Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models

Marc Bocquet, Julien Brajard, Alberto Carrassi, Laurent Bertino

Nonlin. Processes Geophys., 26, 143–162, 2019
https://doi.org/10.5194/npg-26-143-2019

Nonlinear Processes in Geophysics
Open Access
EGU

Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models

Marc Bocquet[1], **Julien Brajard**[2,3], **Alberto Carrassi**[3,4], **and Laurent Bertino**[3]

[1]CEREA, joint laboratory École des Ponts ParisTech and EDF R&D, Université Paris-Est, Champs-sur-Marne, France
[2]Sorbonne University, CNRS-IRD-MNHN, LOCEAN, Paris, France
[3]Nansen Environmental and Remote Sensing Center, Bergen, Norway
[4]Geophysical Institute, University of Bergen, Bergen, Norway

Correspondence: M. Bocquet (marc.bocquet@enpc.fr)

Received: 25 February 2019 – Discussion started: 28 February 2019
Revised: 28 May 2019 – Accepted: 11 June 2019 – Published: 10 July 2019

Abstract. Recent progress in machine learning has shown how to forecast and, to some extent, learn the dynamics of a model from its output, resorting in particular to neural networks and deep learning techniques. We will show how the same goal can be directly achieved using data assimilation techniques without leveraging on machine learning software libraries, with a view to high-dimensional models. The dynamics of a model are learned from its observation and an ordinary differential equation (ODE) representation of this model is inferred using a recursive nonlinear regression. Because the method is embedded in a Bayesian data assimilation framework, it can learn from partial and noisy observations of a state trajectory of the physical model. Moreover, a space-wise local representation of the ODE system is introduced and is key to coping with high-dimensional models.

It has recently been suggested that neural network architectures could be interpreted as dynamical systems. Reciprocally, we show that our ODE representations are reminiscent of deep learning architectures. Furthermore, numerical analysis considerations of stability shed light on the assets and limitations of the method.

The method is illustrated on several chaotic discrete and continuous models of various dimensions, with or without noisy observations, with the goal of identifying or improving the model dynamics, building a surrogate or reduced model, or producing forecasts solely from observations of the physical model.

1 Introduction

1.1 Data assimilation and model error

Data assimilation aims at estimating the state of a physical system from its observation and a numerical dynamical model for it. It has been successfully applied to numerical weather and ocean prediction, where it often consisted in estimating the initial conditions for the state trajectory of chaotic geofluids (Kalnay, 2002; Asch et al., 2016; Carrassi et al., 2018). This objective is impeded by the deficiencies of the numerical model (discretrisation, approximate physical schemes, unresolved scales, and their uncertain parametrisations, e.g. Harlim, 2017) and the difficulty in matching numerical representations of the system with the observations (representation error, Janjić et al., 2018). As a result, the quality of numerical weather predictions based on a methodologically sound data assimilation method crucially depends on both the sensitivity to the initial condition due to the chaotic unstable dynamics and on model error (Magnusson and Källén, 2013).

Model errors can take many forms, and accounting for them depends on the chosen data assimilation scheme. A first class of solutions relies on parametrising model error by, for instance, transforming the problem into a physical parameter estimation problem (e.g. Bocquet, 2012; Aster et al., 2013). Other solutions are based on a weakly parametrised form of model error, for instance when it is assumed to be additive noise. Such techniques have been developed for variational data assimilation (e.g. Trémolet, 2006; Carrassi and Vannitsem, 2010), for ensemble Kalman filters and

smoothers (e.g. Ruiz et al., 2013; Raanes et al., 2015), and for ensemble-variational assimilation (Amezcua et al., 2017; Sakov et al., 2018). In the weakly parametrised form, these methods should be completed by an estimation of the model error statistics (e.g. Pulido et al., 2018; Tandeo et al., 2019). Moreover, a model error's impact can be mitigated, and this is often the case in applications, by multiplicative and additive inflation (e.g. Whitaker and Hamill, 2012; Grudzien et al., 2018; Raanes et al., 2019) and by physically driven stochastic perturbations of model simulations in ensemble approaches (e.g. Buizza et al., 1999) or by stochastic subgrid parametrisations (e.g. Resseguier et al., 2017). This account is very far from exhaustive as this is a vast, multiform, and very active subject of research.

These approaches essentially seek to correct, calibrate, or improve an existing model using observations. Hence, they all primarily make use of data assimilation techniques.

1.2 Data-driven forecast of a physical system

An alternative is to renounce physically based numerical models of the phenomenon of interest and instead to only use observations of that system. Given the huge required datasets, this may seem a far-reaching goal for operational weather and ocean forecasting systems, but recent progress in data-driven methods and convincing applications to geophysical problems of small to intermediate complexity are strong incentives to investigate this bolder approach. Eventually, the perspective of putting numerical models away has a strong practical appeal, even though such a perspective may generate intense debates.

For instance, forecasting of a physical system can be done by looking up past situations and patterns using the techniques of analogues, which can be combined with present observations using data assimilation (Lguensat et al., 2017), or it can rely on a representation of the physical system based on diffusion maps that look for a spectral representation of the dataset (see chapter 6 of Harlim, 2018). An original data-driven stochastic modelling approach has been developed by Kondrashov et al. (2015). The method, recently extended to deal with multi-scale datasets (Kondrashov and Chrekroun, 2017), has been applied to successfully estimate reduced models of geophysical phenomena (see e.g. Kondrashov et al., 2018, and references therein). A fourth route relies on neural networks and deep learning to represent the hidden model and make forecasts from this representation. Examples of such an approach applied to the forecasting of low-order chaotic geophysical models are Park and Zhu (1994), who used a bilinear recurrent neural network and applied it to the three-variable Lorenz model (Lorenz, 1963, hereafter L63), Pathak et al. (2017, 2018), who use reservoir network techniques on the L63 model and on the Kuramoto–Sivashinski model (Kuramoto and Tsuzuki, 1976; Sivashinsky, 1977, hereafter KS), and Dueben and Bauer (2018), who use a neural network on a low-order Lorenz three-scale model and on coarse two-dimensional geopotential height maps at 500 hPa. The last three contributions have to resort to local reservoir networks or convolutional layers, respectively, to cope with the dimensionality of the models. However, all these representations are not mechanistic and the neural network becomes a surrogate for the hidden model. This marks a key distinction with respect to our approach where the dynamics to be determined are explicitly formulated, as will be clarified later.

1.3 Learning the dynamics of a model from its output

Data-driven techniques that seek to represent the model in a more explicit manner, and therefore with a greater interpretability, may use specific classes of nonlinear regression as advocated by Paduart et al. (2010) and Brunton et al. (2016). With a view to forecasting dynamical systems, it is possible to design neural networks in order to reflect the iterative form of a Runge–Kutta (RK) integration scheme. Wang and Lin (1998) proposed and achieved such a goal using classical activation functions, which may however blur the interpretation of the underlying dynamics. Fablet et al. (2018) went further and used a bilinear residual neural network structured so as to mimic a fourth-order RK scheme (RK4) and noise-free data. Using the Keras tool with the TensorFlow backend, their approach proved to be a very effective tool for the L63 model and to a lesser extent for the 40-variable Lorenz model (Lorenz and Emanuel, 1998, hereafter L96). In particular, they retrieved the parameters of the L63 equations to a high precision. Long et al. (2018) sought the operators of the partial differential equations (PDEs) of a physical system by identifying differentiations with convolution operators of a feed-forward neural network. They successfully applied their method to advection–diffusion problems. As opposed to our proposal, described hereafter, none of the aforementioned techniques is embedded in a Bayesian framework, making them less suitable for working with noisy and partial data.

1.4 Goal and outline

From this point on, the physical system under scrutiny will be called the *reference* model. It will be assumed to be known only from observations. We follow a data-driven approach inspired by the works of Paduart et al. (2010) and Fablet et al. (2018) in the sense that we will consider an observed physical reference model, which might be generated by a hidden mathematical model or process. This work is focused on either one or a combination of the following goals: (i) to build a *surrogate* model for the dynamics, (ii) to produce forecasts that emulate those of the reference model, and (iii) to identify the underlying dynamics of the reference model given by a mathematical model. The reference model could be totally unknown or only partially specified.

To achieve these goals, we introduce a surrogate model defined by a set of ordinary differential equations (ODEs):

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{\phi}(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{N_x}$ is the state vector and $\boldsymbol{x} \longmapsto \boldsymbol{\phi}(\boldsymbol{x})$ is a vector field that we shall call the *flow rate*. For the sake of simplicity, the dynamics in this study are supposed to be autonomous, i.e. do not explicitly depend on time. Our technique seeks a fit $\boldsymbol{\phi}$ given observations of the reference model. This is a rather general representation since, for instance, PDEs can be discretised into ODEs. We will restrict ourselves to the case where $\boldsymbol{\phi}$ is at most quadratic in $\{x_n\}_{0\leq n<N_x}$. The numerical integration of Eq. (1) could be based on any RK scheme, but should additionally rely on the composition of such integration steps. As a result, quite general resolvents of Eq. (1) can be built (the resolvent is the model, i.e. the flow rate, integrated in time over a finite time interval).

Importantly, we will not require any machine learning software tool since the adjoint of the model resolvent can be derived without a lot of effort. As opposed to the contributions mentioned in the previous subsections, we embed the technique in a data assimilation framework. From a data assimilation standpoint, the technique can be seen as meant to deal with model error (with or without some prior on the model) and it naturally accommodates partial and noisy observations. Moreover, we will build representations of the dynamics that are either invariant by spatial translation (homogeneous) and/or local (i.e. the flow rate of a variable x_n only depends on neighbouring variables whose perimeter is defined by a stencil). These properties make our technique scalable and thus potentially applicable to high-dimensional systems.

In Sect. 2, we present model identification as a Bayesian data assimilation problem. We first choose an ODE representation of the dynamics, introduce a nonlinear regressor basis, and define the integration schemes we will work with. We describe the local and homogeneous representations as physically based simplifications of the most general case, and we derive the gradient of the problem's cost function based on these representations. We then introduce the Bayesian problem and the resulting cost function used for joint supervised learning of the optimal representation and estimation of the state trajectory. The latter is the standard goal of data assimilation, while the former is that of machine learning. Our approach blends them together using the formalism of data assimilation.

In Sect. 3, we discuss several theoretical issues: the prior of the model, the convergence of the training step, the connection with numerical analysis of integration schemes, the connection with deep learning architectures, and, finally, the pros and cons of our approach.

In Sect. 4, we illustrate the method with several low-order chaotic models (L63, L96, KS, and a two-scale Lorenz model) of various sizes, from a perfectly identifiable model, i.e. where the model used to generate the dataset can be retrieved exactly, to a reduced-order model where the model used to generate the dataset cannot be retrieved exactly, using full or partial, noiseless, or noisy observations. Conclusions are given in Sect. 5.

2 Model identification as a data assimilation problem

2.1 Ordinary differential equation representation

Our surrogate model is chosen to be represented by an ODE system as described by Eq. (1). We additionally assume that the flow rate can be written as

$$\boldsymbol{\phi}_{\mathbf{A}}(\boldsymbol{x}) = \mathbf{A}\boldsymbol{r}(\boldsymbol{x}), \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{N_x \times N_p}$ is a matrix of real coefficients to be estimated and $\boldsymbol{r} : \mathbb{R}^{N_x} \longmapsto \mathbb{R}^{N_p}$ is a map that defines regressor functions of $\boldsymbol{x}$. $\mathbb{R}^{N_p}$ is the latent space of the regressors in which the flow rate is linear.

In the absence of any peculiar symmetry, we choose this map to list all the monomials up to second order built on $\boldsymbol{x}$, i.e. the constant, linear, and bilinear monomials. Let us call $\mathcal{D} = \{0, 1, \ldots, N_x - 1\}$ the set of all variable indices and $\mathcal{P}$ the set of all pairs of variable indices. We introduce the augmented state vector

$$\widetilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \in \mathbb{R}^{N_x+1}, \tag{3}$$

extend $\mathcal{D}$ to $\widetilde{\mathcal{D}} = \mathcal{D} \cup \{N_x\}$, and define $\widetilde{\mathcal{P}}$ as the distinct pairs of variable indices in $\widetilde{\mathcal{D}}$.

As a result, the regressors are compactly defined by

$$\boldsymbol{r}(\boldsymbol{x}) = \left[\{\widetilde{x}_n \widetilde{x}_m\}_{(n,m)\in\widetilde{\mathcal{P}}}\right], \tag{4}$$

where the scalars in the bracket are the entries of the vector $\boldsymbol{r}(\boldsymbol{x})$. We count

$$N_p = \binom{N_x+1}{2} = \frac{1}{2}(N_x+1)(N_x+2) \tag{5}$$

regressors, i.e. the cardinal of $\widetilde{\mathcal{P}}$. For instance, a model with three variables, x_0, x_1, and x_2, such as L63, has 10 such regressors:

$$\left[1, x_0, x_1, x_2, x_0^2, x_0x_1, x_0x_2, x_1^2, x_1x_2, x_2^2\right]. \tag{6}$$

Higher-order regressors, as well as regressors of different functional forms, could be included as in Brunton et al. (2016). However, it is important to keep in mind that we do not seek an expansion of the resolvent of the reference model, but of the flow rate $\boldsymbol{\phi}_{\mathbf{A}}$. As a consequence, higher-order products of the state variables are anyhow generated by the integration schemes and their composition. It is worth mentioning that nonlinear regressions are not widespread in geophysical data assimilation. We are nonetheless aware of at least one noticeable exception that extends traditional Gaussian-based methods (Hodyss, 2011, 2012).

2.2 Local and homogeneous representations

At least two useful simplifications for the ODEs could be exploited if the state $\boldsymbol{x}$ is assumed to be the discretisation of a spatial field.

2.2.1 Locality

First, we use a locality assumption based on the physical locality of the system: all multivariate monomials in the ODEs have variables x_n that belong to a stencil, i.e. a local arrangement of grid points around a given node. This can significantly reduce the number of bilinear monomials in $\boldsymbol{r}(\boldsymbol{x})$. We assume that s_n is the stencil around node n, the pattern being the same for all nodes except the last one. For the node N_x corresponding to the extra variable $\widetilde{x}_{N_x} = 1$, we assume that its stencil consists of all the $N_x + 1$ nodes. We then define $\widetilde{\mathcal{P}}_s \subset \widetilde{\mathcal{P}}$ as the sub-set of all pairs (n, m) of variables for which $m \in s_n$. The set of required monomials can therefore be reduced to

$$\boldsymbol{r}(\boldsymbol{x}) = \left[\{\widetilde{x}_n \widetilde{x}_m\}_{(n,m)\in\widetilde{\mathcal{P}}_s}\right]. \tag{7}$$

Under these conditions, $\mathbf{A}$ becomes sparse. Indeed, for each node n, we assume that $\dot{x}_n$, the time derivative of x_n, is impacted only by linear terms x_m such that $m \in s_n$ and quadratic terms $x_m x_l$ such that $m \in s_n$, $l \in s_n$, and $m \in s_l$. However, to keep a dense matrix, we choose to compactly redefine and shrink $\mathbf{A}$ by eliminating all a priori zero entries due to the locality assumption. The number of columns of $\mathbf{A}$ is then significantly reduced from N_p to N_a. As a consequence of this redefinition of $\mathbf{A}$, the matrix multiplication in between $\mathbf{A}$ and $\boldsymbol{r}(\boldsymbol{x})$ must be changed accordingly. Nonetheless, the operation that assigns coefficients in $\mathbf{A}$ to the monomials in $\boldsymbol{r}(\boldsymbol{x})$ remains linear, and we write it as

$$\boldsymbol{\phi}_\mathbf{A}(\boldsymbol{x}) = \mathbf{A} \cdot \boldsymbol{r}(\boldsymbol{x}). \tag{8}$$

Let us take the example of a one-dimensional extended space as those used in Sect. 4. The domain is supposed to be periodic (circle) and the nodes are indexed by $0 \leq n < N_x$. Recall that the node of index N_x is associated with the extra $\{1\}$. For $0 \leq n < N_x$, the stencil s_n is defined as the set of $2L+1$ nodes of index $n-L, n-L+1, \ldots n+L-1, n+L$, plus the extra node of index N_x. The stencil s_{N_x} consists of all the nodes, i.e. $\mathcal{D}$. We assume $2L+1 \leq N_x$. In that case $\boldsymbol{r}(\boldsymbol{x})$ as defined by Eq. (7) has $N_p = 1 + N_x(2+L)$ monomials. For instance, there are 161 such regressors for a 40-variable model defined on a circular domain, such as L96, with $L = 2$:

$$\begin{aligned}
&[1,\\
&x_0, x_1, \ldots, x_{39},\\
&x_0^2, x_0x_1, x_0x_2,\\
&x_1^2, x_1x_2, x_1x_3,\\
&\vdots\\
&x_{39}^2, x_{39}x_0, x_{39}x_1\Big].
\end{aligned} \tag{9}$$

The row $[\mathbf{A}]_n$ of the dense $\mathbf{A}$ contains the following coefficients for each $0 \leq n < N_x$. First there are $2L+2$ regressors built with $\{1\}$ (the constant and linear regressors). Second, we consider the square monomials x_m^2 with $m \in s_n$, i.e. $\left\{x_m^2\right\}_{n-L\leq m\leq n+L}$ whose number is $2L+1$. Then we consider those separated by one space step, $\{x_m x_{m+1}\}_{n-L\leq m\leq n+L-1}$ whose number is $2L$, followed by those separated by two space steps whose number is $2L-1$, and so on until a separation of L is reached. Quadratic monomials of greater separation are discarded since they do not belong to a common stencil as per the above definition reflecting the locality assumption. Hence there is a total of $N_\mathrm{a} = \sum_{l=L+1}^{2L+2} l = \frac{3}{2}(L+1)(L+2)$ coefficients per grid cell.

In Appendix A, we show in the one-dimensional space case how to compute the reduced form of the product between $\mathbf{A}$ and $\boldsymbol{r}(\boldsymbol{x})$, assuming locality. This type of technical parametrisation is required for a parsimonious representation of the control variables, i.e. the coefficients of $\mathbf{A}$, and is key for a successful implementation with high-dimensional models.

Note that this locality assumption is hardly restrictive. Indeed, owing to the absence of long-range instantaneous interactions (which are precluded in geophysical fluids), farther distance correlations between state variables can be generated by small stencils in the definition of $\boldsymbol{\phi}_\mathbf{A}$ through time integrations. This would not prevent potential specific long-distance dependencies (such as teleconnections).

2.2.2 Homogeneity

Furthermore, a symmetry hypothesis could optionally be used by assuming translational invariance of the ODEs, called *homogeneity* in the following. Because our control parameters, i.e. the coefficients of $\mathbf{A}$, parametrise the flow rate, the symmetry simply translates into the rows $[\mathbf{A}]_n$ of the dense $\mathbf{A}$ being the same for all n. Hence $\mathbf{A}$ simply becomes a vector in $\mathbb{R}^{N_\mathrm{a}}$.

Let us enumerate its coefficients in the case of the L96 model with $L = 2$ and assuming both locality and homogeneity. The coefficients are partitioned into $A^{(0)}$ for the bias, $A_l^{(1)}$ for the linear sector, and $A_{l,m}^{(2)}$ for the bilinear sector. In the linear sector, $l = -2, \ldots, 2$ is the relative position with respect to the current grid point. In the bilinear sector, l, m are the relative positions with respect to the current grid point of

the two variables in the product. Proceeding in the same way we counted them, the $N_\mathrm{a} = 18$ coefficients of $\mathbf{A}$ are

$$\begin{aligned}&\Big[A^{(0)},\\ &A^{(1)}_{-2}, A^{(1)}_{-1}, A^{(1)}_{0}, A^{(1)}_{1}, A^{(1)}_{2},\\ &A^{(2)}_{-2,-2}, A^{(2)}_{-1,-1}, A^{(2)}_{0,0}, A^{(2)}_{1,1}, A^{(2)}_{2,2},\\ &A^{(2)}_{-2,-1}, A^{(2)}_{-1,0}, A^{(2)}_{0,1}, A^{(2)}_{1,2},\\ &A^{(2)}_{-2,0}, A^{(2)}_{-1,1}, A^{(2)}_{0,2},\Big].\end{aligned} \tag{10}$$

Note that while both constraints, locality and homogeneity, apply to the ODEs, they do not apply to the states per se. For instance, ODEs for discretised homogeneous two-dimensional turbulence satisfy both constraints and yet generate non-uniform flows.

For realistic geofluids, the forcing fields (solar irradiance, bathymetry, boundary conditions, friction, etc.) are heterogeneous, so that the homogeneity assumption should be dropped. Nonetheless, the fluid dynamics part of the model would remain homogeneous. As a result, a hybrid approach could be enforced.

2.3 Integration scheme and cycling

The reference model will be observed at time steps t_k, indexed by integer $0 \leq k \leq K$. Hence, we need to be able to express the resolvent of the surrogate model from t_k to t_{k+1}. We assume that $t_{k+1} - t_k$ is a multiple of the integration time step of the surrogate model, $t_{k+1} - t_k = N_\mathrm{c}^k h$, where h is the integration time step and N_c^k is the number of integrations. The time steps $t_{k+1} - t_k$ can be uneven, which is reflected in the dependence of N_c^k on k. Hence, the resolvent of the surrogate model from t_k to t_{k+1} can be written as

$$\boldsymbol{x}_{k+1} = \mathbf{F}_\mathbf{A}^k(\boldsymbol{x}_k), \quad \text{where} \quad \mathbf{F}_\mathbf{A}^k \equiv \mathbf{f}_\mathbf{A}^{N_\mathrm{c}^k} \equiv \underbrace{\mathbf{f}_\mathbf{A} \circ \ldots \circ \mathbf{f}_\mathbf{A}}_{N_\mathrm{c}^k \text{ times}}, \tag{11}$$

i.e. the integration of Eq. (1) from t_k to t_{k+1} using the representation Eq. (2).

We define intermediate state vectors in between $[t_k, t_{k+1}]$: $\boldsymbol{x}_{k,l}$ is the state vector defined at time $t_k + (t_{k+1} - t_k)l/N_\mathrm{c}^k$ for $0 \leq l \leq N_\mathrm{c}^k$, as the result of l compositions of $\mathbf{f}_\mathbf{A}$ on $\boldsymbol{x}_k$: $\boldsymbol{x}_{k,l} = \mathbf{f}_\mathbf{A}^l(\boldsymbol{x}_k)$. Figure 1 is a schematic of the composition of the integration steps, along with the state vectors $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k,l}$.

The operator $\mathbf{f}_\mathbf{A}$ is meant to be an explicit numerical integration scheme. In the following, we shall consider an RK scheme applied to $\boldsymbol{x} \equiv \boldsymbol{x}_{k,l}$, with N_RK steps. This number of steps coincides with the accuracy of the schemes that we will consider: first order for the Euler scheme, second order for RK2, and fourth order for RK4 ($N_\mathrm{RK} = 1, 2$, and 4, respectively). Provided the dynamics are autonomous, a general RK scheme reads as

$$\mathbf{f}_\mathbf{A}(\boldsymbol{x}) = \boldsymbol{x} + h \sum_{i=0}^{N_\mathrm{RK}-1} \beta_i \mathbf{k}_i, \tag{12a}$$

$$\mathbf{k}_i = \boldsymbol{\phi}_\mathbf{A}\left(\boldsymbol{x} + h \sum_{j=0}^{i-1} \alpha_{i,j} \mathbf{k}_j\right), \tag{12b}$$

where the coefficients β_i and $\alpha_{i,j}$ entirely specify the scheme and $h = (t_{k+1} - t_k)/N_\mathrm{c}^k$. Note that $\alpha_{i,j}$ are zero for $j \geq i$, so that Eq. (12b) can be computed iteratively from $\mathbf{k}_0$ to $\mathbf{k}_{N_\mathrm{RK}-1}$, followed by the sum Eq. (12a) to get $\mathbf{f}_\mathbf{A}(\boldsymbol{x})$.

In the following, h will be absorbed into the definition of $\mathbf{A}$ and hence $\boldsymbol{\phi}_\mathbf{A}$, so that we can take $h = 1$ without loss of generality.

2.4 Bayesian analysis

We consider a sequence of observation vectors $\boldsymbol{y}_k \in \mathbb{R}^{N_y^k}$ of the physical system at t_k indexed by $0 \leq k \leq K$. The system state is observed through

$$\boldsymbol{y}_k = \mathbf{H}_k(\boldsymbol{x}_k) + \boldsymbol{\epsilon}_k, \tag{13}$$

where $\mathbf{H}_k$ is the observation operator at time t_k. The observation error $\boldsymbol{\epsilon}_k$ will be assumed to be Gaussian with zero mean and covariance matrix $\boldsymbol{r}_k$. It is also assumed to be white in time. The flow rate $\boldsymbol{\phi}_\mathbf{A}$ is given by the approximation Eq. (2), so that the resolvent $\mathbf{F}_\mathbf{A}$ of the surrogate model should also be considered an approximation of the reference model's resolvent. Hence, we generalise Eq. (11) to

$$\boldsymbol{x}_{k+1} = \mathbf{F}_\mathbf{A}^k(\boldsymbol{x}_k) + \boldsymbol{\eta}_k, \tag{14}$$

where $\boldsymbol{\eta}_k$ are unbiased Gaussian errors of covariance matrices $\mathbf{Q}_k$, supposed to be white in time and uncorrelated from the observation errors. Note that, in all generality, the state space of the surrogate model does not have to match that of the reference model. We will nonetheless take them to coincide here merely for simplicity.

With the goal of identifying a model or building a surrogate of the reference one, we are interested in estimating the probability density function (pdf) $p(\mathbf{A}|\boldsymbol{y}_{0:K})$, where $\boldsymbol{y}_{0:K}$ stands for all observations in the window $[t_0, t_K]$. To obtain a tractable expression for this conditional likelihood, we need to marginalise over the state variables $\boldsymbol{x}_{0:K}$ within the window:

$$p(\mathbf{A}|\boldsymbol{y}_{0:K}) = \int \mathrm{d}\boldsymbol{x}_{0:K}\, p(\mathbf{A}, \boldsymbol{x}_{0:K}|\boldsymbol{y}_{0:K}). \tag{15}$$

An approximate maximum a posteriori for $\mathbf{A}$ could be obtained by using the Laplace approximation of this integral,

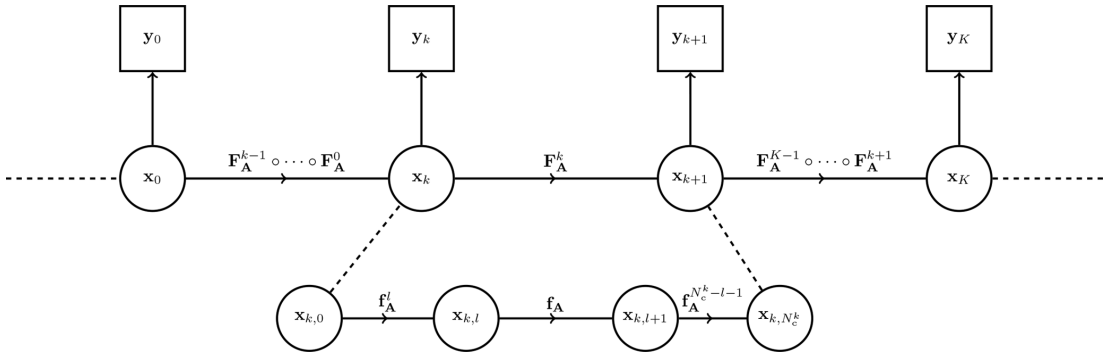

Figure 1. Representation of the data assimilation system as a hidden Markov chain model and of the model resolvents $\mathbf{F}_\mathbf{A}^k$ and $\mathbf{f}_\mathbf{A}$.

which would require finding the maximum of

$$\begin{aligned} p(\mathbf{A}, \boldsymbol{x}_{0:K} | \boldsymbol{y}_{0:K}) &= \frac{p(\mathbf{A}, \boldsymbol{x}_{0:K}, \boldsymbol{y}_{0:K})}{p(\boldsymbol{y}_{0:K})} \\ &= \frac{p(\boldsymbol{x}_{0:K}, \boldsymbol{y}_{0:K} | \mathbf{A}) p(\mathbf{A})}{p(\boldsymbol{y}_{0:K})} \\ &= \frac{p(\boldsymbol{y}_{0:K} | \boldsymbol{x}_{0:K}, \mathbf{A}) p(\boldsymbol{x}_{0:K} | \mathbf{A}) p(\mathbf{A})}{p(\boldsymbol{y}_{0:K})}. \end{aligned} \quad (16)$$

Nonetheless, maximising Eq. (16) rigorously yields the maximum a posteriori of the *joint* variables $\mathbf{A}, \boldsymbol{x}_{0:K}$. The cost function associated with this joint pdf is by definition $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K}) = -\ln(p(\mathbf{A}, \boldsymbol{x}_{0:K} | \boldsymbol{y}_{0:K}))$. Because Eq. (14) is Markovian and given the Gaussian form of both model and observational errors, the cost function reads as

$$\begin{aligned} \mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K}) = & \frac{1}{2} \sum_{k=0}^{K} \left\| \boldsymbol{y}_k - \mathbf{H}_k(\boldsymbol{x}_k) \right\|_{\boldsymbol{r}_k^{-1}}^2 \\ & + \frac{1}{2} \sum_{k=1}^{K} \left\| \boldsymbol{x}_k - \mathbf{F}_\mathbf{A}^{k-1}(\boldsymbol{x}_{k-1}) \right\|_{\mathbf{Q}_k^{-1}}^2 \\ & - \ln p(\boldsymbol{x}_0 | \mathbf{A}) - \ln p(\mathbf{A}), \end{aligned} \quad (17)$$

up to a constant depending on $\mathbf{Q}_{1:K}$ and $\boldsymbol{r}_{0:K}$ only. The vector norm $\|z\|_\mathbf{P}$ is defined as $\sqrt{z^\mathrm{T} \mathbf{P} z}$. This is the cost function of a weak constraint 4D-Var (see Sect. 2.4.3.2 of Asch et al., 2016) with $\mathbf{A}$ and $\boldsymbol{x}_{0:K}$ as control variables.

In the case where the reference model is fully and directly observed, i.e. $\mathbf{H}_k \equiv \mathbf{I}_x$, and in the absence of observation noise, i.e. $\boldsymbol{r}_k \equiv \mathbf{0}$, we have $\boldsymbol{x}_{0:K} \equiv \boldsymbol{y}_{0:K}$ and the cost function simplifies to

$$\begin{aligned} \mathcal{J}(\mathbf{A}) = & \frac{1}{2} \sum_{k=1}^{K} \left\| \boldsymbol{y}_k - \mathbf{F}_\mathbf{A}^{k-1}(\boldsymbol{y}_{k-1}) \right\|_{\mathbf{Q}_k^{-1}}^2 \\ & - \ln p(\boldsymbol{y}_0 | \mathbf{A}) - \ln p(\mathbf{A}), \end{aligned} \quad (18)$$

where $\boldsymbol{y}_{0:K}$ is the fully and perfectly observed state trajectory of the reference model. This is notably similar to a traditional least-square function used in machine and deep learning regression. This connection between machine learning and data assimilation cost functions had been previously put forward by Hsieh and Tang (1998) and Abarbanel et al. (2018), although in a different form. Reciprocally, when the aforementioned hypotheses of noiseless and complete observations do not hold, Eq. (17) can be seen as a natural data assimilation extension of Eq. (18). Note that Eq. (18) only depends on the sequence $\mathbf{Q}_{1:K}$. If, in addition, the dependence on $p(\boldsymbol{y}_0, \mathbf{A}) = p(\boldsymbol{y}_0 | \mathbf{A}) p(\mathbf{A})$ is neglected in Eq. (18), then the maximum a posteriori should not depend on a global rescaling of $\mathbf{Q}_{1:K}$.

The data assimilation system is represented in Fig. 1 as a hidden Markov chain model. This Bayesian view highlights the choice that must be made for $\boldsymbol{r}_{0:K}$ and/or $\mathbf{Q}_{1:K}$ and provides an interpretation in terms of errors. Furthermore, one could implement an objective estimation of these error statistics as in Pulido et al. (2018).

2.5 Gradients and adjoint of the representation

To efficiently minimise the cost function Eq. (17) with a gradient-based optimisation tool, we need to analytically derive the gradient of Eq. (17) with respect to both $\mathbf{A}$ and $\boldsymbol{x}_{0:K}$. As for $\boldsymbol{x}_{0:K}$, we have

$$\begin{aligned} \nabla_{\boldsymbol{x}_0} \mathcal{J} = & - \left(\nabla_{\boldsymbol{x}_0} \mathbf{H}_0 \right)^\mathrm{T} \boldsymbol{r}_0^{-1} \boldsymbol{\delta}_0 - \left(\nabla_{\boldsymbol{x}_0} \mathbf{F}_\mathbf{A}^0 \right)^\mathrm{T} \mathbf{Q}_1^{-1} \boldsymbol{\Delta}_1 \\ & - \nabla_{\boldsymbol{x}_0} \ln p(\boldsymbol{x}_0 | \mathbf{A}), \end{aligned} \quad (19a)$$

$$\begin{aligned} \nabla_{\boldsymbol{x}_k} \mathcal{J} = & - \left(\nabla_{\boldsymbol{x}_k} \mathbf{H}_k \right)^\mathrm{T} \boldsymbol{r}_k^{-1} \boldsymbol{\delta}_k - \left(\nabla_{\boldsymbol{x}_k} \mathbf{F}_\mathbf{A}^k \right)^\mathrm{T} \mathbf{Q}_{k+1}^{-1} \boldsymbol{\Delta}_{k+1} \\ & + \mathbf{Q}_k^{-1} \boldsymbol{\Delta}_k, \quad \text{for} \quad 1 \le k \le K-1, \end{aligned} \quad (19b)$$

$$\nabla_{\boldsymbol{x}_K} \mathcal{J} = - \left(\nabla_{\boldsymbol{x}_K} \mathbf{H}_K \right)^\mathrm{T} \boldsymbol{r}_K^{-1} \boldsymbol{\delta}_K + \mathbf{Q}_K^{-1} \boldsymbol{\Delta}_K, \quad (19c)$$

where $\boldsymbol{\delta}_k = \boldsymbol{y}_k - \mathbf{H}_k(\boldsymbol{x}_k)$ for $0 \le k \le K$ and $\boldsymbol{\Delta}_k = \boldsymbol{x}_k - \mathbf{F}_\mathbf{A}^{k-1}(\boldsymbol{x}_{k-1})$ for $1 \le k \le K$; $\nabla_{x_k} \mathbf{F}_\mathbf{A}^k$ is the tangent linear operator of the resolvent $\mathbf{F}_\mathbf{A}^k$ computed at $\boldsymbol{x}_k$ for $0 \le k < K$; $\nabla_{x_k} \mathbf{H}_k$ is the tangent linear operator of the observation oper-

ator $\mathbf{H}_k$ computed at $\boldsymbol{x}_k$ for $0 \leq k \leq K$. As for $\mathbf{A}$, we have

$$\nabla_{\mathbf{A}} \mathcal{J} = -\sum_{k=1}^{K} \boldsymbol{\delta}_k^{\mathrm{T}} \mathbf{Q}_k^{-1} \nabla_{\mathbf{A}} \mathbf{F}_{\mathbf{A}}^{k-1}(\boldsymbol{x}_{k-1}) - \nabla_{\mathbf{A}} \ln p(\mathbf{A}), \tag{20}$$

assuming $\boldsymbol{x}_0$ is independent of $\mathbf{A}$. Hence, a key technical aspect of the problem is to compute the tangent linear and adjoint operators required by these gradients. In this paper, we assume that the adjoints $\left(\nabla_{\boldsymbol{x}_k} \mathbf{H}_k\right)^{\mathrm{T}}$ of the tangent linear operators of the observation operators are known, for instance if the latter are linear as in Sect. 4.

The computations of the gradients and the required adjoints are developed in Appendix B. These technicalities can be skipped since they are not required to understand the method. Nonetheless, they are critical to its numerical efficiency.

3 Discussion of the theoretical points

In this section, we discuss the prior pdf $p(\mathbf{A})$, the optimisation of the cost function $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K})$, and the connections with deep learning techniques.

3.1 Prior information on the reference model

The goal is either to reconstruct an ODE for the reference model, characterised by the coefficients $\mathbf{A}$ through $\boldsymbol{\phi}_{\mathbf{A}}$, or to build a surrogate model of it. The estimation of $\boldsymbol{x}_{0:K}$ is then accessory even though factually critical to the estimation of $\mathbf{A}$. The alternative would have been to consider the estimation of $\boldsymbol{x}_{0:K}$ as the primary problem, under model error of a prescribed ODE form, the estimation of $\mathbf{A}$ becoming accessory. In both cases, but particularly in the latter one, one may benefit from an informative prior pdf $p(\mathbf{A})$.

The prior pdf $p(\mathbf{A})$ can be used to encode any prior knowledge on the reference model, such as pieces of it that would be known. Indeed, $p(\mathbf{A})$ can formally quantity the uncertainty associated with any part of the surrogate model. For instance, assume that the reference model is partially identifiable, which means that part of the reference model could be represented by an up to bilinear flow rate of the form Eq. (2) and Eq. (4). Moreover, assume that there is one such part of the reference model which is known, i.e. that elements of $\mathbf{A}$ are actually known, while others need to be estimated. Then, the known coefficients can formally be encoded in $p(\mathbf{A})$ with Dirac factors. In practice it could be implemented as a constrained optimisation problem, for instance using an augmented Lagrangian, in order to avoid significantly altering the gradients with respect to $\mathbf{A}$. More generally, assigning a non-trivial prior likelihood, such as a Gaussian one for $\mathbf{A}$, is certainly appealing but may not be practical.

3.2 Numerical optimisation: issues and solutions

The success of the optimisation of $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K})$ depends above all on the ability to evaluate it robustly. In particular, it depends on the stability of the numerical integration scheme $\boldsymbol{x}' = \mathbf{f}_{\mathbf{A}}(\boldsymbol{x})$. In this paper, we chose to rely on one-step explicit schemes which are much simpler to describe and efficient to integrate (a family to which the RK schemes of any N_{RK} belong). These schemes are 0-stable, which means that the finite time error growth goes to zero as the integration step goes to zero. But, as a major drawback, they have a limited absolute (or A-)stability domain (see e.g. chapters 5 and 6 of Gautschi, 2012). For a given state trajectory, there exists a stability domain $\mathcal{D}_{\mathrm{s}} \in \mathbb{R}^{N_x \times N_p}$ out of which the evaluation of $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K})$ is hazardous. A failure to evaluate the cost function also depends on the number of integration steps since the instabilities are likely to increase exponentially with N_{c}^k.

This says that, in the absence of a strong prior $p(\mathbf{A})$, it is safer to start with $\mathbf{A} = \mathbf{0}$ likely to lie close to $\mathcal{D}_{\mathrm{s}}$. Alternatively, if stability constraints are known about $\mathbf{A}$, they could be encoded in $p(\mathbf{A})$. It also says that we should strike an empirical compromise between the composition numbers N_{c}^k and the easiness in evaluating $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K})$. On the one hand, the larger N_{c}^k, the more the iterates of $\mathbf{A}$ in the optimisation must be kept confined in $\mathcal{D}_{\mathrm{s}}$. On the other hand, the longer N_{c}^k, the broader the class of achievable resolvents and hence the ability to build a good surrogate. Moreover, the higher the stability of the integration scheme, the larger $\mathcal{D}_{\mathrm{s}}$, and the easier the optimisation in spite of an increase in its numerical cost.

As for the sensitivity on K, the longer the time window, the more observations are available to constrain the problem. However, the longer K, the higher the chances of having a significant instability: the chances of a successful integration typically decrease exponentially with the length K.

This stability issue can be somehow alleviated by normalising the observations $\boldsymbol{y}_k$ by their mean and variance in order to avoid excessively large value ranges of the regressors. This will not change the fundamental stability of the schemes, yet may delay the occurrences of instabilities due to the nonlinear terms.

Moreover, instabilities can significantly be mitigated by replacing the monomials with smoothed or truncated ones:

$$\boldsymbol{r}(\boldsymbol{x}) = \left[\{\zeta(\widetilde{x}_n)\zeta(\widetilde{x}_m)\}_{(n,m)\in\widetilde{\mathcal{P}}} \right]. \tag{21}$$

One can for instance choose $\zeta(x) = \lambda \tanh(x/\lambda)$, in order to cut off too large values of $|x|$ and hence delay the growth of instabilities. The parameter $\lambda > 0$ is roughly chosen as the typical maximum amplitude of $|x|$ as approximately inferred from the observations. If tanh is deemed to be numerically too costly, one can choose instead $\zeta(x) = -\lambda 1_{]-\infty,-\lambda]} + x 1_{[-\lambda,\lambda]} + \lambda 1_{[\lambda,+\infty[}$ or more generally $\zeta(x) = (-\lambda + \varepsilon(x +$

$\lambda))1_{]-\infty,-\lambda]}+x1_{[-\lambda,\lambda]}+(\lambda+\varepsilon(x-\lambda))1_{[\lambda,+\infty[}$, with $0 \leq \varepsilon \ll 1$ a small trend.

This latter change in variables is the one implemented for all numerical applications described in Sect. 4, together with the normalisation. From these experiments, we learned that these tricks often turned critical in the first iterates of the optimisation when the estimate of $\mathbf{A}$ progressively migrated to the A-stability domain. After a few iterations, however, the integrations are stabilised and the nonlinear regime of the truncations in Eq. (21) is not tapped into anymore.

3.3 Connection and analogies with deep learning architectures

It has recently been advocated that residual deep learning architectures of neural networks can roughly be interpreted as dynamical systems (e.g. Weinan, 2017; Chang et al., 2018). Each layer of the network contributes marginally to the output, so that there exists an asymptotic continuum limit representation of the neural network. Furthermore, as mentioned in the introduction, Wang and Lin (1998) and Fablet et al. (2018) have shown that the architecture of the network can follow that of an integration scheme.

By contrast, we have started here from a pure dynamical system standpoint and proposed to use data assimilation techniques. In order to explore complex model resolvents, applied to each interval $[t_k, t_{k+1}]$ between observations, we need the composition of N_c integration steps. In particular, this allows the resolvent to exhibit more realistic long-range correlations. Even when using a reasonably small stencil, long-range correlations will arise as a result of the integration steps. Nonetheless, the stencil might not be too small so as to model discretised higher-order differential operators. As noted by Abarbanel et al. (2018), each application of $\mathbf{f}_\mathbf{A}$ could be seen as a layer of the neural network. Moreover, within such a layer, there are sublayers corresponding to the steps of the integration scheme. The larger N_c is, the deeper this network is, and the richer the class of resolvents is to optimise on.

Following this analogy, the analysis step where $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K})$ is optimised can be called the training phase. Backpropagation in the network, as coined in machine learning (Goodfellow et al., 2016), corresponds to the computation of the gradient of the network with respect to $\mathbf{A}$ and of the model adjoint derived in Sect. 2. This is a shortcut for the use of machine learning software libraries such as TensorFlow or PyTorch (see Appendix C for a brief discussion).

Because of our complete control of the backpropagation, we hope for a gain in efficiency. However, our method does not have the flexibility of deep learning through established tools. For instance, addition of extra parameters, adaptive batch normalisation, and dropouts are not granted in our approach without further considerations.

Convolutional layers play the role of localisation in neural architecture. In our approach this role is played by the locality assumption and its stencil prescription. Recall that a tight stencil does not prevent longer-range correlations that are built up through the integration scheme and their composition. This is similar to stacking several convolutional layers to account for multiple scales from the reference model which the neural network is meant to learn from.

Finally, we note that, as opposed to most practical deep learning strategies with a huge amount of weights to estimate, we have reduced the number of control variables (i.e. $\mathbf{A}$) as much as possible.

4 Numerical illustrations

4.1 Model setup and forecast skill

In this section, we shall consider four low-order chaotic models defined on a physical one-dimensional space, except for L63, which is 0-dimensional. They will serve as reference models.

1. The L63 model as defined by the ODEs:

$$\frac{\mathrm{d}x_0}{\mathrm{d}t} = \sigma(x_1 - x_0), \tag{22a}$$

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = \rho x_0 - x_1 - x_0 x_2, \tag{22b}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = \rho x_0 x_1 - \beta x_2, \tag{22c}$$

   with the canonical values $(\sigma, \rho, \beta) = (10, 28, 8/3)$. Its Lyapunov time[1] is about 1.10. Besides its intrinsic value, this model is introduced for benchmarking against Fablet et al. (2018). It is integrated using an RK4 scheme with $\delta t_r = 0.01$ as the integration time step.

2. The L96 model as defined by ODEs defined over a periodic domain of variables indexed by $n = 0, \ldots, N_x - 1$ where $N_x = 40$:

$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = (x_{n+1} - x_{n-2})x_{n-1} - x_n + F, \tag{23}$$

   where $x_{N_x} = x_0$, $x_{-1} = x_{N_x-1}$, $x_{-2} = x_{N_x-2}$, and $F = 8$. This model is an idealised representation of a one-dimensional latitude band of the Earth's atmosphere. Its Lyapunov time is 0.60. It is integrated using the RK4 scheme and with $\delta t_r = 0.05$.

3. The KS model, as defined by the PDE:

$$\frac{\partial x}{\partial t} = -x\frac{\partial x}{\partial \alpha} - \frac{\partial^2 x}{\partial \alpha^2} - \frac{\partial^4 x}{\partial \alpha^4}, \tag{24}$$

[1] The Lyapunov time is defined as the inverse of the first Lyapunov exponent, i.e. the typical time over which the error grows by a factor e.

over the periodic domain $\alpha \in [0, 32\pi]$ on which we apply a spectral decomposition with $N_x = 128$ modes. The Lyapunov time of our KS model is 10.2 time units. This model is of interest because, even though it has dynamical properties comparable to that of L96, it is much steeper, so that much more stringent numerical integration schemes are required to efficiently integrate it. It is defined by a PDE, not an ODE system. It is integrated using the EDTRK4 scheme (Kassam and Trefethen, 2005) and $\delta t_\mathrm{r} = 0.05$.

4. The two-scale Lorenz model (L05III, Lorenz, 2005) is given by the two-scale ODEs:

$$\frac{\mathrm{d}x_n}{\mathrm{d}t} = \psi_n^+(\boldsymbol{x}) + F - h\frac{c}{b}\sum_{m=0}^{9} u_{m+10n}, \tag{25a}$$

$$\frac{\mathrm{d}u_m}{\mathrm{d}t} = \frac{c}{b}\psi_m^-(b\mathbf{u}) + h\frac{c}{b}x_{m/10}, \tag{25b}$$

$$\psi_n^\pm(\boldsymbol{x}) = x_{n\mp1}(x_{\pm1} - x_{n\mp2}), \tag{25c}$$

for $n = 0, \ldots, N_x - 1$ with $N_x = 36$ and $m = 0, \ldots, N_u - 1$ with $N_u = 360$. The indices apply periodically over their domain; $m/10$ stands for the integer division of m by 10. We use the original values for the parameters: $c = 10$ for the timescale ratio, $b = 10$ for the space-scale ratio, $h = 1$ for the coupling, and $F = 10$ for the forcing. When uncoupled ($h = 0$), the Lyapunov time of the slow variables $\boldsymbol{x}$ sector of the model Eq. (25) is 0.72, which will be the key timescale when focusing on the slow variables (see e.g. Carlu et al., 2019).

This model is of interest because the variable $\mathbf{u}$ is meant to represent unresolved scales and hence model error when only considering the slow variables $\boldsymbol{x}$. For this reason, it has been used in data assimilation papers focusing on estimating model error (e.g. Mitchell and Carrassi, 2015; Pulido et al., 2018). It is integrated with an RK4 scheme and $\delta t_\mathrm{r} = 0.005$ since it is steeper than the L96 model.

The numerical experiments consist of three main steps. First, the truth is generated, i.e. a trajectory of the reference model is computed. The reference model equations are supposed to be unknown, but the trajectory is observed through Eq. (13) to generate the observation vector sequence $\boldsymbol{y}_{0:K}$.

Next, estimators of the ODE model and state trajectory $\boldsymbol{x}_{0:K}$ are learned by minimising the cost function $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K})$. We choose to minimise it using an implementation of the quasi-Newton BFGS algorithm (Byrd et al., 1995), which critically relies on the gradients obtained in Sect. 2. The default choices for the initial ODE model are $\mathbf{A} = \mathbf{0}$ and $\boldsymbol{x}_{0:K}$ defined as the space-wise linear interpolation of $\boldsymbol{y}_{0:K}$. Note that the minimisation could converge to a local minimum, which may or may not yield satisfactory estimates.

Finally, we can make forecasts using the tentative optimal ODE model $\mathbf{A}^\star$ obtained from the minimisation. With a view to comparing it to the reference model used to generate the data, we will consider a set of forecasts with (approximately) independent initial conditions. Both the reference model and the surrogate one will be forecasted from these initial conditions. The departure from their trajectories, as measured by a root mean square error (RMSE) over the observed variables, will be computed for several forecast lead times. The RMSE is then averaged over all the initial conditions. We will also display the state trajectories of the reference and surrogate models starting from one of the initial conditions.

The integration time step of the truth (reference model) is δt_r over the time window $[t_0, t_K]$. This parameter only matters for the reference model integration since only the training time steps $t_{k+1} - t_k$ and the output of the model $\boldsymbol{y}_{0:K}$ (which may include knowledge of the observation operator) are known to the observer.

The integration time step of the surrogate model within the training time window $[t_0, t_K]$ is δt_a. It is assumed to be an integer divisor of the training time step $\Delta t = t_{k+1} - t_k$, supposed to be constant, i.e. $\Delta t/\delta t_\mathrm{a}$ is a constant integer and the number of compositions N_c, and that is why the index k on N_c^k has been dropped. The integration time step of the surrogate model within the forecast time window $[T, T + T_\mathrm{f}]$ is denoted δt_f. Note that δt_a and δt_f can be distinct and that they are critical to the stability of the training and the forecast step, respectively.

The three steps of the numerical experiments are depicted in Fig. 2. Except when explicitly mentioned, the prior $p(\mathbf{A})$ is disregarded, which means that no explicit regularisation on $\mathbf{A}$ is introduced.

4.2 Inferring the dynamics from dense and noiseless observations: perfectly identifiable models

In the first couple of experiments, we consider a densely observed[2] reference model with noiseless observations. In this case, $\mathbf{H}_k$ is the identity operator, i.e. each grid point value is observed, and $\boldsymbol{r}_k \equiv \mathbf{0}$ so that a uniform rescaling of the $\mathbf{Q}_k$, all chosen to be $\mathbf{I}_x$, is irrelevant, assuming $p(\boldsymbol{y}_0, \mathbf{A}) = p(\boldsymbol{y}_0|\mathbf{A})p(\mathbf{A})$ can be neglected, which is hypothesised here and is generally true for large K. Moreover, we use the same numerical scheme with the same integration time step to generate the reference model trajectory as the one used by the surrogate model. In principle, we should be able to retrieve the reference model, since the reference is identifiable, meaning that it belongs to the set of all possible surrogate models.

Let us first experiment with the L63 model, using an RK4 integration scheme, with $\Delta t = 0.01$ and $K = 10^4$ (this corresponds to about 91 Lyapunov times). We have $N_x = 3$ and $N_p = 10$. We choose $\delta t_\mathrm{a} = \delta t_\mathrm{f} = 0.01$. A convergence to the highest possible precision is achieved after about 120 iterations. The cost function value reaches 0 to machine pre-

[2] We choose the qualifier *densely observed* instead of *fully observed* because there is no way to tell from the observations alone whether the reference model is fully observed.

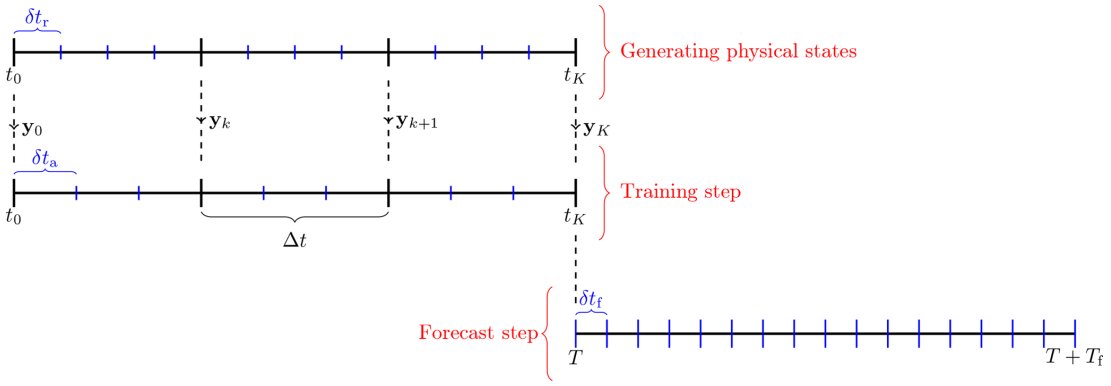

Figure 2. Schematic of the three steps of the experiments, with the associated time steps (see main text). The beginning of the forecast window may or may not coincide with the end of the training window. The lengths of the segments δt_r, δt_a, and δt_f are arbitrary in this schematic.

cision at $\mathbf{A}^\star$. The estimated $\mathbf{A}$ is given by $\mathbf{A}_\mathrm{a} = \mathbf{A}^\star/\delta t_\mathrm{a}$, because, as mentioned above, the optimised $\mathbf{A}$ matrix absorbs the time step. The accuracy of $\mathbf{A}_\mathrm{a}$ is measured by the uniform norm $\|\mathbf{A}_\mathrm{a} - \mathbf{A}_\mathrm{r}\|_\infty$, i.e. the absolute values of the entries of the difference $\mathbf{A}_\mathrm{a} - \mathbf{A}_\mathrm{r}$, where $\mathbf{A}_\mathrm{r}$ is the matrix of the flow rate of L63 (including the zero coefficients). We obtain $\|\mathbf{A}_\mathrm{a} - \mathbf{A}_\mathrm{r}\|_\infty = 8.46 \times 10^{-18}$. To compute the RMSE as a function of the forecast lead time, we average over $N_\mathrm{e} = 10^3$ runs (each one starting from a difference initial condition). The RMSE (not shown) starts significantly, diverging from 0 after 16 Lyapunov time units, and reaches a saturation for a lead time of 23 Lyapunov times.

A similar experiment is carried out with the L96 model, using an RK4 integration scheme, with $\Delta t = 0.05$ and $K = 50$ (this corresponds to about 4.2 Lyapunov times). We choose here to implement the locality and homogeneity assumptions (see Sect. 2.2.1 and 2.2.2). The stencil has a width of 5 (i.e. the local grid points with two points on its left and two points on its right). We have $N_x = 40$, $N_p = 161$, and $N_\mathrm{a} = 18$. We choose $\delta t_\mathrm{a} = \delta t_\mathrm{f} = 0.05$. Through the minimisation, the main coefficients of the L96 model (forcing F, advection terms, dissipation) are retrieved with a precision of a least 8.88×10^{-15}.

To compute the RMSE as a function of the forecast lead time, we average over $N_\mathrm{e} = 10^3$ runs. The RMSE starts significantly, diverging from 0 after 12 Lyapunov times, and reaches a saturation for a lead time of 25 Lyapunov times.

4.3 Inferring the dynamics from dense and noiseless observations: non-identifiable models

In this second couple of experiments, we consider again a densely observed reference model with noiseless observations. The reference model trajectory is generated by the L96 model ($N_x = 40$) integrated with the RK4 scheme, with $\Delta t = 0.05$ and $K = 50$.

As opposed to the reference model, in these non-identifiable model experiments, the surrogate model is based on the RK2 scheme, with N_c compositions. We choose to implement the locality and homogeneity assumptions, with a stencil of width 5. We have $N_p = 161$ and $N_\mathrm{a} = 18$. We choose $\delta t_\mathrm{a} = \delta t_\mathrm{f} = \Delta t/N_\mathrm{c}$. In all the cases, the convergence is reached within a few dozens of iterations. The error on the coefficients of $\mathbf{A}$ (i.e. $\|\mathbf{A}_\mathrm{a} - \mathbf{A}_\mathrm{r}\|_\infty$) is about 4×10^{-2} but with the dominant contribution from F. The RMSE as a function of the forecast lead time is computed for $N_\mathrm{c} = 1, 2, 3, 4, 5$ and is shown in Fig. 3. The error is reduced as N_c is increased. But the improvement saturates at about $N_\mathrm{c} = 5$.

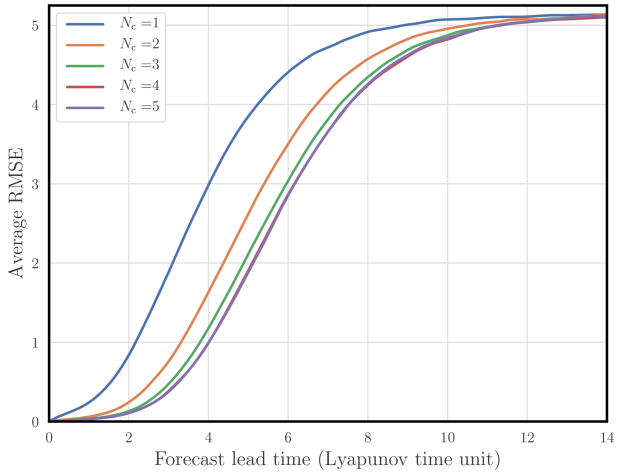

Figure 3. Average RMSE of the surrogate model (L96 with an RK2 structure) compared to the reference model (L96 with an RK4 integration scheme) as a function of the forecast lead time (in Lyapunov time unit) for an increasing number of compositions.

Figure 4 shows the trajectories of the reference and surrogate models starting from the same initial condition, as well as their difference, as a function of the forecast lead time. Their divergence becomes significant after 4 Lyapunov times and saturates after 8 Lyapunov times.

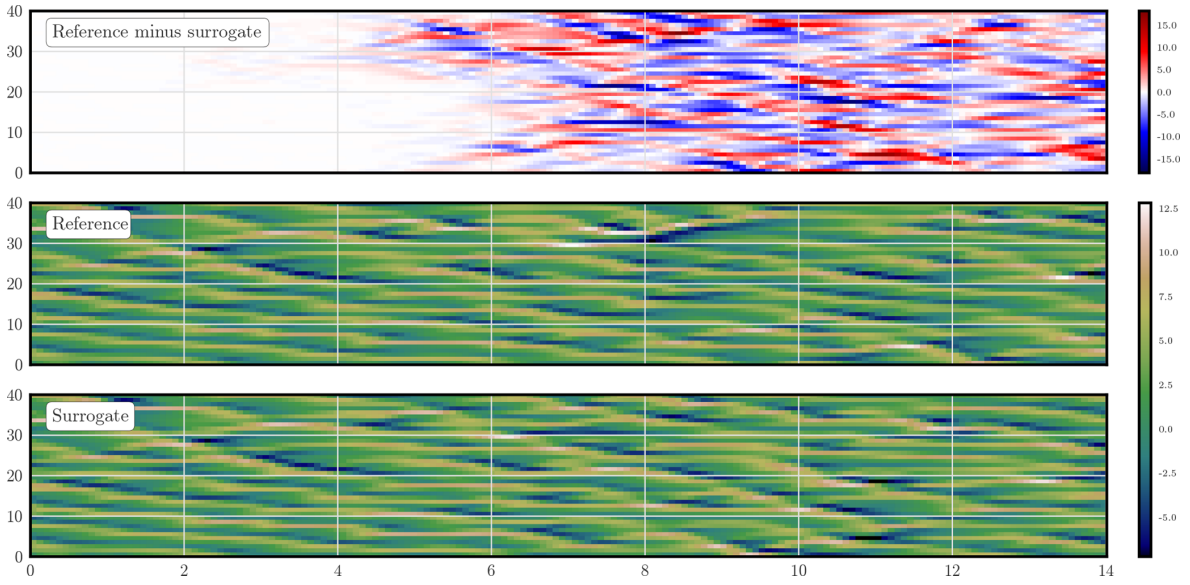

Figure 4. Density plot of the L96 reference and surrogate model trajectories, as well as their difference trajectory, as a function of the forecast lead time (in Lyapunov time unit). The observations are noiseless and dense; the model is not identifiable.

Next, the reference model trajectory is generated by the KS model ($N_x = 128$) integrated with the ETDRK4 scheme, with $\Delta t = 0.05$ and $K = 50$ (this corresponds to about 0.25 Lyapunov time). We choose to implement the locality and homogeneity assumptions, with a stencil of width 9. The surrogate model is based on the RK4 scheme, with $N_c = 2$ compositions. Note that in this experiment, the reference and surrogate models and their integration schemes significantly differ. We have $N_p = 769$ and $N_a = 45$. We choose $\delta t_a = \Delta t / N_c$ and $\delta t_f = 10^{-3}$. The forecast time step δt_f is somehow smaller than δt_a because the KS equations are stiff and so will the surrogate model be. This emphasises once again that we have learned about the intrinsic flow rate of the reference model and not a resolvent thereof. Alternatively, we could use a more robust integration scheme than RK4 such as ETDRK4 for the forecast.

Figure 5 shows the trajectories of the reference and surrogate models starting from the same initial condition, as well as their difference, as a function of the forecast lead time, for a stencil of 9. Their divergence becomes significant after 4 Lyapunov times and saturates after 8 Lyapunov times.

To check whether the PDE of the KS model could be retrieved in spite of the differences in the method of integrations and representations, we have computed a Taylor expansion of all monomials in the surrogate ODE flow rate up to order 4 so as to obtain an approximately PDE equivalent. The coefficients of this PDE (up to order 4 in the expansion) are displayed in Fig. 6 and compared to the coefficients of the reference model's PDE. The match is good and the terms $-x\partial_\alpha x$, $-\partial_\alpha^2 x$, and $-\partial_\alpha^4 x$ are correctly identified as the dominant ones. Nonetheless, there are three non-negligible coefficients for higher-order terms that either might have been generated by the Taylor expansion or may originate from a degeneracy among the higher-order operators, or may simply be identified with a shortcoming of our specific ODE representation.

4.4 Inferring the dynamics from partial and noisy observations

We come back to the L96 model, which is densely observed but with noisy observations that are generated using an independently identically distributed normal noise. The surrogate model is based on an RK4 scheme $N_c = 1$ and a stencil of length 5, which makes the reference model identifiable. In this case, the outcome theoretically depends on the choice for $\boldsymbol{r}_k$ and $\mathbf{Q}_k$, given that Eq. (17) is now used instead of Eq. (18). For the sake of simplicity, we have chosen them to be of the scalar forms $\boldsymbol{r}_k \equiv \sigma_y^2 \mathbf{I}_y$ and $\mathbf{Q}_k \equiv q\mathbf{I}_x$. In these synthetic experiments, σ_y is supposed to be known, while q is not. We only have a qualitative view of the potential mismatch between the reference and the surrogate model. Moreover, a Gaussian additive noise might not even be the best statistical model for such error. Nonetheless holding to the above Gaussian assumptions for model error, the optimal value of q could be determined using an empirical Bayes approach based on, for instance, the expectation–maximisation technique in order to determine the maximum a posteriori of the conditional density of q (see e.g. Dreano et al., 2017; Pulido et al., 2018). The use of advanced methods of that sort to estimate model error will be considered in future works, while, in the following, we have chosen values of q that yield near-to-optimal skill scores (typically $q \in [10^{-4}, 1]\sigma_y^2$).

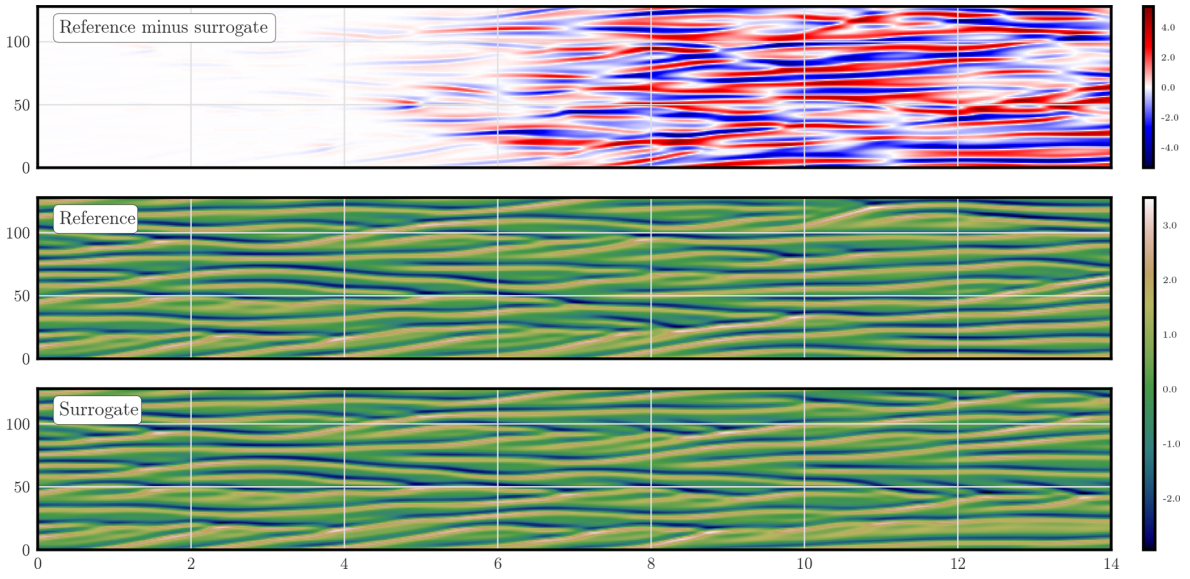

Figure 5. Density plot of the KS reference and surrogate model trajectories, as well as their difference trajectory, as a function of the forecast lead time (in Lyapunov time unit). The observations are noiseless and dense; the model is not identifiable.

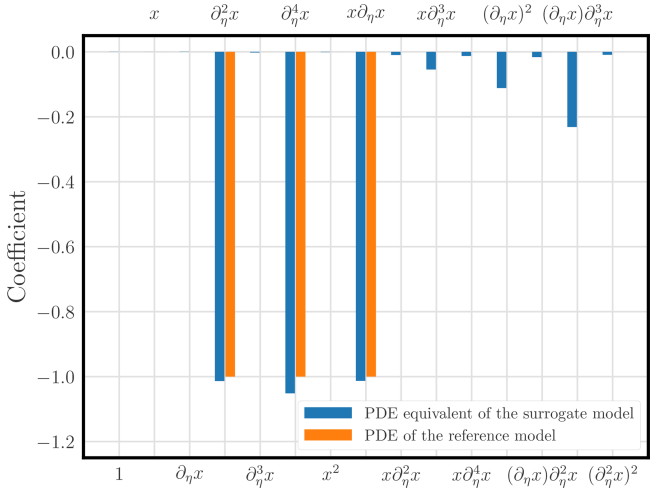

Figure 6. Coefficients of the surrogate PDE model (blue) resulting from the expansion of the surrogate ODEs and compared to the reference PDE's coefficients (orange).

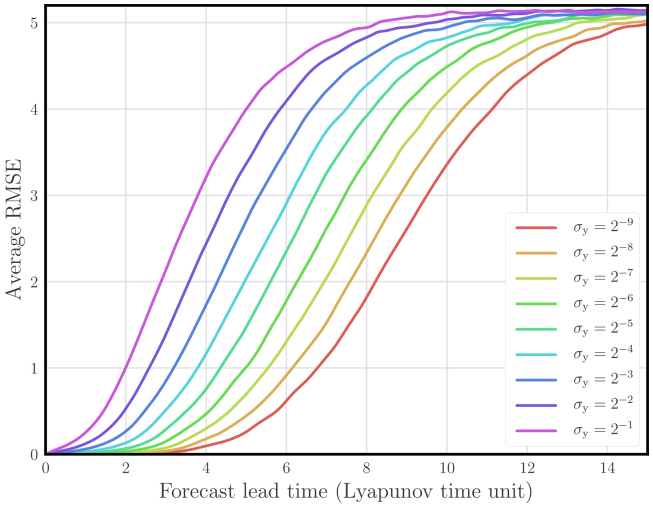

Figure 7. Average RMSE of the surrogate model (L96 with an RK4 structure) compared to the reference model (L96 with an RK4 integration scheme) as a function of the forecast lead time (in Lyapunov time unit) for a range of observation error standard deviations σ_y.

Moreover, note that we have chosen the relatively small $K = 50$. While we expect increasing K to be beneficial, especially with noisy observations, it would force us to address issues related to weak constraint 4D-Var optimisation for long time windows, a topic which is also beyond the scope of this paper. Preliminary results on this aspect are nonetheless discussed later in this section.

Figure 7 shows the forecast skill of the surrogate model as a function of the forecast lead time and for increasing noise in the observations.

Even though, in this configuration, the model is identifiable, the reference value $\mathbf{A}_0$ for $\mathbf{A}$ may not correspond to a minimum of the cost function. The cost function might have several local minima. As a consequence, there is no guarantee, starting from a non-trivial initial value for $\mathbf{A}$, that the model will be identified. Indeed, as seen in Fig. 7, the forecast skill degrades significantly as the observation error standard deviation is increased.

This is confirmed by Fig. 8, where the precisions in identifying the model, measured by either the spectral norm

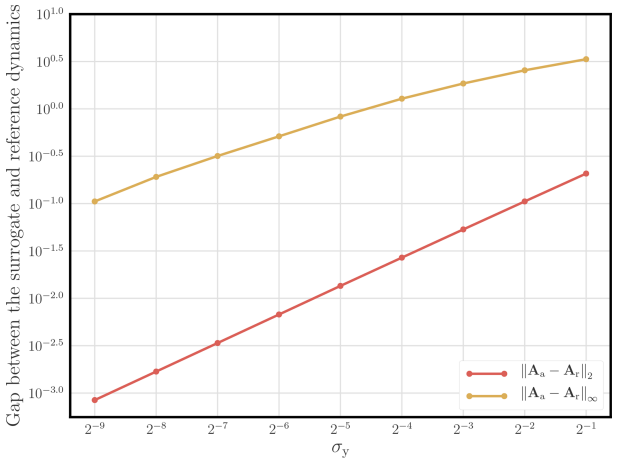

Figure 8. Gap between the surrogate (L96 with an RK4 structure) and the (identifiable) reference dynamics (L96 with an RK4 integration scheme) as a function of the observation error standard deviation σ_y. Note the use of logarithmic scales.

$\|\mathbf{A}_0 - \mathbf{A}\|_2$ or the uniform norm $\|\mathbf{A}_0 - \mathbf{A}\|_\infty$, are plotted as functions of the observation error standard deviation.

Using the same setup, we have also reduced the number of observations. The observations of grid point values are regularly spaced and shifted by one grid cell at each observation time step. The initial $\mathbf{A}$ in the optimisation remains $\mathbf{0}$, while the initial state $\boldsymbol{x}_{0:K}$ is taken as a cubic spline interpolation of the observations over the whole surrogate model grid.

If the observations are noiseless, the reference model is easily retrieved to a high precision down to a density of 1 site over 4. If the observations are noisy, the performance slowly degrades when the density is decreased down to about 1 site over 4, below which the minimisation, trapped in a deceiving local minimum, yields an improper surrogate model.

We would like to point out that in the case of noiseless observations, the performance depends little on the length of the training window, beyond a relatively short length, typically $K = 50$. However, in the presence of noisy observations, the overall performance improves with longer K, as expected since the information content of the observations linearly increases with the length of the window.

Figure 9 displays the values of the coefficients in $\mathbf{A}$ with respect to the minimisation iteration index for the noiseless and fully observed case. As expected, 4 coefficients converge to the value specified by the exact L96 flow rate, while the 14 other coefficients collapse to 0.

Figure 10 shows the same but in the significantly noisy case where $\sigma_y = 1$ and with a significantly longer window $K = 5000$ (about 417 Lyapunov times).

4.5 Inferring reduced dynamics of a multiscale model

In this experiment, we consider the L05III model. With the locality and the homogeneity assumptions, the scalability is typically linear with the size of the system, and we actually consider the 10-fold model where $N_x = 360$ and $N_u = 3600$ to demonstrate that no issues were encountered when scaling up the method. The large-scale variable $\boldsymbol{x}$ of the reference model is noiselessly and fully observed over a short training window ($K = 50$, which corresponds to about 0.35 Lyapunov time), i.e. all slow variable values are observed, whereas the short-scale variable $\mathbf{u}$ is not observed. The surrogate model is based on the RK4 scheme and $N_c = 2$ compositions. We choose to implement the locality and homogeneity assumptions, with a stencil of width 5. We have $N_p = 161$ and $N_a = 18$. We choose $\delta t_a = \delta t_f = \Delta t / N_c$.

Figure 11 shows the trajectories of the reference and surrogate models starting from the same initial condition, as well as their difference, as a function of time.

The emergence of error, i.e. the divergence from the reference, appears as long darker stripes on the density plot of the difference (close to zero difference values appear as white or light colour). We argue that these stripes result from the emergence of sub-scale perturbations that are not properly represented by the surrogate model. Reciprocally there are long-lasting stripes of low error not yet impacted by sub-scale perturbations. As expected, and similarly to the L96 model, the perturbations are transported eastward, as shown by the upward tilt of the stripes in Fig. 11. Clearly, in this case, a flow rate of the form Eq. (2) could be insufficient. Adding a stochastic parametrisation with parameters additionally inferred might offer a solution, as in Pulido et al. (2018). Because of this mixed performance, the RMSE slowly degrades (compared to the other experiments reported so far) with the increase in the forecast lead time (not shown).

5 Conclusions

We have proposed to infer the dynamics of a reference model from its observation using Bayesian data assimilation, which is a new and original scope for data assimilation. Over a given training time window, the control variables are the state trajectory and the coefficients of an ODE representation for the surrogate model. We have chosen the surrogate model to be the composition of an explicit integration scheme (Runge–Kutta typically) applied to this ODE representation. Time invariance, space homogeneity, and locality of the dynamics can be enforced, making the method suitable for high-dimensional systems. The cost function of the data assimilation problem is minimised using the adjoint of the surrogate resolvent which is explicitly derived. Analogies between the surrogate resolvent and a deep neural network have been discussed as well as the impact of stability issues of the reference and surrogate dynamics.

The method has been applied to densely noiseless observed systems with identifiable reference models yielding a perfect reconstruction close to machine precision (L63 and L96 models). It has also been applied to densely or par-

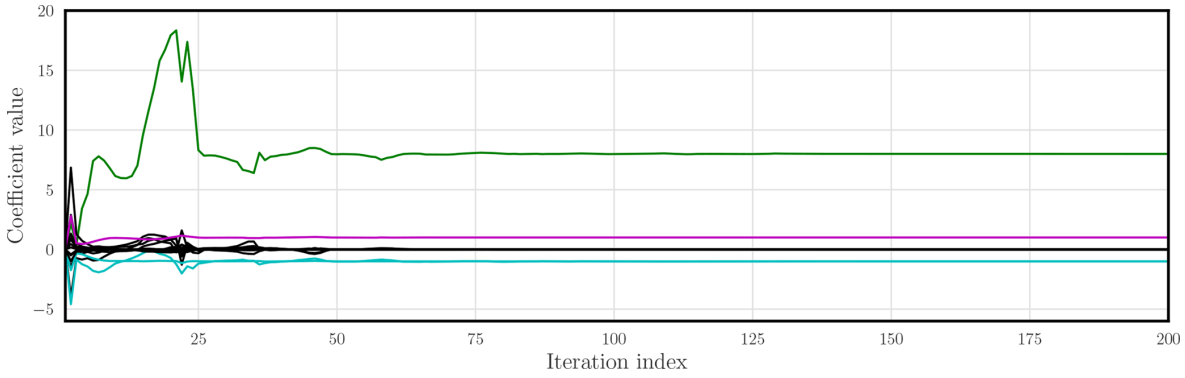

Figure 9. L96 is the reference model, which is fully observed without noise: plot of the $N_a = 18$ coefficients of the surrogate model as a function of the minimisation iteration number. The coefficient of the forcing (F) is in green, the coefficients of the convective terms are in cyan, and the dampening coefficient is in magenta.

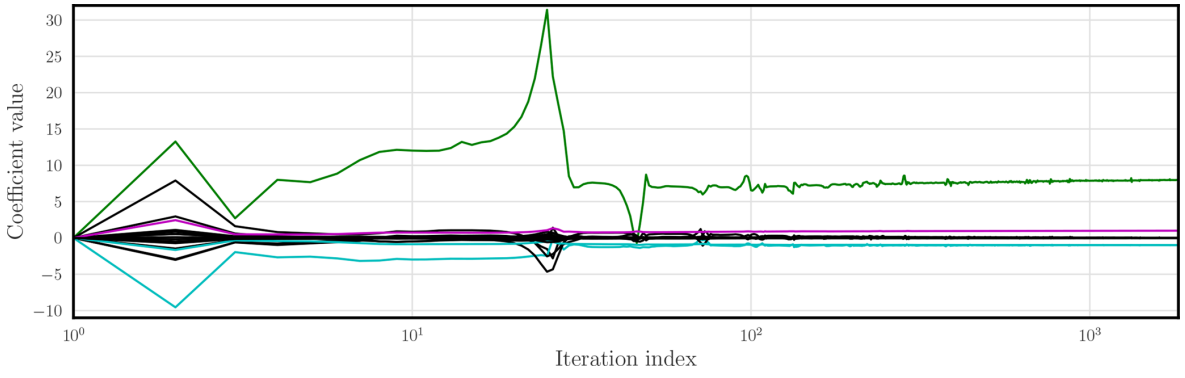

Figure 10. L96 is the reference model, which is fully observed with observation error standard deviation $\sigma_y = 1$: plot of the $N_a = 18$ coefficients of the surrogate model as a function of the minimisation iteration number. Note that the index axis is in logarithmic scale. The coefficient of the forcing (F) is in green, the coefficients of the convective terms are in cyan, and the dampening coefficient is in magenta.

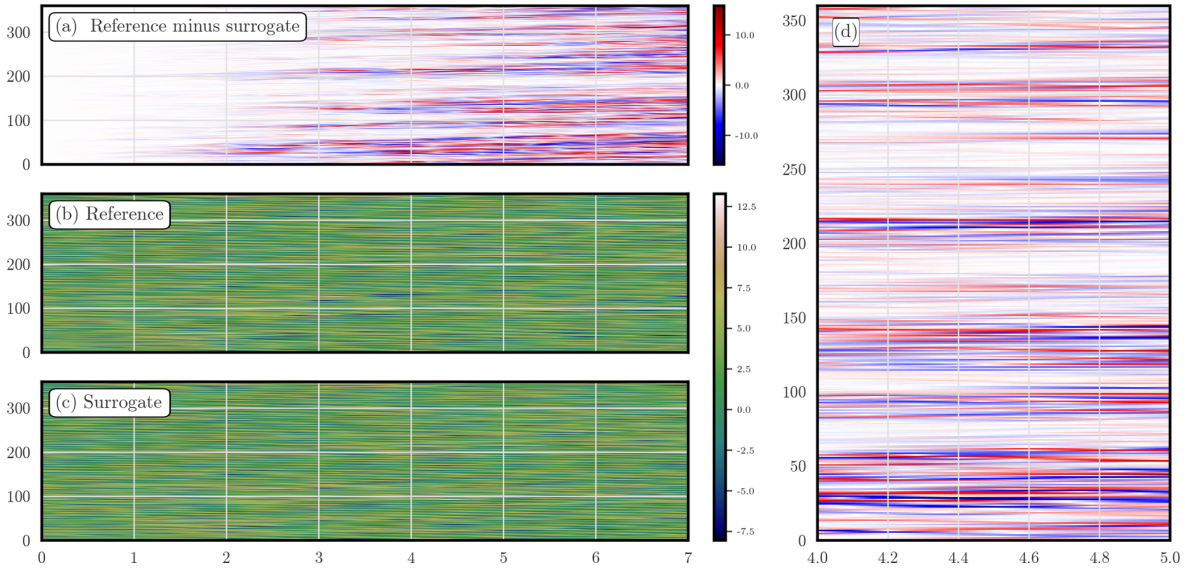

Figure 11. Density plot of the L05III reference and surrogate model trajectories, as well as their difference trajectory, as a function of the forecast lead time (in Lyapunov time unit). Panel **(d)** shows a zoom of the difference between times 4 and 5.

tially observed, identifiable or non-identifiable models with or without noise in the observations (L96 and KS models). For moderate noise and sufficiently dense observation, the method is successful in the sense that the forecast is accurate beyond several Lyapunov times. The method has also been used as a way to infer a reduced model for a multiscale observed system (L05-III model). The reduced model was successful in emulating slow dynamics of the reference model, but could not properly account for the impact of the fast unresolved scale dynamics on the slow ones. A subgrid parametrisation would be required or would have to be inferred.

Two potential obstacles have been left aside on purpose but should later be addressed. First, the model error statistics have not been estimated. This could be achieved using for instance an empirical Bayesian analysis built on an ensemble-based stochastic expectation maximisation technique. This is an especially interesting problem since the potential discrepancy between the reference and the surrogate dynamics is in general non-trivial. Second, we have used relatively short training time windows. Numerical efficient training on longer windows will likely require use of advanced weak constraint variational optimisation techniques.

In this paper, only autonomous dynamics have been considered. We could at least partially extend the method to non-autonomous systems by keeping a static part for the pure dynamics and consider time-dependent forcing fields. We have not numerically explored non-homogeneous dynamics, but we have shown how to learn from them using non-homogeneous local representations.

A promising yet challenging path would be to consider implicit or semi-implicit schemes following for instance the idea in Chen et al. (2018). This idea is known in geophysical data assimilation as the continuous adjoint (see e.g. Bocquet, 2012). This would considerably strengthen the stability of the training and forecast steps at the cost of more intricate mathematical developments.

If observations keep coming after the training time window, then one can perform data assimilation using the ODE surrogate model of the reference model. This data assimilation scheme could only focus on state estimation or it could continue to update the ODE surrogate model for the forecast.

Data availability. No datasets were used in this article.

Appendix A: Parametrisation of $\boldsymbol{\phi}_{\mathbf{A}}$ for local representations defined over a circle

In this Appendix, we show how to parametrise $\boldsymbol{\phi}_{\mathbf{A}}$ assuming locality of the representation, in the case where it is defined over a periodic one-dimensional domain, i.e. a circle. It is of the generic form

$$[\mathbf{A}\cdot\boldsymbol{r}]_n = \sum_{p=0}^{N_p-1} A_{n,\pi(n,p)} r_p, \tag{A1}$$

where $\pi(n,p)$ is an integer such that $0 \leq \pi(n,p) < N_\mathrm{a}$. We can treat the bias, linear, and bilinear monomials separately into sectors, 0, 1, and 2, respectively. Let $0 \leq a_i \leq N_\mathrm{a}-1$ be the indices which span the columns of $\mathbf{A}$ for each of the three sectors i and $0 \leq p_i \leq N_p - 1$ the indices which span the entries of $\boldsymbol{r}$ for each of the three sectors i. Then, Eq. (A1) can be more explicitly written as

$$\begin{aligned}[\mathbf{A}\cdot\boldsymbol{r}]_n =& A_{n,a_0(0)} r_{p_0(0)} + \sum_{l=0}^{2L} A_{n,a_1(n,l)} r_{p_1(n,l)} \\ &+ \sum_{l=0}^{L}\sum_{m=0}^{2L-l} A_{n,a_2(n,l,m)} r_{p_2(n,l,m)},\end{aligned} \tag{A2}$$

where the dummy index l for the linear terms browses the stencil, and the dummy indices l,m for the bilinear monomials browse the stencil, in the same way as we did in Sect. 2.2.1 to enumerate them. By enumeration, we find the following.

- For the bias sector, we have $p_0(0) = 0$ and $a_0(0) = 0$.
- For the linear sector, we have $a_1(n,l) = 1+l$ and $p_1(n,l) = 1+[n+l-L]$, where $[n]$ means the index in $[0, N_x-1]$ congruent to n modulo N_x, in order to respect the periodicity of the domain.
- Finally, for the bilinear sector, we have $a_2(n,l,m) = 1+2L+1+\frac{1}{2}l(4L-l+1)+m$ and $p_2(n,l,m) = 1+N_x+[n-L+m](1+L)+l$.

We observe that these indices $a_0(0), a_1(n,l)$, and $a_2(n,l,m)$ do not depend on the site index n. They only indicate a relative position with respect to n. Hence, if homogeneity is additionally assumed, A_{n,a_0}, A_{n,a_1}, and A_{n,a_2} do not depend on n anymore and $\mathbf{A}$ becomes a vector.

Appendix B: Computations of the gradients with respect to A and $\boldsymbol{x}_{\mathbf{0:}K}$

B1 Differentiation of the RK schemes

It will be useful in the following to consider the variation of each $\mathbf{k}_i$, defined by Eq. (12b), with respect to either $\mathbf{A}$ or $\boldsymbol{x}$:

$$\delta\mathbf{k}_i = \delta\boldsymbol{\phi}_{\mathbf{A},i} + h\sum_{j=0}^{i-1}\alpha_{i,j}\left(\nabla_{\boldsymbol{x}_i}\boldsymbol{\phi}_{\mathbf{A},i}\right)\delta\mathbf{k}_j, \tag{B1}$$

where $\boldsymbol{\phi}_{\mathbf{A},i}$ is $\boldsymbol{\phi}_{\mathbf{A}}$ evaluated at $\boldsymbol{x}_i^{\mathrm{RK}} \equiv \boldsymbol{x} + h\sum_{j=0}^{i-1}\alpha_{i,j}\mathbf{k}_j$ and $\nabla_{\boldsymbol{x}_i}\boldsymbol{\phi}_{\mathbf{A},i}$ is the tangent linear operator with respect to $\boldsymbol{x}$ of $\boldsymbol{\phi}_{\mathbf{A}}$ evaluated at $\boldsymbol{x}_i^{\mathrm{RK}}$. Equation (B1) can be written compactly in the form

$$\mathbf{G}\delta\boldsymbol{\kappa} = \delta\boldsymbol{\varphi}, \tag{B2}$$

where $\mathbf{G}$ is the matrix of size $(N_{\mathrm{RK}}N_x)\times(N_{\mathrm{RK}}N_x)$ defined by its $N_x\times N_x$ blocks $[\mathbf{G}]_{i,j} = \mathbf{I}_x - h\alpha_{i,j}\nabla_{\boldsymbol{x}_i}\boldsymbol{\phi}_{\mathbf{A},i}$, $\boldsymbol{\kappa}$ is the vector of size $N_{\mathrm{RK}}N_x$ which results from the stacking of the $\mathbf{k}_i \in \mathbb{R}^{N_x}$ for $0 \leq i < N_{\mathrm{RK}}$, $\boldsymbol{\varphi}$ is the vector of size $N_{\mathrm{RK}}N_x$ which results from the stacking of the $\boldsymbol{\phi}_{\mathbf{A},i}$ for $0 \leq i < N_{\mathrm{RK}}$, and $\mathbf{I}_x \in \mathbb{R}^{N_x}$ is the identity matrix. The important point is that $\mathbf{G}$ is a lower triangular matrix and describes an iterative construction of the $\mathbf{k}_i$. Moreover, the diagonal entries of $\mathbf{G}$ are 1 by construction so that $\mathbf{G}$ is invertible and

$$\delta\boldsymbol{\kappa} = \mathbf{G}^{-1}\delta\boldsymbol{\varphi}. \tag{B3}$$

This will be used to compute the variations of $\mathbf{f}_{\mathbf{A}}(\boldsymbol{x})$ via Eq. (12a).

B2 Integration step

We first consider the situation when the observation interval corresponds to one integration time step of the surrogate model, i.e. $N_\mathrm{c}^k = 1$: $\boldsymbol{x}' = \mathbf{F}_{\mathbf{A}}(\boldsymbol{x}) = \mathbf{f}_{\mathbf{A}}(\boldsymbol{x})$ with $\boldsymbol{x} \equiv \boldsymbol{x}_k$. As a result, the time index k can be omitted here. We will later consider the composition of several integration schemes ($N_\mathrm{c} \geq 2$). Equation (12a) is written again but as

$$\mathbf{f}_{\mathbf{A}}(\boldsymbol{x}) = \boldsymbol{x} + \mathbf{b}\boldsymbol{\kappa}, \tag{B4}$$

where $\mathbf{b} = \boldsymbol{\beta}\otimes\mathbf{I}_x$ is the matrix of a size $N_x\times(N_{\mathrm{RK}}N_x)$ tensor product of the vector $\boldsymbol{\beta}$ defined by $\boldsymbol{\beta}^{\mathrm{T}} = (\beta_0,\ldots,\beta_{N_{\mathrm{RK}}-1})$, i.e. the coefficients of the RK scheme as defined in Eq. (12a), with the state space identity matrix, and where $\boldsymbol{\kappa}$ is the vector of size $N_{\mathrm{RK}}N_x$ defined after Eq. (B2). Looking first at the gradient with respect to the state variable, and using Eq. (B3), we have

$$\nabla_{\boldsymbol{x}}\mathbf{f}_{\mathbf{A}} = \nabla_{\boldsymbol{x}}\boldsymbol{x} + \mathbf{b}\nabla_{\boldsymbol{x}}\boldsymbol{\kappa} = \mathbf{I}_x + \mathbf{b}\mathbf{G}^{-1}\nabla_{\boldsymbol{x}}\boldsymbol{\varphi}, \tag{B5}$$

which yields the adjoint operator

$$(\nabla_{\boldsymbol{x}}\mathbf{f}_{\mathbf{A}})^{\mathrm{T}} = \mathbf{I}_x + (\nabla_{\mathrm{x}}\boldsymbol{\varphi})^{\mathrm{T}}\mathbf{G}^{-\mathrm{T}}\mathbf{b}^{\mathrm{T}}. \tag{B6}$$

Let us consider an arbitrary vector $\boldsymbol{d} \in \mathbb{R}^{N_x}$; we have

$$(\nabla_{\boldsymbol{x}}\mathbf{f}_{\mathbf{A}})^{\mathrm{T}}\boldsymbol{d} = \boldsymbol{d} + (\nabla_{x}\boldsymbol{\varphi})^{\mathrm{T}}\mathbf{G}^{-\mathrm{T}}\mathbf{b}^{\mathrm{T}}\boldsymbol{d}. \tag{B7}$$

To avoid computing $\mathbf{G}^{-\mathrm{T}}$ explicitly, let us define the vector $\boldsymbol{z} \in \mathbb{R}^{N_x N_{\mathrm{RK}}}$ such that

$$\mathbf{G}^{\mathrm{T}}\boldsymbol{z} = \mathbf{b}^{\mathrm{T}}\boldsymbol{d}. \tag{B8}$$

Because $\mathbf{G}^{\mathrm{T}}$ is upper triangular with diagonal entries of value 1, $\boldsymbol{z}$ is the solution of a linear system easily solvable iteratively, which stands as an adjoint/dual to the RK iterative

construction. Hence, we finally obtain a formula and algorithm to evaluate

$$(\nabla_{\boldsymbol{x}}\mathbf{f}_{\mathbf{A}})^{\mathrm{T}}\boldsymbol{d} = \boldsymbol{d} + (\nabla_{\boldsymbol{x}}\boldsymbol{\varphi})^{\mathrm{T}}\boldsymbol{z}, \tag{B9}$$

which is key to computing Eq. (19). Indeed, Eq. (19) now reads as

$$\begin{aligned}\nabla_{\boldsymbol{x}_0}\mathcal{J} =& -\left(\nabla_{\boldsymbol{x}_0}\mathbf{H}_0\right)^{\mathrm{T}}\boldsymbol{r}_0^{-1}\boldsymbol{\delta}_0 - \mathbf{Q}_1^{-1}\boldsymbol{\Delta}_1 - \left(\nabla_{\boldsymbol{x}_0}\boldsymbol{\varphi}\right)^{\mathrm{T}}\boldsymbol{z}_0 \\ &- \nabla_{\boldsymbol{x}_0}\ln p(\boldsymbol{x}_0|\mathbf{A}),\end{aligned} \tag{B10a}$$

$$\begin{aligned}\nabla_{\boldsymbol{x}_k}\mathcal{J} =& -\left(\nabla_{\boldsymbol{x}_k}\mathbf{H}_k\right)^{\mathrm{T}}\boldsymbol{r}_k^{-1}\boldsymbol{\delta}_k - \mathbf{Q}_{k+1}^{-1}\boldsymbol{\Delta}_{k+1} - \left(\nabla_{\boldsymbol{x}_k}\boldsymbol{\varphi}\right)^{\mathrm{T}}\boldsymbol{z}_k \\ &+ \mathbf{Q}_k^{-1}\boldsymbol{\Delta}_k, \quad \text{for} \quad 1 \leq k \leq K-1,\end{aligned} \tag{B10b}$$

$$\nabla_{\boldsymbol{x}_K}\mathcal{J} = -\left(\nabla_{\boldsymbol{x}_K}\mathbf{H}_K\right)^{\mathrm{T}}\boldsymbol{r}_K^{-1}\boldsymbol{\delta}_K + \mathbf{Q}_K^{-1}\boldsymbol{\Delta}_K, \tag{B10c}$$

where $\boldsymbol{z}_k$ is the iterative solution of the system $\mathbf{G}_k^{\mathrm{T}}\boldsymbol{z}_k = \mathbf{b}^{\mathrm{T}}\mathbf{Q}_{k+1}^{-1}\boldsymbol{\Delta}_{k+1}$ for $0 \leq k \leq K-1$.

Second, let us look at the gradient of $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K})$ with respect to $\mathbf{A}$. From Eqs. (B4) and (B3), and now considering variations with respect to $\mathbf{A}$, we obtain

$$\nabla_{\mathbf{A}}\mathbf{f}_{\mathbf{A}} = \nabla_{\mathbf{A}}\boldsymbol{x} + \mathbf{b}\nabla_{\mathbf{A}}\boldsymbol{\kappa} = \mathbf{b}\mathbf{G}^{-1}\nabla_{\mathbf{A}}\boldsymbol{\varphi}, \tag{B11}$$

which yields, using $\boldsymbol{z}$ as defined by Eq. (B8),

$$\boldsymbol{d}^{\mathrm{T}}\left(\nabla_{\mathbf{A}}\mathbf{f}_{\mathbf{A}}\right) = \boldsymbol{d}^{\mathrm{T}}\mathbf{b}\mathbf{G}^{-1}\left(\nabla_{\mathbf{A}}\boldsymbol{\varphi}\right) = \boldsymbol{z}^{\mathrm{T}}\left(\nabla_{\mathbf{A}}\boldsymbol{\varphi}\right). \tag{B12}$$

For each $0 \leq i < N_{\mathrm{RK}}$, let us introduce $\boldsymbol{r}_i = \boldsymbol{r}(\mathbf{x}_\mathbf{i}) \in \mathbb{R}^{N_p}$, and let us denote $\boldsymbol{z}_i \in \mathbb{R}^{N_x}$ the subvector of $\boldsymbol{z}$ for the ith block of the Runge–Kutta scheme. Then we have, for $0 \leq n < N_x$ and $0 \leq p < N_p$,

$$\begin{aligned}\left[\boldsymbol{z}^{\mathrm{T}}\left(\nabla_{\mathbf{A}}\boldsymbol{\varphi}\right)\right]_{n,p} &= \sum_{m=0}^{N_x-1}\sum_{i=0}^{N_{\mathrm{RK}}-1}[\boldsymbol{z}_i]_m\frac{\partial}{\partial A_{n,p}}\sum_{q=0}^{N_p-1}A_{m,q}[\boldsymbol{r}_i]_q \\ &= \sum_{m=0}^{N_x-1}\sum_{i=0}^{N_{\mathrm{RK}}-1}\sum_{q=0}^{N_p-1}[\boldsymbol{z}_i]_m\delta_{m,n}\delta_{p,q}[\boldsymbol{r}_i]_q \\ &= \sum_{i=0}^{N_{\mathrm{RK}}-1}[\boldsymbol{z}_i]_n[\boldsymbol{r}_i]_p = \sum_{i=0}^{N_{\mathrm{RK}}-1}\boldsymbol{z}_i\boldsymbol{r}_i^{\mathrm{T}}.\end{aligned} \tag{B13}$$

This is key to efficiently computing Eq. (20), which now reads as

$$\nabla_{\mathbf{A}}\mathcal{J} = -\sum_{k=1}^{K}\sum_{i=0}^{N_{\mathrm{RK}}-1}\boldsymbol{z}_{k,i}\boldsymbol{r}_{k,i}^{\mathrm{T}} - \nabla_{\mathbf{A}}\ln p(\mathbf{A}), \tag{B14}$$

where $\boldsymbol{z}_k$ is the solution of $\mathbf{G}_k^{\mathrm{T}}\boldsymbol{z}_k = \mathbf{b}^{\mathrm{T}}\mathbf{Q}_k^{-1}\boldsymbol{\delta}_k$. The index k of $\mathbf{G}_k$ indicates that the operators defined in the entries of $\mathbf{G}$ are evaluated at $\boldsymbol{x}_k$.

B3 Composition of integration steps

We now consider a resolvent which is the composition of $N_{\mathrm{c}}^k \geq 2$ integration steps over $[t_k, t_{k+1}]$: $\boldsymbol{x}' = \mathbf{f}_{\mathbf{A}}^{N_{\mathrm{c}}^k}(\boldsymbol{x})$, where $\boldsymbol{x}$ is an alias to $\boldsymbol{x}_k$. Let us first look at the gradient with respect to the state variable. Within the scope of this section, we define $\boldsymbol{x}_0 \equiv \boldsymbol{x}$ and for $1 \leq l \leq N_{\mathrm{c}}^k$: $\boldsymbol{x}_l \equiv \mathbf{f}_{\mathbf{A}}(\boldsymbol{x}_{l-1})$. Hence, $\boldsymbol{x}' = \boldsymbol{x}_{N_{\mathrm{c}}}$. We also define $\nabla_{\boldsymbol{x}_l}\mathbf{f}_{\mathbf{A}}$ as the tangent linear operator of $\mathbf{f}_{\mathbf{A}}$ at $\boldsymbol{x}_l$. By the Leibniz rule, we obtain

$$\left(\nabla_{\boldsymbol{x}}\mathbf{F}_{\mathbf{A}}^k\right)^{\mathrm{T}} = (\nabla_{\boldsymbol{x}_0}\mathbf{f}_{\mathbf{A}})^{\mathrm{T}}(\nabla_{\boldsymbol{x}_1}\mathbf{f}_{\mathbf{A}})^{\mathrm{T}}\cdots(\nabla_{\boldsymbol{x}_{N_{\mathrm{c}}-1}}\mathbf{f}_{\mathbf{A}})^{\mathrm{T}}. \tag{B15}$$

We can now apply Eq. (B9) to each individual integration step and obtain for any $\boldsymbol{d} \in \mathbb{R}^{N_x}$

$$(\nabla_{\boldsymbol{x}_l}\mathbf{f}_{\mathbf{A}})^{\mathrm{T}}\boldsymbol{d} = \boldsymbol{d} + \left(\nabla_{\boldsymbol{x}_l}\boldsymbol{\varphi}\right)^{\mathrm{T}}\boldsymbol{z}_l, \tag{B16}$$

where $\boldsymbol{z}_l$ is the solution of

$$\mathbf{G}_l^{\mathrm{T}}\boldsymbol{z}_l = \mathbf{b}^{\mathrm{T}}\boldsymbol{d}. \tag{B17}$$

Hence, to compute $\left(\nabla_{\boldsymbol{x}}\mathbf{F}_{\mathbf{A}}^k\right)^{\mathrm{T}}\cdot\boldsymbol{d}$, we define $\widetilde{\boldsymbol{x}}_{N_{\mathrm{c}}} = \boldsymbol{d}$ and for $N_{\mathrm{c}}-1 \geq l \geq 0$: $\widetilde{\boldsymbol{x}}_l = (\nabla_{\boldsymbol{x}_l}\mathbf{f}_{\mathbf{A}})^{\mathrm{T}}\widetilde{\boldsymbol{x}}_{l+1}$. This finally reads as

$$\widetilde{\boldsymbol{x}}_l = \widetilde{\boldsymbol{x}}_{l+1} + \left(\nabla_{\boldsymbol{x}_l}\boldsymbol{\varphi}_l\right)^{\mathrm{T}}\boldsymbol{z}_l, \tag{B18}$$

for $N_{\mathrm{c}}-1 \geq l \geq 0$, where $\boldsymbol{z}_l$ is the solution of

$$\mathbf{G}_l^{\mathrm{T}}\boldsymbol{z}_l = \mathbf{b}^{\mathrm{T}}\widetilde{\boldsymbol{x}}_{l+1}. \tag{B19}$$

To compute the key terms in the gradients Eq. (19), $\boldsymbol{d}$ must be chosen to be $\mathbf{Q}_{k+1}^{-1}\boldsymbol{\Delta}_{k+1}$ where $0 \leq k \leq K-1$ and

$$\left(\nabla_{\boldsymbol{x}}\mathbf{F}_{\mathbf{A}}^k\right)^{\mathrm{T}}\mathbf{Q}_{k+1}^{-1}\boldsymbol{\Delta}_{k+1} = \widetilde{\boldsymbol{x}}_0. \tag{B20}$$

Second, we look at the gradient with respect to $\mathbf{A}$. In this case, the application of the Leibniz rule yields

$$\begin{aligned}\nabla_{\mathbf{A}}\mathbf{F}_{\mathbf{A}}^k &= \sum_{l=0}^{N_{\mathrm{c}}-1}(\nabla_{\boldsymbol{x}_{N_{\mathrm{c}}-1}}\mathbf{f}_{\mathbf{A}})\cdots(\nabla_{\boldsymbol{x}_{l+1}}\mathbf{f}_{\mathbf{A}})\nabla_{\mathbf{A}}\mathbf{f}_{\mathbf{A}}(\boldsymbol{x}_l) \\ &= \sum_{l=0}^{N_{\mathrm{c}}-1}\left(\nabla_{\boldsymbol{x}_{l+1}}\mathbf{f}_{\mathbf{A}}^{N_{\mathrm{c}}-l-1}\right)\nabla_{\mathbf{A}}\mathbf{f}_{\mathbf{A}}(\boldsymbol{x}_l),\end{aligned} \tag{B21}$$

where $\nabla_{\boldsymbol{x}_{l+1}}\mathbf{f}_{\mathbf{A}}^{N_{\mathrm{c}}-l-1} = (\nabla_{\boldsymbol{x}_{N_{\mathrm{c}}-1}}\mathbf{f}_{\mathbf{A}})\cdots(\nabla_{\boldsymbol{x}_{l+1}}\mathbf{f}_{\mathbf{A}})$. But $\nabla_{\mathbf{A}}\mathbf{f}_{\mathbf{A}}(\boldsymbol{x}_l)$, which focuses on a single integration step, is given by Eq. (B11),

$$\nabla_{\mathbf{A}}\mathbf{f}_{\mathbf{A}}(\boldsymbol{x}_l) = \mathbf{b}\mathbf{G}_l^{-1}\nabla_{\mathbf{A}}\boldsymbol{\varphi}_l \tag{B22}$$

and from Eq. (B12),

$$\boldsymbol{d}^{\mathrm{T}}\nabla_{\mathbf{A}}\mathbf{f}_{\mathbf{A}}(\boldsymbol{x}_l) = \boldsymbol{z}_l^{\mathrm{T}}\left(\nabla_{\mathbf{A}}\boldsymbol{\varphi}_l\right). \tag{B23}$$

As a result, we obtain

$$\nabla_{\mathbf{A}}\mathcal{J} = -\sum_{l=0}^{N_{\mathrm{c}}-1}\sum_{k=1}^{K}\sum_{i=0}^{N_{\mathrm{RK}}-1}\boldsymbol{z}_{k,l,i}\boldsymbol{r}_{k,l,i}^{\mathrm{T}} - \nabla_{\mathbf{A}}\ln p(\mathbf{A}), \tag{B24}$$

where $\boldsymbol{z}_{k,l}$ is the solution of

$$\mathbf{G}_{k,l}^{\mathrm{T}}\boldsymbol{z}_{k,l} = \mathbf{b}^{\mathrm{T}}\left(\nabla_{\boldsymbol{x}_{l+1}}\mathbf{f}_{\mathbf{A}}^{N_{\mathrm{c}}-l-1}\right)^{\mathrm{T}}\mathbf{Q}_k^{-1}\boldsymbol{\delta}_k. \tag{B25}$$

All of these results, Eqs. (B10), (B14), (B20), and (B24), allow us to efficiently compute the gradients of the cost function $\mathcal{J}(\mathbf{A}, \boldsymbol{x}_{0:K})$ with respect to both $\mathbf{A}$ and $\boldsymbol{x}_{0:K}$. Note, however, that they have been derived under the simplifying assumption that $\boldsymbol{\phi}_{\mathbf{A}}$ is given by Eq. (2) with a traditional matrix multiplication between $\mathbf{A}$ and $\boldsymbol{r}(\boldsymbol{x})$, but not by the compact Eq. (8). When relying on homogeneity and/or locality, the calculation of the gradient with respect to $\mathbf{A}$ follows the principle described above but requires further adaptations, which can be derived using Eq. (A2), with the asset of strongly reducing the computational burden.

Appendix C: Adjoint differentiation with PyTorch and TensorFlow

As an alternative to the explicit computation of the gradients of Eq. (17) and the associated adjoint models, we have used PyTorch and TensorFlow as automatic differentiation tools. Only the cost function code needs to be implemented. We made a few tests of the experiments of Sect. 4 that showed that the fastest code is a C++ implementation using the explicit gradients, followed by an implementation using Python/Numpy/Numba using the explicit gradients, followed by a much slower implementation with TensorFlow (graph execution), followed by an implementation with TensorFlow (eager execution) and finally with PyTorch. Our experiments made use of a multi-core CPU and/or a GPU. This purely qualitative ranking is not surprising since (i) PyTorch and TensorFlow excel in tensor algebra operations which are not massively used in the way we built our cost function with relatively few but significant parameters, and (ii) the time spent in the development of each approach scales with their speed.

Author contributions. MB first developed the theory, implemented, ran, and interpreted the numerical experiments, and wrote the original version of the manuscript. All authors discussed the theory, interpreted the results, and edited the manuscript. The four authors approved the manuscript for publication.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The authors are grateful to two anonymous reviewers and Olivier Talagrand acting as editor for their comments and suggestions. Marc Bocquet is thankful to Said Ouala and Ronan Fablet for enlightening discussions. CEREA and LOCEAN are members of the Institut Pierre-Simon Laplace (IPSL).

Financial support. This research has been supported by the Norwegian Research Council (project REDDA (grant no. 250711)).

Review statement. This paper was edited by Olivier Talagrand and reviewed by two anonymous referees.

References

Abarbanel, H. D. I., Rozdeba, P. J., and Shirman, S.: Machine Learning: Deepest Learning as Statistical Data Assimilation Problems, Neural Comput., 30, 2025–2055, https://doi.org/10.1162/neco_a_01094, 2018.

Amezcua, J., Goodliff, M., and van Leeuwen, P.-J.: A weak-constraint 4DEnsembleVar. Part I: formulation and simple model experiments, Tellus A, 69, 1271564, https://doi.org/10.1080/16000870.2016.1271564, 2017.

Asch, M., Bocquet, M., and Nodet, M.: Data Assimilation: Methods, Algorithms, and Applications, Fundamentals of Algorithms, SIAM, Philadelphia, 2016.

Aster, R. C., Borchers, B., and Thuber, C. H.: Parameter Estimation and Inverse Problems, Elsevier Academic Press, 2nd Edn., 2013.

Bocquet, M.: Parameter field estimation for atmospheric dispersion: Application to the Chernobyl accident using 4D-Var, Q. J. Roy. Meteor. Soc., 138, 664–681, https://doi.org/10.1002/qj.961, 2012.

Brunton, S. L., Proctor, J. L., and Kutz, J. N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems, P. Natl. Acad. Sci. USA, 113, 3932–3937, https://doi.org/10.1073/pnas.1517384113, 2016.

Buizza, R., Miller, M., and Palmer, T. N.: Stochastic representation of model uncertainties in the ECMWF ensemble prediction system, Q. J. Roy. Meteor. Soc., 125, 2887–2908, 1999.

Byrd, R. H., Lu, P., and Nocedal, J.: A Limited Memory Algorithm for Bound Constrained Optimization, SIAM J. Sci. Stat. Comp., 16, 1190–1208, 1995.

Carlu, M., Ginelli, F., Lucarini, V., and Politi, A.: Lyapunov analysis of multiscale dynamics: the slow bundle of the two-scale Lorenz 96 model, Nonlin. Processes Geophys., 26, 73–89, https://doi.org/10.5194/npg-26-73-2019, 2019.

Carrassi, A. and Vannitsem, S.: Accounting for model error in variational data assimilation: A deterministic formulation, Mon. Weather Rev., 138, 3369–3386, https://doi.org/10.1175/2010MWR3192.1, 2010.

Carrassi, A., Bocquet, M., Bertino, L., and Evensen, G.: Data Assimilation in the Geosciences: An overview on methods, issues, and perspectives, WIREs Climate Change, 9, e535, https://doi.org/10.1002/wcc.535, 2018.

Chang, B., Meng, L., Haber, E., Tung, F., and Begert, D.: Multi-level residual networks from dynamical systems view, in: Proceedings of ICLR 2018, 2018.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D.: Neural ordinary differential equations, in: Advances in Neural Information Processing Systems, 6571–6583, 2018.

Dreano, D., Tandeo, P., Pulido, M., Ait-El-Fquih, B., Chonavel, T., and Hoteit, I.: Estimating model error covariances in nonlinear state-space models using Kalman smoothing and the expectation-maximisation algorithm, Q. J. Roy. Meteor. Soc., 143, 1877–1885, https://doi.org/10.1002/qj.3048, 2017.

Dueben, P. D. and Bauer, P.: Challenges and design choices for global weather and climate models based on machine learning, Geosci. Model Dev., 11, 3999–4009, https://doi.org/10.5194/gmd-11-3999-2018, 2018.

Fablet, R., Ouala, S., and Herzet, C.: Bilinear residual neural network for the identification and forecasting of dynamical systems, in: EUSIPCO 2018, European Signal Processing Conference, Rome, Italy, 1–5, available at: https://hal.archives-ouvertes.fr/hal-01686766 (last access: 8 July 2019), 2018.

Gautschi, W.: Numerical analysis, Springer Science & Business Media, 2nd Edn., 2012.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep learning, The MIT Press, Cambridge Massachusetts, London, England, 2016.

Grudzien, C., Carrassi, A., and Bocquet, M.: Chaotic dynamics and the role of covariance inflation for reduced rank Kalman filters with model error, Nonlin. Processes Geophys., 25, 633–648, https://doi.org/10.5194/npg-25-633-2018, 2018.

Harlim, J.: Model error in data assimilation, in: Nonlinear and stochastic climate dynamics, edited by: Franzke, C. L. E. and O'Kane, T. J., Cambridge University Press, 276–317, https://doi.org/10.1017/9781316339251.011, 2017.

Harlim, J.: Data-driven computational methods: parameter and operator estimations, Cambridge University Press, Cambridge, 2018.

Hodyss, D.: Ensemble State Estimation for Nonlinear Systems Using Polynomial Expansions in the Innovation, Mon. Weather Rev., 139, 3571–3588, https://doi.org/10.1175/2011MWR3558.1, 2011.

Hodyss, D.: Accounting for Skewness in Ensemble Data Assimilation, Mon. Weather Rev., 140, 2346–2358, https://doi.org/10.1175/MWR-D-11-00198.1, 2012.

Hsieh, W. W. and Tang, B.: Applying Neural Network Models to Prediction and Data Analysis in Meteorology and Oceanography, B. Am. Meteorol. Soc., 79, 1855–1870, https://doi.org/10.1175/1520-0477(1998)079<1855:ANNMTP>2.0.CO;2, 1998.

Janjić, T., Bormann, N., Bocquet, M., Carton, J. A., Cohn, S. E., Dance, S. L., Losa, S. N., Nichols, N. K., Potthast, R.,

Waller, J. A., and Weston, P.: On the representation error in data assimilation, Q. J. Roy. Meteor. Soc., 144, 1257–1278, https://doi.org/10.1002/qj.3130, 2018.

Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, Cambridge, 2002.

Kassam, A.-K. and Trefethen, L. N.: Fourth-Order Time-Stepping For Stiff PDEs, Siam J. Sci. Comput., 26, 1214–1233, https://doi.org/10.1137/S1064827502410633, 2005.

Kondrashov, D. and Chrekroun, M. D.: Data-adaptive harmonic spectra and multilayer Stuart-Landau models, Chaos, 27, 093110, https://doi.org/10.1016/j.physd.2014.12.005, 2017.

Kondrashov, D., Chrekroun, M. D., and Ghil, M.: Data-driven non-Markovian closure models, Physica D, 297, 33–55, https://doi.org/10.1063/1.4989400, 2015.

Kondrashov, D., Chrekroun, M. D., Yuan, X., and Ghil, M.: Data-adaptive harmonic decomposition and stochastic modeling of Arctic sea ice, in: Advances in Nonlinear Geosciences, edited by: Tsonis, A., Springer, Cham, 179–205, https://doi.org/10.1007/978-3-319-58895-7_10, 2018.

Kuramoto, Y. and Tsuzuki, T.: Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, Prog. Theor. Phys., 55, 356–369, 1976.

Lguensat, R., Tandeo, P., Ailliot, P., Pulido, M., and Fablet, R.: The Analog Data Assimilation, Mon. Weather Rev., 145, 4093–4107, https://doi.org/10.1175/MWR-D-16-0441.1, 2017.

Long, Z., Lu, Y., Ma, X., and Dong, B.: PDE-Net: Learning PDEs from Data, in: Proceedings of the 35th International Conference on Machine Learning, 2018.

Lorenz, E. N.: Deterministic nonperiodic flow, J. Atmos. Sci., 20, 130–141, 1963.

Lorenz, E. N.: Designing Chaotic Models, J. Atmos. Sci., 62, 1574–1587, https://doi.org/10.1175/JAS3430.1, 2005.

Lorenz, E. N. and Emanuel, K. A.: Optimal sites for supplementary weather observations: simulation with a small model, J. Atmos. Sci., 55, 399–414, https://doi.org/10.1175/1520-0469(1998)055<0399:OSFSWO>2.0.CO;2, 1998.

Magnusson, L. and Källén, E.: Factors influencing skill improvements in the ECMWF forecasting system, Mon. Weather Rev., 141, 3142–3153, https://doi.org/10.1175/MWR-D-12-00318.1, 2013.

Mitchell, L. and Carrassi, A.: Accounting for model error due to unresolved scales within ensemble Kalman filtering, Q. J. Roy. Meteor. Soc., 141, 1417–1428, https://doi.org/10.1175/MWR-D-16-0478.1, 2015.

Paduart, J., Lauwers, L., Swevers, J., Smolders, K., Schoukens, J., and Pintelon, R.: Identification of nonlinear systems using polynomial nonlinear state space models, Automatica, 46, 647–656, https://doi.org/10.1016/j.automatica.2010.01.001, 2010.

Park, D. C. and Zhu, Y.: Bilinear recurrent neural network, IEEE World Congress on Computational Intelligence, Neural Networks, 3, 1459–1464, 1994.

Pathak, J., Lu, Z., Hunt, B. R., Girvan, M., and Ott, E.: Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data, Chaos, 27, 121102, https://doi.org/10.1063/1.5010300, 2017.

Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E.: Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach, Phys. Rev. Lett., 120, 024102, https://doi.org/10.1103/PhysRevLett.120.024102, 2018.

Pulido, M., Tandeo, P., Bocquet, M., Carrassi, A., and Lucini, M.: Stochastic parameterization identification using ensemble Kalman filtering combined with maximum likelihood methods, Tellus A, 70, 1442099, https://doi.org/10.1080/16000870.2018.1442099, 2018.

Raanes, P. N., Carrassi, A., and Bertino, L.: Extending the square root method to account for additive forecast noise in ensemble methods, Mon. Weather Rev., 143, 3857–3873, https://doi.org/10.1175/MWR-D-14-00375.1, 2015.

Raanes, P. N., Bocquet, M., and Carrassi, A.: Adaptive covariance inflation in the ensemble Kalman filter by Gaussian scale mixtures, Q. J. Roy. Meteor. Soc., 145, 53–75, https://doi.org/10.1002/qj.3386, 2019.

Resseguier, V., Mémin, E., and Chapron, B.: Geophysical flows under location uncertainty, Part I Random transport and general models, Geophys. Astro. Fluid, 111, 149–176, https://doi.org/10.1080/03091929.2017.1310210, 2017.

Ruiz, J. J., Pulido, M., and Miyoshi, T.: Estimating model parameters with ensemble-based data assimilation: A Review, J. Meteorol. Soc. Jpn., 91, 79–99, https://doi.org/10.2151/jmsj.2013-201, 2013.

Sakov, P., Haussaire, J.-M., and Bocquet, M.: An iterative ensemble Kalman filter in presence of additive model error, Q. J. Roy. Meteor. Soc., 144, 1297–1309, https://doi.org/10.1002/qj.3213, 2018.

Sivashinsky, G. I.: Nonlinear analysis of hydrodynamic instability in laminar flames-I. Derivation of basic equations, Acta Astronaut., 4, 1177–1206, 1977.

Tandeo, P., Ailliot, P., Bocquet, M., Carrassi, A., Miyoshi, T., Pulido, M., and Zhen, Y.: Joint Estimation of Model and Observation Error Covariance Matrices in Data Assimilation: a Review, available at: https://hal-imt-atlantique.archives-ouvertes.fr/hal-01867958 (last access: 8 July 2019), submitted, 2019.

Trémolet, Y.: Accounting for an imperfect model in 4D-Var, Q. J. Roy. Meteor. Soc., 132, 2483–2504, 2006.

Wang, Y.-J. and Lin, C.-T.: Runge-Kutta neural network for identification of dynamical systems in high accuracy, IEEE T. Neural Networ., 9, 294–307, https://doi.org/10.1109/72.661124, 1998.

Weinan, E.: A proposal on machine learning via dynamical systems, Commun. Math. Stat., 5, 1–11, https://doi.org/10.1007/s40304-017-0103-z, 2017.

Whitaker, J. S. and Hamill, T. M.: Evaluating Methods to Account for System Errors in Ensemble Data Assimilation, Mon. Weather Rev., 140, 3078–3089, https://doi.org/10.1175/MWR-D-11-00276.1, 2012.