolutions. One-dimensional recursive filters, applied sequentially along two perpendicular directions, have been introduced to approximate isotropic Gaussian convolutions [4]. In oceanography, explicit solutions of the diffusion equation have been used to model horizontal error covariances [2], while implicit integration techniques have later been developed to enhance numerical stability and computational efficiency [5]. Higher-order recursive filters have also been explored to improve accuracy with fewer iterations [6]. To overcome the limitations of isotropic covariance models, various studies have introduced anisotropy into spatial correlation structures [7, 8]. Building on these advances, this study presents an efficient method for modeling anisotropic horizontal covariances in 3D-VAR by leveraging a fast anisotropic Gaussian convolution approximated. The proposed approach extends isotropic methods by incorporating an adaptive anisotropic diffusion tensor, **T**, which locally adjusts to spatial heterogeneities. This ensures that the correlation structure aligns with dynamically relevant directions, improving the representation of background error covariances and enhancing the propagation of observational information. A key aspect of this method is the application of a non-orthogonal coordinate transformation, which diagonalizes the diffusion tensor and enables a computationally efficient approximation of anisotropic convolution along anisotropic directions through recursive filtering [9]. This transformation significantly reduces the complexity of direct convolution computations while preserving accuracy. Unlike previous studies where recursive filters were applied only in isotropic settings, the proposed method extends their application to anisotropic diffusion processes, leading to a more accurate representation of background error correlations in geophysical applications. Numerical experiments confirm that the anisotropic recursive filter effectively models horizontal covariances while

[a] University of Naples Federico II, Naples, Italy

maintaining computational efficiency comparable to isotropic filtering. Moreover, the iterative solvers used for 3D-Var scheme exhibit accelerated convergence under the proposed anisotropic modeling. The results demonstrate that the proposed method captures flow-dependent structures and spatially varying correlation scales, making it a viable alternative to traditional isotropic covariance models.

The remainder of this paper is structured as follows: Section 2 presents the mathematical formulation of 3D-VAR and covariance modeling. Section 3 introduces the efficient decomposition of anisotropic Gaussian kernels using a non-orthogonal coordinate transformation. Section 4 details the implementation of recursive filtering techniques for anisotropic convolution. Section 5 provides a comparative analysis of isotropic and anisotropic covariance modeling in 3D-VAR. Finally, Section 6 summarizes the findings and discusses potential extensions toward advanced data assimilation schemes.

2 Mathematical Foundations of 3D-VAR and Covariance Modeling

The 3D-VAR method addresses the problem of combining a background model state and observational data at a given time t_0 to produce an optimal estimate of the true state of a physical system at the same time t_0. It typically minimizes a cost function such as:

$$J(\mathbf{s}) = \frac{1}{2}(\mathbf{s}-\mathbf{s}_b)^T\mathbf{B}^{-1}(\mathbf{s}-\mathbf{s}_b) + \frac{1}{2}(\mathbf{y}-\mathcal{H}(\mathbf{s}))^T\mathbf{R}^{-1}(\mathbf{y}-\mathcal{H}(\mathbf{s})), \tag{1}$$

where $\mathbf{s}$ is the state vector, $\mathbf{s}_b$ is the background model state, $\mathbf{B}$ is the background error covariance matrix, $\mathbf{y}$ is the observation state, $\mathcal{H}$ is the observation operator, which maps the model state to the observation space, ensuring consistency between model predictions and observed data, and $\mathbf{R}$ is the observation error covariance matrix, assumed to be diagonal, reflecting the assumption that observation errors are uncorrelated between different measurements. [1, 11, 15].

Let the physical domain $\Omega \subset \mathbb{R}^3$ be discretized into a finite set of grid nodes

$$\mathcal{G} = \{\mathbf{x}_i \in \Omega : i = 1, \ldots, N\},$$

where N denotes the total number of spatial degrees of freedom, i.e., the dimension of the state space. The discrete state vector is then

$$\mathbf{s} = \big(s(\mathbf{x}_1), s(\mathbf{x}_2), \ldots, s(\mathbf{x}_N)\big)^T \in \mathbb{R}^N,$$

and the background error covariance matrix $\mathbf{B} \in \mathbb{R}^{N\times N}$ operates on this space.

In the particular case of a structured Cartesian grid with uniform resolution $(\Delta x, \Delta y, \Delta z)$, one has $N = N_x \cdot N_y \cdot N_z$. More generally, when the discretization is irregular with variable grid spacing and possibly non-uniform element geometry, the indexing $\{\mathbf{x}_i\}_{i=1}^N$ is arbitrary and the above formulation remains valid, since it depends only on the total number of degrees of freedom N and not on the mesh topology. In this framework, the grid spacing parameters can be interpreted as spatially varying mesh scales $\Delta x(\mathbf{x}_i), \Delta y(\mathbf{x}_i), \Delta z(\mathbf{x}_i)$, allowing Equation (1) to be consistently defined on variable-resolution grids.

For simplicity and clarity in the following discussion, we assume a regular grid with uniform resolution. By linearizing $\mathcal{H}(\mathbf{s})$ around $\mathbf{s}_b$, the cost function in 1 becomes quadratic and convex [4], ensuring the existence of a unique global minimum. The linearized observation operator is expressed as:

$$\mathcal{H}(\mathbf{s}) \approx \mathcal{H}(\mathbf{s}_b) + \mathbf{H}(\mathbf{s}-\mathbf{s}_b),$$

where $\mathbf{H} \in \mathbb{R}^{M\times N}$ is the Jacobian of $\mathcal{H}$ evaluated at $\mathbf{s}_b$, and M denotes the dimension of the observation space. Taking the gradient of $J(\mathbf{s})$ and setting it to zero leads to the following preconditioned linear system:

$$(\mathbf{I} + \mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\delta\mathbf{s} = \mathbf{B}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}, \tag{2}$$

where $\delta\mathbf{s} = \mathbf{s} - \mathbf{s}_b \in \mathbb{R}^N$ represents the correction to the background model state, and $\mathbf{d} = \mathbf{y} - \mathcal{H}(\mathbf{s}_b) \in \mathbb{R}^M$ is the misfit between the model-predicted data and the observed data. By applying the Woodbury matrix identity, the solution of system 2 can be written explicitly as

$$\delta\mathbf{s} \;=\; \mathbf{B}\mathbf{H}^T\big(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R}\big)^{-1}\mathbf{d}, \tag{3}$$

which coincides with the linear Kalman update [16]. However, in large-scale geophysical applications such as OceanVar [11], NEMOVAR [17], the Met Office global system [18], and more recently JEDI-MPAS [15], the number of observations is often comparable to the dimension of the state space ($M \approx N$). In such cases, the matrix $\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R}$ is prohibitively large to invert explicitly. Moreover, the background covariance matrix $\mathbf{B}$ is never assembled or stored explicitly, due to its dimensionality and memory requirements.

For these reasons, the system in 2 is efficiently solved using iterative methods, which require repeated matrix-vector products with the large covariance matrix $\mathbf{B}$, about $O(N^2)$ operations at each iteration of the minimizer. To deal with this computational challenge, efficient techniques are employed to approximate these products, often by representing $\mathbf{B}$ as $\mathbf{B} = \mathbf{V}^T\mathbf{V}$, ensuring that the background error covariance matrix remains symmetric and positive semi-definite.

The matrix $\mathbf{V}$ is in turn split into a sequence of linear operators [2], where each captures a specific physical aspect of the considered system. For example, in oceanographic problems, such as OceanVar [11], the matrix $\mathbf{V}$ is defined as follows:

$$\mathbf{V} = \mathbf{V}_D \mathbf{V}_{uv} \mathbf{V}_g \mathbf{V}_H \mathbf{V}_V, \tag{4}$$

In Equation 4, the linear operator $\mathbf{V}_V$ transforms coefficients which multiply vertical EOFs into vertical profiles of temperature and salinity defined at the model vertical levels. $\mathbf{V}_H$ applies horizontal covariances on fields of temperature and salinity, $\mathbf{V}_g$ calculates the sea surface height error covariance from three-dimensional fields of temperature and salinity, $\mathbf{V}_{uv}$ calculates velocity from sea surface height, temperature, and salinity, and $\mathbf{V}_D$ applies a divergence damping filter on the velocity field.

In 3D-VAR data assimilation, the background-error covariance operator for horizontal correlations $\mathbf{V}_H$, acts as a smoothing operator. It spreads the background error across neighboring grid points, imposing realistic horizontal spatial correlations on the analysis variables. Mathematically, under isotropic assumptions, i.e., the same correlation structure in all horizontal directions, $\mathbf{V}_H$ can be represented as a discrete convolution with a Gaussian kernel of a given correlation length scale σ. Specifically, a common choice for the covariance between two horizontal grid points $\mathbf{x}_i$ and $\mathbf{x}_j$ (at a fixed vertical level $z = z_k$) is a Gaussian function:

$$G_H(i,j) \;=\; g(\mathbf{x}_i - \mathbf{x}_j;\, \sigma) \;=\; \frac{1}{2\pi\,\sigma^2}\,\exp\!\left(-\,\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\,\sigma^2}\right), \tag{5}$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|$ is the horizontal Euclidean distance between the two points. In practice, applying $\mathbf{V}_H$ with this kernel corresponds to replacing the error at each grid point with a weighted average of the surrounding points, with weights defined by the Gaussian distribution (nearby points receive higher weight, and distant points negligible weight). This Gaussian smoothing effect (see Figure 1a) ensures that the background errors vary smoothly in space, as is physically expected.

However, forming $\mathbf{V}_H$ as an explicit matrix based on the kernel above is computationally infeasible for large systems. If there are $N_x \times N_y$ horizontal grid points, then at each vertical level the matrix G_H is of size $(N_x N_y) \times (N_x N_y)$. When extended over N_z vertical levels (often assuming separability so that horizontal correlations are applied independently at each level), $\mathbf{V}_H$ becomes block-diagonal in the vertical and the full operator has size $N \times N$ with $N = N_x N_y N_z$. A direct application of this operator would require $O(N^2)$ operations, which is prohibitive in large-scale geophysical applications. Instead, in practice $\mathbf{V}_H$ is never assembled explicitly as a dense matrix. The Gaussian smoothing is applied iteratively through efficient approximations, such as sequential recursive filters [10, 12, 6] or implicit diffusion solvers [3, 5, 13, 14], which reduce the computational cost from $O(N^2)$ to $O(N)$ operations.

However, isotropic covariance models, despite being computationally efficient, fail to capture directional dependencies that are essential in complex flow systems such as ocean currents or atmospheric jet streams [3]. In reality, background error correlations can be elongated in one direction (e.g. along a coastline or a jet stream) and narrower in the perpendicular direction. Recent studies, such as [14] on flow-dependent anisotropic diffusion filters, demonstrate that anisotropy can be efficiently represented and handled within the diffusion-based framework.

To overcome the limitation of isotropy, $\mathbf{V}_H$ is extended to model anisotropic horizontal covariances by using an oriented Gaussian convolution, represented by the following horizontal kernel:

$$G_H(i,j) \;=\; g(\mathbf{x}_i - \mathbf{x}_j;\, \sigma_u,\, \sigma_v,\, \theta) = \frac{1}{2\pi\sigma_u\sigma_v}\,\exp\left(-\tfrac{1}{2}\,(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{T}^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right), \tag{6}$$

where $\mathbf{T}$ is the diffusion (covariance) tensor that defines the anisotropic Gaussian shape. In practice, one constructs:

$$\mathbf{T} = \hat{\mathbf{R}}\,\mathbf{D}_{uv}\,\hat{\mathbf{R}}^T, \qquad \text{where} \quad \hat{\mathbf{R}} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{7}$$

is the rotation matrix that rotates coordinates by an angle θ with respect to the original x- and y-axes, defining new principal axes u and v, i.e., the change of basis from the (u,v)-system to the original (x,y)-system, where the anisotropy manifests (Figure 1b). The diagonal matrix $\mathbf{D}_{uv} = \mathrm{diag}(\sigma_u^2,\, \sigma_v^2)$ contains the squared correlation lengths (variances) along the rotated u- and v-directions.

This anisotropic Gaussian convolution formulation allows the background-error covariance to stretch or contract differently in horizontal directions, aligning the smoothing effect with known physical structures. For example, by choosing θ to follow the dominant flow direction, $\mathbf{V}_H$ will spread errors farther along the flow and less across it, yielding more realistic error covariances aligned with geophysical dynamics. The diffusion tensor $\mathbf{T}$ can also account for non-uniform grid spacing (dx and dy) by incorporating the grid metrics into σ_u and σ_v. In summary, extending $\mathbf{V}_H$ to an anisotropic Gaussian convolution enables the 3D-VAR system to more accurately capture spatial error variability by aligning the covariance kernel with key physical characteristics, such as dominant flow directions and environmental gradients, while still benefiting from the efficient convolution-based implementation of the covariance operator.

3 Efficient Decomposition of Anisotropic Gaussian Kernels for Computational Convolution

The objective in this paper is to realize the anisotropic Gaussian convolution in (6) by means of fast recursive Gaussian filters; to this end, we employ the Infinite Impulse Response (IIR) approximation to Gaussian smoothing [10], in particular its accurate third-order formulations successfully adopted in 3D-VAR systems [6]. The natural eigenframe (u, v), aligned with the principal axes of anisotropy (Figure 1b), is not computationally convenient on a Cartesian mesh: the principal directions are typically oblique with respect to the grid, so each filtering sweep along u and v would, at every horizontal level, require off-grid evaluations with repeated one-dimensional linear (or bilinear on the Cartesian mesh) interpolations at non-integer locations in both causal and anti-causal passes, increasing computational cost and introducing additional numerical smoothing. In the classical (x, y)-orthogonal system, the convolution kernel in Equation 6 is not separable due to the non-diagonal nature of the anisotropic diffusion tensor $\mathbf{T}$. This lack of separability along the x- and y-axes prevents the direct application of recursive filters [10]. Then, following [9], we therefore introduce an oblique change of variables (x, t) $(t \neq u)$ (Figure 1c) via an LDL^T factorization of $\mathbf{T}^{-1}$ such that the convolution kernel in Equation 6 became separable along x- and t-axes and preserves the x-axis (a shear), so that one pass is grid-aligned (along x) and only one pass is oblique (along t). This choice is particularly suited to the present work, where we address the case of spatially varying anisotropy. In such a setting, it is advantageous to define a reference frame in which at least one axis (here, the x-axis) remains fixed and aligned with the computational grid, while the other axis (t) adapts locally to the varying orientation of anisotropy. In this way, the filtering procedure retains computational efficiency by exploiting the grid alignment in one direction, while still capturing the local variability of anisotropic structures along the oblique direction. In the (x, y)-orthogonal system, the t-axis is defined by $t: \; y = \tan(\phi)\, x$ with $\phi \neq 0$ (see Figure 1c) and, in general, $\phi \neq \theta$, hence $t \neq u$. We seek a linear change of coordinates that diagonalizes the quadratic form $\mathbf{x}^{\top}\mathbf{T}^{-1}\mathbf{x}$ and decouples it into independent contributions along two axes. This is achieved by a change-of-basis matrix $\mathbf{L}^T$ such that

$$\tilde{\mathbf{x}} = \mathbf{L}^T\mathbf{x}, \qquad \tilde{\mathbf{x}}^{\top} = (x, t), \quad \mathbf{x}^{\top} = (x, y).$$

Under this mapping, the quadratic form has to diagonalize and decouple as:

$$\mathbf{x}^{\top}\mathbf{T}^{-1}\mathbf{x} = \mathbf{x}^{\top}\mathbf{L}\mathbf{D}_{xt}^{-1}\mathbf{L}^{\top}\mathbf{x} = \tilde{\mathbf{x}}^{\top}\mathbf{D}_{xt}^{-1}\tilde{\mathbf{x}} = \frac{x^2}{\sigma_x^2} + \frac{t^2}{\sigma_t^2},$$

where $\mathbf{D}_{xt}^{-1} = \mathrm{diag}\big(1/\sigma_x^2,\, 1/\sigma_t^2\big)$. This decoupling is realized by applying an $LDL^{\top}$ factorization to $\mathbf{T}^{-1}$, which yields the diagonal matrix

$$\mathbf{D}_{xt}^{-1} = \begin{bmatrix} \dfrac{\sin^2\theta}{\sigma_v^2} + \dfrac{\cos^2\theta}{\sigma_u^2} & 0 \\ 0 & \dfrac{1}{\sigma_u^2\sin^2\theta + \sigma_v^2\cos^2\theta} \end{bmatrix},$$

Consequently, the effective variances along the x– and t–axes are

$$\sigma_x^2 = \frac{\sigma_u^2\sigma_v^2}{\sigma_u^2\sin^2\theta + \sigma_v^2\cos^2\theta}, \qquad \sigma_t^2 = \sigma_u^2\sin^2\theta + \sigma_v^2\cos^2\theta. \tag{8}$$

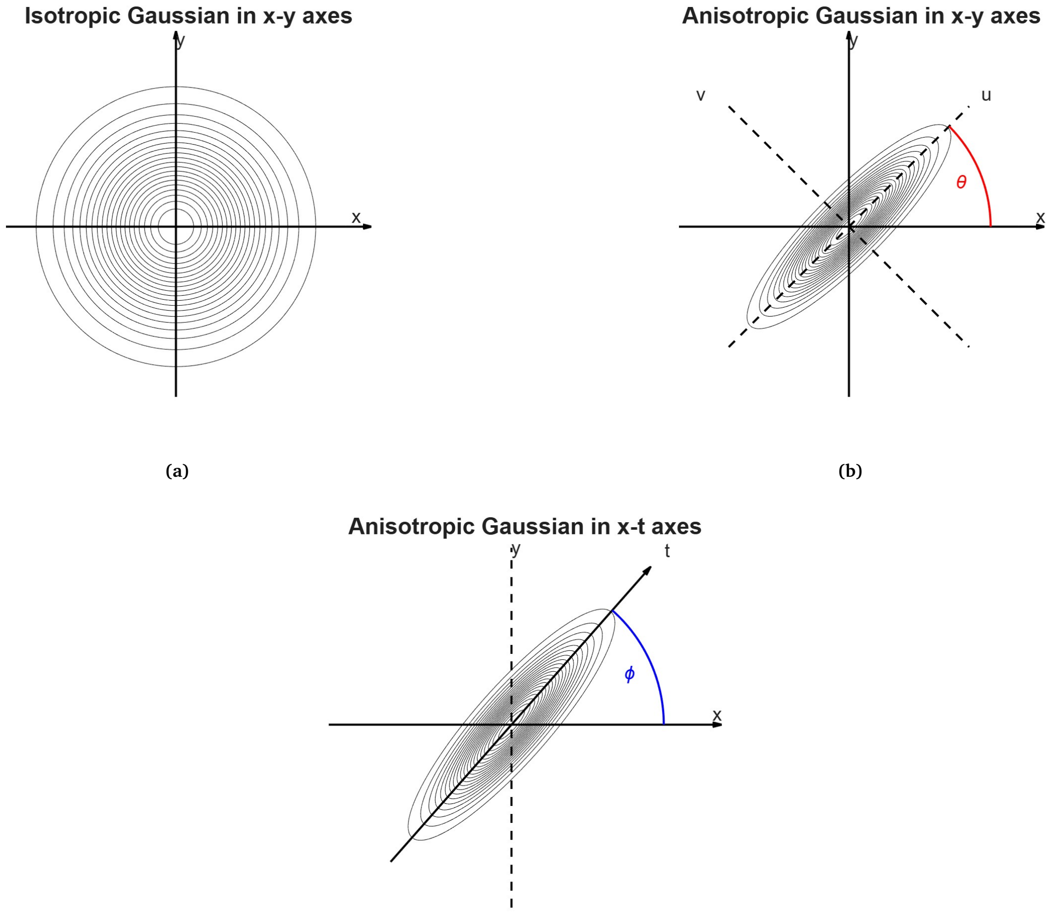
The determinant of $\mathbf{T}^{-1}$ is preserved under the change of reference frame, hence, $\sigma_x^2\sigma_t^2 = \sigma_u^2\sigma_v^2$. Since the matrix $\mathbf{L}^{-T}$ encodes the coordinates of the x- and t-axes in the x-y reference system, we can determine the angle ϕ, which defines the inclination of the t-axis relative to the x-axis (as illustrated in Figure 1c). To achieve this, we compute the matrix $\mathbf{L}^{-T}$ and derive the corresponding expression for ϕ:

$$\mathbf{L}^{-T} = \begin{bmatrix} 1 & \dfrac{(\sigma_u^2 - \sigma_v^2)\cos(\theta)\sin(\theta)}{\sigma_v^2\cos^2(\theta) + \sigma_u^2\sin^2(\theta)} \\ 0 & 1 \end{bmatrix}, \qquad \phi = \arctan\left(\frac{\sigma_v^2\cos^2(\theta) + \sigma_u^2\sin^2(\theta)}{(\sigma_u^2 - \sigma_v^2)\cos(\theta)\sin(\theta)}\right), \tag{9}$$

where $\sigma_u \neq \sigma_v$ and $\theta \notin \{0, \pi/2, \pi, 3\pi/2\}$. Then, we obtain $\mathbf{T}^{-1} = \mathbf{L}\mathbf{D}_{\mathbf{xt}}^{-1}\mathbf{L}^T$and the Gaussian function in the x-t coordinate system can be represented in the following decoupled, quadratic form:

$$g(\tilde{\mathbf{x}}, \sigma_x, \sigma_t, \phi) = \frac{1}{2\pi\sigma_x\sigma_t}\exp\left(-\frac{1}{2}\tilde{\mathbf{x}}^T\mathbf{D}_{xt}^{-1}\tilde{\mathbf{x}}\right), \tag{10}$$

This approach ensures an accurate representation of anisotropic diffusion orientation in the transformed x-t coordinate system and the decomposition of the anisotropic Gaussian kernel into independent components, enabling an efficient approximation of Gaussian convolution starting from arbitrary orientations.


Isotropic Gaussian in x-y axes
y
x
Anisotropic Gaussian in x-y axes
y
v
u
θ
x
Anisotropic Gaussian in x-t axes
y
t
ϕ
x

(a)

(b)

(c)

Figure 1: Contour line representations of Gaussian functions: (a) isotropic Gaussian function in the x-y coordinate system, (b) anisotropic Gaussian function in the x-y coordinate system, and (c) anisotropic Gaussian function in the x-t coordinate system.

4 Fast Recursive Filter for Anisotropic Gaussian Convolution with Spatially Varying Coefficients

Isotropic convolution in the 3D-VAR method for modeling horizontal covariances has been efficiently approximated using first- and third-order IIR recursive Gaussian filters [11, 6]. In this study, we extend this approach to the anisotropic case, where the degree of anisotropy is not uniform but may vary from point to point across the grid. This spatial variability prevents the use of a single global filter: instead, a local filtering strategy is required.

A key advantage of the (x, t) system, defined in Section 3, is that the x-axis remains the same for all points of the domain, while the t-axis may change locally depending on the anisotropy tensor. Consequently, along the x-axis the filtering is always grid-aligned, and only the recursive filter coefficients vary from point to point, since they depend on the local standard deviation $\sigma_x(i, j)$ (see Equation 8). In contrast, the filtering along the t-axis must be performed locally because both the orientation $\phi(i, j)$ (see Equation 9) and the effective scale $\sigma_t(i, j)$ (see Equation 8) vary spatially.

The recursive filter is applied to a discretized grid of $N_x \times N_y$ points at each vertical level $z = z_k$, $k = 1, \ldots, N_z$, independently. This is because the horizontal covariance operator $\mathbf{V}_H$ (defined in Section 2) is block–diagonal with respect to the vertical index. To keep the notation light, we adopt a purely two–dimensional notation. Consequently, the two 1-D filtering passes are performed level-wise: along the x-axis the filter is applied globally (since the axis is common to all grid points and only the coefficients depend on the local variance $\sigma_x(i, j)$), while along the t-axis the filter must be applied locally, because both the orientation $\phi(i, j)$ and the effective scale $\sigma_t(i, j)$ vary from point to point.

4.1 Filtering along the x-axis

Specifically, the x-axis is aligned with the horizontal grid, whereas the t-axis is inclined at a local angle $\phi(i,j)$ relative to the x-axis, as defined in Equation 9. Along the x-axis, the filtering with local standard deviation $\sigma_x(i,j)$ acts in two stages: first, a forward (causal) filtering step is performed for each $j = 1, \ldots, N_y$, progressing from $i = 1$ to $i = N_x$. Subsequently, a backward (anti-causal) filtering step is applied in reverse, starting from $i = N_x$ and proceeding back to $i = 1$. The recursive relationships that govern these procedures [6, 10] are expressed in Box 1.

BOX 1 — Third-order recursive Gaussian filter along the x-axis $(\forall\, j = 1, \ldots, N_y)$

Forward boundary initialization. Define $p_x(1)$, $p_x(2)$, $p_x(3)$ according to Equation (11) in Subsection 4.4, using the local coefficients $\alpha_1(i,j), \alpha_2(i,j), \alpha_3(i,j), \beta(i,j)$, $i = 1, 2, 3$ evaluated at the corresponding grid points.
Forward (causal) pass.

$$p_x(i) = \beta(i,j)s_{0,x}(i,j) + \alpha_1(i,j)p_x(i-1) + \alpha_2(i,j)p_x(i-2) + \alpha_3(i,j)p_x(i-3), \qquad i = 4, \ldots, N_x.$$

Backward boundary initialization. Define $s_x(N_x)$, $s_x(N_x+1)$, $s_x(N_x+2)$ by the exact backward initialization as described in Subsection 4.4 Equation (19), again with local coefficients evaluated at the point (N_x, j).
Backward (anti-causal) pass.

$$s_x(i) = \beta(i,j)p_x(i) + \alpha_1(i,j)s_x(i+1) + \alpha_2(i,j)s_x(i+2) + \alpha_3(i,j)s_x(i+3), \qquad i = N_x - 1, \ldots, 1.$$

Output mapping.

$$s_{f,x}(i,j) = s_x(i), \qquad i = 1, \ldots, N_x.$$

Remark. The coefficients $\alpha_1(i,j), \alpha_2(i,j), \alpha_3(i,j)$ and $\beta(i,j)$ are computed pointwise from Box 3 in Subsection 4.3, using the local standard deviation $\sigma_x(i,j)$ (see Equation 8).

In Box 1, $s_{0,x}(i,j)$ is the input function, $p_x(i)$ is the forward-filtered function, $s_x(i)$ is the backward-filtered function, and $s_{f,x}(i,j)$ represents the result of the filtering along the x-axis, mapped back to the grid at position (i,j). The coefficients $\alpha_1(i,j), \alpha_2(i,j), \alpha_3(i,j)$ and $\beta(i,j)$ in Box 1 are computed from Box 3 in Subsection 4.3 as functions of the local variance $\sigma_x(i,j)$, following the stable IIR formulations proposed in [10] and successfully applied in 3D-VAR systems [6].

4.2 Filtering along the t-axis

Before describing the implementation of the recursive filter along the t-axis, it is important to emphasize why this operation is performed locally rather than globally. In this work we address the case of spatially varying anisotropy, where both the orientation and the strength of anisotropy may change from point to point across the domain. In such a situation, applying a single global filter along t would not be consistent, since the oblique direction that best captures anisotropy is not uniform. Instead, a local filtering strategy is adopted: at each grid point $(\bar{i}, \bar{j})$, the oblique axis t is defined according to the local anisotropy tensor, ensuring that the smoothing procedure follows the spatial variability of the field. This makes the convolution both computationally feasible and physically consistent, aligning the filtering operation with the dominant local anisotropic structures.

For these reasons, filtering along the t-axis is carried out in the following way. At every grid point $(\bar{i}, \bar{j})$ on the grid of size $N_x \times N_y$, the recursive Gaussian filter is applied to a discrete interval $I_{(\bar{i},\bar{j})}$, centered at $(\bar{i}, \bar{j})$, with a radius of $3\lfloor \sigma_t(\bar{i}, \bar{j}) \rfloor$, where $\sigma_t(\bar{i}, \bar{j})$ is the standard deviation derived in Equation 8, and $\lfloor \cdot \rfloor$ denotes the integer part of a real number. The filtering operation at each point $(\bar{i}, \bar{j})$ of the $N_x \times N_y$ grid is restricted only to an interval of approximately $[-3\sigma_t(\bar{i}, \bar{j}), +3\sigma_t(\bar{i}, \bar{j})]$ along the t-direction, since within this range the Gaussian kernel contains over 99% of its probability mass. This ensures that the smoothing effect of the filter is both accurate and computationally efficient, as contributions outside this interval are negligible.

The filtering is performed along the local oblique line $\bar{t} : y - \bar{j} = \tan(\phi(\bar{i}, \bar{j}))(x - \bar{i})$, so that both the scale and the orientation adapt to the anisotropy at the given grid point. The procedure consists of two stages: first, a forward (causal) filtering step is performed, progressing from $j = 1$ to $j = 6\lfloor \sigma_t(\bar{i}, \bar{j}) \rfloor + 1$. Subsequently, a backward (anti-causal) filtering step is applied in reverse, starting from $j = 6\lfloor \sigma_t(\bar{i}, \bar{j}) \rfloor + 1$ and proceeding back to $j = 1$ (see Figure 2). The recursive relationships governing these procedures [9] are expressed in Box 2.

In Box 2, the term $s_{f,x}\big(x(j), y(j)\big)$ represents values obtained by linear interpolation between the two nearest grid points along the x-axis at the ordinate $y(j)$ (see Figure 2), for each $j = 1, \ldots, 6\lfloor \sigma_t(\bar{i}, \bar{j}) \rfloor + 1$, provided that $(x(j), y(j))$ lies within the grid boundaries; otherwise, it is set to zero. The functions $p_t(j)$ and $s_t(j)$ denote the intermediate results of the forward and backward filtering operations, respectively. The final output $s_{f,t}\big(x(j), y(j)\big)$ maps the filtered values back onto the grid using the same interpolation scheme applied along the x-axis, and these contributions are then accumulated with the values previously computed at the neighboring points.

BOX 2 — Third-order recursive Gaussian filter along the t-axis $(\forall\,(\bar{i},\bar{j})$ on the $N_x \times N_y$ grid)

Sampling coordinates along the local t-line. For a given center point $(\bar{i},\bar{j})$ and local parameters $\sigma_t(\bar{i},\bar{j})$, $\phi(\bar{i},\bar{j})$ (Equations 8–9), we parametrize

$$\bar{t}:\; y-\bar{j}=\tan\big(\phi(\bar{i},\bar{j})\big)\big(x-\bar{i}\big)$$

over $j=1,\dots,6\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+1$ as

$$\boxed{\begin{aligned} x(j) &= \frac{j-\big(3\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+1\big)}{\tan\big(\phi(\bar{i},\bar{j})\big)}+\bar{i},\\ y(j) &= j+\bar{j}-\big(3\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+1\big).\end{aligned}}$$

Sampling & interpolation at $(x(j),y(j))$. The input samples $s_{f,x}(x(j),y(j))$ are obtained by *horizontal two-point linear interpolation* at the ordinate $y(j)$ (see Figure 2):

$$i_0=\lfloor x(j)\rfloor,\qquad w=x(j)-i_0,\qquad s_{f,x}(x(j),y(j))=(1-w)s_{f,x}(i_0,\,y(j))+w\,s_{f,x}(i_0+1,\,y(j)).$$

If the sampling point $(x(j),y(j))$ lies outside the grid, we use zero-padding: both the sample $s_{f,x}(x(j),y(j))$ and the local variance $\sigma_t(x(j),y(j))$ (used to compute the filter coefficients from Box 3 in Subsection 4.3) are set to zero.

Forward boundary initialization. Define $p_t(1),p_t(2),p_t(3)$ as described in Subsection 4.4 Equation (11), using local coefficients $\alpha_1(x(j),y(j)),\alpha_2(x(j),y(j)),\alpha_3(x(j),y(j))$ and $\beta(x(j),y(j))$ evaluated at $j=1,2,3$.
Forward (causal) pass.

$$\begin{aligned} p_t(j) &= \beta(x(j),y(j))s_{f,x}\big(x(j),y(j)\big)\\ &\quad+\alpha_1(x(j),y(j))\,p_t(j-1)+\alpha_2(x(j),y(j))\,p_t(j-2)+\alpha_3(x(j),y(j))\,p_t(j-3),\\ &\qquad\qquad j=4,\dots,6\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+1.\end{aligned}$$

Backward boundary initialization. Define $s_t(6\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+1)$, $s_t(6\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+2)$, $s_t(6\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+3)$ as in Subsection 4.4 Equation (19), with the local coefficients evaluated at the point $6\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+1$.
Backward (anti-causal) pass.

$$\begin{aligned} s_t(j) &= \beta(x(j),y(j))\,p_t(j)\\ &\quad+\alpha_1(x(j),y(j))s_t(j+1)+\alpha_2(x(j),y(j))s_t(j+2)+\alpha_3(x(j),y(j))s_t(j+3),\\ &\qquad\qquad j=6\lfloor\sigma_t(\bar{i},\bar{j})\rfloor,\dots,1.\end{aligned}$$

Output mapping.

$$s_{f,t}\big(x(j),y(j)\big)=s_t(j),\qquad j=1,\dots,6\lfloor\sigma_t(\bar{i},\bar{j})\rfloor+1.$$

Remark. The coefficients $\alpha_1(x(j),y(j)),\alpha_2(x(j),y(j)),\alpha_3(x(j),y(j))$ and $\beta(x(j),y(j))$ are computed from Box 3 in Subsection 4.3, using the local scale $\sigma_t(x(j),y(j))$ (Equation 8).

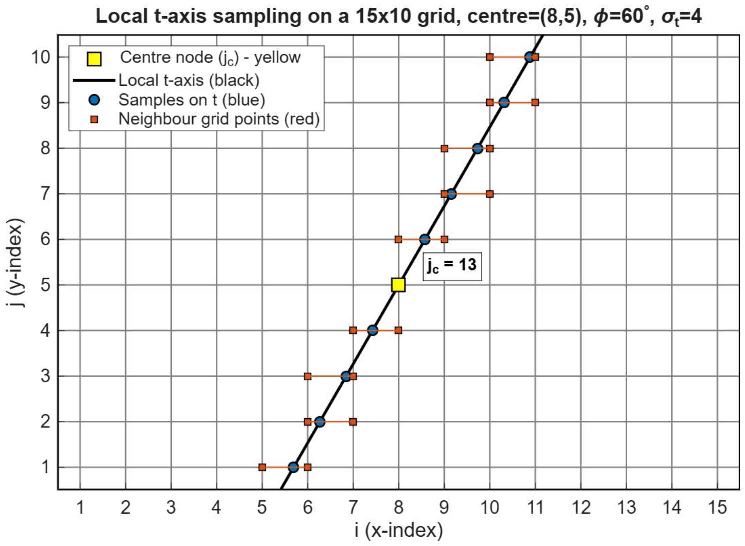


Figure 2: Local oblique filtering scheme on a 15 × 10 grid with center $(\bar{i},\bar{j})=(8,5)$, orientation $\phi=60°$, and scale $\sigma_t=4$. Black line: oblique t-axis; blue circles: sampling points $(x(j),y(j))$ for $j=1,\dots,25$ along $[-3\sigma_t,+3\sigma_t]$; yellow square: central node ($j_c=13$); red squares: neighbouring grid nodes for linear interpolation.

4.3 Computation of filter coefficients and stability analysis

Following [10], the recursive Gaussian filter introduced in Boxes 1–2 is realised as a cascade of a *causal* IIR filter (forward pass) and an *anti–causal* IIR filter (backward pass). For the causal transfer function to be Bounded-Input Bounded-Output (BIBO) stable, its poles must lie strictly inside the unit circle in the complex z-plane, whereas for the anti-causal transfer function to be BIBO-stable, its poles must lie strictly outside the unit circle.
In our formulation, the coefficients are *spatially variable*:

$$\alpha_m(i,j),\ \beta(i,j) \text{ for the } x\text{–axis pass}, \qquad \alpha_m(x(j),y(j)),\ \beta(x(j),y(j)) \text{ for the } t\text{–axis pass},\ m=1,2,3.$$

These coefficients depend pointwise on the local standard deviations $\sigma_x(i,j)$ and $\sigma_t(x(j),y(j))$, and are computed according to the analytical formulation reported in [6] and summarised in Box 3. The stability criterion is satisfied locally: at every grid location, the causal filter retains all poles strictly within $|z|<1$, whereas the anti–causal filter has all poles strictly within $|z|>1$ in the complex z-plane [10]. This spectral separation ensures that the combined causal–anti–causal process remains BIBO–stable under arbitrary spatial variations of the correlation length scales, providing a stable and accurate recursive approximation of the anisotropic Gaussian convolution.

BOX 3 — Recursive filter coefficients as a function of the standard deviation σ

$$\alpha_1=\frac{a_1}{a_0},\quad \alpha_2=\frac{a_2}{a_0},\quad \alpha_3=\frac{a_3}{a_0},\quad \beta=\sqrt[4]{2\pi\sigma^2}\,(1-\alpha_1-\alpha_2-\alpha_3),$$

$$\begin{aligned}
a_0 &= 3.738128+5.788982\,\sigma+3.382473\,\sigma^2+1.000000\,\sigma^3,\\
a_1 &= 5.788982\,\sigma+6.764946\,\sigma^2+3.000000\,\sigma^3,\\
a_2 &= -3.382473\,\sigma^2-3.000000\,\sigma^3,\\
a_3 &= 1.000000\,\sigma^3.
\end{aligned}$$

Note. The spatial index (i,j) is omitted from σ for readability; in practice, all coefficients are computed pointwise from the local standard deviation $\sigma_x(i,j)$ or $\sigma_t(x(j),y(j))$.

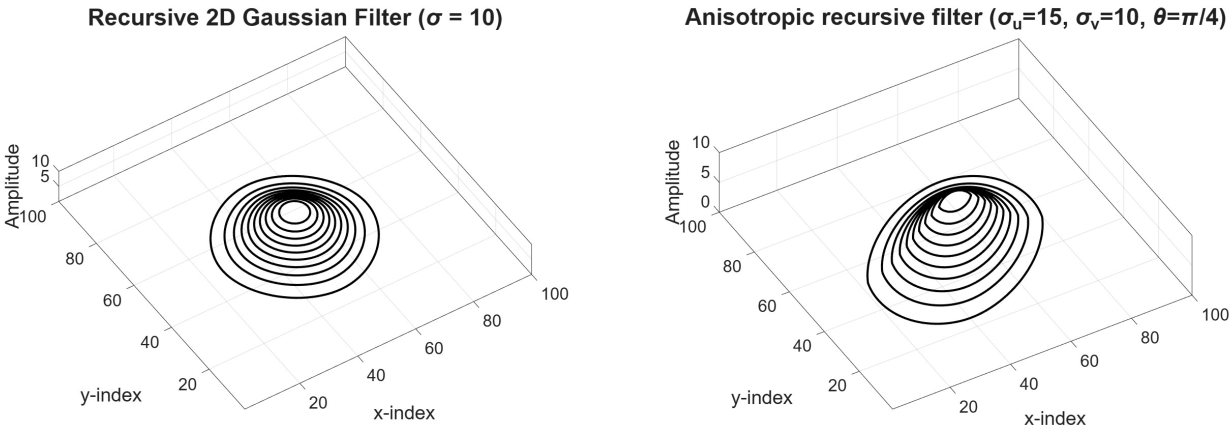
To illustrate the behaviour of the isotropic and the anisotropic recursive Gaussian filters introduced in Boxes 1–2, we now present two representative examples on a 100 × 100 grid with a Dirac impulse located at $(i_0,j_0)=(50,50)$. Both filters are implemented using the third–order causal/anti–causal formulation derived in [6, 10], with coefficients defined in Box 3.
In the isotropic case (Figure 3a), the standard deviation $\sigma=10$ is constant and the diffusion tensor reduces to $\sigma^2\mathbf{I}$. The two one–dimensional recursive passes (Box 1 along x, followed by the symmetric pass along y) produce a separable and radially symmetric Gaussian smoothing. The resulting level sets are circular and concentric, confirming the isotropic nature of the convolution.
In contrast, the anisotropic case (Figure 3b), where the diffusion tensor is expressed as $\mathbf{T}=\hat{\mathbf{R}}\mathbf{D}_{uv}\hat{\mathbf{R}}^T$ (see Section 2), uses spatially constant but unequal principal standard deviations $(\sigma_u,\sigma_v)=(15,10)$ and an orientation $\theta=\pi/4$ where the anisotropy manifests. Here, the filtering along the oblique axis t (Box 2) operates in a rotated coordinate frame (x,t), with coefficients locally adapted according to Equations 8–9. The output exhibits elongated and tilted level sets—elliptical contours aligned with the diffusion tensor—which demonstrate the directional smoothing induced by anisotropy. Together, these examples visually validate the theoretical formulation of the isotropic and anisotropic recursive filters.

4.4 Exact boundary conditions for third-order recursive filters

Incorrect initial conditions for the backward procedures in the recursive Gaussian filter cause significant amplitude and phase (geometric position) distortion for all points within about three standard deviations from the last point of the right boundary. To address this issue, we implement appropriate boundary conditions for the filters in Box 1 and Box 2 [19], which correct these distortions in recursive filtering.
The boundary conditions for the forward procedures in Box 1 and Box 2 are derived in [10] under the assumption of a stationary input signal to the left of the filtering interval of length N. They are formulated as:

$$\begin{bmatrix} p(1)\\ p(2)\\ p(3)\end{bmatrix}=\begin{bmatrix}\beta(1) & 0 & 0\\ 0 & \beta(2) & 0\\ 0 & 0 & \beta(3)\end{bmatrix}\begin{bmatrix} s_0(1)\\ s_0(2)\\ s_0(3)\end{bmatrix}+\begin{bmatrix}0 & 0 & 0\\ \alpha_1(2) & 0 & 0\\ \alpha_2(3) & \alpha_1(3) & 0\end{bmatrix}\begin{bmatrix} p(1)\\ p(2)\\ p(3)\end{bmatrix}, \tag{11}$$

where s_0 denotes the input to the filter and p represents the intermediate result of the forward (causal) filtering step. The coefficients $\alpha_k(i)$ for $k=1,2,3$ and $\beta(i)$ (with spatial index j omitted for readability) are position-dependent filter coefficients corresponding to the point being calculated, as specified in Box 3 (Subsection 4.3).



(a) **(b)**

Figure 3: Comparison of the recursive Gaussian filtering in the isotropic and anisotropic cases on a 100 × 100 grid with a Dirac delta at $(50, 50)$: (a) isotropic case ($\sigma = 10$) with circular, concentric level sets showing radially symmetric smoothing, and (b) anisotropic case with $(\sigma_u, \sigma_v) = (10, 5)$ and $\theta = \pi/4$ showing elliptical, rotated level sets. In (b), Boxes 1–2 implement a separable scheme in the (x, t) frame, producing oriented elliptical smoothing governed by $(\sigma_u, \sigma_v, \theta)$.

To eliminate amplitude and phase distortions near the right boundary, the backward initial conditions $s(N)$, $s(N+1)$, $s(N+2)$ are set using the exact linear correction method proposed in [19], assuming here that the input signal s_0 is stationary and equal to zero to the right of the filtering interval.
The method extends the forward–backward recursion to $i \to \infty$. For the third-order filter, the forward state evolves as

$$\begin{bmatrix} p(i) \\ p(i-1) \\ p(i-2) \end{bmatrix} = \mathbf{A} \begin{bmatrix} p(i-1) \\ p(i-2) \\ p(i-3) \end{bmatrix} = \mathbf{A}^{i-N} \begin{bmatrix} p(N) \\ p(N-1) \\ p(N-2) \end{bmatrix}, \qquad \mathbf{A} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad i > N, \tag{12}$$

and the backward state as

$$\begin{bmatrix} s(i) \\ s(i+1) \\ s(i+2) \end{bmatrix} = \mathbf{B} \begin{bmatrix} p(i) \\ p(i-1) \\ p(i-2) \end{bmatrix} + \mathbf{A} \begin{bmatrix} s(i+1) \\ s(i+2) \\ s(i+3) \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad i \geq N. \tag{13}$$

Combining Equations (12) and (13), we have

$$\begin{bmatrix} s(N) \\ s(N+1) \\ s(N+2) \end{bmatrix} = \left(\sum_{i=0}^{\infty} \mathbf{A}^i \mathbf{B} \mathbf{A}^i \right) \begin{bmatrix} p(N) \\ p(N-1) \\ p(N-2) \end{bmatrix}. \tag{14}$$

We need to calculate the 3 × 3 matrix $\mathbf{M} \equiv \sum_{i=0}^{\infty} \mathbf{A}^i \mathbf{B} \mathbf{A}^i$ that links the initial backward state $\begin{bmatrix} s(N) \\ s(N+1) \\ s(N+2) \end{bmatrix}$ to the final forward state $\begin{bmatrix} p(N) \\ p(N-1) \\ p(N-2) \end{bmatrix}$. By **M**'s recursive definition,

$$\mathbf{M} = \mathbf{B} + \mathbf{A}\mathbf{M}\mathbf{A}, \tag{15}$$

solving this 9 × 9 linear system (15) for **M** (see [19]) yields the exact boundary correction for the third-order recursive filter. The transition matrix **M** is explicitly defined by:

$$\mathbf{M} = \frac{1}{D} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}, \tag{16}$$

where the matrix elements are given by:

$$\begin{aligned}
m_{11} &= \alpha_1\alpha_3\beta + \alpha_2\beta + \alpha_3^2\beta - \beta,\\
m_{12} &= -\alpha_1^2\alpha_3\beta - \alpha_1\alpha_2\beta - \alpha_1\alpha_3^2\beta - \alpha_2\alpha_3\beta,\\
m_{13} &= -\alpha_1\alpha_3\beta - \alpha_2\alpha_3^2\beta,\\
m_{21} &= -\alpha_1\beta - \alpha_2\alpha_3\beta,\\
m_{22} &= \alpha_1\alpha_2\alpha_3\beta - \alpha_1\alpha_3\beta + \alpha_2^2\beta - \alpha_2\beta,\\
m_{23} &= \alpha_1\alpha_3^2\beta + \alpha_2\alpha_3\beta + \alpha_3^3\beta - \alpha_3\beta,\\
m_{31} &= -\alpha_1^2\beta - \alpha_1\alpha_3\beta + \alpha_2^2\beta - \alpha_2\beta,\\
m_{32} &= -\alpha_1\alpha_2\beta + \alpha_1\alpha_3^2\beta - \alpha_2^2\alpha_3\beta + \alpha_2\alpha_3\beta + \alpha_3^3\beta - \alpha_3\beta,\\
m_{33} &= -\alpha_1\alpha_3\beta - \alpha_2\alpha_3^2\beta.
\end{aligned} \tag{17}$$

The denominator D in Equation (16) is given by:

$$\begin{aligned}
D = \alpha_1^3\alpha_3 + \alpha_1^2\alpha_2 + \alpha_1^2\alpha_3^2 + \alpha_1^2 - \alpha_1\alpha_2^2\alpha_3 + 4\alpha_1\alpha_2\alpha_3 - \alpha_1\alpha_3^3 + \alpha_1\alpha_3\\
- \alpha_2^3 + \alpha_2^2\alpha_3^2 + \alpha_2^2 - \alpha_2\alpha_3^2 + \alpha_2 - \alpha_3^4 + 2\alpha_3^2 - 1.
\end{aligned} \tag{18}$$

The coefficients $\beta, \alpha_1, \alpha_2, \alpha_3$ appearing in Equations (17) and (18) are evaluated at the last grid point (N, j), i.e., the rightmost point where the signal exists before extending to zero, using the formulation provided in Box 3 with the local standard deviation at that point.

The final forward-filtered values $p(N)$, $p(N-1)$ and $p(N-2)$ are used to compute the initial state of the backward (anti-causal) pass through the transition matrix **M**, such that:

$$\begin{bmatrix} s(N) \\ s(N+1) \\ s(N+2) \end{bmatrix} = \mathbf{M} \begin{bmatrix} p(N) \\ p(N-1) \\ p(N-2) \end{bmatrix}. \tag{19}$$

This implementation ensures numerical stability and avoids loss of convergence in iterative solvers used for solving Equation 2 [12].

4.5 Computational complexity analysis

The computational efficiency of the anisotropic filtering approach is a critical factor in its practical applicability to large-scale data assimilation systems. While the anisotropic Gaussian convolution is naturally expressed in the principal (u, v) eigenframe, where the diffusion tensor is diagonal and the directional correlation scales are explicitly defined, implementing the filter in this coordinate system would require two oblique (sheared) filtering passes one along each principal axis both incurring interpolation costs at every grid point. To reduce computational overhead while preserving anisotropic structure, we adopt the non-orthogonal (x, t) factorization introduced in Section 3, which requires only one oblique pass (along t) combined with a single grid-aligned pass (along x). This section quantifies the computational cost of this strategy and compares it against both the isotropic Gaussian filter (the standard baseline in operational 3D-VAR systems) and the alternative (u, v) implementation.

From a computational standpoint, we consider an anisotropic third-order recursive Gaussian filter whose per-sample update has constant cost $c_f \simeq 14$ flops (including both causal and anti-causal passes). The filtering along the x-axis is applied over the entire grid of size $N_x \times N_y$ without truncation. In contrast, the filtering along the oblique t-axis is performed locally at each grid point (i, j) and truncated to an interval of approximately $6\lfloor\sigma_t(i, j)\rfloor$ samples centered at that point, since the t-direction varies spatially according to the local anisotropy. Within this $\pm 3\sigma_t(i, j)$ range, the Gaussian kernel contains over 99% of its probability mass, while outside this interval the filtering effects are negligible. When a filtering pass is aligned with the grid (e.g., along x) no interpolation is required; when a pass is oblique (sheared along t) we include a linear gather+scatter overhead c_{int} per sample per oblique pass (in our implementation $c_{\text{int}} \simeq 6$ flops).

To assess computational efficiency, we compare three possible filtering configurations: (i) isotropic filtering with two grid-aligned passes along x and y, (ii) anisotropic filtering in the non-orthogonal (x, t) frame with one grid-aligned and one oblique pass, and (iii) anisotropic filtering in the principal (u, v) frame with two oblique passes. Denoting by $N = N_x N_y$ the number of horizontal grid points and by $\bar{\sigma}_t$, $\bar{\sigma}_u$, $\bar{\sigma}_v$ the spatial means of the effective standard deviations along the oblique axis t and along the principal axes (u, v), the three configurations behave as follows.

(i) Isotropic (two grid-aligned passes). Both passes are aligned with the grid, so the cost is simply the sum of two updates and scales as

$$C_{\text{iso}} \approx 2\, c_f\, N \ \ \text{(about } 28N \text{ flops)}.$$

(ii) Anisotropic in the non-orthogonal (x, t) frame (one grid-aligned + one oblique pass). The pass along x is grid-aligned and costs $c_f N$. The oblique pass along t touches, on average, $6\bar{\sigma}_t$ samples per node, so its filtering part contributes $6\bar{\sigma}_t\, c_f\, N$; the linear gather+scatter overhead for this single oblique pass adds $6\bar{\sigma}_t\, c_{\text{int}}\, N$. In compact form, the total cost is

$$C_{\text{aniso},(x,t)} \approx \big[(1+6\bar{\sigma}_t)\,c_f + 6\bar{\sigma}_t\, c_{\text{int}}\big]N,$$

which is $\mathcal{O}(N)$ with a multiplicative constant proportional to the effective oblique support $6\bar{\sigma}_t$.

(iii) Anisotropic in the principal (u,v) frame (two oblique passes). Here both passes are oblique. The filtering parts contribute $6\bar{\sigma}_u\, c_f\, N$ and $6\bar{\sigma}_v\, c_f\, N$, and the linear gather+scatter overhead is paid twice (once per oblique pass), adding $6\bar{\sigma}_u\, c_{\text{int}}\, N + 6\bar{\sigma}_v\, c_{\text{int}}\, N$. The resulting cost is

$$C_{\text{aniso},(u,v)} \approx \big[6\,(\bar{\sigma}_u+\bar{\sigma}_v)\,(c_f + c_{\text{int}})\big]N.$$

which is $\mathcal{O}(N)$ with a multiplicative constant proportional to the effective sum of the oblique supports $6\bar{\sigma}_u$ and $6\bar{\sigma}_v$.

As general remark, all three costs are linear in N and proportional to the effective stencil lengths. We now compare the alternatives.

Comparison: anisotropic (x,t) vs isotropic. A convenient indicator is

$$\frac{C_{\text{aniso},(x,t)}}{C_{\text{iso}}} = \frac{(1+6\,\bar{\sigma}_t)\,c_f + 6\,\bar{\sigma}_t\, c_{\text{int}}}{2\,c_f}.$$

With c_f=14 and c_{int}=6 this gives

$$\frac{C_{\text{aniso},(x,t)}}{C_{\text{iso}}} \simeq \frac{14+84\,\bar{\sigma}_t+36\,\bar{\sigma}_t}{28} = \frac{14+120\,\bar{\sigma}_t}{28} = \frac{1}{2}+\frac{30}{7}\,\bar{\sigma}_t,$$

showing that the extra cost scales with the effective oblique support $6\,\bar{\sigma}_t$: one oblique pass (plus one grid-aligned) is more expensive than two grid-aligned passes by a factor that grows approximately as $\frac{30}{7}\,\bar{\sigma}_t \approx 4.3\,\bar{\sigma}_t$ relative to C_{iso}.

Comparison: anisotropic (u,v) vs anisotropic (x,t). For comparable scales $\bar{\sigma}_u \approx \bar{\sigma}_v \approx \bar{\sigma}_t$, the ratio

$$\frac{C_{\text{aniso},(u,v)}}{C_{\text{aniso},(x,t)}} = \frac{6(\bar{\sigma}_u+\bar{\sigma}_v)\,(c_f+c_{\text{int}})}{(1+6\bar{\sigma}_t)\,c_f + 6\bar{\sigma}_t\, c_{\text{int}}} = \frac{6(\bar{\sigma}_u+\bar{\sigma}_v)\,(c_f+c_{\text{int}})}{c_f + 6\bar{\sigma}_t\,(c_f+c_{\text{int}})}.$$

With c_f=14, c_{int}=6, and assuming $\bar{\sigma}_u=\bar{\sigma}_v=\bar{\sigma}_t=\bar{\sigma}$, this simplifies to

$$\frac{C_{\text{aniso},(u,v)}}{C_{\text{aniso},(x,t)}} = \frac{12\bar{\sigma}\cdot 20}{14+6\bar{\sigma}\cdot 20} = \frac{240\bar{\sigma}}{14+120\bar{\sigma}} \xrightarrow{\bar{\sigma}\gg 1} 2.$$

Intuitively, the (u,v) formulation requires two oblique passes (and thus pays two interpolation overheads), whereas the (x,t) formulation uses only one oblique pass plus a cheap grid-aligned pass, yielding approximately half the cost for large $\bar{\sigma}$.

5 Comparative Analysis of Isotropic and Anisotropic Horizontal Covariances in 3D-VAR

This section evaluates the performance of isotropic versus anisotropic horizontal covariance modeling in the 3D-VAR framework through numerical experiments on a synthetic Black Sea application. Subsection 5.1 describes the experimental setup, including the construction of the synthetic dataset, the mathematical formulation of the reduced variational problem, and the numerical implementation of both isotropic and anisotropic Gaussian filtering schemes. Subsection 5.2 presents the comparative results in terms of conjugate-gradient convergence and relative accuracy metrics, demonstrating the benefits of flow-aligned anisotropic covariance modeling.

5.1 Experimental Setup and Numerical Implementation

In this subsection, we analyze the optimal estimate $\mathbf{s}_b + \delta\mathbf{s}$, where $\delta\mathbf{s}$ is the correction to the background state obtained as the solution to the linear system in Equation 2, comparing the performance of isotropic and anisotropic convolutions in modeling horizontal covariances.

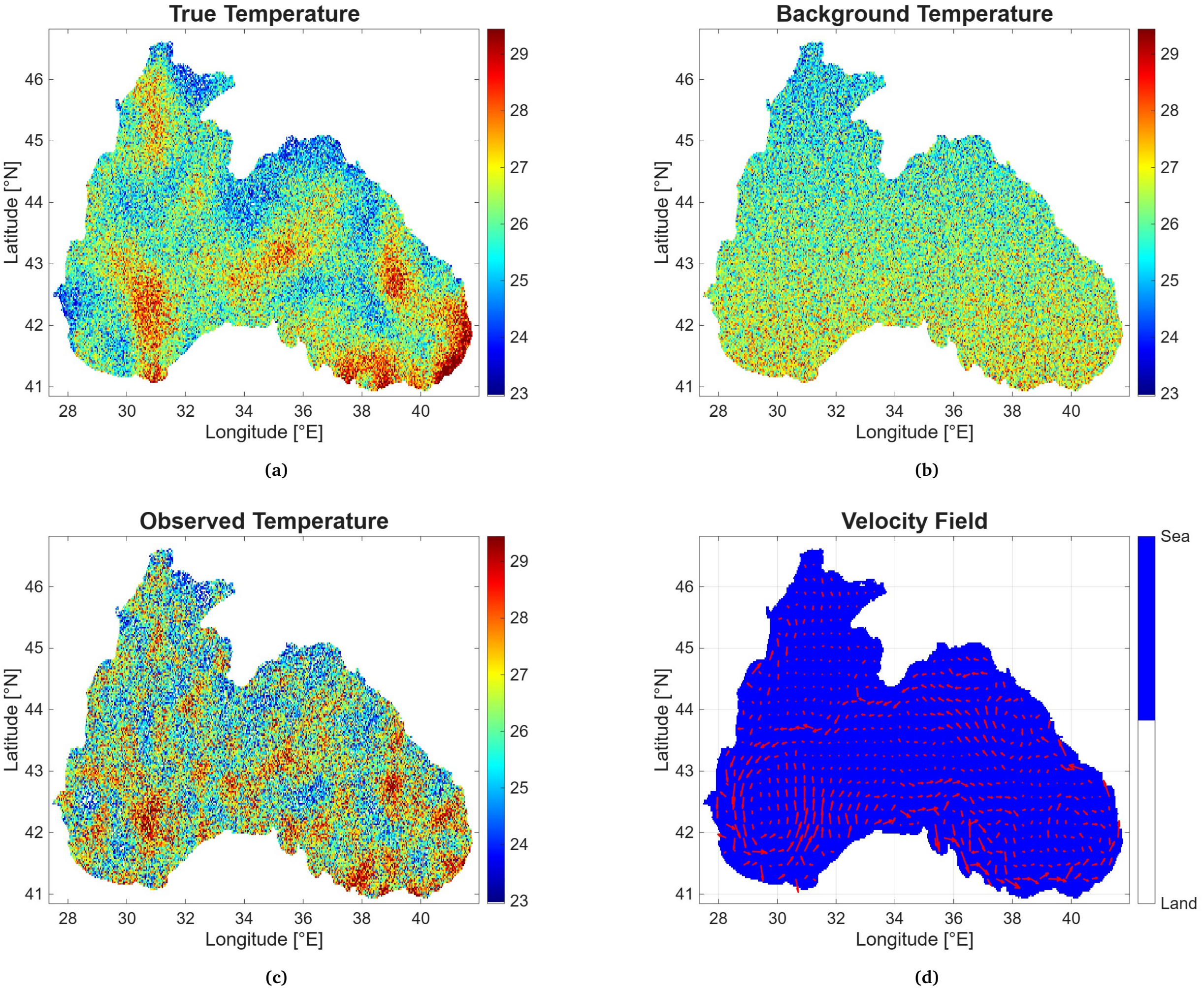
The tests are performed on a synthetic application over the Black Sea domain, designed to provide a controlled yet realistic environment for assessing the behavior of the convolution filters. The geographical grid and the surface velocity fields are extracted from CMEMS (Copernicus Marine Environment Monitoring Service) for 11 July 2015 (surface layer). The temperature fields used in the test refer to synthetic sea-surface temperature (SST) values generated on this grid.

We consider three sea-surface temperature (SST) fields. The true state $\mathbf{s}_{\text{true}}$ at time t_0 (Figure 4a) is constructed as a smooth field with mean 27 °C—matching the basin-wide average SST over the Black Sea on 11 July 2015—superimposed on a weak meridional (north–south) gradient and small-amplitude, zero-mean random perturbations with standard deviation 1 °C. To imprint dynamically consistent structure, we superimpose flow-aligned mesoscale anomalies oriented along the CMEMS surface-velocity directions (cf. Figure 4d). These anomalies are generated via anisotropic Gaussian smoothing of white noise, elongated along the local current and scaled by its magnitude, and we add a small high-frequency component (≈ 10% of the anomaly amplitude) to introduce fine-scale variability.

The background field $\mathbf{s}_b$ (Figure 4b) represents the model state at t_0. It is obtained from $\mathbf{s}_{\text{true}}$ by removing the flow-aligned mesoscale anomalies (and the fine-scale component), yielding a field that retains only the large-scale mean and weak meridional gradient. This construction mimics a typical background lacking mesoscale variability.

The observation field $\mathbf{y}$ (Figure 4c) is generated by $\mathbf{s}_{\text{true}}$, adding independent zero-mean Gaussian noise with variance σ_o^2 (i.e., $\mathbf{R} = \sigma_o^2\mathbf{I}$), consistent with the error model adopted in the assimilation experiments.

In Figure 4c, we display the pointwise CMEMS surface velocity field $\mathbf{v}$ over the Black Sea using a thinned quiver plot to avoid visual clutter from small-scale variability. Arrows indicate the local flow direction, sampled on a regular subgrid in index space, with land points masked. This representation preserves the native pointwise information while remaining legible at figure scale. The resulting dataset background, noisy observations and velocity field, provides the initial inputs for the variational assimilation test.



(a)

(b)

(c)

(d)

Figure 4: Synthetic dataset over the Black Sea ($N_x \times N_y = 395 \times 295$): (a) true temperature field $\mathbf{s}_{\text{true}}$, (b) background temperature $\mathbf{s}_b$ obtained by removing flow-aligned mesoscale anomalies from $\mathbf{s}_{\text{true}}$, (c) observed temperature $\mathbf{y} = \mathbf{s}_{\text{true}} + \varepsilon$ with independent and identically distributed noise ($\sigma_o = 1.25\,^\circ\text{C}$), and (d) CMEMS surface velocity field with arrows showing flow direction. Land points are set to `NaN` in (a–c) and masked in (d); all temperature maps use identical color scales.

In this subsection the goal is to resolve the linear system in Equation 2 under the following assumptions: the background-error covariance, modelling horizontal correlations only, is written $\mathbf{B} = \mathbf{V}_H^T\mathbf{V}_H$, where $\mathbf{V}_H$ is a (isotropic or anisotropic) Gaussian convolution operator and $\mathbf{V}_H^T$ its adjoint and $\mathbf{s}_b$ and $\mathbf{y}$ are defined on the same grid and represent the same physical variable, so the observation operator $\mathcal{H}$ (Section 2) and its Jacobian $\mathbf{H}$ reduce to the identity. Under these assumptions, Equation 2 reduces to

$$(\mathbf{I} + \mathbf{B}\mathbf{R}^{-1})\,\delta\mathbf{s} \;=\; \mathbf{B}\mathbf{R}^{-1}\mathbf{d}, \qquad \mathbf{d} = \mathbf{y} - \mathbf{s}_b, \tag{20}$$

and, since $\mathbf{B}$ and $\mathbf{R}^{-1} = (1/\sigma_o^2)\mathbf{I}$ are symmetric and semi-definite and definite positive respectively, the matrix $\mathbf{I} + \mathbf{B}\mathbf{R}^{-1}$ is symmetric positive definite (SPD). Consequently, an approximation of δs is computed by the Conjugate Gradient (CG) method [20] in the

Algorithm 1 Conjugate Gradient for $(\mathbf{I}+\mathbf{B}\mathbf{R}^{-1})\,\delta\mathbf{s}=\mathbf{B}\mathbf{R}^{-1}\mathbf{d}$ with $\mathbf{B}=\mathbf{V}_H^T\mathbf{V}_H$

```
Input: Background state $\mathbf{s}_b$; observations $\mathbf{y}$ (same grid); diffusion tensor $\mathbf{T}$ (isotropic or anisotropic case). scalar observation
    precision $\mathbf{R}^{-1}=\frac{1}{\sigma_o^2}\mathbf{I}$; tolerance tol; maximum iterations kmax.
Misfit: $\mathbf{d}\leftarrow\mathbf{y}-\mathbf{s}_b$.
Right-hand side: $\mathbf{b}\leftarrow\mathbf{V}_H^T\mathbf{V}_H(\mathbf{R}^{-1}\mathbf{d})$        ▷ Gaussian convolutions with tensor $\mathbf{T}$
Initialization: $\delta\mathbf{s}_0\leftarrow\mathbf{0},\quad \mathbf{r}_0\leftarrow\mathbf{b},\quad \mathbf{p}_0\leftarrow\mathbf{r}_0$.
for $k=0,1,2,\ldots,$ until kmax do
    $\mathbf{q}_k\leftarrow\mathbf{p}_k+\mathbf{V}_H^T\mathbf{V}_H(\mathbf{R}^{-1}\mathbf{p}_k)$        ▷ Gaussian convolution with tensor $\mathbf{T}$
    $\alpha_k\leftarrow\dfrac{\mathbf{r}_k^\top\mathbf{r}_k}{\mathbf{p}_k^\top\mathbf{q}_k}$
    $\delta\mathbf{s}_{k+1}\leftarrow\delta\mathbf{s}_k+\alpha_k\,\mathbf{p}_k$
    $\mathbf{r}_{k+1}\leftarrow\mathbf{r}_k-\alpha_k\,\mathbf{q}_k$
    if $\|\mathbf{r}_{k+1}\|_2<$ tol then
        break
    end if
    $\beta_k\leftarrow\dfrac{\mathbf{r}_{k+1}^\top\mathbf{r}_{k+1}}{\mathbf{r}_k^\top\mathbf{r}_k}$
    $\mathbf{p}_{k+1}\leftarrow\mathbf{r}_{k+1}+\beta_k\,\mathbf{p}_k$
end for
Output: correction $\delta\mathbf{s}$;   final estimate $\mathbf{s}_b+\delta\mathbf{s}$.
```

Krylov subspace:

$$\mathcal{K}_K\big((\mathbf{I}+\mathbf{B}\mathbf{R}^{-1}),\,\mathbf{B}\mathbf{R}^{-1}\mathbf{d}\big)=\operatorname{span}\big\{\mathbf{B}\mathbf{R}^{-1}\mathbf{d},\,(\mathbf{I}+\mathbf{B}\mathbf{R}^{-1})\mathbf{B}\mathbf{R}^{-1}\mathbf{d},\,(\mathbf{I}+\mathbf{B}\mathbf{R}^{-1})^2\mathbf{B}\mathbf{R}^{-1}\mathbf{d},\,\ldots,\,(\mathbf{I}+\mathbf{B}\mathbf{R}^{-1})^{K-1}\mathbf{B}\mathbf{R}^{-1}\mathbf{d}\big\},\quad K\le N.$$

In practice, matrix–vector products with $\mathbf{B}\mathbf{R}^{-1}$ are evaluated as $\mathbf{V}_H^T\mathbf{V}_H(\mathbf{R}^{-1}\cdot)$. Isotropic $\mathbf{V}_H$ is efficiently realized by high-order recursive Gaussian filters (RF) [10, 6, 11]; the anisotropic construction follows the non-orthogonal factorization and the filtering strategy developed in Sections 3–4.

Algorithm 1 presents the conjugate–gradient (CG) solver specialized for Equation 20. Inputs: background state $\mathbf{s}_b$, observations $\mathbf{y}$, scalar observation precision $\mathbf{R}^{-1}=\sigma_o^{-2}\mathbf{I}$, and a diffusion tensor $\mathbf{T}$ that parameterizes the Gaussian preconditioner used to approximate the background–error covariance action $\mathbf{B}$ at each iteration. The solver also takes the tolerance tol and the maximum iteration count kmax. Outputs: the analysis increment $\delta\mathbf{s}$ and the final estimate $\mathbf{s}_b+\delta\mathbf{s}$. The CG iteration stops when $\|\mathbf{r}\|_2\le$ tol or when kmax is reached.

From a numerical standpoint, the isotropic reference adopts a grid-aligned metric. Let $\mathbf{v}(i,j)=\big(v_x(i,j),\,v_y(i,j)\big)^\top$ denote the pointwise horizontal velocity (see Figure 4d) ; we set

$$\mathbf{T}_{\text{iso}}(i,j)=\begin{bmatrix}\sigma_x^2(i,j) & 0\\ 0 & \sigma_y^2(i,j)\end{bmatrix},\qquad \sigma_x(i,j)=\sigma\left(1+\gamma\,\frac{|v_x(i,j)|}{\|v_x\|_{\max}}\right),\quad \sigma_y(i,j)=\sigma\left(1+\gamma\,\frac{|v_y(i,j)|}{\|v_y\|_{\max}}\right),$$

with $\|v_x\|_{\max}=\max_{(i,j)}|v_x(i,j)|$ and $\|v_y\|_{\max}=\max_{(i,j)}|v_y(i,j)|$, i.e., the maximum absolute values over the grid under examination. Here, $\sigma\in\mathbb{R}^+$ is the base correlation length scale, and $\gamma\in\mathbb{R}^+$ is a dimensionless parameter that modulates the velocity-dependent scaling of the horizontal covariances. This yields an isotropic-in-orientation yet scale-varying Gaussian diffusion aligned with the grid directions (x,y). As in [10, 6], the horizontal covariance operator $\mathbf{B}$ is realized by a third-order causal/anti-causal recursive Gaussian filter and its adjoint (transpose), a standard practice in oceanographic 3D-VAR [11, 12]. With a positive-definite observation-error precision $\mathbf{R}^{-1}=(1/\sigma_o^2)\mathbf{I}$, the normal operator $\mathbf{A}=\mathbf{I}+\mathbf{B}\mathbf{R}^{-1}$ is SPD, enabling CG convergence.

The anisotropic reference orients diffusion along the local flow. Let $\theta_{ij}=\operatorname{atan}\big(v_y(i,j),v_x(i,j)\big)$ be the flow direction. In the natural eigenframe (u,v) (see Figure 1b), we set

$$\mathbf{T}_{ij}=\hat{\mathbf{R}}(\theta_{ij})\,\mathbf{D}_{uv}(i,j)\,\hat{\mathbf{R}}(\theta_{ij})^\top,\qquad \mathbf{D}_{uv}(i,j)=\operatorname{diag}\big(\sigma_u^2(i,j),\sigma_v^2(i,j)\big),$$

with

$$\sigma_u(i,j)=\sigma\left(1+\gamma\,\frac{\|\mathbf{v}(i,j)\|}{\|\mathbf{v}\|_{\max}}\right),\qquad \sigma_v(i,j)=\sigma,$$

where $\sigma\in\mathbb{R}^+$ is the base correlation length and $\gamma\ge 0$ is a dimensionless parameter that modulates the along/across–flow correlation-length ratio: for $\gamma=0$ the model tends to a rotated isotropic filter, while for $\gamma>0$ it elongates covariances proportionally to the local speed. As detailed in Sections 3–4, the associated Gaussian convolution operator $\mathbf{V}_H$ and its transposed $\mathbf{V}_H^T$ are implemented in a local, non-orthogonal (x,t) system (fixed x; oblique t with inclination $\phi(i,j)$) by a pair of stable

1D recursive filters (and adjoints) along x and t, which preserve the spatial variability of the metric while keeping the matrix $\mathbf{A} = \mathbf{I} + \mathbf{B}\mathbf{R}^{-1}$ SPD.

In this paper, all experiments use $\sigma = 15$ pixels and $\gamma \in \{3,4,5\}$ on the Black Sea domain; $\sigma = 15$ was chosen so that the background horizontal correlation radius is a small fraction (about $\frac{1}{20}$) of the shortest grid dimension—a pragmatic tuning in variational oceanography that spreads increments at mesoscale while avoiding domain-scale correlations [3, 11].

5.2 Results and Discussion

Having established the experimental setup and numerical framework, we now evaluate the comparative performance of isotropic and anisotropic covariance modeling through convergence analysis and accuracy metrics.

The comparative behaviour of the isotropic and anisotropic variants of the Gaussian recursive preconditioner was investigated through a series of experiments performed over the CMEMS Black Sea domain. Both configurations were embedded in the Conjugate–Gradient (CG) solver for Equation 20, with the same convergence tolerance ($\texttt{tol} = 10^{-6}$) and maximum iteration count ($\texttt{kmax} = 100$). All experiments started from the same background temperature field ((Figure 4b) and used identical observational perturbations (Figure. 4c). The isotropic baseline adopts a velocity-informed but grid-aligned covariance metric, with correlation scales that vary spatially along the computational (x, y) axes according to the local velocity components, allowing the filter to respond to flow intensity while preserving axis alignment. For the anisotropic configurations, the horizontal-covariance metric was flow-informed, with local orientation and stretching derived from the surface-velocity field shown in Figure 4d.

We contrast a velocity-informed isotropic baseline, in which grid-aligned scales (σ_x, σ_y) vary pointwise with the velocity components (v_x, v_y) on the (x, y) frame, with a flow-aligned anisotropic filter, where principal scales (σ_u, σ_v) depend on $\|\mathbf{v}\|$ in the (u, v) eigenframe and are mapped to the local (x, t) coordinates. Convergence is evaluated from the CG residual histories (see Figure 5), while skill is quantified by relative L^2 errors (see Figure 6) of the optimal estimate $\mathbf{s}_b + \delta\mathbf{s}$ (see Figure 7) against the true field on sea points (see Figure 4a).

The anisotropic preconditioner attains the tolerance 10^{-6} in $\sim$ 14–17 iterations for $\gamma = 3$–5, whereas the isotropic variable-scale baseline typically requires $\sim$ 23–29 iterations; for $\gamma = 5$ the isotropic run exhibits a mild late-iteration stagnation (Figure 5).

In terms of accuracy, we report in Figure 6 relative L^2 errors over sea points, computed with respect to the true temperature field $\mathbf{s}_{\text{true}}$, for the background $(\mathbf{s}_b)$, the observations $(\mathbf{y})$, and the isotropic/anisotropic analyses, We obtain that the best anisotropic analysis ($\gamma = 5$) (see Figure 7f) improves upon the best isotropic baseline ($\gamma = 3$) (see Figure 7a) by $\approx 4.8\%$, and remains robust across γ.

Figure 7 compares qualitatively optimal estimates at fixed $\sigma = 15$ for $\gamma \in \{3,4,5\}$. The isotropic panels (7a, 7c, 7e)—with grid-aligned, variable scales—exhibit excessive cross-stream diffusion and overly smoothed frontal gradients. In contrast, the anisotropic solutions (7b, 7d, 7f), oriented along the surface-flow, preserve sharper fronts and coherent, flow-parallel filaments while limiting cross-stream spreading. Increasing γ strengthens along-flow coherence and further suppresses spurious diffusion, yielding analyses that qualitatively align better with $\mathbf{s}_{\text{true}}$.

Aligning the diffusion tensor with the local velocity field yields faster CG convergence and uniformly better analyses than the velocity-informed isotropic baseline, while remaining fully SPD and inexpensive thanks to recursive 1D factorizations. In summary, replacing isotropic with flow-aligned anisotropic covariance modelling acts as an effective preconditioner for the CG solve—reducing iteration counts via improved conditioning of $\mathbf{A} = \mathbf{I} + \mathbf{B}\mathbf{R}^{-1}$ while simultaneously lowering the relative error L^2 with respect to $\mathbf{s}_{\text{true}}$. The anisotropic scheme further exhibits greater robustness as the anisotropy parameter γ increases (i.e., stronger along-flow elongation), sustaining stable convergence where the velocity-informed isotropic baseline tends to stagnate.

6 Conclusion

This study introduces a fast anisotropic Gaussian convolution for 3D-VAR data assimilation, leveraging recursive filtering to efficiently approximate horizontal covariances. The proposed method improves covariance modeling by incorporating an adaptive anisotropic diffusion tensor, aligning correlation structures with dynamically relevant directions while maintaining computational efficiency. Numerical experiments confirm that this approach enhances information propagation and captures spatial anisotropy effectively. Moreover, the analysis of CG residuals demonstrates that the flow-aligned anisotropic covariance operator significantly improves the conditioning of the background error covariance matrix, leading to faster convergence of iterative solvers compared to conventional isotropic approaches. This improved conditioning translates into substantial computational benefits for operational data assimilation systems, while simultaneously delivering superior accuracy in the analysis increment.

Future research will extend this framework by integrating the Gabor filter to selectively extract relevant velocity structures and spatial scales, enhancing data assimilation through targeted spectral filtering and improved covariance representation.

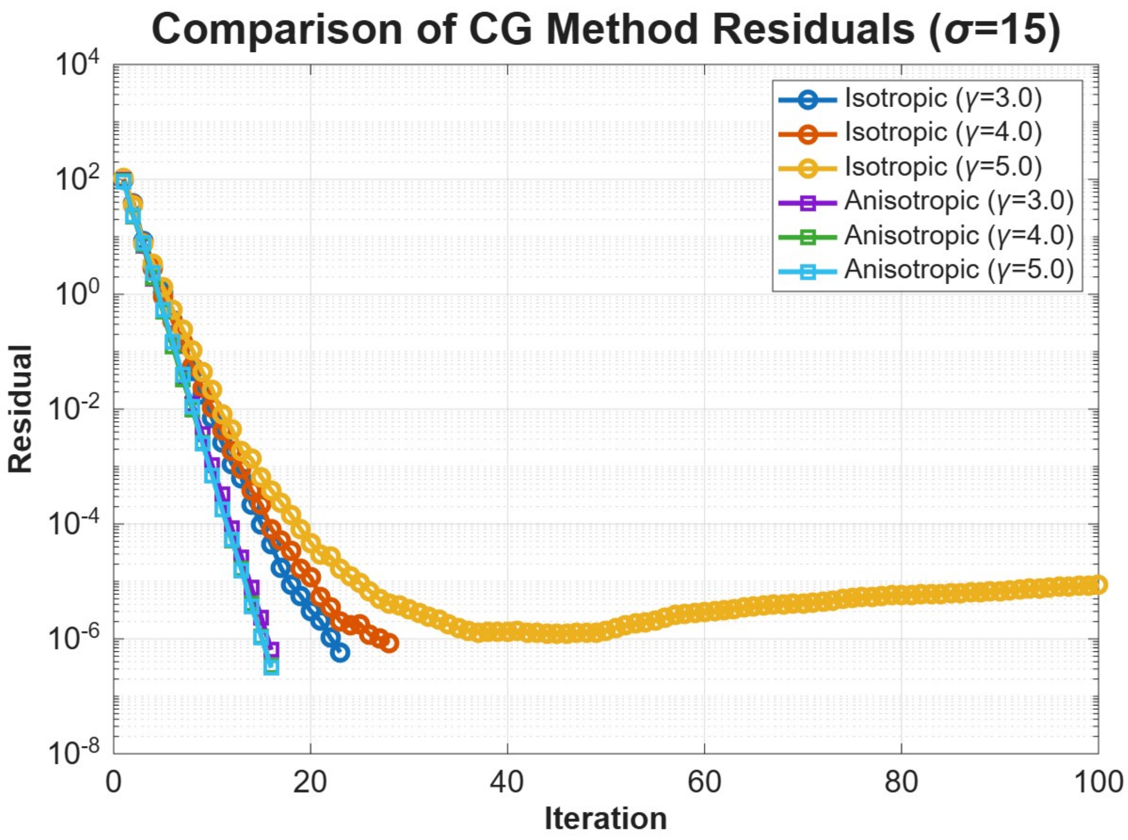


Figure 5: CG convergence (residual, log scale). Anisotropic runs reach 10^{-6} in $\sim$ 14–17 iterations; isotropic runs require $\sim$ 23–29 iterations, with mild stagnation for $\gamma = 5$.

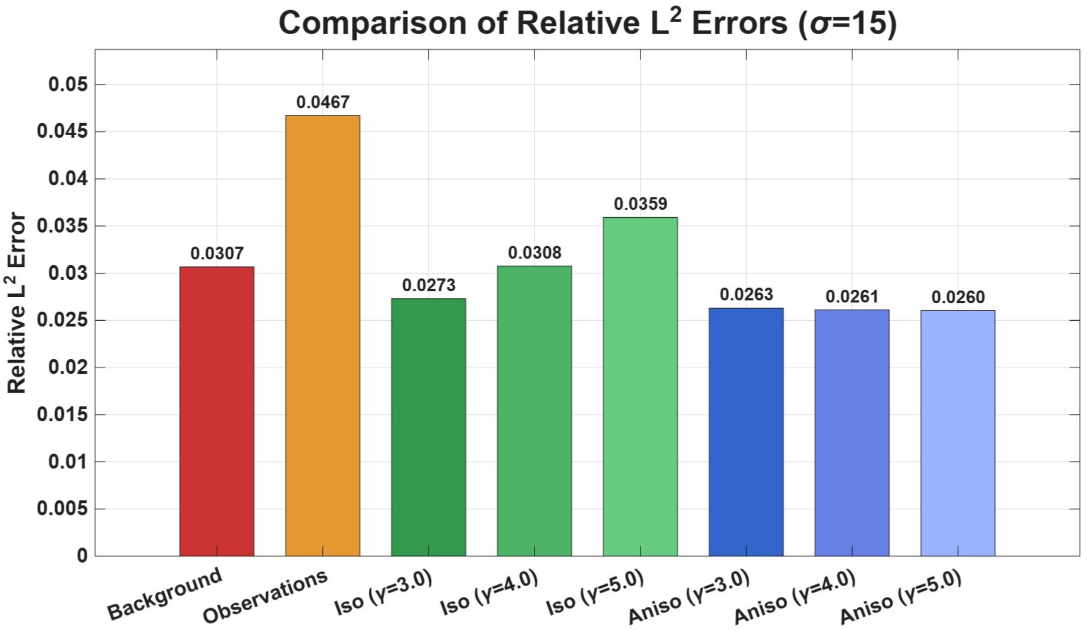


Figure 6: Relative L^2 errors with respect to true state for background, observations, and 3D-VAR analyses at $\sigma = 15$, $\gamma \in \{3, 4, 5\}$, and $\sigma_o = 1.0\,^\circ$C. Errors computed over sea points only. Anisotropic solutions uniformly outperform isotropic across all γ.

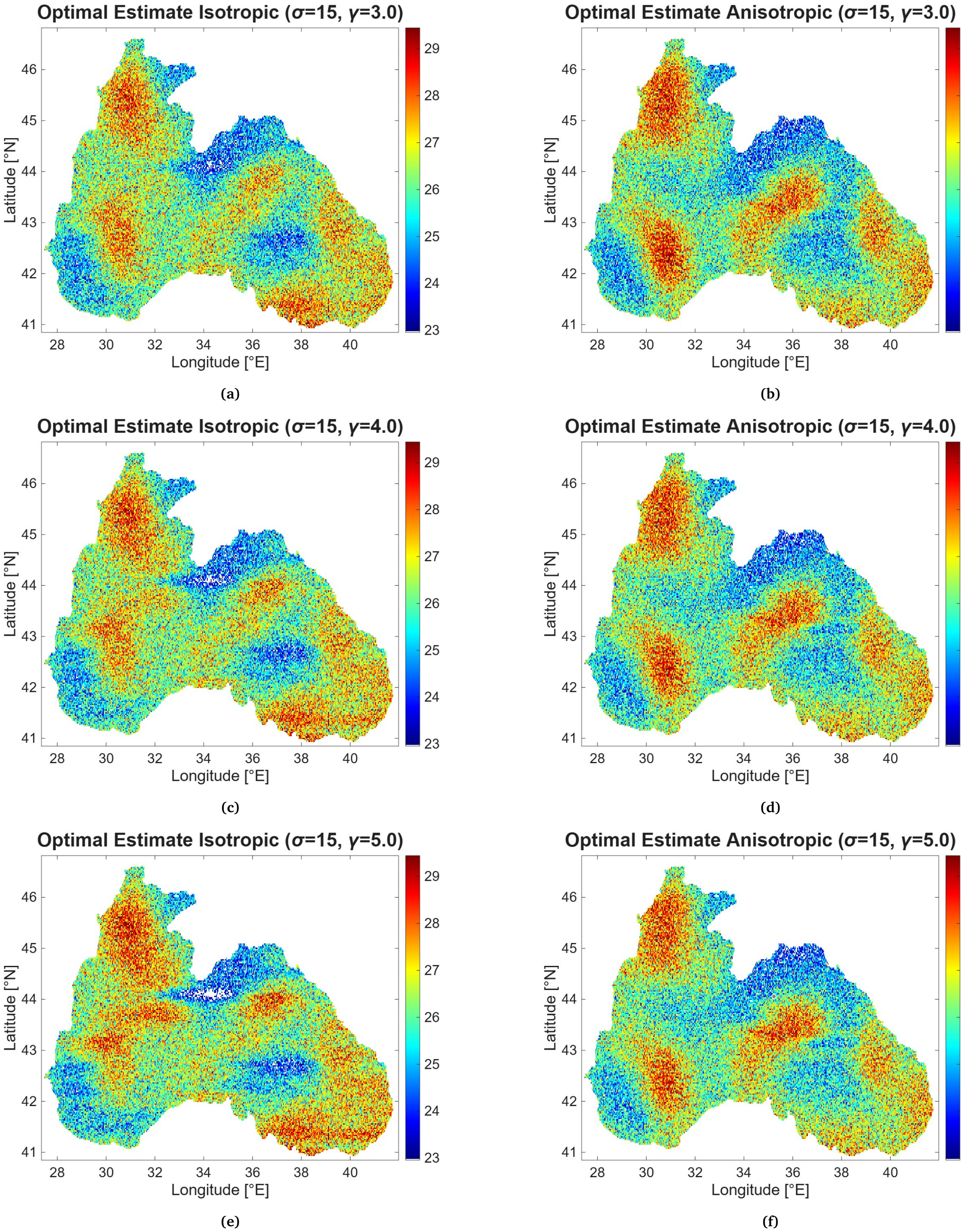

Optimal Estimate Isotropic (σ=15, γ=3.0)
Optimal Estimate Anisotropic (σ=15, γ=3.0)
Optimal Estimate Isotropic (σ=15, γ=4.0)
Optimal Estimate Anisotropic (σ=15, γ=4.0)
Optimal Estimate Isotropic (σ=15, γ=5.0)
Optimal Estimate Anisotropic (σ=15, γ=5.0)
Latitude [°N]
Longitude [°E]
41
42
43
44
45
46
28
30
32
34
36
38
40
23
24
25
26
27
28
29
(a)
(b)
(c)
(d)
(e)
(f)

Figure 7: Pairwise comparison of optimal estimates at fixed $\sigma = 15$ and $\gamma \in \{3,4,5\}$: (a) isotropic variable-scale baseline with $\gamma = 3.0$, (b) flow-aligned anisotropic filter with $\gamma = 3.0$, (c) isotropic baseline with $\gamma = 4.0$, (d) flow-aligned anisotropic filter with $\gamma = 4.0$, (e) isotropic baseline with $\gamma = 5.0$, and (f) flow-aligned anisotropic filter with $\gamma = 5.0$. The anisotropic analyses preserve frontal sharpness and limit cross-stream diffusion.

References

[1] Lorenc, A. C., "Analysis methods for numerical weather prediction," *Quarterly Journal of the Royal Meteorological Society*, vol. 112, pp. 1177–1194, 1986.

[2] Weaver, A., Vialard, J., and Anderson, D., "3D and 4D variational assimilation in a tropical Pacific OGCM," *Monthly Weather Review*, vol. 131, pp. 1360–1378, 2003.

[3] Weaver, A., and Courtier, P., "Correlation modelling on the sphere using diffusion," *Quarterly Journal of the Royal Meteorological Society*, vol. 127, pp. 1815–1846, 2001.

[4] Lorenc, A. C., "Development of an operational variational assimilation scheme," *Journal of the Meteorological Society of Japan*, vol. 75, pp. 339–346, 1997.

[5] Mirouze, I., and Weaver, A. T., "Representation of correlation functions in variational assimilation using an implicit diffusion operator," *Quarterly Journal of the Royal Meteorological Society*, vol. 136, pp. 1421–1443, 2010.

[6] Farina, R., Dobricic, S., Storto, A., Masina, S., and Cuomo, S., "A revised scheme to compute horizontal covariances in a 3D-VAR system," *Journal of Computational Physics*, vol. 284, pp. 631–647, 2015.

[7] Allard, D., Senoussi, R., and Porcu, E., "Anisotropy models for spatial data," *Mathematical Geosciences*, vol. 48, pp. 305–328, 2015.

[8] Honda, T., Honda, Y., Tsuyuki, T., Iwasaki, K., Kotsuki, S., Saito, K., and Kunii, M., "JMA Operational Hourly Hybrid 3DVar with Singular Vector-Based Ensemble Forecasts," *Journal of the Meteorological Society of Japan*, vol. 102, no. 2, 2024. doi:10.2151/jmsj.2024-006.

[9] Geusebroek, J.-M., Smeulders, A. W. M., and van de Weijer, J., "Fast anisotropic Gauss filtering," *IEEE Transactions on Image Processing*, vol. 12, pp. 938–943, 2003.

[10] Young, I. T., and Van Vliet, L. J., "Recursive implementation of Gaussian filters," *Signal Processing*, vol. 44, pp. 139–151, 1995.

[11] Dobricic, S., and Pinardi, N., "An oceanographic 3D-VAR assimilation scheme," *Ocean Modelling*, vol. 22, pp. 89–105, 2008.

[12] Cuomo, S., Farina, R., Galletti, A., and Marcellino, L., "Numerical effects of Gaussian recursive filters in 3DVAR," *Numerical Mathematics: Theory, Methods and Applications*, vol. 10, pp. 520–540, 2017.

[13] Weaver, A., Mirouze, I., and Piacentini, A., "An evaluation of methods for normalizing diffusion-based covariance operators used for background-error modelling," *Quarterly Journal of the Royal Meteorological Society*, vol. 147, no. 741, pp. 2267–2285, 2021. doi:10.1002/qj.3918.

[14] Hu, Y., Zhang, X., Li, D., Li, W., Zhang, L., Fu, H., and Zhang, L., "Anisotropic diffusion filters for flow-dependent variational data assimilation of sea surface temperature," *Ocean Modelling*, vol. 184, 102233, 2023. doi:10.1016/j.ocemod.2023.102233.

[15] Jung, B.-J., Menetrier, B., Snyder, C., Liu, Z., Guerrette, J. J., Ban, J., Banos, I. H., Yu, Y. G., and Skamarock, W. C., "Three-dimensional variational assimilation with a multivariate background error covariance for the Model for Prediction Across Scales–Atmosphere with the Joint Effort for Data assimilation Integration (JEDI-MPAS 2.0.0-beta)," *Geoscientific Model Development*, vol. 17, no. 9, pp. 3879–3895, 2024. doi:10.5194/gmd-17-3879-2024.

[16] Bannister, R. N., "A review of operational methods of variational and ensemble-variational data assimilation," *Quarterly Journal of the Royal Meteorological Society*, vol. 143, no. 703, pp. 607–633, 2017.

[17] Mogensen, K., Balmaseda, M. A., and Weaver, A. T., "The NEMOVAR ocean data assimilation system as implemented in the ECMWF ocean analysis system," *ECMWF Technical Memorandum*, no. 668, 2012.

[18] Rawlins, F., Ballard, S. P., Bovis, K. J., Clayton, A. M., Li, D., Inverarity, G. W., Lorenc, A. C., and Payne, T. J., "The Met Office global four-dimensional variational data assimilation scheme," *Quarterly Journal of the Royal Meteorological Society*, vol. 133, no. 623, pp. 347–362, 2007.

[19] Triggs, B., and Sdika, M., "Boundary conditions for recursive filtering," *IEEE Transactions on Signal Processing*, vol. 54, pp. 2365–2367, 2006.

[20] Hestenes, M. R., and Stiefel, E. L., "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.