IMA Journal of Applied Mathematics (2022) **87**, 935–963
https://doi.org/10.1093/imamat/hxac026
Advance Access publication on 20 October 2022

The maximum likelihood ensemble smoother for the Kuramoto–Sivashinsky equation

Christopher Hurst*
Computational Fluid Dynamics and Propulsion Laboratory, Colorado State University, Fort Collins, CO 80523, USA
*Corresponding author: hurstcl@colostate.edu

Milija Zupanski
Cooperative Institution for Research in the Atmosphere, Colorado State University, Fort Collins, CO 80523, USA

and

Xinfeng Gao
Computational Fluid Dynamics and Propulsion Laboratory, Colorado State University, Fort Collins, CO 80523, USA

[Received on 28 July 2021; revised on 19 August 2022; accepted on 5 September 2022]

Data assimilation (DA) aims to combine observations/data with a model to maximize the utility of information for obtaining the optimal estimate. The maximum likelihood ensemble filter (MLEF) is a sequential DA method or a filter-type method. Weaknesses of the filter method are assimilating time-integrated observations and estimating empirical parameter estimation. The reason is that the forward model is employed outside of the analysis procedure in this type of DA method. To overcome these weaknesses, the MLEF is now extended as a smoother and the novel maximum likelihood ensemble smoother (MLES) is proposed. The MLES is a smoothing method with variational-like qualities, specifically in the cost function. Rather than using the error information from a single temporal location to solve for the optimal analysis update as done by the MLEF, the MLES can include observations and the forward model within a chosen time window. The newly proposed DA method is first validated by a series of rigorous and thorough performance tests using the Lorenz 96 model. Then, as DA is known to be used extensively to increase the predictability of the commonly chaotic dynamical systems seen in meteorological applications, this study demonstrates the MLES with a model chaotic problem governed by the 1D Kuramoto–Sivashinky (KS) equation. Additionally, the MLES is shown to be an effective method in improving the estimate of uncertain empirical model parameters. The MLES and MLEF are then directly compared and it is shown that the performance of the MLES is adequate and that it is a good candidate for increasing the predictability of a chaotic dynamical system. Future work will focus on an extensive application of the MLES to highly turbulent flows.

Keywords: Data assimilation; maximum likelihood ensemble smoother; maximum likelihood ensemble filters; chaos; Kuramoto–Sivashinsky equation; fluid dynamics.

1. Nomenclature

CFD	Computational fluid dynamics
KS	Kuramoto–Sivashinsky
MLES	Maximum Likelihood Ensemble Smoother
MLEF	Maximum Likelihood Ensemble Filter
NWP	Numerical weather prediction
PDE	Partial differential equation
PDF	Probability density function
KS	Kuramoto–Sivashinsky
$J(\cdot)$	Cost function
$\hat{\mathbf{Q}}_n$	State vector at time n
$\hat{\mathbf{Q}}_n^f$	Forecast state vector at time n
$\mathbf{P}_f$	Forecast error covariance
N_{mles}	Observations taken in smoother window
$\mathbf{O}$	Synthesized observation data
η	Random perturbation for $\mathbf{O}$
σ_{obs}	Normal distribution obs. variance
$\mathscr{H}(\cdot)$	Observation operator
$\mathbf{R}$	Observation error covariance
$\mathscr{M}(\cdot)$	Non-linear forward model
ψ	Non-dimensional control variable
s	Number of control variable types
N_G	Dimension of grid space
N_S	Dimension of state space
k	Minimization step index
$\mathbf{p}_i^f$	Elements of $\mathbf{P}_f^{1/2}$
$\mathbf{Z}$	Notation for gradient and Hessian
$\mathbf{z}^i$	Elements of $\mathbf{Z}$
$\mathbf{g}$	Gradient of $J(\psi)$
$\mathbf{C}$	Hessian of $J(\psi)$
ξ	Used for change of variable in ψ
$\mathbf{c}^i$	Elements of $\mathbf{C}^{-\mathrm{T}/2}$
$\mathbf{P}_a$	Global analysis error covariance
$\mathbf{p}_i^a$	Elements of $\mathbf{P}_a^{1/2}$
a, b, c	Model parameters for KS equation

2. Introduction

The age-old issue of increasing the predictability in chaotic dynamical systems has long been a central part of research in many facets of physical and numerical modelling. Data assimilation (DA) is a mathematical discipline that seeks to enhance the ability to predict the long-term future using available data in conjunction with a model to enhance the model's prediction. Numerical weather prediction is a field in which DA is widely used (e.g. Kalnay (2003) and Asch *et al.* (2016)). In terms of chaotic behaviour of turbulent fluid dynamics, the models used in weather prediction share many similarities to

the models used in computational fluid dynamics (CFD) for engineering applications and thus a natural extension is to apply DA to CFD. In particular, reduced-order modelling of turbulence is a prominent area of research in the CFD community due to the impracticality of direct numerical simulations. Turbulence models such as the Reynolds-averaged Navier–Stokes or large-eddy simulations have been employed to solve these problems, but their accuracy for highly turbulent flows is limited due to the fact that the models and model parameters are dependent on the evolving flow conditions and, ideally, would need dynamic tuning. This dynamic tuning is a major advantage of DA because it uses all available data to perform statistical error analysis which is then used to tune the model state and parameters as the system evolves.

To this end, much research has been seen in this increasingly popular field of integrating DA with CFD (simply referred to as DA+CFD herein) (e.g. Gao & Liu (2014), Gao *et al.* (2015), Kato *et al.* (2015), Reagan *et al.* (2016), Gao *et al.* (2016), Gao *et al.* (2017), He *et al.* (2018), Hurst & Gao (2020), Wang *et al.* (2021), and Wang *et al.* (2020)). Specifically, considerable improvement has been made by way of the maximum likelihood ensemble filter (MLEF) (Carrassi *et al.* (2008), Wang *et al.* (2020)). Like most ensemble-based DA methods, the MLEF is a sequential, e.g. filtering method. While these methods can address most of the needs of DA, there are several weaknesses of filters that may adversely impact some applications. The major weakness is the assimilation of time-integrated observations, such as accumulated precipitation, for example (Cohn *et al.* (1994)). Another drawback is the empirical parameter estimation. This is particularly problematic for non-linear problems. The reason for impediment is that, in filters, the prediction model is employed outside of the analysis procedure. Much better equipped for such problems are the smoothers, in which the prediction model is an integral part of the analysis procedure. Particularly, smoothers adjust the initial conditions at the beginning of a time window that, unlike the MLEF, includes multiple observations and a necessary model integration within the window. Aside from mentioned benefits of smoothers over filters, there are some challenges for smoothers. One is related to nonlinearity. Since the prediction models are generally non-linear and they are included in the analysis procedure, it is critical to address nonlinearity in smoothers. The second challenge is due to increased cost of calculation since the use of the prediction model can add considerable computational overhead for smoothers. Further reading for the advancements in smoother DA methods can be found in many references (e.g. Evensen (2000), Chen & Oliver (2012), Bocquet & Sakov (2014), Posselt *et al.* (2014) and Evensen (2018)).

The objective of this study is to introduce such a smoother to address these challenges. The novel maximum likelihood ensemble smoother (MLES) is devised, implemented and tested in this study. The MLES transforms the MLEF (Zupanski (2005)) from filter to smoother and is then implemented in our existing DA+CFD system (Gao *et al.* (2017), Wang *et al.* (2021), Wang *et al.* (2020)). Subsequently, the MLES is verified and validated by the 1D Kuramoto–Sivashinsky (KS) equation in order to understand the potential advantages of the new algorithm. The KS equation is chosen because it is known to be chaotic and a good candidate to assess the performance of the proposed MLES. Additionally, the KS equation has previously been used to compare the ensemble Kalman filter, MLEF and a Monte Carlo particle filter (Jardak *et al.* (2010)) and provides a good foundation in comparison with other DA methods. Our overarching goal is to apply the valid MLES to improve the state predictions and the estimate of parameters of turbulence models in CFD for practical highly turbulent flows in engineering applications. However, this study is the inception of the long-term investigation.

In Section 3, the methodology behind the new algorithm is described. The difference between the smoother and filter is emphasized. Furthermore, a brief description of the forward model and the three main steps for DA are given; also the pseudocode for the MLES algorithm is given. Section 4 includes a rigorous derivation of the cost function and the process of minimization in order to obtain the optimum

state. Section 5 validates the MLES method using the Lorenz 96 model as a testbed. The results of the MLES and MLEF are presented and compared in Section 6 using the KS equation.

3. A brief overview of the methodology

This study transforms the MLEF to the MLES from a filter method to a smoother one. To put the transformation in context, the MLEF method is reviewed. Then, the methodology of the MLES is introduced.

3.1 *Overview of the MLEF*

The MLEF is based on the Bayesian framework which provides a systematic approach for model state estimate. Bayes theorem tells us that the posterior probability density function (PDF), $p(\hat{\mathbf{Q}}|\mathbf{O})$, is proportional to the prior PDF, $p(\hat{\mathbf{Q}})$, multiplied by the likelihood PDF $p(\mathbf{O}|\hat{\mathbf{Q}})$,

$$p(\hat{\mathbf{Q}}|\mathbf{O}) \propto p(\mathbf{O}|\hat{\mathbf{Q}})p(\hat{\mathbf{Q}}), \tag{3.1}$$

where $\mathbf{O}$ is the observation vector, and the control vector, $\hat{\mathbf{Q}}$, is a vector that contains all of the variables of interest, representing the optimal estimate of the state. Following the standard definition of multivariate Gaussian-type distributions, we assume both background (prediction) error and observation error are Gaussian PDFs, so that

$$p(\hat{\mathbf{Q}}) = N_{\hat{\mathbf{Q}}}\left(\hat{\mathbf{Q}}^f, \mathbf{P}_f\right) \propto \exp\left(-\frac{1}{2}\left(\hat{\mathbf{Q}}-\hat{\mathbf{Q}}^f\right)^{\mathrm{T}}\mathbf{P}_f^{-1}\left(\hat{\mathbf{Q}}-\hat{\mathbf{Q}}^f\right)\right), \tag{3.2}$$

and

$$p(\mathbf{O}|\hat{\mathbf{Q}}) = N_{\mathbf{H}(\hat{\mathbf{Q}})}(\mathbf{O}, \mathbf{R}) \propto \exp\left(-\frac{1}{2}\left(\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})\right)\right). \tag{3.3}$$

Therefore, the posterior PDF is given as

$$p(\hat{\mathbf{Q}}|\mathbf{O}) \propto \exp\left(-\frac{1}{2}\left(\hat{\mathbf{Q}}-\hat{\mathbf{Q}}^f\right)^{\mathrm{T}}\mathbf{P}_f^{-1}\left(\hat{\mathbf{Q}}-\hat{\mathbf{Q}}^f\right) - \frac{1}{2}\left(\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})\right)\right). \tag{3.4}$$

Approximating the maximum of the posterior $p(\hat{\mathbf{Q}}|\mathbf{O})$ involves finding the mode of the PDF curve, which leads to

$$\max\ p(\hat{\mathbf{Q}}|\mathbf{O}) = \max\ \alpha\exp\left(-J(\hat{\mathbf{Q}})\right) \Rightarrow \operatorname{argmin}\ J(\hat{\mathbf{Q}}), \tag{3.5}$$

where $J(\hat{\mathbf{Q}})$ is the cost function and defined as

$$J(\hat{\mathbf{Q}}) = \frac{1}{2}\left(\hat{\mathbf{Q}}-\hat{\mathbf{Q}}^f\right)^{\mathrm{T}}\mathbf{P}_f^{-1}\left(\hat{\mathbf{Q}}-\hat{\mathbf{Q}}^f\right) + \frac{1}{2}\left(\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})\right). \tag{3.6}$$

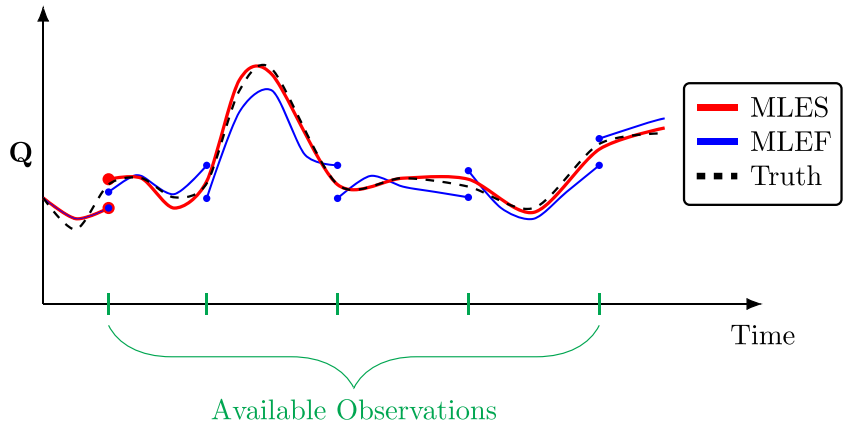

Fig. 1. Graphical description of the differences between the MLES and MLEF. The y-axis, $\mathbf{Q}$, represents just one state value in the entire spatial domain. The MLES is updated at the beginning of the smoother window which is the first location of available observations, represented by the green dashes. The forward model is employed in this entire region for the control and each additional member of the ensemble. The state of the MLEF is updated sequentially over multiple times at each available observation and the model is not included the analysis.

The MLEF is a low-rank ensemble-based method where the analysis is performed in ensemble space (Zupanski (2005), Zupanski *et al.* (2008)). The members in the ensemble dynamically propagate the errors of the system to each analysis time and the performance is highly relied upon the quality of the members.

3.2 *Overview of the MLES*

The inherent differences between the MLES and MLEF are (i) the forward model is employed within the analysis process in the MLES; (ii) the state is optimized for that of the beginning of the time window in the MLES. Figure 1 illustrates these differences. The continuous line for the MLES in the figure shows that the model is coupled in the analysis process. The red-filled circle, which is marked at the beginning of the time window, is the optimal state produced by the MLES, while the blue-filled circles are the optimal states at each observation time produced by the MLEF.

As in the MLEF method, three main steps in the MLES method includes initialization, forecast and analysis. The process is illustrated briefly by Fig. 2. First, initialization of the control vector and the ensemble must be performed in such a way that is realistic and applicable. This involves perturbing the control and the ensemble from the arbitrarily chosen initial condition, then performing a 'burn-in' period in which the forward model is run until each state is fully developed and indistinguishable from the initial condition, at which point the main DA algorithm is started. The next step is the forecast step. This entails applying the forward model to the start of the DA window and calculating the forecast parameters. The last step is the analysis step where the control vector and the ensemble are updated. The key ingredient for this step is the statistical error from the ensemble and forecast values and their statistical error information is necessary to update the control. Without this step, the chaotic nature of the models may eventually make the control and true run deviate. Then, once the analysis state has been achieved, the cycle of forecast and analysis steps are repeated.

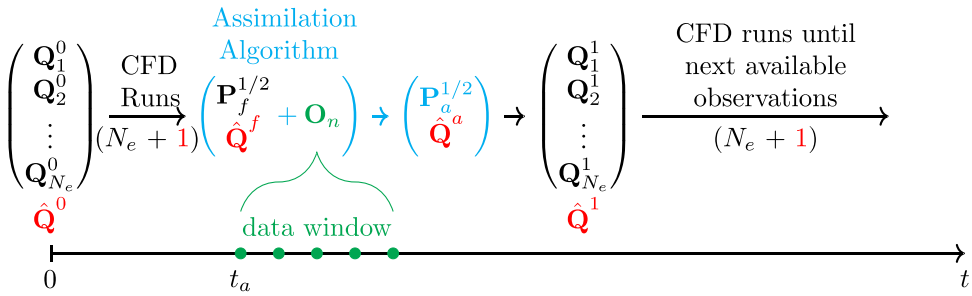

Fig. 2. The flowchart shows the DA +CFD process. The superscripts (e.g. 0, 1) indicate the DA cycle number except for $\mathbf{P}_f$ where the superscript 1/2 represents the square-root. The subscripts (e.g. $1, \ldots, N_e$) stand for the number of members in the ensemble. The subscripts or superscripts f and a represent forecast and analysis, respectively. t_a is the analysis time location. The red coloured $\hat{\mathbf{Q}}$ is the control vector.

The initialization of the truth, control and ensemble is based on the lagged forecast initialization (Kalnay (2003)) and involves perturbing the control and the ensemble from the arbitrarily chosen initial condition of the truth state. In this study, the numerical simulation regarded as the 'truth' is obtained assuming there is no uncertainty in the initial and boundary conditions as well as in the physical models and model parameters employed by the forward model. Once the truth run is initialized, the forward model propagates it to the desired solution time. After a 'burn-in' period, the main DA algorithm is started. The ensemble is generated by adding perturbations to reference conditions and the control is chosen to be the average of the ensemble. The forward model is denoted by $\mathscr{M}(\cdot)$ which in general can be a computational prediction model, an empirical relation or an analytical model. In this study, our forward model is the CFD model. Note that DA methods do not depend on $\mathscr{M}(\cdot)$.

In this study, the truth is used to synthesize the observations. Synthesized observations are commonly used to verify new DA methods, particularly when physical observations are not available, and in this study there are no physical observations. In our case, we synthesize the observations from the known truth using

$$\mathbf{O}_n = \mathscr{H}\left(\hat{\mathbf{Q}}_{\text{true}}\right) + \boldsymbol{\eta}, \tag{3.7}$$

where $\hat{\mathbf{Q}}_{\text{true}}$ is the state from the perfect CFD scenario, meaning that a particular initial condition is chosen and the resulting space-time solution is used to synthesize the observations. n is the time index, and $\boldsymbol{\eta}$ is a random perturbation taken from a normal distribution of a chosen variance, σ_{obs}, and mean of 0. The observation operator, $\mathscr{H}(\cdot)$, is generally non-linear and maps the state space to the observation space. Note that each element in the vector $\mathbf{O}_n$ receives a specific random perturbation from the result of the observation operator on the true state. The observations are saved and used later for the analysis step.

The states of the control and the ensemble are obtained by the forward model to the start of the first DA cycle. At this point, the square-root forecast error covariance is calculated and the current state is used as the forecast state. The forecast error covariance $\mathbf{P}_f$ is defined in ensemble subspace where the square-root forecast error covariance matrix, $\mathbf{P}_f^{1/2}$, is approximated as

$$\mathbf{P}_f^{1/2} = \left.\mathbf{Q}_i^f - \hat{\mathbf{Q}}^f\right|_{i=1,\cdots,N_e} = \mathbf{Q}_1^f - \hat{\mathbf{Q}}^f \;\; \mathbf{Q}_2^f - \hat{\mathbf{Q}}^f \;\; \cdots \mathbf{Q}_{N_e}^f - \hat{\mathbf{Q}}^f \quad , \tag{3.8}$$

where N_e is the number of members in the ensemble. $\hat{\mathbf{Q}}_i^f$ represents the entire vector of the state variables for the i-th member in the ensemble and $\hat{\mathbf{Q}}^f$ is the entire control vector of the forecast state of the MLES.

The main goal of the analysis step is to minimize the cost function by finding the optimum control state. The cost function, as described in the next section, is a mathematical measurement that represents the error between the current state, the observations and the forecast state. The newly found optimum state is then used to formulate the analysis error covariance which is used to update the members in the ensemble and then the forecast step is used to run the models to the next analysis step.

4. Mathematical formulation of the cost function and error covariance for the MLES

The cost function for the new MLES method begins by extending the existing MLEF method (Zupanski (2005), Zupanski *et al.* (2008)) to an ensemble smoother. Mathematically, one defines a cost function similar to that of a four-dimensional variational DA method, such as

$$J(\hat{\mathbf{Q}}_0) = \frac{1}{2}\left(\hat{\mathbf{Q}}_0 - \hat{\mathbf{Q}}_0^f\right)^{\mathrm{T}} \mathbf{P}_f^{-1}\left(\hat{\mathbf{Q}}_0 - \hat{\mathbf{Q}}_0^f\right) + \frac{1}{2}\sum_{n=1}^{N_{\mathrm{mles}}}\left(\mathbf{O}_n - \mathscr{H}\left(\mathscr{M}_0^n(\hat{\mathbf{Q}}_0)\right)\right)^{\mathrm{T}} \mathbf{R}_n^{-1}\left(\mathbf{O}_n - \mathscr{H}\left(\mathscr{M}_0^n(\hat{\mathbf{Q}}_0)\right)\right), \tag{4.1}$$

where n is the time index and there are N_{mles} observation times in the assimilation time window. $\mathbf{R}$ is the observation error covariance, and $\mathscr{H}(\cdot)$ is the observation operator as defined previously. The state vector $\hat{\mathbf{Q}}_0$ is defined at initial time of the window ($t = t_0$) and $\mathbf{O}_n$ is the observation vector at the specified time index. By updating the initial forecast state at the beginning of the DA time window, the cost function can be minimized, which will represent the algorithm's most optimal state based on the available observations. Note that

$$\mathscr{M}_0^n\left(\hat{\mathbf{Q}}_0\right) \tag{4.2}$$

denotes the non-linear prediction model integrated from time t_0 to time t_n. To reiterate, the MLEF only applies the analysis update at each instance in time where observations are available, whereas the MLES takes observations that are temporally spread out in a time window. Indeed, the difference between Eq. 3.6 and Eq. 4.1 is the lack of the summation term and the forward model application in the former.

4.1 *First change of variable*

In DA problems, the magnitude of the physical units inside the control vector can widely vary and imply a large condition number of the Hessian of the unmodified cost function. Therefore, a standard change of variable is commonly used in any DA method to remove the dependence on the large magnitude differences in the original problem. Specifically for the MLES, it transforms the control variable from dimensional ($\hat{\mathbf{Q}}_0$) to non-dimensional ($\boldsymbol{\psi}$), given by

$$\hat{\mathbf{Q}}_0 = \hat{\mathbf{Q}}_0^f + \mathbf{P}_f^{1/2}\boldsymbol{\psi}, \tag{4.3}$$

which implies that one can now distinguish between the state space (with dimension N_S) and the grid space (with size of N_G). The ensemble square-root forecast error covariance is typically written in

column-vector form

$$\mathbf{P}_f^{1/2} = \left[\mathbf{p}_1^f, \ldots, \mathbf{p}_{N_e}^f\right], \tag{4.4}$$

where each $\mathbf{p}_i^f : i = 1, \ldots, N_e$ is a column vector of dimension N_S. The transformed control variable definition implies that there is another cost function defined for $\boldsymbol{\psi}$

$$J(\boldsymbol{\psi}) = \frac{1}{2}\boldsymbol{\psi}^{\mathrm{T}}\boldsymbol{\psi} + \frac{1}{2}\sum_{n=1}^{N_{\mathrm{mles}}} \left[\mathbf{O}_n - \mathcal{H}\left[\mathcal{M}_0^n(\hat{\mathbf{Q}}_0^f + \mathbf{P}_f^{1/2}\boldsymbol{\psi})\right]\right]^{\mathrm{T}} \mathbf{R}_n^{-1} \left[\mathbf{O}_n - \mathcal{H}\left[\mathcal{M}_0^n(\hat{\mathbf{Q}}_0^f + \mathbf{P}_f^{1/2}\boldsymbol{\psi})\right]\right]. \tag{4.5}$$

Taking the gradient we get

$$\mathbf{g} = \boldsymbol{\psi} - \sum_{n=1}^{N_{\mathrm{mles}}} \mathbf{P}_f^{\mathrm{T}/2}\mathbf{H}_n^{\mathrm{T}}\mathbf{R}_n^{-1}\left[\mathbf{O}_n - \mathcal{H}\left[\mathcal{M}_0^n(\hat{\mathbf{Q}}_0^f + \mathbf{P}_f^{1/2}\boldsymbol{\psi})\right]\right], \tag{4.6}$$

where

$$\mathbf{H} = \frac{\partial \mathcal{H}(\hat{\mathbf{Q}})}{\partial \hat{\mathbf{Q}}}. \tag{4.7}$$

Now we take the gradient of the above to get the Hessian of $J(\boldsymbol{\psi})$,

$$\mathbf{C} = \mathbf{I} + \sum_{n=1}^{N_{\mathrm{mles}}} \mathbf{P}_f^{\mathrm{T}/2}\mathbf{H}_n^{\mathrm{T}}\mathbf{R}_n^{-1}\mathbf{H}_n\mathbf{P}_f^{1/2}. \tag{4.8}$$

Before introducing the optimal conditioning, at each time $t = t_n$ one can define the matrix $\mathbf{Z}_n(\mathbf{Q}_0)$

$$\mathbf{Z}_n(\mathbf{Q}_0) = \left[\mathbf{z}_n^1, \ldots, \mathbf{z}_n^{N_e}\right], \tag{4.9}$$

where

$$\mathbf{z}_n^i(\mathbf{Q}_0) = \mathbf{R}_n^{-1/2}\mathbf{H}_n\mathbf{P}_f^{1/2} \approx \mathbf{R}_n^{-1/2}\left[\mathcal{H}\left[\mathcal{M}_0^n(\mathbf{Q}_0 + \mathbf{p}_i^f)\right] - \mathcal{H}\left[\mathcal{M}_0^n(\mathbf{Q}_0)\right]\right], \quad i = 1, \ldots, N_e. \tag{4.10}$$

$\mathbf{Z}_n(\mathbf{Q}_0)$ is the linear approximation that is used in place of $\mathbf{H}$, where $\mathbf{Q}_0$ is the state's initial condition at the start of the smoother window. For the rest of the study, the notation of $\mathbf{Z}_n(\mathbf{Q}_0)$ will be dropped and $\mathbf{Z}_n$ will be used instead, but this term must be evaluated for different initial states of the DA window. This is prevalent in the minimization step where the continual updating of the initial state means a recalculation of the term. The formulation is the same as that in the MLEF, except that the summation term appears in the smoother and the terms within the summation have time indices associated with them. With this

notation, we have the gradient and Hessian of Eq. (4.5).

$$\mathbf{g} = \boldsymbol{\psi} - \sum_{n=1}^{N_{\text{mles}}} \mathbf{Z}_n^{\mathrm{T}} \mathbf{R}_n^{-1/2} \left[\mathbf{O}_n - \mathcal{H} \left[\mathcal{M}_0^n (\hat{\mathbf{Q}}_0^f + \mathbf{P}_f^{1/2} \boldsymbol{\psi}) \right] \right], \tag{4.11}$$

$$\mathbf{C} = \mathbf{I} + \sum_{n=1}^{N_{\text{mles}}} \mathbf{Z}_n^{\mathrm{T}} \mathbf{Z}_n . \tag{4.12}$$

Eq. (4.12) is another deviation from the MLEF. It contains error information for the state for the entire DA window, not just the single point in time as in the MLEF.

4.2 *Second change of variable*

A second change of variable is more specific to the MLES as it further addresses the Hessian preconditioning of the transformed cost function. Therefore, the second change of variable is introduced with the goal of optimally reducing the condition number of the Hessian, referred to as the optimal Hessian preconditioning. Following the MLEF, it is given as

$$\boldsymbol{\psi} = \mathbf{C}^{-\mathrm{T}/2} \boldsymbol{\xi}, \tag{4.13}$$

where the Hessian preconditioner, $\mathbf{C}$, results in the optimized control vector $\boldsymbol{\xi}$ and eventually the optimized global control vector $\hat{\mathbf{Q}}_0$. Using the optimal Hessian preconditioning term, the new cost function is defined:

$$\begin{aligned} J(\boldsymbol{\xi}) = & \frac{1}{2} \left[\mathbf{C}^{-\mathrm{T}/2} \boldsymbol{\xi} \right]^{\mathrm{T}} \mathbf{C}^{-\mathrm{T}/2} \boldsymbol{\xi} \\ & + \frac{1}{2} \sum_{n=1}^{N_{\text{mles}}} \left[\mathbf{O}_n - \mathcal{H} \left[\mathcal{M}_0^n (\hat{\mathbf{Q}}_0^f + \mathbf{P}_f^{1/2} \mathbf{C}^{-\mathrm{T}/2} \boldsymbol{\xi}) \right] \right]^{\mathrm{T}} \mathbf{R}_n^{-1} \left[\mathbf{O}_n - \mathcal{H} \left[\mathcal{M}_0^n (\hat{\mathbf{Q}}_0^f + \mathbf{P}_f^{1/2} \mathbf{C}^{-\mathrm{T}/2} \boldsymbol{\xi}) \right] \right] . \end{aligned} \tag{4.14}$$

Both the first derivative and second derivative with respect to $\boldsymbol{\xi}$ can now be calculated. The first derivative is found using the chain rule and yields

$$\nabla_{\boldsymbol{\xi}} J(\hat{\mathbf{Q}}_k) = \mathbf{C}^{-\mathrm{T}/2} \nabla_{\boldsymbol{\psi}} J(\hat{\mathbf{Q}}_k) = \mathbf{C}^{-\mathrm{T}/2} \mathbf{g} . \tag{4.15}$$

Expanding the above equation to purely represent the gradient in terms of $\boldsymbol{\xi}$ gives

$$\nabla_{\boldsymbol{\xi}} J(\hat{\mathbf{Q}}_k) = \mathbf{C}^{-\mathrm{T}} \boldsymbol{\xi}_k - \mathbf{C}^{-\mathrm{T}/2} \sum_{n=1}^{N_{\text{mles}}} \mathbf{Z}_n^{\mathrm{T}} \mathbf{R}^{-1/2} \left[\mathbf{O}_n - \mathcal{H} \left[\mathcal{M}_0^n (\hat{\mathbf{Q}}_k) \right] \right] . \tag{4.16}$$

The Hessian derivation starts with a Taylor series expansion of $J(\boldsymbol{\psi})$:

$$J(\boldsymbol{\psi} + \delta\boldsymbol{\psi}) = J(\boldsymbol{\psi}) + \mathbf{g}^{\mathrm{T}} \delta\boldsymbol{\psi} + \frac{1}{2} \delta\boldsymbol{\psi}^{\mathrm{T}} \mathbf{C} \delta\boldsymbol{\psi}, \tag{4.17}$$

where the higher order terms have been dropped and is a small positive number. After introducing the change of variable from Eq. (4.13) and applying the first variation,

$$\delta\boldsymbol{\psi} = \mathbf{C}^{-\mathrm{T}/2}\delta\boldsymbol{\xi}, \tag{4.18}$$

the second-order term in the Taylor expansion becomes

$$\frac{1}{2}\delta\boldsymbol{\psi}^{\mathrm{T}}\mathbf{C}\delta\boldsymbol{\psi} = \frac{1}{2}\left(\mathbf{C}^{-\mathrm{T}/2}\delta\boldsymbol{\psi}\right)^{\mathrm{T}}\mathbf{C}\left(\mathbf{C}^{-\mathrm{T}/2}\delta\boldsymbol{\psi}\right) = \frac{1}{2}\delta\boldsymbol{\xi}^{\mathrm{T}}\left(\mathbf{C}^{-1/2}\mathbf{C}\mathbf{C}^{-\mathrm{T}/2}\right)\delta\boldsymbol{\xi}. \tag{4.19}$$

Therefore, the Hessian reduces to

$$\nabla^2_{\boldsymbol{\xi}} J(\hat{\mathbf{Q}}_k) = \mathbf{C}^{-1/2}\mathbf{C}\mathbf{C}^{-\mathrm{T}/2} = \mathbf{C}^{-1/2}\left(\mathbf{C}^{1/2}\mathbf{C}^{-\mathrm{T}/2}\right)\mathbf{C}^{-\mathrm{T}/2} = \mathbf{I}\,. \tag{4.20}$$

4.3 *Minimization*

The MLES employs the forward model within the minimization process. The forward model must be applied to the control and the members in the ensemble for the length of the smoother window. While this is advantageous for considerations of nonlinearity, it also induces computational expense in the analysis procedure. Nevertheless, with the optimal Hessian preconditioning, Eq. 4.20 shows that when the observation operator is linear, the Hessian is the identity matrix and thus only a single minimization iteration is needed. Indeed, even with nonlinear observation operators, a sufficient number of minimization iterations is typically only two to three (Zupanski (2005)).

In this study, the minimization method used is the Newton method. Of course, more complex minimization techniques can be used (Alekseev *et al.* (2009)) and are more efficient for larger nonlinear unconstrained optimization problems. However, using the optimum control variable and the straightforward models in this study, the Newton method proves to be an efficient method of minimizing the optimized cost function (Zou *et al.* (1993)). The problem of minimizing a function of n variables,

$$\min f(\mathbf{x}), \quad \mathbf{x} \in \mathrm{IR}^{\mathrm{n}\times 1}\,, \tag{4.21}$$

where $f(\mathbf{x})$ is a smooth function can be solved iteratively using

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k\,. \tag{4.22}$$

k indicates the number of iterations, and α_k is defined as the step-length value that can vary in each iteration and is determined by the minimization method. If using the Newton's method, $\alpha_k = 1$ and the descent direction $\mathbf{d}$ can be solved using

$$\mathbf{d}_k = -\left(\nabla^2 f(\mathbf{x})\right)^{-1}\nabla f(\mathbf{x})\,. \tag{4.23}$$

Using these steps and following Eq. (4.22) and Eq. (4.23), our minimization update equations are

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \mathbf{d}_k\,, \tag{4.24}$$

and the descent direction is

$$\mathbf{d}_k = -\left[\nabla^2_{\boldsymbol{\xi}} J(\hat{\mathbf{Q}}_k)\right]^{-1} \nabla_{\boldsymbol{\xi}} J(\hat{\mathbf{Q}}_k)\,. \tag{4.25}$$

The process starts with an initial guess of $\boldsymbol{\xi}_{k=0} = 0$, which implies

$$\hat{\mathbf{Q}}_{k=0} = \hat{\mathbf{Q}}^f\,. \tag{4.26}$$

4.4 *State update*

After the optimal value of $\boldsymbol{\xi}_{k+1}$ is obtained, the analysis deterministic state vector $\hat{\mathbf{Q}}^a$ is updated by

$$\hat{\mathbf{Q}}^a = \hat{\mathbf{Q}}^f + \mathbf{P}_f^{1/2}\left[\mathbf{I} + \sum_{n=1}^{N_{\text{mles}}} \mathbf{Z}_n^{\mathrm{T}}(\hat{\mathbf{Q}}_0^f)\mathbf{Z}_n(\hat{\mathbf{Q}}_0^f)\right]^{-\mathrm{T}/2} \boldsymbol{\xi}_k\,. \tag{4.27}$$

The local Hessian preconditioning term $\mathbf{C}_k^{\mathrm{T}/2}$ is estimated at the optimal analysis point and is then used to form the analysis error covariance

$$\mathbf{P}_a^{1/2} = \mathbf{P}_f^{1/2}\left[\mathbf{I} + \sum_{n=1}^{N_{\text{mles}}} \mathbf{Z}_n^{\mathrm{T}}(\hat{\mathbf{Q}}_0^a)\mathbf{Z}_n(\hat{\mathbf{Q}}_0^a)\right]^{-\mathrm{T}/2} = \mathbf{P}_f^{1/2}\mathbf{C}_k^{-\mathrm{T}/2}\,. \tag{4.28}$$

Finally, the ensemble members are reinitialized for the forecast step in the next DA cycle by

$$\hat{\mathbf{Q}}_0^i = \hat{\mathbf{Q}}_0 + \mathbf{p}_i^a\,, \quad i = 1, \ldots, N_e\,, \tag{4.29}$$

where $\mathbf{p}_i^a$ is the columns of $\mathbf{P}_a^{1/2}$.

4.5 *MLES pseudocode*

The pseudocode for the MLES is provided in Algorithm 1, showing three major steps of DA: initialization, forecast and analysis. Additionally, the solution process is illustrated by the flow chart in Fig. 3.

Algorithm 1 MLES method detailing the three major steps

Step 1: Initialization

procedure LAGGED FORECAST($\mathbf{Q}_0^{1:N_e}$,$\hat{\mathbf{Q}}_0$)
  Select the initial conditions of uncertainties.
  Propagate $\mathcal{M}(\mathbf{Q}^{1:N_e})$ and $\mathcal{M}(\hat{\mathbf{Q}})$ for $t \in [0, t_\tau]$.
  Construct $\mathbf{Q}_0^{1:N_e}$ from random picks in $t \in [0, t_\tau]$.
  Select $\hat{\mathbf{Q}}_0$from $t = t_\tau/2$.
end procedure

Step 2: Forecast

procedure FORECAST STEP($\mathbf{Q}^f$,$\mathbf{P}^f$)
  Propagate $\mathcal{M}(\mathbf{Q}^{1:N_e})$ and $\mathcal{M}(\hat{\mathbf{Q}})$ to next analysis time.
  Obtain $\mathbf{Q}^f$ and $\hat{\mathbf{Q}}^f$.
  Calculate $\mathbf{P}_f^{1/2}$ by Eq. (4.4).
end procedure

Step 3: Analysis

procedure ANALYSIS STEP($\hat{\mathbf{Q}}^a$,$\mathbf{P}^a$)
  Set $\hat{\mathbf{Q}}_{k=0} = \hat{\mathbf{Q}}^f$ for the first minimization.
  Apply change of variable $\boldsymbol{\xi}_{k=0}$.
  Evaluate the Hessian preconditioner by Eq. (4.13).
  while $1 \leqslant k \leqslant n$ **do** ▷ Start minimization.
    Calculate $\mathbf{d}_k$ by Eq. (4.25).
    Update $\boldsymbol{\xi}_k$ by Eq. (4.24).
    Update $\hat{\mathbf{Q}}_k$ by Eq. (4.27).
  end while ▷ End minimization
    Update control analysis, $\hat{\mathbf{Q}}^a = \hat{\mathbf{Q}}_k$.
    Calculate analysis covariance, $\mathbf{P}_a^{1/2}$ by Eq. (4.28).
    Update $\mathbf{Q}_i^a$ for $i = 1, \ldots, N_e$ by Eq. (4.29).
  end procedure

Repeat Steps 2-3.

5. Validation and performance assessment

The framework is developed and validated by the well-known Lorenz 96 model (Lorenz (1995)). This model is chosen because it is used in many validation cases for DA and has literature data for the performance assessment of the framework (e.g. Anderson (2001), Anderson (2007), Hoteit *et al.* (2012), Ott *et al.* (2004), Whitaker & Hamill (2002)). Additionally, by choosing a correct forcing term, the model is commonly known to cause chaotic behaviour. The result of the MLES algorithm is compared with those of the MLEF algorithm. The performance of the MLES algorithm is measured by χ^2 diagnostics, the normalized cost function and a root-mean-square error (RMSE) analysis. The following sections provide details of the numerical experimental set-up, descriptions of the tests and the results of the numerical experiments.

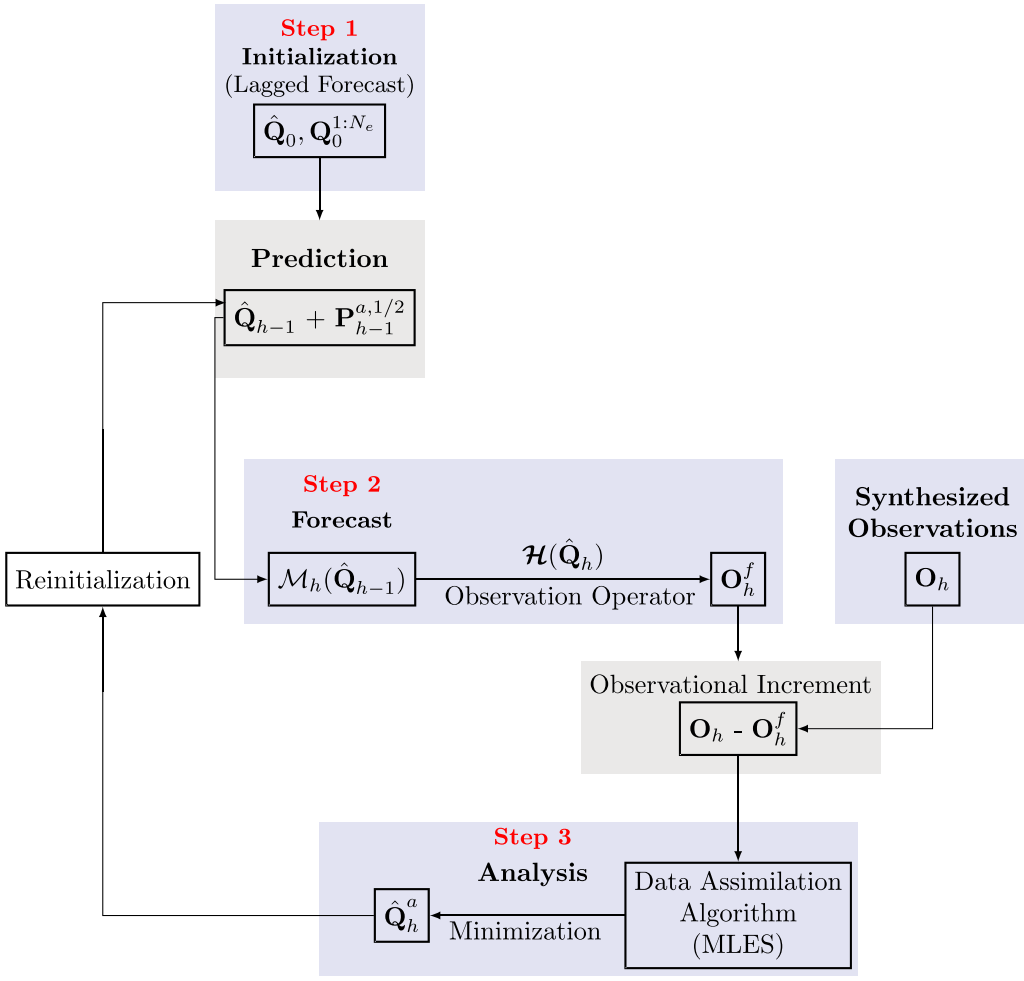

Fig. 3. A flowchart showing the steps for the MLES algorithm. Note that these steps correspond exactly to the steps as shown in Algorithm 1.

5.1 *Description of the test model*

The Lorenz 96 model is given as

$$\frac{dx_i}{dt} = x_{i+1} - x_{i-2}\ x_{i-1} - x_i + F\,, \quad i = 1, \ldots, M\,. \tag{5.1}$$

A value of $F = 8$ is known to cause chaotic behaviour and chosen as such for the validation. In this study, the number of variables in the test is chosen to be $M = 40$. The chosen forward model to advance the solution in time, $\mathcal{M}(\cdot)$, is a second-order finite volume method with the standard four-stage Runge–Kutta explicit time marching method.

Table 1 *The Lorenz 96 model experiment's data assimilation parameters.*

Variable	Value
N_e	15 members
Observation time interval	0.05 time units
N_O	40 (every computation cell)
N_{mles}	10
σ_{obs}	0.05
Diagonal elements of $\mathbf{R}$	0.05^2
Minimization steps	3

5.2 *Data assimilation configuration*

There are a number of parameters specific to DA and are explained herein. N_e is the number of members in the ensemble, excluding the control. The choice of N_e greatly determines the ability of the ensemble-based DA algorithms to accurately capture the error in the current system. The time between observations is conveniently chosen to be the same as the forward model time step size. This also dictates the length of a DA cycle. The MLEF is performed at every observation point in time and the MLES DA window length is determined by the observation interval as well as N_{mles}, or the number of observations taken in the MLES window. N_O is the number of observations. σ_{obs} is the variance of the normal distribution used for synthesizing observations taken from the truth as shown in Eq. (3.7). The diagonal elements of the observation covariance, $\mathbf{R}$, are unchanging with values σ^2_{obs}. The number of steps to be used in the minimization procedure is based on Zupanski (2005). The DA configuration parameters for the validation are given in Table 2.

These values are used to gather statistics for the tests in the following sections. Note that the observation error covariance matrix $\mathbf{R}$ only has diagonal elements specified in the table above and stays constant through the entire simulation, i.e. the random variance for the gathered observation data remains the same for each DA cycle. Note that for the MLEF, one DA cycle consists of running the model to each observation available. For the MLES, however, one DA cycle for this numerical experiment is 10 times the length of the DA cycle in the MLEF because the number of observations taken in the time window is $N_{\text{mles}} = 10$.

5.3 *DA performance measures*

The χ^2 diagnostics test (Menard *et al.* (2000), Zupanski (2005), Wang *et al.* (2020)) evaluates the correctness of the innovation covariance matrix. It is given, in observation space, as

$$\chi^2 = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{N_O}, \quad \mathbf{X} = \left[\mathbf{R} + \mathbf{H}(\hat{\mathbf{Q}})\mathbf{P}_f\mathbf{H}^{\mathrm{T}}(\hat{\mathbf{Q}})\right]^{-1/2}\left[\mathbf{O} - \mathscr{H}(\hat{\mathbf{Q}})\right], \tag{5.2}$$

where $\mathbf{H}(\hat{\mathbf{Q}})$ is the tangent linear model of the observation operator, $\mathscr{H}(\cdot)$. $\hat{\mathbf{Q}}$ is the analysis state at each DA cycle. For a Gaussian distribution of innovations and a linear observation operator, the expected values should be equal to 1.0. However, due to the calculation of $\mathbf{P}_f$ wherein the non-linear model is used, and also due to the finite number of observations per cycle, the actual values for the assimilated runs will not be exactly 1.0. The result of the test is shown in Fig. 4. Every DA cycle in the MLES

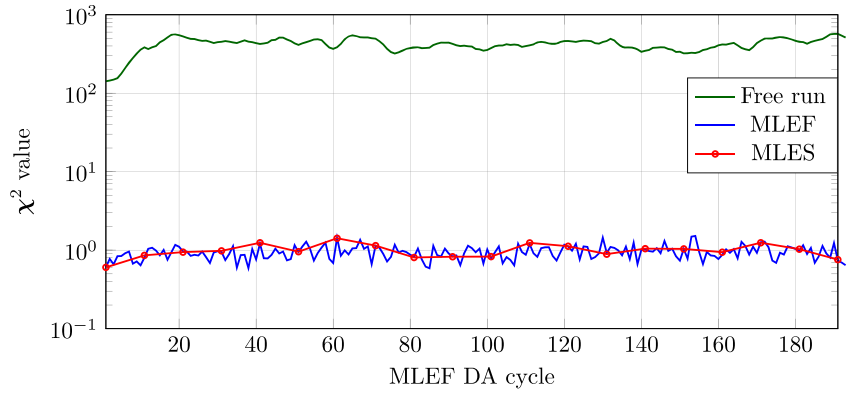

Fig. 4. The χ^2 diagnostic test for the Lorenz 96 model experiment. The y-axis is in log-scale. The x-axis is normalized DA cycles by the MLEF one; correspondingly, the MLES updates are shown on the coarser temporal locations as indicated by the red circles where the MLES performs the analysis.

is performed at a time interval that is equivalent to 10 DA cycles in the MLEF. For convenience, the x-axis is normalized by the MLEF DA cycles. Accordingly, the MLES is shown on a coarser temporal scale as indicated by the red circles. Nevertheless, both numerical experiments show the same solution time at the same corresponding coordinate locations. Both MLEF and MLES show χ^2 values close to 1.0 for each DA cycle. The mean χ^2 values for the entire run are 0.9509 for the MLEF and 1.088 for the MLES, respectively. Either by running further numerical experiments or by increasing the number of Gaussian distributed observations, one would expect each subsequent run to have a mean close to 1.0. The free run χ^2 values are expectedly much larger and emphasizes the chaotic behaviour of the numerical experiment's model.

We examine the normalized cost function where the number of observations taken in the spatial domain is the normalization factor and is calculated by

$$\frac{J}{N_O} = \frac{1}{N_O}\,\frac{1}{2}\,\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}})^{\mathrm{T}}\,\mathbf{R}_j^{-1}\,\mathbf{O}-\mathcal{H}(\hat{\mathbf{Q}}) \quad . \tag{5.3}$$

The cost function J and the control vector $\hat{\mathbf{Q}}$ are calculated after the minimization process has occurred and therefore the analysis state is used for Eq. (5.3). The normalized cost function is used to assess the minimization in a multi-variate problem and is related to the χ^2 test. The uncertainty error is reduced and thus the expected values are expected to be approximately 0.5. The results can be seen in Fig. 5 and every value is close to 0.5.

The RMSE is also used to measure the difference between the analysis and the truth. The truth is simply the deterministic simulation by the forward model using perfect conditions and parameters. The RMSE is calculated by

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{m=1}^{N_S}\left(\mathbf{Q}_{\mathrm{true}}-\hat{\mathbf{Q}}^a\right)_m^2}{N_S}}\,. \tag{5.4}$$

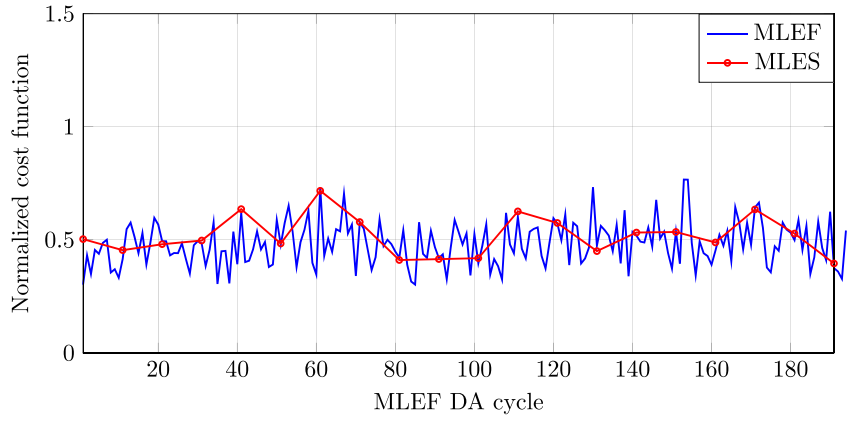

Fig. 5. The normalized cost function values for the MLEF and MLES. The *x*-axis is normalized DA cycles by the MLEF one; correspondingly, the MLES updates are shown on the coarser temporal locations as indicated by the red circles where the MLES performs the analysis.

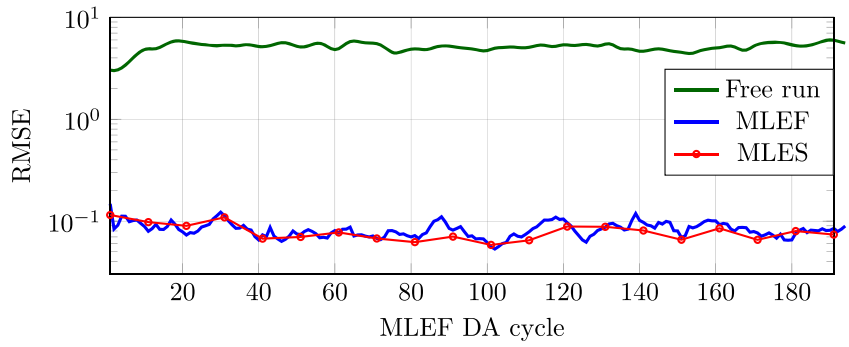

Fig. 6. The RMSE for the free run, MLEF, and MLES. The *y*-axis is in log-scale. The *x*-axis is normalized DA cycles by the MLEF one; correspondingly, the MLES updates are shown on the coarser temporal locations as indicated by the red circles where the MLES performs the analysis.

The results of the RMSE are seen in Fig. 6 for each DA cycle and compared between the MLEF and MLES. In addition, the RMSE of the free run is shown and clearly larger than that of the MLEF and MLES, which is expected. Both the MLEF and MLES have similar values and, overall, the RMSE shows a stable convergence with the truth as well as having stable performance throughout the numerical experiment.

Additionally, the first state value, Q_1, is plotted as a function of the time step for the MLES and MLEF and the synthesized data for this particular state value are also shown in Fig. 7. This particular case set-up is the same as described in Section 5.2. The first DA update is located at the 11 th time step and is the corresponding analysis update for the MLES of the time window. After the MLES analysis update, it is freely forecasted to an end time of 120 time steps. The first analysis point for the MLEF is also at the 11th time step but is updated at every available observation until the last observation which is

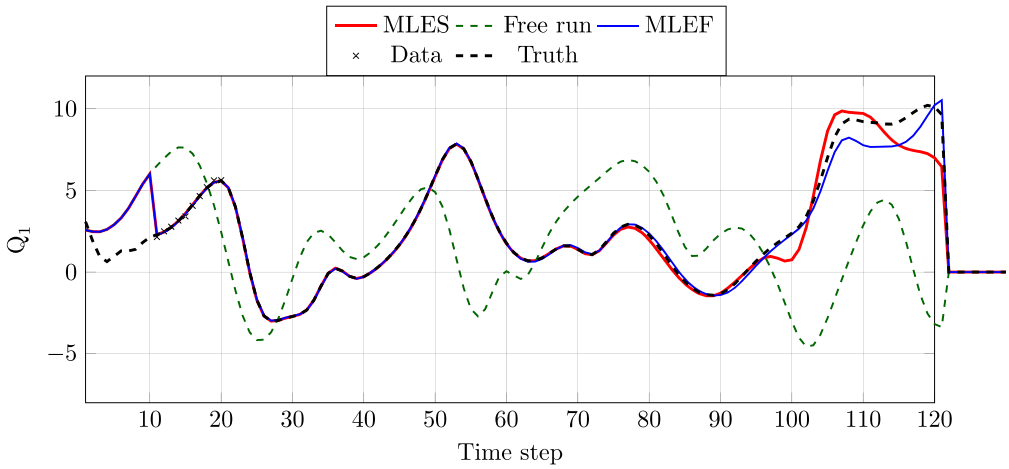

Fig. 7. The trajectory of Q_1 for the truth, MLES, MLEF, free run and observations of the Lorenz 96 model over time. Both the MLEF and MLES are first updated at the 10^{th} time step and the MLEF is continually updated until the last observation available. The free run is simply a simulation by the forward model with the same steps as the MLES or MLEF but without DA.

located at the 20^{th} time step. At this point, it is freely forecasted to the same end time as the MLES. A free run that does not have any DA updates is also included.

Fundamentally, Fig. 7 displays the increase in predictability after DA is applied. Both the MLEF and MLES diverge from the truth at approximately the 100^{th} time step, while the free run completely diverges from the truth very early. The DA simulations have good approximations of the truth state but their trajectories start to deviate from the truth as a consequence of the nature of the chaotic system. It is worth mentioning that the MLEF DA method is competent and efficient in general; it is encouraging and promising to have shown the MLES to have comparable performance as the MLEF for these cases. By design, it is expected that the MLES demonstrates superior performance over the MLEF for practical applications as mentioned earlier, but this will be part of a future follow-up study.

6. Results and discussions

The validated MLES algorithm is now applied to solve the 1D KS equation. Originally developed for modelling instabilities in flame fronts (Sivashinsky (1977), Kuramoto (1978)), this simulation provides insight to the MLES application to turbulent CFD problems. The algorithm is tested and compared against the original MLEF algorithm.

The model equation is

$$\frac{\partial y}{\partial t} = -a\frac{\partial^2 y}{\partial x^2} - by\frac{\partial y}{\partial x} - c\frac{\partial^4 y}{\partial x^4}, \tag{6.1}$$

where, in the truth run, the model parameter coefficients are $a = 1$, $b = 1$ and $c = 1$. The boundary conditions are chosen to be periodic and the computational mesh has 256 cells uniformly distributed over $x \in [-100, 100]$. The initialization of the ensemble is based on the lagged forecast method (Kalnay (2003)). That is, given a time interval length of $t \in [0, t_\tau]$, a 'burn-in' simulation of length t_τ is run

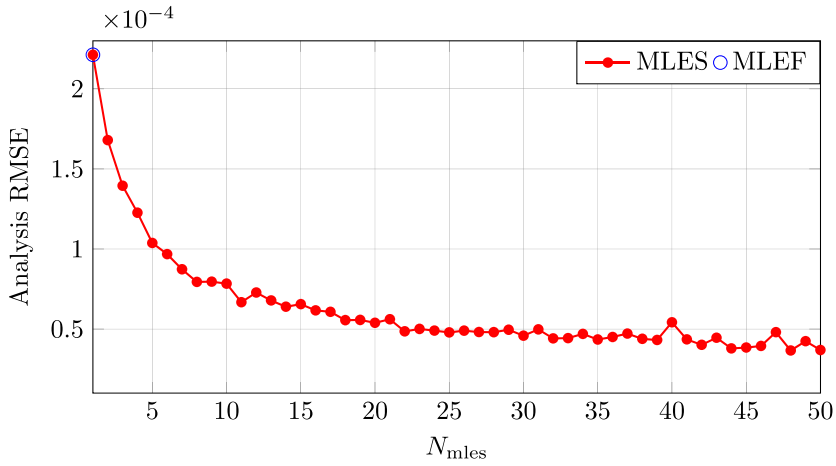

Fig. 8. The analysis RMSE of the MLES by varying N_{mles} from 1 to 50. The red dotted points are the statistical average of 100 runs of the RMSE of the analysis of the MLES as a function of N_{mles}. The blue circle is the MLEF at the analysis point and, as expected, matches the MLES when only one observation is taken in the window.

using an arbitrarily chosen initial condition. Given an ensemble of size N_e, one picks N_e randomly distributed temporal points over the $[0, t_\tau]$ interval to ensure that the uncertainties of the dynamical system are properly represented. At each of the chosen temporal points, there is a corresponding state vector, $\mathbf{Q}$, from the 'burn-in' simulation which is then used as the initial conditions for the members in the ensemble. The true state is set to be the simulation in the middle of the simulation length and the initial state of the control vector is set to be the state average of the members in the ensemble. t_τ is set to be 5 time units. Where applicable, initial uncertainty of the model parameters is based on a normally distributed PDF. The variance for the synthesized observations is chosen to be $\sigma_{\text{obs}} = 0.001$. The time step of the forward model is $\Delta t = 0.005$ time units.

6.1 *Number of time points with observations*

We first perform a numerical experiment to study the effect of the MLES as a function of N_{mles} (the number of observations used within a smoothing window). That is, using a fixed time interval between observations, N_{mles} is varied and the RMSE of the analysis update of the initial DA cycle (i.e. the beginning of the smoothing window) is then calculated for each value of N_{mles}. Note that by fixing the time interval between observations and adjusting N_{mles}, the total length of the entire smoothing window changes. Figure 8 shows the statistical average of the RMSE calculated over 100 individual runs. N_{mles} varies from 1 to 50. The fixed time interval between observations is $\Delta t = 0.05$ time units and the domain is fully observed. Also, $N_e = 20$ for each run in this numerical experiment.

As a check, the MLEF is included and shows the analysis RMSE matches the MLES when $N_{\text{mles}} = 1$ which is to be expected. Then, as N_{mles} is increased, the RMSE values of the analysis drop until there is a saturation of performance. We expect the overall performance to eventually degrade as N_{mles} is increased beyond a certain threshold as the distance between the last observation and the location of the analysis becomes larger. In other words, the introduction of nonlinearity (mainly from the forward model) of a larger smoother window will eventually cause an increase in the error. Nevertheless, Fig. 8 shows that given more information within a smoothing window, the error is reduced. It is worth noting that while

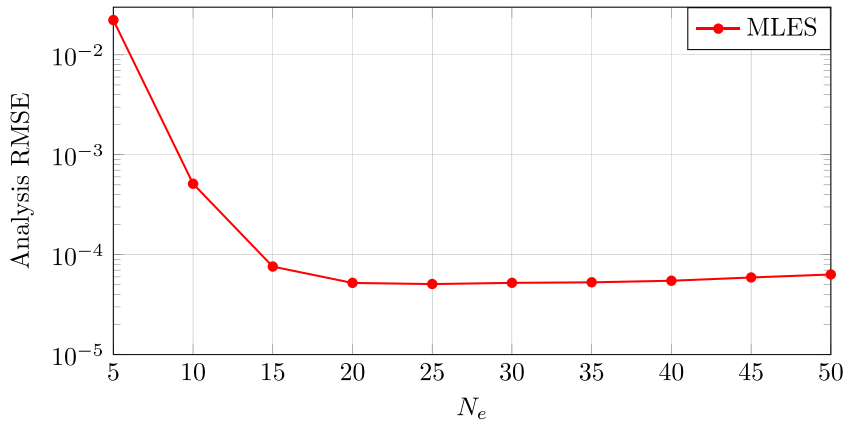

Fig. 9. MLES performance as a function of N_e using a constant smoother window length of $N_{\text{mles}} = 25$.

there are small oscillations in the RMSE values (see points $N_{\text{mles}} = 11$, 40 and 46, for example), it is expected that this curve will smooth out as more and more cases are run to help with the averaging. The important result is that the general trend of error reduction in the MLES goes down as more information is provided within a smoothing window. Furthermore, there is a balance between the optimal state and the computational cost for assimilating observations. For example, $N_{\text{mles}} = 18$ provides the desired optimal state.

6.2 *Ensemble size*

The ensemble size is varied from $N_e = 5$ to $N_e = 50$ and a similar test is run as described in Section 6.1. The smoother window is kept constant using $N_{\text{mles}} = 25$. The RMSE of the analysis is reported in Fig. 9. The performance of the MLES is severely diminished when a small ensemble size is used. The performance is found to be relatively stable after approximately N_e is chosen to be 15 or greater. However, more members than 20 are not necessary for this case according to the sensitivity study. This finding is indeed consistent with what is to be expected of other ensemble DA methods (Asch *et al.* (2016))

6.3 *Number of observations*

The MLES is now used to directly compare results with the MLEF. The current test cases are performed by varying the number of available observations in the domain (N_O) while keeping N_{mles} constant for each case. In doing so, the predictability limits of both methods are directly compared as a function of the number of observations in the domain. Spatially, these observations are evenly spaced. For the MLES, only one smoother window (and therefore only one analysis update) is used. Once the optimal analysis state has been found, the control of the MLES is forecasted to a temporal location equal to twice the MLES smoother window length and the RMSE is calculated. In order to directly compare with the MLES, the control of the MLEF is updated using N_{mles} DA cycles. At the last point of the MLEF update, the MLEF control state is forecasted to the same ending point of the MLES and the RMSE is calculated. This means that the time between the updated analysis location for the MLES and the end of the simulation is equivalent to two smoother window lengths, whereas the last MLEF analysis

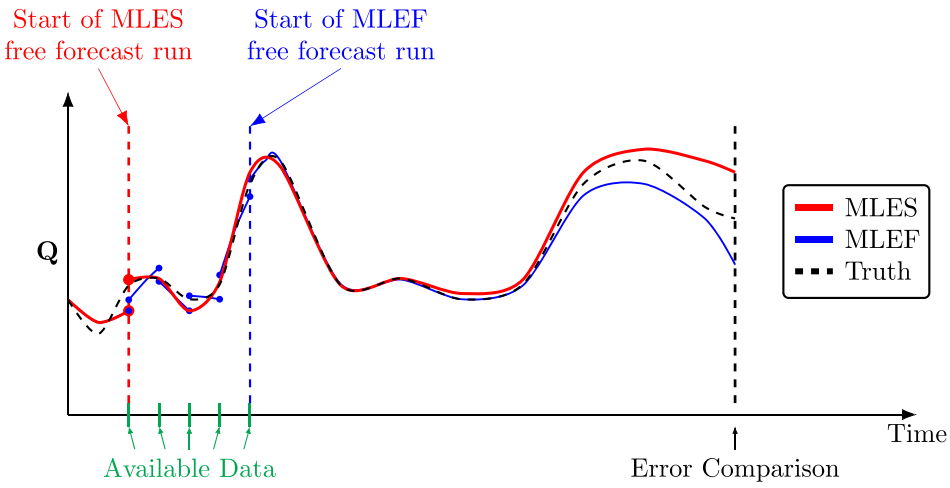

Fig. 10. Graphical description of the error comparison with the MLES and MLEF. The MLES is updated at the vertical dashed red line and freely run to the vertical black dashed line. The MLEF is updated at each of the green hashes which represent available observations. The last update for the MLEF is located at the vertical blue dashed line and run to the same vertical black dashed line. At the vertical black dashed line, the error of the MLES and MLEF is calculated with respect to the truth and compared. This value is saved and the mean of these values is taken over 100 simulations. In the figure, the *y*-axis, **Q**, represents just one state value in the entire spatial domain.

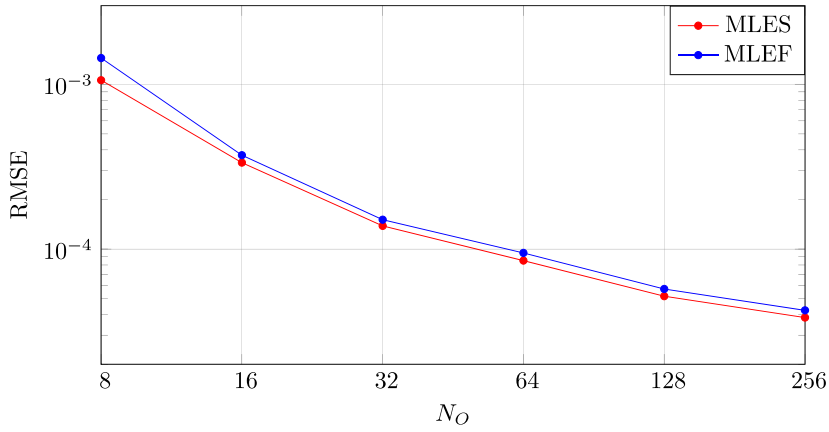

Fig. 11. Performance comparison between the MLES and MLEF using a smaller amount of evenly spaced observations in the domain. The RMSE is taken at the end of a time window length equal to two smoother window lengths after the first initial MLES analysis update.

update is exactly one smoother window length. The process is graphically shown in Fig. 10. The size of the ensemble is set to $N_e = 20$ members and the smoother window is set to be $N_{\text{mles}} = 25$. The other simulation parameters are the same as presented in Section 6.1. The result of the averaged RMSE over 100 independent runs for each N_O is shown in Fig. 11.

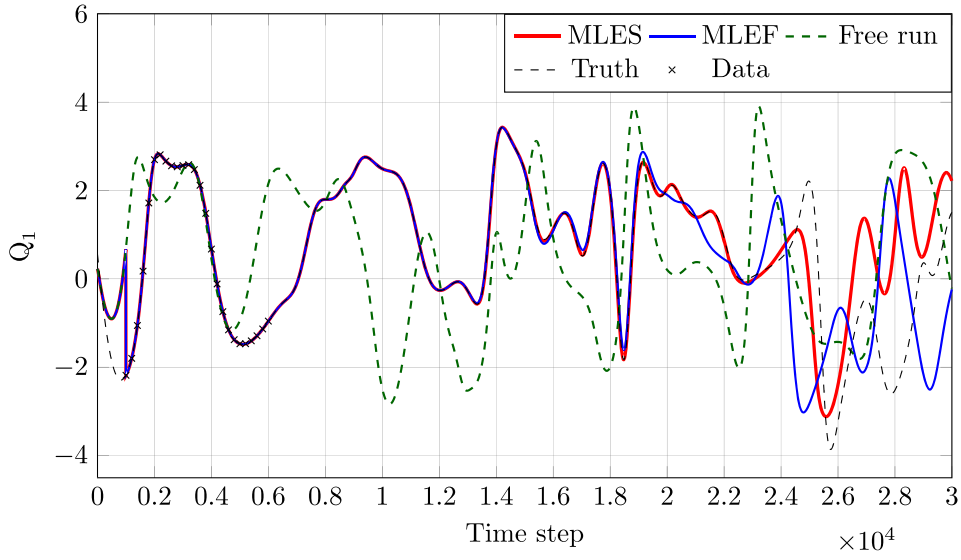

Fig. 12. The model state and observation data at the first spatial location.

The limit of predictability for both algorithms decreases with more available observations which is to be expected. The error reduction for the MLES is slightly better for all cases. This is promising and interesting as both DA algorithms have been provided with the same information, while the main difference is that the MLES has a longer time between the location of the RMSE calculation and the initial analysis. This will be an area of emphasis for future studies in order to better understand the impact of the observations on the performance.

6.4 *State quantities and RMSE*

Three state quantities are presented to compare the improvement of predictability for both the MLES and MLEF. Using $N_O = 64$, the state variable at spatial domain locations of 1, 65 and 221 for the MLES, MLEF, free run and truth run are shown in Figs 12–14. Also shown are the observed quantities at these locations as well. The observation interval is chosen to be 200 time steps in this case. Each DA algorithm correctly matches the truth state beginning at the first analysis cycle located at time step 0.1×10^4. Deviation for both runs can be clearly distinguished at time step 2×10^4. For each spatial location presented, the MLES does a slightly better job at predicting the truth for a slightly longer time than the MLEF. It is worth noting that at time step 1.8×10^4, the MLES does slightly diverge from the truth but is still a good approximation of the truth.

The RMSE value is also given for the current simulation and is shown in Fig. 15. The MLEF updates can clearly be seen from the vertical drops after the initial analysis state update. The plot shows good initial convergence to the truth and accurately depicts the chaotic nature of the KS equation as both DA algorithms eventually diverge from the truth. Note that this is a single run and previous results have been collected from an average of these runs.

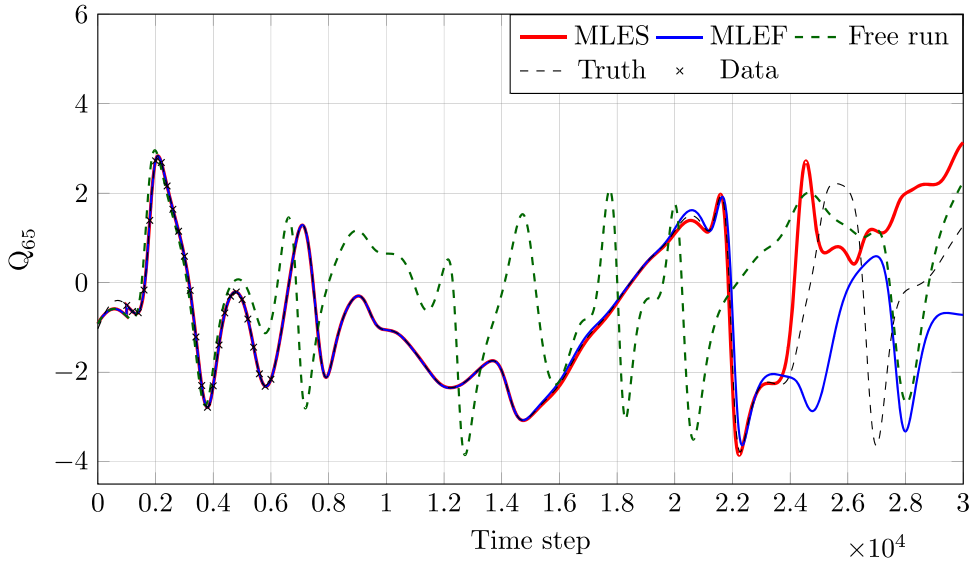

Fig. 13. The model state and observation data at the 65 th spatial location.

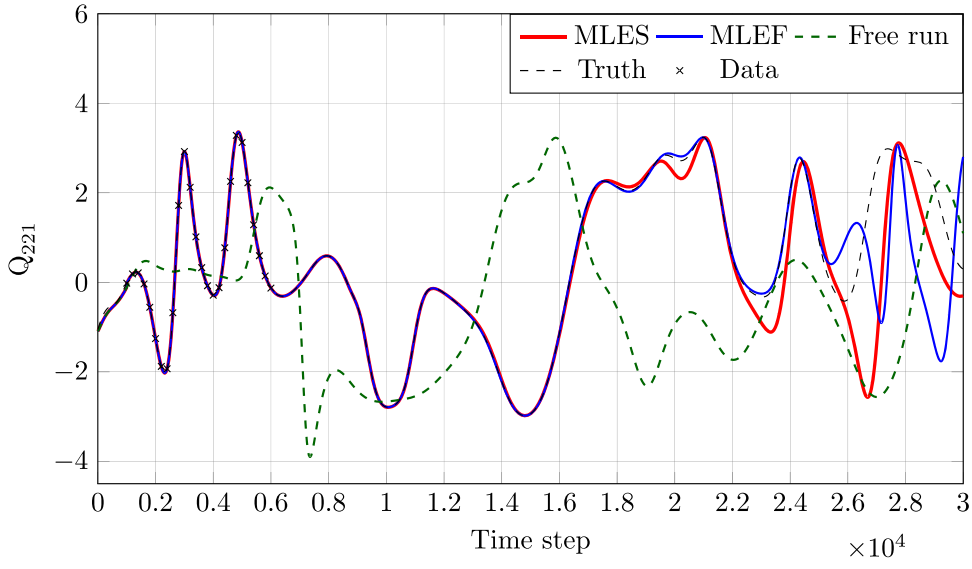

Fig. 14. The model state and observation data at the 221 st spatial location.

6.5 *Model parameter estimation*

The estimation of model parameters of the KS equation, a, b and c, are now included in the DA process and therefore added to the control vector $\mathbf{Q}$. The initial uncertainty of the parameters for each member in the ensemble is chosen by a random Gaussian PDF with mean 0.50 and variance of 0.05. For reference,

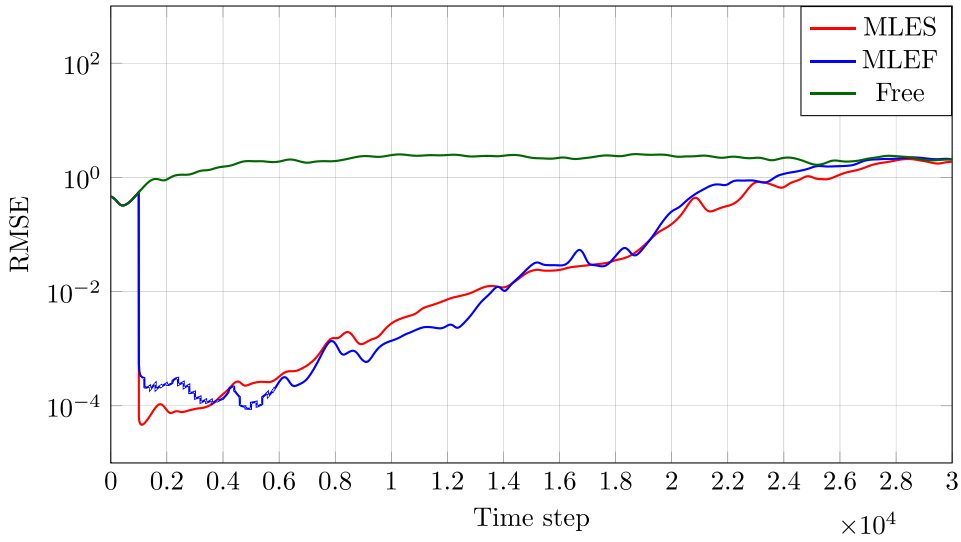

Fig. 15. The RMSE values for the MLES, MLEF and free run.

Table 2 *The estimated model parameters for the MLES and MLEF by taking the average of the last analysis model parameters over* 100 *different cases. The mean of the initialization of the model parameters is set to be* 50% *less than that of the truth.*

Model Param.	**Truth**	**MLES**	**MLEF**
a	1.00	1.0003	1.0010
b	1.00	1.0013	1.0009
c	1.00	1.0001	1.0012

the truth of these parameters is known as $a = b = c = 1$. A total of 10 DA cycles for the MLES is used and the corresponding amount of DA cycles is used for MLEF. The initial 'burn-in' period is reduced to 2 time units. The rest of the case setup is similar to that in Section 6.1. Note that the model parameters are not observed as this is often the case. The statistical average of the mean of the final analysis update of the model parameters for the MLES and MLEF after 100 independent cases is shown in Table 3. All three model parameters for both the MLES and MLEF to converge to approximately the true values.

Additionally, the plots of the model parameter update over time is also shown in Figs 16–18 for a single run. Although the simulations over the entire solution time are not shown for the sake of space, both MLES and MLEF converge to the true model parameters quickly. Both DA algorithms show good prediction of the true model parameters even with a poor initial uncertainty of the truth. The MLES does better for a and c, while slightly less for b when compared with the MLEF. More complex chaotic systems will be investigated to see whether the MLES consistently does better than the MLEF in general.

6.6 *Averaged observations*

In fluid dynamics problems that involve chaotic turbulence, rather than providing instantaneous data, the time-averaged flow data are desirable. Thus, it is of interest to use time-averaged data. To synthesize

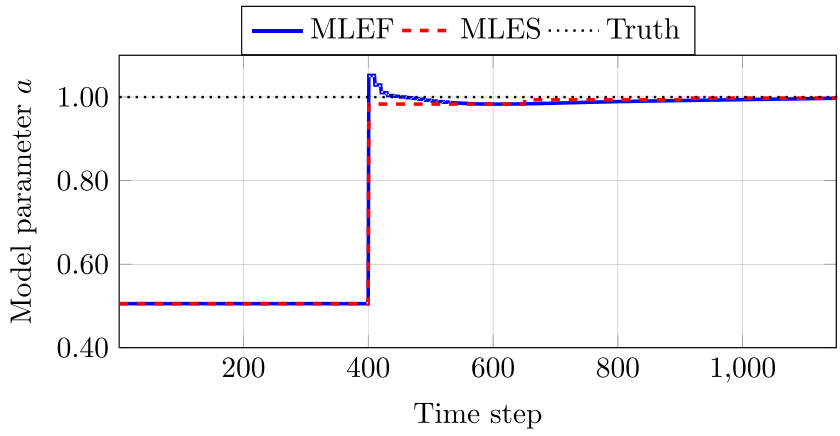

Fig. 16. Model parameter a convergence.

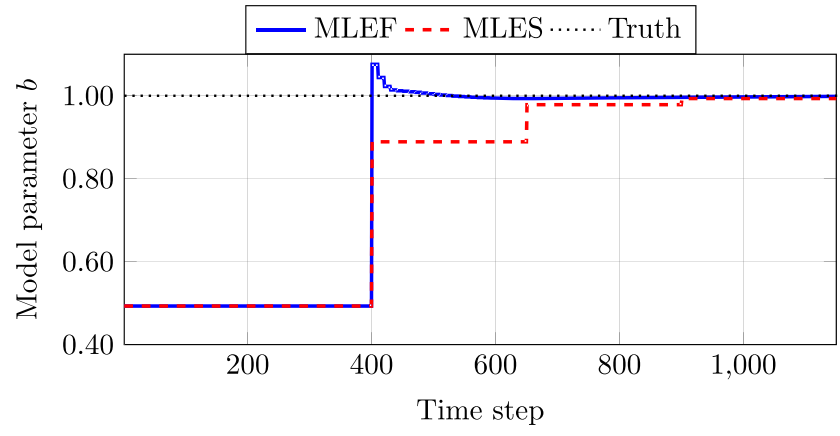

Fig. 17. Model parameter b convergence.

the data, a time-averaged true value is used in the observation operator, but a random perturbation η is applied to the averaged value as 'measurement error'. The acquisition of these data is shown in Fig. 19. Apart from the averaged observations, the case set-up is identical to that in Section 6.5 where the statistical average of the model parameters are found after running 100 cases. Table 4 presents the model parameter estimates by the MLES and MLEF. All three model parameters for both the MLES and MLEF converge to the true values and the MLES performs slightly better for a and c than the MLEF.

The plots of the model parameter update over time are also shown in Figs 20–22 for a single run. Note that only three MLES DA cycles (and the equivalent number of MLEF DA cycles) of the simulation are shown as both MLES and MLEF converge to the true model parameters quickly. Both DA algorithms show good prediction of the true model parameters even with a poor initial uncertainty of the truth. Notice that the MLES performs slightly better in this study. Applications to more complex problems will be investigated in a follow-up study.

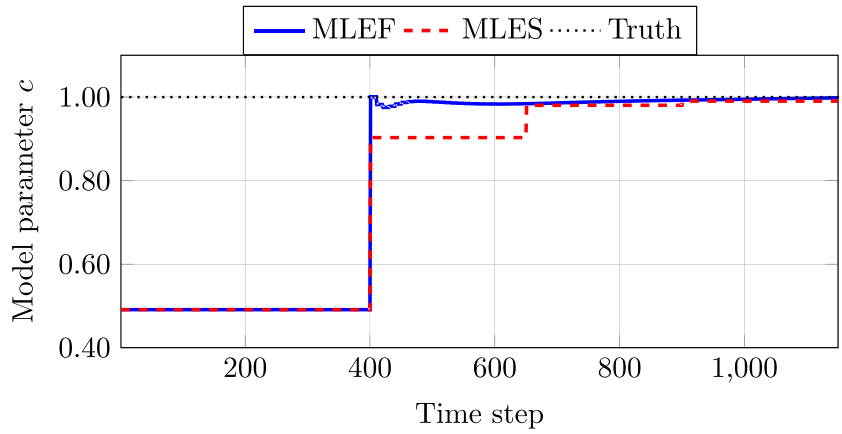

Fig. 18. Model parameter c convergence.

Table 3 *The estimated model parameters for the MLES and MLEF using the averaged observations method.*

Model Param.	Truth	MLES	MLEF
a	1.00	0.9997	1.0004
b	1.00	1.0011	1.0006
c	1.00	0.9996	1.0006

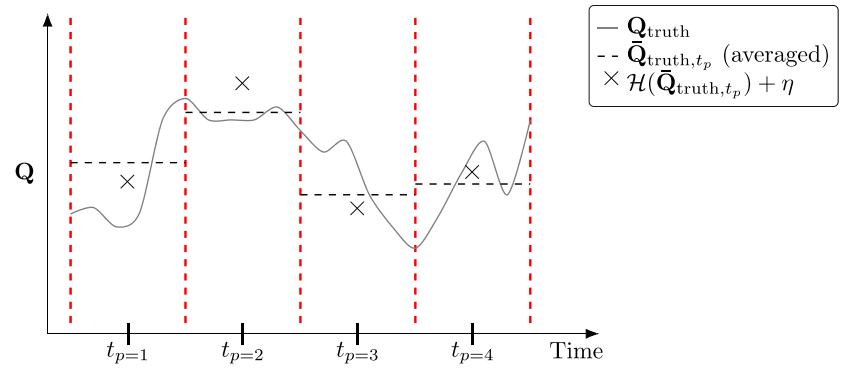

Fig. 19. To gather the averaged data, an average of the true values is taken within an interval and then the observation operator is applied and perturbed using random variance. The interval is shown as the vertical red dashed lines and the black horizontal dashed lines are the averaged true values. The black crosses represent the data obtained after applying the observation operator $\mathcal{H}(\cdot)$ and adding random variance, η. The y-axis, $\mathbf{Q}$, represents just one state value in the entire spatial domain.

7. Conclusions and future work

The study introduces a new MLES algorithm and compares it alongside its sister sequential filtering algorithm, the MLEF, using the 1D KS equation. Both algorithms show positive impact on the analysis

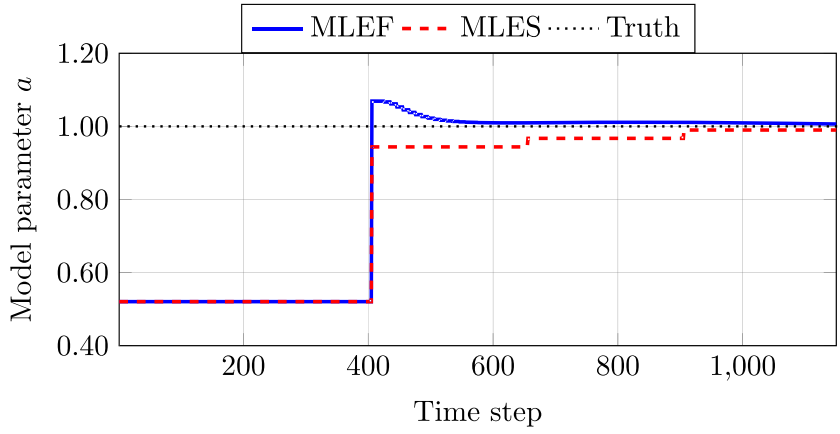

Fig. 20. Model parameter a convergence for the time-averaged synthesized observations.

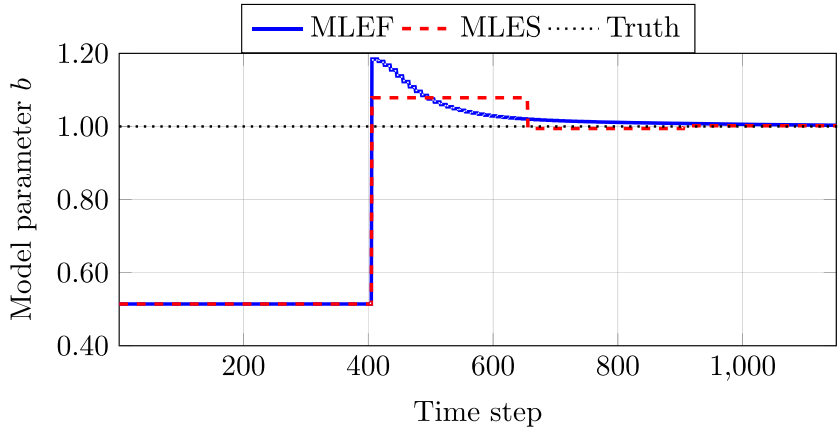

Fig. 21. Model parameter b convergence for the time-averaged synthesized observations.

states for each of the numerical experiments presented, while the MLES is at least comparable or slightly better than the MLEF. This inceptive study shows that it is a promising extension of the MLEF.

Aside from the numerical experimental results shown in Section 6, many similar numerical experiments have been performed and it has been observed that the initial starting point of the model parameters for the members in the ensemble and the control have the most impact on the quality of results. Quality, in this case, means the closeness to the true values. It is also seen that the large changes in the observation variance do not impact the results as much as the model parameters do. This leads to the idea that the gathered data do not have as much of an impact, but rather the empirical knowledge of the dynamical system is more important in performing such numerical experiments.

To reiterate, the model parameter estimation experiment is under the assumption that the initialization of the parameters is not exactly the truth. This is an interesting concept because in reality there would be no notion of a 'true' model parameter because model parameters cannot be observed and the

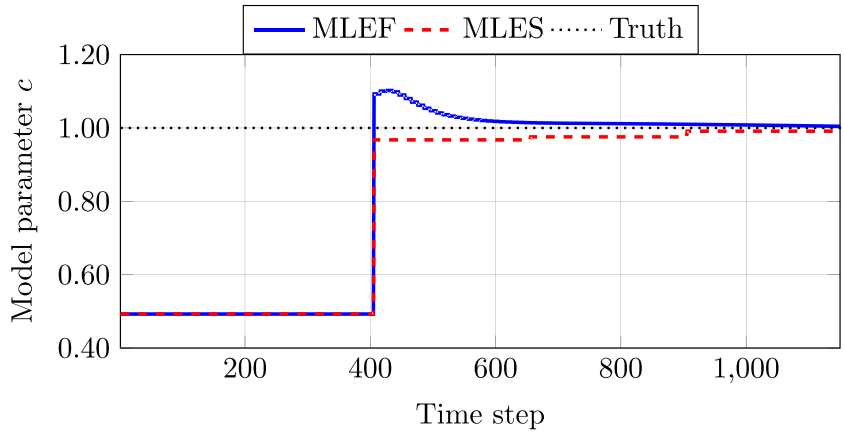

Fig. 22. Model parameter c convergence for the time-averaged synthesized observations.

data gathered would be from physical experiments. Thus, the concept of converging to the 'true' model parameters using synthesized data is inherently different than an actual physical scenario. Nevertheless, the methodology is applicable when using physical experimental data.

Future work will predominantly focus on the application of the MLES in a larger scale DA +CFD framework for flows with strong turbulence. In this study, the computational costs of the MLES and MLEF are comparable. For future larger systems, the computational performance will be compared as well. Improvements to the algorithm for computational efficiency must also be considered such as the usage of either the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) minimization method or a conjugate-gradient minimization approach, and covariance localization. Such improvements are out of scope of this study as the focus of this study is to develop and test the MLES method.

Conflict of interest

The authors declare that they have no conflict of interest.

References

Alekseev, A., Navon, I. & Steward, J. (2009) Comparison of advanced large-scale minimization algorithm for the solution of inverse ill-posed problems. *Optim. Methods Softw.*, **24**, 503–528.

Anderson, J. L. (2001) An ensemble adjustment Kalman filter for data assimilation. *Mon. Wea. Rev.*, **129**, 2884–2903.

Anderson, J. L. (2007) An adaptive covariance inflation error correction algorithm for ensemble fitlers. *Tellus*, **59A**, 520–526.

Asch, M., Bocquet, M. & Nodet, M. (2016) *Data Assimilation: Methods, Algorithms, and Applications*. Society for Industrial and Applied Mathematics.

Bocquet, M. & Sakov, P. (2014) An iterative ensemble Kalman smoother. *Q. J. R. Meterol. Soc.*, **140**, 1521–1535.

Carrassi, A., Vannitsem, S., Zupanski, D. & Zupanski, M. (2008) The maximum likelihood ensemble filter performances in chaotic systems. *Tellus A: Dynmanic Meteorology and Oceanography*, **61**, 587–600.

Chen, Y. & Oliver, D. S. (2012) Ensemble randomized maximum likelihood method as an iterative ensemble smoother. *Math. Geosci.*, **44**, 1–26.

Cohn, S. E, Sivakumaran, N. S.& Todling, R. (1994) A Fixed-Lag Kalman Smoother for Retrospective Data Assimilation. *Mon. Wea. Rev.*, **122**, 2838–2867.

Evensen, G. (2000) An Ensemble Kalman Smoother for Nonlinear Dynamics. *Mon. Wea. Rev.*, **128**, 1852–1867.

Evensen, G. (2018) Analysis of iterative ensemble smoothers for solving inverse problems. *Comput. Geosci.*, **22**, 885–908.

Gao, X. & Liu, J. (2014) Data Assimilation for Computational Fluid Dynamics Modeling of Combustion. *26th International Conference on Parallel Computational Fluid Dynamics*. Trondheim, Norway.

Gao, X., Wang Y., Overton, N. & May, I. (2015) Estimation of Flame Speed Model Parameter Using Ensemble Kalman Filter Algorithm. *Fall Meeting of the Western States Section of the Combustion Institute*. Provo, Utah, USA: Western States Section/Combustion Institute, pp. WSSCI 2015–134IE-0013.

Gao, X., Wang Y., Overton, N., Zupanski, M. & Tu, X. (2016) Properties of a Modified Ensemble Kalman Filter Algorithm for Combustion Application. *46th AIAA Fluid Dynamics Conference, Washington, D.C., USA*. Reston, Virginia, USA: American Institute of Aeronautics and Astronautics, Inc.

Gao, X., Wang Y., Overton, N., Zupanski, M. & Tu, X. (2017) Data-assimilated computational fluid dynamics modeling of convection-diffusion-reaction problems. *Journal of Computational Science*, **21**, 38–59.

He, C., Liu, Y. & Gan, L. (2018) A data assimilation model for turbulent flows using continuous adjoint formulation. *Phys. Fluids*, **30**, 105108-1–105108-13.

Hoteit, I., Luo, X. & Pham, D. (2012) Particle Kalman filtering: A nonlinear Bayesian framework for ensemble Kalman filters. *Mon. Wea. Rev.*, **140**, 528–542.

Hurst, C. & Gao, X. (2020) *Increasing the Finite Limit of Predictability in Turbulence by Data Assimilation*. Reston, Virginia, USA: Technical report, American Institute of Aeronautics and Astronautics, Inc.

Jardak, M., Navon, I. M. & Zupanski, M. (2010) Comparison of sequential data assimilation methods for the Kuramoto-Sivashinsky equation. *Int. J. Numer. Meth. Fluids*, **62**, 374–402.

Kalnay, E. (2003) *Atmospheric modeling, data assimilation and predictabilty*, 1st edn. Cambridge, United Kingdom: Cambridge University Press.

Kato, H., Yoshizawa, A., Ueno, G. & Obayashi, S. (2015) A data assimilation methodology for reconstructing turbulent flows around aircraft. *J. Comput. Phys.*, **283**, 559–581.

Kuramoto, Y. (1978) Diffusion-Induced Chaos in Reaction Systems. *Progress of Theoretical Physics Supplement*, **64**, 346–367.

Lorenz, E. N. (1995) *Predictability: A Problem Partly Solved*. Reading, United Kingdom: Seminar, ECMWF, Shinfield Park, Reading.

Menard, R., Cohn, S. E., Chang, L.-P. & Lyster, P. M. (2000) Assimilation of stratospheric chemical tracer observations using a Kalman filter. Part I: Formulation. *Mon. Wea. Rev.*, **128**, 2654–2671.

Ott, E., Hunt, B., Szunyogh, I., Zimin, A., Kostelich, E., Corazza, M., Kalnay, E., Patil, D. & Yorke, J. (2004) A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, **56**, 318.

Posselt, D. J., Hodyss, D., Bishop, C. H.& Pope, S. B.(2014) Errors in ensemble Kalman smoother estimates of cloud microphysical parameters. *Mon. Wea. Rev.*, **142**, 1631–1654.

Reagan, A. J., Dubief, Y., Dodds, P. S.& Danforth, C. M.(2016) Predicting Flow Reversals in a Computational Fluid Dynamics Simulated Thermosyphon Using Data Assimilation. *PLoS ONE*, **11**, 1–19.

Sivashinsky, G. I.(1977) Nonlinear analysis of hydrodynamic instability in laminar flames–I. Derivation of basic equations. *Acta Astronomica*, **4**, 1177–1206.

Wang Y., Walters, S., Guzik, S. & Gao, X. (2020) Data-Assimilated Large Eddy Simulation of Bluff-Body Stabilized Premixed Flames. *2020 AIAA SciTech Forum, Orlando, Florida, USA*. Reston, Virginia, USA: American Institute of Aeronautics and Astronautics, Inc.

Wang Y., Guzik, S., Zupanski, M. & Gao, X. (2021) The maximum likelihood ensemble filter for computational flame and fluid dynamics. *IMA J. Appl. Math.*, **86**, 631–661.

Whitaker, J. S. & Hamill, T. M. (2002) Ensemble data assimilation without perturbed observations. *Mon. Wea. Rev.*, **130**, 1913–1924.

Zou, X., Navon, I. M., Berger, M., Phua, K. H., Schlick, T. & Le Dimet, F. X. (1993) Numerical experience with limited-memory quasi-Newton and truncated Newton methods. *SIAM J. Contr. Optim.*, **3**, 582–608.

Zupanski, M. (2005) Maximum Likelihood Ensemble Filter: Theoretical aspects. *Mon. Wea. Rev.*, **133**, 1710–1726.

Zupanski, M., Navon, I. M. & Zupanski, D. (2008) The Maximum Likelihood Ensemble Filter as a non-differentiable minimization algorithm. *Q. J. R. Meteorol. Soc.*, **134**, 1039–1050.