Meta-Learning Fourier Neural Operators for Hessian Inversion and Enhanced Variational Data Assimilation

Hamidreza Moazzami, Asma Jamali
School of Computational Science and Engineering,
McMaster University, Hamilton, Canada

Nicholas Kevlahan
Mathematics and Statistics,
McMaster University, Hamilton, Canada
kevlahan@mcmaster.ca

Rodrigo A. Vargas-Hernández
Department of Chemistry and Chemical Biology,
McMaster University, Hamilton, Canada
vargashr@mcmaster.ca

Abstract

Data assimilation (DA) is crucial for enhancing solutions to partial differential equations (PDEs), such as those in numerical weather prediction, by optimizing initial conditions using observational data. Variational DA methods are widely used in oceanic and atmospheric forecasting, but become computationally expensive, especially when Hessian information is involved. To address this challenge, we propose a meta-learning framework that employs the Fourier Neural Operator (FNO) to approximate the inverse Hessian operator across a family of DA problems, thereby providing an effective initialization for the conjugate gradient (CG) method. Numerical experiments on a linear advection equation demonstrate that the resulting FNO-CG approach reduces the average relative error by 62% and the number of iterations by 17% compared to the standard CG. These improvements are most pronounced in ill-conditioned scenarios, highlighting the robustness and efficiency of FNO-CG for challenging DA problems.

1 Introduction

Partial differential equation (PDE) models, such as those used in weather prediction, often rely on initial conditions that are unknown or only approximately known, which makes them a significant source of error [1]. Data assimilation (DA) addresses this by optimally estimating the initial state using sparse, arbitrarily distributed observations in space and time. Widely applied in oceanic and atmospheric forecasting, DA methods are typically categorized as sequential or variational [2–4]. The most common variational method, 4D-Var, frames DA as an optimal control problem, minimizing the discrepancy between model simulations and observations over time and space [5].

4D-Var optimization typically relies on gradient-based methods, but convergence and accuracy can be improved by incorporating Hessian information [6]. Direct computation of the Hessian, or even Hessian–vector products, is often prohibitively expensive, particularly in ill-conditioned settings. To mitigate this, several strategies have been explored, including preconditioning [7], reduced-order modelling [8], and Hessian approximation methods [9, 10].

Here, FNOs are leveraged to improve the efficiency and accuracy of 4D-Var. FNOs, known for learning PDE operators efficiently [11–13], have been applied to DA and PDE-constrained optimization problems. For instance, Singh et al. [14] combined FNO-based predictions with correction operators for DA, while Kang et al. [15] employed FNO surrogate solvers for PDEs to accelerate gradient computations. In this work, we reinterpret the FNO through the lens of meta-learning [16], training

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

it to model the inverse Hessian operator across a distribution of 4D-Var problems. In doing so, the FNO becomes a reusable, task-adaptive solver that shapes the optimization trajectory, accelerating convergence while retaining the rigour of classical solvers. This data-driven approach offers a computationally efficient alternative to traditional iterative schemes, where the model implicitly captures the structure of the inverse Hessian to advance the state of DA.

2 Methodology

4D-Var framework. Variational DA methods are based on minimizing a cost function by tuning the model trajectory to the observations, which reduces cumulative errors over time. The time evolution of the model state from time step k to $k+1$ is defined as, $\mathbf{u}_{k+1} = \mathcal{M}_{k+1}(\mathbf{u}_k)$, where $\mathbf{u}$ is the model state-vector and $\mathcal{M}$ is a nonlinear model operator. The 4D-Var loss function is defined as,

$$J(\mathbf{u}) = \frac{1}{2}(\mathbf{u}_0 - \mathbf{u}^b)^T\mathbf{B}^{-1}(\mathbf{u}_0 - \mathbf{u}^b) + \frac{1}{2}\sum_{k=0}^{K}(\mathbf{H}\mathbf{u}_k - \mathbf{u}_k^{obs})^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{u}_k - \mathbf{u}_k^{obs}) \tag{1}$$

where the first term is a regularization term that minimizes the distance between the initial state, $\mathbf{u}_0$, and the background state, $\mathbf{u}^b$, which is the prior information about the study system. $\mathbf{H}$ is the linear observation operator that interpolates the model forecasts into observations. $\mathbf{B}$ and $\mathbf{R}$ are the background error covariance matrix and the observation error covariance matrix, respectively. The second term is simply the difference between observations, $\mathbf{u}_k^{obs}$ and the PDE simulation, $\mathbf{u}_k$, over time. The gradient of the loss function is given by,

$$\nabla J(\mathbf{u}_0) = \mathbf{B}^{-1}(\mathbf{u}_0 - \mathbf{u}^b) - \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{d}_0 + \mathbf{M}_1^T\left[\mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{d}_1 + \mathbf{M}_2^T\left[\mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{d}_2 + ... + \mathbf{H}_K^T\mathbf{R}_K^{-1}\mathbf{d}_K\right]\right], \tag{2}$$

where $\mathbf{d}_k$ is the distance between the observations and the model forecast, and is defined as,

$$\mathbf{d}_k = (\mathbf{u}_k^{obs} - \mathbf{H}_k\mathcal{M}_k\mathcal{M}_{k-1}...\mathcal{M}_2\mathcal{M}_1\mathbf{u}_0), \tag{3}$$

where $\mathbf{M}_k$ is the tangent linear model (TLM) of the nonlinear model, $\mathcal{M}_k$, and $\mathbf{M}_k^T$ is its adjoint model, which evolves the state backward in time [17]. If the PDE is linear ($\mathbf{M}_k$), the operators $\mathbf{G}_k$ containing both $\mathbf{H}$ and $\mathbf{M}_k$ and $\mathbf{G}_k^T$ containing $\mathbf{H}^T$ and $\mathbf{M}_k^T$ are defined as,

$$\mathbf{G}_k = \mathbf{H}_k\mathbf{M}_k\mathbf{M}_{k-1}...\mathbf{M}_2\mathbf{M}_1 \quad \text{and} \quad \mathbf{G}_k^T = \mathbf{M}_1^T\mathbf{M}_2^T...\mathbf{M}_{k-1}^T\mathbf{M}_k^T\mathbf{H}_k^T. \tag{4}$$

Since the model and the observation operator are linear and the cost function is quadratic, the Hessian of the loss function, $\nabla^2 J$ can be used to write the exact first-order Taylor expansion of the gradient, $\nabla J(\mathbf{u}_0) = \nabla J(\mathbf{u}_0 = 0) + \nabla^2 J\mathbf{u}_0$, where $\mathbf{u}_0$ is optimum at $\nabla J = 0$. Therefore, the optimal initial state, $\mathbf{u}_0^{opt}$ is achieved via:

$$\nabla^2 J\ \mathbf{u}_0^{opt} = \mathbf{f} \tag{5}$$

where $\mathbf{f}$ is the solution of the gradient equation in Eq. (2) by setting $\mathbf{u}_0 = \mathbf{0}$,

$$\mathbf{f} = -\nabla J(\mathbf{u}_0 = \mathbf{0}) = \mathbf{B}^{-1}\mathbf{u}^b + \sum_{k=0}^{K}\mathbf{G}_k^T\mathbf{R}_k^{-1}\mathbf{u}_k^{obs}. \tag{6}$$

Although Eq. (5) is derived for a linear PDE, incremental 4D-Var extends it to nonlinear PDEs by updating the nonlinear model in the outer loop and using its TLM in the inner loop to define a convex quadratic cost minimized via the Gauss–Newton approximation [18, 19]. This highlights the potential applicability of our approach to nonlinear PDEs.

FNO for Hessian-based Data Assimilation. The inverse of the Hessian in Eq. (5) is computationally intensive for high-dimensional systems and, therefore, is usually solved iteratively [10]. The CG method typically requires a large number of iterations to solve Eq. (5), especially when the condition number of the Hessian is high [20]. The multigrid method can be an efficient alternative approach for solving Eq. (5). However, it is restricted to elliptic problems, which limits its applicability [21, 22].

In this study, a surrogate FNO for the Hessian inverse operator is introduced, allowing Eq. (5) to be solved in a single step, without relying on an iterative process,

$$\mathbf{u}_0^{opt} = [\nabla^2 J]^{-1}_{\text{FNO}}(\mathbf{f}). \tag{7}$$

In Eq. (7), the FNO model plays the role of the inverse Hessian, $[\nabla^2 J]^{-1}_{\text{FNO}}$, that maps $\mathbf{f}$ to the target variable, $\mathbf{u}_0$. $\mathbf{f}$ is computed by running the standard adjoint model on the observations, $\mathbf{u}_k^{\text{obs}}$ (Eq. (6)). Recall that the observations are sparse and distributed arbitrarily in space and time. An important advantage of the proposed approach is that, unlike most previous ML-based DA studies [23], the sparse observations $\mathbf{u}_k^{\text{obs}}$ are implicit to the FNO, embedded in $\mathbf{f}$ and fed into the model, eliminating the need for a separate observation term in the loss and simplifying training. The FNO uses multiple spectral convolution layers that apply a learned transformation to low-frequency modes, efficiently capturing long-range spatial dependencies and maintaining flexibility across resolutions.

The FNO model was trained on 5,400 points using the Adam optimizer with a learning rate of 10^{-4} and a 32 batch size. The implementation was based on the `fourier_neural_operator` library, with 16 Fourier modes in the architecture. All experiments were performed on an Apple M2 chip.

3 Experimental Setup

The DA framework in this study approximates the initial condition $\mathbf{u}_0$, of a linear advection equation solved on a periodic domain:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \quad x \in \left[-\frac{x_{\max}}{2}, \frac{x_{\max}}{2}\right), \quad t \in [0, T] \tag{8}$$

where $x_{max} = 100\, m$, $T = 90\ s$, and $c = 0.92\, m.s^{-1}$. The true initial state, $\mathbf{u}_0^T$, is obtained by perturbing the background state, $\mathbf{u}^b$, with spatially periodic noise, η, composed of a finite sum of modulated cosine functions with varying frequencies, $\mathbf{u}_0^T = \mathbf{u}^b + \mathbf{B}^{1/2}\ \eta$. The FNO is trained with pairs of ($\mathbf{u}_0^T$, $\mathbf{f}$), which are generated by varying multiple factors. The general form of the background state is,

$$u^b(x) = 0.5 + \beta \sin(\alpha \frac{2\pi x}{x_{\max}} + \phi) \tag{9}$$

where α, β, and ϕ take the following values: $\alpha \in \{2, 4, 6\}$, $\beta \in \{0.1, 0.3, 0.5, 0.7, 1\}$, and $\phi \in \left\{0, \frac{\pi}{3}, \frac{\pi}{4}\right\}$. The second-order autoregressive (SOAR) distribution is employed to model the background error covariance, $\mathbf{B}$ [24, 25]:

$$B_{ij} = \sigma_b^2 \left(1 + \frac{D_{ij}}{L}\right) \exp\left(-\frac{D_{ij}}{L}\right), \tag{10}$$

where D_{ij} is the distance between each pair of grid points, σ_b^2 is the variance of the background error and L is the correlation length scale. To generate the training data, the correlation length scale L is chosen from the set $\{5\Delta x, 10\Delta x, 15\Delta x, 20\Delta x, 25\Delta x\}$, where Δx is the spatial grid's resolution.

Based on the above assumptions for the parameters in $\mathbf{u}^b$ and the correlation length scale L in $\mathbf{B}$, different true initial state vectors $\mathbf{u}_0^T$ were generated. Each $\mathbf{u}_0^T$ can be paired with multiple $\mathbf{f}$ vectors by considering various observation configurations, all sampled from the true solution. We assumed the number of spatial observations is selected from the set $\{2, 4, 6, 8\}$, with all observations equally spaced, and their temporal availability is specified by the set $\{1, 4, 6, 10, 15, 20\}$, where each value denotes the interval (in time steps) between consecutive observations. Samples were generated by discretizing the PDE in Eq. (8) with the Lax—Wendroff scheme and computing $\mathbf{f}$ in Eq. 6 via its adjoint, with training on $[0, T]$ and testing on $[T, 2T]$, each containing $5,400$ samples.

4 Results and Discussion

In this section, we demonstrate that although the $\mathbf{u}_0$ predicted with an FNO model, Eq. (7), is typically less accurate than that of the standard CG solver, their combination improves results compared to using either method alone. To measure the difference between the true solution, $\mathbf{u}_0^T$ and the predicted $\mathbf{u}_0$, we used the relative error:

$$\mathcal{E}(\mathbf{u}_0) = \frac{||\mathbf{u}_0^T - \mathbf{u}_0||}{||\mathbf{u}_0^T||}. \tag{11}$$

Fig. 1a presents the difference between the relative errors of the standalone FNO model and the CG method, $\Delta\mathcal{E}_{\text{FNO}} = \mathcal{E}_{\text{FNO}} - \mathcal{E}_{\text{CG}}$, plotted against the condition number of the Hessian, κ, for the test dataset consisting of 5,400 samples. The distribution displayed to the right of Fig. 1a illustrates that,

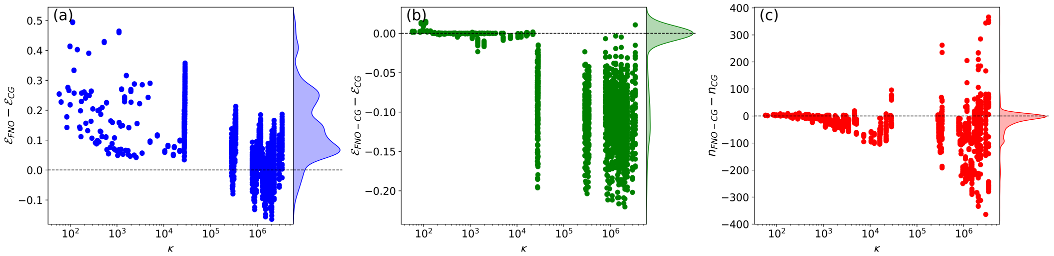

Figure 1: (a) Difference between the relative errors of the FNO and the CG model. (b) Difference between the relative errors of the FNO-CG and the CG model. (c) Difference in the number of iterations between the FNO-CG and the CG model. All panels are as a function of the condition number of the Hessian, κ.

for the majority of cases, $\Delta\mathcal{E}_{\text{FNO}}$ is positive, indicating that the standalone FNO model generally does not outperform the CG method.

As previously discussed, the CG method is widely used in the DA framework, but its performance is highly sensitive to the choice of the initial guess: poor initialization can substantially slow convergence. To overcome this limitation, we propose using FNOs to provide an effective initialization for solving Eq. (5), leading to the FNO-CG approach. Table 1 compares the standard CG method, initialized with the background state $\mathbf{u}^b$ (Eq. (9)), against FNO-CG, where the initializer is the FNO-predicted state $\mathbf{u}_0$. The results show that FNO-CG consistently improves both accuracy and efficiency. Fig. 1b presents the distribution of $\Delta\mathcal{E}_{\text{FNO-CG}} = \mathcal{E}_{\text{FNO-CG}} - \mathcal{E}_{\text{CG}}$. Most samples yield negative values, indicating smaller relative errors with FNO-CG. Cases with positive $\Delta\mathcal{E}_{\text{FNO-CG}}$ remain close to zero and are predominantly associated with well-conditioned problems, where CG alone is already sufficient. This underscores the robustness of FNO-CG in handling poorly conditioned DA problems. Further, Fig.1c illustrates the difference in the number of iterations, $\Delta n_{\text{FNO-CG}} = n_{\text{FNO-CG}} - n_{\text{CG}}$, which is predominantly negative across the dataset, confirming the efficiency gain. On average, FNO-CG reduces the relative error by 62% and the iteration count by 17%.

Table 1: Comparison of CG, FNO, and FNO-CG results for the test data.

	Average Relative Error ↓	**Average Total Iterations ↓**
CG	4.26×10^{-2}	127
FNO-CG	$\mathbf{1.61 \times 10^{-2}}$	**105**
FNO	1.86×10^{-1}	–

Fig. 2 compares CG, FNO, and FNO-CG on three randomly selected test samples. For Sample 1, CG and FNO-CG achieve similar accuracy, but FNO-CG converges with 24 fewer iterations. For Samples 2 and 3, FNO-CG attains 39% and 76% higher accuracy, while also reducing the iteration count by 17 and 103, respectively. The inset panels show the corresponding $\mathbf{u}_0$ predicted by each method. Although the standalone FNO model exhibits a relatively high average relative error, 1.86×10^{-1} (see Table 1), it still captures the overall structure of $\mathbf{u}_0$, providing a useful initialization for CG.

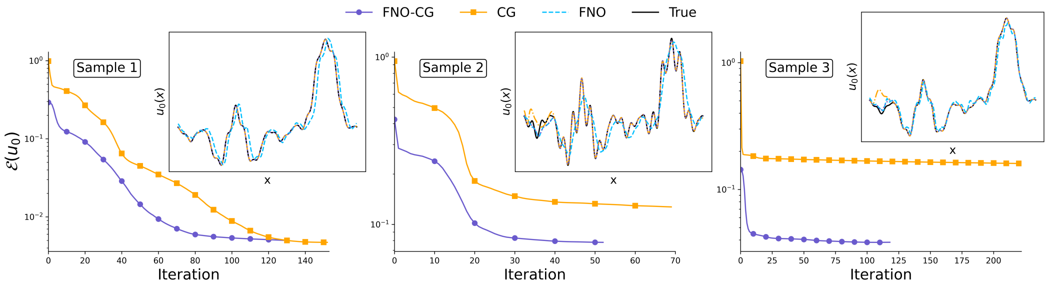

Figure 2: Main panels: relative error as a function of CG iterations when the initial $\mathbf{u}_0$ is set to the background state $\mathbf{u}_b$ or to the initialization predicted by the FNO model. Inset panels: predicted $\mathbf{u}_0$ obtained with CG, FNO, and FNO-CG methods for three randomly selected test samples.

5 Conclusion

We introduced a meta-learning framework based on Fourier Neural Operators to enhance variational data assimilation by approximating the inverse Hessian operator. The proposed FNO-CG method combines the predictive power of FNOs with the robustness of the classical CG algorithm, yielding substantial improvements: a 62% reduction in average relative error and a 17% decrease in iteration count compared to standard CG. Distribution analyses show that these gains are most pronounced in ill-conditioned problems, where optimization is typically more difficult. Beyond this application, our results demonstrate that FNOs can approximate operators beyond PDE solvers, here targeting the inverse Hessian, a central object in many inverse problems. By learning the mapping from $\mathbf{f}$ to an effective initial state $\mathbf{u}_0$, FNO-CG reduces the need for expensive iterative procedures. Future work will extend this meta-learning approach to nonlinear PDEs, with a particular focus on challenging chaotic systems such as the Kuramoto–Sivashinsky equation.

Acknowledgment

NKRK was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant no. RGPIN-2024-05282. RAVH acknowledges the support from the Digital Research Alliance of Canada and NSERC Discovery Grant No. RGPIN-2024-06594.

References

[1] Linus Magnusson, Jan-Huey Chen, Shian-Jiann Lin, Linjiong Zhou, and Xi Chen. Dependence on initial conditions versus model formulations for medium-range forecast error variations. *Quarterly Journal of the Royal Meteorological Society*, 145(722):2085–2100, 2019.

[2] Ionel M Navon. Data assimilation for numerical weather prediction: a review. *Data assimilation for atmospheric, oceanic and hydrologic applications*, pages 21–65, 2009.

[3] Seon Ki Park and Liang Xu. *Data assimilation for atmospheric, oceanic and hydrologic applications (Vol. II)*. Springer, 2013.

[4] Patricia de Rosnay, Philip Browne, Eric de Boisséson, David Fairbairn, Yoichi Hirahara, Kenta Ochi, Dinand Schepers, Peter Weston, Hao Zuo, Magdalena Alonso-Balmaseda, et al. Coupled data assimilation at ecmwf: current status, challenges and future developments. *Quarterly Journal of the Royal Meteorological Society*, 148(747):2672–2702, 2022.

[5] François-Xavier Le Dimet and Olivier Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, 38(2):97–110, 1986.

[6] Coralia Cartis, Maha H Kaouri, Amos S Lawless, and Nancy K Nichols. Convergent least-squares optimization methods for variational data assimilation. *Optimization*, 73(11):3451–3485, 2024.

[7] Stephen A Haben, Amos S Lawless, and Nancy K Nichols. Conditioning and preconditioning of the variational data assimilation problem. *Computers & Fluids*, 46(1):252–256, 2011.

[8] Răzvan Ştefănescu, Adrian Sandu, and Ionel Michael Navon. Pod/deim reduced-order strategies for efficient four dimensional variational data assimilation. *Journal of Computational Physics*, 295:569–595, 2015.

[9] Alexandru Cioaca, Adrian Sandu, and Eric de Sturler. Efficient methods for computing observation impact in 4d-var data assimilation. *Computational Geosciences*, 17(6):975–990, 2013.

[10] Kylen Solvik, Stephen G Penny, and Stephan Hoyer. 4d-var using hessian approximation and backpropagation applied to automatically differentiable numerical and machine learning models. *Journal of Advances in Modeling Earth Systems*, 17(4):e2024MS004608, 2025.

[11] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[12] Surbhi Khetrapal and Jaswin Kasi. A numerical study of chaotic dynamics of ks equation with fnos. *arXiv preprint arXiv:2410.12280*, 2024.

[13] Byoung-Ju Choi, Hong Sung Jin, and Bataa Lkhagvasuren. Applications of the fourier neural operator in a regional ocean modeling and prediction. *Frontiers in Marine Science*, 11:1383997, 2024.

[14] Ashutosh Singh, Ricardo Augusto Borsoi, Deniz Erdogmus, and Tales Imbiriba. Learning semilinear neural operators: A unified recursive framework for prediction and data assimilation. *arXiv preprint arXiv:2402.15656*, 2024.

[15] Chanik Kang, Joonhyuk Seo, Ikbeom Jang, and Haejun Chung. Adjoint method-based fourier neural operator surrogate solver for wavefront shaping in tunable metasurfaces. *iScience*, 28(1), 2025.

[16] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44 (9):5149–5169, 2021.

[17] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data Assimilation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016.

[18] Emilie Neveu. *Applications des méthodes multigrilles à l'assimilation de données en géophysique*. PhD thesis, Université de Grenoble, 2011.

[19] AS Lawless, Serge Gratton, and NK Nichols. Approximate iterative methods for variational data assimilation. *International journal for numerical methods in fluids*, 47(10-11):1129–1135, 2005.

[20] Mike Fisher, Jorge Nocedal, Yannick Trémolet, and Stephen J Wright. Data assimilation in weather forecasting: a case study in pde-constrained optimization. *Optimization and Engineering*, 10(3):409–426, 2009.

[21] Laurent Debreu, Emilie Neveu, Ehouarn Simon, François-Xavier Le Dimet, and Arthur Vidard. Multigrid solvers and multigrid preconditioners for the solution of variational data assimilation problems. *Quarterly Journal of the Royal Meteorological Society*, 142(694):515–528, 2016.

[22] Emilie Neveu, Laurent Debreu, and François-Xavier Le Dimet. Multigrid methods and data assimilation—convergence study and first experiments on non-linear equations. *Revue Africaine de Recherche en Informatique et Mathématiques Appliquées*, 14, 2011.

[23] QiZhi He, David Barajas-Solano, Guzel Tartakovsky, and Alexandre M Tartakovsky. Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Advances in Water Resources*, 141:103610, 2020.

[24] Laura M Stewart, Sarah L Dance, and Nancy K Nichols. Data assimilation with correlated observation errors: experiments with a 1-d shallow water model. *Tellus A: Dynamic Meteorology and Oceanography*, 65(1):19546, 2013.

[25] N Bruce Ingleby. The statistical structure of forecast errors and its representation in the met. office global 3-d variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 127(571):209–231, 2001.