CS6404: Computational Data Assimilation

Project 3 - 4D-Var

Sam Donald

December 7, 2022

1 Project Outline

This project applies the 4D-Var data assimilation method to the Lorenz-63 system, whereby initial conditions are computed through minimizing the negative log posterior distribution, conditioned by observations of the systems first two states. The report is structured as follows, section 2 provides an overview of the 4D-Var data assimilation process along with verification of its components. The 4D-Var method is then evaluated within section 3 and discussed within section 4. The process of generating the Lorenz-63 trajectories, along with descriptions of parameters used, are additionally provided within appendix A.

2 Applied Methods

2.1 4D-Var Machinery

2.1.1 Runge-Kutta

A Runge-Kutta iterative method was developed using a fixed step size $h = dt/200$ and $s = 4$ in accordance with eqs. (1) to (3), using coefficients described by the Butcher tableau outlined by eq. (4) corresponding with a traditional RK4 method.

$$T_{k;i} = t_{k-1} + c_i h_k \tag{1}$$

$$\boldsymbol{Y}_{k;i} = x_{k-1} + h_k \sum_{j=1}^{s} a_{i,j} \boldsymbol{f}(T_{k;j}, \boldsymbol{Y}_{k;j}), \quad i = 1, ..., s; \tag{2}$$

$$x_k = x_{k-1} + h_k \sum_{j=1}^{s} b_i \boldsymbol{f}(T_{k;j}, \boldsymbol{Y}_{k;j}) \tag{3}$$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \tag{4}$$

This Runge-Kutta implementation can be found within the associated $rk4.m$ file, and is validated in section 2.2.

2.1.2 Adjoint computations

Tangent linear models (TLMs) describe how infinitesimally small perturbations of a systems initial state propagate forward in time. The adjoint of this TLM is instead used to map a sensitivity gradients backwards in time. The adjoint variable evolution can be constructed as the adjoint of the specific numerical discretization method applied to the forward model, and is called the discrete adjoint method. For the RK4 discretization method implemented, it can be shown that the adjoint solution is moved backwards in time from t_k to t_{k-1} by eqs. (5) and (6).

$$\boldsymbol{u}_i = h_k \boldsymbol{f}_x^T(T_{k;i}, \boldsymbol{Y}_{k;j}) \cdot (b_i \lambda_k + \sum_{j=1}^{s} a_{j,i} \boldsymbol{u}_j), \quad i = s, ..., 1; \tag{5}$$

$$\lambda_{k-1} = \lambda_k + \sum_{j=1}^{s} \boldsymbol{u}_j \tag{6}$$

This discrete adjoint method is found within the associated $discrete\_adj\_rk4.m$ file.

2.1.3 Cost function and its gradients

Direct differentiation of the 4D-Var cost function, eq. (7), provides the gradient described by eq. (8).

$$\psi(x_0) = \frac{1}{2}\|x_0 - x_0^b\|^2_{\boldsymbol{B_0^{-1}}} + \frac{1}{2}\sum_{i=0}^{N}\|\mathcal{H}_i(x_i) - y_i^{obs}\|^2_{\boldsymbol{R_i^{-1}}} \tag{7}$$

$$\nabla_{x_0}\psi(x_0) = \boldsymbol{B}_0^{-1}(x_0 - x_0^b) + \sum_{i=0}^{N} \boldsymbol{M}_{0,i}^T \boldsymbol{H}_i^T \boldsymbol{R}_i^{-1}(\mathcal{H}_i(x_i) - y_i^{obs}) \tag{8}$$

This gradient can be efficiently computed through running the adjoint model, eqs. (5) and (6), in accordance with eqs. (9) and (10) to assimilate observations. The the background term is then accounted for within equation eq. (11) to provide the final computed gradient.

$$\lambda_N = \boldsymbol{H}_i^T \boldsymbol{R}_N^{-1}(\mathcal{H}_N(x_N) - y_N^{obs}) \tag{9}$$

$$\lambda_{i-1} = \boldsymbol{M}_{i-1,i}^T \lambda i + \boldsymbol{H}_i^T \boldsymbol{R}_i^{-1}(\mathcal{H}_i(x_i) - y_i^{obs}) \tag{10}$$

$$\nabla_{x_0}\psi(x_0) = \boldsymbol{B}_0^{-1}(x_0 - x_0^b) + \lambda_0 \tag{11}$$

The process of computing the graditent of the cost function is summarized by algorithm 1, validated in section 2.2, and found within the $FourD\_Var.m$ file.

2.1.4 4D-Var optimization

Using the gradient of the cost function ψ, the initial condition x_0 is updated to minimize ψ via MATLABs inbuilt unconstrained optimization method $fminunc$. This optimization is repeated using an updated background condition from the previous round of optimizations until a satisfactory solution is found.

Algorithm 1 Adjoint algorithim for calculating gradients in 4D-Var data assimilation

Input: Initial condition x_0, Observations $(t_i, y_i, \boldsymbol{R}_i)_{0 \leq i \leq N}$
Output: Cost function gradient $\nabla_{x_0} \psi$

for $i = 1$ to N **do** ▷ For each time interval
  for $k = 1$ to K **do** ▷ Run forward Runge-Kutta solver
    Run one step of Runge-Kutta solver to advance solution

$$x_{k-1} \mapsto x_k$$

    Save solution x_{k-1} and all stage variables
  end for
end for

Initialize adjoint model

$$\lambda_N^- = \boldsymbol{H}_N^T \boldsymbol{R}_N^{-1} (\mathcal{H}_N(x_N) - y_N^{obs})$$

for $i = N$ down to 1 **do** ▷ For each time interval
  for $k = K$ down to 1 **do** ▷ Run discrete adjoint Runge-Kutta solver
    Load forward solution $x_{k-1;i}$ and all stage variables

    Run adjoint model according to eqs. (5) and (6)

$$\lambda_k \mapsto \lambda_{k-1}$$

  end for
  Add observation forcing term at t_{i-1}

$$\lambda_{i-1}^- = \lambda_{i-1}^+ + \boldsymbol{H}_i^T \boldsymbol{R}_i^{-1} (\mathcal{H}_i(x_i) - y_i^{obs})$$

end for

Account for background term

$$\nabla_{x_0} \psi = \boldsymbol{B}_0^{-1} (x_0 - x_0^b) + \lambda_0^-$$

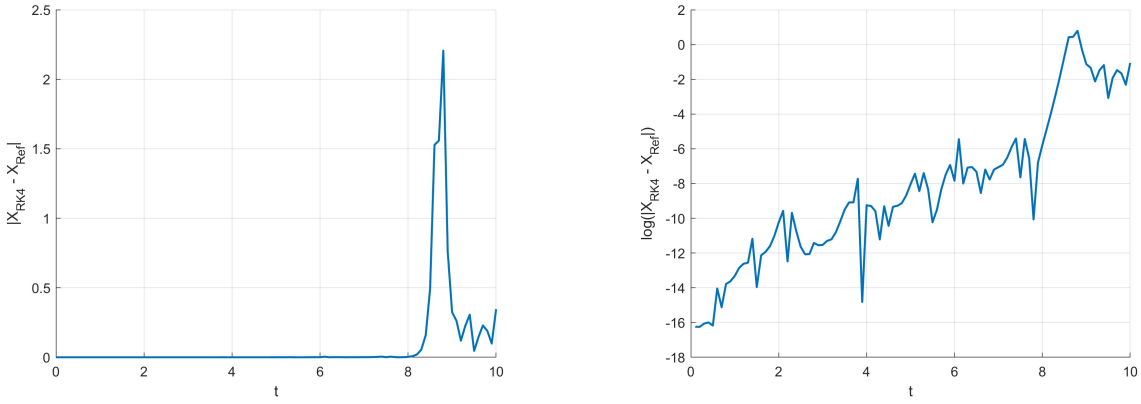

Figure 1: Absolute error (left) and log of error (right) between x^{Ref} and x^{RK4}

2.2 Validation

2.2.1 Runga-Kutta

The Runge-Kutta method was validated through comparison with MATLABs inbult ODE45 method, with errors displayed within fig. 1.

2.2.2 Cost function and its gradients

The adjoint gradient computation is validated through running 4 separate computations. One with inital condition x_0, and three additional runs with a perturbation ϵ applied to the k-th entry of the initial condition x_0. The k-th component of the adjoint gradient is then compared to the finite difference approximation outlined by equation eq. (12), with results and percentage errors displayed in table 1.

$$\frac{d\psi}{dx_{0;k}} = (\nabla_{x_0}\psi(x_0))_k \approx \epsilon^{-1}(\psi(x_0 + \epsilon \boldsymbol{e}_k) - \psi(x_0)) \tag{12}$$

Term	**Finite Difference Grad**	**ADJ Grad**	**Percent Error**
$\boldsymbol{x}_{0,k=1}$	40.7848	40.7758	0.022
$\boldsymbol{x}_{0,k=2}$	-85.7761	-85.7887	-0.0147
$\boldsymbol{x}_{0,k=2}$	-63.6810	-63.6838	-0.0043

Table 1: Term-wise adjoint gradient comparison

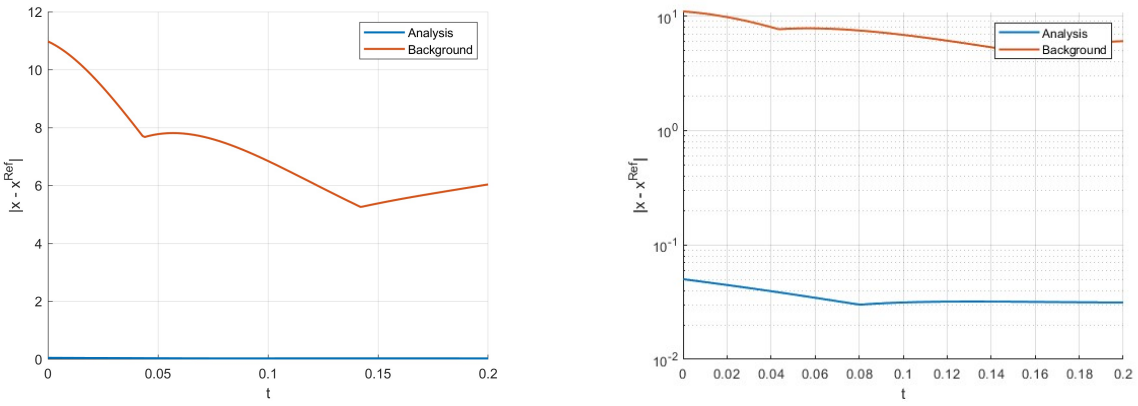

Figure 2: Absolute error (left) and log scaled error (right) compared to reference

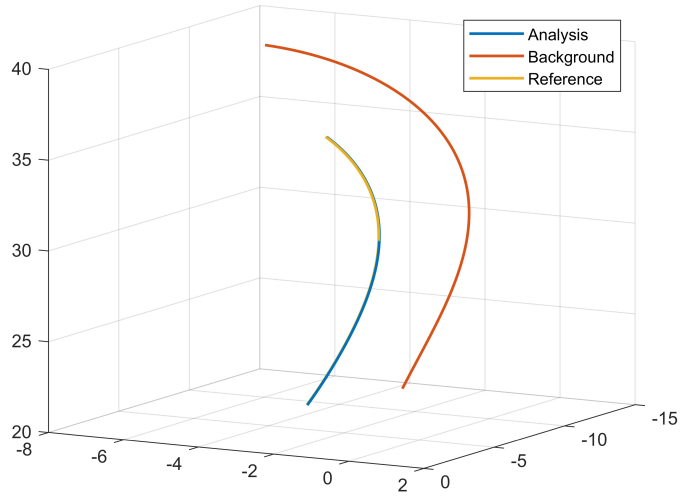

Figure 3: Background, analysis, and reference trajectories (note analysis and reference overlap)

3 Results

Using an assimilation window of $[t_0, t_N]$ with $N = 2$, ($[0, 0.2]$), synthetic observations were generated in accordance with section A. The inital background state x_0^b was drawn from a normal distribution $\mathcal{N}(x_0, \boldsymbol{B}_0)$, and used to generate the background trajectory $x_{0:N}^b$in accordance with the forward model. The 4D-Var optimization process outlined by algorithm 1 was then used to optimize the inital state resulting in the analysis inital state x_0^a.

High resolution trajectories were generated using the updated x_0^a, x_0^b, and x_0^{ref} and displayed within fig. 3. The background and analysis trajectories errors with respect to the reference solution are additionally provided within fig. 2.

4 Discussion

4.1 Overall performance

As displayed within fig. 3, the analysis trajectory very closely matches that of the reference trajectory and is a significant improvement over the background trajectory. This high level of accuracy is further confirmed by fig. 2, where the analysis trajectory maintains a absolute error of less than 4×10^{-2} compared to the reference trajectory across the assimilation window.

4.2 RK4

The RK4 method implemented is deemed to be accurate based on the small errors with respect to to the reference solution generated by MATLABs ODE45 solver, displayed within fig. 1. The RK4 method maintains an absolute error below 10^{-6} prior to time $t = 8$, where the error rises sharply. This along with the gradual increase in error over time is expected for a chaotic system such as Lorenz-63, as small differences within system states diverge exponentially over time.

4.3 Cost Function Gradients

When comparing gradients generated via adjoint method and the finite difference approximation, the percentage error for all states is below 0.025%. The adjoint gradient is therefore deemed to be sufficiently accurate.

Appendix A Lorenz-63 Trajectory Configuration

A.1 Model description

The Lorenz-63 model is a dynamical system commonly used to benchmark various data assimilation methods, in part due to its chaotic nature. Trajectories generated by this system are used throughout this project, and are summarized by eqs. (13) to (15).

$$\frac{dx}{dt} = \sigma(y - x) \tag{13}$$

$$\frac{dy}{dt} = x(\rho - z) - y \tag{14}$$

$$\frac{dz}{dt} = xy - \beta z \tag{15}$$

A.2 Model parameters

Parameter values used to generate the various Lorenz-63 trajectories are outlined within table 2.

Parameter	Value	Description
N_{state}	3	Number of states
N_{obvs}	2	Number of observed states
N	2	Assimilation window size
H	[1, 1, 0]	Observation operator
Δt	0.1	Time step size
h	$\frac{\Delta t}{200}$	RK4 step size
r	0.025	Observation noise
R	$r^2 I_{N_{obs}}$	Observation error matrix
x_0	$\begin{bmatrix} -10.0375 \\ -4.3845 \\ 34.6514 \end{bmatrix}$	Inital system state
$\boldsymbol{B}_0$	$\begin{matrix} 12.4294 & 12.4323 & -0.2139 \\ 12.4294 & 16.0837 & -0.0499 \\ -0.2139 & -0.0499 & 14.7634 \end{matrix}$	Climatological covariance matrix
σ	10	L63 parameter
ρ	28	L63 parameter
β	8/3	L63 parameter

Table 2: Descriptions of parameters used within project

A.3 Trajectory generation process

A.3.1 Observation trajectory

A reference trajectory was generated by running the forward model over the interval $[0, T_{max}]$ using initial conditions $\mathbf{x}_0^{true}$, with data saved every Δt time units. The observation operator was then applied to the reference trajectory (selecting the odd states), before adding random noise drawn from $\mathcal{N}(0, R)$ resulting in the observation trajectory.