3DVar with Lorenz'63

Testing a 3DVar twin experiment

Jan Stüwe, Pavitra Chandarana

July 25, 2025

Overview

Lorenz '63

An example of deterministic chaos in low-dimensional systems

The Lorenz system was originally derived as a simplified model for atmospheric convection. It exhibits sensitive dependence on initial conditions.

Lorenz equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x),\\ \dot{y} &= x(\rho - z) - y,\\ \dot{z} &= xy - \beta z\end{aligned}$$

Standard parameters:

$$\sigma = 10, \quad \rho = 28, \quad \beta = \frac{8}{3}$$

The Setup

Simulating the True Trajectory ("Nature Run")

We integrate the Lorenz '63 system over the time interval $[0, 200]$ using a fixed time step $\Delta t = 0.01$ with the classical RK4 method:

- Initial condition: $\mathbf{x}_0 = [1.0,\ 1.0,\ 1.0]$
- Number of steps: $n_{\text{truth}} = \frac{T}{\Delta t} = 20000$
- Time vector: `org_time` $\in [0,\ 200]$ with 20000 points

3D Plot

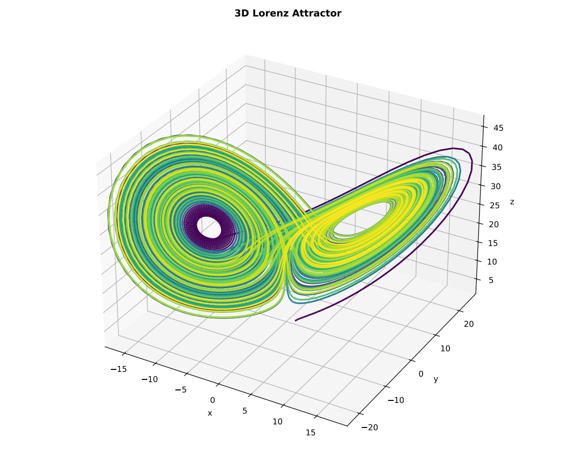

Figure 1: Lorenz attractor in phase space

Observation Generation with Noise

$$\text{Observation interval:} \quad \Delta t_{\text{obs}} = 0.02 \quad (\texttt{stride} = 2)$$

$$\text{Observation operator:} \quad H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{Noisy observations:} \quad \mathbf{y}_{\text{obs}} = H\mathbf{x}_{\text{true}} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1)$$

These noisy observations will later be used as “real-world observations” for 3DVar and learning tasks in the full 3D state space.

Ensemble Forecast & Covariance Estimation

- **Ensemble generation:**

$$N_{\mathrm{ens}} = 200, \quad \mathbf{x}_0^{(i)} \sim \mathcal{U}([-20, 20]^3), \quad i = 1, \dots, N_{\mathrm{ens}}.$$

- **Free runs:**

$$\mathbf{x}_{0:T}^{(i)} = \mathrm{batch\_integrate}\big(\mathbf{x}_0^{(i)},\, n_{\mathrm{truth}}\big), \quad \mathrm{free\_runs} \in \mathbb{R}^{N_{\mathrm{ens}} \times (T+1) \times 3}.$$

- **Ensemble mean & RMSE over time:**

$$\bar{\mathbf{x}}_t = \frac{1}{N_{\mathrm{ens}}} \sum_{i=1}^{N_{\mathrm{ens}}} \mathbf{x}_t^{(i)}, \qquad \mathrm{RMSE}_t = \sqrt{\frac{1}{N_{\mathrm{ens}}} \sum_{i=1}^{N_{\mathrm{ens}}} \|\mathbf{x}_t^{(i)} - \bar{\mathbf{x}}_t\|^2}.$$

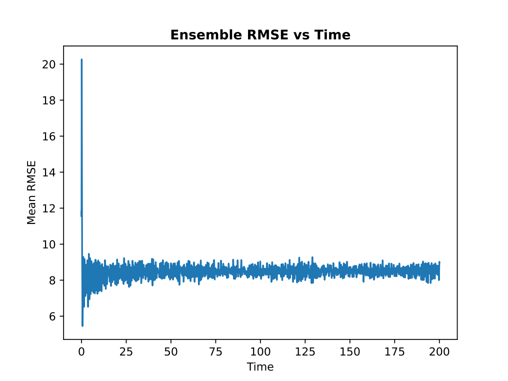

Figure 2: Ensemble Root Mean Squared Error

Covariance Matrix B:

$$B \approx \begin{pmatrix} 62.4 & 62.4 & 0.1 \\ 62.4 & 80.8 & 0.1 \\ 0.1 & 0.1 & 74.3 \end{pmatrix}$$

$$X' = \begin{bmatrix} x_T^{(1)} - \bar{x}_T \\ \vdots \\ x_T^{(N_{\mathrm{ens}})} - \bar{x}_T \end{bmatrix}$$

$$B = \frac{1}{N_{\mathrm{ens}} - 1} X'^{\top} X'$$

Twin Experiment

- **Observation error covariance:**

$$R = \sigma_o^2 I_3, \quad R^{-1} = \frac{1}{\sigma_o^2} I_3$$

- **Constant Kalman gain:**

$$K = BH^T \left(HBH^T + R\right)^{-1}$$

- **Forecast & Analysis cycle:**
  1. *Forecast:* propagate background state $x_b^{(k-1)} \to x_b^{(k)} = \mathcal{M}\left(x_a^{(k-1)}\right)$
  2. *Innovation:* $d^{(k)} = y^{(k)} - H\, x_b^{(k)}$
  3. *Analysis update:* $x_a^{(k)} = x_b^{(k)} + K\, d^{(k)}$

- **Outputs:** series of backgrounds x_b, analyses x_a, innovations d, increments $x_a - x_b$.

MLP-Based Increment Correction

- **Dataset construction:**

$$X^{(k)} = [x_b^{(k)}, y^{(k)}] \in \mathbb{R}^6, \quad \Delta^{(k)} = x_{\text{true}}^{(k)} - x_b^{(k)} \in \mathbb{R}^3$$

  split into training (80%) and validation (20%).

- **Model architecture:**

$$\underbrace{6}_{\text{input}} \to 64 \to 64 \to 32 \to 16 \to \underbrace{3}_{\text{output}} \quad \text{with ReLU activations}$$

- **Training setup:**
  - Optimizer: Adam, $\mathrm{lr} = 10^{-3}$
  - Loss: MSELoss
  - Early stopping: patience=50 epochs
- **Result:** monitor $\mathrm{MSE}_{\text{val}}$, report best validation score (≈ 0.36).

RMSE Comparison: 3D-Var vs. MLP Correction

- **Loop over observations:**
  1. *Forecast:* $x_b^{(k)} = \mathcal{M}\left(x_a^{(k-1)}\right)$
  2. *3D-Var increment:* $\delta_{3D}^{(k)} = K\left(y^{(k)} - H\,x_b^{(k)}\right)$
  3. *MLP increment:* $\delta_{ML}^{(k)} = \mathrm{MLP}\left[x_b^{(k)}, y^{(k)}\right]$
  4. *Analysis updates:* $\begin{cases} x_a^{3D,(k)} = x_b^{(k)} + \delta_{3D}^{(k)} \\ x_a^{ML,(k)} = x_b^{(k)} + \delta_{ML}^{(k)} \end{cases}$
  5. *Error metric:* $\mathrm{SE}_*^{(k)} = \left\| x_a^{*,(k)} - x_{\mathrm{true}}^{(k)} \right\|$
  6. *Update forecast state:* $x_b^{(k+1)} = x_a^{3D,(k)}$
- **Results:**

$$\overline{\mathrm{L}^2}_{ML} \approx 0.9, \quad \overline{\mathrm{L}^2}_{3D} \approx 1.5$$

Framework for RMSE Comparison

- **General cycle:**

$$\text{run\_cycle}(x_0, \mathit{gain}) : \begin{cases} x_b^{(k)} = \mathcal{M}\left(x_a^{(k-1)}\right) \\ x_a^{(k)} = \mathit{gain}(x_b^{(k)}, y^{(k)}) \\ \mathrm{L}^{2(k)} = \|x_a^{(k)} - x_{\mathrm{true}}^{(k)}\| \end{cases}$$

- **Gain functions:**

  classic_gain 3D-Var: $x_a = x_b + K\left(y - Hx_b\right)$

  ml_gain ML-predicted: $x_a = x_b + \delta_{\mathrm{ml}}(x_b, y)$

- **Experiment:** initialize x_0 poorly, run both gains, record RMSE over assimilation cycles

3D Plot

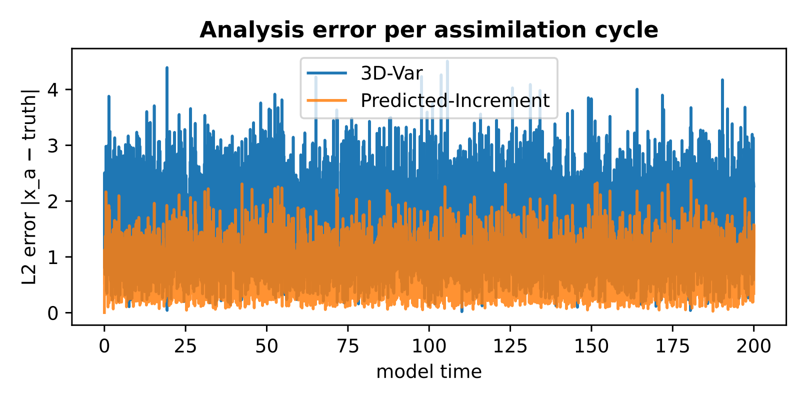

Figure 3: Analysis Error: MLP vs 3DVar

Sensitivity to Observation Error Variance

- **Setup:** for each $\sigma_r^2 \in \{1, 2, 5\}$,

$$R = \sigma_r^2 I_3, \quad y^{(k)} = x_{\text{true}}^{(k)} + \varepsilon, \ \varepsilon \sim \mathcal{N}(0, R)$$

- **Experiment:**
  1. Run `run_cycle` for each method from same x_0
  2. Record mean RMSE over assimilation cycles
- **Analysis:** Compare mean L^2 error vs. R to assess robustness of each method.

3D Plot

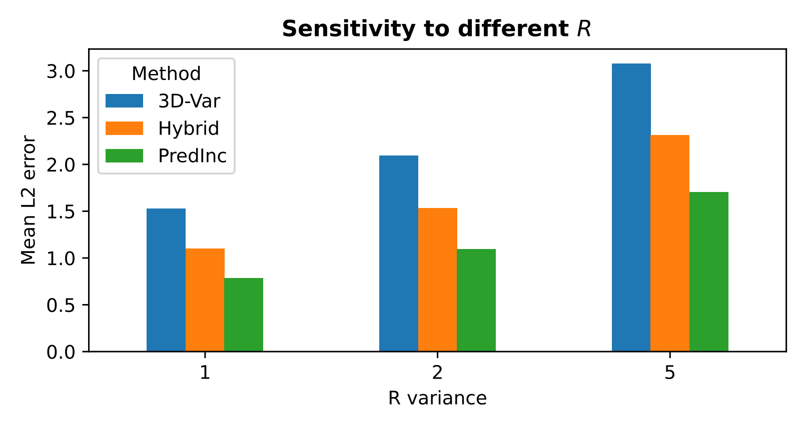

Figure 4: Analysis Error: MLP vs 3DVar vs Hybrid

Partial-Observation Assimilation

- **Observation model:**

$$H = [1\ 0\ 0], \quad R = \sigma_o^2, \quad y^{(k)} = x_{\mathrm{true}}^{(k)} + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, R)$$

- **Kalman gain (3×1):**

$$K = B\,H^T \big(H\,B\,H^T + R\big)^{-1}$$

- **Assimilation loop:**
  1. Forecast: $x_b^{(k)} = \mathcal{M}(x_a^{(k-1)})$
  2. Innovation: $d^{(k)} = y^{(k)} - H\,x_b^{(k)}$
  3. Analysis: $x_a^{(k)} = x_b^{(k)} + K\,d^{(k)}$
- **Visualization:** 3D plot of truth vs. partial-obs analysis trajectory.

Thank You!

Questions and Discussion Welcome