



**HAL**  
open science

# State-dependent Preconditioning for Data Assimilation

Victor Trappler, Arthur Vidard

► **To cite this version:**

Victor Trappler, Arthur Vidard. State-dependent Preconditioning for Data Assimilation. ISDA 2023 - 9th International Symposium on Data Assimilation, Oct 2023, Bologne, Italy. , pp.1-1, 2023. hal-04309242

**HAL Id: hal-04309242**

**<https://hal.science/hal-04309242v1>**

Submitted on 27 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



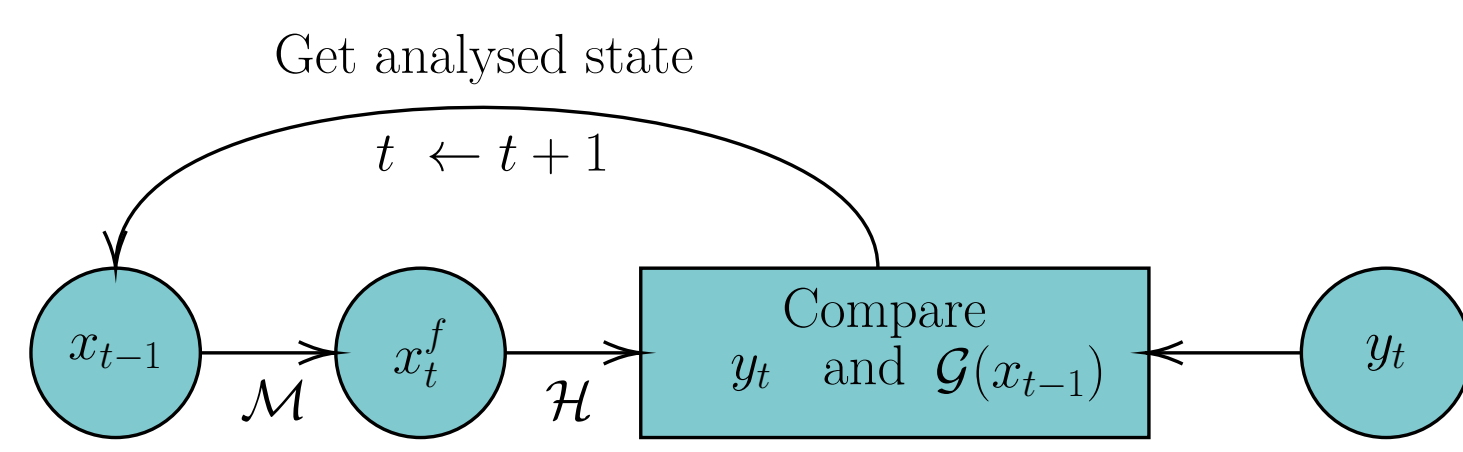
Distributed under a Creative Commons Attribution 4.0 International License

## In short

Variational Data Assimilation requires the optimization of a high-dimensional function. This is usually done by linearizing the problem, and then inverting the Gauss-Newton matrix  
¿ Can we build a state-dependent preconditioner which would help for the inversion of the linear system ?

## Variational Data Assimilation

- $x \in \mathbb{R}^n$ : state
- $\mathcal{M} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ : time propagator
- $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ : observation operator
- $y \in \mathbb{R}^m$ : observations



- $\mathcal{G}$  composes the forward model and the observation operator, to compare with the available observation

$$\mathcal{G} : \mathbb{X} \subseteq \mathbb{R}^n \rightarrow \mathbb{X} \rightarrow \mathbb{R}^m \quad (1)$$

$$x \mapsto \mathcal{M}(x) \mapsto (\mathcal{H} \circ \mathcal{M})(x) = \mathcal{G}(x)$$

- The cost function to optimize to get the analysis is

$$J_{4D}(x) = \frac{1}{2} \|\mathcal{G}(x) - y\|_{R^{-1}}^2 + \frac{1}{2} \|x - x^b\|_{B^{-1}}^2 \quad (2)$$

and

$$x_{t-1}^a = \arg \min_{x \in \mathbb{X}} J_{4D}(x) \quad (3)$$

## Incremental 4DVar

### Outer and Inner loops: Minimization as a sequence of Linear Systems

- Linearize  $J$  around  $x$  (Linear Inverse Problem):

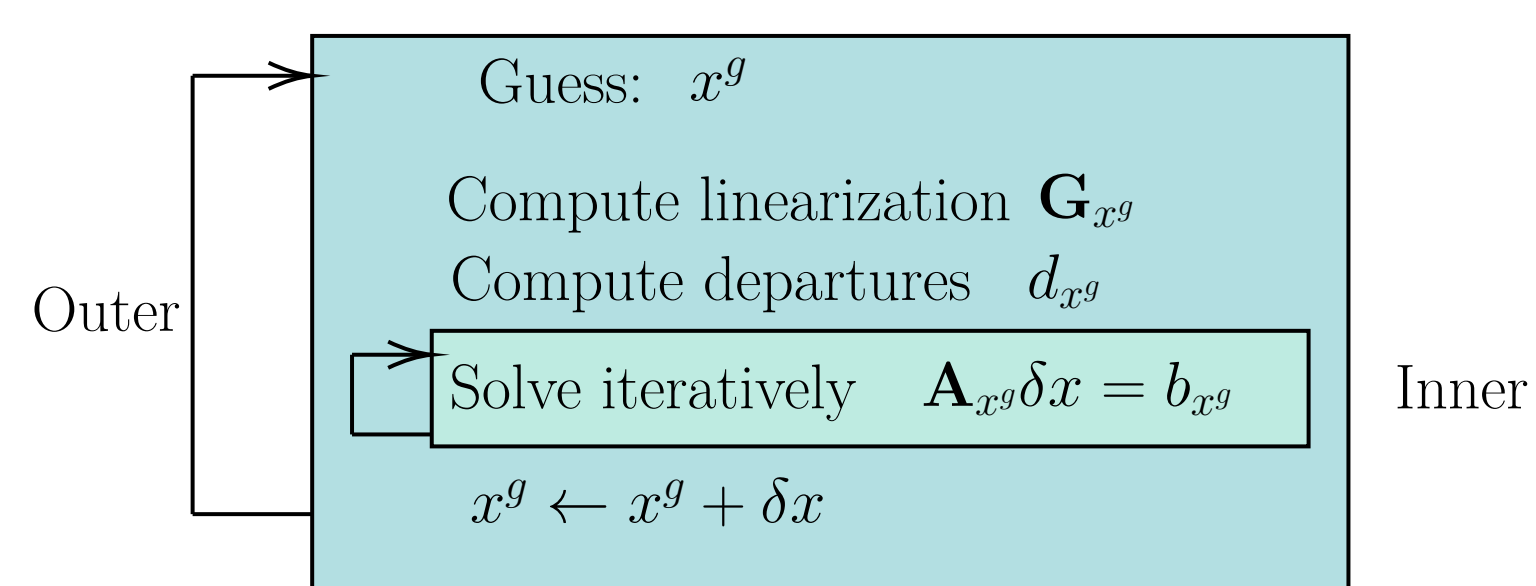
$$J_{\text{incr}}(x, \delta x) = \frac{1}{2} \|\mathbf{G}_x \delta x + \underbrace{(\mathcal{G}(x) - y)}_{-d_x}\|_{R^{-1}}^2 + \frac{1}{2} \|\delta x + x - x^b\|_{B^{-1}}^2 \quad (4)$$

- The optimal increment solves

$$\underbrace{(\mathbf{G}_x^T R^{-1} \mathbf{G}_x + B^{-1})}_{\mathbf{A}_x} \delta x = \underbrace{-\mathbf{G}_x^T R^{-1} d_x - B^{-1} (x - x^b)}_{b_x} \quad (5)$$

where  $\mathbf{A}_x$  Gauss-Newton Matrix  $\iff$  Inverse of the posterior error cov matrix

$$\mathbf{A}_x = \mathbf{G}_x^T R^{-1} \mathbf{G}_x + B^{-1} \in \mathbb{R}^{n \times n} \text{ symmetric and spd} \quad (6)$$



### In the Inner Loop

- $\mathbf{A}_x$  is spd, so **Conjugate Gradient** can be used
- Convergence rate depends on the spectrum of  $\mathbf{A}_x$  ie
  - Condition number:  $\kappa(\mathbf{A}_x) = \sigma_{\max} / \sigma_{\min} = \|\mathbf{A}_x\| \cdot \|\mathbf{A}_x^{-1}\|$
  - Clustering of eigenvalues at 1

## State Dependent Preconditioner

### Preconditioning

Instead of solving  $\mathbf{A}_x \delta x = b_x$ , solve  $(L^T \mathbf{A}_x L) z = L^T b$  and  $\delta x = L z$

- $H = LL^T$  symmetric, positive definite, cheap to compute and to apply
- $H$  should be close to  $\mathbf{A}_x^{-1}$
- $1 \leq \kappa(H \mathbf{A}_x) \leq \kappa(\mathbf{A}_x)$

"One-fits-all" preconditioner do not exist, most include information on *spectrum* of  $\mathbf{A}_x$ , which depends on  $x$

### State-dependent preconditioner

We propose to construct a mapping

$$x \mapsto H(x) \quad (7)$$

where  $H(x)$  is a preconditioner well-suited for the linear system  $\mathbf{A}_x \delta x = b_x$

### Challenges

- $H(x) \in \mathbb{R}^{n \times n}$  is spd (ie  $n(n+1)/2$  "free" parameters)
- $\mathbf{A}_x$  is not stored explicitly (only accessible as  $\text{TL}(x, z) = \mathbf{A}_x z$ ) and high-dimensional
- Independence with respect to the observations (thus to  $b_x$ )
- $H(x)$  should contain spectral information of  $\mathbf{A}_x$

## ML framework

### Objective

- Construct a preconditioner using DNN, which requires **no** call to  $\mathbf{A}_x$  when in use
- No access to  $\mathbf{A}_x^{-1}$  during the training, no explicit construction of  $\mathbf{A}_x$

### Low-rank approximation

Let  $U_\theta \in \mathbb{R}^{n \times r}$  whose columns are orthonormal vectors, and  $\Lambda_\theta = \text{diag}(\lambda_1, \dots, \lambda_r) \in \mathbb{R}^{r \times r}$

$$\tilde{\mathbf{A}}_\theta(x) = U_\theta \Lambda_\theta U_\theta^T \quad (8)$$

Low-rank approximation is optimal wrt to the Frobenius norm

$$\min_\theta \|\mathbf{A}_x - \tilde{\mathbf{A}}_\theta(x)\|_F^2 \quad (9)$$

but  $\mathbf{A}_x$  cannot be explicitly constructed in practice: estimation of F-norm

$$\hat{\mathcal{L}}(\theta, x_i) = \frac{1}{n_z} \sum_{j=1}^{n_z} \|\mathbf{A}_{x_i} z_j - \tilde{\mathbf{A}}_\theta(x_i) z_j\|_2^2 \quad \text{with } z_j \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_n) \quad (10)$$

Possibility of **online training**:

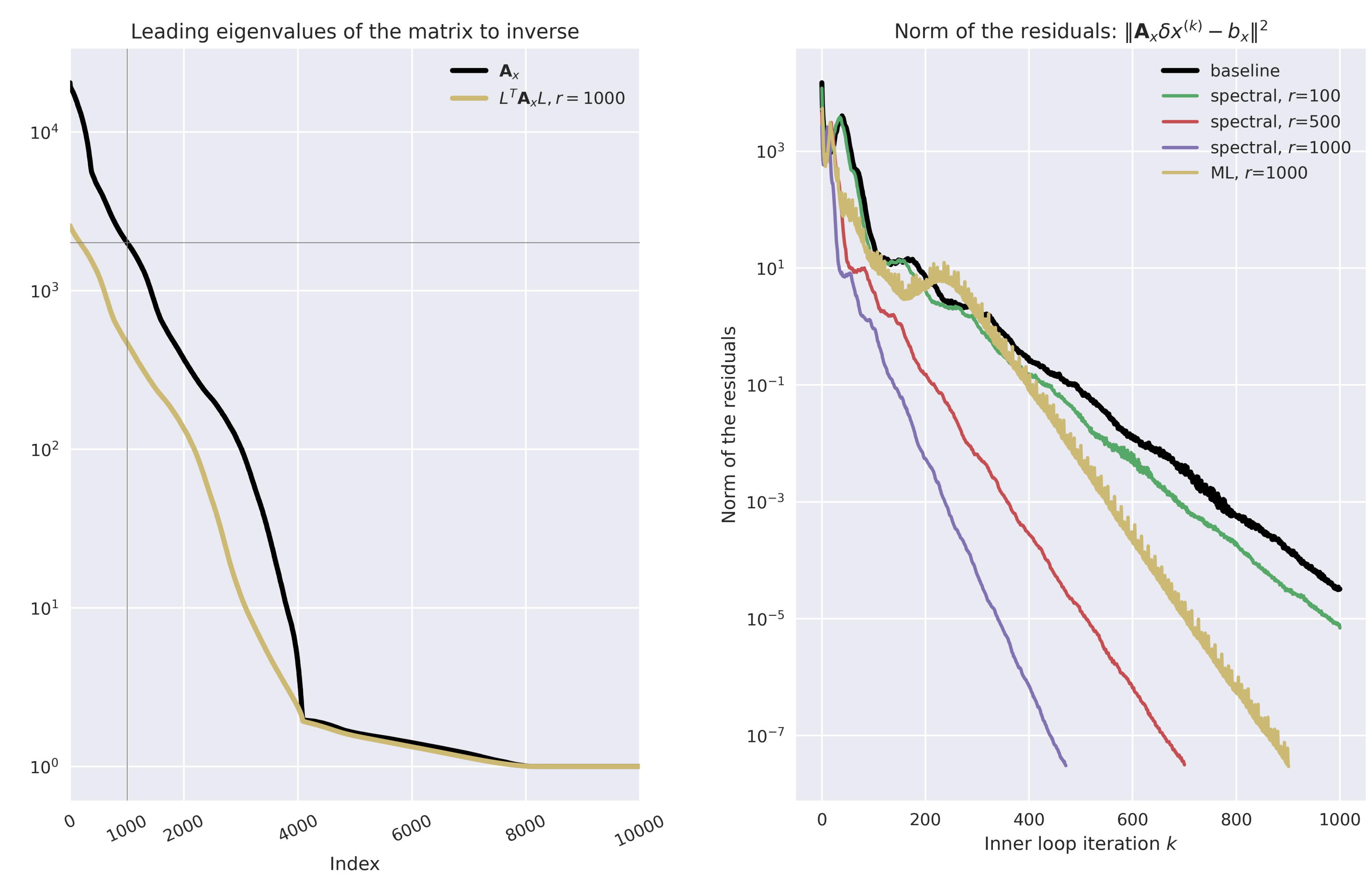
- Only  $\mathbf{A}_x z_j$  needed instead of constructing and storing  $\mathbf{A}_x$  explicitly

When trained, we can construct the preconditioner as ([5])

$$L_\theta = U_\theta \left( \Lambda_\theta^{-\frac{1}{2}} - I_r \right) U_\theta^T + I_n, \quad \text{sp}(H_\theta) = (1, \dots, 1, \lambda_r^{-1}, \dots, \lambda_1^{-1}) \quad (11)$$

## Numerical Results

SW assimilation system,  $n \approx 12000$ , UNet-like architecture:  $\eta, u, v$  as features



## Conclusion and further work

- We propose to use DNN in order to build a preconditioner for inverting the Gauss-Newton matrix, which is **state-dependent** (or parametrized spd matrices in general)
- Make use of spectral information
- Use this information for dimension reduction (with Bayesian inverse problem point of view) [3]
- Appropriate ML architecture to output large numbers of eigenvectors ?
- How to deal with changing observation operator ?

## References

- [1] N. Bousserez and Daven K. Henze. Optimal and scalable methods to approximate the solutions of large-scale Bayesian problems: Theory and application to atmospheric inversion and data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(711):365–390, 2018.
- [2] S. Gratton, A. S. Lawless, and N. K. Nichols. Approximate Gauss-Newton Methods for Nonlinear Least Squares Problems. *SIAM Journal on Optimization*, 18(1):106–132, January 2007.
- [3] Alessio Spantini, Antti Solonen, Tiangang Cui, James Martin, Luis Tenorio, and Youssef Marzouk. Optimal low-rank approximations of Bayesian linear inverse problems. *arXiv:1407.3463 [math, stat]*, July 2015.
- [4] Jemima M. Tabart, Sarah L. Dance, Stephen A. Haben, Amos S. Lawless, Nancy K. Nichols, and Joanne A. Waller. The conditioning of least-squares problems in variational data assimilation. *Numerical Linear Algebra with Applications*, 25(5):e2165, 2018.
- [5] J. Tshimanga, S. Gratton, A. T. Weaver, and A. Sartenaer. Limited-memory preconditioners, with application to incremental four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 134(632):751–769, 2008.