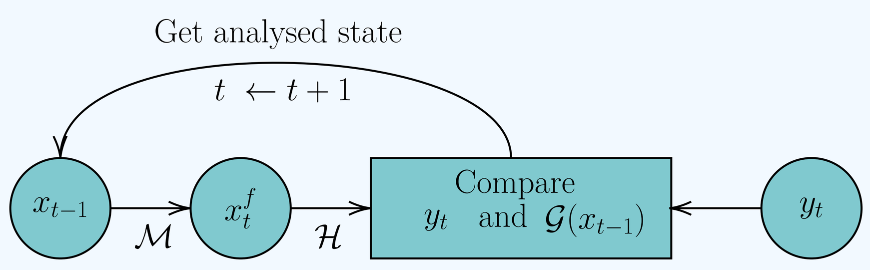
State-dependent Preconditioning for Data Assimilation

Victor Trappler [1,2] Arthur Vidard [1]

[1]AIRSEA Research team – Inria – Laboratoire Jean Kuntzmann
[2]AI4Sim R&D team – Eviden (ex Bull/Atos)

Variational Data Assimilation

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



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

$$\begin{aligned}\mathcal{G}: \mathbb{X} \subseteq \mathbb{R}^n &\longrightarrow \mathbb{X} \longrightarrow \mathbb{R}^m \\ x &\longmapsto \mathcal{M}(x) \longmapsto (\mathcal{H}\circ\mathcal{M})(x) = \mathcal{G}(x)\end{aligned} \tag{1}$$

- The cost function to optimize in order to get the analysis is

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

and

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

Incremental 4DVar

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

- Linearize J around x (Linear Inverse Problem):

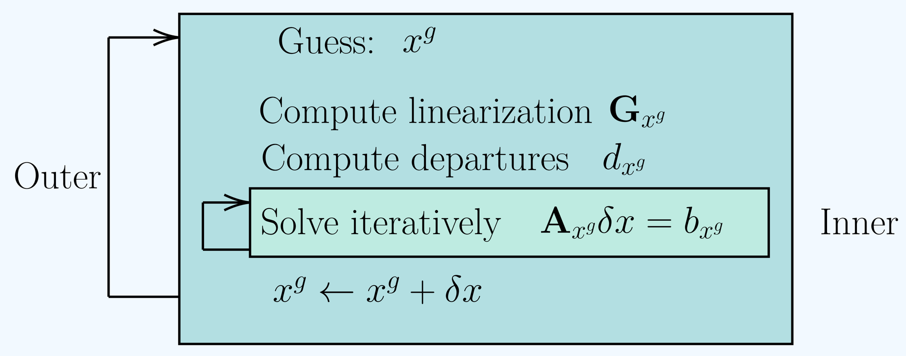
$$J_{\text{incr}}(x, \delta x) = \frac{1}{2}\|\mathbf{G}_x\delta x + \underbrace{(\mathcal{G}(x) - y)}_{-d_x}\|^2_{R^{-1}} + \frac{1}{2}\|\delta x + x - x^b\|^2_{B^{-1}} \tag{4}$$

- The optimal increment solves

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

where $\mathbf{A}_x$ Gauss-Newton Matrix $\iff$ Inverse posterior covariance Matrix

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



In the Inner Loop

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

State Dependent Preconditioner

Preconditioning

Instead of solving $\mathbf{A}_x\delta x = b$, solve $H\mathbf{A}_x\delta x = Hb$ instead

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

But "one-fits-all" preconditioner do not exist, most include information on *spectrum* of $\mathbf{A}_x$.

State-dependent preconditioner

We propose to construct a mapping

$$x \longmapsto H(x) \tag{7}$$

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 $\mathrm{TL}(x, z) = \mathbf{A}_x z$) 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 $x \mapsto H_\theta$ using DNN, which require **no** call to $\mathbf{A}_x$ when in use
- No access to $\mathbf{A}_x^{-1}$ during the training

Limited Memory Preconditioners [4]/Balancing Preconditioners

Let $S, S' \in \mathbb{R}^{n\times r}$, $S' = \mathbf{A}_x S$

$$H_{\text{LMP}}(S, S') = (I_n - S(S^TS')^{-1}S'^T)(I_n - S'(S^TS')^{-1}S^T) + S(S^TS')^{-1}S^T \tag{8}$$

We define the preconditioner as

$$H_\theta : x \mapsto H_{\text{LMP}}(S_\theta(x), \tilde{A}_\theta(x)S_\theta(x)) \tag{9}$$

and H_θ^{-1} available in a similar way

Loss function, Estimation of Frobenius norm

If we constrain the norm of $\|H_\theta\|$ (e.g. by choosing S_θ as eigenvectors)

$$\text{"minimize}\|\mathbf{A}_x - H_\theta^{-1}(x)\|^2_{\text{F}}\text{"} \tag{10}$$

Sample n_z random $z_j \sim \mathcal{N}(0, I_n)$, and an estimation of the loss at a state x_i is

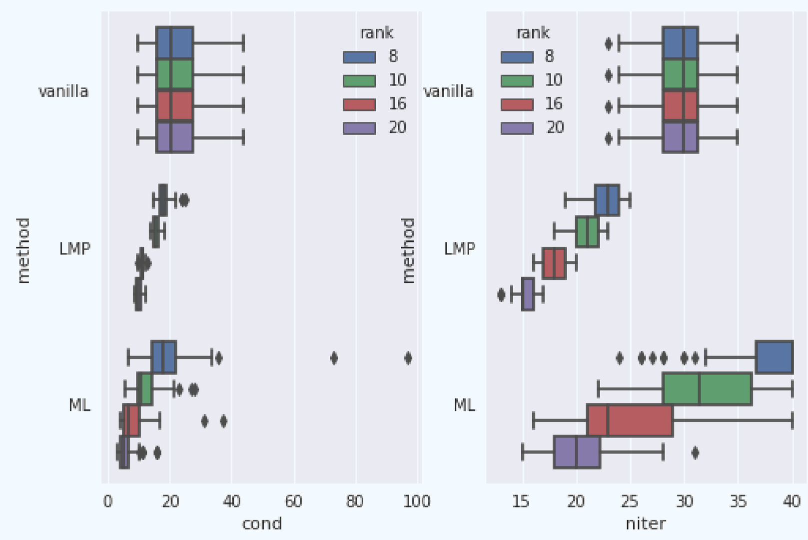
$$\hat{\mathcal{L}}(\theta, x_i) = \frac{1}{n_z}\sum_{j=1}^{n_z}\|\mathbf{A}_{x_i}z_j - H_\theta^{-1}(x_i)z_j\|_2^2 + \text{regul}(\theta) \tag{11}$$

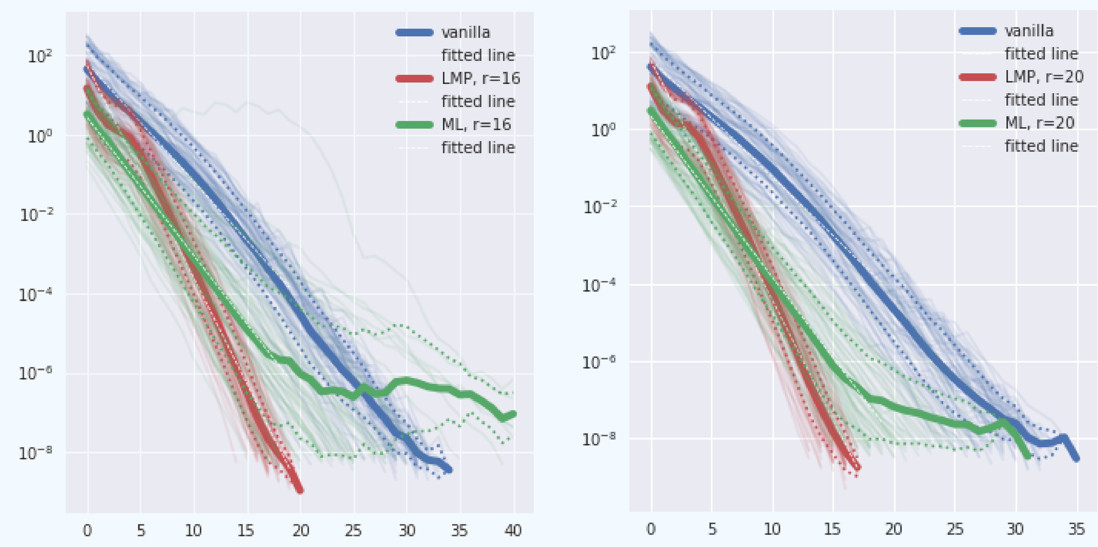
Possibility of **online training**:

- In a DA system, when $\mathbf{A}_x$ is available, evaluate $\mathbf{A}_x z_j$
- Less storage required

Numerical Results

Lorenz96 system, n dimension, state is "spatially" distributed and periodic $\Rightarrow$ CNN





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)
- Use of different metric/regularization for the training of the DNN (Förstner distance...)
- Directly looking for a low-rank / spectral decomposition of $\mathbf{A}_x$ might be of interest
- Use this information for dimension reduction (with Bayesian inverse problem point of view) [2]

References

[1] 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.

[2] 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.

[3] Jemima M. Tabeart, 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.

[4] 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.