Continuous Hyper-parameter OPtimization (CHOP) in an ensemble Kalman filter

Xiaodong Luo, NORCE, Norway; Chuan-An, Xia, Fuzhou University, China

Preprints available from https://arxiv.org/abs/2206.03050

Outline

- Why CHOP?
- A proposed workflow to CHOP problems
- Numerical examples
- Summary and future plans

Why CHOP

Algorithmic hyper-parameters in ensemble data assimilation (DA) algorithms

- Ensemble Kalman filer (EnKF):

$$m_j^a = m_j^b + S_m S_d^T \left(S_d S_d^T + C_d\right)^{-1} \left(d^o - g(m_j^b)\right)$$
$$= m_j^b + K(d^o - g(m_j^b))$$
$$K = S_m S_d^T \left(S_d S_d^T + C_d\right)^{-1}$$

Auxiliary techniques for improved performance:

- ❑ Covariance inflation: $m_j^b \rightarrow m_j^b + (1+\delta)(m_j^b - \bar{m}^b)$
- ❑ Model-space/observation-space/Kalman localization, e.g.,

$S_m S_m^T \rightarrow L(\lambda) \circ (S_m S_m^T)$ ($\circ$ for Schur/element-wise product)

$K \rightarrow L(\lambda) \circ K$

Why CHOP

- Many ways that hyper-parameters can be introduced to DA algorithms
- Tailored or empirical methods for hyper-parameters tuning
- Possibility to have a more automated workflow? (for Continuous Hyper-parameter OPtimization)

Outline

- Why CHOP?
- A proposed workflow to CHOP problems
- Numerical examples
- Summary and future plans

Main idea

NORCE

- Treat a DA algorithm as a parametric mapping:

$$m_j^a = f_\theta(m_j^b) = f(\theta; m_j^b)$$

f: a generic DA algorithm
θ: vector of hyper-parameters

Main idea

$$m_j^a = f_\theta(m_j^b) = f(\theta; m_j^b)$$

- Optimality criterion: Find one or multiple set(s) of θ such that the corresponding simulated observations $g(m_j^a)$ match the real observations d^o to a good extent

- g: observation operator

- Assuming that θ contains continuous hyper-parameters

Continuous Hyper-parameter Optimization (CHOP) problem

NORCE

Formulating a minimum-average-cost (MAC) problem

$$argmin_{\{\theta_j^a\}} \frac{1}{N_e} \sum_j L\left(\theta_j^a | d_j^o, \theta_j^b, m_j^b, \gamma\right), j = 1, 2, \ldots, N_e$$

$$L\left(\theta_j^a | d_j^o, \theta_j^b, m_j^b, \gamma\right) = \frac{1}{2}\left(d^o - g\left(f(\theta_j^a; m_j^b)\right)\right)^T C_d^{-1}\left(d^o - g\left(f(\theta_j^a; m_j^b)\right)\right) + \frac{\gamma}{2}\left(\theta_j^a - \theta_j^b\right)^T C_\theta^{-1}\left(\theta_j^a - \theta_j^b\right)$$

$\frac{1}{2}\left(d^o - g\left(f(\theta_j^a; m_j^b)\right)\right)^T C_d^{-1}\left(d^o - g\left(f(\theta_j^a; m_j^b)\right)\right)$: Data mismatch term measuring the distance between $g\left(m_j^a\right)$ and d^o ($m_j^a = f(\theta_j^b; m_j^b)$)

$\frac{\gamma}{2}\left(\theta_j^a - \theta_j^b\right)^T C_\theta^{-1}\left(\theta_j^a - \theta_j^b\right)$: Regularization term for improving numerical stability/promoting certain desired properties in the solution

Approximate solution to the CHOP problem

$$\theta_j^a = \theta_j^b + S_\theta S_{g\circ f}^T \left(S_{g\circ f} S_{g\circ f}^T + \gamma C_d\right)^{-1} \left(d_j^o - g\left(f\left(\theta_j^b; m_j^b\right)\right)\right), j = 1,2,\dots,N_e$$

$$S_\theta \equiv \frac{1}{\sqrt{N_e - 1}}\left[\theta_1^b - \bar{\theta}^b, \theta_2^b - \bar{\theta}^b, \dots, \theta_{N_e}^b - \bar{\theta}^b\right];\ \bar{\theta}^b = \frac{1}{N_e}\sum_j \theta_j^b;$$

$$S_{g\circ f} \equiv \frac{1}{\sqrt{N_e - 1}}\left[g\left(f\left(\theta_1^b; m_1^b\right)\right) - g\left(f\left(\bar{\theta}^b; \bar{m}^b\right)\right), g\left(f\left(\theta_2^b; m_2^b\right)\right) - g\left(f\left(\bar{\theta}^b; \bar{m}^b\right)\right), \dots, g\left(f\left(\theta_{N_e}^b; m_{N_e}^b\right)\right) - \left(f\left(\bar{\theta}^b; \bar{m}^b\right)\right)\right];$$

- Applying an iterative ensemble smoother (IES) to estimate θ, independent of the assimilation algorithm f
- Estimating an ensemble of θ, rather than a single value. Benefits:
  - ❑ Derivative free (e.g., no need for $\partial f / \partial\theta$)
  - ❑ UQ

Localization in the CHOP problem

NORCE

$$\theta_{\mathrm{j}}^{\mathrm{a}} = \theta_j^{\mathrm{b}} + K\left(d_j^o - g\left(f\left(\theta_j^b; m_j^b\right)\right)\right)$$

$$K = S_\theta S_{g\circ f}^T \left(S_{g\circ f} S_{g\circ f}^T + \gamma C_d\right)^{-1}$$

⇩ Kalman-type localization

$$\theta_{\mathrm{j}}^{\mathrm{a}} = \theta_j^{\mathrm{b}} + (L \circ K)\left(d_j^o - g\left(f\left(\theta_j^b; m_j^b\right)\right)\right)$$

- θ contains hyper-parameters, which may not possess physical locations
- Correlation-based adaptive localization (AutoAdaLoc*) applied to construct the localization matrix L

*Luo, X., & Bhakta, T. (2020). Automatic and adaptive localization for ensemble-based history matching. *JPSE*, *184*, 106559.

Outline

NORCE

- Why CHOP?
- A proposed workflow to CHOP problems
- Numerical examples
- Summary and future plans

Numerical example 1

Experiment settings	
Type of DA problem	Sequential DA
DA algorithm (for analysis / model state estimation)	Stochastic ensemble Kalman filter (EnKF) with perturbed observations
Dynamical system (for forecast)	40-dimensional Lorentz 96 (testbed for sequential DA algorithms)

Numerical example 1

Stochastic EnKF

- Plain algorithm: $m_j^a = m_j^b + S_m S_d^T \left(S_d S_d^T + C_d\right)^{-1} \left(d_j^o \ - g\left(m_j^b\right)\right)$

- Auxiliary techniques often introduced for improved DA performance

Numerical example 1

Two Auxiliary techniques used in stochastic EnKF

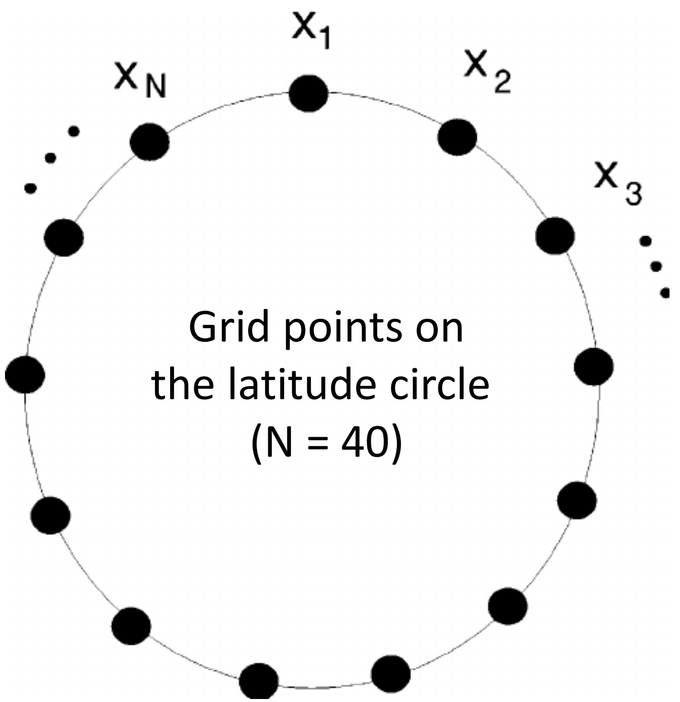
1. Covariance inflation (inflation factor δ): $m_j^b \rightarrow m_j^b + (1+\delta)(m_j^b - \bar{m}^b)$

2. Distance-based Kalman-gain localization (length scale λ):

$$K = S_m S_d^T (S_d S_d^T + C_d)^{-1} \rightarrow L(\lambda) \circ K \text{ (} \circ \text{ for Schur product)}$$

The elements of $L(\lambda)$ depend on the distances between the grid points on a latitude circle (N = 40 here) and λ.

λ corresponding to the critical length scale for localization



Numerical example 1

Stochastic EnKF

- Original algorithm: $m_j^a = m_j^b + S_m S_d^T (S_d S_d^T + C_d)^{-1} \left(d_j^o - g(m_j^b)\right)$

- Modified algorithm:

$$m_j^a = m_j^b + (1+\delta)(m_j^b - \overline{m}^b) + K(\delta,\lambda)\left(d_j^o - g\left(m_j^b + (1+\delta)(m_j^b - \overline{m}^b)\right)\right)$$

$$K(\delta,\lambda) = L(\lambda) \circ [(1+\delta)S_m S_d(\delta)^T (S_d(\delta) S_d(\delta)^T + C_d)^{-1}]$$

Numerical example 1

Performance of the modified stochastic EnKF

Grid search method used to tune (δ, λ) such that the updated model state best <u>matches the true model state</u>

Unrealistic in practice, but can be used as a reference

versus

An ensemble (δ, λ) to be estimated by the CHOP workflow to reduce data mismatch

Realistic and applicable in practice

Numerical example 1

Part of the results obtained by the grid search method (averaged over 20 repetitions)

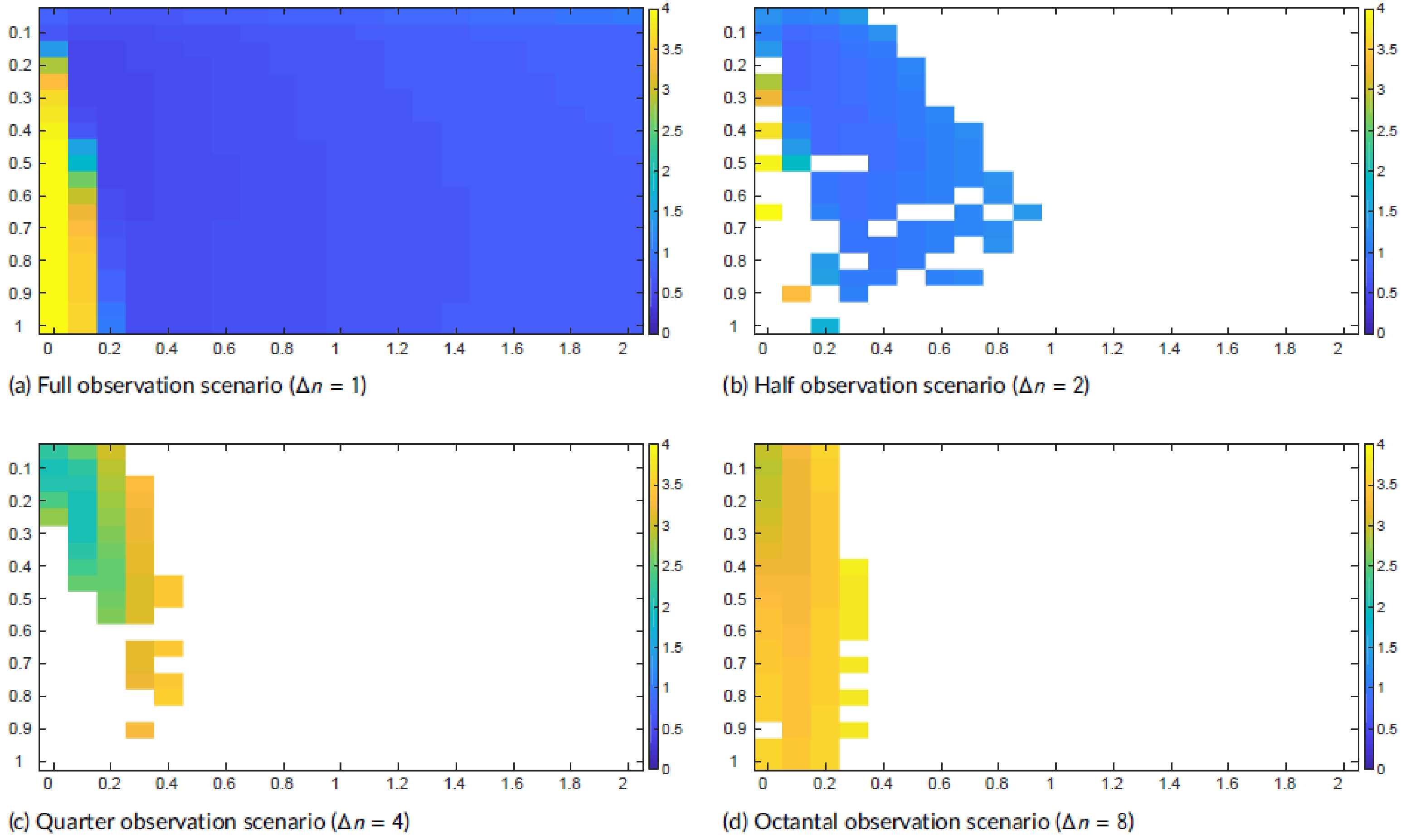
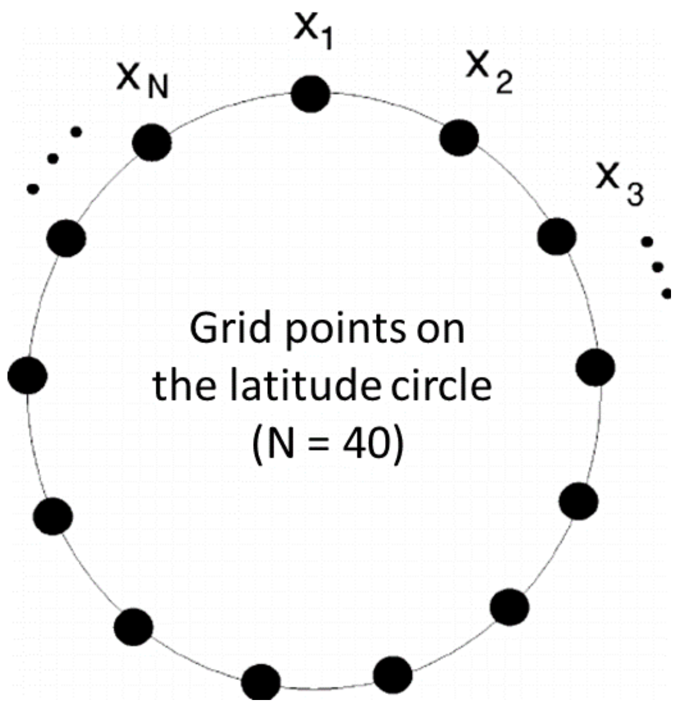


FIGURE 2 As in Figure 1, but for average RMSEs obtained by the grid search method in the half ($\Delta n = 2$, $N^{freq} = 4$), quarter ($\Delta n = 4$, $N^{freq} = 4$) and octantal ($\Delta n = 8$, $N^{freq} = 4$) observation scenarios, respectively, with the ensemble sizes $N_e = 30$. For ease of comparison, the results of the full observation scenario ($\Delta n = 1$, $N^{freq} = 4$, $N_e = 30$) in Figure 1 are re-plotted here.



Numerical example 1

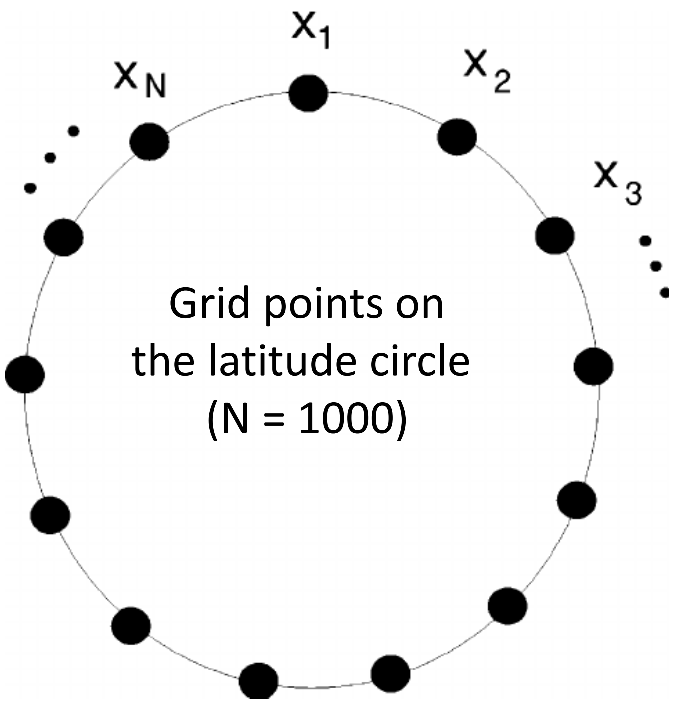
Result comparison: grid search (best results) vs CHOP

TABLE 2 As in Table 1, but for performance comparison between the grid search method and the CHOP workflow with full, half, quarter and octantal observations, respectively, whereas the ensemble size and the observation frequency are set to 30 and 4, respectively, in all experiments.

Observation density	Grid search		CHOP
	Minimum average RMSE (mean ± STD)	$(\delta_{min}, \lambda_{min})$	Average RMSE (mean ± STD)
Full ($\Delta n = 1$)	0.4560 ± 0.0100	(0.10, 0.20)	0.4766 ± 0.0096
Half ($\Delta n = 2$)	0.7975 ± 0.0257	(0.10, 0.20)	0.8763 ± 0.0418
Quarter ($\Delta n = 4$)	2.0100 ± 0.0773	(0.10, 0.25)	2.3596 ± 0.1248
Octantal ($\Delta n = 8$)	2.9129 ± 0.0353	(0.05, 0.10)	3.2437 ± 0.0419

Numerical example 2

Experiment settings	
Type of DA problem	Sequential DA
DA algorithm (for analysis / model state estimation)	Stochastic ensemble Kalman filter (EnKF) with perturbed observations
Dynamical system (for forecast)	1000-dimensional Lorentz 96 (testbed for sequential DA algorithms)



For illustration, we will consider a much larger set of hyper-parameters

Numerical example 2

NORCE

Stochastic EnKF

- Original algorithm: $m_j^a = m_j^b + S_m S_d^T (S_d S_d^T + C_d)^{-1} \left(d_j^o - g(m_j^b)\right)$

- Modified algorithm 1: single-inflation-factor δ (SIF) as in numerical example 1

$$m_j^a = m_j^b + (1+\delta)(m_j^b - \bar{m}^b) + K(\delta,\lambda)\left(d_j^o - g\left(m_j^b + (1+\delta)(m_j^b - \bar{m}^b)\right)\right)$$

$$K(\delta,\lambda) = L(\lambda) \circ [(1+\delta) S_m S_d(\delta)^T (S_d(\delta) S_d(\delta)^T + C_d)^{-1}]$$

Numerical example 2

Stochastic EnKF

- Modified algorithm 2: multiple-inflation-factor Δ (MIF) as in numerical example 1

$$\tilde{m}_j^b = m_j^b + (1 + \Delta) \circ \left(m_j^b - \bar{m}^b\right)$$
$$m_j^b = \left[m_{1,j}^b, m_{2,j}^b, \ldots, m_{1000,j}^b\right]^{\mathrm{T}}$$
$$\Delta = [\delta_1, \delta_2, \ldots, \delta_{1000}]^{\mathrm{T}}$$

$$m_j^a = \tilde{m}_j^b + K(\delta, \lambda)\left(d_j^o - g\left(\tilde{m}_j^b\right)\right)$$
$$K(\delta, \lambda) = L(\lambda) \circ \left[\tilde{S}_m \tilde{S}_d^T \left(\tilde{S}_d \tilde{S}_d^T + C_d\right)^{-1}\right]$$

Numerical example 2

Algorithms in comparison	EnKF + Grid search	EnKF + SIF + localization	EnKF + MIF + localization
Hyper-parameters	Single-inflation-factor (SIF): δ Localization length scale: λ Total No.: 2	Single-inflation-factor (SIF): δ Localization length scale: λ Total No.: 2	Multiple-inflation-factor (MIF): Δ Localization length scale: λ Total No.: 1001

Numerical example 2

TABLE 4 Performance comparison between the grid search method and the CHOP workflow in the 1000-dimensional L96 model.

Grid search		CHOP (SIF)	CHOP (MIF)
Minimum average RMSE (mean ± STD)	$(\delta_{min}, \lambda_{min})$	Average RMSE (mean ± STD)	Average RMSE (mean ± STD)
2.7667 ± 0.0099	(0.10, 0.05)	3.4213 ± 0.0552	3.0264 ± 0.0116

Outline

- Why CHOP?
- A proposed workflow to CHOP problems
- Numerical examples
- Summary and future plans

Summary

- The CHOP workflow can be used to:
  - ❑ estimate continuous hyper-parameters for a given DA algorithm
  - ❑ handle a large amount of hyper-parameters

- The CHOP workflow cannot be directly used to:
  - ❑ develop new DA algorithms
  - ❑ design new auxiliary techniques

The CHOP workflow enables the development of more sophisticated DA algorithms (not necessary ensemble-based)

Future work

- Applications to reservoir DA problems:
  - ❑ Modified iterative ensemble smoother (IES) algorithms
  - ❑ Estimate length scales in correlation-based adaptive localization schemes

Q&A

NORCE

X. Luo acknowledges financial supports from the NORCE research project “Assimilating 4D Seismic Data: Big Data Into Big Models” which is funded by industry partners, Equinor Energy AS, Lundin Energy Norway AS, Repsol Norge AS, Shell Global Solutions International B.V., TotalEnergies EP Norge AS, and Wintershall Dea Norge AS, as well as the Research Council of Norway (project number: 295002).

C. Xia acknowledges financial supports from the National Nature Science Foundation of China (Grant No. 42002247) and the Nature Science Foundation of Guangdong Province, China (Grant No. 2020A1515111054).